НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ
ИНСТИТУТ ТЕЛЕКОММУНИКАЦИЙ И ГЛОБАЛЬНОГО
ИНФОРМАЦИОННОГО ПРОСТРАНСТВА

На правах рукописи

ВАСЯНИН ВЛАДИМИР АЛЕКСАНДРОВИЧ

УДК 519.854:004.9:004.422.63

МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ МНОГОПРОДУКТОВЫХ КОММУНИКАЦИОННЫХ СЕТЕЙ С ДИСКРЕТНЫМИ ПОТОКАМИ

01.05.02 – математическое моделирование и вычислительные методы

Диссертация на соискание ученой степени доктора технических наук

Научный консультант – член корреспондент
НАН Украины, доктор технических наук,
профессор Трофимчук Александр Николаевич

Киев – 2017

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ …… ….. ….7
РАЗДЕЛ 1. ОСНОВНЫЕ МЕТОДОЛОГИЧЕСКИЕ ПОДХОДЫ К ПОСТРОЕНИЮ МНОГОПРОДУКТОВЫХ КОММУНИКАЦИОННЫХ СЕТЕЙ И ЗАДАЧИ ОБРАБОТКИ И РАСПРЕДЕЛЕНИЯ МЕЛКОПАРТИОННЫХ ДИСКРЕТНЫХ ПОТОКОВ КОРРЕСПОНДЕНЦИЙ……21
1.1. Общие положения …… ….21
1.2. Зонально-узловая структура транспортной сети. Принципы организации сортировки и транспортировки мелкопартионных корреспонденций. Математическая модель многопродуктовой иерархической сети ……..31
1.3. Технология сортировки мелкопартионных корреспонденций, транспортные средства для перевозки контейнеров, контейнерный терминал……..…… 38
1.4. Инфраструктура пространственных данных транспортной сети. Структурная схема информационно-аналитической системы поддержки принятия решений…… 40
Выводы по первому разделу…… 54
РАЗДЕЛ 2. ЗАДАЧИ МАРШРУТИЗАЦИИ ВНУТРИУЗЛОВЫХ ПЕРЕВОЗОК…… 55
2.1. Организация внутриузловых перевозок……... 56
2.1.1. Основные принципы и схемы перевозок……... 56
2.1.2. Функции затрат и содержательная постановка задачи……. 58
2.2. Классификация задач маршрутизации, краткий обзор методов и алгоритмов их решения…… 63
2.3. Математические модели перевозок мелкопартионных грузов во внутренней зоне магистрального узла…… 77
2.3.1. Формулировка задачи построения доставочных маршрутов (the Delivery Route, DR) для перевозки грузов из магистрального узла к клиентам. Однопродуктовая модель1……..…. 80
2.3.2. Формулировка задачи построения доставочных маршрутов для

перевозки грузов из магистрального узла к клиентам. Однопродуктовая модель 2……. 84
2.3.3. Формулировка задачи построения доставочных маршрутов для перевозки грузов из магистрального узла к клиентам. Двухпродуктовая модель. ……. 85
2.3.4. Формулировка задачи построения доставочных маршрутов для перевозки грузов из магистрального узла к клиентам. Модели на разбиении множества…….. 87
2.3.5. Формулировка задачи построения сборочных маршрутов (the Collection Route, CR) для перевозки грузов от клиентов в магистральный узел…….. 91
2.3.6. Формулировка задачи построения комбинированных (комплексных доставочно-сборочных) маршрутов (the Delivery and Collection Route, DCR) и обзор методов ее решения…… 92
2.3.7. Формулировка задачи построения маршрутов с расщепленной доставкой и неоднородным парком транспортных средств……... 102
2.3.8. Задача построения доставочных маршрутов с неоднородным парком транспортных средств, размещением депо и временными окнами…….. 105
Выводы по второму разделу …… 107
РАЗДЕЛ 3. ЗАДАЧИ УПАКОВКИ И ВЫБОРА СТРУКТУРЫ МАГИСТРАЛЬНОЙ СЕТИ…….. 110
3.1. Справочная матрица объединения потоков……... 112
3.1.1. Постановка задачи ……..……... 112
3.1.2. Основные результаты и алгоритм решения……....…… 114
3.2. Задача упаковки без функций затрат…… …120
3.2.1. Постановка задачи…… ..121
3.2.2. NP-полнота задачи…….. 123
3.2.3. Численный эксперимент…… 127
3.3. Задача упаковки с функциями затрат и алгоритмы ее решения…….. 134
3.3.1. Постановка задачи и особенности ее решения…… 135

3.3.2. Алгоритмы решения задачи…….. 140
3.3.3. Вычислительный эксперимент……. 150
3.4. Задача упаковки с ограничениями на пропускные способности дуг….. 155
3.4.1. Постановка и математическая модель задачи……. ...156
3.4.2. Вычислительный эксперимент…….…… ……...163
3.4.3. Задача выбора пропускных способностей дуг…… ….167
3.4.3.1. Алгоритмы решения задачи выбора пропускных способностей. 168
3.4.3.2. Аппроксимационный алгоритм……..…… ….172
3.4.3.3. Алгоритм, основанный на методе последовательного анализа и отсеивания вариантов……..…….. 174
3.4.3.4. Экспериментальное сравнение алгоритмов… ……... 177
3.5. Задача выбора структуры сети……. . ….178
3.5.1. Постановка и общая схема решения задачи…… 179
3.5.2. Экспериментальные исследования решения задачи…….…. 181
Выводы по третьему разделу……... 191
РАЗДЕЛ 4. ЗАДАЧИ ПОСТРОЕНИЯ МАРШРУТОВ В МАГИСТРАЛЬНОЙ СЕТИ……. 194
4.1. Общая задача распределения и маршрутизации потоков транспортных блоков со смешанными вложениями и нелинейными затратами…… 196
4.1.1. Математическая модель……... 197
4.1.2. Особенности задачи……. 201
4.1.3. Декомпозиция задачи к совокупности задач о ранце…… 206
4.2. Задача распределения и маршрутизации потоков с заданными тарифами в узлах и на дугах……. 211
4.2.1. Постановка и математическая модель задачи……. 212
4.2.2. Полиномиальное преобразование задачи с тарифами к задаче целочисленного программирования……. 214
4.3. Задачи построения кратчайших путей……... 221
4.3.1. Постановка задачи и алгоритмы решения……... 221
4.3.2. Экспериментальное исследование алгоритмов……... 234

4.4. Задачи построения остовных деревьев……... 237
4.4.1. Постановка задачи, обзор методов и алгоритмов ее решения……... 237
4.4.2. Алгоритмы построения MST-деревьев…… 244
4.4.3. Экспериментальное исследование алгоритмов……... 261
4.5. Алгоритмы распределения и маршрутизации потоков…… 272
4.6. Численный пример решения задачи распределения и маршрутизации потоков……290
Выводы по четвертому разделу…… 294
РАЗДЕЛ 5. ЗАДАЧИ ТЕКУЩЕГО ПЛАНИРОВАНИЯ КАК МЕТОДОЛОГИЧЕСКАЯ ОСНОВА — ЯДРО ДЛЯ ПЕРСПЕКТИВНОГО РАЗВИТИЯ УЗЛОВ И ТРАНСПОРТНЫХ МАРШРУТОВ И ОПЕРАТИВНОГО УПРАВЛЕНИЯ РАСПРЕДЕЛЕНИЕМ ПОТОКОВ В МАГИСТРАЛЬНОЙ СЕТИ……... 297
5.1. Введение в постановку задачи и основные постулаты…….. 298
5.2. Методика получения исходных данных для построения динамических моделей поэтапного развития узлов и транспортных маршрутов магистральной сети. Схема взаимодействия задач текущего планирования и оперативного управления……..... 302
Выводы по пятому разделу…… 308
РАЗДЕЛ 6. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ РЕШЕНИЯ ЗАДАЧ ТЕКУЩЕГО ПЛАНИРОВАНИЯ В МАГИСТРАЛЬНОЙ СЕТИ…… 310
6.1. Методика решения задач текущего планирования……... 310
6. 2. Компьютерная программа решения задачи упаковки и выбора структуры сети ……... 318
6.3. Компьютерная программа решения задачи распределения и маршрутизации потоков ……... 326
6.4. Обобщенный пример проектирования автотранспортной сети……... 335
Выводы по шестому разделу …… 340
ЗАКЛЮЧЕНИЕ…….. 342
СПИСОК ЛИТЕРАТУРЫ..…… 347

Приложение А. Типовые задания по разработке информационного обеспечения и автоматизации процессов поддержки принятия решений…… 393
Приложение Б. Задача распределения и упаковки мелкопартионных потоков корреспонденций в зональных сетях…… 398
Б.1. Постановка задачи…….……..……. …399
Б.2. Математическая модель и алгоритмы решения задачи……. 400
Приложение В. Задача развозки порожних контейнеров……... 409
В.1. Постановка и математическая модель задачи…… 409
В.2. Алгоритм и методы решения задачи……... 415
В.3. Вычислительный эксперимент…… 427
Приложение Г. Обзор методов построения кратчайших путей……. 430
Приложение Д. Структуры данных и процедуры редукции маршрутов в задачах распределения потоков……. 447
Д.1. Постановка задачи и алгоритм решения……..…… 447
Д.2. Пример формирования справочной структуры… ……... 458
Д.3. Результаты вычислительного эксперимента……..…….. .461
Приложение Е. Вычислительный эксперимент к пятому разделу……... 464
Приложение Ж. Акты о внедрении и использовании результатов диссертационной работы……... 483
Приложение З. Копии свидетельств про авторское право……. 488

ВВЕДЕНИЕ

Актуальность темы. Одними из приоритетных областей развития экономики Украины определены транспорт и связь повышение эффективности функционирования которых требует создания прикладного математического обеспечения для решения задач оптимизации транспортных процессов и информационно-аналитических систем поддержки принятия решений (ИАС ППР) для управления обработкой и распределением транспортных потоков. В связи с важностью рассматриваемой проблематики для интенсификации рыночных преобразований в Украине и, в частности, в сфере перевозок мелкопартионных грузов, в диссертационной работе исследуются задачи математического моделирования и проектирования многопродуктовых коммуникационных сетей с дискретными потоками мелкопартионных корреспонденций. Существующие и проектируемые территориально-распределенные коммуникационные сети, как правило, имеют многоуровневую структуру, где число уровней иерархии определяется административным делением территории, подчинением органов территориального управления, принятой технологией обработки и распределения потоков грузов и информации и др. В большинстве случаев они состоят из децентрализованной распределенной сети верхнего уровня — магистральной, и низовых сетей — зональных и внутренних.

Проектирование многоуровневых сетей должно отражать характерную для сложных систем невозможность полной централизации в одном звене обработки информации и принятия решений по управлению процессами их развития и функционирования. Это приводит к необходимости формирования иерархической структуры системы автоматизированного управления ресурсами и распределением потоков в сети. Проектирование иерархических сетевых структур обычно имеет нисходящий характер. На верхнем уровне решаются задачи структурного синтеза и перспективного развития сети, для которых используются крупноагрегированные модели распределения потоков. На низших уровнях проектирования детализация объектов должна увеличиваться с целью наиболее адекватного описания их функционирования и принятия рациональных

решений. Это обуславливает итерационно-циклический характер процессов проектирования и управления, включающих процедуры синтеза и анализа возможных решений на всех уровнях сети. Поскольку решения принимаются в условиях неопределенности, связанных с неполнотой имеющейся информации и колебаниями потоков во времени, а также с огрублением математических моделей, необходимо разделять решения на перспективные, текущие и оперативные. В этой связи представляется актуальной разработка комплекса взаимосвязанных многоуровневых моделей перспективного развития, текущего планирования и оперативного управления, отображающих иерархию сети и соответствующую ей степень агрегирования показателей.

В теории проектирования крупномасштабных сетевых структур, решения задач структурного анализа, синтеза и оптимизации, управления распределением потоков в многопродуктовых сетях, наукой накоплен значительный теоретический материал и практический опыт. Причем весомый вклад в развитие моделей и методов управления сетевыми ресурсами сделали зарубежные и отечественные ученые, среди которых, в первую очередь, следует выделить таких как Green P., Lucky R., Kleinrock L., Gallager R., Ford L.R., Fulkerson D.R., Hu T., Tomlin J., Assad A.A., Kennington J.L., Fratta L., Gerla M., Castro J., Nabona N., Frangioni A., Gratzer F.J., Steiglitz K., Takahara Y., Ahuja R.K., Magnanti T.L., Orlin J.B., Steenbrink P.A., Vutukury S., Моисеев Н.Н., Краснощеков П.С., Вишневский В.М., Олифер В.Г., Олифер Н.А., Малашенко Ю.Е., Новикова Н.М., Бусленко Н.П., Гнеденко Б.В., Глушков В.М., Михалевич В.С., Сергиенко И.В., Зайченко Ю.П. и многие другие. Как правило, в работах этих авторов рассматриваются детерминированные модели и модели, основанные на математических методах теорий массового обслуживания и случайных марковских процессов. При этом фактически предполагается либо детерминированность, либо стационарность потоков на отдельных промежутках времени распределения ресурсов. Реальные потоки этих свойств не имеют. Кроме того, в большинстве случаев, потоки и параметры элементов сети являются дискретными величинами. Математический аппарат для моделирования и анализа функционирования

многопродуктовых сетей с дискретными потоками развит недостаточно. Необходима разработка математических моделей и методов решения, учитывающих немарковский характер случайных процессов, протекающих в многопродуктовых сетях, неоднородность, нестационарность и дискретность распределяемых потоков. В большинстве из известных работ, непосредственно посвященных распределению многопродуктовых потоков и связанных с задачами анализа и синтеза многопродуктовых сетей предлагаются линейные и нелинейные модели и соответствующие методы решения в основном для непрерывных переменных и параметров элементов сети. Кроме того, рассматриваемые модели являются, как правило, довольно грубыми математическими описаниями функционирования сложной сетевой структуры и не учитывают целый ряд физических характеристик и параметров, присущих реальным системам (т.е. являются сильно агрегированными).

Вместе с тем, в случае распределения мелкопартионных потоков в многопродуктовой сети, сама постановка общей проблемы и отдельных задач управления сетевыми ресурсами существенно изменяется из-за необходимости учета ряда общих, одновременно действующих факторов. К ним, в первую очередь, следует отнести: организацию процессов сортировки мелкопартионных потоков корреспонденций в узлах сети, которые непосредственно связаны с их транспортировкой в узлы назначения; ограничения на время доставки корреспонденций конечным получателям; ограничения на пропускные способности транспортных линий связи (грузоподъемность транспортных средств, пропускную способность каналов связи); ограничения на объемы обработки транспортируемых единиц (транспортных блоков, контейнеров) в промежуточных пунктах линий связи; нелинейность приведенных (эксплуатационных и капитальных) затрат на обработку и транспортировку потоков.

Поэтому актуальной остается проблема разработки методологии моделирования и проектирования многопродуктовых коммуникационных сетей, обобщающей полученные ранее результаты и позволяющей с использованием моделей различной степени агрегирования и для всех уровней иерархии успеш-

но решать практические задачи перспективного развития, текущего планирования и оперативного управления для сетей большой размерности с дискретными потоками и дискретными параметрами элементов сети.

Связь работы с научными программами, планами, темами.

Разработанная в диссертации методология математического моделирования функционирования и развития многопродуктовых сетей с дискретными потоками является дальнейшим продолжением исследований, начатых автором в кандидатской диссертации. Все математические модели, методы и алгоритмы, рассмотренные в кандидатской диссертации, получили дальнейшее развитие и для них достигнуты новые научные результаты.

Результаты, включенные в диссертационную работу, получены при выполнении научно-исследовательских работ по созданию инструментария и информационной платформы (портала) для автоматизации процессов принятия решений в управлении развитием и функционированием многопродуктовых коммуникационных сетей с дискретными потоками. Работы проводились Институтом телекоммуникаций и глобального информационного пространства НАН Украины в рамках научных работ по ведомственной тематике «Розробка індикативних методів моделювання та моніторингу процесів технологічного розвитку в Україні» (2013-2015 рр., № ДР 0112U007445), «Розробка нових програмних засобів і технологій пошуку, впорядкування, обробки та представлення інформації у інформаційному просторі» (2014-2016 рр., № ДР 0114U000798), «Розробка програмно-технічних комплексів взаємодії новітніх телекомунікаційних технологій з мережами загального користування» (2014-2016 рр., № ДР 0114U004983), одним из исполнителей которых являлся соискатель.

Цель и задачи исследования. Повышение эффективности функционирования многопродуктовых транспортных сетей за счет снижения дефицитных материальных, сырьевых, энергетических, финансовых и трудовых затрат на основе предложенной методологии моделирования и проектирования процессов обработки и распределения мелкопартионных дискретных потоков и комплекса мер информационно-аналитического обеспечения и автоматизации про-

цедур принятия решений в управлении транспортными потоками. Для достижения намеченной цели в диссертационной работе были поставлены следующие основные задачи.

1. Всесторонне исследовать и определить структуру и свойства многопродуктовых сетей с мелкопартионными потоками, критерии и стоимостные функции обработки и распределения потоков.

2. Провести анализ современного состояния и проблем организации распределения мелкопартионных потоков в транспортных системах и сетях передачи данных.

3. Разработать теоретико-методологические основы построения многоуровневой структуры многопродуктовой сети и ее математическую модель. Определить принципы и концепции обработки и распределения мелкопартионных потоков в многоуровневой (зонально-узловой) структуре многопродуктовой сети.

4. Провести обзор и анализ развития теории и практики проектирования территориально-распределенных коммуникационных сетей с непрерывными и дискретными потоками, математических моделей и методов решения задач обработки, распределения и маршрутизации потоков на всех уровнях иерархической сети. Обосновать выбор приближенных и эвристических методов для решения задач моделирования и проектирования многопродуктовой сети с мелкопартионными дискретными потоками.

5. Провести обзор и анализ современных информационных платформ и инструментария для построения ИАС ППР для управления сложными территориально-распределенными сетями. Определить основные задачи ИАС ППР и разработать методологию проектирования структуры системы управления процессами обработки и распределения мелкопартионных дискретных потоков в многопродуктовых коммуникационных сетях.

6. Провести обзор и анализ основных моделей, методов и алгоритмов для решения отдельных задач текущего планирования, построения кратчайших путей и остовных деревьев.

7. Разработать методологию математического моделирования процессов обработки и распределения мелкопартионных дискретных потоков на магистральном, зональном и внутреннем уровнях многопродуктовой сети для периодов текущего планирования функционирования сети. Разработать математические модели, методы и алгоритмы решения задач текущего планирования. Экспериментально исследовать решения задач на чувствительность, провести параметрический и постоптимальный численный анализ решений, определить временную сложность алгоритмов решения задач.

8. Определить основные задачи перспективного развития и оперативного управления на магистральном уровне сети. Показать взаимосвязь их решения с задачами текущего планирования в случаях предельного колебания нагрузок, перегрузок, отказов узлов и линий связи и возникновения других внештатных ситуаций. Разработать методику и схему принятия решений по стабилизации работы магистральной сети в штатном режиме.

9. Разработать методику решения комплекса задач текущего планирования в среде ИАС ППР и компьютерные программы для решения задач выбора структуры магистральной сети и схем сортировки, распределения и маршрутизации потоков. Исследовать разработанные методы, алгоритмы и программы на числовых примерах, провести анализ результатов цифрового моделирования решения задач на сетях большой размерности.

Объект исследования — процессы обработки и распределения мелкопартионных дискретных потоков в многопродуктовых сетях.

Предмет исследования — методология математического моделирования и проектирования многопродуктовых коммуникационных сетей с мелкопартионными дискретными потоками, включающая понятийный базис, концепции, принципы, математические модели, теоретические методы, алгоритмы, технологии, методики, схемы и сценарии, нацеленные на эффективное использование ресурсного потенциала сетей при заданных ограничениях.

Методы исследования. Основой научных исследований в диссертационной работе являлись общая методология системного анализа и моделирования

сложных систем, теория распределения потоков в многопродуктовых сетях, методы исследования операций и математического программирования, комбинаторный и эвристический анализ, теория графов, теория множеств, методология построения информационно-аналитических систем поддержки принятия решений.

Научная новизна полученных результатов. В диссертационной работе получены следующие основные результаты, имеющие научную новизну:

1. Впервые предложены теоретико-методологические основы построения распределенной иерархической многопродуктовой коммуникационной сети с мелкопартионными дискретными потоками, включающие: понятийный базис; принципы, концепции и критерии обработки и распределения мелкопартионных потоков, методологию синтеза и анализа многоуровневой структуры сети. В отличие от существующих подходов, предложенная методология позволяет логически увязать процессы обработки мелкопартионных потоков в узлах сети с их последующим распределением по сети и доставкой получателям и повысить экономическую эффективность функционирования и развития транспортной системы в целом за счет снижения приведенных затрат как в узлах так и на транспортных магистралях сети при заданных ограничениях.

2. Впервые для управления нестационарными процессами обработки и распределения мелкопартионных дискретных потоков в многопродуктовой коммуникационной сети предложены основанные на комплексном решении задач текущего планирования, оперативного управления и перспективного развития концепция и методология построения многоуровневой ИАС ППР, которая функционирует в масштабе реального времени, что открывает новые направления для создания интеллектуальных методов и методик в управлении сложными процессами и объектами в транспортных системах.

3. Усовершенствованы типовые математические модели задач построения доставочных, сборочных и комбинированных маршрутов транспортных средств, которые отличаются от известных моделей большей адекватностью реальным процессам перевозки мелкопартионных грузов во внутренней зоне

магистрального узла.

4. Получила дальнейшее развитие ранее предложенная автором математическая модель задачи упаковки мелкопартионных корреспонденций в транспортные блоки, в которой учитываются новые ограничения на количество объединений корреспонденций, время их доставки конечному получателю, пропускные способности дуг и среднее время задержки потоков в сети. Впервые доказана NP-трудность задачи, разработаны эвристические методы и алгоритмы решения задачи и доказана их сходимость к локальному оптимуму. Получены оценки временной сложности алгоритмов. Доказано, что решение задачи упаковки с дополнительными ограничениями на пропускные способности дуг и среднее время задержки потоков в сети можно свести к последовательному решению задачи упаковки без указанных ограничений и задачи выбора пропускных способностей дуг.

5. Впервые предложен эвристический метод решения задачи выбора иерархической структуры многопродуктовой сети в соответствии с принципами, концепциями и критериями, заложенными в основу обработки и распределения мелкопартионных дискретных потоков, который отличается от существующих своей оригинальностью и позволяет проектировщику сети проводить числовое моделирование, анализ и выбор структуры сети при изменении различных начальных данных, параметров и ограничений в интерактивном режиме.

6. Получила дальнейшее развитие ранее предложенная автором математическая модель задачи распределения и маршрутизации потоков транспортных блоков в многопродуктовой магистральной сети. Разработана совокупность новых методов и алгоритмов для решения задачи при ограничениях на пропускные способности узлов и транспортных линий связи, объемы обработки транспортных блоков в транзитных узлах линий связи, среднее время задержки потоков в сети и время доставки мелкопартионных корреспонденций получателю. Получены асимптотические оценки временной сложности алгоритмов. Предложенные методы и алгоритмы, в отличие от известных подходов к решению задач распределения целочисленных потоков в многопродуктовой сети, опира-

ются на эвристические приемы решения системы многомерных задач о ранце со связывающими ограничениями и на эффективные способы представления структур данных, которые генерируются в процессе решения задачи, и позволяют проектировать новые маршруты транспортировки для разветвленных и неразветвленных потоков.

7. Впервые всесторонне исследована группа задач текущего планирования, теоретически и экспериментально обосновано их использование как методологической основы — ядра для решения задач перспективного развития многопродуктовой сети и оперативного управления процессами обработки и транспортировки потоков.

8. Для отдельных подзадач, возникающих при решении основных задач текущего планирования, разработан новый лексикографический двухкритериальный алгоритм построения кратчайших путей, а также предложена эффективная реализация алгоритма Краскала для нахождения минимального остовного леса графа или сети. По сравнению с алгоритмом Флойда-Уоршелла, модифицированным для нахождения двухкритериальных кратчайших путей между всеми n узлами сети с временной сложностью $O(n^3)$, предложенный алгоритм всегда имеет меньшую трудоемкость и с увеличением плотности сети быстрее первого в несколько раз. Временная сложность усовершенствованной версии алгоритма Краскала практически доведена до порядка $O(e)$, где e — число ребер графа. Получена оценка сравнимости быстродействия предложенного алгоритма с алгоритмом Прима, из которой следует, что для связных графов предложенный алгоритм лучше алгоритма Прима при $e \le 0{,}27n^2$, где n — число вершин графа.

Обоснованность научных положений достигнута путем системного подхода к рассмотрению исследуемой проблемы обработки и распределения мелкопартионных дискретных потоков в многопродуктовых сетях, достаточно полного учета исходных данных, параметров и ограничений, характерных для функционирования реальных распределенных сетей, сочетания формальных и неформальных методов исследования.

Достоверность научных положений обеспечена: использованием апробированных теоретических положений и методов системного анализа и проектирования сложных систем, методов исследования операций; статистической достаточностью исходного объема анализируемых литературных источников; непротиворечивостью известных и предложенных в работе теоретических, практических и нормативных положений, принципов, концепций и закономерностей; математической корректностью поставленных задач и сходимостью теоретических предположений с результатами экспериментальных исследований; многочисленными результатами цифрового моделирования разработанных в рамках диссертационной работы новых методов и алгоритмов для решения поставленных задач на сетях большой размерности (до 1000 и более узлов).

Теоретическая значимость диссертационной работы состоит в разработке теоретико-методологических основ математического моделирования и проектирования распределенной иерархической многопродуктовой коммуникационной сети с мелкопартионными дискретными потоками, включающих: понятийный базис; принципы, концепции и критерии обработки и распределения мелкопартионных потоков, методологию синтеза и анализа многоуровневой структуры сети и методологию построения многоуровневой ИАС ППР, базирующейся на решении задач текущего планирования, оперативного управления и перспективного развития и функционирующей в масштабе реального времени. В совокупности, предложенная методология, математический и программный инструментарий, а также комплекс мер информационно-аналитического обеспечения и автоматизации процедур принятия решений в управлении транспортными потоками представляют новое системно изложенное, научно-обоснованное решение крупной научно-прикладной проблемы, имеющей важное народнохозяйственное значение для повышения эффективности функционирования многопродуктовых транспортных сетей за счет снижения дефицитных материальных, сырьевых, энергетических, финансовых и трудовых затрат.

Практическое значение полученных результатов. Практическая значимость полученных в диссертационной работе результатов заключается в дове-

дении их до уровня практической реализации — создания макетного образца комплекса программ ИАС ППР для управления процессами обработки и транспортировки мелкопартионных грузов в многопродуктовой транспортной сети в реальном времени. Внедрение предложенной методологии и разработка на ее основе промышленных образцов ИАС ППР будет способствовать дальнейшему развитию транспортной отрасли Украины, созданию благоприятных условий для повышения конкурентоспособности экономики страны на внешнем и внутреннем рынках, улучшению качества обслуживания предприятий и населения транспортными услугами.

Отдельные результаты диссертации могут с успехом применяться и к другим сетевым структурам — сетям передачи данных, сетям сотовой связи и др. Так, например, похожая технология виртуальных контейнеров, при которой в один виртуальный контейнер могут объединяться сообщения с разными адресами назначения, применяется при построении перспективных крупномасштабных общенациональных и интернациональных сетей передачи данных на основе сверхширокополосных каналов и схем типа опорной сети. В настоящее время при проектировании таких сетей используются метод коммутации каналов, европейская технология синхронной цифровой иерархии (Synchronous Digital Hierarchy, SDH) или американская технология синхронных оптических сетей (Synchronous Optical NETs, SONET) с применением оптоволоконных линий святи и спектрального уплотнения каналов WDM (wavelength-division multiplexing), DWDM (dense WDM), HDWDM (high dense WDM), позволяющих одновременно передавать несколько информационных каналов по одному оптическому волокну на разных несущих частотах и мультиплексировать от 18 до 64 и более спектральных каналов с пропускной способностью от 40 и более Гбит/с. Поэтому математические модели задач обработки и транспортировки мелкопартионных потоков корреспонденций, а также предложенные методы и алгоритмы их решения могут быть использованы и для проектирования магистральных сетей передачи данных с технологией виртуальных контейнеров.

Важной особенностью предложенной и реализованной в работе методоло-

гии является также то, что она позволяет проводить математическое моделирование и оптимизацию функционирования не только для «чисто» многопродуктовых сетей, в которых каждый узел сети обменивается мелкопартионными грузами со всеми остальными узлами, но и для традиционных логистических производственных и транспортно-складских сетевых систем, включающих узлы поставщиков сырья, производства товаров, склады и конечных потребителей.

Внедрение результатов диссертационной работы в деятельность транспортных предприятий, занимающихся перевозкой различной мелкопартионной продукции, дает возможность научно обоснованного решения важных для транспортных систем задач: рассчитать основные технико-эксплуатацион-ные и экономические показатели функционирования транспортной сети; повысить эффективность функционирования сети в реальном масштабе времени за счет оптимизации схем обработки и перевозки грузов и снижения приведенных затрат на ее создание и эксплуатацию; оперативно управлять процессами обработки и перевозки грузов в случаях возникновения внештатных ситуаций за счет своевременного их обнаружения и перераспределения потоков; планировать поэтапное развитие сети с учетом коммерческого риска вложения инвестиций в условиях нестабильности экономики и девальвации ресурсов; снизить затраты на проектирование сети и с минимальными затратами труда моделировать различные варианты транспортной сети, в интерактивном режиме оптимизации изменяя топологию, иерархическую структуру, потоки, маршруты транспортных средств, всевозможные параметры и ограничения транспортной модели и из семейства полученных результатов выбирать наилучший вариант с учетом выбранной функции цели и принятых ограничений.

Кроме того, полученные результаты работы могут быть использованы в учебном процессе при подготовке специалистов в области разработки и проектирования сложных распределенных многопродуктовых сетей с дискретными потоками.

Внедрение результатов. Результаты диссертационных исследований использованы и внедрены в нескольких частных международных транспортных компаниях и в учебном процессе факультета «Инженерии и транспорта» Академии транспорта, информатики и коммуникаций Республики Молдова и факультета информационно-компьютерных технологий Житомирского государственного технологического университета, что подтверждается соответствующими документами, имеющимися в диссертационной работе.

Личный вклад соискателя. Все основные научные результаты, изложенные в диссертации, получены автором лично. В научных работах, опубликованных в соавторстве [12, 13, 15,16, 21, 24, 26, 32, 118, 119, 125, 127, 129, 131, 462], автору диссертации принадлежит: определение понятийного базиса, принципов, концепций и критериев обработки и распределения мелкопартионных потоков, методология синтеза и анализа многоуровневой структуры сети и ИАС ППР, формулировки математических постановок, доказательство утверждений, разработка методов, алгоритмов, абстрактных структур данных и компьютерных технологий решения поставленных задач.

Апробация результатов диссертации. Основные результаты диссертации докладывались и прошли апробацию на: VII и VIII Международных научно-технических конференциях «Інформаційно-комп'ютерні технології» (Житомир, 2014, 2016); Conferinţe Internaţionale «Modelare matematică, optimizare şi tehnologii informaţionale» (Chişinău, 2014, 2016); Conferinţe Internaţionale «Sisteme de transport şi logistică» (Chişinău, 2013, 2015); 14 Международной научно-практической конференции «Сучасні інформаційні технології управління екологічною безпекою, природокористуванням, заходами в надзвичайних ситуаціях» (Киев, Пуща-Водица, 2015); V and VI International Conferences on Optimization Methods and Applications «Optimization and applications» — OPTIMA-2014, 2015, (Petrovac, Montenegro, 2014, 2015); VI Всеукраинской научно-практической конференции с международным участием «Інформатика та системні науки» — ІСН-2015, (Полтава, 2015); VIII Всероссийской научной конференции с международным участием «Математическое моделирование разви-

вающейся экономики, экологии и технологий (ЭКОМОД-2014)», посвященной 80-летию академика А.А. Петрова и 100-летию академика Г.С. Поспелова (Москва, ВЦ РАН, 2014); International Scientific Practical Conferences «Information Control Systems and Technologies» (ICST-ODESSA-2013-2014), (Одесса, 2013, 2014); 15-th and 16-th International conferences on System Analysis and Information Technologies (SAIT2013, SAIT2014), (Киев, 2013, 2014); XI Международной научно-практической конференции «Математичне та програмне забезпечення інтелектуальних систем» (MPZIS-2013), (Днепропетровск 2013); научной конференции «Информационные технологии в управлении сложными системами – 2013» (Днепропетровск, 2013); Международной научной конференции «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту» (Евпатория, 2008).

Кроме того результаты диссертации докладывались на научных семинарах в Институте телекоммуникаций и глобального информационного пространства НАН Украины и Институте кибернетики имени В.М. Глушкова НАН Украины.

Публикации. Всего по тематике диссертации опубликовано 51 научная работа, в том числе: 21 статья из перечня научных специализированных изданий Украины (из них 5 в изданиях Украины, включенных в международные наукометрические базы); 2 статьи в зарубежных научных периодических изданиях; 23 в сборниках материалов научно-технических конференций (из них 9 в зарубежных изданиях); получено 5 свидетельств про регистрацию авторского права на программный продукт.

Структура и объем диссертации. Диссертация состоит из введения, шести разделов и заключения, изложенных на 326 страницах основного текста и 20 страницах рисунков и таблиц и включает восемь приложений на 105 страницах. Всего работа содержит 66 рисунков, 31 таблицу и список литературы из 493 наименований на 46 страницах. Общий объем диссертации занимает 497 страниц.

РАЗДЕЛ 1. ОСНОВНЫЕ МЕТОДОЛОГИЧЕСКИЕ ПОДХОДЫ К ПОСТРОЕНИЮ МНОГОПРОДУКТОВЫХ КОММУНИКАЦИОННЫХ СЕТЕЙ И ЗАДАЧИ ОБРАБОТКИ И РАСПРЕДЕЛЕНИЯ МЕЛКОПАРТИОННЫХ ДИСКРЕТНЫХ ПОТОКОВ КОРРЕСПОНДЕНЦИЙ

В разделе рассматриваются методологические подходы к построению многопродуктовых иерархических коммуникационных сетей и определены основные задачи обработки и распределения мелкопартионных дискретных потоков корреспонденций, позволяющие создать благоприятные условия для сокращения материальных, финансовых и трудовых затрат в транспортных системах при дальнейшей механизации и автоматизации производства. Приводятся некоторые аспекты создания и внедрения инфраструктуры пространственных данных, геоинформационной системы и специального математического обеспечения для транспортных сетей и сетей передачи данных. В приложении А1 определены основные задания по разработке комплекса мер информационного обеспечения и автоматизации процессов принятия решений для сетевых структур подобного класса.

Основные материалы и результаты раздела опубликованы в работах [12-14] и докладывались на конференциях [112-116, 460].

Ключевые слова: иерархические многопродуктовые коммуникационные сети, мелкопартионные потоки корреспонденций, принципы и технология обработки и транспортировки потоков в зонально-узловой структуре сети, задачи долгосрочного и текущего планирования и оперативного управления, информационно-аналитическая система поддержки принятия решений

1.1. Общие положения

Современный период развития социально-экономических процессов в Украине характеризуется их динамикой и сложностью, совершенствованием структуры управления органов государственной власти, трансформацией сложив-

шихся традиционных и появлением новых информационных общественных отношений, обуславливающих переход общества от индустриального к информационному. Это создает необходимость повышения роли информации и новых информационных технологий, формирования единого информационного пространства страны, совершенствования информационного обеспечения деятельности органов государственной власти и хозяйствующих субъектов.

В соответствии с Законом Украины «О Концепции Национальной программы информатизации» (от 4 февраля 1998 года № 75/98-ВР, текущая редакция от 11 августа 2013 года) информационные технологии определены как составные части национальной информационной инфраструктуры и призваны обеспечить экономический подъем страны. Осуществление процесса информатизации в период становления Украины как независимого государства в условиях кризисного состояния экономики и ограниченности ресурсов возможно лишь путем определения приоритетных сфер и направлений с концентрацией на них финансовых, материальных и трудовых ресурсов. В Национальной программе информатизации одной из приоритетных областей экономики определен транспорт, повышение эффективности функционирования которого требует создания комплекса автоматизированных систем обработки данных и управления разного уровня и назначения. Поэтому автоматизация информационных процессов в управлении транспортными потоками в корпоративных и государственных транспортных сетях, является перспективным направлением в достижении качественно нового уровня управления транспортом и интеграции Украины в Европейское Сообщество. В связи с важностью транспортного комплекса для интенсификации рыночных преобразований в Украине, предлагаемая в диссертации методология анализа и проектирования коммуникационных сетей в основном направлена на решение задач обеспечения нормального функционирования транспортных систем в инфраструктурных комплексах крупных городов, регионов и страны в целом. Однако она может с успехом применяться и к другим сетевым структурам — сетям передачи данных, сетям сотовой и почтовой связи и др.

В большинстве случаев существующие и проектируемые территориально-распределенные коммуникационные сети — транспортные, информационно-вычислительные, топливно-энергетические, почтовые, телеграфные, телефонные и пр. являются многоуровневыми и состоят из децентрализованной распределенной сети (магистральной) и низовых фрагментарных сетей (зональных и внутренних) на нижних уровнях иерархии. В данной работе рассматриваются многопродуктовые транспортные сети (ТС) и сети передачи данных (СПД), для которых характерно наличие множества источников и стоков потоков мелкопартионных корреспонденций (продуктов или требований). Под корреспонденцией (продуктом, требованием) понимается пара различных узлов сети, между которыми имеется направленный (адресный) дискретный поток элементов (например, неделимых грузов унифицированного размера, бит или символов) заданной величины. В многопродуктовой сети все потоки корреспонденций подлежат единовременной передаче из источников в стоки. В общем случае на сети может быть задано некоторое множество видов (категорий) корреспонденций, отличающихся весом, габаритами и другими характеристиками, но имеющих общие источники и стоки.

Эффективность работы коммуникационных сетей во многом определяет экономические и социальные показатели функционирования инфраструктуры большинства хозяйствующих субъектов. Стабилизация и совершенствование экономики страны, особенно в условиях мирового экономического кризиса, требуют дальнейшего развития и модернизации сетевых структур. Для того чтобы возможные инвестиции, выделенные на развитие инфраструктур, были использованы наиболее эффективно, нужен строгий технико-экономический расчет, охватывающий основные альтернативные схемы их развития и функционирования. Необходимо выполнить оценку и отбор различных вариантов совершенствования инфраструктур на основе процедур оптимизации и системного анализа эффектов, возникающих внутри связей инфраструктур, а также вне их, опосредованно влияющих на развитие и состояние экономики отдельных территориальных регионов и страны в целом. Решение задач перспективного

развития инфраструктуры позволит поэтапно, с учетом величины полученных инвестиций, осуществить формирование оптимизированной структуры объектов и связей транспортной сети.

В настоящее время для существующих сетевых инфраструктур в различных отраслях хозяйства, характерно то, что на всех уровнях управления уже введены различные автоматизированные и информационные системы. Предусматривается дальнейшее их развитие с использованием новейших информационных технологий; современного методического, технического, и математического обеспечений; систем поддержки принятия решений, рационально сочетающих формальные и неформальные методы принятия решений и интерактивный режим анализа и выбора оптимальных решений. Сложилась и структура информационных и вычислительных сетей, обеспечивающих функционирование коммуникационных систем, включающая глобальные, корпоративные и локальные сети передачи данных, аппаратуру коммутации и передачи данных и т.д.

Поскольку физическая пространственная структура большинства существующих сетей уже сложилась, в первую очередь, наибольший интерес представляет решение задач текущего планирования и оперативного управления, нацеленных главным образом на оптимизацию их функционирования при имеющихся ресурсах.

В рамках проектов автоматизации процессов принятия решений по оптимальному развитию и функционированию иерархических коммуникационных сетей должны быть рассмотрены:

• математические модели, методы и алгоритмы решения задач оптимизации многоэтапного управления развитием инфраструктуры таких сетей с учетом всех возможностей их качественного организационно-технического совершенствования;

• математические модели, методы и алгоритмы решения задач оптимального функционирования сетевых структур на уровнях текущего планирования и оперативного управления;

• методы анализа решений с использованием баз знаний и привлечением профессиональных экспертов, позволяющие принимать решения в интерактивном режиме для указанных выше задач;

• информационное обеспечение, базирующееся на самых последних новациях в области создания баз данных и систем управления базами данных.

На современном этапе развития информационного обеспечения во многих сферах государственной, производственной и научной деятельности наибольшее распространение стали получать перспективные технологии создания и внедрения инфраструктуры пространственных данных (ИПД) и геоинформационных систем (ГИС). По официальному определению, ИПД это система базовых пространственных данных и метаданных, организационных структур, механизмов правового регулирования, методической базы, технологий и технических средств, обеспечивающая широкий доступ и эффективное использование пространственных данных гражданами, организациями и органами власти [3]. Так, например, Правительством РФ уже одобрена «Концепция создания и развития инфраструктуры пространственных данных Российской Федерации» [311], которая предусматривает переход к цифровым технологиям получения и использования пространственных данных. Согласно «Концепции…», в стране должна быть создана иерархическая территориально-распределенная система сбора, обработки, хранения и предоставления базовых пространственных данных и метаданных, включающая подсистемы уровней государственной власти и местного самоуправления и предоставляющая пользователям удаленный доступ к цифровым базам пространственных данных и метаданным. Задачами ближайшего будущего являются апробация системы в рамках пилотных проектов на уровне субъектов РФ, создание организационной структуры ИПД РФ и массива базовых пространственных данных. К сожалению, в Украине технология ИПД начинает только развиваться и нашла свою реализацию в немногочисленных силовых ведомствах и министерствах.

Признанным лидером в создании и продвижении технологий ИПД и ГИС является компания ESRI, которая недавно выпустила семейство программных

продуктов нового поколения — ArcGIS 10 [312]. Платформа ArcGIS является оптимальным решением для построения корпоративной ГИС, фундамента информационной системы эффективного управления крупными государственными и коммерческими организациями. ArcGIS построена на основе стандартов компьютерной отрасли, включая объектную архитектуру COM, .NET, Java, XML, SOAP, что обеспечивает поддержку общепринятых стандартов, гибкость предлагаемых решений, широкие возможности взаимодействия. Архитектура ArcGIS обеспечивает ее использование во многих прикладных сферах и на разных уровнях организации работы: на персональных компьютерах, на серверах, через Web, или в «полевых» условиях.

Новые информационные технологии нашли применение и в транспортных системах. Так, например, для оптимизации маршрутов грузовых и пассажирских перевозок появилась новая версия программного продукта ArcLogistics — инструмента для планирования и оптимизации работы парка транспортных средств: импорта заказов, расчета оптимальных маршрутов, создания маршрутных листов, построения отчетов, анализа эффективности работы. В ArcLogistics реализован новый алгоритм сетевого анализа VRP (vehicle routing problem), появившийся в последней версии модуля ArcGIS Network Analyst [313]. ArcLogistics построен на базе общего ядра с продуктами семейства ArcGIS. При оптимизации маршрутов движения достигается снижение затрат, связанных с транспортировкой (затраты на горючее, стоимость рабочего времени водителей). По оценкам, полученным на основе опыта внедрения программного продукта ArcLogistics (более тысячи лицензий за время существования продукта), ежегодное снижение затрат составляет от 10 до 25%. Кроме уменьшения пробега и количества транспортных средств уменьшается и время планирования. При этом достигается повышение производительности до 10-15% за счет увеличения числа клиентов, обслуживаемых тем же парком транспортных средств и уменьшения времени реагирования на поступающие заказы.

Одним из главных факторов роста эффективности функционирования транспортных сетей с мелкопартионными дискретными потоками корреспонден-

ций является внедрение контейнерной технологии перевозок (КТП), при которой мелкопартионные корреспонденции (МК) перед отправкой получателю подлежат упаковке в транспортные блоки (контейнеры). При этом в один транспортный блок могут попасть элементы с разными адресами получателей. К очевидным преимуществам контейнерной технологии можно отнести: повышение уровня механизации и автоматизации погрузочно-разгрузочных работ; сокращение времени на выгрузку и погрузку грузов в транзитных узлах на маршрутах следования транспортных средств; снижение объемов ручного труда и уменьшение числа неквалифицированных работников; повышение сохранности грузов и сопроводительных документов; возможность хранения контейнеров на открытых площадках. При этом в качестве «контейнера» может использоваться разовая пленочная упаковка, в которую запаковывается заданное количество единичных корреспонденций. Применение дешевой, но достаточно прочной пленочной упаковки исключает необходимость содержания парка специализированных дорогих контейнеров и решения задачи оптимизации развозки порожних контейнеров на уровнях текущего планирования и оперативного управления при управлении парком контейнеров. Кроме того, значительно возрастает полезная загрузка транспортных средств. Размер (объем) транспортного блока может варьироваться в определенных пределах в зависимости от различных характеристик единичных корреспонденций — размеров, стоимости, приоритета и пр. Размеры транспортных блоков должны быть кратными вместимости (грузоподъемности) транспортных средств. Предполагается, что имеется множество транспортных средств с различной грузоподъемностью, например, 10, 15, 20 условных транспортных блоков и т.д.

Несмотря на очевидную «привлекательность» контейнерной технологии, на протяжении последних лет объемы контейнерных перевозок мелкопартионных корреспонденций в Украине увеличиваются недостаточно интенсивно. Для примера, в США в год перевозится более 15 миллиардов тонн мелкопартионных грузов на общую сумму более 9 триллионов долларов, а доходы от транспортировки мелкопартионных грузов составляют около 11% от валовой нацио-

нальной продукции США. Для перевозки таких грузов в трейлерах (контейнерах) в США ежедневно используются десятки тысяч автомашин и сотен реактивных самолетов [214]. Основными причинами этого являются организационно-техническая неподготовленность транспортных предприятий к внедрению современных информационно-аналитических систем, технологий и инструментальных средств автоматизированного проектирования, управления и поддержки принятия решений, в том числе и с использованием ситуационных центров обработки и представления информации. Кроме того, в существующих в настоящее время автоматизированных логистических транспортных системах не реализованы научно обоснованные методы математического моделирования перевозок мелкопартионных грузов и соответственно отсутствуют инструментальные средства для управления такими перевозками. Внедрение результатов работы позволит ликвидировать существующий пробел в этой области и улучшить обслуживание граждан и различных частных и государственных предприятий транспортными услугами по перевозке мелкопартионных грузов от «двери до двери».

Важное место в проектировании перевозок мелкопартионных корреспонденций занимает задача разработки схемы перевозок по транспортным магистралям, соединяющим крупные узловые предприятия транспортной сети. Ее решение позволяет определить основные технико-эксплуатационные и экономические показатели функционирования сети перевозок. Существующие практические методы разработки схемы магистральных перевозок предполагают использование в основном субъективной оценки качества перевозки грузов диспетчерами. Поэтому эффективность схемы перевозок сильно зависит от квалификации, опыта и интуиции работников, занятых ее разработкой. Ограниченность возможностей практических методов отрицательно влияет на важнейшие показатели качества транспортных сообщений на магистралях: устойчивость, регулярность, отсутствие необменов и провозов, соблюдение контрольных сроков продвижения (доставки) корреспонденций. В большинстве из известных работ (обзор которых приведен в [15, 117] и в диссертации при рассмотрении

конкретных задач) непосредственно посвященных распределению потоков и связанных с задачами анализа и синтеза многопродуктовых сетей предлагаются линейные и нелинейные модели и соответствующие методы решения в основном для непрерывных переменных и параметров элементов сети. Кроме того, рассматриваемые в литературе модели являются, как правило, довольно грубыми математическими описаниями функционирования сложной сетевой структуры и не учитывают целый ряд физических характеристик и параметров, присущих реальным системам (т.е. являются сильно агрегированными). В математических моделях проектирования и анализа схемы перевозок мелкопартионных корреспонденций необходимо учитывать целый ряд важнейших факторов, характерных для обработки и перевозки потоков в реальных сетях. Прежде всего, к ним следует отнести: 1) процессы сортировки мелкопартионных корреспонденций в узловых транспортных предприятиях, которые непосредственно связаны с их транспортировкой в пункты назначения; 2) контрольные сроки продвижения (доставки) корреспонденций; 3) провозные возможности (грузоподъемность) магистрального транспорта; 4) время (объемы) погрузки и выгрузки в пунктах следования транспортных средств; 5) нелинейность приведенных (эксплуатационных и капитальных) затрат на обработку и перевозку.

Поэтому, актуальной остается проблема разработки методологии проектирования многопродуктовых транспортных сетей, обобщающей полученные ранее результаты и позволяющей с использованием моделей различной степени агрегирования и для всех уровней иерархии успешно решать практические задачи перспективного развития, текущего планирования и оперативного управления для сетей большой размерности с дискретными потоками и дискретными параметрами элементов сети.

В заключение подраздела отметим, что похожая технология виртуальных контейнеров, при которой в один виртуальный контейнер могут объединяться сообщения с разными адресами назначения, применяется при построении перспективных крупномасштабных общенациональных и интернациональных сетей передачи данных на основе сверхширокополосных каналов и схем типа опор-

ной сети (backbone, известна, например, Европейская опорная сеть E — bone). В настоящее время при проектировании таких сетей используются метод коммутации каналов, европейская технология синхронной цифровой иерархии (Synchronous Digital Hierarchy, SDH) или американская технология синхронных оптических сетей (Synchronous Optical NETs, SONET) с применением оптоволоконных линий связи. В сетях с коммутацией каналов любая линия связи имеет пропускную способность кратную элементарному каналу — базовой технической характеристики, представляющей фиксированное значение пропускной способности (в телефонных сетях, как правило, пропускная способность элементарного канала составляет 64 Кбит/с). В сетях с коммутацией каналов пропускная способность каждой линии связи должна быть равна целому числу элементарных каналов. Это позволяет создавать надежные сети и гибко формировать цифровые каналы в широком диапазоне скоростей — от нескольких мегабит до десятков гигабит в секунду. Так, например, в системах спектрального уплотнения каналов WDM (wavelength-division multiplexing), DWDM (dense WDM), HDWDM (high dense WDM), позволяющих одновременно передавать несколько информационных каналов по одному оптическому волокну на разных несущих частотах, возможно мультиплексировать от 18 до 64 и более спектральных каналов с пропускной способностью от 40 и более Гбит/с. Эти технологии дают возможность существенно увеличить пропускную способность линий связи, причем они позволяют использовать уже проложенные волоконно-оптические линии, организовать двустороннюю многоканальную передачу трафика по одному волокну. Преимуществом DWDM-систем является возможность передачи высокоскоростного сигнала на сверхдальние расстояния без регенерации сигнала усилителями в промежуточных пунктах. Мультиплексоры SDH с волоконно-оптическими линиями связи между региональными подсетями, к которым подключены локальные сети, образуют среду, в которой формируются цифровые каналы между точками подключения к магистральной сети. Сети SDH относятся к классу сетей с коммутацией каналов на базе синхронного мультиплексирования с разделением по времени (Time Division Multiplexing, TDM), при

котором адресация информации от отдельных абонентов определяется ее относительным временным положением внутри составного кадра, а не явным адресом, как это происходит в сетях с коммутацией пакетов. Каналы SDH относятся к классу полупостоянных (semipermanent) — формирование (provisioning) канала происходит по инициативе оператора сети SDH, пользователи же лишены такой возможности, поэтому такие каналы обычно применяются для передачи достаточно устойчивых во времени потоков. Из-за полупостоянного характера соединений в технологии SDH чаще используется термин «кросс-коннект» (cross-connect), а не коммутация. С помощью каналов SDH обычно объединяют большое количество периферийных (и менее скоростных) каналов плезиохронной цифровой иерархии (Plesiochronous Digital Hierarchy, PDH).

Поэтому отдельные результаты диссертации, а именно математические модели задач обработки и транспортировки мелкопартионных потоков корреспонденций могут быть использованы и для проектирования магистральных сетей передачи данных с технологией виртуальных контейнеров.

1.2. Зонально-узловая структура транспортной сети. Принципы организации сортировки и транспортировки мелкопартионных корреспонденций. Математическая модель многопродуктовой иерархической сети

Масштабность сети перевозок, многообразие и сложность выполняемых функций, функционирование в условиях неполной информации и воздействия случайных факторов, вынуждают рассматривать ее как сложную систему. Основные понятия теории и методология системной оптимизации сложных систем изложены в работах Г. Гуда (H. Good), Р. Маккола (R. Machol), М. Месаровича (M. Mesarovic), Д. Мако (D. Macko), Я. Такахара (Y. Takahara), Д. Лэсдона (D. Lesdon), Н.П. Бусленко, Дж. Клира (G. Klir). В основу методологии системной оптимизации положены принципы последовательной агрегации нормативов, связывающих воедино все уровни планирования; проектирования динамических моделей в статику; многокритериальной оптимизации планов; согласования плановых решений по всем уровням иерархии планирования. Свое даль-

нейшее развитие методы проектирования крупномасштабных сетевых структур получили при решении задач структурного анализа, синтеза и оптимизации для различных типов сетей в работах П. Грина (P. Green), Р. Лакки (R. Lucky), Л. Клейнрока (L. Kleinrock), В.М. Глушкова, В.М. Вишневского, Ю.П. Зайченко, И.В. Кузьмина, Е.Г. Петрова, М. Шварца, А.Д. Цвиркуна, Б.Я. Эттингера, Е.А. Якубайтиса, Г.Ф. Янбыха и др.

Проектирование иерархических сетевых структур имеет, как правило, нисходящий характер. На верхнем уровне решаются задачи структурного синтеза и перспективного развития сети, для которых используются крупноагрегированные модели распределения потоков. На низших уровнях проектирования детализация объектов должна увеличиваться с целью наиболее адекватного описания их функционирования и принятия рациональных решений. Это обуславливает итерационно-циклический характер процессов проектирования и управления, включающих процедуры синтеза и анализа возможных решений на всех уровнях сети. Поскольку решения принимаются в условиях неопределенности, связанных с неполнотой имеющейся информации, а также с огрублением математических моделей, необходимо разделять решения на перспективные, текущие и оперативные.

Как известно, большинство математических моделей управления, описывающих сложную систему как единое целое, не являются адекватными реальным процессам из-за трудностей формального описания многих ограничивающих условий. Для сети перевозок неадекватность общих моделей объясняется также невозможностью оперативной корректировки моделей в изменяющихся условиях протекания процесса, а также необоснованностью выбора критериев или заменой ряда критериев одним, имеющим сложный условно-экономический характер. Попытки представить модель на уровне агрегированных показателей ведут к потере специфики организации перевозок и, как правило, к неадекватности модели. Составление развернутых моделей, когда управление на нижних уровнях представлено детализированными моделями, приводит к необозримости получаемых решений и некомпетентности специа-

листов, ответственных за организацию перевозок. Разрешение такой ситуации может быть в создании комплекса взаимосвязанных многоуровневых моделей планирования и управления, отображающих иерархию сети перевозок и соответствующую ей степень агрегирования показателей.

Руководствуясь общей методологией системной оптимизации, определим принципы зонально-узловой организации сети перевозок при контейнерной технологии обработки и перевозки мелкопартионных корреспонденций. Выделим в сети перевозок три уровня иерархии — магистральный, зональный и внутренний и четыре типа узлов — узлы первого, второго, третьего и четвертого типов. Узлы первого, второго и третьего типа, находящиеся на транспортных магистралях или поблизости от них и соединяющие их участки маршрутов транспортных средств (железнодорожного, автомобильного, авиа и водного транспорта) составляют магистральный уровень и определяют магистральную сеть. Узлы первого типа являются крупными узловыми транспортными предприятиями, обладающими мощной материально-технической базой. В узлах этого типа уровень концентрации мелкопартионных корреспонденций позволяет формировать потоки транспортных блоков в адреса обслуживаемых узлов второго и третьего типа, других предприятий первого типа и некоторые дополнительные направления. Узлы второго и третьего типа формируют транспортные блоки только до ближайших узлов первого типа. Эти узлы являются, как правило, областными, автономными, краевыми и районными транспортными центрами. Уровень механизации и автоматизации в таких предприятиях ниже, чем в узлах первого типа. Узлы четвертого типа являются доставочными предприятиями (районные, городские и другие выделенные транспортные узлы) и находятся во внутренней зоне обслуживания какого-либо магистрального узла. Как правило, узлы четвертого типа — это пункты приема и выдачи мелкопартионных грузов (либо полностью автоматизированные, либо являются автоматизированными рабочими местами приемщиков грузов в отделениях внутренней зоны магистрального узла).

Все магистральные узлы осуществляют магистральную сортировку корре-

споненций. В узлах четвертого типа корреспонденции не сортируются. Узлы четвертого типа доставляют все исходящие потоки на обслуживающие их магистральные узлы и от них же получают входящие потоки. Зона обслуживания (ЗОУ) любого магистрального узла представляет собой связный фрагмент магистральной сети, построенный по следующим правилам: 1) от узла проводятся все простые (в терминах теории графов) пути через узлы второго и третьего типа до первых встретившихся узлов первого типа и тупиковых (висячих) узлов второго и третьего типа; 2) определенные по правилу 1) узлы первого типа и тупиковые узлы соединяются условной границей зоны обслуживания. Определенные таким образом области ЗОУ представляют зональные уровни сети (зональные сети). Внутренние уровни сети (внутренние сети) представлены магистральными узловыми транспортными предприятиями и обслуживаемыми ими доставочными предприятиями, не лежащими на магистралях.

В узлах третьего типа запрещена обработка транзитных потоков единичных корреспонденций и потоков транзитных транспортных блоков. В узлах второго типа запрещена только обработка транзитных потоков единичных корреспонденций. Исходящие и входящие потоки узлов второго и третьего типа обрабатываются в соответствующих им узлах первого типа.

Каждый узел в сети имеет географические координаты, название, уникальный почтовый индекс и порядковый номер. Каждому магистральному узлу можно поставить во взаимно-однозначное соответствие множество уникальных почтовых индексов обслуживаемых доставочных предприятий, находящихся в зоне обслуживания этого узла.

На рис. 1.1 показаны фрагменты сетей, а также примеры ЗОУ для узлов первого, второго и третьего типа. Фактическое формирование ЗОУ и определение типов узлов в магистральной сети выполняется при решении задачи упаковки и выбора структуры сети, рассмотренной в подразделе 3.5.

Рассмотрим принципы сортировки и транспортировки мелкопартионных корреспонденций, положенные в основу организации системы контейнерных перевозок. Магистральными направлениями сортировки из каждого магистра-

льного узла называются направления сортировки на узлы первого типа. Зональные направления сортировки из каждого магистрального узла представлены направлениями сортировки на магистральные узлы второго и третьего типа, лежащие в ЗОУ или на границе ЗОУ этого узла. Внутренние направления сортировки — это направления сортировки из каждого магистрального узла на сеть доставочных предприятий этого узла.

Схема сортировки потоков мелкопартионных корреспонденций для каждого магистрального узла полностью определяется количеством магистральных, зональных и внутренних направлений сортировки.

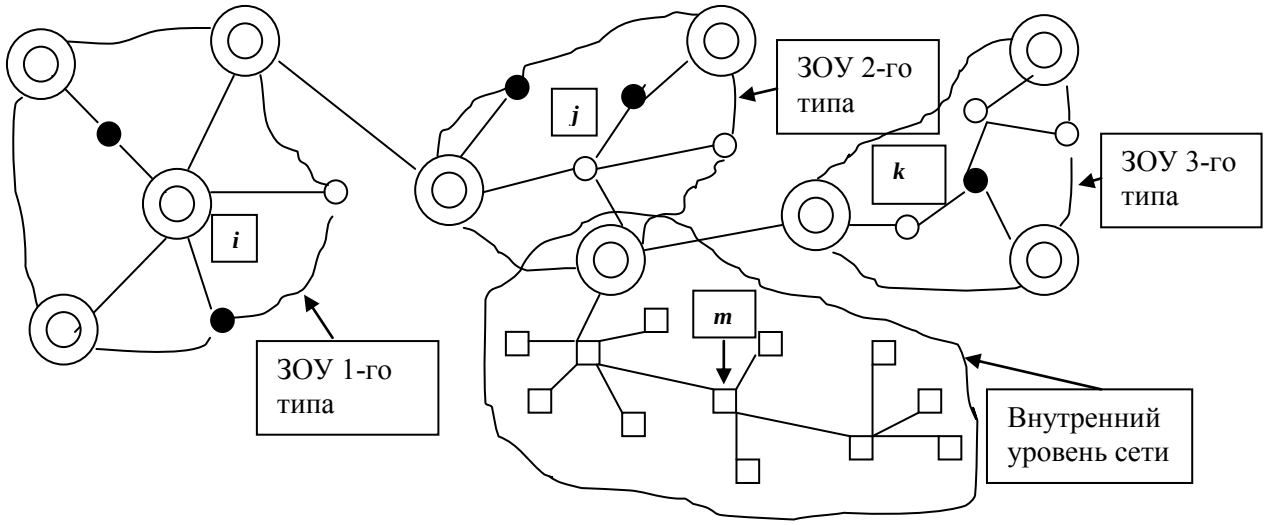


Рис. 1.1. Где *i*, *j*, *k*, *m* — соответственно узлы первого, второго, третьего и четвертого типа

Количество магистральных и зональных направлений сортировки для узлов первого, второго и третьего типа определяется после решения задачи упаковки и формирования потоков транспортных блоков (подраздел 3.3). Перевозка мелкопартионных корреспонденций, отсортированных по магистральным и зональным направлениям и упакованных в транспортные блоки, осуществляется на основании схемы транспортировки, полученной после решения задачи распределения и маршрутизации сформированных потоков транспортных блоков (раздел 4).

Количество внутренних направлений сортировки для любого магистрального узла определяется числом обслуживаемых узлов четвертого типа во внут-

ренней сети. Перевозка корреспонденций по внутренним направлениям выполняется оптимизированными автомобильными маршрутами, полученными задачами текущего планирования внутренних перевозок (раздел 2).

Поток корреспонденций, поступающий в узел первого типа от зональных и внутренних предприятий, может быть в виде «россыпи» или в транспортных блоках, а от магистральных предприятий вне ЗОУ — только в транспортных блоках. Весь входящий поток, поступивший в адрес узла первого типа, рассортировывается аналогично исходящему.

Исходящие и входящие потоки в узлах второго и третьего типа обрабатывается так же, как и в узлах первого типа, с той лишь разницей, что магистральные направления сортировки для исходящих потоков ограничены ЗОУ. Потоки из этих узлов, адресованные за пределы ЗОУ, поступают в узлы первого типа, лежащие на границе ЗОУ, и дополнительно рассортировываются.

Отсортированные по магистральным и зональным направлениям мелкопартионные потоки корреспонденций упаковываются в транспортные блоки. На магистральном уровне транспортировка потоков возможна только в транспортных блоках. Узел первого типа может отправлять транспортные блоки в любые другие узлы первого типа, а также в узлы второго и третьего типа, находящиеся в зоне его обслуживания. Например, на рис. 1.1 узел первого типа i может отправлять транспортные блоки во все другие узлы первого типа, один узел второго типа и два узла третьего типа в зоне своего обслуживания. Узел второго типа j может отправлять транспортные блоки в три узла первого типа, один узел второго типа и один узел третьего типа. Узел третьего типа k может отправлять транспортные блоки только в три узла первого типа и три узла второго типа. Входящие потоки в транспортных блоках в узлы второго и третьего типа могут поступать только от узлов, расположенных в зоне обслуживания этих узлов. То есть не может быть потока из узла i и узла j в узел k. Потоки в узел k могут поступать только от трех узлов первого типа и трех узлов второго типа, находящихся в зоне обслуживания узла k. В магистральной сети не может быть прямого потока транспортных блоков между узлами первого типа и

узлами второго или третьего типа и наоборот, если только они не находятся в одной зоне обслуживания.

Согласно введенным понятиям определим математическую модель транспортной сети. Пусть физическая сеть задана связным графом $\hat{G}(\hat{N},\hat{P})$ с множеством узлов $\hat{N}=N_1\cup N_2\cup N_3\cup N_4$, $\hat{n}=\left|\hat{N}\right|$ и множеством дуг $\hat{P}$, $\hat{p}=\left|\hat{P}\right|$, где N_1,N_2,N_3,N_4 — множества узлов первого, второго, третьего и четвертого типа, $\hat{n}$ и $\hat{p}$ соответственно число узлов и дуг сети, а $\left|\cdot\right|$ — знак мощности множества. Дуги представляют собой участки дорог или линии связи и могут быть как ориентированными, так и неориентированными. Длины дуг заданы матрицей $\hat{R}=\left\|\hat{r}_{ij}\right\|_{\hat{n}\times\hat{n}}$, $\hat{r}_{ij}\in R^1$, $(i,j)\in\hat{P}$, где R^1 — множество действительных чисел. Потоки мелкопартионных корреспонденций между всеми парами узлов заданы целочисленной матрицей $\hat{A}=\left\|\hat{a}_{ij}\right\|_{\hat{n}\times\hat{n}}$, где $\hat{a}_{ij}$ — поток из узла i в узел j, $i,j=\overline{1,\hat{n}}$, $i\neq j$. Определим также магистральную сеть $G(N,P)$ с множеством узлов $N=N_1\cup N_2\cup N_3$, $n=\left|N\right|$, множеством дуг P, $p=\left|P\right|$ и внутриузловые сети $\tilde{G}_k(\tilde{N}_k,\tilde{P}_k)$, $\tilde{N}_k=\{N_0^k,N_4^k\}$, $N_0^k\in N$, $N_4^k\subseteq N_4$, $\left|N_0^k\right|=1$, $n_k=\left|\tilde{N}_k\right|=1+\left|N_4^k\right|$, $p_k=\left|\tilde{P}_k\right|$, $k=\overline{1,n}$.

Для внутриузловых сетей выполняется условие $\bigcap_{k=1}^{n}\tilde{G}_k(\tilde{N}_k,\tilde{P}_k)=\varnothing$. Для полной сети имеется процедура редукции $\phi:\hat{G}(\hat{N},\hat{P})\Rightarrow G(N,P)+\bigcup_{k=1}^{n}\tilde{G}_k(\tilde{N}_k,\tilde{P}_k)$, которая в соответствии с принципами сортировки и транспортировки мелкопартионных корреспонденций в иерархической сети преобразовывает потоковую матрицу $\hat{A}=\left\|\hat{a}_{ij}\right\|_{\hat{n}\times\hat{n}}$ в матрицы $A=\left\|a_{ij}\right\|_{n\times n}$ и $\tilde{A}_k=\left\|\tilde{a}_{ij}^k\right\|_{n_k\times n_k}$, а матрицу длин дуг $\hat{R}=\left\|\hat{r}_{ij}\right\|_{\hat{n}\times\hat{n}}$ соответственно в матрицы $R=\left\|r_{ij}\right\|_{n\times n}$ и $\tilde{R}_k=\left\|\tilde{r}_{ij}^{\,k}\right\|_{n_k\times n_k}$, $k=\overline{1,n}$. В матрице A значения a_{ii}, $i=\overline{1,n}$ определяют внутриузловые объемы на обработку потоков в магистральной сети.

Процедура редукции позволяет автоматизированным способом преобразовать данные исходной полной сети в данные магистральной и внутриузловых сетей при изменении ее структуры и соподчинения узлов четвертого типа магистральным узлам.

1.3. Технология сортировки мелкопартионных потоков корреспонденций, транспортные средства для перевозки контейнеров, контейнерный терминал

Постоянный рост обмена потоками в транспортной сети влечет за собой увеличение грузопотоков и объемов погрузочно-разгрузочных работ. Увеличение обрабатываемых объемов потоков приводит к необходимости создания новой высокопроизводительной техники для сортировки, перевозки и погрузки-выгрузки грузов, дальнейшего развития автоматизации и механизации технологических процессов их обработки и перевозки.

Для сортировки мелкопартионных корреспонденций необходимо использовать высокопроизводительные поточные системы автоматизированной обработки грузов (САОМГ), схема работы которых проиллюстрирована на рис. 1.2. Входной поток единичных корреспонденций из общего накопителя (или накопителей) поступает на линии сортировки. Число таких линий регулируется в зависимости от общего объема корреспонденций, ежесуточно поступающего на сортировку. Входной поток включает исходящие, входящие и транзитные корреспонденции и обрабатывается по мере их поступления в непрерывном режиме или с предварительным накоплением в зависимости от интенсивности потока. Устройство считывания уникальных почтовых индексов корреспонденций (или их расширенных штрих - кодов), расположенное в начале каждой линии, считывает индексы и передает их в компьютер. Расширенный штрих - код может содержать данные об индексах предприятий отправителя и получателя; о кодах юридических или физических лиц отправителей и получателей; о дате приема корреспонденции в транспортном предприятии отправителя и пр. Такая расширенная информация о каждой корреспонденции дает возможность авто-

матизированного отслеживания ее продвижения от отправителя к получателю. В памяти компьютера хранятся сортировочные таблицы, содержащие группы индексов, соответствующие каждому направлению сортировки. Общее число

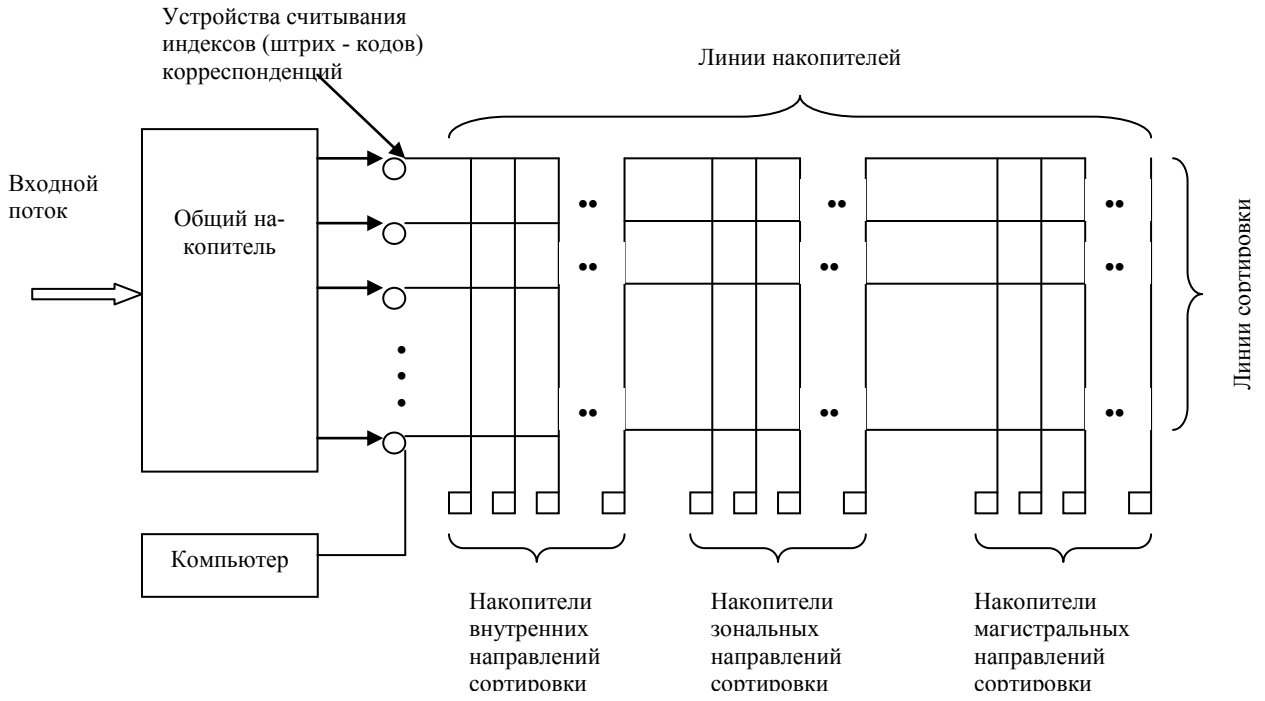


Рис. 1.2

направлений сортировки определяется суммой магистральных, зональных и внутренних направлений. В соответствии с расшифрованным индексом корреспонденция направляется в нужный накопитель. Состав групп индексов, соответствующих внутренним и зональным направлениям сортировки, практически постоянен и изменяется только при введении новых предприятий, обслуживаемых данным узлом или изменении зоны обслуживания узла. Число и состав магистральных направлений сортировки определяются после решения задачи упаковки и формирования потоков транспортных блоков.

В узлах первого типа предполагается установка отечественных или импортных поточных сортировочных систем. В остальных магистральных узлах возможна установка менее производительных, но более дешевых систем. На этапе опытного внедрения, при отсутствии дорогостоящего оборудования, для сортировки корреспонденций могут быть использованы ручные сканеры штрих - кодов на рабочих местах сортировщиков.

Для перевозки мелкопартионных корреспонденций (грузов) в контейнерах необходима разработка и отечественное промышленное производство (или приобретение у других стран) современных контейнеров, погрузчиков и транспортных средств для перевозки контейнеров, оснащенных средствами полной механизации и автоматизации процессов погрузки-выгрузки контейнеров. Кроме того, в магистральных узловых предприятиях необходимо строительство и оборудование контейнерных терминалов, способных обеспечить необходимую производственную мощность для обработки входящих, исходящих и транзитных контейнеров.

Таким образом, очевидно, что контейнерная технология обработки грузов требует весьма значительных капитальных затрат и должна внедряться поэтапно с учетом ежегодно выделяемых инвестиций.

1.4. Инфраструктура пространственных данных транспортной сети. Структурная схема информационно-аналитической системы поддержки принятия решений

Для создания и распространения механизмов хозяйствования и управления в зонально-узловой транспортной сети необходимы организационная, информационная, телекоммуникационная инфраструктуры связей, а также распределенные вычислительные ресурсы, охватывающие все уровни иерархии сети. Отсутствие или бедность инфраструктуры тормозит процесс установления связей, способствует ее монополизации и приводит к фактической консервации существующей технологии функционирования.

За счет многочисленных связей и зависимостей в транспортной сети часто проявляется «каскадный эффект», когда перегрузка одного или нескольких магистральных узлов или маршрутов приводит к перегрузке и выходу из строя многих других элементов сети. Все это обуславливает необходимость создания инфраструктуры пространственных данных (ИПД) и геоинформационной системы (ГИС) сети. Для транспортной сети ИПД и ГИС должны обеспечить всем диспетчерским службам интерактивный картографический доступ к наборам

пространственных данных, характеризующих состояние узлов и транспортных магистралей в реальном времени, а также к решению различных задач прогнозирования, анализа, перспективного развития, текущего планирования и оперативного управления на всех уровнях иерархии сети. Доступ может осуществляться через ведомственные каналы связи или Интернет посредством веб-серверов и обычного веб-браузера (например, Internet Explorer).

Кроме того, для эффективного управления процессами обработки и транспортировки мелкопартионных корреспонденций необходима разработка экспертно-аналитических систем поддержки принятия решений для задач перспективно развития зонально-узловой структуры сети; проектирования текущих (среднесрочных) схем сортировки и транспортировки грузов; оперативного управления потоками и распределения ресурсов между узлами сети; анализа возможностей улучшения работы сети за счет рационального перераспределения потоков и др. Специфика задач принятия решений в транспортных системах определяется, прежде всего, неопределенностью и наличием общих транспортных магистралей, разделяемых между различными транспортными ведомствами и предприятиями. В условиях дефицита ресурсов сети возникают отказы, потери, задержки и ухудшение качества работы всей транспортной системы, поэтому необходимо минимизировать различные факторы риска и гарантировать конечный результат — доставку корреспонденций адресату. Помимо объективной (физической) неопределенности, необходимо учитывать и субъективную неопределенность, когда некоторые текущие параметры сети (загрузка узлов, транспортных магистралей) известны отдельным диспетчерским службам, но не известны главной диспетчерской службе, ответственной за принятие окончательного решения. Схема функционирования транспортной сети может отличаться от расчетной, полученной на этапе проектирования, поэтому естественна ее корректировка в процессе перераспределения потоков, т.е. характерна многоэтапность процедуры принятия решений.

Методика проектирования транспортной системы должна отражать невозможность полной централизации в одном звене обработки информации и при-

нятия решений по управлению процессами сортировки и перевозки грузов. Это приводит к необходимости формирования иерархической структуры информационно-аналитической системы поддержки принятия решений (ИАС ППР) для проектирования и управления, соответствующей уровневой декомпозиции сети перевозок (магистральный, зональный и внутренний уровни) и этапной декомпозиции самого процесса проектирования и управления (модели перспективного развития, текущего планирования, оперативного управления). На низших уровнях детализация проектируемых объектов должна увеличиваться с целью уточнения параметров и характеристик текущего функционирования и оперативного управления и принятия наиболее рациональных решений.

Решение указанных проблем неразрывно связано с созданием и внедрением уникальной информационной системы сбора и хранения данных о транспортной системе, обеспечивающей требуемое качество и соответствующий статус циркулирующей в ней информации. Это, в свою очередь, определяет необходимость обеспечения соответствующих требований к показателям качества функционирования ИАС, и в первую очередь к ее надежности. Недостаточная надежность ИАС ухудшает в целом ряд показателей качества обрабатываемой в ней информации, в том числе ее физическую целостность, достоверность, полноту, безопасность и приводит к угрозе принятия неэффективных решений должностными лицами в процессе управления.

Как уже отмечалось выше, ИАС должна представлять собой информационно-картографическую систему на базе новейших информационных технологий — инфраструктуры пространственных данных, геоинформационных систем и распределенных вычислительных сетей. При этом на главном узле управления (главном центре магистральных перевозок) размещаются: главный геопортал, распределенная база данных и метаданных, управляемые системой управления базами данных (СУБД), база знаний, специализированное программное обеспечение, ГИС-сервер, веб-сервер, прикладное программное обеспечение магистрального уровня. На всех магистральных узлах размещаются локальные (зональные и внутренние) базы данных, специализированное и приклад-

ное программное обеспечение.

ИАС должна включать глобальную сеть передачи данных (ведомственные каналы связи или Интернет), соединяющую все предприятия в транспортной сети и локальные вычислительные сети (ЛВС) в магистральных узлах. Информационно-картографическая система предоставляет возможность работать с многослойной электронной картой узлов и транспортных магистралей сети перевозок. На карте можно одновременно или в любой комбинации видеть узлы, зоны обслуживания узлов, транспортные магистрали, выделенные фрагменты сети (полигоны), маршруты движения транспортных средств, плановую и текущую загрузку узлов и транспортных средств на маршрутах движения, объемы неотправленных вовремя грузов в узлах сети и множество другой информации, характеризующей сеть перевозок. При этом можно изменять масштаб изображения, переключаться на разные слои электронной карты, вызывать различные прикладные программы.

На рис. 1.3 представлена примерная структурная схема ИАС для транспортной сети. Как видно из рисунка, при разработке такой системы возникает проблема создания специализированного математического обеспечения, периферийных информационных сетей и систем, обеспечивающих прием, передачу, обработку и хранение информации. Решение проблемы включает следующее: разработку программного комплекса управления обработкой и транспортировкой мелкопартионных корреспонденций на каждом уровне планирования; выбор структуры региональной сети вычислительных машин, которая должна отображать иерархию транспортной сети; выбор систем связи, аппаратных средств и режимов передачи информации; создание автоматизированных систем управления технологическими процессами сортировки мелкопартионных грузов (САОМГ) и обработки контейнеров на контейнерном терминале (КТ).

Приведем некоторые особенности решения и характеристики основных задач ИАС. В настоящее время одним из главных направлений в развитии автоматизированных систем управления, систем обработки данных, является создание такой вычислительной среды, при которой пользователи системы работали

бы наиболее эффективно с минимальными непроизводительными затратами.

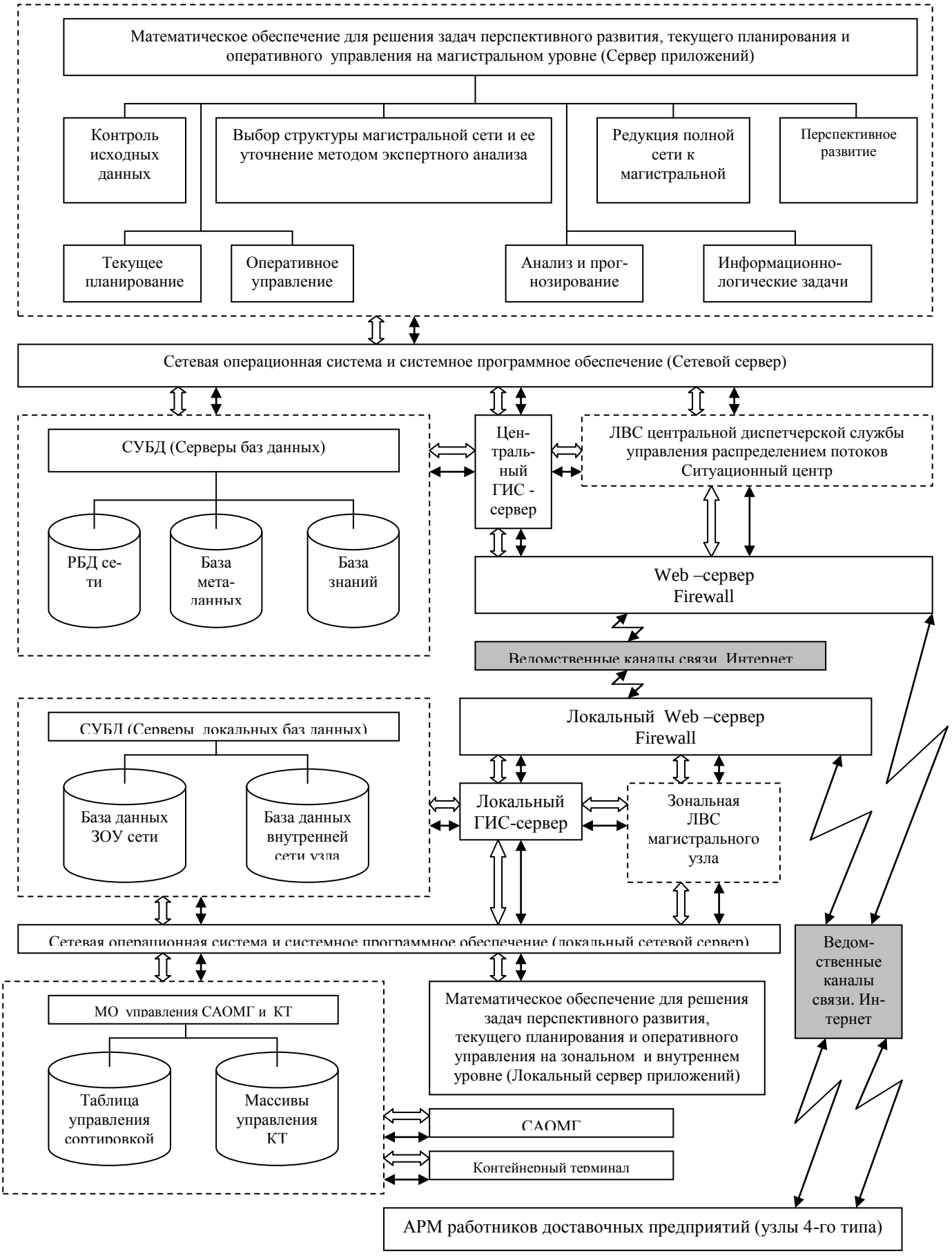


Рис. 1.3

Таким образом, возникает задача автоматизации функций информационного обеспечения в управлении транспортной системой. Методологической основой этого является применение комплексного подхода к проектированию переработки информации, подразумевающего создание такой системы, функционирующей на единой информационной базе, в которой процессы получения справочной информации и процессы управления переработкой и хранением информации функционально и организационно отделены друг от друга. Поэтому комплекс программ управления транспортной системой должен использовать специализированные базы данных (БД), основная функция которых состоит в поддержании в системе динамической информационной модели сложного управляемого и исследуемого объекта и обеспечение коллективного доступа к этой информации со стороны заинтересованных пользователей. При использовании баз данных, процесс формирования, накопления и ведения информационной базы выделяется в автономный, который связан с программами прикладных задач только информационно.

Вся необходимая оперативная информация о сети (объемы суточных потоков в узлах сети, маршруты и провозные возможности магистрального транспорта, пропускные способности узловых предприятий и множество другой текущей информации) поступает в распределенную базу данных (РБД) через аппаратуру передачи данных (АПД) и ведомственные каналы связи или Интернет от магистральных узлов и доставочных предприятий. В РБД хранится и обновляется с определенной периодичностью вся информация, необходимая для решения задач управления транспортной системой на магистральном уровне. Назначение базы данных заключается не только в поддержании в актуальном состоянии исходных информационных массивов, но и в хранении результатов решения всех задач управления на магистральном уровне. База метаданных содержит стандартизованный набор сведений — каталог (например, в стандарте метаданных ISO19139) обо всех существующих информационных ресурсах сети и их расположении в ИАС. В базе знаний располагается и накапливается информация для оперативного реагирования и принятия решений в случае возник-

новения внештатных ситуаций. База знаний и ситуационный центр используется в основном в экспертно-обучающей системе диспетчеров при анализе и выборе альтернативных вариантов выхода из критических состояний транспортной сети (перегрузках маршрутов и узлов, отказов сортировочного оборудования и контейнерных терминалов и т.п.).

Данные в базах хранятся в виде геоданных — модели, определяющей структуру и правила хранения различных видов данных — векторных и растровых, адресных точек, данных геодезических измерений, числовых, текстовых, табличных и других. Центральный ГИС сервер осуществляет требуемую функциональную геообработку и визуализацию данных на рабочих местах ЛВС центральной диспетчерской службы магистральных перевозок и экране ситуационного центра. Управление базами данных осуществляется системами управления базами данных (СУБД). Web-серверы с межсетевой защитой (Firewall) обеспечивают разграниченный доступ различных пользователей к ресурсам системы управления.

Рассмотрим кратко состав и функции математического обеспечения планирования и управления на магистральном уровне.

Комплекс программ контроля исходных данных предназначен для контроля и диагностики исходных массивов информации и позволяет выявить синтаксические и логические ошибки, допущенные при подготовке данных, а также обеспечивает контроль объективности информации, поступающей по каналам связи от удаленных абонентов.

При решении задачи выбора структуры магистральной транспортной сети определяется количественный и качественный состав ее узлов (узлы первого, второго и третьего типа). Поскольку при решении задачи не удается формализовать все факторы, влияющие на оптимальную структуру сети, для ее окончательного выбора используется опыт практических работников транспортных предприятий.

Программы редукции предназначены для автоматизированного преобразования данных полной сети в данные для магистральной сети. Как указывалось

выше, организация системы управления требует наличия централизованной базы данных, поддерживающей информационную модель полной транспортной сети, а также сети и средств автоматизированного сбора данных. В связи с этим возникает вопрос, какие узлы следует включить в сеть сбора данных? Естественное стремление к уменьшению числа пунктов сбора данных и концентрации обработки информации в крупных узловых предприятиях (например, в узлах первого, второго и третьего типов) с точки зрения организации системы сбора и передачи данных наиболее приемлемо. Однако следует учитывать, что, во-первых, для выбора структуры магистральной сети а также для решения разнообразных задач на зональных и внутренних уровнях сети, необходима информация о полной сети транспортных предприятий; во-вторых, без решения задачи структурной оптимизации априори трудно судить о том, какие узлы должны быть узлами сбора данных. Кроме того, для постоянно развивающейся транспортной сети неизбежно периодическое изменение ее структуры с переводом узловых предприятий с одного уровня на другой. Перестройка структуры сети вызывает определенные трудности в организации системы сбора данных в крупных узловых предприятиях, так как сбор данных должен осуществляться с учетом принципов зонально- узловой сортировки. В этом случае возникает также задача перестройки информационной базы данных транспортной сети. Поэтому наиболее правильным решением будет то, когда все предприятия транспортной системы (узлы первого, второго, третьего и четвертого типа) будут являться источниками информации. Тогда перестройка структуры сети не вызовет никаких изменений в организации и обработке данных на уровне узлов, а все функции, связанные с переработкой данных в соответствии с иерархической структурой выбранной магистральной сети, могут быть выполнены процедурами редукции централизованно автоматизированным способом. Редукции подлежат все данные, используемые для решения задач перспективного развития, текущего планирования и оперативного управления на магистральном уровне. При преобразовании данных используются принципы иерархического соподчинения узлов, что обуславливает и логическое преобразование данных.

К перспективным задачам, прежде всего, относится задача оптимизации поэтапного управления развитием производственно-транспортной системы с учетом всех возможностей качественного организационно-технического совершенствования ее структуры при наличии ограничений на капитальные вложения и возможности их освоения. Кроме того, в эту группу включены задача оптимизации поэтапного плана развития контейнерных транспортных маршрутов и задача корректировки структуры сети с учетом развития промышленно-территориальных комплексов. В первой задаче подразумевается построение новых, ранее не существовавших маршрутов, а во второй — расширение магистральной сети с учетом строительства и освоения новых хозяйственных объектов на территории Украины.

Важное место в составе математического обеспечения транспортной системы на магистральном уровне занимают задачи текущего (тактического) планирования, которые дают рациональные решения на среднесрочные планируемые периоды и нацелены, главным образом, на эффективное использование имеющихся ресурсов транспортных предприятий. Именно этот класс задач является первоочередным объектом исследований и составляет основу — ядро для решения задач перспективного развития и оперативного управления. Первая задача возникает при сортировке мелкопартионных корреспонденций (грузов) в магистральных узлах. Она заключается в определении оптимального количества магистральных направлений сортировки и называется задачей упаковки и формирования потоков транспортных блоков (контейнеров). Результатом решения этой задачи является схема сортировки потоков мелкопартионных грузов в каждом магистральном узле и схема адресации сформированных потоков транспортных блоков. Вторая задача характерна только для транспортных сетей и заключается в балансировке результирующей магистральной матрицы потоков транспортных блоков, необходимость которой возникает из-за неравенства исходящих и входящих потоков в отдельных узлах сети. Окончательно сформированная магистральная матрица потоков является исходной для задачи распределения и маршрутизации, решение которой определяет схему транспор-

тировки транспортных блоков на сети маршрутов транспортных средств.

Среди задач оперативного управления на магистральном уровне выделим наиболее важные: анализ очередей в узлах и загрузки транспортных магистралей и выработка управляющих решений; корректировка схем сортировки мелкопартионных грузов в узлах и схемы транспортировки транспортных блоков при возникновении перегрузок, отказов и т.п.; оперативный учет и планирование парка транспортных средств для устойчивого управления процессами обработки и транспортировки мелкопартионных потоков корреспонденций.

Группа задач анализа и прогнозирования обеспечивает диспетчерский аппарат управления некоторой дополнительной информацией, позволяющей анализировать технико-экономические показатели функционирования транспортной системы. Сюда относятся задачи анализа загрузки элементов транспортной сети, расчета времени доставки корреспонденций конечному потребителю, получения различных характеристик транспортных сетей, прогнозирования исходящих потоков и многие другие. Задачи анализа и прогнозирования тесно связаны с задачами оперативного управления магистральными перевозками и, по сути, являются информационным обеспечивающим звеном для своевременного реагирования и принятия решений в случае возникновения внештатных ситуаций. Результаты решения задач прогнозирования используются для составления перспективных схем магистральных перевозок и являются исходными данными для задачи поэтапного управления развитием сети.

Группа информационно-логических задач входит в состав информационно-справочной подсистемы ИАС, работающей в режиме диалога, управляемого диспетчером-экспертом и предназначена для выдачи на экраны персональных компьютеров и ситуационного центра конкретно запрашиваемой информации в картографическом виде. Так, например, на картографической схеме могут быть отображены: загрузка выбранных транспортных магистралей с перечислением проходящих по ним транспортных средств, адресные потоки транспортных блоков и содержащиеся в них единичные грузы; схемы сортировки мелкопартионных корреспонденций для указанных узлов; маршруты следования отдель-

ных корреспонденций и транспортных блоков с указанием транспортных средств, узлов перегрузки и времени доставки конечному потребителю; объемы исходящих, входящих и транзитных мелкопартионных корреспонденций и транспортных блоков в узлах сети; технико-экономические показатели функционирования узлов и транспортных средств (величина приведенных затрат на сортировку, погрузку-выгрузку и транспортировку, рабочий парк транспортных средств и пр.). Доступ к информации возможен с локальных и удаленных ПЭВМ и разграничен между пользователями системой паролей. Многие функции задач анализа и прогнозирования, а также оперативного управления выполняются посредством кодирования специальных запросов к группе информационно-логических задач. Выходная информация может быть получена и в виде печатных документов, например, в удобной и привычной для диспетчеров форме маршрутных листов, схем детальной сортировки грузов в узловых транспортных предприятиях и т.п.

Математическое обеспечение управления зональными и внутренними перевозками включает: задачи контроля, учета и анализа; задачи перспективного развития, текущего планирования и оперативного управления в зонах обслуживания магистральных узлов и на внутренних уровнях сети перевозок; информационно-логические задачи. Главная цель оптимизационных задач заключается в получении схем транспортировки потоков грузов в пределах зон обслуживания магистральных узлов и в разработке рациональных маршрутов автотранспортных перевозок по доставочным предприятиям этих узлов. Функции остальных задач во многом аналогичны функциям задач на магистральном уровне.

Кроме общего математического обеспечения планирования и управления в магистральных узлах необходимо иметь работающие в реальном масштабе времени системы управления технологическими процессами обработки потоков мелкопартионных грузов (САОМГ) и обработки исходящих, входящих и транзитных транспортных блоков (контейнеров) на контейнерном терминале (КТ). Для управления САОМГ и КТ используются таблицы управления сортировкой и информационные массивы управления контейнерным терминалом. Эта

управляющая информация является результатом решения задач упаковки и формирования потоков транспортных блоков, распределения и маршрутизации сформированных потоков транспортных блоков и задач оперативного управления зонального уровня.

На рис. 1.4 показана последовательность решения задач текущего планирования, иллюстрирующая их взаимосвязь с задачами перспективного развития и оперативного управления, возникающими как на магистральном, так и на зональном уровнях. Процесс решения задач начинается этапом контроля и диагностики исходных данных о полной транспортной сети, включающей узлы первого, второго, третьего и четвертого типа. На втором этапе, после успешного контроля, решаются задачи анализа, позволяющие проверить связность сети и получить ее предварительные характеристики, и осуществляется формирование внутренних сетей с уточнением их границ опытными экспертами. На третьем этапе осуществляется процедура редукции сети. Данные автоматически преобразовываются в соответствии с принятой структурой сети и принципами иерархического соподчинения узлов, формируются базы данных зональных и внутренних сетей. Информационные массивы магистральной сети перед выводом на внешний накопитель подвергаются дополнительному контролю, поскольку после процедуры редукции магистральная сеть может оказаться несвязной. На четвертом этапе решаются оптимизационные задачи текущего планирования, и определяется схема магистральных перевозок грузов. Результаты решения первой задачи используются для разработки сортировочных таблиц, управляющих магистральной сортировкой в узловых предприятиях, а также в задачах оперативного управления на магистральном уровне. Схема магистральных перевозок и схема распределения порожних контейнеров являются исходными данными для задач планирования перевозок в зонах обслуживания узла и для задач оперативного управления магистральными перевозками.

Поскольку схема перевозок дает полную картину не только процесса перевозки контейнеров, но и план их погрузки-выгрузки в узловых предприятиях, на ее основе разрабатываются информационные массивы управления контей-

нерным терминалом (КТ), определяющие порядок обработки исходящих, входящих и транзитных контейнеров.

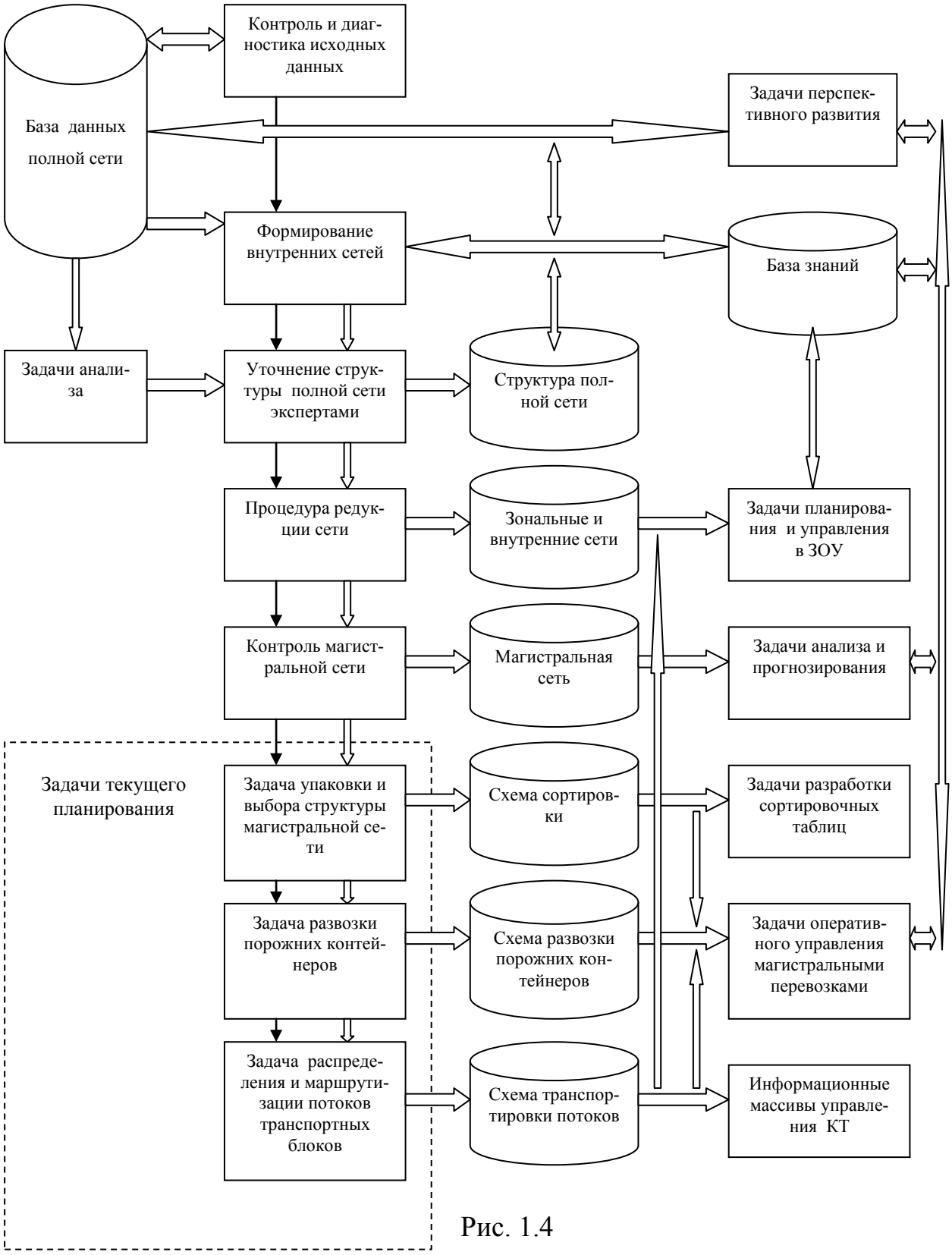


Рис. 1.4

Совместное использование результатов решения задач анализа, прогнози-

рования и проектирования схемы перевозок позволяет решать задачи перспективного развития транспортной системы. Так, например, результаты решения задач текущего планирования, при использовании прогнозируемых на перспективу магистральных потоков являются исходными данными для развития узловых предприятий и построения новых маршрутов магистрального транспорта (раздел 5).

Важным вопросом в системе управления транспортной системой является выбор дискреты времени планирования для различных задач. Как правило, перспективные планы разрабатываются на пятилетия и более долгосрочные периоды. Для задач текущего планирования характерно построение схем перевозок для нескольких периодов на протяжении года. На практике наиболее часто разработка схемы магистральных перевозок выполняться пять раз в год в зависимости от сезонных колебаний потоков. При этом в качестве мгновенных потоков, распределяемых по сети, рекомендуется использовать среднесуточные планируемые потоки, рассчитанные для конкретных периодов. Использование среднесуточных потоков в задачах текущего планирования связано с двумя важными факторами, сопутствующими обработке и перевозке мелкопартионных грузов: ритмичностью производственных процессов и стремлением связать текущее планирование с задачами оперативного управления, где в качестве дискреты времени принимаются одни сутки. Первый фактор наиболее сильно проявляется при сортировке потоков и связан с контрольными сроками обработки грузов в узловых предприятиях, а также с расписанием движения магистрального транспорта. В этом смысле ежедневное движение транспортных средств через узловые предприятия на всех маршрутах является наиболее приемлемым, так как позволяет установить четкую ритмичность отправки контейнеров, сформированных за сутки. Второй фактор обусловлен тем, что иногда в реальных условиях управления граница между текущими и оперативными решениями стирается. Например, задача распределения порожних контейнеров должна решаться как на уровне текущего планирования, так и при оперативном управлении. Обобщая сказанное, можно сделать вывод о том, что взаимосвязь

задач, наличие обратной связи в системе управления, стирание граней в отдельных случаях между задачами текущего планирования и оперативного управления позволяет гибко управлять процессами перевозки на всех уровнях иерархии сети.

Выводы по первому разделу

1. Приведены методологические основы построения распределенной иерархической многопродуктовой коммуникационной сети с мелкопартионными дискретными потоками корреспонденций. Определены основные принципы и технология организации сортировки и транспортировки потоков в зонально-узловой структуре сети. В отличие от существующих подходов, предложенная методология основывается на комплексном решения задач долгосрочного, текущего и оперативного планирования и управления и позволяет логически увязать процессы обработки потоков в узлах сети с их последующим распределением на сети.

2. Для управления нелинейными и нестационарными процессами обработки и распределения потоков в сети предложен концептуальный подход к построению многоуровневой информационно-аналитической системы поддержки принятия решений, функционирующей в режиме реального времени и открывающей новые возможности для создания интеллектуальных методов и информационных технологий в управлении сложными процессами и объектами.

3. Определены основные задачи текущего планирования, составляющие методологическую основу — ядро для решения задач оперативного управления процессами обработки и распределения потоков, а также для решения задач перспективного развития сети. Показана взаимосвязь задач текущего планирования с задачами оперативного управления и перспективного развития.

4. Определены основные задания по разработке комплекса мер информационного обеспечения и автоматизации процессов принятия решений для управления процессами обработки и распределения потоков в иерархической сети (Приложение А).

РАЗДЕЛ 2. ЗАДАЧИ МАРШРУТИЗАЦИИ ВНУТРИУЗЛОВЫХ ПЕРЕВОЗОК

В разделе обсуждаются вопросы организации перевозок во внутренней зоне магистрального узла, приводятся основные принципы и схемы, заложенные в основу построения маршрутов транспортных средств. Рассматриваются технические и экономические особенности и характеристики реальных транспортных процессов, которые должны быть учтены при формировании целевых функций математических моделей задач маршрутизации на уровнях долгосрочного и текущего планирования и оперативного управления. Приводится классификация задач маршрутизации по основополагающим признакам и библиографический обзор литературы, посвященной их решению. Предложено несколько вариантов математических постановок задачи, которые основаны на известных типовых моделях и могут быть использованы при проектировании рабочего парка и маршрутов транспортных средств для обслуживания узлов четвертого типа во внутренней зоне магистрального узла. Обсуждаются методы, алгоритмы и пакеты программ, применяемые в настоящее время для решения задач подобного класса. Сформулированы основные выводы.

Материалы этого раздела составляют методологическую основу для разработки современного математического обеспечения решения задач долгосрочного и текущего планирования и оперативного управления во внутренних зонах магистральных узлов.

При изложении материала раздела используются более общие термины: транспортное средство — ТС (автомобили и автомобильные прицепы); депо (места размещения парка ТС во внутренней зоне магистрального узла первого, второго или третьего типа); клиенты (узлы четвертого типа во внутренней зоне магистрального узла).

Результаты раздела опубликованы в работах [16, 17].

Ключевые слова: задачи комбинаторной оптимизации, математические

модели маршрутизации транспортных средств, автотранспортные перевозки, мелкопартионные грузы

2.1. Организация внутриузловых перевозок

2.1.1. Основные принципы и схемы перевозок

Прежде всего, приведем основные принципы организации внутриузловых перевозок:

- для выполнения перевозок используются только кольцевые (замкнутые, циклические) маршруты ТС с одним или несколькими депо во внутренней зоне магистрального узла;
- каждый клиент может обслуживаться только одним ТС (возможно с одним или двумя прицепами);
- кузова автомобилей и прицепов должны быть оборудованы так, чтобы давать возможность погрузки и выгрузки грузов с боковых бортов и заднего борта кузова;
- все ТС и грузовые поддоны после обслуживания клиентов должны возвращаться в депо. Загруженный в магистральном узле прицеп может быть оставлен на некоторое время в местах дислокации клиентов для выгрузки и погрузки больших объемов грузов. Загруженный клиентом прицеп возвращается в магистральный узел и депо тягловым ТС при очередном посещении клиента;
- могут использоваться ТС разных типов с разной грузоподъемностью и начальной продажной стоимостью, т.е. допускается неоднородность (гетерогенность) рабочего парка транспортных средств (РП ТС);
- в течение рабочего дня одно и то же ТС может быть использовано для перевозки грузов несколько раз и не обязательно по одному и тому же маршруту;
- в отдельных случаях при оперативном перераспределении потоков грузов и маршрутов ТС из-за возникновения временных перегрузок и непредвиденных ситуаций в сети перевозок допускается обслуживание любого клиента несколькими ТС;

• магистральный узел сортирует грузы (поступающие в него из магистральной сети и адресованные во внутреннюю зону узла) по местам дислокации клиентов во внутренней зоне. Отсортированные грузы могут быть сгруппированы и погружены на поддоны для каждого клиента. Грузы на поддонах перевозятся только в грузовых отсеках автомобиля, доступ к которым возможен со всех бортов. Для мелких партий грузов, перевозимых без поддонов, должна применяться погрузка по правилу «последний пришел, первый вышел» (Last Input, First Output, LIFO). Большие партии грузов, предназначенные одному клиенту, перевозятся общей массой (навалом), без упорядочения;

• клиенты не сортируют грузы по адресам назначения, а отправляют их в магистральный узел, загружая ТС общей массой (навалом), поэтому обмен грузами между клиентами во внутренней зоне возможен только через магистральный узел этой зоны;

• все грузы из магистрального узла к клиентам и все грузы от клиентов в магистральный узел должны быть доставлены в течение 24 часов (принимается восьмичасовой рабочий день водителей в дневную и ночную смену). Здесь предполагается, что во внутреннюю зону каждого магистрального узла включены только такие клиенты, которые могут быть обслужены в течение рабочего дня водителя с учетом времени возвращения ТС в депо;

• могут быть заданы жесткие временные горизонты (временные окна – Time Window, TW) для отправления и получения грузов в магистральном узле и для получения и отправления грузов в местах дислокации клиентов.

На рис. 2.1 проиллюстрированы фрагменты внутриузловой сети, схематично показаны циркулирующие потоки и пример кольцевых маршрутов.

Для организации перевозок во внутренних зонах магистральных узлов можно применять в основном следующие три схемы перевозки грузов на проектируемых маршрутах ТС:

• доставочные маршруты для перевозки грузов из магистрального узла к клиентам;

• сборочные маршруты для перевозки грузов от клиентов к магистрально-

му узлу;

• комбинированные маршруты, на которых разрешено выгружать и погружать грузы в местах дислокации клиентов.

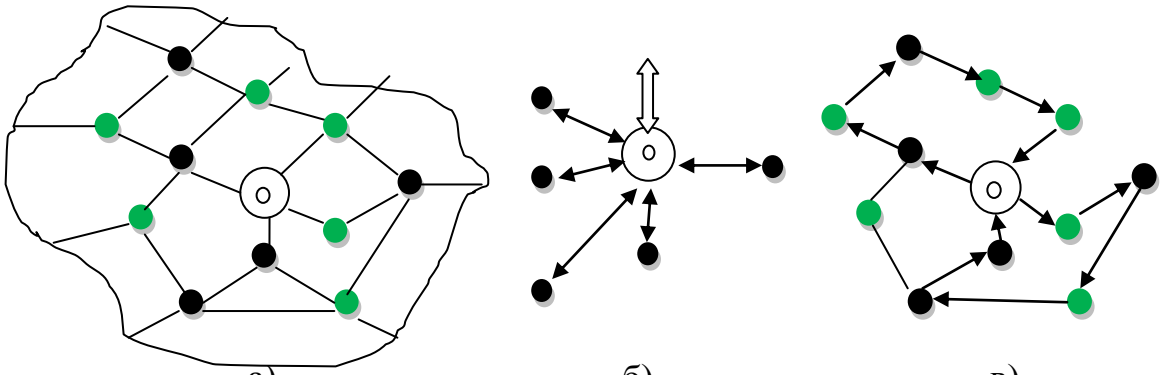


Рис. 2.1. а) внутриузловая сеть с центральным магистральным узлом и депо, места дислокации клиентов закрашены черным цветом, а дорожные узлы – зеленым, участки дорог показаны линиями; б) входящие и исходящие магистральные потоки (широкая стрелка), и внутриузловые потоки к клиентам и от клиентов (простая стрелка); в) два циклических маршрута транспортных средств.

2.1.2. Функции затрат и содержательная постановка задачи

В этом подразделе обсуждаются вопросы, связанные с методикой расчета приведенных затрат при решении задачи маршрутизации внутриузловых перевозок на долгосрочные и среднесрочные периоды времени. Целью решения такой задачи является определение общего количества и состава транспортных средств по типам и грузоподъемности, необходимого для выполнения всех перевозок во внутренней зоне магистрального узла при минимизации капитальных затрат на приобретение транспортных средств и эксплуатационных расходов на транспортировку грузов. При решении задачи нужно определить рабочий парк неоднородных транспортных средств (гетерогенный рабочий парк) и найти схему распределения потоков грузов и маршрутизации транспортных средств во внутриузловой сети перевозок. В зарубежной литературе эти задачи принято называть Fleet Size and Mix Vehicle Routing (FSMVRP) и Heterogeneous

Fleet Vehicle Routing Problem (HFVRP) в зависимости от ограничений на количество транспортных средств каждого типа. В дальнейшем будем использовать эти аббревиатуры для краткости изложения материала по всем вопросам решения таких задач.

Для транспортных предприятий (компаний) ключевыми факторами при расчете затрат являются ожидаемые объемы перевозок, цены на транспортные средства и себестоимость перевозок. Транспортные средства с большей грузоподъемностью, как правило, имеют более низкую себестоимость перевозки единицы груза, чем транспортные средства с меньшей грузоподъемностью, при условии, что коэффициент загрузки транспортного средства достаточно высок. Следует учитывать и то, что для подержанных транспортных средств затраты на амортизацию ниже, а затраты на содержание выше, чем у новых. Объемы перевозок и цены на транспортные средства изменяются во времени, поэтому у транспортных предприятий возникает и задача управления размером рабочего парка не только на периоды текущего планирования, но и на перспективу. Чрезмерно большой парк при снижении спроса на перевозки вынуждает продавать или сдавать в аренду излишек транспортных средств, увеличение спроса приводит к необходимости приобретения новых транспортных средств или их аренды. В любом случае при долгосрочном планировании ожидаемые доходы должны быть больше ожидаемых затрат. Регулирование размера рабочего парка, зависящего от приведенных и других случайных факторов должно осуществляться при решении динамических стохастических задач долгосрочного планирования, математические модели которых более агрегированные, чем у задач текущего планирования. Соответственно в этом случае используются и более агрегированные функции приведенных затрат.

В математических моделях текущего (тактического) планирования, как правило, рассматриваются усредненные потоки, и цены для заданных периодов времени на протяжении года, рассчитанные статистическими методами. Неопределенность в этом случае уменьшается, и задача маршрутизации должна рассматриваться на более детальном уровне с учетом всех значимых факторов. Как

и при долгосрочном планировании, в условиях колебания спроса, основные решения связаны с приобретением и арендой транспортных средств или с продажей и сдачей в аренду имеющихся излишек. Однако при принятии тактических решений следует больше уделять внимания корректировке грузоподъемности транспортных средств. Поэтому при решении задач маршрутизации на среднесрочные периоды, как правило, создаются резервы грузоподъемности транспортных средств на всех маршрутах транспортировки грузов, т.е. может быть задан коэффициент максимальной загрузки транспортного средства (ТС не загружаются «под завязку» с коэффициентом загрузки равным единице). Это приводит к использованию транспортных средств с большей грузоподъемностью и увеличению капитальных и эксплуатационных затрат, но позволяет выполнять заказы клиентов в условиях колебания спроса в определенных пределах. Задача состоит в определении «золотой середины». При текущем планировании точно известны размер и состав РП ТС, подробная схема распределения потоков грузов и транспортных средств по маршрутам движения. Известны также заключенные контракты, и тарифы на перевозку грузов и другие виды операций, влияющих на текущее финансовое состояние транспортного предприятия. В этой связи при неопределенном колебании спроса может быть поставлена задача оптимизации РП ТС и портфеля заказов — необходимо отобрать наиболее выгодные контракты так, чтобы их можно было бы выполнить при имеющемся потенциале провозных возможностей транспортного предприятия. Следует отметить, что не во всех случаях при текущем планировании нужно объединять решение задачи определения РП ТС с решением задачи управления активами и портфелем заказов транспортного предприятия, так как последняя может быть решена независимо при известных колебаниях нагрузок в сети перевозок.

На оперативном уровне задача транспортного предприятия, как правило, состоит в перераспределении потоков и реоптимизации схемы маршрутизации при превышении провозной возможности или возникновении отказов у клиентов и на маршрутах движения, а также при различных непредвиденных ситуа-

циях и стихийных бедствиях. В этом случае может быть использован внутренний резерв транспортных средств или привлечены арендные транспортные средства.

Для транспортных предприятий чрезвычайно важно, чтобы при решении задач текущего планирования они могли бы получить реальные оценки транспортных издержек. Поэтому в математических моделях, используемых для текущего планирования перевозок, в функциях затрат должны отражаться основные статьи эксплуатационных расходов, связанные с содержанием парка транспортных средств и транспортировкой грузов.

В большинстве зарубежных и отечественных работах, посвященных решению задач замкнутой маршрутизации (построению циклических маршрутов), авторы рассматривают затраты на дугах маршрута как заданные константы, и предполагают, что эти затраты не зависят от типа и грузоподъемности транспортного средства. Более того, принимается также, что затраты одинаковы при прохождении транспортной дуги в прямом и обратном направлении. Однако такие допущения не реалистичны. На практике затраты на дугах зависят от длины, состояния и географических особенностей участков дороги, типа и грузоподъемности транспортного средства, его текущей загрузки и скорости движения. Все это, в свою очередь, влияет на расход топлива. Кроме того, при транспортировке грузов нужно учитывать дополнительные издержки, связанные со случайными факторами — вынужденными простоями транспортных средств из-за непредвиденных ситуаций на дороге, стихийных проявлений природы и др. Дополнительные издержки на практике обычно проявляются в виде штрафных санкций за нарушение сроков доставки грузов, их утерю или повреждение.

В задачах построения циклических маршрутов всегда известна начальная и конечная точка маршрута, поэтому будут найдены все узлы и дуги, через которые проходит маршрут. Это дает возможность, при условии аддитивности затрат, задавать их на дугах (i, j) и в узлах i функциями f_{ij}^{k} и f_{i}^{k} для каждого ТС типа k. В эти функции кроме основных параметров, определяющих длину дуги, грузоподъемность и скорость движения ТС, должны быть включены коэф-

фициенты, отражающие фактическую загрузку ТС на дуге и ее географические особенности (например, подъем или спуск соответствующего участка дороги). Функции f_{ij}^{k} могут зависеть и от одного, обобщенного параметра — расхода топлива на дуге (i, j). Вычисляемые в процессе решения задачи значения этих функций должны определять реальные эксплуатационные затраты на перевозку грузов по всему маршруту, проходящему по физическим участкам дорог (см. рис. 2.1а, в).

Ясно, что расходы на приобретение, содержание и эксплуатацию рабочего парка транспортных средств при решении задачи маршрутизации должны рассчитываться только с учетом тех параметров, которые явно или неявно входят в математическую модель задачи. Поэтому эти расходы будут составлять только часть себестоимости перевозок транспортного предприятия. Напомним, что себестоимость перевозок — это выраженная в денежной форме величина эксплуатационных расходов транспортного предприятия, приходящихся в среднем на единицу продукции транспорта. На грузовом автомобильном транспорте себестоимость перевозок определяется, как правило, за перевезенную тонну грузов на один километр (т/км) или за использование одного автомобиля в час (а/ч). Расчет себестоимости необходим для определения тарифов на перевозку грузов и ожидаемой прибыли предприятия. Оптимизация маршрутов перевозки грузов позволяет снизить себестоимость перевозок в основном за счет устранения нерациональных маршрутов, увеличения коэффициента загрузки и уменьшения потребляемого топлива и порожнего пробега ТС, уменьшения времени доставки грузов потребителю.

Поскольку после решения задачи известны капитальные затраты на приобретение РП ТС, то возникает вопрос: какие статьи эксплуатационных затрат на перевозку нужно учитывать в общей формуле расчета приведенных затрат при решении задачи? Представляется разумным, что для расчета эксплуатационных затрат в формуле необходимо учитывать только те статьи затрат, которые непосредственно связаны с перевозкой грузов, а общехозяйственные (накладные) расходы предприятия можно не учитывать. Перечислим основные статьи этих

затрат: топливо, смазочные и прочие эксплуатационные материалы; техническое обслуживание и ремонт подвижного состава (автомобилей и прицепов); амортизация по восстановлению подвижного состава; износ и ремонт автомобильной резины; заработная плата водителей вместе с отчислениями на социальные нужды.

Помимо перечисленных, в расчетной общей формуле приведенных затрат могут быть учтены и другие статьи эксплуатационных затрат, связанные с движением (дорожные сборы, оплата платных дорог и пр.). Главное — чтобы величина эксплуатационных затрат, рассчитанная по выведенной формуле при решении задачи, была бы как можно ближе сопоставима (адекватна) с величиной той части затрат, которые были фактически понесены предприятием только на перевозку грузов по оптимизированным маршрутам движения транспортных средств. Определение общей формулы расчета приведенных затрат представляет отдельную, непростую задачу для экономистов транспортного предприятия, которая должна быть решена до проведения численного решения задачи нахождения состава и количества транспортных средств, распределения потоков грузов и маршрутизации транспортных средств. Во многих практических случаях функции приведенных затрат нелинейные, несепарабельные и неаддитивные, и это следует учитывать при построении математической модели конкретно рассматриваемой задачи.

2.2. Классификация задач маршрутизации, краткий обзор методов и алгоритмов их решения

Трудно утверждать, что единственная формулировка задачи будет достаточной для всех реальных приложений. В частности, как уже упоминалось, для долгосрочного планирования из-за неопределенности информации существует большая вероятность принятия различных решений о возможном составе РП ТС. Для таких задач должны использоваться более агрегированные модели распределения потоков грузов и маршрутизации ТС. В задачах текущего планирования и оперативного управления детализация моделей маршрутизации должна

увеличиваться для получения оценок приведенных затрат, адекватных затратам, понесенным на процессы обработки и транспортировки грузов в реальных сетях.

В большинстве известных работ, посвященных решению задач маршрутизации на транспорте, рассматриваются идеализированные (академические) математические модели, в которых не учитываются многие факторы, присущие реальным процессам обработки и транспортировки грузов. Например, такие важные параметры как расстояние, время поездки, транспортные расходы, время обслуживания и другие, присутствуют в некотором абстрактном виде. Во многих моделях принимается евклидово расстояние между узлами сети перевозок, когда выполняется правило треугольника. Скорость и затраты, за немногими исключениями стохастических моделей, моделируются константами. В реальной жизни для внутриузловых перевозок должны рассчитываться действительные расстояния, время и приведенные затраты на транспортировку грузов. В частности, в городских районах, представляющих внутреннюю сеть перевозок, время в пути является весьма неопределенным, так как скорость транспортных средств может изменяться с течением времени из-за перегрузки улиц. Поэтому для расчета времени доставки грузов представляется разумным использовать среднюю скорость движения транспортных средств, однако надо учитывать, что с увеличением средней скорости на объездных дорогах могут возрасти эксплуатационные затраты. В любом случае в моделях внутриузловых перевозок нужно учитывать зависимость между скоростью и затратами. Кроме того при расчете времени доставки клиентам необходимо учитывать время на сортировку и погрузку мелкопартионных грузов в магистральном узле и время на разгрузку и погрузку грузов в местах дислокации клиентов.

Часто рассматриваются задачи, в которых рабочий парк транспортных средств является однородным. На практике, как правило, транспортные предприятия имеют неоднородный парк транспортных средств. Есть и другие важные производственные особенности помимо неоднородности парка, так например, задача маршрутизации должна рассматриваться вместе с задачей оптими-

зации распределения водителей между транспортными средствами с учетом ограничений на длительность рабочего дня. В классической постановке задачи маршрутизации, существует единое депо, где каждое транспортное средство начинает и заканчивает движение. При этом для каждого транспортного средства допускается только один циклический маршрут (тур) движения. В реальных задачах транспортировки грузов, может быть несколько депо, и водители могут выполнять множество различных туров в день.

Приведенные особенности характерны для внутриузловых перевозок, когда транспортные средства доставляют грузы между магистральным узлом и клиентами внутренней зоны и наоборот и, возможно, между различными клиентами внутренней зоны.

Можно условно выделить несколько основных классов задач маршрутизации на транспорте, которые наиболее часто рассматриваются в литературе:

- классическая задача маршрутизации транспортных средств (The Vehicle Routing Problem, VRP) и ее разновидности:

простая пикап - доставка (Vehicle Routing Problem with Pickup and Delivery, VRPPD);

доставка с упорядочением загружаемых грузов по правилу «последним пришел, первым вышел» (VRPPD with LIFO);

доставка с временными окнами (Vehicle Routing Problem with Time Windows, VRPTW);

доставка с ограниченной грузоподъемностью транспортных средств на маршрутах или с ограниченной пропускной способностью дуг сети (Capacitated Vehicle Routing Problem, CVRP или Capacitated Arc Routing Problem, CARP);

доставка с одновременной сборкой грузов для депо (Vehicle Routing Problem with Simultaneous Delivery and Pick-up, VRPSDP);

доставка с неоднократным посещением клиентов (Split Delivery Vehicle Routing Problem, SDVRP);

доставка, когда одни и те же транспортные средства могут использоваться многократно на разных маршрутах (Vehicle Routing Problem with Multiple Trips,

VRPMT);

доставка, когда транспортные средства могут иметь более одного депо (Vehicle Routing Problem with Multiple Depots, VRPMD);

открытую задачу маршрутизации, когда транспортные средства не обязаны возвращаться в депо (Open Vehicle Routing Problem, OVRP);

задачу маршрутизации грузовика с прицепом (The Truck and Trailer Routing Problem, TTRP);

задачу маршрутизации транспортных средств с прицепами, когда прицепы могут оставаться в узлах под разгрузку и погрузку (The Rollon-Rolloff Vehicle Routing Problem, RRVRP).

- обобщенная задача формирования (определения размера и состава) гетерогенного рабочего парка транспортных средств и схемы распределения потоков грузов и маршрутизации транспортных средств (Heterogeneous Fleet Vehicle Routing Problem, HFVRP или The Fleet Size and Mix Vehicle Routing Problem, FSMVRP) и ее разновидности в зависимости от ограничений на размер парка;
- обобщенная задача формирования гетерогенного рабочего парка транспортных средств, размещения депо и схемы распределения потоков грузов и маршрутизации транспортных средств (The Fleet Size and Mix Location Routing Problem, FSMLRP);
- задача проектирования сети (Network Design Problems, NDP) и задача сервисного обслуживания сети (Service Network Design problems, SND) в которой маршрутизация связана с частотой, режимами и графиками проведения сервисных работ.

Второй класс задач иногда обобщают под названием — Fleet Composition and Routing Problems, FCRP. На практике все эти задачи могут быть связаны между собой и образовывать различные гибриды в зависимости от необходимости учета фактических ограничений. Рассмотренные классы задач можно разделить и упорядочить по нескольким основополагающим признакам. Например, таким как количество депо и клиентов, технические характеристики ТС и физические характеристики сети дорог и перевозимых грузов (непрерывные и

дискретные потоки, однородные и смешанные грузы), схемы и условия перевозки грузов, ограничения по времени и т.п.

Впервые задача VRP была сформулирована в 1959 г. Данцигом и Рамсером (G.B. Dantzig, J.H. Ramser) и впоследствии стала называться классической. Эта задача является обобщением широко известной NP-трудной задачи коммивояжера (Traveling Salesman Problem, TSP), для которой до сих пор неизвестны методы и алгоритмы, позволяющие за полиномиальное время находить точные или приближенные решения с заданной оценкой погрешности [429] (теорема S. Sahni, T. Gonzalez, 1976 г.). Для метрической TSP в 1976 г. Кристофидесом (N. Christofides) [211] был предложен алгоритм с оценкой погрешности не более 50% и этот результат пока не улучшаем (однако имеются аномальные примеры, когда алгоритм Кристофидеса дал отклонение от точного решения на 150% [216]). NP-трудность классической задачи VRP была доказана в 1981 г. Ленстрой и Каном (J.K. Lenstra, A.H.G. Rinnooy Kan) [364]. Поэтому в настоящее время утвердилось мнение, что для большинства разновидностей задачи VRP и общей задачи TSP не существует полиномиальной и, более того, полностью полиномиальной приближенной схемы решения (Polynomial and Fully Polynomial Time Approximation scheme, PTAS and FPTAS), если только P≠NP. Напомним, что PTAS определяет класс алгоритмов, которые для любого наперед заданного вещественного $\varepsilon > 0$, позволяют находить $1+\varepsilon$ константное приближенное решение за полиномиальное время от размера входа любой индивидуальной задачи (экземпляра оптимизационной задачи) и ε. Аппроксимационные алгоритмы PTAS могут быть экспоненциальными по $1/\varepsilon$. Алгоритмы класса FPTAS находят $1+\varepsilon$ приближенное решение (для задач на минимум) за полиномиальное время от размера входа любой индивидуальной задачи и от $1/\varepsilon$.

В 1964 г. Кларком и Райтом (G. Clarke and J. W. Wright) впервые был предложен приближенный алгоритм для решения задачи VRP. В последующие годы началось интенсивное исследование методов и алгоритмов решения этой задачи и ее разновидностей [358].

Для задач с небольшим числом клиентов $n \le 135$, были разработаны точ-

ные алгоритмы, основанные на методах динамического программирования и ветвей и границ (1981, Christofides, Mingozzi, Toth); ветвей, отсечений и цен (branch-and-cut, branch-and-cut-and-price — 1985, Laporte, Nobert, Desrochers; 1994, Fisher, Ralphs et al., Augerat et al.,; 2000, Blasum, Hochstättler; 2006, Fukasawa et al.; 2008, Baldacci, Christofides, Mingozzi); сетевых методах (2004, Baldacci, Hadjiconstantinou, Mingozzi).

К классическим приближенным (эвристическим) алгоритмам можно отнести [359, 360]: *конструктивные алгоритмы*, которые постепенно наращивают решение без последующего этапа его улучшения; *двухэтапные кластерные алгоритмы*, сочетающие группировку клиентов в кластер для построения маршрута и решение задачи TSP для каждого кластера (существуют две разновидности — TSP решается для всего множества клиентов и затем строятся маршруты и наоборот, когда сначала определяются кластеры, а затем маршруты); *улучшающие алгоритмы*, в которых находится начальное допустимое решение, а затем оно улучшается на множестве комбинаторных перестановок узлов и дуг для одного или нескольких проектируемых маршрутов. Среди этих алгоритмов наиболее известными являются: savings (1965, Clarke, Wright); sweep (1974, Gillett, Miller); cluster first, route second (1981, Fisher, Jaikumas); intra-route improvement methods based on TSP heuristics (interchanges — 1993, Osman; cyclic exchanges — 1993, Thompson, Psaraffis; edge exchange schemes — 1997, Kindervater, Savelsbergh; ejection chains —1996, 1998, Xu, Kelly, Rego, Roucairol; very large neighbourhood search — 2003, Ergun et al.; SERR — 2004, De Franceschi, Fischetti, Toth).

С середины 1990-х годов и по настоящее время для решения различных задач VRP активно разрабатываются метаэвристические алгоритмы. В эти алгоритмы заложена способность преодоления локальных экстремумов для расширенного поиска наилучшего решения. Поэтому в отдельных случаях и для некоторых индивидуальных задач они могут оказаться эффективнее (по быстродействию и точности получаемого решения) по сравнению с классическими алгоритмами.

Принято разделять метаэвристические алгоритмы на три основные группы [270, 271]: *локального поиска* (local search) — моделируемый отжиг (Simulated Annealing, SA), детерминированный отжиг (Deterministic Annealing, DA), поиск с запретами (Tabu Search, TS); *эволюционные* (population search), основанные на генерации популяций особей — процедуры с адаптивной (адаптивным управлением) памятью (adaptive memory procedures), генетические (genetic search, Genetic Algorithms, GA); *самообучающиеся* (learning mechanisms) — нейронные сети (neural networks), муравьиные колонии (ant colony systems). К основным метаэвристикам относят также процедуру жадного адаптивного рандомизированного поиска (Greedy Randomized Adaptive Search Procedure, GRASP) и поиска (спуска) с чередующимися окрестностями (Variable Neighborhood Search (Descent), VNS, VND).

Исследованием метаэвристических алгоритмов занимались очень многие зарубежные ученые, поэтому выделим только некоторых из них:

поиск с запретами, моделируемый и детерминированный отжиг — Willard (1989), Gendreau, Hertz, Laporte (1991), Taillard, Osman (1993), Gendreau, Hertz, Laporte (1994), Cordeau, Laporte, Mercier (2001), Toth, Vigo (2003), Li, Golden, Wasil (2004);

эволюционные алгоритмы — Rochat, Taillard (1995), Rego, Roucairol (1996), Tarantilis, Kiranoudis (2002), Prins, Mester and Bräysy (2004), Mester, Bräysy (2005);

самообучающиеся алгоритмы — Reinmann, Doerner, Hartl (2004);

комбинированные алгоритмы — Very large neighbourhood search (Ergun, Orlin, Steele-Feldman, Ergun et al., 2003, 2006), Attribute based hill climbing (Derigs, Kaiser, 2007), Genetic search + very large neighbourhood search (Mester, Bräysy, 2007), Guided very large neighbourhood search (Kytöjoki et al., 2007), Adaptive very large neighbourhood search (Pisinger, Ropke, 2007), Memetic algorithm (Nagata, 2007), Local search limitation strategies (Nagata, Bräysy, 2008), Memetic algorithm (Nagata, Bräysy, 2009), Greedy Randomized Adaptive Search Procedure (GRASP) + Evolutionary search (Prins, 2009).

В странах СНГ исследованиями задач маршрутизации занимались И.И. Меламед, И.Х. Сигал, С.И. Сергеев, А.А. Лазарев, Е.М. Бронштейн, Э.Х. Гимади, Л.Ф. Гуляницкий, Д.И. Соломон, А.В. Панишев и др.

Среди последних отечественных работ и работ ближнего зарубежья, посвященных исследованию задач VRP, отметим учебное пособие Л.Ф. Гуляницкого и О.Ю. Мулеса [44], монографию А.В. Панишева и А.В. Морозова [94] и диссертации М.С. Пожидаева [99], С.В. Чернышева [133], Р.В. Гиндуллина [41], А.К. Перцовского [97] и А.В. Хмелева [132].

Различные варианты классификации задач VRP и более поздние обзоры по методам, алгоритмам и программам их решения можно найти в статьях [7, 181, 191, 226, 248, 356, 414, 415, 423], монографиях [286, 459] и на сайте http://neo.lcc.uma.es/vrp/bibliography-on-vrp/.

В отличие от классической задачи VRP и ее разновидностей, задачи семейства FCRP изучались менее интенсивно. В этих задачах рассматриваются не только транспортные расходы, но и расходы, связанные с приобретением неоднородного (гетерогенного) рабочего парка транспортных средств и они наиболее интересны для проектирования перевозок во внутренних зонах магистральных узлов. В зарубежной литературе принято разделять задачи с неоднородным парком в зависимости от ограниченности или неограниченности количества транспортных средств каждого типа, учета фиксированной стоимости транспортных средств и переменной стоимости транспортировки грузов, а также от того зависят ли цены на транспортировку от типа транспортных средств. Согласно зарубежной классификации выделено пять основных классов задач FCRP:

Heterogeneous VRP with Fixed Costs and Vehicle Dependent Routing Costs (HVRPFD) — ограниченный парк, фиксированная и переменная стоимость учитываются, цены зависят от типа транспортного средства;

Heterogeneous VRP with Vehicle Dependent Routing Costs (HVRPD) — ограниченный парк, учитывается только переменная стоимость, цены зависят от типа транспортного средства;

Fleet Size and Mix VRP with Fixed Costs and Vehicle Dependent Routing Costs

(FSMFD) — неограниченный парк, фиксированная и переменная стоимость учитываются, цены зависят от типа транспортного средства;

Fleet Size and Mix VRP with Fixed Costs (FSMF) — неограниченный парк, фиксированная и переменная стоимость учитываются, цены не зависят от типа транспортного средства;

Fleet Size and Mix VRP with Vehicle Dependent Routing Costs (FSMD) — неограниченный парк, учитывается только переменная стоимость, цены зависят от типа транспортного средства.

Первыми исследованиями задач FCRP принято считать работы G.B. Dantzig, D.R. Fulkerson (1954) [223] и D. Kirby (1959) [343]. В 1984 г. вышла обзорная статья B.L. Golden, A. Assad и др. [287], где обсуждаются возможные модели и методы исследования операций, применимые к решению задач FCRP. В частности в этой работе рассматривается задача с неограниченным парком транспортных средств, известная также как Fleet Size and Mix (FSM), в которой определяются оптимальные состав и схема маршрутизации транспортных средств. Для решения задачи авторами предложены два эвристических алгоритма, основанных на savings-алгоритме Кларка и Райта (G. Clarke и J.W. Wright) [212] и построении большого тура коммивояжера. Они также сформулировали математическую модель для задачи FSMF и представили некоторые нижние границы ее решения.

Обзор ранних работ по задачам FCRP приведен S. Salhi и G.K. Rand в 1993 г. [430]. I.H. Osman и S. Salhi [407] исследовали и проанализировали работы до 1996 г., посвященные решению задач с помощью методов локального поиска. Y.H. Lee и др. (2008) [363] описывают несколько эвристических подходов к решению различных вариантов задач FCRP, основанных на методах поиска с запретами (tabu search) и разбиения множества (set partitioning). Более поздние обзоры по задачам формирования неоднородного парка транспортных средств приведены R. Baldacci и др. [167, 168] и A. Hoff и др. [307, 308]. В обзорах рассматриваются в основном работы, связанные с гетерогенным ограниченным и неограниченным размером парка транспортных средств с различным сочетани-

ем учета фиксированной стоимости транспортных средств и переменной стоимости транспортировки грузов. При этом особое внимание уделяется методам получения нижних оценок решения задач и эвристическим алгоритмам. Обсуждается ряд приближенных алгоритмов, разработанных в разное время Gheysens (1984), Golden (1984), Gheysens (1986), Desrochers и Verhoog (1991), Salhi и Rand (1993), Osman и Salhi (1996), Ochi (1998), Taillard (1999), Gendreau (1999), Liu и Shen (1999), Wassan и Osman (2002), Dullaert (2002), Renaud и Boctor (2002), Tarantilis и др. (2003, 2004), Lima, Goldbarg и Goldbarg (2004), Lee, Kim, Kang и Kim (2006), Yaman (2006), Choi и Tcha (2007), Pessoa, Poggi de Aragão и Uchoa (2007), Brandão (2007), Dell'Amico (2007), Li и др. (2007), и приводится сравнительный анализ их вычислительной эффективности на типовых контрольных примерах, общедоступных в сети Интернет.

Методы получения нижних границ решения задачи были предложены Golden и др. (1984) [287], Yaman (2006) [478] и Choi и Tcha (2007) [210]. В работе [478] на основе методов отсечений предложена методика получения нижних границ для шести математических формулировок задачи FSMF. Для последних двух формулировок, в которых введены трех - индексные потоковые переменные для типов транспортных средств и заданной пары клиентов получены наилучшие результаты. Для устранения подциклов, не проходящих через депо, использовались потоковые переменные, а для улучшения нижних границ вводились дополнительные ограничения в виде «правильных» отсекающих неравенств. В [210] для задачи, сформулированной на основе разбиения множества (set partitioning formulation), предложена методика вычисления нижних границ решения с помощью метода генерации столбцов. В этом случае множеству генерируемых столбцов взаимно однозначно соответствует множество маршрутов, где каждый маршрут представлен циклом, включающим депо и некоторое подмножество клиентов с суммой заказов, не превышающей грузоподъемности транспортного средства, выделенного на маршрут. Используя такой подход, авторы разработали очень эффективный эвристический алгоритм, который комбинирует столбцы в оптимальном решении релаксированной задачи. В [416]

для решения задач FCRP приведен робастный алгоритм ветвей, отсечений и цен (branch-cut-and-price algorithm), основанный на алгоритме Fukasawa и др. (2006) [258]. В сформулированной задаче используется метод генерации столбцов с новым семейством отсекающих неравенств, улучшающих оценки нижних границ решения. Отмечается, что алгоритм позволяет найти точное решение для многих тестовых задач, в которых число клиентов не превышает 75.

Одни из последних работ R. Baldacci и др. [169-172] посвящены исследованию вопросов улучшения нижних границ, обзору точных методов и сравнению вычислительной эффективности алгоритмов решения задач VRP и FCRP. В [170] предложен один из лучших в настоящее время точный алгоритм, основанный на разбиении множеств и позволяющий находить оптимальные решения задач FCRP с числом клиентов до 100. В алгоритме используется три типа процедур лагранжевой и LP-релаксации исходной математической формулировки задачи, которые позволяют значительно сократить число переменных задачи и использовать для ее решения известные пакеты программ целочисленного линейного программирования. В статье приводятся результаты вычислений для тестовых экземпляров задач FCRP, которые показали новые лучшие нижние границы решения для отдельных индивидуальных задач, и что предложенный точный алгоритм впервые позволил решить несколько не решенных ранее тестовых примеров.

В заключение обзора приведем несколько работ, посвященных метаэвристическим алгоритмам (см. Gendreau и др. «Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography» в [286] стр. 143-169). Широко известны метаэвристики, основанные на эволюционной стратегии (Evolutionary Strategies, ES). Ochi и др. [403] (1998) разработали гибридную эволюционную эвристику, сочетающую в себе генетический алгоритм и процедуру Scatter Search (SS) [275] для решения задачи FSMF. Параллельный вариант, основанный на той же модели, представлен в [404]. Для задачи HVRP, впервые предложенной Taillard [443] (1999), автор разработал алгоритм, основанный на методах AMP (Adaptive Memory Procedure), поиска с запретами (Tabu

Search, TS) и генерации столбцов (Column Generation, CG), который применим и к задаче FSM. Несколько эвристик с запретами (TS) были предложены для решения задач FSMF и FSMD. Gendreau и др. [272] (1999) разработали TS-алгоритм, объединяющий GENIUS-подход (Generalized Insertion Procedure with Unstringing and Stringing processes) и AMP-процедуру. Han и Cho [301] (2001) для решения задачи FSMVRP рассмотрели новый метаэвристический подход (Generic Intensification and Diversification Search, GIDS), основанный на интеграции универсальных методов поиска, таких как Threshold Accepting (TA) и Great Deluge Algorithm (GDA) и мета-стратегии интенсификации и диверсификации для интеллектуального поиска. Метод включает в себя три компонента: множественный конструктор инициализации (Multiple Initialization Constructor, MIC); интенсификации поиска (Generic Search for Intensification, GSI); возмущения и диверсификации поиска (Perturbation Search for Diversification, PSD). Для проверки метода была протестирована группа из двадцати известных задач FSMVRP несколькими предложенными алгоритмами. Для двух тестовых примеров были обновлены рекорды, а для остальных — среднее отклонение от лучших решений составило 0,598%. Для задачи FSMF Renaud и Boctor [425] (2002) предложили улучшенный конструктивный эвристический алгоритм sweep-based heuristic, используемый для решения задач VRP. Wassan и Osman [474] (2002) для задачи FSMF разработали новые варианты метаэвристики поиска с запретами (TS), в которых используется концепция реактивного поиска (reactive tabu search), переменные окрестности (variable neighbourhoods), специальные структуры данных и хэширование памяти. Предложенный подход позволяет переключаться между стратегиями интенсификации и диверсификации и эффективно определять переменные окрестности поиска. Отмечается конкурентоспособность предложенных алгоритмов с другими известными алгоритмами. Tarantilis и др. [447] (2003) решили задачу HVRPD с помощью модификации метода моделируемого отжига (Simulated Annealing, SA), в котором худшее решение принимается только тогда, когда оно находится в пределах заданного порогового значения. Этими же авторами [448] (2004) также были

предложены другие пороговые процедуры для решения некоторых вариантов задачи. Меметический алгоритм (Memetic Algorithm, MA) для решения задачи FSMF был предложен Lima и др. [368] (2004). Для задач HVRPFD и HVRPD Li F. и др. [365] (2007) представили переборный record-to-record алгоритм, который использует детерминированный вариант SA. Lee и др. [363] (2008) для задачи FSM разработали алгоритм, основанный на методах TS и разбиения множества (Set Partitioning, SP). Tarantilis и др. [449] (2008) представлен новый алгоритм решения задачи HVRPFD методом ведомого поиска с запретами (Guided Tabu Search, GTS), основанного на поиске с запретами, контролируемого непрерывным направляющим механизмом для изменения целевой функции задачи. Роль этой направляющей стратегии заключается в расширении поиска и избегании локальных оптимумов. Гибридный генетический алгоритм (A hybrid Genetic Algorithm, GA), в котором применяется локальный поиск как метод мутации особей популяции был предложен Liu и др. [369] (2009) для решения задач FSMF и FSMD. Imran и др. [323] (2009) развили метод поиска с переменными окрестностями (Variable Neighborhood Search, VNS), который использует алгоритм Дейкстры и sweep-алгоритм для генерации начального допустимого решения и несколько различных структур окрестностей в фазе локального поиска. Авторы рассмотрели все варианты задачи FSM. В [192] Brandão (2009) для задачи FSM был предложен детерминированный алгоритм поиска с запретами (TS), который позволяет использовать различные процедуры для генерации начальных допустимых решений. Для решения всех вариантов задач FSM и HVRPD Prins [422] (2009) были разработаны две эффективные меметические эвристические процедуры. Для задачи HVRPFD Li X. и др. [366] (2010) предложили мульти-стартовую адаптивную процедуру многостраничного запуска в комбинации с перекомпоновкой путей и модифицированным методом поиска с запретами (TS). Позднее Brandão [193] (2011) был предложен алгоритм поиска с запретами (TS) для решения задачи HVRP с некоторыми дополнительными функциями.

Из последних отметим работу Mutingi и Mbohwa [399] (2012), посвящен-

ную разработке группы генетических алгоритмов (GA), с уникальными генетическими операторами. Как указывают авторы, алгоритмы на ряде известных тестов конкурентоспособны по точности решения и временным затратам с другими эвристиками и могут быть использованы для решения практических задач HVRP. Subramanian и др. [441] (2012) предложили гибридный эвристический итерационный алгоритм локального поиска (Iterated Local Search, ILS), основанный на методе разбиения множества (Set Partitioning, SP). Сформулированная SP задача решается в интерактивном режиме с помощью пакета программ смешанного целочисленного программирования. Разработанный алгоритм был протестирован на контрольных примерах, содержащих до 360 клиентов. В работе Karagül и Myo [332] (2014) предложен новый конструктивный алгоритм маршрутизации для задачи FSMF, в котором для выбора маршрута главную роль играют остаточные издержки, а не типы транспортных средств. Метаэвристический bio-inspired-алгоритм, основанный на концепции мембранных вычислений, предложен Juanjuan и Tao [330] (2014) для решения задач FSMVRP. В алгоритме клеточно-подобная структура и формальные правила обновления, связи, записи и отбора объединяются для улучшения глобального поиска. Решение ряда эталонных задач с учетом и без учета фиксированной стоимости показало пригодность предложенного алгоритма для высококачественного решения рассмотренных задач. Самые свежие обзоры по решению задач из семейства FCRP можно найти в работах [324, 345].

В работах [194, 346, 370, 413] представлен обширный библиографический обзор по решению задач с неоднородным парком транспортных средств, доставкой грузов и временными окнами (Time Window, TW). В этих задачах для каждого клиента заданы промежуток времени для обслуживания и время обслуживания, а для каждой дуги известно время ее проезда, которое, в большинстве случаев, принимается одинаковым для всех типов транспортных средств. Для задач с временными окнами принято рассматривать несколько вариантов в зависимости от целевой функции, которая может представлять собой: минимум суммы фиксированной стоимости всех используемых транспортных средств и

переменной стоимости транспортировки грузов и минимум суммарной временной продолжительности маршрутов без учета заданного времени обслуживания клиентов (время обслуживания используется только для проверки допустимости маршрута и корректировки времени выезда из депо для минимизации времени ожидания обслуживания); минимум суммы фиксированной и переменной стоимости и минимум суммарной длины маршрутов.

2.3. Математические модели перевозок мелкопартионных грузов во внутренней зоне магистрального узла

Пусть физическая сеть внутриузловых перевозок (см. рис. 2.1.а) задана связным ориентированным графом $\tilde{G}(\tilde{N},\tilde{P})$. Множество вершин $\tilde{N}=\{0\}\cup\{1,\ldots,n\}\cup\{n+1,\ldots,\tilde{n}\}$, перенумерованное от 0 до $\tilde{n}$ включает вершину $\{0\}$ — депо и магистральный узел, $\{1,\ldots,n\}$ — множество клиентов и $\{n+1,\ldots,\tilde{n}\}$ — множество транзитных пунктов, связывающих отдельные участки дорог. Транзитные пункты вводятся для того, чтобы учесть особенности отдельных участков дорог, соединяющих узлы четвертого типа во внутренней зоне магистрального узла. Участки дорог представлены противоположно направленными ориентированными дугами $(i,j),(j,i)$, $i,j\in\tilde{N}$, $i\neq j$, одна из которых может отсутствовать (например, из-за одностороннего движения по участку дороги или запрета проезда грузовым видам транспорта). Длины дуг заданы матрицей $\tilde{R}=\|\tilde{r}_{ij}\|_{\tilde{n}+1\times\tilde{n}+1}$, $\tilde{r}_{ij}\in R^1$, $(i,j)\in\tilde{P}$. Пусть для дуг сети известны коэффициенты $\tilde{k}_{ij}\in R^1$, $\tilde{k}_{ij}\geq 1.0$, характеризующие географические особенности (подъем, спуск и пр.) и физическое состояние (качество покрытия) участков дорог. Определим элементы преобразованной матрицы $\tilde{R}'$ как $\tilde{r}'_{ij}=\tilde{k}_{ij}\tilde{r}_{ij}$, $(i,j)\in\tilde{P}$ и построим на ней все кратчайшие пути в $\tilde{G}(\tilde{N},\tilde{P})$. В результате получим матрицу длин кратчайших путей $\tilde{L}=\|\tilde{l}_{ij}\|_{\tilde{n}+1\times\tilde{n}+1}$ и справочную матрицу $\tilde{C}=\|\tilde{c}_{ij}\|_{\tilde{n}+1\times\tilde{n}+1}$, каждый элемент которой $\tilde{c}_{ij}$, $i\neq j$ определяет номер предпоследнего узла на

кратчайшем пути от i до j, $\tilde{c}_{ii} = 0$, $i = \overline{1,\tilde{n}+1}$. С помощью справочной матрицы можно легко определить кратчайший путь между любыми вершинами исходного графа. По матрице $\tilde{L}$ для множества $\{0\} \cup \{1,\ldots,n\}$ построим полный ориентированный граф $G(N,A)$ с множеством вершин $N = \{0,1,\ldots,n\}$ и множеством дуг $A = \{(i,j) : \forall i,j \in N, i \neq j\}$ с известными длинами l_{ij}, $(i,j) \in A$. Очевидно, что для дуг построенного графа выполняется правило треугольника $l_{ik} + l_{kj} \geq l_{ij}$. В дальнейшем все задачи будем формулировать на графе $G(N,A)$.

Пусть для каждого клиента j, $j = \overline{1,n}$ задано среднесуточное количество $a_j > 0$ и $b_j > 0$, $a_j, b_j \in Z^+$ единиц мелкопартионных грузов унифицированного размера, которое нужно в течение суток доставить клиенту из магистрального узла (депо) и отправить от клиента в магистральный узел ($a_0 = b_0 = 0$). Принимается, что на протяжении некоторого периода текущего планирования длительностью T суток среднесуточные потоки изменяются незначительно и за T суток может быть перевезено $T\sum_{j=1}^{n}(a_j + b_j)$ грузов.

В депо имеется K типов транспортных средств с различной грузоподъемностью $Q_k \in Z^+$, $k = \overline{1,K}$. Грузоподъемность измеряется в тех же единицах, что и потоки мелкопартионных грузов. Предполагается, что $\max\limits_j \{a_j, b_j\} \leq Q_K$, а количество транспортных средств каждого типа может быть ограничено величиной m_k, $k = \overline{1,K}$ и не ограничено. Для каждого типа транспортного средства известна стоимость его приобретения и среднестатистическая стоимость обслуживания в сутки. В стоимость обслуживания включаются расходы на смазочные и прочие эксплуатационные материалы, ремонт и амортизацию подвижного состава, износ и ремонт автомобильной резины, заработная плата водителей вместе с отчислениями на социальные нужды. Пусть для расчета этих затрат задана функция $F_V = f(S_k, R_k)$, $k = \overline{1,K}$ где S_k — фиксированная стоимость приобретения транспортного средства типа k, а R_k — расходы на обслуживания одного транспортного средства типа k в сутки.

После сортировки в магистральном узле грузы для каждого клиента могут перевозиться в транспортных средствах на поддонах, для погрузки и выгрузки которых необходимо использовать механизированные автопогрузчики. Грузы от клиентов в магистральный узел перевозятся «навалом». При большом объеме перевозимых грузов затраты на операции погрузки и выгрузки могут быть значительными и их необходимо учитывать при решении задачи. Для депо и клиентов задана функция затрат на погрузку-выгрузку u_j грузов $F_L = f(Q_k, u_j)$, $j = \overline{0,n}$, $k = \overline{1,K}$. На практике принято считать, что эта функция не зависит от типа транспортного средства, а зависит только от объема груза и наличия погрузочно-разгрузочного оборудования в депо и у клиентов. В депо затраты на погрузку и выгрузку несет транспортное предприятие, клиенты могут выполнять эти операции за свой счет. В дальнейшем предполагается, что места дислокации клиентов являются доставочными и сборочными пунктами транспортного предприятия, поэтому затраты на погрузку и выгрузку грузов у клиентов также относятся к общим затратам предприятия. Как правило такие затраты моделируются непрерывной вогнутой функцией $F_L = f(u_j)$, $j = \overline{0,n}$ от объема грузов. В отдельных случаях может быть принята линейная зависимость $F_L = c_L u_j$, где c_L — стоимость погрузки-выгрузки единицы мелкопартионных грузов. Ясно, что если затраты на погрузку и выгрузку грузов не зависят от типа транспортного средства, то они не должны фигурировать в целевой функции задачи, и могут быть вычислены независимо от переменных следующим образом: $F_{full} = f(\sum_{i=1}^{n}(a_i + b_i)) + \sum_{i=1}^{n} f(a_i + b_i)$ или $F_{full} = 2c_L \sum_{i=1}^{n}(a_i + b_i)$. Если же такая зависимость существует, то эти затраты должны быть связаны с неизвестными переменными задачи.

Пусть стоимость проезда транспортного средства типа k по дуге $(i, j) \in A$ задана функцией $F_M = f(l_{ij}, Q_k, V_{av}^k, k_L)$, где l_{ij} — длина дуги в километрах, V_{av}^k — средняя скорость движения (км/час), k_L — коэффициент загрузки транспортного средства на дуге. Значение $k_L = u_{ij} / Q_k \in [0,1]$, где u_{ij} — текущая загрузка

транспортного средства. По существу функция F_M определяет стоимость израсходованного топлива на передвижение по дуге (i, j) для каждого транспортного средства типа k. Во многих задачах маршрутизации принимается $F_M = c_k l_{ij}$, где c_k — удельная стоимость топлива транспортного средства типа k на один километр пути. Часто рассматривается случай, когда затраты на передвижение по дуге не зависят от типа транспортного средства и одинаковы для прямого и обратного направления движения. В этом случае вводятся дополнительные условия для грузоподъемности $Q_1 < Q_2 ... < Q_K$ и фиксированной стоимости $F_1 < F_2 ... < F_K$.

Все функции затрат должны быть приведены к сопоставимому виду, например, за одни сутки или за заданный период времени текущего планирования T. Если построенные функции затрат F_V, F_L и F_M реально отражают производственные издержки, то их сумма будет достаточна близка к фактическим расходам транспортного предприятия на приобретение и эксплуатацию рабочего парка транспортных средств без учета накладных расходов. Примем, что при решении задачи эти функции рассчитываются и их соответствующие числовые значения F_k, f_j^k и f_{ij}^k, $i, j = \overline{0,n}$, $k = \overline{1,K}$ используются в целевой функции.

2.3.1. Формулировка задачи построения доставочных маршрутов (the Delivery Route, DR) для перевозки грузов из магистрального узла к клиентам. Однопродуктовая модель 1

Формулировка задачи основана на моделях, предложенных Gheysens и др. [274], Golden и др. [287], Baldacci и др. [167] и Salhi и Rand [430]. Задача заключается в определении множества непересекающихся по вершинам клиентов гамильтоновых циклических маршрутов, начинающихся и заканчивающихся в депо, с минимальными суммарными затратами. При этом требуется, чтобы все запросы клиентов были удовлетворены, а грузоподъемности транспортных средств на всех маршрутах не нарушались.

Обозначим $C=\{1,\ldots,n\}$ — множество клиентов, $V=\{1,\ldots,K\}$ — множество типов транспортных средств. Введем потоковые переменные y_{ij}, определяющие количество грузов в транспортном средстве при его проезде к клиенту j после посещения клиента i, $i,j\in N$ и булевы переменные x_{ij}^k, $x_{ij}^k=1$, если транспортное средство типа k движется от клиента i к клиенту j и $x_{ij}^k=0$ в противном случае. Пусть $\sum_{j=1}^{n} x_{0j}^k$ представляет собой суммарное количество используемых транспортных средств типа k.

Требуется найти минимум функции

$$F_{DR}=\sum_{k\in V}F_k\sum_{j\in C}x_{0j}^k+\sum_{k\in V}\sum_{j\in N}f_j^k\Big(\sum_{i\in C}\big|(y_{ij}-y_{ji})\big|x_{ji}^k\Big)+\sum_{k\in V}\sum_{i,j\in N}f_{ij}^k x_{ij}^k, \qquad (2.1)$$

при ограничениях

$$\sum_{k\in V}\sum_{i\in N}x_{ij}^k=1,\ \ \forall j\in C, \qquad (2.2)$$

$$\sum_{i\in N}x_{ij}^k-\sum_{i\in N}x_{ji}^k=0,\ \ \forall j\in C,\ \ \forall k\in V, \qquad (2.3)$$

$$\sum_{j\in C}y_{0j}=\sum_{j\in C}a_j,\ \ \sum_{j\in C}y_{j0}=0, \qquad (2.4)$$

$$\sum_{i\in N}y_{ij}-\sum_{i\in N}y_{ji}=a_j,\ \ \forall j\in C, \qquad (2.5)$$

$$y_{0j}\le\sum_{k\in V}Q_k x_{0j}^k,\ \ \forall j\in C, \qquad (2.6)$$

$$y_{ij}\le\sum_{k\in V}(Q_k-a_i)x_{ij}^k,\ \ \forall i\in C,\ \ \forall j\in N,\ \ i\ne j, \qquad (2.7)$$

$$x_{ij}^k\in\{0,1\},\ \ y_{ij}\ge 0 \text{ и целые},\ \ \forall i,j\in N,\ \ \forall k\in V. \qquad (2.8)$$

Ограничения (2.7) можно записать в расширенном виде

$$a_j x_{ij}^k\le y_{ij}\le(Q_k-a_i)x_{ij}^k,\ \ \forall i\in C,\ \ \forall j\in N,\ \ i\ne j,\ \ \forall k\in V. \qquad (2.9)$$

Если количество транспортных средств типа k ограничено величиной m_k, то к задаче могут быть добавлены условия

$$\sum_{j\in C}x_{0j}^k\le m_k,\ \ \forall k\in V. \qquad (2.10)$$

При необходимости могут быть установлены ограничения на максимальную продолжительность маршрутов по времени. Для этого включим в модель вре-

менные параметры T_k — максимальное время поездки транспортного средства типа k и t_{ij} — время проезда по дуге (i, j), а также непрерывные переменные r_{ij}, определяющие разность между значением T_k и текущим значением времени движения после проезда по дуге (i, j). Тогда к задаче добавятся ограничения

$$r_{ij} \le \sum_{k \in V} T_k x_{ij}^k, \quad \forall (i, j) \in A, \tag{2.11}$$

$$r_{0j} = \sum_{k \in V} T_k x_{0j}^k - \sum_{k \in V} t_{0j} x_{0j}^k, \quad \forall j \in C, \tag{2.12}$$

$$\sum_{i \in N} r_{ip} - \sum_{j \in N} r_{pj} = \sum_{k \in V} \sum_{j \in N} t_{pj} x_{pj}^k, \quad \forall p \in C, \tag{2.13}$$

$$r_{ij} \ge 0, \quad \forall (i, j) \in A. \tag{2.14}$$

Первая часть целевой функции (2.1) определяет фиксированные затраты на приобретение и обслуживание используемых транспортных средств, а вторая и третья — затраты на погрузку-выгрузку и доставку всех грузов. Ограничения (2.2) и (2.3) гарантируют, что каждый клиент j посещается только один раз каким-либо транспортным средством типа k, и это транспортное средство после прибытия к клиенту и выгрузки его груза a_j должно обязательно покинуть клиента. Ограничения (2.4) означают, что общее количество грузов на выходе из депо равно суммарным требованиям всех клиентов, и что никакие грузы не возвращаются в депо. Ограничения (2.5) означают, что при посещении клиента j груз a_j должен быть выгружен, т.е. количество груза в транспортном средстве после посещения клиента j уменьшается на величину a_j. Эти ограничения обеспечивают выполнение требований всех клиентов и исключают циклы, не проходящие через депо. Ограничения (2.6) определяют, что количество грузов y_{0j}, транспортируемых до клиента j не должно превышать грузоподъемности транспортного средства, назначенного для доставки груза этому клиенту. Ограничения (2.7) или (2.9) связывают переменные y_{ij} и x_{ij}^k, и означают, что никакие грузы не перевозятся из i в j, если ни одно транспортное средство не обслуживает связь между этими узлами, т.е. если $x_{ij}^k = 0 \quad \forall k \in V$. Ограничения

(2.8) устанавливают область изменения переменных. Ограничения (2.11) означают, что оставшееся время движения транспортного средства после проезда по дуге (i, j) не может превышать максимального времени движения транспортного средства. Условия (2.12) обеспечивают то, чтобы оставшееся время движения транспортного средства после выезда из депо было равно разности между максимальным временем движения и временем, необходимым для перемещения к клиенту j. Ограничения (2.13) указывают на тот факт, что каждый раз, когда транспортное средство проезжает между двумя клиентами, оставшееся время движения уменьшается на время проезда между этими клиентами.

Для формулировки задачи (2.1)-(2.8) с ограничениями на временные окна, депо представим двумя узлами с номерами 0 и $n+1$. Тогда $N = \{0\} \cup \{1,...,n\} \cup \{n+1\}$. Пусть $[t_i^S, t_i^C]$ и s_i соответственно определяют промежуток времени (временное окно), на протяжении которого возможно обслуживание, и время обслуживания клиента i, $t_i^S, t_i^C, s_i \in R^1$. Введем переменные $t_i^k \in R^1$, означающие точное время начала обслуживания транспортного средства типа k у клиента i. Принимается, что если транспортное средство прибывает к клиенту i ранее t_i^S, то оно ожидает начала обслуживания. В задачу (2.1)-(2.8) добавятся (см. [194]) — в целевую функцию слагаемое $\sum_{k \in V} (t_{n+1}^k - t_0^k)$, и ограничения:

$$\sum_{j \in N} x_{0j}^k = 1, \ \sum_{i \in N} x_{i,n+1}^k = 1, \ \forall k \in V;$$

$$t_j^k \geq t_i^k + s_i + t_{ij} - M_{ij}(1 - x_{ij}^k), \ \forall i, j \in N, \ \forall k \in V,$$

$$\text{где } M_{ij} = \max\{0, t_i^C + s_i + t_{ij} - t_j^S\};$$

$$t_i^S \sum_{j \in N} x_{ij}^k \leq t_i^k, \ t_i^k \leq t_i^C \sum_{j \in N} x_{ij}^k, \ \forall i \in N, \ \forall k \in V;$$

$$x_{i0}^k = 0, \ x_{ii}^k = 0, \ x_{n+1,i}^k = 0, \ \forall i \in N, \ \forall k \in V.$$

В записи первых ограничений предполагается, что если транспортное средство не используется, то оно совершает пустой тур, т.е. $x_{0,n+1}^k = 1, \ \forall k \in V$. Кроме того, $a_0 = a_{n+1} = s_0 = s_{n+1} = 0$, $[t_0^S, t_0^C] = [t_{n+1}^S, t_{n+1}^C] = [Start, Close]$, где *Start* и *Close* —

время выезда и приезда в депо. Задача с временными окнами является задачей смешанного целочисленного программирования.

2.3.2. Формулировка задачи построения доставочных маршрутов для перевозки грузов из магистрального узла к клиентам. Однопродуктовая модель 2

Введем вместо переменных y_{ij} булевы переменные $y_{ik}=1$, если транспортное средство типа k посещает клиента или депо i, и $y_{ik}=0$ в противном случае. Для исключения подциклов при построении маршрутов введем целочисленные переменные $u_{ik}\geq 0$, значения которых определяют порядковый номер клиента i в маршруте транспортного средства типа k. Впервые вариант исключения подциклов был предложен в 1960 г. в работе [386]. Сформулируем задачу. Требуется найти минимум функции

$$F_{DR}=\sum_{k\in V}F_k\sum_{j\in C}x_{0j}^k+\sum_{k\in V}f_0^k\Big(\sum_{i\in C}a_i y_{0k}\Big)+\sum_{k\in V}\sum_{i\in C}f_i^k(a_i y_{ik})+\sum_{k\in V}\sum_{i,j\in N}f_{ij}^k x_{ij}^k, \quad (2.15)$$

при ограничениях

$$\sum_{k\in V}y_{ik}=1,\ \ \forall i\in C, \quad (2.16)$$

$$\sum_{j\in N}x_{ij}^k=y_{ik},\ \ \sum_{j\in N}x_{ji}^k=y_{ik},\ \ \forall i\in N,\ \ i\neq j,\ \ \forall k\in V, \quad (2.17)$$

$$\sum_{i\in C}a_i y_{ik}\leq Q_k,\ \ \forall k\in V, \quad (2.18)$$

$$u_{ik}-u_{jk}+nx_{ij}^k\leq n-1,\ \ \forall i,j\in C,\ \ \forall k\in V, \quad (2.19)$$

$$x_{ij}^k, y_{ik}\in\{0,1\},\ \ \forall i,j\in N,\ \ \forall k\in V, \quad (2.20)$$

$$u_{ij}\geq 0 \text{ и целые},\ \ \forall i,j\in C. \quad (2.21)$$

Отметим, что условия запрета подциклов можно записать в виде

$$\sum_{i\in S}\sum_{j\in S\setminus\{i\}}x_{ij}^k\leq |S|-1,\ \ \forall S\subseteq C,\ \ |S|\geq 2,\ \ \forall k\in V, \quad (2.22)$$

где S — кластер (подмножество клиентов), обслуживаемый одним транспортным средством типа k.

Ограничения (2.16) и (2.17) гарантируют, что каждый клиент посещается только одним транспортным средством, и что транспортное средство после по-

сещения клиента или депо должно его покинуть. Условия (2.18) и (2.19) или (2.22) устанавливают ограничения на загрузку транспортных средств и запрещают подциклы, не проходящие через депо. В задачу также можно ввести дополнительные условия, ограничивающие количество транспортных средств каждого типа и временные ограничения на маршруты.

2.3.3. Формулировка задачи построения доставочных маршрутов для перевозки грузов из магистрального узла к клиентам. Двухпродуктовая модель

Рассмотрим двухпродуктовую формулировку задачи, предложенную в [169], для случая, когда $f_{ij}^{k_1} = f_{ji}^{k_2} = f_{ij} = f_{ji} \quad \forall i, j \in N$, $k_1 \neq k_2 \in V$, $f_i^k = f_i$ $\forall i \in N, \forall k \in V$. В модель не включаются затраты на погрузку-выгрузку, поскольку предполагается, что они не зависят от типа транспортного средства и могут быть вычислены по формуле для F_{full}.

Пусть $G(N', E)$ — полный неориентированный граф, полученный из графа $G(N, A)$ следующим образом. Множество узлов N' включает множество узлов клиентов, K копий узла депо, ассоциированных с K типами транспортных средств и один общий узел депо. Пусть узлы N' занумерованы и $N' = \{1,\ldots,n\} \cup \{n+1,\ldots,n+K\} \cup \{n+K+1\}$. Обозначим, как и ранее, $C = \{1,\ldots,n\}$, $V = \{n+1,\ldots,n+K\}$, $n' = n+K+1$. Пусть $\pi(i)$, $i \in V$ означает транспортное средство, связанное с узлом i. Определим стоимости дуг $c_{ij} \in E$ следующим образом: $c_{ij} = F_{\pi(j)} + f_{0i}$, для $a_i \leq Q_{\pi(j)}$, $j \in V$, $i \in C$; $c_{ij} = f_{ij}$, для $a_i + a_j \leq Q_K$, $i, j \in C$, $i < j$; $c_{in'} = f_{0i}$, для $i \in C$; $c_{ij} = +\infty$ в противном случае. Предполагается, что $a_i = 0$, $\forall i \in V \cup \{n'\}$.

В модели используется две потоковые переменные y_{ij} и y_{ji}, связанные с каждой дугой $(i, j) \in E$. Переменная y_{ij} определяет загрузку транспортного средства, а переменная $y_{ji} = Q_K - y_{ij}$ представляет свободную емкость транспортного средства с наибольшей грузоподъемностью. Значение $y_{ji} - (Q_K - Q_k)$ означает свободную емкость транспортного средства типа k. Пусть для каждой

дуги $(i, j) \in E$ определены булевы переменные $x_{ij} = 1$, если дуга (i, j) включена в решение и $x_{ij} = 0$ в противном случае.

Определим $\Omega = \{S : S \subseteq C, |S| \geq 2\}$ и пусть для подмножества (кластера) $S \in \Omega$, $\overline{S} = C \setminus S$, а $\delta(S) = \{(i, j) \in E : i \in S \wedge j \notin S \vee i \notin S \wedge j \in S\}$. Обозначим $a(S) = \sum_{i \in S} a_i$ — суммарный поток требований клиентов в кластере S.

Сформулируем задачу. Требуется найти минимум функции

$$\sum_{(i,j) \in E} c_{ij} x_{ij} \tag{2.23}$$

при ограничениях

$$\sum_{j \in N'} (y_{ji} - y_{ij}) = 2a_i, \ \forall i \in C, \tag{2.24}$$

$$\sum_{i \in V} \sum_{j \in C} y_{ij} = a(C), \quad \sum_{j \in C} y_{jn'} = 0, \tag{2.25}$$

$$\sum_{\{i,j\} \in \delta(p)} x_{ij} = 2, \quad \forall p \in C, \tag{2.26}$$

$$\sum_{i \in V} \sum_{j \in C} x_{ij} = \sum_{j \in C} x_{jn'}, \tag{2.27}$$

$$y_{ij} + y_{ji} = Q_K x_{ij}, \ \forall (i, j) \in E, \tag{2.28}$$

$$\sum_{\{i,j\} \in \delta(S)} x_{ij} \geq \sum_{i \in V} \sum_{j \in C} x_{ij}, \ S \subseteq V \cup C, \ V \subset S, \tag{2.29}$$

$$y_{ij} \leq Q_{\pi(i)}, \ \forall i \in V, \ \forall j \in C, \tag{2.30}$$

$$y_{ij}, y_{ji} \geq 0 \text{ и целые}, \ x_{ij} \in \{0,1\} \ \forall (i, j) \in E. \tag{2.31}$$

Ограничения (2.24), (2.25) и (2.31) определяют допустимый поток. Ограничения (2.26) гарантируют, что в любом допустимом решении у каждого клиента будут только две инцидентных ему дуги. Условия (2.27) означают, что если $q = \sum_{i \in V} \sum_{j \in C} x_{ij}$ транспортных средств выехало из множества узлов V, то все они обязательно должны приехать в общее депо. Ограничения (2.28) устанавливают связь между переменными в допустимом решении задачи, а условия (2.29) запрещают образование простых путей, начинающихся и заканчивающихся в узлах множества V. Ограничения на грузоподъемность для различных типов транспортных средств заданы условиями (2.30).

В [169] доказана теорема о том, что множество допустимых решений задачи (2.24)-(2.31) однозначно соответствует множеству решений задачи FSMF (задачи с независимыми ценами дуг от типа транспортного средства).

2.3.4. Формулировка задачи построения доставочных маршрутов для перевозки грузов из магистрального узла к клиентам. Модели на разбиении множества

Рассмотрим постановки задач HVRP, предложенные в [170] и [430], основанные на ранней работе M. Balinski и R. Quandt [173]. Определим маршрут транспортного средства как пару (R,k), где $R=(i_1,i_2,...,i_{|R|})$, $i_1=i_{|R|}=0$, $\{i_2,i_3,...,i_{|R|-1}\}\subseteq C$, $\left|\{i_2,i_3,...,i_{|R|-1}\}\right|\geq 1$ представляет простой цикл, начинающийся и заканчивающийся в депо, а k — определяет тип транспортного средства, связанного с R. Ссылка на маршрут R используется для обозначения как последовательности посещаемых клиентов, так и для подмножества этих клиентов, включая депо. Маршрут (R,k) является *допустимым*, если суммарный запрос клиентов, посещаемых на этом маршруте, не превышает грузоподъемности транспортного средства Q_k, выделенного для этого маршрута, $\sum_{h=2}^{|R|-1} a_{i_h} \leq Q_k$. Затраты на маршруте (R,k) включают F_k (суммарные затраты на приобретение и обслуживание транспортного средства типа k) и сумму затрат на дугах маршрута $\sum_{h=1}^{|R|-1} f^{k}_{i_h i_{h+1}}$.

Пусть R_k представляет множество всех возможных допустимых маршрутов для транспортного средства типа $k\in V$. Для каждого маршрута $l\in R_k$ обозначим f_{lk} — ассоциируемые маршрутные затраты (сумму затрат на дугах маршрута), f^{k}_{l} — затраты на выгрузку грузов клиентам на маршруте. Пусть $R_{ik}\subset R_k$ — подмножество маршрутов транспортного средства типа k, покрывающих (охватывающих) клиентов $i\in C$, а r_{lk} определяет подмножество клиентов, посеща-

емых на маршруте l, $r_{lk} = \{i_1, i_2, ..., i_h\}$, $\{i_1, i_2, ..., i_h\} \subseteq C$. Определим $a_{lk} = \sum_{i \in r_{lk}} a_i$, $\forall l \in R_k$, $\forall k \in V$. Введем булевы переменные $x_{lk} = 1$, если и только если маршрут $l \in R_k$ включен в оптимальное решение и $x_{lk} = 0$ в противном случае.

Требуется найти минимум

$$\sum_{k \in V} \sum_{l \in R_k} (F_k + f_{lk}) x_{lk} + 2 \sum_{k \in V} \sum_{l \in R_k} f_l^k (a_{lk} x_{lk}) \tag{2.32}$$

при ограничениях

$$\sum_{k \in V} \sum_{l \in R_{ik}} x_{lk} = 1, \ \forall i \in C, \tag{2.33}$$

$$\sum_{l \in R_k} x_{lk} \le m_k, \ \forall k \in V, \tag{2.34}$$

$$x_{lk} \in \{0,1\}, \ \forall l \in R_k, \ \forall k \in V. \tag{2.35}$$

Вторая составляющая целевой функции умножается на два, так как все выгружаемые грузы из транспортных средств типа k на маршруте l, предназначенные клиентам i, загружаются в эти же транспортные средства в депо, а затраты на погрузку и выгрузку единицы груза предполагаются одинаковыми для транспортных средств одинакового типа. Ограничения (2.33) гарантируют, что каждый клиент может быть обслужен только одним транспортным средством, а условия (2.34) ограничивают количество используемых транспортных средств каждого типа (эти условия могут отсутствовать при неограниченном рабочем парке).

Приведем несколько измененную формулировку [430], в которой учитываются время обслуживания клиента и ограничения на общее время движения транспортного средства. Определим множество $\tilde{V} = \{1,2,...,K\}$ всех типов транспортных средств (количество транспортных средств каждого типа не ограничено), занумерованных индексом $k = \overline{1,K}$. Пусть F_k, Q_k и T_k соответственно означают затраты на приобретение и обслуживание, грузоподъемность и максимальное время движения транспортного средства типа $k \in \tilde{V}$. Введем переменную решения задачи v, указывающую общее количество транспортных средств, включенных в состав рабочего парка $V = \{1,2,...,v\}$, $V \subseteq \tilde{V}$. Введем пе-

ременные затраты α_k и временной фактор β_k в расчете на единицу расстояния для каждого типа транспортного средства $k \in \tilde{V}$. Пусть t_i — время обслуживания клиента $i \in C$, R_p — множество клиентов, обслуживаемых транспортным средством с номером $p \in V$, а σ — функция $\sigma : V \to \tilde{V}$, однозначно отображающая множество номеров из V на множество типов транспортных средств $\tilde{V}$ и $\sigma(p)$ указывает на тип транспортного средства с наименьшей грузоподъемностью $Q_{\sigma(p)}$, которое может обслужить клиентов из R_p. Пусть π_p является кратчайшим маршрутом коммивояжера (TSP-маршрутом) с отправной и конечной точкой в депо, который обслуживает клиентов из R_p, а $\pi_p(i)$ указывает положение (позицию) клиента i в π_p. Введем для маршрута π_p обозначения: $D(\pi_p)$ — общее расстояние, $T(\pi_p)$ — общее время в пути, $C(\pi_p)$ — сумма переменной и фиксированной стоимости. Обозначим через l_{ij} расстояние между клиентами i и j. Пусть S является допустимым решением и определяется как $S = \{R_1, ..., R_v\}$, а Π представляет множество всех маршрутов коммивояжера в S, $\Pi = \{\pi_1, ..., \pi_v\}$. Сформулируем следующую задачу оптимизации:

$$\min_{S,\Pi,v} C(S) = \sum_{p \in V} C(\pi_p), \tag{2.36}$$

при ограничениях

$$\bigcup_{p \in V} R_p = C,\ R_p \cap R_q = \varnothing,\ \forall p \neq q \in V, \tag{2.37}$$

$$\sum_{i \in R_p} a_i \leq Q_{\sigma(p)},\quad \forall p \in V, \tag{2.38}$$

$$D(\pi_p) = \sum_{i \in R_p \cup \{0\}} l_{i\pi_p(i)},\quad \forall p \in V, \tag{2.39}$$

$$T(\pi_p) = \beta_{\sigma(p)} D(\pi_p) + \sum_{i \in R_p} t_i \leq T_{\sigma(p)},\quad \forall p \in V, \tag{2.40}$$

$$C(\pi_p) = F_{\sigma(p)} + f_0^{\sigma(p)}\left(\sum_{i \in R_p} a_i\right) + \sum_{i \in R_p} f_i^{\sigma(p)}(a_i) + \alpha_{\sigma(p)} D(\pi_p),\ \forall p \in V. \tag{2.41}$$

Вместо слагаемого $\alpha_{\sigma(p)} D(\pi_p)$ можно использовать $\sum_{i \in R_p \cup \{0\}} f^{\sigma(p)}_{i\pi_p(i)\,i\pi_p(i)}$, а вместо $\beta_{\sigma(p)} D(\pi_p)$ — $D(\pi_p)/\theta_{\sigma(p)}$, где $\theta_{\sigma(p)}$ — средняя скорость движения транспорт-

ного средства типа $\sigma(p)$. Целевая функция (2.36) определяет общую сумму затрат на всех маршрутах. Ограничения (2.37) и (2.38) означают, что каждый клиент обслуживается только одним маршрутом транспортного средства и что суммарный объем требований клиентов, обслуживаемых каждым маршрутом не может превышать грузоподъемности выделенного транспортного средства. Уравнения (2.39) и (2.41) определяют соответственно общую длину и затраты для каждого маршрута. Ограничения (2.40) означают, что время проезда по каждому маршруту не должно превышать заданной максимальной величины.

Как уже отмечалось, в настоящее время известны три точных алгоритма для решения задач с неоднородным парком транспортных средств. Choi и Tcha (2007) [210] для линейной релаксации исходной задачи разработали алгоритм на основе метода генерации столбцов. Они модифицировали несколько алгоритмов динамического программирования для решения классической задачи VRP с целью эффективной генерации допустимых столбцов, а затем применили процедуру ветвей и границ для получения целочисленного решения. Результаты численного экспериментирования с разработанными алгоритмами показали их превосходство по сравнению с существующими алгоритмами по качеству получаемых решений и по времени вычислений. Baldacci и Mingozzi [170] (2009) предложили универсальный точный алгоритм, использующий метод разбиения множества (set partitioning) для решения всех задач из семейства FCRPF. Ими были использованы три типа процедур основанные на LP и лагранжевой релаксации исходной задачи и получены новые более точные нижние границы решения задач. Третий точный алгоритм предложен Baldacci, Bartolini, Mingozzi и Roberti [171] (2010). Он сочетает в себе несколько итеративных двойственных процедур для генерации близко оптимальных двойственных нецелочисленных решений для множества разбиений (кластерных решений) и добавляет в координирующую задачу эффективные отсекающие неравенства для сокращения перебора в алгоритме генерации столбцов и получения целочисленного решения. Для окончательного двойственного решения определяется допустимая область, содержащая только целочисленные решения. Как утверж-

дается в [171] этот алгоритм эффективнее всех других известных точных алгоритмов.

2.3.5. Формулировка задачи построения сборочных маршрутов (the Collection Route, CR) для перевозки грузов от клиентов в магистральный узел

Сформулируем задачу аналогично (2.1)-(2.8). Пусть для каждого клиента j, $j=\overline{1,n}$ задано среднесуточное количество $b_j>0$, $b_j\in Z^+$ единиц мелкопартионных грузов унифицированного размера, которое нужно в течение суток отправить от клиента в магистральный узел (депо). Введем потоковые переменные z_{ji}, определяющие количество грузов в транспортном средстве при его проезде к клиенту i после посещения клиента j, $i,j\in N$ и булевы переменные x_{ij}^k, $x_{ij}^k=1$, если транспортное средство типа k движется от клиента i к клиенту j и $x_{ij}^k=0$ в противном случае. Пусть $\sum_{j=1}^{n} x_{0j}^k$ представляет собой суммарное количество используемых транспортных средств типа k.

Требуется найти минимум функции

$$F_{CR}=\sum_{k\in V}F_k\sum_{j\in C}x_{0j}^k+\sum_{k\in V}\sum_{j\in N}f_j^k\Big(\sum_{i\in C}\big|(z_{ji}-z_{ij})\big|x_{ij}^k\Big)+\sum_{k\in V}\sum_{i,j\in N}f_{ij}^k x_{ij}^k, \tag{2.42}$$

при ограничениях

$$\sum_{k\in V}\sum_{i\in N}x_{ij}^k=1,\ \ \forall j\in C, \tag{2.43}$$

$$\sum_{i\in N}x_{ij}^k-\sum_{i\in N}x_{ji}^k=0,\ \ \forall j\in C,\ \ \forall k\in V, \tag{2.44}$$

$$\sum_{j\in C}z_{0j}=0,\ \ \sum_{j\in C}z_{j0}=\sum_{j\in C}b_j, \tag{2.45}$$

$$\sum_{i\in N}z_{ji}-\sum_{i\in N}z_{ij}=b_j,\ \ \forall j\in C, \tag{2.46}$$

$$z_{j0}\le\sum_{k\in V}Q_k x_{j0}^k,\ \ \forall j\in C, \tag{2.47}$$

$$z_{ij}\le\sum_{k\in V}(Q_k-b_j)x_{ij}^k,\ \ \forall i\in C,\ \ \forall j\in N,\ \ i\ne j, \tag{2.48}$$

$$x_{ij}^k\in\{0,1\},\ \ z_{ij}\ge 0 \text{ и целые},\ \ \forall i,j\in N,\ \ \forall k\in V. \tag{2.49}$$

Ограничения (2.48) можно записать в виде

$$b_i x_{ij}^k\le z_{ij}\le(Q_k-b_j)x_{ij}^k,\ \ \forall i\in C,\ \ \forall j\in N,\ \ i\ne j,\ \ \forall k\in V.$$

Так же как и в задаче (2.1)-(2.8) в постановку задачи (2.42)-(2.49) могут быть введены ограничения на используемое количество транспортных средств каждого типа и на максимальную продолжительность маршрутов по времени. Первая часть целевой функции (2.42) определяет фиксированные затраты на приобретение и обслуживание используемых транспортных средств, вторая — затраты на погрузку грузов у клиентов и их выгрузку в депо, и третья — переменные затраты на доставку всех грузов в депо. Ограничения (2.43) и (2.44) гарантируют, что каждый клиент j посещается только один раз каким-либо транспортным средством типа k, и это транспортное средство после прибытия к клиенту и погрузки его груза b_j должно обязательно покинуть клиента. Ограничения (2.45) означают, что общее количество грузов на выходе из депо равно нулю, а на входе в депо равно суммарным требованиям всех клиентов. Ограничения (2.46) означают, что при посещении клиента j груз b_j должен быть погружен, т.е. количество груза в транспортном средстве после посещения клиента j увеличивается на величину b_j. Эти ограничения обеспечивают выполнение требований всех клиентов и исключают циклы, не проходящие через депо. Ограничения (2.47) гарантируют, что грузоподъемность транспортных средств не будет превышена. Ограничения (2.48) связывают переменные z_{ij} и x_{ij}^k, и означают, что никакие грузы не перевозятся из i в j, если ни одно транспортное средство не обслуживает связь между этими узлами, т.е. если $x_{ij}^k = 0$ $\forall k \in V$. Ограничения (2.49) устанавливают область изменения переменных.

Отметим, что для сборочных маршрутов могут быть сформулированы и математические модели, аналогичные приведенным в подразделах 2.3.2-2.3.4.

2.3.6. Формулировка задачи построения комбинированных (комплексных доставочно-сборочных) маршрутов (the Delivery and Collection Route, DCR) и обзор методов ее решения

Обозначим общую сумму затрат транспортного предприятия как $F_{DR} + F_{CR}$ при проектировании только доставочных и сборочных маршрутов и рассмот-

рим комплексную задачу построения комбинированных маршрутов. В зарубежной литературе такая задача называется The Heterogeneous Vehicle Routing Problem with Simultaneous Pickup and Delivery (HVRPSPD). Главное отличие этой задачи от рассмотренных ранее заключается в том, что при построении маршрутов важен порядок посещения клиентов с учетом грузоподъемности транспортного средства и возможности погрузки грузов в магистральный узел в зависимости от количества выгруженного груза в каждом узле маршрута. В этом случае для отдельных индивидуальных задач оптимизации может оказаться, что общие затраты F_{DCR} при решении комплексной задачи меньше или равны суммарным затратам $F_{DR} + F_{CR}$. Не исключен и вариант $F_{DR} + F_{CR} < F_{DCR}$. Поэтому при проектировании маршрутов транспортное предприятие должно использовать доставочные, сборочные и комплексные модели.

Сформулируем задачу. Требуется найти минимум функции

$$F_{DCR} = \sum_{k\in V} F_k \sum_{j\in C} x_{0j}^k + \sum_{k\in V}\sum_{j\in N} f_j^k \Big(\sum_{i\in C} \big|(y_{ij} - y_{ji})\big| x_{ji}^k\Big) + \sum_{k\in V}\sum_{j\in N} f_j^k \Big(\sum_{i\in C} \big|(z_{ji} - z_{ij})\big| x_{ij}^k\Big) +$$

$$+ \sum_{k\in V}\sum_{i,j\in N} f_{ij}^k x_{ij}^k, \qquad (2.50)$$

при ограничениях

$$\sum_{k\in V}\sum_{i\in N} x_{ij}^k = 1, \ \ \forall j \in C, \qquad (2.51)$$

$$\sum_{i\in N} x_{ij}^k - \sum_{i\in N} x_{ji}^k = 0, \ \ \forall j \in C, \ \ \forall k \in V, \qquad (2.52)$$

$$\sum_{j\in C} y_{0j} = \sum_{j\in C} a_j, \ \ \sum_{j\in C} z_{j0} = \sum_{j\in C} b_j, \qquad (2.53)$$

$$\sum_{i\in N} y_{ij} - \sum_{i\in N} y_{ji} = a_j, \ \ \forall j \in C, \qquad (2.54)$$

$$\sum_{i\in N} z_{ji} - \sum_{i\in N} z_{ij} = b_j, \ \ \forall j \in C, \qquad (2.55)$$

$$\sum_{k\in V} a_j x_{ij}^k \le y_{ij} \le \sum_{k\in V} (Q_k - a_i) x_{ij}^k, \ \ \forall i, j \in N, \ \ i \ne j, \qquad (2.56)$$

$$\sum_{k\in V} b_i x_{ij}^k \le z_{ij} \le \sum_{k\in V} (Q_k - b_j) x_{ij}^k, \ \ \forall i, j \in N, \ \ i \ne j, \qquad (2.57)$$

$$y_{ij} + z_{ij} \le \sum_{k\in V} (Q_k - \max\{0, b_j - a_j, a_i - b_i\}) x_{ij}^k, \ \ \forall i, j \in N, \ \ i \ne j, \ \ \forall k \in V, \qquad (2.58)$$

$$x_{ij}^k \in \{0,1\}, \ \ y_{ij}, z_{ij} \ge 0 \text{ и целые}, \ \ \forall i, j \in N, \ \ \forall k \in V. \qquad (2.59)$$

Так же как и в задаче (2.1)-(2.8) в постановку задачи (2.50)-(2.58) могут быть добавлены ограничения на используемое количество транспортных средств каждого типа, максимальную продолжительность маршрутов по времени и временные окна. Первая часть целевой функции (2.50) определяет фиксированные затраты на приобретение и обслуживание используемых транспортных средств, вторая — затраты на погрузку и выгрузку грузов, и третья — переменные затраты на транспортировку грузов. Ограничения задачи имеют тот же смысл, что и соответствующие ограничения задач (2.1)-(2.8) и (2.42)-(2.49).

Для задачи (2.50)-(2.59) могут быть сформулированы двухпродуктовые постановки, аналогично задаче VRPSPD (Vehicle Routing Problem with Simultaneous Pickup and Delivery) из работы [442].

Задача проектирования комбинированных маршрутов может рассматриваться как один из вариантов задачи пикап-доставки (Pickup and Delivery Problem, PDP). Общая схема классификации таких задач предложена Berbeglia и др. [181]. В соответствии с этой схемой задача VRPSPD отнесена к классу «один - ко многим - к одному» (one-to-many-to-one) с комбинированными требованиями клиентов. Задача VRPSPD впервые была сформулирована Min [389] в 1989 г. Для решения задачи распределения и сбора книг публичной библиотеки автор предложил эвристические алгоритмы, основанные на кластеризации клиентов в соответствии с их потребностями и грузоподъемностью транспортных средств и решении задачи коммивояжера (TSP) для каждого кластера.

Обзор литературы по задаче VRPSPD до 2012 г. можно найти в диссертационной работе A. Subramanian [442].

В настоящее время в литературе описано немного точных аппроксимаций для решения задачи VRPSPD. Так, например Dell'Amico и др. [227] (2006), разработан алгоритм ветвей и цен (branch-and price), в котором использованы две различные стратегии для решения подзадач ценообразования — точные методы динамического программирования и методы релаксации. Авторам удалось найти оптимальные решения для задач с 40 клиентами при использовании только идентичных транспортных средств. Ранее Angelelli и Manisini [150] (2002) так-

же применили метод ветвей и цен к формулировке задачи VRPSPD в виде разбиения множества с дополнительными ограничениями на временные окна (time-windows constraints). В их работе подзадачи сформулированы как задачи нахождения кратчайших путей с ресурсными ограничениями и использован алгоритм расстановки меток. Были найдены точные решения для задач с 20 клиентами. Трех индексные формулировки VRPSPD предложены Dethlof [236] (2001) и Montané и Galvão [392] (2006), однако только последние авторы провели экспериментальные исследования. Для решения задачи в их постановке был использован пакет CPLEX 9.0, с помощью которого за два часа счета удалось получить нижние границы для контрольных экземпляров индивидуальных задач с 50-400 клиентами. Subramanian [437] (2008) представил двухпродуктовую потоковую формулировку задачи VRPSPD, но никаких практических экспериментов не проводил.

Эвристические методы решения задачи VRPSPD были предложены в диссертации Halse [300] (1992), в которой автор разработал двухэтапный алгоритм, основанный на концепции cluster-first, route-second. Позднее Salhi и Nagy [431] (1999) предложили cluster insertion эвристики для задачи VRPB (VRP with Backhauls), которые в последующих работах использовались для решения задачи VRPSPD. Значительное число работ по задаче VRPSPD появилось после 2000 г. Так в [236] (2001) Dethloff была предложена эвристика, основанная на вставках самого дешевого допустимого маршрута по критерию остаточной грузоподъемности. Кроме того, автор исследовал связь задачи VRPSPD и другими вариантами задачи VRP. Røpke и Pisinger [428] (2006) разработали Large Neighborhood Search (LNS) эвристику, связанную с метаэвристической процедурой, аналогичной Variable Neighborhood Search (VNS) для решения различных вариантов задач VRPB и VRPSPD. Для задачи VRPPD (Vehicle Routing Problem with Pickups and Deliveries) Nagy и Salhi [400] (2005) разработали эвристический алгоритм, в котором используется общая методология решения задачи VRP, но модифицировали ее для сокращения множества допустимых решений.

Большинство эвристик, разработанных для решения задачи VRPSPD, ос-

нованы на методе поиска с запретами (TS). Crispim и Brandão [220] (2005) предложили метаэвристическую процедуру, объединяющую методы поиска с запретами (TS) и Variable Neighborhood Descent (VND). Montané и Galvão [392] (2006) разработали алгоритм TS с применением схемы локального поиска в окрестностях, характерной для решения традиционной задачи VRP и привели результаты экспериментальных расчетов для набора из 87 тестовых задач с числом клиентов от 50 до 400. Chen и Wu [204] (2006) предложили процедуру локального поиска на основе подхода record-to-record и списков запрета (tabu lists). Вычислительные эксперименты с гибридным эвристическим алгоритмом показали его высокую эффективность для решения задач небольшой размерности. Chen [205] (2006) представил эвристику, основанную на методах моделируемого отжига (Simulated Annealing, SA) и поиска с запретами (TS), в которой использована новая процедура параллельной вставки, позволяющая уменьшить число итераций алгоритма между получением начального допустимого решения и «оптимальным» решением. Bianchessi и Righini [186] (2007) предложили ряд конструктивных и локальных эвристик поиска, а также процедуру поиска с запретами (TS), которая использует переменную структуру окрестностей (variable neighborhood structure) с комбинированной схемой обмена узлов и дуг (node-exchange и arc-exchange) в проектируемых маршрутах. Wassan и др. [475] (2008) представили reactive-TS алгоритм поиска со следующими правилами построения окрестности: перемещение во вспомогательную структуру данных (Auxiliary Data Structure, ADS), обмен двух клиентов между двумя различными маршрутами и изменение направления движения по маршруту. Предложенный подход позволил сократить число возвратов для проведения повторного поиска. Zachariadis и др. [485] (2009) разработали алгоритм, который сочетает в себе принципы метаэвристик TS и ведомого локального поиска (Guided Local Search, GLS). Достоинство предложенного метода заключается в расширении пространства поиска оптимального решения и интенсификации поиска только в перспективных областях, избегая локальных оптимумов. Эффективность алгоритма была проверена на эталонных задачах с числом клиентов от 50 до 400. Те

же авторы [486] (2010) предложили алгоритм, основанный на использовании Adaptive Memory Procedure (AMP) и гранулированной TS эвристики. Рассмотренная стратегия использует инновационный механизм памяти и систематически максимизирует количество маршрутной информации, извлеченной из адаптивной памяти для ведения поиска в направлении перспективных областей пространства решений. Метаэвристический алгоритм был протестирован на эталонных задачах (число клиентов от 50 до 400).

Для решения задачи VRPSPD были также разработаны некоторые эволюционные стратегии (Evolutionary Strategies, ES). Vural [469] (2003) в диссертационной работе предложил два генетических алгоритма, первый из которых основан на случайном поиске (random key representation [177]), а второй состоит из улучшенной эвристики «Or-opt» Dethloff [236].

Gajpal и Abad [262] (2009) разработали муравьиный алгоритм (Ant Colony, AC), который состоит из двух основных этапов: интенсивность следа и параметры инициализируются с использованием исходного решения, полученного в ближайшей окрестности конструктивной эвристики; муравей - решение генерируется для каждого муравья с использованием интенсивности следа, за которым следует локальный поиск в каждом муравьином решении и обновлении элитарных муравьев и интенсивности следа. Авторами был проведен обширный численный эксперимент на эталонных задачах, который показал хорошие результаты. Çatay [199] (2010) предложил муравьиный алгоритм, основанный на новых saving-based функции видимости и процедуре обновления феромона.

Ai и Kachitvichyanukul [146] (2009) предложили алгоритм роя частиц (Particle Swarm Optimization, PSO) [338] с множественными структурами, который использует случайный поиск (random key representation). Способ декодирования в алгоритме начинается с преобразования роя частиц в приоритетный список клиентов и матрицу приоритетов транспортных средств, которые далее используются для построения маршрутов. Предложенный алгоритм протестирован и показал конкурентоспособность с другими известными алгоритмами решения задачи VRPSPD.

Mingyong и Erbao [390] (2010) предложен улучшенный алгоритм дифференциальной эволюции (improved differential evolution algorithm) для решения задачи VRPSPD с временными окнами (VRPSPDTW). В алгоритме было принято новое десятичное кодирование для построения начальной популяции и использовались некоторые улучшенные дифференциальные операторы эволюции в отличие от существующих алгоритмов. В операции мутации применялся критерий целого порядка на основе метода кодирования натурального числа. Для неосуществимых решений вводились штрафные коэффициенты, а для итераций алгоритма был разработан вероятностный самоадаптирующийся кроссовер.

Subramanian [437] (2008) и Subramanian и др. [438] (2008) предложили итерационный алгоритм локального поиска (Iterated Local Search, ILS), который использует метод спуска с переменными окрестностями (Variable Neighborhood Descent, VND) с детерминированным упорядочением окрестностей в локальной фазе поиска. Параллельный вариант этого алгоритма, в котором процедура VND имеет случайное упорядочение окрестностей (Variable Neighborhood Descent with Random neighborhood, RVND), был впоследствии разработан Subramanian и др. [440] (2010). Алгоритм состоит из процедуры спуска по случайно упорядоченным переменным окрестностям и встроенной итерационной процедуры локального поиска. Эксперименты с алгоритмом проводились на кластере с многоядерной архитектурой, использующей до 256 ядер. В результате испытаний были получены новые рекорды для некоторых известных тестовых задач. Subramanian и Cabral [439] (2008) также применили комбинацию ILS-VND для решения VRPSPD с ограничениями на временную продолжительность маршрутов. Единая структура VRP задачи, разработанная в этой работе послужила основой для дальнейшего развития предложенного метода. Souza и др. [436] (2010) также реализовали алгоритм ILS, но в сочетании с GENIUS [273] подходом. Предложенный ими алгоритм, протестированный на 72 эталонных задачах, для 9 задач показал новые улучшенные решения и для 49 задач ранее известные решения.

Новый алгоритм локального поиска на основе метаэвристики был предло-

жен Zachariadis и Kiranoudis [487] (2011). Алгоритм способен исследовать широкие окрестности решения за счет формирования специальных структур данных. Для того чтобы избежать зацикливания и индуцировать диверсификацию, общий поиск координируется использованием концепции, основанной на методологии поиска с запретами с заданной константой времени, ограничивающей проведение локального поиска.

Hezer и Kara Y [304] (2011) для решения задачи VRPSPD предложили бактериальный алгоритм (Bacterial Foraging Optimization Algorithm, BFOA), который представляет собой новый алгоритм оптимизации на основе поведения бактерии при поиске пищи. В основу алгоритма заложено поведение бактерии при поиске питательных веществ с наибольшей концентрацией, что может рассматриваться как процесс оптимизации при поиске решения.

Для решения задачи VRPSPD Tasan и Gen [454] (2012) предложили генетический алгоритм, а Wang и Chen [470] (2012) разработали генетический алгоритм для решения задачи VRPSPD с временными окнами, сформулированной в виде задачи смешанного целочисленного программирования. Работоспособность и вычислительная эффективность последнего алгоритма проверена на ряде тестовых задач Соломона (Solomon’s benchmark) с использованием программного обеспечения CPLEX.

В работе Jun и Kim [331] (2012) обсуждается эвристический алгоритм, состоящий из процедур начального построения и улучшения маршрутов, и процедуры возмущения полученного решения. Алгоритм основан на построении лучших начальных решений и схем модификации маршрутов, за счет перестановок узлов и дуг внутри маршрутов и между различными маршрутами. Процедура возмущения используется для избегания локальных оптимумов путем удаления некоторых маршрутов и вставки новых в текущее решение. Авторы провели вычислительные эксперименты на эталонных задачах и получили ряд лучших решений.

Goksal и др. [291] (2013) предложили модификацию алгоритма роя частиц (particle swarm optimization PSO), в котором локальный поиск осуществляется

методом спуска с чередующимися окрестностями (VND). В алгоритме для сохранения разнообразия роя реализована стратегия отжига. Эффективность алгоритма исследована экспериментально.

Liu R. и др. [372] (2013) рассматривают задачу планирования маршрутов транспортных средств для материально-технического обеспечения медицинского обслуживания на дому. Задача состоит из доставки лекарственных препаратов и изделий медицинского назначения от аптеки и больницы по домам пациентов и обратной доставки неиспользованных лекарственных средств, изделий медицинского назначения и биологических образцов от пациентов. Задача сформулирована в виде VRPSPDTW и для ее решения предложены две модели смешанного целочисленного программирования, генетический алгоритм (GA) и алгоритм на основе метода поиска с запретами (TS). Генетический алгоритм основан на перестановке хромосом, процедуре расщепления и локальном поиске. Второй алгоритм для выбора маршрутов использует атрибуты пациентов, дополненные функцией затрат, реоптимизацию маршрутов и уровни аспирации на основе атрибутов пациентов. Алгоритмы апробированы на известных тестовых экземплярах VRPTW задач.

Kecеci и др. [336] (2014) предложены математическая модель задачи VRPSPD и гибридный эвристический алгоритм, основанный на методах моделируемого отжига (SA) и локального поиска (LS) для решения задач средней и большой размерности. По утверждению авторов, результаты проведенной серии экспериментов показали вычислительную эффективность и пригодность разработанного алгоритма для получения достаточно хороших решений.

Работа Wang и др. [471] (2015) адресована решению задачи VRPSPDTW, сформулированной в виде смешанного целочисленного программирования. Предложена параллельная версия алгоритма моделируемого отжига (SA), включающая вставки на основе эвристик остаточной грузоподъемности (Residual Capacity) и радиальной доплаты (Radial Surcharge). Результаты расчетов представлены для 65 эталонных тестовых задач Wang и Chen [470], решенных генетическим алгоритмом (GA). Разработанный алгоритм SA для числа клиен-

тов от 10 до 50 показал те же результаты (100% совпадение) по количеству используемых транспортных средств, что и генетический алгоритм Wang и Chen. Для 100 клиентов алгоритмом SA были получены лучшие результаты для 12 индивидуальных задач, а для остальных — те же решения. Из 44 тестовых задач с одинаковыми решениями по количеству транспортных средств в 16 задачах получены лучшие решения по суммарному пройденному расстоянию, чем у алгоритма GA, а для 7 задач — равные решения. Кроме того, были найдены решения для 30 задач с числом клиентов 200, 400, 600, 800 и 1000, которые могут служить новым ориентиром для задачи VRPSPDTW.

Avci и Topaloglu [164] (2016) для задачи VRPSPD с гетерогенным парком транспортных средств разработали гибридный алгоритм локального поиска, в котором интегрированы немонотонная стратегия регулировки порога и поиск с запретами. Пороговая функция, используемая в алгоритме, имеет адаптивный характер, что делает ее само - настраиваемой. Реализация алгоритма очень проста, так как не требует настройки параметров за исключением длины списка запретов. Предложенный алгоритм протестирован на ряде случайно сгенерированных индивидуальных задач.

В работе Zachariadis и др. [488] (2016) рассматривается задача VRPSPD (Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries and Two-Dimensional Loading Constraints, 2L-SPD), в которой одновременно с маршрутизацией выполняется упаковка прямоугольных грузов с различной длиной и шириной в заданную полезную площадь грузового отсека транспортных средств. Предполагается использование однородного парка транспортных средств с одинаковой грузоподъемностью и полезной площадью для погрузки и выгрузки. Габариты грузов не могут перекрываться, а грузы упаковываются только в плоскости грузового отсека. В предложенном подходе решения задачи объединены модели маршрутизации и двумерной упаковки (Two-Dimensional Bin Packing Problem) и используется двухэтапный алгоритм, состоящий из конструктивной эвристики локального поиска для генерации начального решения (в котором требования всех клиентов могут быть не удовлетворены) и трех опера-

торов поиска окончательного решения. Для диверсификации поиска применяется простая структурированная схема на основе критериев аспирации метода поиска с запретами (tabu search). При решении задачи минимизируются издержки маршрутизации и определяются возможные шаблоны для погрузки перевозимых грузов для обеспечения выполнения требований всех клиентов. Шаблоны конструируются эвристическим алгоритмом двумерной упаковки. Кроме основной модели авторами предложена ее модификация с учетом правила погрузки LIFO. Эффективность предложенных алгоритмов проверена на эталонных задачах.

2.3.7. Формулировка задачи построения маршрутов с расщепленной доставкой и неоднородным парком транспортных средств

В классической задаче с расщепленной (раздельной) доставкой (The Split Delivery Vehicle Routing Problem, SDVRP), формулируемой на полном неориентированном графе $G(N, E)$, предполагается использование неограниченного однородного рабочего парка транспортных средств с одинаковой грузоподъемностью. Каждый клиент может посещаться несколькими транспортными средствами, а запрос клиентов может быть больше грузоподъемности транспортных средств. Каждое транспортное средство начинает и заканчивает свой тур в одном и том же депо. Заданы стоимости проезда по ребрам (длины ребер), которые удовлетворяют неравенству треугольника. Задача заключается в построении маршрутов, обслуживающих запросы всех клиентов без нарушения ограничений на грузоподъемность транспортных средств и имеющих минимальную стоимость.

Естественно предположить, что расщепленная перевозка грузов наиболее характерна для реальных транспортных сетей и необходима при оперативном управлении перевозками в условиях ограниченной провозной возможности имеющегося парка транспортных средств и значительного колебания потоков. Впервые задача SDVRP была рассмотрена M. Dror и P. Trudeau в 1989 г. [244, 245], которые указали на возможность сокращения транспортных издержек за

счет дробления потоков и их расщепленной доставки клиентам. C. Archetti и др. [151, 152] провели анализ максимально возможной экономии транспортных издержек и представили результаты вычислительных экспериментов, показывающих, как экономия зависит от характеристик индивидуальных задач оптимизации. Было показано, что расщепленная доставка в отдельных случаях позволяет улучшить решение нерасщепленной задачи до 2 раз. Обзоры по классической задаче SDVRP можно найти в работах [153, 299], а отдельные методы и алгоритмы ее решения в [147, 154, 155, 178, 182, 190, 206, 340, 341].

Сформулируем задачу SDVRP с неограниченным и ограниченным неоднородным парком транспортных средств. Известны объемы доставляемых мелкопартионных грузов $a_i \in Z^+$, $i = \overline{1,n}$, измеряемые количеством единиц, количество m_k и грузоподъемность $Q_k \in Z^+$ транспортных средств типа $k \in V$, $V = \{1,\ldots,K\}$, $k = \overline{1,K}$. Пусть, как и ранее $N = \{0,1,\ldots,n\}$, $C = \{1,\ldots,n\}$, а объемы грузов могут превышать грузоподъемность транспортных средств. Известны F_k, f_i^k и f_{ij}^k, $i, j = \overline{0,n}$, $k = \overline{1,K}$ — соответственно фиксированные затраты на приобретение и обслуживание транспортных средств, переменные затраты на обработку и транспортировку грузов. Введем переменные y_i^k, определяющие количество единиц груза доставленных клиенту i транспортным средством k, целочисленные переменные u_i^k для исключения подциклов, которые определяют порядковый номер элемента i в маршруте k и булевы переменные x_{ij}^k, $x_{ij}^k = 1$, если транспортное средство k движется от клиента i к клиенту j и $x_{ij}^k = 0$ в противном случае. Требуется найти минимум функции

$$F_{CDVRP} = \sum_{k \in V} F_k \sum_{j \in C} x_{0j}^k + \sum_{k \in V} \sum_{i \in N} f_i^k(y_i^k) + \sum_{k \in V} \sum_{i,j \in N} f_{ij}^k x_{ij}^k, \tag{2.60}$$

при ограничениях

$$\sum_{k \in V} \sum_{i \in N} x_{ij}^k \geq 1, \quad \forall j \in N, \tag{2.61}$$

$$\sum_{i \in N} x_{ij}^k - \sum_{i \in N} x_{ji}^k = 0, \quad \forall j \in N, \quad \forall k \in V, \tag{2.62}$$

$$u_i^k - u_j^k + nx_{ij}^k \le n-1, \ \ \forall i, j \in C, \ \forall k \in V, \tag{2.63}$$

$$y_i^k \le a_i \sum_{j \in N} x_{ij}^k, \ \ \forall i \in C, \ \forall k \in V, \tag{2.64}$$

$$\sum_{k \in V} y_i^k = a_i, \ \ \forall i \in C, \tag{2.65}$$

$$\sum_{i \in C} y_i^k \le Q_k, \ \ \forall k \in V, \tag{2.66}$$

$$x_{ij}^k \in \{0,1\}, \ \ \forall i, j \in N, \ \ \forall k \in V, \tag{2.67}$$

$$y_i^k, u_i^k \ge 0 \text{ и целые}, \ \ \forall i \in C, \ \ \forall k \in V. \tag{2.68}$$

Ограничения (2.63) на запрет подциклов также как и в задаче (2.15)-(2.21) можно записать в виде

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \le |S| - 1, \ \ \forall S \subseteq C, \ \ |S| \ge 2, \ \ \forall k \in V,$$

где S — кластер (подмножество клиентов), обслуживаемый транспортным средством типа k. Для ограниченного парка транспортных средств к задаче добавятся ограничения

$$\sum_{j \in C} x_{0j}^k \le m_k, \ \ \forall k \in V. \tag{2.69}$$

Целевая функция (2.60) определяет общие затраты маршрутизации. Ограничения (2.61) и (2.62) гарантируют, что каждый клиент j посещается хотя бы одним транспортным средством типа k, и это транспортное средство после прибытия к клиенту и выгрузки его груза должно обязательно покинуть клиента. Ограничения (2.63) запрещают подциклы, не проходящие через депо, а ограничения (2.64)-(2.66) распределяют запросы клиентов среди транспортных средств.

Постановка задачи (2.60)-(2.69) может быть расширена на случаи построения комбинированных маршрутов [484] и дополнительных ограничений на временные окна, продолжительность и длину маршрутов и т.п.

Одними из первых задачу с неоднородным парком транспортных средств и расщепленными поставками рассмотрели Tavakkoli и др. (2007) [455], в которой стоимость используемого рабочего парка зависит от общей незаполненной грузоподъемности и количества транспортных средств. Авторы сформулировали

задачу в виде смешанного целочисленного линейного программирования и для ее решения разработали гибридный алгоритм моделируемого отжига (SA). Для экспериментального исследования алгоритма были сгенерированы тестовые задачи с числом клиентов от 6 до 100. Для задач небольшой размерности разработанный алгоритм показал конкурентоспособные результаты с алгоритмом ветвей и границ. На задачах большой размерности сравнивались нижние границы решения, полученные путем построения большого тура с посещением всех клиентов. Belfiore и Yoshizaki (2013) [179] разработали scatter-search алгоритм для решения задачи с неограниченным рабочим парком и временными окнами. В их алгоритме начальные решения формируются с помощью двух конструктивных эвристик, а процедуры scatter-search используются для диверсификации и интенсификации поиска лучших решений. Авторы протестировали свой алгоритм на эталонных задачах FSMTW Liu и Shen (1999) [371] и эталонных задачах Ho и Haugland (2004) [306] со 100 клиентами для которых получен ряд известных решений.

2.3.8. Задача построения доставочных маршрутов с неоднородным парком транспортных средств, размещением депо и временными окнами

В заключение этого раздела рассмотрим содержательную постановку важной задачи, в которой требуется выбрать места дислокации депо, определить состав парка транспортных средств и построить маршруты обслуживания клиентов с учетом ограничений на временные окна (The Fleet Size and Mix Location-Routing Problem with Time Windows, FSMLRPTW). Такая задача может возникнуть при перспективном планировании распределенной сети перевозок мелкопартионных грузов в крупных магистральных узлах, когда на территории магистрального узла нет возможности разместить депо, а количество клиентов очень велико (например, в столице страны и областных центрах). Предполагается, что места размещения клиентов во внутренней зоне магистрального узла известны и заданы географические координаты точек строительства потенциальных депо. Задача заключается в определении подмножества депо, назначении

каждому депо обслуживаемых клиентов и построении для каждого депо кольцевых маршрутов неоднородных транспортных средств. Каждый клиент должен посещаться только один раз в пределах заданного временного окна, а загрузка транспортного средства не должна превышать его грузоподъемности. Целевая функция задачи на минимум включает четыре составляющих: капитальные затраты на строительство депо, фиксированные затраты на приобретение и обслуживание транспортных средств, затраты на погрузку и выгрузку грузов и переменные эксплуатационные расходы на транспортировку. Все затраты должны быть приведены к сопоставимому виду за заданный период времени.

В диссертационной работе Ç. Koç [347] и статье Ç. Koç и др. [348] приведены обзор по задаче FSMLRPTW, ее формулировка в виде целочисленного программирования с системой ограничений, представленной линейными неравенствами и гибридный алгоритм эволюционного поиска (Hybrid Evolutionary Search Algorithm, HESA) с введением нескольких адаптивных процедур нахождения окрестностей поиска локальных экстремумов целевой функции (Location Heterogeneous Adaptive Large Neighborhood Search, L-HALNS). Предложены процедуры построения начальных решений, разбиения допустимых решений, обучения и схема диверсификации через процедуру мутации решений. В работах представлены также результаты решения задачи на известных эталонных примерах. Целочисленная задача решалась пакетом CPLEX 12.5. Алгоритм HESA был реализован на C++. Главной целью эксперимента было сравнение результатов решения задачи полученных с помощью пакета CPLEX 12.5 и алгоритмом HESA для индивидуальных задач малой, средней и большой размерности. Численные результаты на наборе эталонных задач, содержащих до 100 клиентов и 10 потенциальных депо, показали, что предложенный авторами алгоритм способен найти приближенное решение в пределах 0,05% от оптимального для задач малой размерности и за приемлемое время дает лучшие решения по сравнению с решателем CPLEX для задач большей размерности, что позволяет использовать разработанный алгоритм в практических приложениях.

Выводы по второму разделу

1. Для эффективного управления процессами обработки и транспортировки мелкопартионных грузов во внутренних зонах магистральных узлов государственные и частные транспортные предприятия должны оптимизировать долгосрочные, тактические и оперативные решения, используя современные методы исследования операций, комбинаторной оптимизации и информационно-аналитические системы поддержки принятия решений (ИАС ППР). Уменьшение приведенных эксплуатационных затрат за счет оптимизации решений позволяет снижать тарифы на перевозку грузов, поддерживать здравую конкуренцию среди перевозчиков на оказание транспортных услуг и постоянно повышать качество обслуживания хозяйственных предприятий и населения.

2. При определении структуры внутриузловой сети перевозок и ее математической модели должны учитываться реальные географические особенности и характеристики участков дорог транспортной сети, трудно-формализуемые внешние факторы и возможные последствия на транспортные процессы от непредсказуемых стихийных бедствий. Модель физической структуры сети должна формироваться при участии опытных экспертов и диспетчеров транспортных перевозок для каждого магистрального узла. Для крупных магистральных узлов должна быть поставлена задача о территориальном размещении нескольких депо и привязки к ним обслуживаемых узлов четвертого типа.

3. Предложены основные принципы и схемы организации перевозок во внутренних зонах магистральных узлов и определены технические и экономические особенности реальных транспортных процессов обработки и транспортировки мелкопартионных грузов, которые должны быть учтены при формировании целевых функций математических моделей задач маршрутизации на уровнях долгосрочного и текущего планирования и оперативного управления.

4. В большинстве известных работ, посвященных решению задач маршрутизации на транспорте, рассматриваются идеализированные математические модели, в которых не учитываются многие ограничения, присущие реальным процессам обработки и транспортировки грузов. Часто принимается евклидово

расстояние между узлами сети перевозок, когда выполняется правило треугольника, а такие важные параметры как расстояние, время поездки, транспортные расходы, время обслуживания и другие, присутствуют в некотором абстрактном виде и моделируются константами. В постановках задач внутриузловых перевозок должны присутствовать все ограничения и параметры, которые позволят рассчитать близкие к фактическим технические и экономические показатели функционирования перевозок во внутренних зонах магистральных узлов.

5. Для решения задач построения маршрутов с неоднородным парком транспортных средств наиболее часто применяются классические эвристические и метаэвристические алгоритмы, что объясняется с одной стороны NP-трудностью решаемых задач, а с другой — относительно низкой трудоемкостью разработки таких алгоритмов. Однако следует учитывать, что большинство эвристических алгоритмов на разных экземплярах индивидуальных задач оптимизации могут дать решения, сколь угодно отличающиеся от глобального оптимума. Поэтому для решения задач большой размерности (более 100 клиентов) предпочтительнее использовать гибридные алгоритмы, в которых сочетаются в различных комбинациях точные (ветвей и границ, ветвей и отсечений, ветвей отсечений и цен, генерации столбцов, разбиения множества, динамического программирования) и многочисленные эвристические и метаэвристические методы и подходы. Сегодня можно утверждать, что разработка гибридных алгоритмов для решения NP-трудных кластерных задач маршрутизации стала общепризнанной в мировой практике [286, 459]. В последние годы наблюдается также тенденция к построению унифицированных алгоритмов и портала-сервера, способных решать большой класс задач маршрутизации с возможностью учета многих реальных ограничений и параметров [468].

6. На основании проведенного обзора и анализа известных математических моделей разработано несколько вариантов математических постановок задачи проектирования доставочных, сборочных и комбинированных маршрутов транспортных средств для перевозки мелкопартионных грузов. Для решения поставленных задач могут быть применены точные, эвристические и метаэвристиче-

ские методы и алгоритмы, реализованные в многочисленных коммерческих, например, IBM ILOG CPLEX, GAMS , AIMMS, Gurobi Optimizer [314] и некоммерческих, например, ABACUS, COIN-OR, GLPK, lp_solve [315] пакетах программ смешанного и целочисленного линейного программирования. Многие из них бесплатно доступны на сервере NEOS (https://neos-server.org/neos/). Учитывая иерархическую структуру магистральной сети перевозок и, как следствие, небольшое количество узлов четвертого типа во внутренних зонах магистральных узлов, предпочтение следует отдать точным и гибридным методам и алгоритмам. В частности, с успехом могут использоваться алгоритмы, предложенные R. Baldacci и др. [169-172], которые в настоящее время принято считать одними из лучших точных универсальных алгоритмов, способных находить оптимальные решения многих задач маршрутизации с числом клиентов до 100.

7. К перспективным направлениям решения задач маршрутизации следует отнести разработку стохастических моделей и алгоритмов для долгосрочного планирования, учитывающих риски вложения инвестиций в развитие парка транспортных средств, динамических моделей текущего планирования и оперативного управления для определения границ экономической эффективности уже полученных решений на заданные промежутки времени при колебании потоков грузов и изменении параметров транспортной модели. Оперативная информация в этом случае может поступать в ИАС от GPS с бортов транспортных средств, системы электронных заказов по Интернет и устройств мобильной связи (сотовых телефонов, планшетов, смартфонов и т.п.). Большой интерес представляет также создание единой общенациональной базы данных в стандартизированном формате на основе одного структурированного языка (например, XML) с трудными тестовыми примерами для различных классов типовых задач маршрутизации.

8. Материалы этого раздела составляют методологическую основу для разработки современного математического обеспечения решения задач долгосрочного, текущего и оперативного планирования и управления во внутренних зонах магистральных узлов.

РАЗДЕЛ 3. ЗАДАЧИ УПАКОВКИ И ВЫБОРА СТРУКТУРЫ МАГИСТРАЛЬНОЙ СЕТИ

В первом подразделе предложен способ формирования справочной матрицы объединения мелкопартионных потоков корреспонденций при решении задачи оптимизации упаковки в многопродуктовой сети с ограничениями на время доставки и (или) число объединений потоков. На основании доказанных утверждений разработаны эффективные (полиномиальные) вычислительные алгоритмы для определения узлов объединения и объединенных потоков для всех корреспондирующихся пар в сети. Алгоритмы могут быть использованы внутри основных схем оптимизации для расчета времени доставки корреспонденций получателю и проверки соответствующих ограничений при решении различных задач упаковки в транспортных сетях и опорных сетях передачи данных.

Во втором подразделе рассмотрена задача упаковки элементов квадратной матрицы, заданных целыми положительными числами, в блоки фиксированного размера. Предложена постановка задачи и исследована трудоемкость полного перебора ее решений. Доказано, что задача является NP-полной, путем полиномиального сведения к ней NP-полной целочисленной задачи о многопродуктовом потоке минимальной стоимости.

В третьем подразделе предложена постановка задачи упаковки с нелинейными функциями затрат, которая возникает при сортировке и упаковке мелкопартионных корреспонденций в транспортные блоки (физические или виртуальные контейнеры) в магистральных транспортных сетях и сетях передачи данных. Обсуждаются особенности задачи и приведены алгоритмы ее решения. Проведен анализ сходимости и временной сложности алгоритмов. С помощью вычислительных экспериментов исследована сравнительная эффективность алгоритмов. Экспериментально показано, что результаты решения задачи, полученные различными стратегиями оптимизации на сетях, содержащих до 500 узлов, отличаются не более чем на 2.65%.

В четвертом подразделе рассматривается и исследуется дискретная задача выбора пропускных способностей дуг, которая возникает при решении задачи упаковки мелкопартионных потоков. Доказано, что при решении задачи упаковки нелинейная функция транспортных затрат может быть заменена на функцию удельной стоимости транспортировки транспортных блоков от расстояния и грузоподъемности транспортных средств или пропускной способности каналов связи, а задача выбора пропускных способностей может решаться независимо при оптимизации распределения сформированных потоков транспортных блоков.

В пятом подразделе рассматривается задача выбора структуры магистральной сети, в результате решения которой определяются состав и количество узлов каждого типа. Решение задачи основано на применении ранее рассмотренных алгоритмов упаковки. Приведены общая схема решения задачи и результаты вычислительного эксперимента.

В приложении Б рассматриваются модели и алгоритмы решения задачи распределения и упаковки мелкопартионных корреспонденций в зональных сетях, являющихся отдельными фрагментами магистральной сети.

В приложении В рассматриваются два способа балансировки матрицы контейнерных потоков при решении задачи перевозки мелкопартионных грузов в транспортной сети. Необходимость балансировки возникает из-за неравенства суммы исходящих и входящих потоков контейнеров в узлах сети. Предложена математическая модель и алгоритм решения задачи развозки порожних контейнеров, которые могут быть использованы для балансировки матрицы контейнерных потоков и последующего решения задачи распределения и маршрутизации потоков груженых и порожних контейнеров. Проведен обзор и анализ современных методов и алгоритмов решения транспортной задачи. Экспериментально показано, что оптимальная балансировка по сравнению с симметричной балансировкой позволяет значительно сократить суммарные затраты на транспортировку и обработку порожних контейнеров (на сетях от 100 до 4000 узлов в 17 и 174 раза соответственно).

Ключевые слова: оптимизация упаковки, целочисленный многопродуктовый поток, трудоемкость полного перебора, полиномиальная сводимость, NP-полные задачи, магистральные коммуникационные сети, мелкопартионные потоки корреспонденций, эвристические алгоритмы, сходимость и временная сложность алгоритмов, транспортная сеть, модели и алгоритмы решения транспортной задачи

3.1. Справочная матрица объединения потоков

Задача оптимизации упаковки возникает при проектировании и анализе функционирования многопродуктовых сетей с мелкопартионными дискретными потоками корреспонденций, передаваемыми по сети в некоторых транспортных блоках заданного размера (объема). При этом размеры отдельно взятых потоков, выраженные количеством единиц потока из узлов отправления (источников) в узлы назначения (стоки), значительно меньше размера транспортного блока. Задача оптимизации упаковки заключается в объединении (слиянии) нескольких исходящих из каждого узла потоков корреспонденций с разными адресами назначения в общие транспортные блоки. В качестве критерия оптимизации в таких задачах может быть принят, например, минимум приведенных затрат на транспортировку (передачу) и обработку всех потоков, заданных на сети, а в ограничениях могут учитываться время на доставку потоков получателю (адресату), пропускные способности узлов и ребер сети. При решении задачи некоторые потоки могут несколько раз объединяться в разных узлах с другими потоками, поэтому необходимо иметь такую структуру данных, которая позволяла бы эффективно запоминать и определять узлы объединения и объединенные потоки для всех корреспондирующихся пар.

Целью данного подраздела является выбор структуры данных для представления процесса объединения и разработка алгоритмов нахождения узлов объединения и объединенных потоков.

3.1.1. Постановка задачи

Рассмотрим содержательную постановку задачи [18, 19]. Пусть задана не-

ориентированная сеть $G(N,P)$ с множеством узлов N, $n=\left|N\right|$ и множеством ребер P, $p=\left|P\right|$, где n и p соответственно число узлов и ребер сети, а $\left|\cdot\right|$ — знак мощности множества. На сети задана целочисленная матрица потоков $A=\left\|a_{ij}\right\|_{n\times n}$, $a_{ii}=0$, $i=\overline{1,n}$, которые подлежат единовременной передаче из источников i в стоки j, $i,j=\overline{1,n}$. Потоки должны передаваться по сети в транспортных блоках объема $\omega >> a_{ij}$. При этом предполагается, что каждый поток может быть упакован в транспортный блок только целиком.

При решении оптимизационной задачи над потоками a_{ij} итеративно выполняются следующие преобразования:

$$\begin{aligned} a_{ik}^{'} &\leftarrow a_{ik}+a_{ij}, \\ a_{kj}^{'} &\leftarrow a_{kj}+a_{ij}, \\ c_{ij} &\leftarrow k, a_{ij}\leftarrow 0. \end{aligned} \qquad (3.1)$$

Где «$\leftarrow$» знак операции присваивания; k — узел, через который выполняется преобразование; c_{ij} — элементы справочной матрицы объединения потоков $C=\left\|c_{ij}\right\|_{n\times n}$, которые определяются так:

$$c_{ij}=\begin{cases} k, \text{если поток } a_{ij} \text{ сливается с потоком } a_{ik}, \\ i, \text{если поток } a_{ij} \text{ непосредственно направляется в узел } j, \\ 0, \text{если } i=j. \end{cases} \qquad (3.2)$$

На различных шагах работы алгоритма оптимизации необходимо (например, для расчета времени доставки корреспонденций получателю) определять последовательность $\Omega_{ij}=\{(i,k_1),(k_1,k_2),...,(k_m,j)\}$ с промежуточными узлами $\{k_1,k_2,...,k_m\}$, в которых выполняется дополнительная обработка каждого из потоков a_{ij} и их общее число $\nu_{ij}=\left|\{k_1,k_2,...,k_m\}\right|$. Кроме того, при выборе очередного потока a_{ij} для преобразования (3.1) может потребоваться нахождение множества других потоков $\{a_{ij}^{*}\}$, которые уже были слиты с потоком a_{ij} на предыдущих итерациях.

3.1.2. Основные результаты и алгоритм решения

Докажем следующие утверждения.

Теорема 3.1. Для любого $a_{ij} \in \{a_{ij} \mid c_{ij} \neq i \wedge c_{ij} \neq 0\}$ построение матрицы C в соответствии с (3.2) позволяет найти последовательность Ω_{ij} и установить число дополнительных обработок потока a_{ij}, где «$\wedge$» — знак конъюнкции (логического «и»).

Доказательство. Для доказательства теоремы сначала покажем, что определение Ω_{ij} сводится к построению и последующему прохождению в глубину бинарного дерева с корнем в вершине (i, j). Для этого рассмотрим пример. Пусть поток $(1,9)$ был подвергнут преобразованиям так, как показано на рис. 3.1. Тогда элементы матрицы C, отражающие это преобразование, будут следующими: $c_{19}=3$, $c_{13}=2$, $c_{12}=1$, $c_{23}=2$, $c_{39}=8$, $c_{89}=8$, $c_{38}=5$, $c_{35}=4$, $c_{34}=3$, $c_{45}=4$, $c_{58}=6$, $c_{56}=5$, $c_{68}=6$. Зная эти элементы, легко построить дерево с корнем в $(1,9)$, описывающее преобразование потока $(1,9)$ (рис. 3.2). При построении дерева сначала строятся левые поддеревья до тех пор, пока не будет найден элемент $c_{ij}=i$, затем происходит возврат на уровень выше и строятся правые поддеревья. Для любого корня правого поддерева процесс построения поддеревьев может быть продолжен, если для этого корня соответствующий $c_{ij} \neq i$.

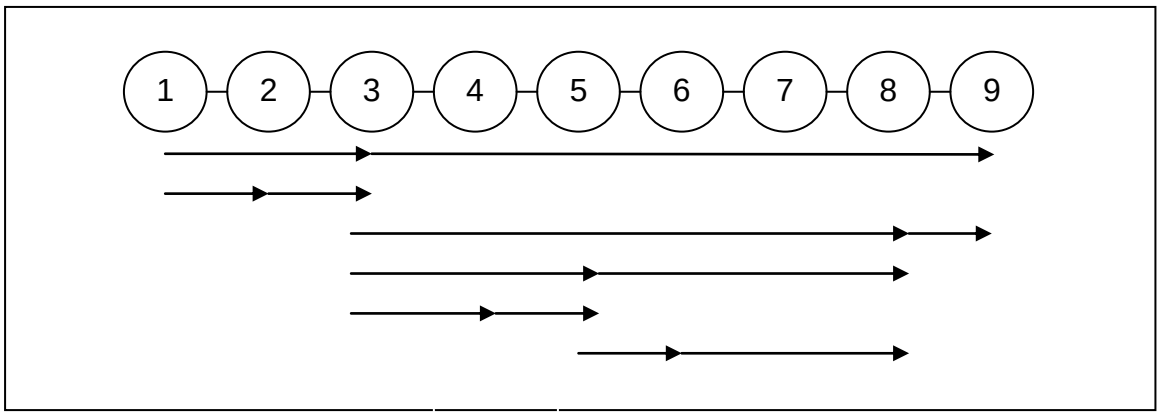


Рис. 3.1. Пример преобразования потока $(1,9)$

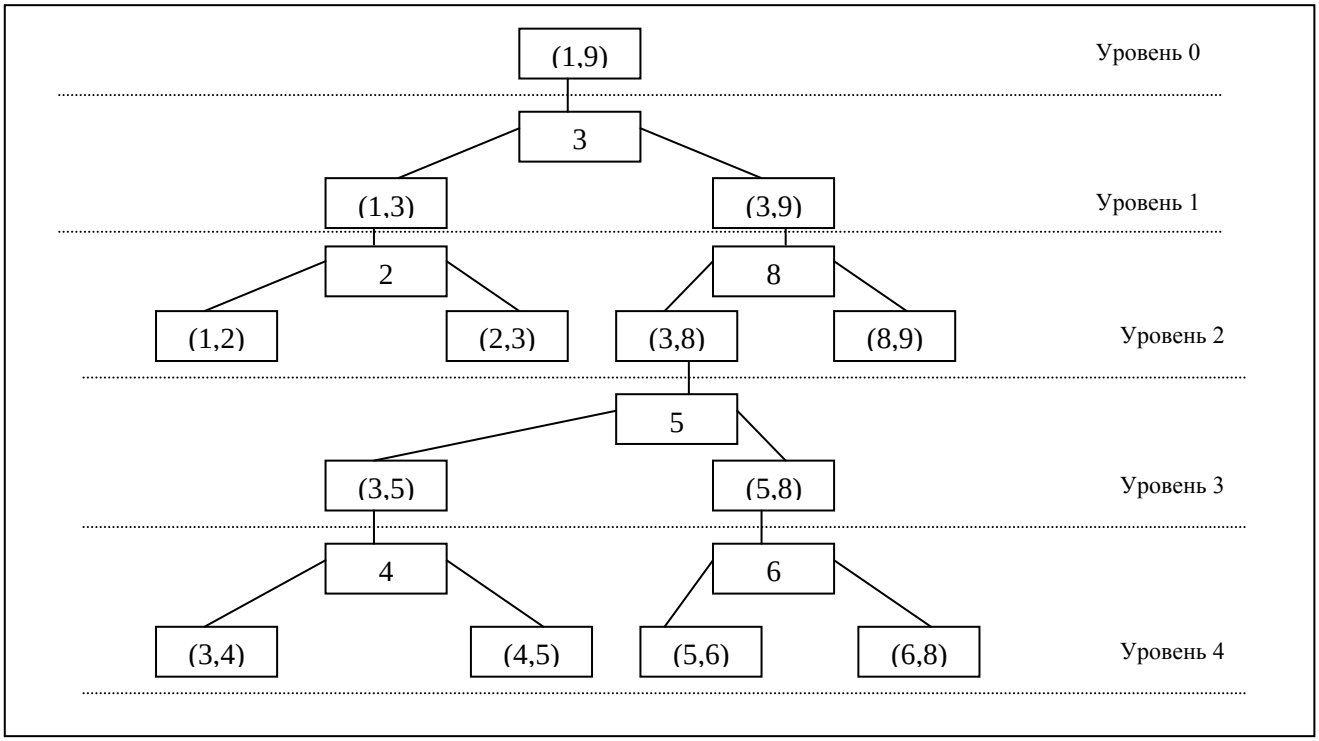


Рис. 3.2. Дерево преобразования потока (1,9)

Лемма 3.1. Листья бинарного дерева, построенного прохождением в глубину с использованием элементов матрицы C, порождают последовательность Ω_{ij} при их просмотре слева направо. Доказательство леммы следует из правил построения бинарного дерева и того факта, что для каждого листа дерева соответствующий $c_{ij} = i$. Последовательность листьев c_{ik_1}, $c_{k_1k_2}$,…, $c_{k_{m-1}k_m}$, c_{k_mj} дает искомую последовательность $\Omega_{ij} = \{(i,k_1),(k_1,k_2),\ldots,(k_m,j)\}$. ■

Для примера на рис. 3.1 такая последовательность будет следующей:

$$\Omega_{19} = \{(1,2),(2,3),(3,4),(4,5),(5,6),(6,8),(8,9)\}.$$

Утверждение леммы позволяет построить эффективный алгоритм построения и прохождения бинарного дерева, в котором поддеревья генерируются в порядке их просмотра. Пусть $St(\xi)$ — стек, обслуживаемый по правилу «последний пришел - первый вышел», где ξ — указатель текущего размещения стека; $F(m)$ — вектор, в котором формируется последовательность узлов $\{i,k_1,k_2,\ldots,k_m,j\}$; «$\vee$» — знак дизъюнкции (логического «или»).

Алгоритм 3.1

1. $m \leftarrow 1;\ F(m) \leftarrow i$.

2. $\xi \leftarrow 1;\ St(\xi) \leftarrow j$.

3. $k \leftarrow i;\ l \leftarrow j$.

4. Если $c_{kl} \neq k \wedge ((a_{kl} \neq 0 \wedge \xi = 1) \vee a_{kl} = 0)$, то выполнить пп. 5-7, иначе перейти к п. 8.

5. $\xi \leftarrow \xi + 1$.

6. $St(\xi) \leftarrow c_{kl}$.

7. $l \leftarrow c_{kl}$; перейти к п. 4.

8. $m \leftarrow m + 1$.

9. $F(m) \leftarrow l$.

10. Если $\xi > 1$, то перейти к п. 11, иначе к п. 14.

11. $k \leftarrow St(\xi)$.

12. $l \leftarrow St(\xi - 1)$.

13. $\xi \leftarrow \xi - 1$; перейти к п. 4.

14. $\nu_{ij} \leftarrow m - 2$.

15. Стоп.

В строке 4 алгоритма 3.1 дополнительные условия $((a_{kl} \neq 0 \wedge \xi = 1) \vee a_{kl} = 0)$ введены для ограничения роста бинарного дерева в случае, если для вновь найденного подкорня (k,l) соответствующий $a_{kl} \neq 0$.

После работы алгоритма значение ν_{ij} равно числу транзитных узлов от i до j, в которых выполняется дополнительная обработка потока a_{ij}.

Вернемся к доказательству теоремы. Для этого рассмотрим связь формирования матрицы C с преобразованием потоковой матрицы A. Преобразование любого потока a_{ij} заключается в выполнении действий (3.1). Поскольку дробление потока запрещено, потоки $(a_{ik} + a_{ij})$ и $(a_{kj} + a_{ij})$ можно рассматривать как единые потоки, возникшие соответственно в узлах i, k и адресованные в узлы

k и j. Применяя операцию (3.1) для потоков $a_{ik}^{'}$ и $a_{kj}^{'}$ получим новое разбиение, для которого также можно выполнить преобразование. По индукции легко показать, что во всех последующих разбиениях учитывается первоначальный поток a_{ij}. Для любого потока, подвергнутого преобразованию, значение $c_{ij} = k \neq i$ и указывает узел, в котором поток a_{ij} первый раз был соединен с потоками a_{ik} и a_{kj}. Из леммы следует, что для получения первого узла, в котором выполняется дополнительная обработка потока a_{ij}, необходимо полностью построить каждое левое поддерево, используя в качестве корня дерева элемент c_{ij}. Восстановление всех остальных промежуточных узлов, в которых обрабатывается поток a_{ij}, осуществляется алгоритмом 3.1 при прохождении всего дерева, что и доказывает теорему. ■

При работе со справочной матрицей объединения потоков в основном алгоритме оптимизации при выборе очередного потока a_{ij} для преобразования (3.1) может потребоваться нахождение множества других потоков $\{a_{ij}^{*}\}$, которые уже были объединены с потоком a_{ij} на предыдущих итерациях, так как преобразование a_{ij} приводит к увеличению времени доставки потока получателю не только для a_{ij}, но и для множества $\{a_{ij}^{*}\}$. Оказывается, что для определения множества $\{a_{ij}^{*}\}$ для любого a_{ij}, достаточно воспользоваться справочной матрицей C, формируемой согласно (3.2).

Теорема 3.2. Для любого потока a_{ij}, над которым выполняется преобразование (3.1), множество $\{a_{ij}^{*}\}$ может быть определено из справочной матрицы C.

Доказательство. Покажем, что задача определения $\{a_{ij}^{*}\}$ эквивалентна построению некоторого дерева поиска и прохождению его снизу вверх. Прежде всего, отметим, что поток a_{ij} может быть представлен в дереве поиска самым нижним подкорнем (i, j) с двумя листьями: левым — (i,k) и правым — (k, j), где k — узел, через который выполняется преобразование потока a_{ij}. Рассмот-

рим далее, какие потоки образуют множество $\{a_{ij}^{*}\}$. Из построения C ясно, что для первых потоков, охватывающих a_{ij}, будут выполняться условия:

$$c_{\xi j} = i,\ \xi = \overline{1,n}, \tag{3.3}$$

$$c_{i\xi} = j,\ \xi = \overline{1,n}, \tag{3.4}$$

причем (3.3) означает охват слева, а (3.4) — охват справа. Для дерева поиска выполнение (3.3) означает, что для a_{ij} найден левый подкорень, находящийся на уровень выше, а выполнение (3.4), — что найден правый верхний подкорень. Поскольку в (3.3) и (3.4) $\xi = \overline{1,n}$, то может найтись несколько левых и правых подкорней, находящихся на одном уровне дерева. Это говорит о том, что дерево, которое строится, не является бинарным. Для каждого вновь полученного подкорня процесс построения новых подкорней продолжается до тех пор, пока будут выполняться условия (3.3) или (3.4).

Заметим, что число дополнительных обработок, определяющее время t_{ij} доставки любого потока, представленного подкорнем в дереве поиска, может быть получено с использованием процедуры построения и прохождения бинарного дерева в глубину, приведенной при доказательстве теоремы 3.1. Таким образом, как только найден очередной подкорень (k,l), для него необходимо построить бинарное дерево, определяющее последовательность Ω_{kl}, исходя из которой для соответствующего потока a_{kl} может быть рассчитано время доставки.

На рис. 3.3 показан пример дерева поиска для определения потоков, охватывающих поток a_{58}, который преобразовывается через узел 6 (дерево построено на сети, изображенной на рис. 3.1). Как видно из рисунка, для подкорня $(5,8)$ найдены подкорни: $(3,8)$, $(3,9)$, $(1,9)$. Соответствующие им последовательности Ω_{38}, Ω_{39}, Ω_{19} строятся прохождением бинарного дерева в глубину:

$$\Omega_{38} = \{(3,4),(4,5),(5,6),(6,8)\},\ \Omega_{39} = \{\Omega_{38},(8,9)\},\ \Omega_{19} = \{(1,2),(2,3),\Omega_{39}\}.$$

Приведем рекурсивную процедуру, осуществляющую построение дерева

исчерпывающего поиска для определения множества $\{a_{ij}^*\}$.

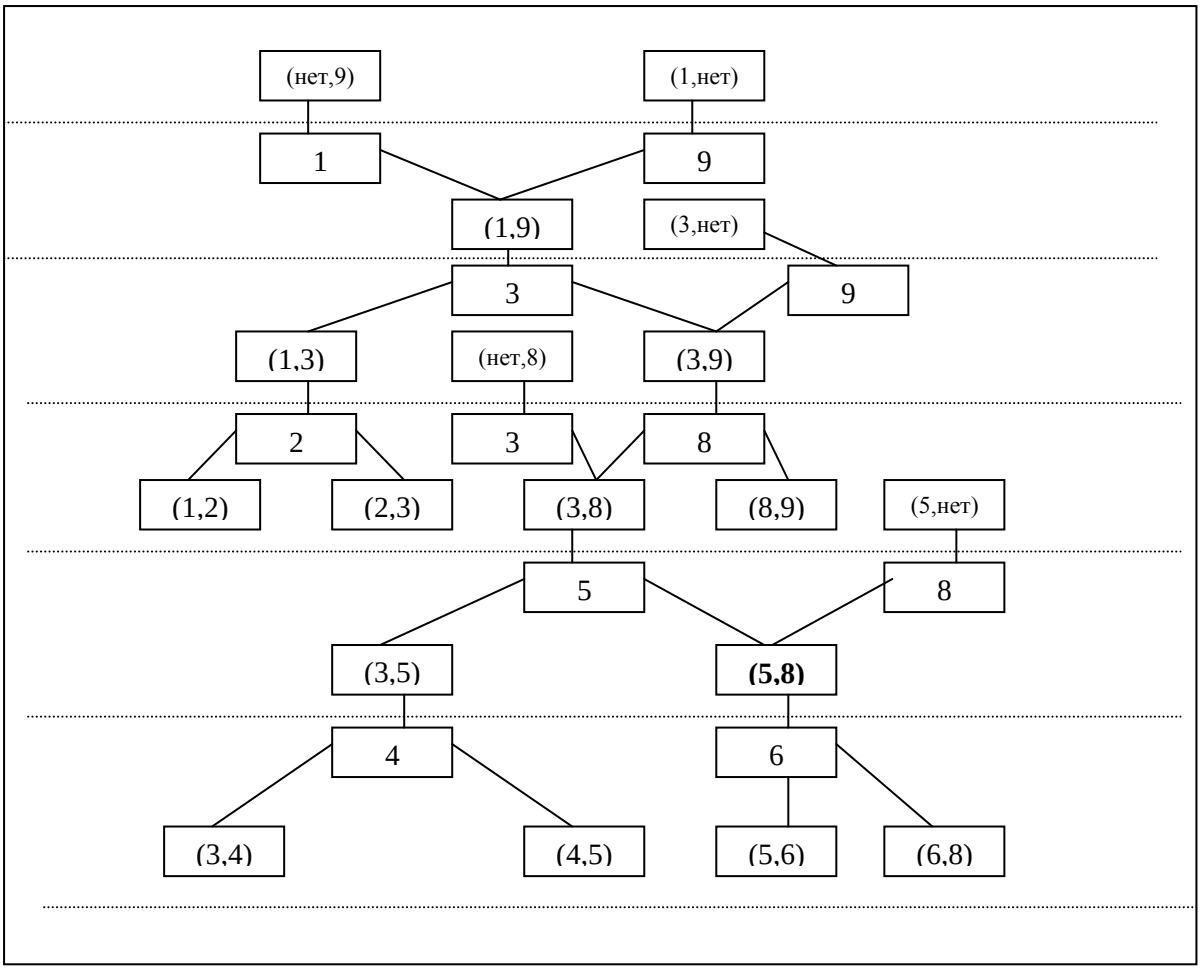


Рис. 3.3. Дерево поиска потоков, слитых с потоком $(5,8)$

Алгоритм 3.2

1. ВХОД (i, j) рекурсивный.

2. $k \leftarrow i;\ l \leftarrow j$.

3. Для $\{\xi \mid \xi = \overline{1,n}\}$ выполнить пп. 4-14.

4. Если $c_{\xi l} = k \wedge a_{\xi l} = 0$, то перейти к п.5, иначе к п. 9.

5. $\{a_{ij}^*\} = \{a_{ij}^*\} \cup a_{\xi l}$. Используя алгоритм 3.1 определить $\Omega_{\xi l}$ и $\nu_{\xi l}$.

6. Рассчитать $t_{\xi l}$.

7. Если $(t_{\xi l} > T_{\xi l} \vee \nu_{\xi l} > \nu_g)$, то $\eta \leftarrow \eta + 1$.

8. Вызвать ВХОД (ξ, l).

9. Если $c_{k\xi} = l \wedge a_{k\xi} = 0$, то перейти к п. 10, иначе к п. 14.

10. $\{a_{ij}^*\} = \{a_{ij}^*\} \cup a_{k\xi}$. Используя алгоритм 3.1 определить $\Omega_{k\xi}$ и $\nu_{k\xi}$.

11. Рассчитать $t_{k\xi}$.

12. Если $(t_{k\xi} > T_{k\xi} \vee \nu_{k\xi} > \nu_g)$, то $\eta \leftarrow \eta + 1$.

13. Вызвать ВХОД (k, ξ).

14. Перейти к п. 3.

15. ВЫХОД.

В строках 4 и 9 введены дополнительные условия $a_{\xi l} = 0$ и $a_{k\xi} = 0$ для случая, когда в матрице C имеются элементы $c_{ij} = k$, $k \neq j$, а соответствующие им потоки $a_{ij} \neq 0$, что может привести к зацикливанию алгоритма.

В приведенном алгоритме: k, l, ξ — рекурсивные переменные, для которых организуются поколения данных при каждом новом вхождении в процедуру; «$\cup$» — знак объединения множеств; t_{ij} — расчетное время доставки потоков адресату; ν_{ij} — определяет число промежуточных узлов, в которых выполняется дополнительная обработка соответствующего потока; ν_g — допустимое число дополнительных обработок; η — признак нарушения ограничений на заданное время доставки T_{ij} или число дополнительных обработок потока ν_g. Значение η указывает число таких нарушений.

Анализ алгоритма 3.2 делает прозрачным доказательство теоремы (которое легко провести по индукции), поскольку на всех уровнях рекурсии и для каждого вновь найденного подкорня выполняется исчерпывающий поиск новых подкорней, для которых справедливо (3.3) или (3.4). Это означает, что для любого потока a_{ij} алгоритм 3.2 корректно строит множество потоков $\{a_{ij}^*\}$, которые были слиты с потоком a_{ij} на предыдущих итерациях основного алгоритма оптимизации упаковки. ■

3.2. Задача упаковки без функций затрат

В подразделе рассматривается задача оптимизации упаковки, в которой элементы строк и столбцов квадратной матрицы, заданные целыми положите-

льными числами, определенным образом упаковываются в блоки фиксированного размера [118, 461]. Такая задача возникает при формализации и оптимизации процессов сортировки и упаковки мелкопартионных корреспонденций (грузов или сообщений) в транспортные блоки (физические или виртуальные контейнеры) в транспортных сетях или в сетях передачи данных. Суть прикладной задачи заключается в нахождении рациональной схемы объединения исходящих из узлов сети мелкопартионных потоков однородных грузов (сообщений) с различными адресами назначения и упаковки их в транспортные блоки (контейнеры) заданного размера в целях минимизации количества последних для транспортировки мелкопартионных корреспонденций во всей сети при определенных условиях ее функционирования.

В данном подразделе рассматривается постановка «чистой» задачи упаковки без функций затрат, которые могут быть введены для оценки стоимости процессов обработки и транспортировки мелкопартионных корреспонденций в реальных приложениях. При этом прикладная задача сводится к задаче над элементами квадратной матрицы. Исследуется переборная сложность задачи и доказано, что она является NP-трудной. Приводятся данные экспериментальных исследований, показывающие зависимость результатов решения задачи от границ изменения значений входных данных и схем перебора вариантов решения.

3.2.1. Постановка задачи

Задана матрица $A = \| a_{ij} \|_{n \times n}$ с целыми положительными элементами вне диагонали и с нулевой диагональю. Над элементами матрицы можно выполнять некую операцию объединения, соблюдая определенные ограничения. Для выполнения отдельной операции выбирается элемент $a_{ij} > 0$ и непустое множество непересекающихся индексов $\{k_1, k_2, \ldots, k_m\}$, отличных от i и j ($\bigcap_{m=1}^{n-2} k_m = \varnothing$, $n \geq 3$). В результате элементы a_{ik_1}, $a_{k_1k_2}, \ldots, a_{k_mj}$ увеличиваются на величину a_{ij} и элемент a_{ij} обнуляется. Каждый преобразованный a_{ik_1}, $a_{k_1k_2}, \ldots, a_{k_mj}$ или не прео-

бразованный элемент a_{ij} (когда $\{k_1,k_2,...,k_m\}=\varnothing$) размещается («упаковывается») в целое количество блоков размера $\omega \in Z^+$. При изменении матрицы изменяется суммарное количество блоков, необходимых для размещения всех элементов. Требуется найти такую последовательность операций объединения, чтобы в результате получить матрицу с минимальным числом блоков. Содержательно исходная матрица A задает потребности в транспортировке, а операция объединения означает, что если есть прямой поток a_{ij}, то его можно направить его через транзитные узлы $k_1,k_2,...,k_m$, соответственно изменив элементы a_{ik_1}, $a_{k_1k_2},\ldots,a_{k_mj}$, a_{ij}.

Приведем формальную постановку задачи, разделив исходное состояние матрицы и измененное. Имеется матрица целочисленных переменных $X=\| x_{ij} \|_{n\times n}$, начальные значения которой совпадают со значениями матрицы A. Требуется найти минимум функции

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\lceil x_{ij}/\omega \rceil,\ x_{ij}=\begin{cases} a_{ij}+\sum_{rs} a^*_{rs}, \text{если } a_{ij} \text{ непосредственно не объединялся} \\ \text{с другими элементами}, i,j=\overline{1,n}, \\ 0 \text{ в противном случае}, \end{cases} \tag{3.5}$$

где $\lceil x \rceil$ — наименьшее целое, большее или равное x, $\{a^*_{rs}\}$ — множество элементов (может быть пустым), объединенных с a_{ij}. Условия баланса имеют вид

$$\sum_{j=1}^{n} x_{ij}-\sum_{j=1}^{n} a_{ij}=\sum_{j=1}^{n} x_{ji}-\sum_{j=1}^{n} a_{ji},\ i=\overline{1,n}. \tag{3.6}$$

Ограничения на величину значений x_{ij} определятся так:

$$x_{ij}\le\left\lceil\frac{a_{ij}}{\omega}\right\rceil\omega,\ \ i,j=\overline{1,n}. \tag{3.7}$$

Оценим трудоемкость решения сформулированной задачи. При построении алгоритма нужно решить, как выбирать элементы x_{ij} и индексы $\{k_1,k_2,...,k_m\}$ для выполнения операции объединения. При решении задачи (3.5)-(3.7) в матрице X имеется $n(n-1)$ вариантов для выбора первого элемен-

та x_{ij}. Ясно, что после выбора этого элемента и какого либо множества индексов $\{k_1, k_2, ..., k_m\}$ существует огромное число последующих вариантов выбора x_{ij} и индексов. Учитывая, что объединение элементов x_{ij} может выполняться многократно, число вариантов полного перебора решения задачи (3.5)-(3.7), которое будет расти экспоненциально, непросто выразить в виде формулы от размерности матрицы X. Если сформулировать (3.5)-(3.7) как задачу перечисления, т.е. ответить на вопрос, сколько имеется у нее решений, то, скорее всего, она принадлежит к классу KP-полных задач [45, стр. 210]. При решении индивидуальной задачи существенную роль играют границы изменения значений элементов a_{ij} по сравнению со значением ω. Так, если $a_{ij} \geq \omega \ \ \forall i, j = \overline{1,n}$, то очевидно, что верхняя граница минимума равна $\sum_{i=1}^{n} \sum_{j=1}^{n} \lceil a_{ij} / \omega \rceil$ и определить это можно за время $O(n^2)$, поскольку операции объединения элементов могут только увеличить значение функции цели. Если выполняется $\sum_{j=1}^{n} a_{ij} \leq \omega$ $\forall i = \overline{1,n}$ и $\sum_{i=1}^{n} a_{ij} \leq \omega \ \ \forall j = \overline{1,n}$, то в этом крайнем случае существует простой алгоритм, не учитывающий ограничения (3.7), со сложностью $O(n^3)$ для нахождения нижней границы минимума функции $\sum_{i=1}^{n} \sum_{j=1}^{n} \lceil x_{ij} / \omega \rceil = 2n - 2$ с минимальным значением транзитных объединенных элементов $y = \sum_{i=1}^{n} y_i$, $y_i = \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} a_{ij}$ (см. приведенный далее алгоритм unconditional packing).

Лемма 3.2. Максимальное число элементов, преобразуемых с помощью операции объединения, не может превышать значения $n^2 - 3n + 2$.

Доказательство. Число элементов в матрице равно $n^2 - n$. Поскольку любую матрицу с ненулевыми элементами (кроме диагональных) с помощью операции объединения можно привести к виду, в котором она будет содержать ровно $2n - 2$ элемента, число преобразуемых элементов определяется как $n^2 - n - (2n - 2) = n^2 - 3n + 2$. ■

3.2.2. NP-полнота задачи

Отметим, что задача (3.5)-(3.7), сформулированная как задача распознава-

ния: существует ли последовательность объединений такая, что значение функции (3.5) не превышает $B \in Z^+$, принадлежит к классу NP, так как для нее всегда найдется сертификат с длиной, ограниченной полиномом $n^2 - 3n + 2$ (лемма 3.2). Покажем, что NP-полную целочисленную задачу о многопродуктовом потоке минимальной стоимости (minimum-cost integer multicommodity flow problem, MC IMCF) можно полиномиально свести к задаче (3.5)-(3.7). В работе [249] NP-полнота задачи MC IMCF даже для двухпродуктового целочисленного потока доказана путем сведения к ней задачи 3-выполнимости (3-SAT). Под сводимостью здесь и далее подразумевается сводимость по Карпу для соответствующих задач распознавания [45].

Приведем постановку задачи MC IMCF в форме «узлы-дуги» (node-arc or conventional). Пусть $G(N, E)$ — сеть с множеством узлов N, $n = \mid N \mid$ и множеством ориентированных дуг $ij \in E$, $e = \mid E \mid$. Задано множество продуктов (commodity) P, $p = \mid P \mid$, каждый из которых характеризуется тройкой (s_m, t_m, a_m), $m = 1, 2, ..., p$, $a_m \in Z^+$, где s_m, t_m и a_m — соответственно узлы отправления (origin) и назначения (destination), а также количество (demand) продукта $m \in P$, которое нужно доставить из узла s_m в узел t_m. При транспортировке продукта запрещено его разветвление (расщепление, дробление на части). Поскольку все продукты однородны, предполагается, что единица любого продукта при транспортировке по дуге использует единицу ее пропускной способности. Для каждой дуги сети $ij \in E$ заданы пропускная способность $w_{ij} \in Z^+$ и стоимость $c_{ij}^m > 0$, $c_{ij}^m \in Z^+$, транспортировки единицы продукта m. Введем булевы переменные x_{ij}^m, $ij \in E$, $m \in P$. Так как разветвление потоков запрещено, то $x_{ij}^m = 1$, если продукт m транспортируется по дуге $ij \in E$ и $x_{ij}^m = 0$ в противном случае.

Требуется найти

$$\min \sum_{m \in P} \sum_{ij \in E} c_{ij}^m a_m x_{ij}^m \tag{3.8}$$

при ограничениях

$$\sum_{j:ij\in E} x_{ij}^{m} - \sum_{j:ji\in E} x_{ji}^{m} = \begin{cases} 1 \text{ при } i = s_m, \\ 0 \text{ при } i \neq s_m \neq t_m, \ \forall m \in P, \forall i \in N; \\ -1 \text{ при } i = t_m, \end{cases} \quad (3.9)$$

$$\sum_{m\in P} a_m x_{ij}^{m} \leq w_{ij} \quad \forall ij \in E; \quad (3.10)$$

$$x_{ij}^{m} \in \{0,1\} \ \forall ij \in E, \ \forall m \in P. \quad (3.11)$$

Отметим, что для задачи MC IMCF известно много разновидностей постановок, например с матричной записью ограничений баланса (3.9) с использованием матрицы инциденций узлы-дуги сети [144]. Формулировки в виде «дуги-пути» (arc-path or column generation) приведены в [174, 175]. Все постановки задач MC IMCF при различных дополнительных ограничениях можно легко трансформировать одну в другую. В [174, 175] приведен также обширный библиографический обзор развития методов и алгоритмов решения многопродуктовых задач.

Теорема 3.3. Задача упаковки (3.5)-(3.7) является NP-полной.

Доказательство. Для сведения задачи (3.8)-(3.11) к задаче (3.5)-(3.7) рассмотрим индивидуальную задачу (3.8)-(3.11) на полной сети, когда индексы m всех продуктов заменены номерами узлов s_m и t_m, совпадающими с индексами дуг (т.е. для каждого продукта $st \in P$ имеется дуга $ij \in E$ и $s = i$, $t = j$).

Индивидуальная задача. Пусть $G(N, E)$ — полная сеть без петель с множеством узлов N, $n = \mid N \mid$ и множеством ориентированных дуг $ij \in E$, $e = \mid E \mid = n^2 - n$, т.е. $\forall ij \in E \ \exists ji \in E$. Каждая дуга ij имеет пропускную способность $w_{ij} \in Z^+$. Задана матрица требований (demand) $A = \| a_{st} \|_{n\times n}$ и число ω, $a_{st}, \omega \in Z^+$, $a_{st} = 0$ при $s = t$. Каждый продукт $st \in P$, $\mid P \mid = n^2 - n$ в количестве a_{st} должен быть доставлен из узла отправления s в узел назначения t. Пусть стоимости транспортировки единицы продукта $c_{ij}^{st} = 1 \ \forall ij \in E$ и $\forall st \in P$. Переменные $x_{ij}^{st} = 1$, если продукт $st \in P$ транспортируется по дуге $ij \in E$, и

$x_{ij}^{st} = 0$ в противном случае. Пусть $w_{ij} = \left\lceil \frac{a_{ij}}{\omega} \right\rceil \omega \ \forall ij \in E$. Обозначим $x_{ij} = \sum_{st \in P} a_{st} x_{ij}^{st}$ $\forall ij \in E$. Тогда индивидуальная задача записывается так

$$\min \sum_{ij \in E} \left\lceil \frac{x_{ij}}{\omega} \right\rceil \tag{3.12}$$

при ограничениях

$$\sum_{j:ij \in E} a_{st} x_{ij}^{st} - \sum_{j:ji \in E} a_{st} x_{ji}^{st} = \begin{cases} a_{st} & \text{при } i = s, \\ 0 & \text{при } i \neq s \neq t, \\ -a_{st} & \text{при } i = t, \end{cases} \quad \forall st \in P, \forall i \in N; \tag{3.13}$$

$$x_{ij} \leq \left\lceil \frac{a_{ij}}{\omega} \right\rceil \omega \quad \forall ij \in E; \tag{3.14}$$

$$x_{ij}^{st} \in \{0,1\} \ \forall ij \in E, \ \forall st \in P. \tag{3.15}$$

Сложив для каждого i равенства (3.13) по всем продуктам $st \in P$, получим уравнения баланса для узлов i, выраженные через переменные x_{ij}:

$$\sum_{j:ij \in E} \sum_{st \in P} a_{st} x_{ij}^{st} - \sum_{j:ji \in E} \sum_{st \in P} a_{st} x_{ji}^{st} = \sum_{t:it \in P} a_{it} - \sum_{s:si \in P} a_{si} \qquad \text{или} \qquad \sum_{j:ij \in E} x_{ij} - \sum_{t:it \in P} a_{it} = \sum_{j:ji \in E} x_{ji} - \sum_{s:si \in P} a_{si},$$

$\forall i \in N$, что идентично условиям (3.6).

Таким образом, задача (3.5)-(3.7) имеет решение тогда и только тогда, когда имеет решение построенная задача (3.12)-(3.15). Приведенное сведение можно выполнить за полиномиальное время $O(n^2)$, преобразовав любую сеть с заданными требованиями $A = \| a_{st} \|_{n \times n}$ в полную с дугами, для которых $c_{ij}^{st} = 1$ и $w_{ij} = \left\lceil \frac{a_{ij}}{\omega} \right\rceil \omega$ для всех индексов st и ij. Пусть имеется любое (не обязательно оптимальное) решение x_{ij}^{st}, $st \in P$, $ij \in E$ задачи (3.12)-(3.15). Набор значений x_{ij}^{st} однозначно определяет $x_{ij} = \sum_{st \in P} a_{st} x_{ij}^{st}$ $\forall ij \in E$ — значения переменных в (3.5)-(3.7). Поэтому очевидно, что решения задач (3.12)-(3.15) и (3.5)-(3.7) совпадают. Обратно. Пусть получено какое- либо решение x_{ij}, $i, j = \overline{1,n}$ задачи (3.5)-(3.7). По сохраненным цепочкам объединения элементов всегда можно

определить значения x_{ij}^{st}, отличные от нуля. Таким образом, по решению одной задачи можно получить аналогичное решение другой задачи и наоборот. Очевидно, что задача (3.5)-(3.7) при формулировке ее в виде задачи MC IMCF на полной сети полиномиально сводится к задаче (3.12)-(3.15). В силу NP-полноты (3.12)-(3.15) задача (3.5)-(3.7) также является NP-полной. Теорема доказана. ■

Сравнивая различные постановки задачи для полной сети с числом продуктов $p = n^2 - n$ и с числом дуг $e = n^2 - n$ по количеству ограничений и переменных (табл. 1), можно утверждать, что для разработки эвристических комбинаторных алгоритмов решения задачи лучшей является исходная формулировка, так как она содержит значительно меньше ограничений и переменных. Постановка (3.5)-(3.7) дает возможность конструировать эффективные (полиномиальные) алгоритмы локальной оптимизации (алгоритмы метода вектора спада, рандомизированные, метаалгоритмы, гибридные и пр.) без булевых переменных (т.е. не использовать в качестве базы только известные методы решения целочисленной задачи о многопродуктовом потоке минимальной стоимости [174, 175]).

Таблица 3.1

Постановка задачи	Количество	
	ограничений	переменных
Исходная (3.5)-(3.7)	$n + (n^2 - n)$	$(n^2 - n)$
В виде MC IMCF узлы-дуги (3.12)-(3.15)	$(n+1)(n^2 - n)$	$(n^2 - n)^2$

3.2.3. Численный эксперимент

Покажем на примере простых тестовых алгоритмов упаковок (Unconditional packing и Conditional packing), как влияют изменение границ входных данных и различные переборные схемы решения задачи на конечные результаты. Эти алгоритмы не предназначены для оптимального решения задачи (3.5)-(3.7) и тем более для решения практических задач упаковок с заданными функциями затрат на обработку и транспортировку мелкопартионных корреспонденций, в основе которых лежит решение задачи (3.5)-(3.7). Результаты работы этих алго-

ритмов только указывают, на что нужно обратить внимание при дальнейшей разработке рабочих алгоритмов для решения задач упаковок в реальных коммуникационных сетях.

Введем некоторую фиксированную операцию объединения элементов при решении задачи (3.5)-(3.7), которая формально записывается так: над любой тройкой элементов $x_{ik}, x_{kj}, x_{ij} \neq 0,\ k,i,j = \overline{1,n},\ k \neq i \neq j$ матрицы X могут выполняться действия

$$x_{ik} \leftarrow x_{ik} + x_{ij},\ x_{kj} \leftarrow x_{kj} + x_{ij},\ c_{ij} \leftarrow k,\ y_k \leftarrow y_k + x_{ij},\ x_{ij} \leftarrow 0, \qquad (3.16)$$

где «$\leftarrow$» — знак операции присваивания; k — индекс столбца и строки, через который выполняется объединение x_{ij}, причем x_{ij} должен объединяться с x_{ik} и x_{kj} только целиком; c_{ij} — элементы справочной матрицы объединений элементов $C = \| c_{ij} \|_{n\times n}$; $Y = \| y_k \|,\ k = \overline{1,n},$ — вектор сумм транзитных объединенных элементов, $y_k = \sum_{j=1}^{n} x_{kj} - \sum_{j=1}^{n} a_{kj}$. В начальном состоянии для всех i элементы $c_{ij} = i\ \forall j = \overline{1,n}$, а $Y = 0$. Из матрицы C на любом шаге можно определить с помощью алгоритмов, приведенных в подразделе 3.1 последовательность $\Omega_{ij} = \{(i,k_1),(k_1,k_2),...,(k_m, j)\}$ с промежуточными индексами столбцов и строк $\{k_1,k_2,...,k_m\}$, через которые выполнялось объединение элемента a_{ij}, а также найти множество других элементов $\{a^*_{rs}\}$, которые уже были объединены с a_{ij} на предыдущих итерациях, если $c_{ij} = i$.

Лемма 3.3. Если

$$\Delta = \left\lceil \frac{x_{ik}}{\omega} \right\rceil + \left\lceil \frac{x_{kj}}{\omega} \right\rceil + \left\lceil \frac{x_{ij}}{\omega} \right\rceil - \left(\left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil + \left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil \right), \qquad (3.17)$$

то ее значение не может превышать единицы.

Доказательство. В (3.17) возможны три случая: $\Delta < 0,\ \Delta = 0$ и $\Delta = 1$. Последний следует из того, что при условиях $\left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil = \left\lceil \frac{x_{ik}}{\omega} \right\rceil$ и $\left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil = \left\lceil \frac{x_{kj}}{\omega} \right\rceil$

разность в (3.17) не может превышать 1, так как при $x_{ij} \ge \omega$ и $\left\lceil \frac{x_{ij}}{\omega} \right\rceil \ge 1$ она всегда будет отрицательной. ■

Таким образом, за одну операцию объединения (3.16) при $\Delta = 1$ можно уменьшить значение функции (3.5) на единицу.

Рассмотрим алгоритмы Unconditional packing и Conditional packing и результаты их работы при изменении границ значений $\min \le (x_{ij} = a_{ij}) \le \max$ при фиксированном ω. В алгоритмах используются очевидное свойство $\sum_i \lceil x_i / \omega \rceil \ge \left\lceil (\sum_i x_i) / \omega \right\rceil$ и то, что любую матрицу X с помощью преобразований (3.16) можно для любых строки i и столбца j, $i = j$, привести к виду (в примере $i = j = 1$):

$$\begin{Vmatrix} 0 & \sum_i x_{i2} & \dots & \sum_i x_{in} \\ \sum_j x_{2j} & 0 & 0 & 0 \\ \dots & 0 & 0 & 0 \\ \sum_j x_{nj} & 0 & 0 & 0 \end{Vmatrix}.$$

Алгоритм Unconditional packing имеет сложность $O(n^3)$, не проверяет выполнение ограничений (3.7) и значения Δ в (3.17). Алгоритм Conditional packing проверяет значение Δ и при $\Delta \ge 0$ допускает ослабление ограничений (3.7). Сложность второго алгоритма больше и составляет $O(fs \cdot n^3)$, где fs — число входов в цикл по k.

Алгоритм 3.3. Unconditional packing

1. $OPT_X \leftarrow A$; $OPT_CS \leftarrow C$; $opt_cont_sum \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{n} \lceil a_{ij} / \omega \rceil$; $OPT_Y \leftarrow 0$; $opt_y_sum \leftarrow 0$.

2. Для $\{ k \mid k = \overline{1,n} \}$ выполнить пп. 3-11.

3. $X \leftarrow A$; $CS \leftarrow C$; $Y \leftarrow 0$.

4. Для $\{ i \mid i = \overline{1,n},\ i \ne k \}$ выполнить пп. 5-8.

5. Для $\{ j \mid j = \overline{1,n},\ j \neq i \wedge j \neq k \}$ выполнить пп. 6-7.

6. $x_{ik} \leftarrow x_{ik} + x_{ij};\ x_{kj} \leftarrow x_{kj} + x_{ij};\ cs_{ij} \leftarrow k;\ y_k \leftarrow y_k + x_{ij};\ x_{ij} \leftarrow 0.$

7. Перейти к п. 5. *** Конец цикла по j

8. Перейти к п. 4. *** Конец цикла по i

9. $cont_sum \leftarrow \sum_{r=1}^{n} \sum_{s=1}^{n} \lceil x_{rs} / \omega \rceil,\ y_sum \leftarrow \sum_{r=1}^{n} y_r$.

10. Если

$cont_sum < opt_cont_sum \vee cont_sum = opt_cont_sum \wedge y_sum < opt_y_sum$

, то $OPT_X \leftarrow X;\ OPT_CS \leftarrow CS;\ OPT_Y \leftarrow Y;\ opt_cont_sum = cont_sum;$

$opt_y_sum \leftarrow y_sum.$

11. Перейти к п. 2. *** Конец цикла по k

12. Конец алгоритма.

Алгоритм 3.4. Conditional packing

1. $OPT_X \leftarrow A;\ OPT_CS \leftarrow C;\ opt_cont_sum \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{n} \lceil a_{ij} / \omega \rceil;$

$OPT_Y \leftarrow 0;\ fs \leftarrow 0;\ fs_cont_sum \leftarrow opt_cont_sum;\ FS_Y \leftarrow 0;$

$fs_y_sum \leftarrow 0.$

2. Для $\{ k \mid k = \overline{1,n} \}$ выполнить пп. 3-12.

3. $X \leftarrow A;\ CS \leftarrow C;\ Y \leftarrow FS_Y;\ opt_y_sum \leftarrow fs_y_sum.$

4. Для $\{ i \mid i = \overline{1,n},\ i \neq k \}$ выполнить пп. 5-9.

5. Для $\{ j \mid j = \overline{1,n},\ j \neq i \wedge j \neq k \wedge x_{ik} \neq 0 \wedge x_{kj} \neq 0 \wedge x_{ij} \neq 0 \}$ выполнить пп. 6-8.

6. $\Delta \leftarrow \left\lceil \frac{x_{ik}}{\omega} \right\rceil + \left\lceil \frac{x_{kj}}{\omega} \right\rceil + \left\lceil \frac{x_{ij}}{\omega} \right\rceil - \left(\left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil + \left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil \right).$

7. Если $\Delta > 0$, то $x_{ik} \leftarrow x_{ik} + x_{ij};\ x_{kj} \leftarrow x_{kj} + x_{ij};\ cs_{ij} \leftarrow k;\ y_k \leftarrow y_k + x_{ij};$

$x_{ij} \leftarrow 0.$

*** Возможно и $\Delta \geq 0$

8. Перейти к п. 5. *** Конец цикла по j

9. Перейти к п. 4. *** Конец цикла по i

10. $cont_sum \leftarrow \sum_{r=1}^{n}\sum_{s=1}^{n}\lceil x_{rs}/\omega\rceil$, $y_sum \leftarrow \sum_{r=1}^{n} y_r$.

11. Если

$cont_sum < opt_cont_sum \vee cont_sum = opt_cont_sum \wedge y_sum < opt_y_sum$

, то $OPT_X \leftarrow X$; $OPT_CS \leftarrow CS$; $OPT_Y \leftarrow Y$; $opt_cont_sum = cont_sum$;

$opt_y_sum \leftarrow y_sum$.

12. Перейти к п. 2. *** Конец цикла по k

13. Если $opt_cont_sum < fs_cont_sum$, то $fs \leftarrow fs+1$;

$fs_cont_sum \leftarrow opt_cont_sum$; $A \leftarrow OPT_X$; $C \leftarrow OPT_CS$;

$FS_Y \leftarrow OPT_Y$; $fs_y_sum \leftarrow opt_y_sum$; перейти к п. 2.

14. Конец алгоритма.

В записи алгоритмов прописными буквами обозначены матрицы и векторы (например, OPT_X, OPT_Y), а строчными (например, opt_cont_sum, opt_y_sum) — простые переменные, знаком *** отмечены комментарии. В алгоритме Unconditional packing Δ из (3.17) не вычисляется, в этом случае ограничения (3.7) не учитываются. В алгоритме Conditional packing в п. 7 допускается $\Delta > 0$ и $\Delta \geq 0$, что соответствует строгому соблюдению ограничений (3.7), и их ослаблению.

При проведении эксперимента матрица A генерировалась датчиком псевдослучайных чисел для восьми интервалов. Результаты эксперимента для n = 100 и ω = 40 приведены в табл. 3.2, в которой приняты следующие обозначения: opt_cont_sum — значение функции цели (3.5); opt_y_sum — минимальное значение $y = \sum_{k=1}^{n} y_k$; K_{av} — средний коэффициент загрузки блока; N_{av} — среднее число ненулевых элементов в строке матрицы X; fs — число входов в цикл по k в алгоритме Conditional packing; t_{sol} — время решения задачи. Для каждого интервала данной таблицы результаты соответствуют работе алгоритмов: до оптимизации — первая строка; Unconditional packing — вторая строка;

Conditional packing при $\Delta \ge 0$ и $\Delta > 0$ — третья и четвертая строки.

Значения $K_{av} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\delta_i}\sum_{j=1}^{n}x_{ij} \Big/ \left(\left\lceil \frac{x_{ij}}{\omega} \right\rceil \omega \right)$, $\delta_i = \sum_{j=1}^{n}\delta_{ij}$, и $N_{av} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$, $i = \overline{1,n}$

Таблица 3.2

Интервалы значений a_{ij}, $i, j = \overline{1,n}$	**Результаты работы алгоритмов Unconditional packing и Conditional packing**					
	opt_cont_sum	opt_y_sum	K_{av}	N_{av}	fs	t_{sol}, с
[1, 5]	9900	0	0,075	99	-	-
	1569	29057	0,936	1	1	0,17
	1570	29126	0,937	1	3	0,97
	2391	22477	0,519	23	30	6,46
[1, 10]	9900	0	0,138	99	-	-
	2793	53253	0,963	1	1	0,16
	2790	53278	0,942	2	3	0,97
	3552	32890	0,597	35	34	8,75
[1, 20]	9900	0	0,262	99	-	-
	5238	101619	0,982	1	1	0,17
	5219	100735	0,866	2	7	1,54
	4510	47325	0,819	45	59	17,02
[1, 40]	9900	0	0,512	99	-	-
	9900	0	0,512	99	1	0,16
	8039	92435	0,833	35	31	7,13
	6734	28156	0,854	67	75	29,36
[1, 60]	13232	0	0,550	99	-	-
	13232	0	0,550	99	1	0,17
	12069	149615	0,887	37	45	10,94
	9931	43588	0,868	65	98	37,47
[1, 80]	14869	0	0,632	99	-	-
	14869	0	0,632	99	1	0,16
	14057	119340	0,889	51	36	10,81
	12334	32342	0,872	73	93	39,91
[1, 100]	17827	0	0,657	99	-	-
	17827	0	0,657	99	1	0,16
	17049	122630	0,893	54	40	12,93
	15344	37448	0,873	74	93	40,34
[1, 120]	19791	0	0,701	99	-	-
	19791	0	0,701	99	1	0,16
	19160	102874	0,900	61	39	14,04
	17726	30478	0,881	78	92	42,46

вычислялись по результирующей матрице OPT_X, где $\delta_{ij} = 1$, если $x_{ij} \ne 0$, и $\delta_{ij} = 0$ в противном случае.

В табл. 3.3 приведены результаты работы алгоритма Conditional packing при $\Delta \ge 0$, $\Delta > 0$ и перестановке индексов основного тройного цикла (пп. 2, 4, 5 алгоритма) при n = 200, ω = 100, $a_{ij} \in [1,120]$, $i, j = \overline{1,n}$. В первой строке даны результаты работы алгоритма Conditional packing до оптимизации, во второй и

третьей строках — соответственно при $\Delta \geq 0$ и $\Delta > 0$. Из табл. 3.2 видно, что алгоритм Unconditional packing может при малых границах изменения a_{ij} дать лучшие решения, чем алгоритм Conditional packing. При больших значениях ω с помощью этого алгоритма можно получить и оптимальное решение $opt_cont_sum = 2n - 2 = 198$ с минимальным значением opt_y_sum. При увеличении границ изменения a_{ij} лучшие результаты показывает алгоритм Conditional packing при $\Delta \geq 0$ и $\Delta > 0$. Из табл. 3.3 можно видеть, что различные переборные схемы объединения элементов в алгоритме Conditional packing дают разные сильно отличающиеся результаты.

Таблица 3.3

Последовательность индексов тройного цикла	**Результаты работы алгоритма Conditional packing**					
	opt_cont_sum	opt_y_sum	K_{av}	N_{av}	fs	t_{sol}, с
$k,\ i,\ j$	46481 40729 32220	0 1438516 434132	0,514 0,856 0,877	199 60 127	- 65 196	- 106,63 573,91
$k,\ j,\ i$	46481 40729 32220	0 1438516 434132	0,514 0,856 0,877	199 60 127	- 65 196	- 105,89 571,85
$i,\ k,\ j$	46481 41463 32779	0 1537721 501658	0,514 0,934 0,890	199 64 130	- 192 200	- 339,69 591,79
$j,\ k,\ i$	46481 41431 32798	0 1532645 497041	0,514 0,921 0,880	199 64 130	- 191 199	- 338,27 584,35
$i,\ j,\ k$	46481 42599 34729	0 1595685 670593	0,514 0,941 0,886	199 80 140	- 169 200	- 332,05 644,47
$j,\ i,\ k$	46481 42645 34692	0 1601494 661718	0,514 0,919 0,881	199 80 140	- 175 200	- 362,27 700,57

Анализ результатов, приведенных в табл. 3.2 и 3.3, позволяет сделать следующие выводы:

— при малых значениях a_{ij}, $i, j = \overline{1,n}$ (интервалы [1, 5], [1, 10]) лучшие значения функции цели можно получить с помощью алгоритма Unconditional packing, когда Δ не вычисляется и ограничения (3.7) не учитываются, а начиная с интервала [1, 40] этот алгоритм неприменим, так как не приводит к улучшению значения целевой функции;

— для интервалов [1, 5], [1, 10] неплохие результаты показывает алгоритм

Conditional packing при $\Delta \geq 0$, в этом случае сумма значений транзитных объединенных элементов (opt_y_sum) всегда намного больше, чем в случае $\Delta > 0$, что необходимо учитывать при разработке рабочих алгоритмов для реальных задач упаковки со сложной целевой функцией;

— безусловная и условная при $\Delta \geq 0$ стратегии могут расширять границы поиска локального минимума при случайном их использовании в алгоритме Conditional packing и привести к лучшему результату по сравнению с непрерывным применением стратегии $\Delta > 0$;

— средний коэффициент загрузки блока K_{av} не может являться критерием поиска лучшего решения;

— данные эксперимента подтверждают сильную зависимость результатов решения задачи от выбора элементов x_{ij} и индексов k для выполнения операций объединения (3.16) и от границ изменения входных данных — значений n, a_{ij}, ω, и указывают на необходимость поиска и разработки эффективных эвристических алгоритмов.

Все задачи решались на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.

3.3. Задача упаковки с функциями затрат и алгоритмы ее решения

В магистральных транспортных сетях и опорных сетях передачи данных с мелкопартионными потоками корреспонденций от отправителей к получателям возникает задача упаковки, сущность которой заключается в концентрации потоков грузов и информации между узлами сети [119]. Исходящие из узлов сети мелкопартионные корреспонденции с разными адресами назначения могут объединяться одна с другой и упаковываться в общие транспортные блоки (контейнеры или виртуальные контейнеры). В результате объединения потоков уменьшается количество транспортных блоков для транспортировки корреспонденций, сокращается число направлений сортировки корреспонденций в узлах сети в другие узлы, увеличивается коэффициент загрузки транспортных блоков

и транспортных средств, более продуктивно используется высокая пропускная способность магистральных каналов связи. В то же время в отдельных узлах сети возникают транзитные потоки мелкопартионных корреспонденций, что приводит к увеличению общей стоимости обработки потоков в сети и увеличению времени доставки корреспонденций получателю.

В настоящем подразделе предлагаются алгоритмы для решения практических задач упаковки с функциями затрат на обработку и транспортировку мелкопартионных корреспонденций, которые основаны на дискретном аналоге метода локального спуска, когда окрестности метрического пространства допустимых решений выбираются из эвристических (интуитивных) соображений с учетом специфики структур данных и особенностей решаемой задачи. Обсуждаются вопросы сходимости и временной сложности алгоритмов, а также на численных примерах исследуется их сравнительная вычислительная эффективность.

3.3.1. Постановка задачи и особенности ее решения

Пусть $G(N,E)$ — неориентированная магистральная сеть с множеством узлов N, $n=\mid N \mid$, и множеством дуг E, $e=\mid E \mid$, $\mid\cdot\mid$ — знак мощности множества. Задано множество корреспонденций, каждая из которых характеризуется узлом отправления i, узлом назначения j и объемом корреспонденции $a_{ij} \in \mathrm{Z}^{+}$, $i,j=\overline{1,n}$. Все корреспонденции должны транспортироваться по сети в некоторых транспортных блоках фиксированной емкости $\omega \in \mathrm{Z}^{+}$. Величины a_{ii}, $i=\overline{1,n}$ определяют внутриузловые объемы на обработку корреспонденций, которые в транспортные блоки не упаковываются и по сети не транспортируются. Все корреспонденции однородны (одного типа), при транспортировке могут объединяться в различных узлах и упаковываться в транспортные блоки только целиком, т.е. запрещается их разветвление (расщепление, дробление на части) и транспортировка по нескольким путям. Пусть объемы корреспонденций заданы исходной матрицей требований $A=\| a_{ij} \|_{n\times n}$, $a_{ij} \in \{1,\ldots,\omega,\ldots,a_{\max}\}$, $i,j=\overline{1,n}$, в ко-

торой строки соответствуют узлам отправления, а столбцы — узлам назначения, и имеется некоторая преобразованная матрица $X = \| x_{ij} \|_{n \times n}$, элементы которой являются целочисленными искомыми переменными задачи упаковки.

Требуется найти минимум функции

$$\sum_{ij \in S} C_{tr}^{ij}(u_{ij}, d_{ij}) + \sum_{i=1}^{n} C_{sort}^{i}(x_i, q_i) + \sum_{i=1}^{n} C_{load}^{i}(u_i), \tag{3.18}$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1, j \neq i}^{n} a_{ij} = \sum_{j=1}^{n} x_{ji} - \sum_{j=1, j \neq i}^{n} a_{ji}, \; i = \overline{1,n}, \tag{3.19}$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1, j \neq i}^{n} a_{ij} \leq h_i, \; i = \overline{1,n}, \tag{3.20}$$

$$x_{ij}, u_{ij} \geq 0 \text{ и целые}, \; \forall ij \in S, \tag{3.21}$$

где S — множество пар индексов ij корреспонденций, определенное на декартовом произведении $n \times n$; $x_{ij} = a_{ij} + \sum_{rs} a_{rs}^{*}$, если корреспонденция a_{ij} не объединялась ни с какой другой корреспонденцией ($\{a_{rs}^{*}\}$ — множество корреспонденций, объединенных с корреспонденцией a_{ij}), и $x_{ij} = 0$, если корреспонденция a_{ij} объединялась с какой-либо другой корреспонденцией или $i = j$; $C_{tr}^{ij}(u_{ij}, d_{ij})$ — нелинейная функция транспортных затрат, зависящая от количества транспортных блоков $u_{ij} = \left\lceil \dfrac{x_{ij}}{\omega} \right\rceil$, ($\lceil x \rceil$ — наименьшее целое, большее или равное x) и длины d_{ij} — пути их транспортировки между узлами i и j; $C_{sort}^{i}(x_i, q_i)$ — нелинейная функция затрат от суммарного объема $x_i = a_{ii} + \sum_{j=1, j \neq i}^{n} (a_{ij} + a_{ji}) + \sum_{j=1}^{n} x_{ij} - \sum_{j=1, j \neq i}^{n} a_{ij}$ и количества направлений сортировки $q_i = q_{in}^{i} + \sum_{j=1}^{n} \delta_{ij}$ мелкопартионных корреспонденций, обрабатываемых в узле i ($\delta_{ij} = 1$, если $x_{ij} \neq 0$, и $\delta_{ij} = 0$, если $x_{ij} = 0$, а q_{in}^{i} определяет заданное количество направлений сортировки для обработки корреспонденций a_{ii}, $i = \overline{1,n}$); $C_{load}^{i}(u_i)$ — нелинейная функция затрат от суммарного количества транспортных блоков

$u_i = \sum_{j=1}^{n}(u_{ij} + u_{ji})$, обрабатываемых в узле i; h_i, $i = \overline{1,n}$, — максимальная пропускная способность i-го узла по обработке транзитных мелкопартионных корреспонденций.

При решении задачи также учитываются ограничения на время доставки $t_{ij} \le T_{ij} \;\; \forall ij \in S$ и число транзитных объединений $\nu_{ij} \le \nu_{\max} \;\; \forall ij \in S$ мелкопартионных корреспонденций при их транспортировке из узлов отправления в узлы назначения, где T_{ij} и $\nu_{\max}$ — соответственно заданное время доставки мелкопартионных корреспонденций получателю и максимально допустимое число транзитных объединений корреспонденций.

Первая составляющая функции (3.18) определяет транспортные затраты, вторая — затраты на сортировку, а третья — затраты на обработку транспортных блоков. Выражения (3.19), (3.20) и (3.21) представляют условия баланса, ограничения на пропускные способности узлов и значения переменных x_{ij}.

Рассмотрим особенности сформулированной задачи. В транспортных сетях мелкопартионные грузы, как правило, перевозятся в контейнерах, на поддонах, в пленочной или другой упаковке и т.п. Это связано с рядом факторов, наиболее важными из которых являются сохранность грузов, возможность их автоматизированной и механизированной обработки, хранение на открытых площадках, упрощение обработки сопроводительной документации и пр. Перед транспортировкой мелкопартионные грузы в узлах транспортной сети рассортировываются по адресам их доставки в сортировочных центрах, оснащенных высокопроизводительными автоматизированными сортировочными системами. Технологический процесс сортировки заключается в следующем. Грузы из общего накопителя или накопителей поступают на линии сортировки, в начале которых установлены считыватели штрих-кодов или другие более современные устройства распознавания адреса груза и сопроводительной информации о нем. В соответствии с расшифрованной информацией груз направляется в нужный адресный накопитель. Количество адресных накопителей соответствует количеству направлений сортировки грузов. В сетях передачи данных также выполняются

подобные технологические операции сортировки информационных потоков. Современные опорные сети передачи данных с волоконно-оптическими линиями связи и технологиями спектрального уплотнения каналов WDM, DWDM, HDWDM (wavelength-division multiplexing, dense WDM, high dense WDM) позволяют мультиплексировать от 16 до 64 и более спектральных каналов в одной линии связи и передавать объемы информации до 27 Тбит/с (1Тбит = 10^{12} бит) и более. В таких сетях в роли «сортировочной машины» выступают мультиплексоры, которые объединяют отдельные информационные потоки в виртуальные контейнеры.

При практическом проектировании и анализе коммуникационных сетей должны использоваться реальные стоимостные показатели, например среднегодовые приведенные затраты на обработку и транспортировку потоков. На практике существует множество различных функций, используемых для расчета затрат на функционирование сети. По мере организационно-технического совершенствования сети эти функции постоянно подвергаются изменениям и уточнениям. В каждом случае определение адекватных функций затрат является сложной задачей, которая должна быть решена отдельно перед проведением численного моделирования. Очевидно, что при решении задач оптимизации в целевую функцию должны быть включены только наиболее важные составляющие затрат, зависящие от искомых переменных. В сформулированной задаче предполагается использование во всех составляющих целевой функции капитальных и эксплуатационных затрат, приведенных к сопоставимому виду.

Как правило, в математических моделях, описывающих процессы обработки и транспортировки потоков, затраты связывают с величиной потока по дугам сети или путям передачи потока. Для сетей передачи данных, где дуги ассоциируются с каналами связи, такие постановки оказываются достаточно приемлемыми. В случае транспортных сетей значительно сложнее адекватно определить стоимостные функции, например $C_{tr}^{ij}(u_{ij}, d_{ij})$, а значит, и получить в результате решения задачи достоверный ответ. Представление потоковых задач в виде задач линейного программирования со стоимостными коэффициентами

вообще может оказаться абсурдным с практической точки зрения. По убеждению автора, в реальных задачах нужно рассчитывать транспортные затраты по маршрутам транспортных средств, т.е. связывать объемы и пути распределения потоков с множеством искомых «оптимальных» маршрутов [11, 120, 467]. Для каждого определенного в результате решения маршрута, зная его характеристики (длину, грузоподъемность и тип транспортного средства и пр.), легко рассчитать среднегодовые приведенные затраты для каждого маршрута и получить более достоверную оценку транспортных затрат для всей сети перевозок. «Маршрутный» подход приемлем и для сетей передачи данных, так как каждый маршрут — это либо ориентированная дуга, либо последовательность таких дуг (скоммутированные каналы). В задаче (3.18)-(3.21) не вводятся «маршрутные» функции затрат, поскольку она является одной из первых подзадач обобщенной задачи обработки, распределения и маршрутизации мелкопартионных корреспонденций [14]. При ее решении рассчитывается только ориентировочная оценка транспортных затрат и нижняя граница затрат на обработку транспортных блоков. Поэтому $C_{tr}^{ij}(u_{ij}, d_{ij})$ может быть принята как функция удельной стоимости транспортировки потока величиной u_{ij} на расстояние d_{ij} от грузоподъемности транспортного средства или пропускной способности канала связи, задаваемой в виде параметра $w_\xi \in \{w_1, w_2, ..., w_\alpha\}$, $\xi = \overline{1, \alpha}$. Например, можно считать, что $C_{tr}^{ij}(u_{ij}, d_{ij}) = u_{ij}(k_1^\xi + k_2^\xi d_{ij}) / w_\xi \quad \forall ij \in S$, где k_1^ξ, k_2^ξ — заданные коэффициенты. В последующих подразделах будет показана допустимость такой замены. Учитывая, что функции C_{sort}^i и C_{load}^i должны отражать реальные затраты, при моделировании задачи необходимо так подобрать значения k_1^ξ, k_2^ξ, w_ξ, чтобы транспортные затраты были соизмеримы с затратами на сортировку мелкопартионных корреспонденций и обработку транспортных блоков в узлах сети, т.е. чтобы порядок величины транспортных затрат был близок к фактическим. Реальные транспортные затраты и затраты на обработку транспортных блоков рассчитываются при решении задачи распределения и маршрутизации потоков (см. раздел 4).

Еще одной особенностью рассматриваемой модели являются ограничения $t_{ij} \le T_{ij}$ на время доставки мелкопартионных корреспонденций конечным получателям и ограничения $\nu_{ij} \le \nu_{\max}$ на максимально допустимое число объединений корреспонденций с другими корреспонденциями. При решении задачи (3.18)-(3.21) рассчитываются только предварительные значения t_{ij}, так как до решения задачи распределения и маршрутизации потоков транспортных блоков неизвестны фактические пути следования мелкопартионных корреспонденций от отправителей к получателям.

3.3.2. Алгоритмы решения задачи

В общем случае целевая функция (3.18) будет невыпуклой и многоэкстремальной. Целочисленность переменных задачи и наличие ограничений $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max} \quad \forall ij \in S$ приводит к необходимости поиска и разработки комбинаторных приближенных методов, использующих специфику структуры задачи, эвристические приемы и интерактивный режим поиска решения. Из постановки задачи (3.18)-(3.21) следует, что можно выписать начальное допустимое решение: $X = A$, где a_{ii}, $i = \overline{1,n}$ заменены нулями. Пусть в процессе решения задачи над элементами $x_{ik}, x_{kj}, x_{ij} \ne 0$, $k,i,j = \overline{1,n}$, $k \ne i \ne j$, матрицы X могут итеративно выполняться следующие фиксированные операции:

$$x_{ik} \leftarrow x_{ik} + x_{ij},\ x_{kj} \leftarrow x_{kj} + x_{ij},\ c_{ij} \leftarrow k,\ y_k \leftarrow y_k + x_{ij},\ x_{ij} \leftarrow 0, \qquad (3.22)$$

где символ «$\leftarrow$» обозначает операцию присваивания; k — номер узла, в котором выполняется объединение корреспонденции x_{ij}; c_{ij} — элементы справочной матрицы объединения корреспонденций $C = \| c_{ij} \|_{n \times n}$; y_k — элементы вектора сумм транзитных корреспонденций $Y = \| y_k \|$, $k = \overline{1,n}$. Элементы матрицы C определяются так:

$$c_{ij} = \begin{cases} k, \text{если корреспонденция } x_{ij} \text{ объединяется с корреспонденцией } x_{ik}, \\ i, \text{если } x_{ij} \text{ не объединяется ни с какой другой корреспонденцией}, \\ 0, \text{если } i = j. \end{cases} \quad (3.23)$$

С помощью матрицы C и алгоритмов, приведенных в подразделе 3.1, можно определить последовательность $\Omega_{ij} = \{(i,k_1),(k_1,k_2),...,(k_m,j)\}$ с промежуточными узлами $\{k_1,k_2,...,k_m\}$ ($\nu_{ij} = |\{k_1,k_2,...,k_m\}|$), в которых выполнялось объединение каждой корреспонденции a_{ij}, а также найти множество других корреспонденций $\{a^*_{rs}\}$, которые уже были объединены с a_{ij} на предыдущих итерациях.

Предположим, что известна справочная матрица путей транспортировки потоков транспортных блоков $C^* = \| c^*_{ij} \|_{n\times n}$, полученная при построении кратчайших путей в сети G по ступенчатому критерию — минимум транзитных узлов на пути следования потока, минимум длины пути (см. раздел 6). Элементы матрица C^* определяются так:

$$c^*_{ij} = \begin{cases} 0, \text{если } i = j, \\ k, \text{если } u_{ij} \text{ следует в узел } j \text{ с перегрузкой в узле } k, \\ i, \text{если } u_{ij} \text{ следует в узел } j \text{ без перегрузки.} \end{cases}$$

Отметим, что кратчайшие пути, построенные от узла i, представляют собой дерево с корнем в i, которое может быть однозначно описано строкой справочной матрицы C^*, а k является предпоследним узлом на кратчайшем пути от i до j. Поэтому с помощью этой матрицы легко определить число транзитных узлов на пути следования любого потока. Известна также матрица длин построенных путей $D = \| d_{ij} \|_{n\times n}$. Матрицы C^* и D используются при расчете значений функции $C^{ij}_{tr}(u_{ij},d_{ij})$. Пусть t_a — заданное время на сортировку мелкопартионных корреспонденций в узле, t_b — среднее время на транзитную перегрузку или перекоммутацию транспортных блоков, V_{av} — средняя скорость движения транспортных средств (км/час) или скорость распространения сигна-

ла по каналам связи (км/с). Требуется рассчитать значения t_{ij}. Для предварительной оценки времени доставки достаточно использовать справочную матрицу объединения корреспонденций $C = \| c_{ij} \|_{n \times n}$ и матрицы C^* и D. Приведем формулу для расчета t_{ij} $\forall ij \in S$, $i \neq j$:

$$t_{ij} = \begin{cases} 2t_a + d_{ij}/(V_{av}\theta) + \psi_{ij} t_b, \text{ если } c_{ij} = i, \\ t_a(\nu_{ij} + 2) + \sum\limits_{\xi\eta \in \Omega_{ij}} (d_{\xi\eta}/(V_{av}\theta) + \psi_{\xi\eta} t_b), \text{ если } c_{ij} \neq i, \end{cases} \quad (3.24)$$

где $\psi_{\xi\eta}$ — число транзитных перегрузок u_{ij} на участке (ξ, η); θ — коэффициент нормирования времени (для транспортных сетей θ = 24 часа, для сетей передачи данных θ = 1).

В предыдущем подразделе приведены результаты экспериментального исследования двух простых алгоритмов упаковки при изменении границ для начальных значений $MIN \le (x_{ij} = a_{ij}) \le MAX$ при фиксированном ω. Опять рассмотрим три возможных случая при выполнении преобразования (3.22) для значения $\Delta^u_{ij,k} = u_{ik} + u_{kj} + u_{ij} - (u'_{ik} + u'_{kj})$ при $\Delta^u_{ij,k} < 0$, $\Delta^u_{ij,k} = 0$ и $\Delta^u_{ij,k} = 1$, где

$$u_{ik} = \left\lceil \frac{x_{ik}}{\omega} \right\rceil,\ u_{kj} = \left\lceil \frac{x_{kj}}{\omega} \right\rceil,\ u_{ij} = \left\lceil \frac{x_{ij}}{\omega} \right\rceil,\ u'_{ik} = \left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil,\ u'_{kj} = \left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil.$$

Было показано, что для большинства интервалов изменения значений a_{ij} лучшие решения могут быть получены при строгом соблюдении ограничений (3.21) и их ослаблении, т.е. при $\Delta^u_{ij,k} > 0$ и $\Delta^u_{ij,k} \ge 0$.

Для описания алгоритма введем следующие обозначения. Пусть

$$\Delta F^-_{ij}(k) = \sum_{ij \in \{ik, kj, ij\}} C^{ij}_{tr}(u_{ij}, d_{ij}) + \sum_{i \in \{i,k\}} C^i_{sort}(x_i, q_i) + \sum_{i \in \{i,k,j\}} C^i_{load}(u_i),\ u_i = \sum\nolimits_{l=1}^{n} (u_{il} + u_{lj});$$

$$\Delta F^+_{ij}(k) = \sum_{ij \in \{ik, kj\}} C^{ij}_{tr}(u'_{ij}, d_{ij}) + C^i_{sort}(x_i, q_i - 1) + C^k_{sort}(x_k + x_{ij}, q_k) + \sum_{i \in \{i,k,j\}} C^i_{load}(u'_i),$$

$u'_i = \sum_{l=1}^{n} (u'_{il} + u'_{li})$, $u'_{ij} = 0$; $k \in K$, K — множество узлов-претендентов, через которые может быть выполнено объединение корреспонденции x_{ij} без нарушения ограничений (3.20) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$. Для возможности «прорыва» из точек локального минимума в другие точки минимума во множество K можно

включать узлы, для которых $\Delta^{u}_{ij,k} \geq 0$.

Приведем общую схему алгоритма. Знаки «$\wedge$» и «$\vee$» соответственно означают операции конъюнкции и дизъюнкции. Знаком «***» отделяются комментарии.

Алгоритм 3.5. Optpac1

1. С помощью алгоритма из подраздела 6.2 для заданных длин дуг сети G построить для всех корреспонденций кратчайшие пути по ступенчатому критерию — минимум транзитных узлов в пути, минимум длины пути и получить матрицы C^{*} и D.

2. Рассчитать полные затраты ZP (значение функции (3.18)) до оптимизации. $DELF \leftarrow ZP$.

3. Выбрать значение $\nu_{\max}$; $\nu \leftarrow 1$; $ZMIN \leftarrow \infty$.

4. Ввести исходную матрицу A; $X \leftarrow A$; установить $c_{ij} \leftarrow i \quad \forall i \neq j$ и $x_{ii} \leftarrow 0$, $c_{ii} \leftarrow 0$, $i, j = \overline{1,n}$. *** c_{ij} — элементы справочной матрицы объединения корреспонденций (3.23)

5. Матрицы $\Lambda \leftarrow 0$, $B \leftarrow 1$; $b_{ii} \leftarrow 0$, $i = \overline{1,n}$; $C_1 \leftarrow 0$. *** $\Lambda = \| \Delta_{ij} \|_{n \times n}$, $B = \| b_{ij} \|_{n \times n}$ — рабочие матрицы, C_1 — счетчик входов в пп. 6-20.

6. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 7-11.

7. Для $\{ j \mid j = \overline{1,n} \}$ выполнить пп. 8-10.

8. Если $i \neq j \wedge b_{ij} = 1 \wedge x_{ij} \neq 0$, то $k \leftarrow i$, перейти к п. 9, иначе перейти к п. 10.

9. Найти множество K узлов-претендентов, с помощью которых может быть выполнено объединение корреспонденции x_{ij} без нарушения ограничений (3.20), $\Delta^{u}_{ij,k} \geq 0$ и ограничений $t_{ij} \leq T_{ij}$, $\nu_{ij} \leq \nu$, $t_{rs} \leq T_{rs}$, $\nu_{rs} \leq \nu$ для a_{ij} и множества корреспонденций $\{a^{*}_{rs}\}$ уже объединенных с a_{ij}; $\Delta_{ij} \leftarrow \max\limits_{k \in K} (\Delta F^{-}_{ij}(k) - \Delta F^{+}_{ij}(k))$, $c_{ij} \leftarrow \arg\max\limits_{k \in K} (\Delta F^{-}_{ij}(k) - \Delta F^{+}_{ij}(k))$.

10. Перейти к п. 7 *** Конец цикла по j

11. Перейти к п. 6 *** Конец цикла по i

12. Найти $\Delta_{ij}^{\max} = \max\limits_{ij \in S} \Delta_{ij}$. Если $\Delta_{ij}^{\max} > 0$, то $k \leftarrow c_{ij}$, перейти к п. 13, иначе перейти к п. 21.

13. Выполнить преобразование $x_{ik} \leftarrow x_{ik} + x_{ij}$, $x_{kj} \leftarrow x_{kj} + x_{ij}$, $y_k \leftarrow y_k + x_{ij}$, $x_{ij} \leftarrow 0$, $\Delta_{ij} \leftarrow 0$.

14. Матрица $B \leftarrow 0$. Для $\{ l \mid l = \overline{1,n} \}$ выполнить $b_{lk} \leftarrow 1$; $b_{kl} \leftarrow 1$; $b_{lj} \leftarrow 1$; $b_{jl} \leftarrow 1$; $b_{il} \leftarrow 1$; $b_{li} \leftarrow 1$.

15. Для $\{ l \mid l = \overline{1,n} \}$ выполнить пп. 16-19.

16. Для $\{ m \mid m = \overline{1,n} \}$ выполнить пп. 17-18.

17. Если $c_{lm} = i \vee c_{lm} = k \vee c_{lm} = j$, то $b_{lm} \leftarrow 1$, иначе перейти к п. 18.

18. Перейти к п. 16 *** Конец цикла по m

19. Перейти к п. 15 *** Конец цикла по l.

20. $C_1 \leftarrow C_1 + 1$. Перейти к п. 6.

21. По формуле (3.24) рассчитать время доставки $t_{ij} \ \forall ij \in S$. *** Временная сложность $O(3n^2)$

22. Рассчитать полные затраты ZP (значение функции (3.18)) после оптимизации. $DELF1 \leftarrow DELF$; $DELF \leftarrow DELF - ZP$. *** Временная сложность $O(n^2)$

23. Если $DELF > 0$, то перейти к п. 24, иначе к п. 29.

24. Если $ZP < ZMIN$, то перейти к п. 25, иначе к п. 26.

25. $ZMIN \leftarrow ZP$; $ZSTOP \leftarrow ZP$; $\nu_{opt} \leftarrow \nu$; $\nu \leftarrow \nu + 1$. Если $\nu > \nu_{\max}$, то перейти к п. 30, иначе перейти к п. 4.

26. Найден локальный минимум $ZMIN$ при значении ν_{opt}. Если $\nu = \nu_{opt}$, то перейти к п. 30, иначе перейти к п. 27.

27. Если $ZSTOP \neq ZP$, то перейти к п. 28, иначе к п. 30.

28. $ZSTOP \leftarrow ZP$; $\nu \leftarrow \nu + 1$. Если $\nu > \nu_{\max}$, то перейти к п. 30, иначе перейти к п. 4.

29. Если $\nu = 1$, то найден локальный минимум $DELF1$ при значении $\nu_{opt} = \nu - 1$, иначе найден локальный минимум $ZMIN$ при значении ν_{opt}.

30. Конец алгоритма.

В алгоритме п. 5 $\Lambda = \| \Delta_{ij} \|_{n \times n}$ — рабочая матрица для значений $\Delta_{ij} = \Delta F_{ij}^{-}(k) - \Delta F_{ij}^{+}(k)$, а $B = \| b_{ij} \|_{n \times n}$ используется для ограничения пересчета оценок в Λ, так как после преобразования x_{ij} оценки в Λ должны быть пересчитаны только для множества корреспонденций $\{x_{rs}\}$, для которых полученные оценки были связаны с преобразованной корреспонденцией x_{ij}. В п. 9 $\forall x_{ij}$, $ij \in S$ выбирается наилучшая оценка Δ_{ij} на множестве K, а в п. 12 определяется корреспонденция x_{ij}, подлежащая преобразованию. Если $\Delta_{ij} = 0$ $\forall$ $ij \in S$, то выполняется выход из внешнего цикла пп. 6-20 и оптимизация для текущего значения ν завершена. В противном случае выполняется преобразование x_{ij} при $\Delta_{ij}^{\max} = \max\limits_{ij \in S} \Delta_{ij}$ (п. 13), пересчитываются оценки в Λ (пп. 14-19), на единицу увеличивается счетчик C_1 и осуществляется переход на начало внешнего цикла 6-20. В пп. 21-29 определяются результаты оптимизации для текущего значения ν и проверяются условия останова алгоритма. В случае нахождения локального минимума выводится его значение. Алгоритм заканчивает работу, если $\nu > \nu_{\max}$ или $ZSTOP = ZP$, т.е. когда значение целевой функции перестает изменяться. Если эти условия не выполняются, алгоритм продолжает работу, начиная с п. 4, для нового значения $\nu \leftarrow \nu + 1$.

Приведенный алгоритм очень похож на алгоритм метода вектора спада И.В. Сергиенко [108]. Как отмечается в [108], для многих комбинаторных задач (коммивояжера, различных задач упаковок и пр.) очень трудно определить понятие окрестности для дискретного метрического пространства допустимых решений и процесс нахождения оптимального решения может свестись к пол-

ному перебору таких решений. Поэтому окрестности для таких задач зачастую вводятся из интуитивных эвристических соображений с учетом специфики структур данных и особенностей решаемых задач. Нетрудно видеть, что в нашем случае, радиус окрестности в алгоритме метода вектора спада — это нарастающие значения $\nu \leftarrow \nu + 1$, а действительные значения вектора спада — это значения Δ_{ij} матрицы Λ. Если $\Delta_{ij} = 0 \ \ \forall ij \in S$, то выполняется переход на увеличение $\nu \leftarrow \nu + 1$ до тех пор, пока $\nu \le \nu_{\max}$ или $ZSTOP \ne ZP$.

Пусть F_0 — значение функции (3.18) до оптимизации (т.е. значение (3.18) при $X = A$).

Теорема 3.4. Если множество M допустимых решений задачи (3.18)-(3.21), определяемых ограничениями (3.20), (3.21) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$, не пусто, то с помощью алгоритма Optpac1 в худшем случае за асимптотическое время $O(Cn^5)$ будет получено некоторое текущее решение $F_\nu < F_0$ или найден локальный минимум $F_{\min}$ функции (3.18).

Доказательство. Функция (3.18) ограничена сверху значением F_0, и при $M = \varnothing$ это значение является решением задачи. Функция (3.18) ограничена также и снизу, поскольку при итеративном выполнении преобразований (3.22) при условии $\Delta^u_{ij,k} > 0$ или $\Delta^u_{ij,k} \ge 0$ и даже полном ослаблении ограничений (3.20) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$ (т.е. при $h_i = \infty$, $i = \overline{1,n}$, $T_{ij} = \infty$, $ij \in S$, $\nu_{\max} = \infty$) будет достигнуто то состояние, когда дальнейшее выполнение преобразования (3.22) для элементов матрицы X не может привести к уменьшению количества транспортных блоков. Во вложенных циклах по i и j (пп. 6-11) для каждой корреспонденции x_{ij} определяется множество K (п. 9) и по мощности этого множества выбирается максимальная оценка $\Delta_{ij} = \max\limits_{k \in K}(\Delta F^-_{ij}(k) - \Delta F^+_{ij}(k))$. Трудоемкость определения K и выбора Δ_{ij} составляет $O(C_2 n)$, где C_2 — некоторая константа, максимальное значение которой зависит от $\left|\{a^*_{rs}\}\right|$ и числа рекурсий алгоритмов из подраздела 3.1 для нахождения $\{a^*_{rs}\}$ и проверки ограничений $t_{rs} \le T_{rs}$ и

$\nu_{rs} \le \nu$. Поэтому выполнение действий в пп. 6-11 потребует порядка $O(C_2 n^3)$ времени. Операции в пп. 12, 14-19 имеют сложность порядка $O(2n^2 + n)$, поэтому сложность выполнения пп. 6-19 составит $O(C_2 n^3 + 2n^2 + n)$. На выполнение пп. 6-20 потребуется $O(C_1 C_2 n^3 + 2C_1 n^2 + C_1 n)$ времени для каждого значения $\nu = 1,...,\nu_{\max}$. Внешний цикл в пп. 6-20 конечен, так как при заданных или незаданных значениях h_i и T_{ij} и текущем значении ν все элементы матрицы Λ в итоге получат нулевые значения, т.е. все x_{ij} будут преобразованы и определится текущее значение F_ν.

Внешний цикл по $\nu = 1,...,\nu_{\max}$ (пп. 4-30) также конечен, что обеспечивается в пп. 23-28. Таким образом, в последовательности получаемых решений $F_1, F_2, ..., F_{\nu_{\max}}$ возможны следующие случаи:

1) $F_1 > F_2 > ... > F_{\nu_{\max}}$, т.е. значения функции цели монотонно уменьшаются и значение $F_{\nu_{\max}}$ является лучшим текущим решением;

2) $F_1 > F_2 > ... > F_{\xi-1} > F_\xi = F_{\xi+1}$, $\xi + 1 \le \nu_{\max}$, т.е. при ξ найден локальный минимум $F_\xi = F_{\min}$;

3) последовательность $F_1, F_2, ..., F_{\nu_{\max}}$ не монотонна, тогда алгоритм среди нескольких локальных минимумов (пп. 23-28) выберет наилучший при значении $\nu = \nu_{opt}$.

Во всех случаях останов алгоритма во внешнем цикле по ν гарантирован, поскольку в итоге будет выполнено одно из условий: $\nu > \nu_{\max}$ или $ZSTOP = ZP$.

Общее время работы алгоритма с учетом операций в пп. 21 и 22 в худшем случае составит $O(\nu_{\max}[C_1 C_2 n^3 + 2C_1 n^2 + C_1 n) + 4n^2])$. Величина C_1 существенно зависит от границ изменения значений a_{ij} по сравнению со значением ω, а также от области допустимых решений, определяемой ограничениями на h_i и T_{ij}. Значение C_1 может быть достаточно большим по сравнению со значением $\nu_{\max} C_2$ и достигать $(n^2 - n)$. Таким образом, в асимптотике время работы алго-

ритма не лучше, чем $O(Cn^5)$, где $C = \nu_{\max} C_2$.

Теорема доказана. ■

Кроме оценки временной сложности алгоритма значительный интерес представляет оценка точности получаемого решения. Как известно [144], разработка приближенных алгоритмов с заданной абсолютной погрешностью для многих NP-трудных задач является проблемой настолько же сложной, как и поиск точного решения. В настоящее время наиболее перспективными направлениями в разработке приближенных (аппроксимационных) алгоритмов являются: поиск полиномиальной или полностью полиномиальной схемы приближенного решения задачи (polynomial or fully polynomial time approximation schemes — PTAS or FPTAS) [157, 162, 163, 351, 480]; разработка вероятностных приближенных алгоритмов [158, 310, 376]; использование для разработки приближенных алгоритмов различных метаэвристик, таких как Simulated Annealing (SA), Tabu Search, Genetic Algorithms, Evolutionary Computation, Variable Neighborhood Search, Ant Colony Optimization, Greedy Randomized Adaptive Search Procedure и др. [187, 195, 302, 446]. Отметим также интересные работы [54, 73], посвященные методам локального поиска для дискретных задач и исследованию эволюционных методов решения задач комбинаторной оптимизации. Поскольку анализ возможного применения к задаче (3.18)-(3.21) алгоритмов с гарантированной оценкой точности, вероятностных (рандомизированных) и метаэвристических алгоритмов это большая отдельная тема, выходящая за рамки настоящей работы, отметим только, что надежда построить для задачи (3.18)-(3.21) алгоритм с гарантированной оценкой точности за полиномиальное время, по-видимому, остается иллюзорной.

Учитывая большую сложность приведенного алгоритма для решения задачи (3.18)-(3.21), в работе [20] предложены ограниченный вариант алгоритма Optpac1 — Optpac2 и ряд других эвристических алгоритмов, разработка которых обоснована тем, что для реальных коммуникационных сетей трудно определить функции C_{tr}^{ij}, C_{sort}^{i} и C_{load}^{i} достаточно адекватно характеризующие затраты на процессы транспортировки и обработки потоков корреспонденций. Расс-

мотрим только несколько алгоритмов из [20], которые при проведении экспериментальных исследований оказались лучшими по границам минимального и максимального отклонения локальных оптимумов и по вычислительной эффективности (быстродействию). Первая группа алгоритмов, в которую включены алгоритмы Optpac1 и Optpac2 применяется для решения задачи в случае, когда можно утверждать, что заданные функции затрат на транспортировку и обработку потоков адекватно отражают стоимостные зависимости реальных процессов от основных изменяемых параметров, вторая группа Optpac3, Optpac4 и Optpac5 — в случае, когда нет твердой уверенности в адекватности используемых функций затрат, а также в случае, когда функции затрат вообще не заданы. Алгоритмы второй группы ориентированы на максимальное сокращение транспортных блоков в сети и не используют в своей работе функции затрат. Однако полученное с помощью этих алгоритмов решение задачи по-прежнему оценивается. Для оценки выполняется расчет по заданным конкретным функциям годовых приведенных затрат.

Алгоритм Optpac2 отличается от Optpac1 ограниченным выбором узлов во множество K, в которое включаются только узлы, находящиеся на кратчайших путях, определяемых из справочной матрицы путей транспортировки потоков транспортных блоков C^*. Поэтому временная сложность действий в п. 9 может быть понижена до $O(C_2 n_{av})$, где n_{av} — среднее число транзитных узлов в пути от i до j в сети G и общая оценка будет не лучше $O(Cn^4 n_{av})$.

В алгоритмах Optpac3 и Optpac4 выбор x_{ij} из X происходит в двойном цикле по i и j из условия $ij \leftarrow \arg\min_{i,j=\overline{1,n}} x_{ij}$ до тех пор, пока возможны преобразования (3.22), а в качестве k выбирается первый узел из полного или ограниченного множества K, если для k выполняются ограничения (3.20), $\Delta^u_{ij,k} \geq 0$ и ограничения $t_{ij} \leq T_{ij}$, $\nu_{ij} \leq \nu$, $t_{rs} \leq T_{rs}$, $\nu_{rs} \leq \nu$ для a_{ij} и множества корреспонденций $\{a^*_{rs}\}$ уже объединенных с a_{ij}. Временная сложность алгоритмов Optpac3 и Optpac4 соответственно составляет $O(\nu_{\max}[C_1(n^2 + C_2 n) + 4n^2])$ и

$O(\nu_{\max}[C_1(n^2 + C_2 n_{av}) + 4n^2])$, а так как для этих алгоритмов $C_1 = n^2 - n$, окончательно получаем

$$O(\nu_{\max}[(n^4 + C_2 n^3 - n^3 - C_2 n^2) + 4n^2]) \text{ и}$$

$$O(\nu_{\max}[(n^4 + C_2 n^2 n_{av} - n^3 - C_2 n n_{av}) + 4n^2]).$$

Алгоритм Optpac5 отличается от алгоритмов Optpac3 и Optpac4 тем, что первоначально элементы матрицы X (когда X совпадает с A) сортируются по неубыванию их значений алгоритмом «карманной» (bucket sort) сортировки с построением вектора указателей на однонаправленные линейные списки элементов x_{ij} с одинаковым значением. Длина вектора указателей определяется максимальным значением $C_L = \lceil a_{\max} / \omega \rceil \omega$ в X. Индекс указателя в векторе указателей равен значению x_{ij}, а сам указатель содержит ссылку на первый элемент списка. Каждый элемент в списке содержит ссылку на следующий элемент и индексы i и j корреспонденции x_{ij}. Первоначально индексы корреспонденций в списках упорядочены в соответствии с просмотром элементов матрицы X слева направо и сверху вниз. В процессе работы алгоритма для изменяемых значений x_{ik} и x_{kj} эти списки динамически корректируются; этим поддерживается постоянное упорядочение. Выбор x_{ij} для выполнения преобразования (3.22) проводится в основном цикле по длине вектора указателей C_L в порядке возрастания значений x_{ij}. Узел k выбирается из ограниченного множества так же, как и в алгоритме Optpac4. Поэтому временная сложность алгоритма Optpac5 составляет $O(\nu_{\max}[C_1(n^2 n_L + C_L n_L C_2 n_{av}) + 4n^2]) = O(\nu_{\max}[n^2 n_L + C_L n_L C_2 n_{av} + 4n^2])$, так как $C_1 = 1$, где n_L определяется средним числом элементов в списке, имеющих одинаковое значение x_{ij}.

3.3.3. Вычислительный эксперимент

Во время проведения эксперимента сначала исследовалось поведение алгоритма Optpac1 при изменении границ значений матрицы корреспонденций A по сравнению со значением ω, определяющим емкость транспортного блока, а

затем сравнивались алгоритмы Optpac1- Optpac5 по быстродействию и полученным результатам. Исследования выполнялись на транспортной сети с длинами дуг r_{ij}, $ij \in E$, в пределах от 80 до 300 км. Значения матрицы корреспонденций и длины дуг генерировались датчиком псевдослучайных чисел. Использовались реальные функции годовых приведенных затрат C_{tr}^{ij}, C_{sort}^{i} и C_{load}^{i}, характерные для автотранспортных предприятий. В табл. 3.4 приведены результаты работы алгоритма Optpac1 при заданных параметрах n = 20, val = 5 (степень узлов), $r_{ij} \in [80, 300]$, $ij \in E$, ω = 40, w_{ξ} = 40, t_a = 24ч, t_b = 12ч, V_{av} = 80км/ч при изменении интервалов значений a_{ij}, $i, j = \overline{1,n}$. Значения h_i, $i = \overline{1,n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ не ограничены.

Таблица 3.4

Результаты работы алгоритма Optpac1

Интервалы значений a_{ij}, $i, j = \overline{1,n}$	F_0, у.е.	$F_0 - F_{\min}$, у.е., (экономия, %)	u_0, $u_{\min}$	ν_{opt}, ν	K_{av}	N_{av}	C_1	t_{sr}, сут.	t_{sol}, с
[1, 5]	518083	304316 (58,74%)	380, 95	5, 7	0,522	4	285	3,67	43,52
[5,10]	569149	242289 (42,57%)	380, 144	4, 6	0,762	7	238	3,34	33,10
[5, 120]	1521120	70517,6 (4,64%)	738, 674	1, 4	0,889	14	89	2,86	2,32
[1, 10]	548326	269046 (49,07%)	380, 123	4, 7	0,715	6	260	3,45	42,15
[10, 20]	651824	151972 (23,31%)	380, 230	1, 6	0,794	11	152	3,03	19,61
[10, 120]	1527020	61987,0 (4,065)	738, 681	2, 3	0,887	15	80	2,85	1,47
[1, 20]	607324	207486 (34,16%)	380, 180	5, 6	0,787	8	204	3,20	30,65
[20, 40]	810251	0 (0,00%)	380, 380	0, 1	0,690	19	0	2,57	0,00
[20, 120]	1548660	30307,9 (1,96%)	738, 712	2, 3	0,884	15	73	2,78	0,86
[1, 40]	722062	115118 (15,94%)	380, 270	5, 6	0,804	13	115	2,96	8,77
[40, 60]	1428870	40289,4 (2,82%)	754, 754	1, 2	0,872	9	193	2,98	6,02
[40, 120]	1774980	18348,4 (1,03%)	871, 871	1, 2	0,892	14	91	2,77	1,09
[1, 60]	954599	109222 (11,44%)	495, 387	3, 5	0,836	13	118	2,99	8,27
[60, 80]	1639140	0 (0,00%)	760, 760	0, 1	0,855	19	0	2,57	0,00
[60,120]	1950360	709,750 (0,04%)	934, 934	1, 2	0,869	18	18	2,62	0,09
[1, 80]	1132660	86884,9 (7,67%)	562, 479	5, 6	0,858	14	97	2,96	5,80
[80, 100]	2222300	0 (0,00%)	1134, 1134	0, 1	0,744	19	0	2,57	0,02
[80, 120]	2283520	0 (0,00%)	1135, 1135	0, 1	0,801	19	0	2,57	0,00
[1, 100]	1342400	83320,0 (6,21%)	664, 586	3, 4	0,872	14	99	2,91	3,81
[100, 120]	2399780	0 (0,00%)	1140, 1140	0, 1	0,907	19	0	2,57	0,00
[1, 120]	1519720	70138,6 (4,62%)	738, 674	3, 4	0,887	14	90	2,90	2,31

В табл. 3.5 приведены результаты сравнения алгоритмов Optpac1- Optpac5 при увеличении n для val = 5, $r_{ij} \in [80, 300]$, $ij \in E$, ω = 40, w_{ξ} = 4000, t_a = 24ч,

t_b = 12ч, V_{av}= 80км/ч, $a_{ij} \in [1, 120]$, $i, j = \overline{1,n}$; (O), (S) — соответственно означает, что алгоритм работает на полном или ограниченном множестве узлов претендентов для преобразования (3.22).

Таблица 3.5

Результаты работы алгоритмов Optpac1-Ooptpac5

Число узлов, n	**Алгоритм**	$F_{\min}$, у.е.	$\Delta F_{\min}$, %	ν_{opt}, ν	K_{av}	N_{av}	C_1	t_{sr}, сут.	t_{sol}, с.
1	2	3	4	5	6	7	8	9	10
25	optpac1 (O)	0,150411E+07	0,00	2, 5	0,891	18	147	3,17	8,66
	optpac2 (S)	+16530	1,10	1, 2	0,805	20	90	2,79	0,25
	optpac3 (O)	+32730	2,18	1, 2	0,902	16	600	3,19	0,02
	optpac4 (S)	+31970	2,13	1, 2	0,832	19	600	2,82	0,05
	optpac5 (S)	+31970	2,13	1, 3	0,832	19	1	2,82	0,06
50	optpac1 (O)	0,588447E+07	0,00	4, 5	0,906	36	642	3,61	390,55
	optpac2 (S)	+86450	1,47	3, 4	0,838	38	513	3,15	9,61
	optpac3 (O)	+100510	1,71	3, 5	0,910	31	2450	3,68	0,36
	optpac4 (S)	+111360	1,89	2, 3	0,845	37	2450	3,16	0,16
	optpac5 (S)	+111160	1,89	2, 3	0,845	37	1	3,16	0,09
75	optpac1 (O)	0,129714E+08	0,00	2, 6	0,919	53	1540	4,00	4262,56
	optpac2 (S)	+234700	1,81	2, 3	0,848	57	1228	3,43	46,77
	optpac3 (O)	+167900	1,29	2, 4	0,917	46	5550	4,06	0,90
	optpac4 (S)	+261000	2,01	2, 3	0,848	55	5550	3,44	0,48
	optpac5 (S)	+261100	2,01	2, 3	0,848	55	1	3,44	0,12
100	optpac1 (O)	0,225900E+08	0,00	3, 7	0,921	71	2748	4,20	22998,70
	optpac2 (S)	+416600	1,84	3, 4	0,850	76	2220	3,59	224,25
	optpac3 (O)	+231300	1,02	4, 5	0,920	61	9900	4,25	3,10
	optpac4 (S)	+462000	2,05	2, 3	0,851	73	9900	3,61	1,40
	optpac5 (S)	+460700	2,04	2, 3	0,851	73	1	3,61	0,16
150	optpac2 (S)	+684400	1,35	3, 4	0,853	114	5173	3,88	1364,03
	optpac3 (O)	0,506015E+08	0,00	2, 4	0,924	91	22350	4,60	11,31
	optpac4 (S)	+745700	1,47	3, 4	0,853	110	22350	3,89	9,02
	optpac5 (S)	+749000	1,48	3, 4	0,853	110	1	3,89	0,44
200	optpac2 (S)	+1495500	1,67	2, 4	0,854	152	9240	4,07	4750,62
	optpac3 (O)	0,892978E+08	0,00	2, 5	0,926	121	39800	4,83	43,96
	optpac4 (S)	+1566600	1,75	3, 4	0,853	146	39800	4,09	28,97
	optpac5 (S)	+1567500	1,76	3, 4	0,853	146	1	4,10	0,84
250	optpac2 (S)	+2724000	1,97	3, 5	0,853	190	14523	4,24	16306,00
	optpac3 (O)	0,138052E+09	0,00	2, 4	0,928	150	62250	5,05	84,68
	optpac4 (S)	+2736000	1,98	3, 4	0,853	182	62250	4,26	72,34
	optpac5 (S)	+2735000	1,98	3, 4	0,853	182	1	4,26	1,65
300	optpac3 (O)	0,197984E+09	0,00	3, 4	0,929	179	89700	5,19	186,39
	optpac4 (S)	+4243000	2,14	3, 4	0,853	219	89700	4,37	163,72
	optpac5 (S)	+4258000	2,15	3, 4	0,853	219	1	4,37	3,12
350	optpac3 (O)	0,268714E+09	0,00	3, 4	0,930	209	122150	5,34	355,67
	optpac4 (S)	+6258000	2,33	3, 4	0,853	255	122150	4,51	317,09
	optpac5 (S)	+6257000	2,33	3, 4	0,853	255	1	4,51	5,99
400	optpac3 (O)	0,349885E+09	0,00	2, 4	0,931	238	159600	5,46	616,22
	optpac4 (S)	+8555000	2,45	3, 4	0,854	291	159600	4,61	551,29
	optpac5 (S)	+8554000	2,44	4, 5	0,854	291	1	4,60	15,01
450	optpac3 (O)	0,441988E+09	0,00	3, 4	0,932	267	202050	5,57	1009,65
	optpac4 (S)	+11290000	2,55	3, 4	0,853	327	202050	4,69	1031,15
	optpac5 (S)	+11292000	2,55	3, 4	0,853	327	1	4,69	25,02
500	optpac3 (O)	0,544746E+09	0,00	2, 4	0,932	297	249500	5,68	1831,16
	optpac4 (S)	+14418000	2,65	3, 4	0,853	363	249500	4,79	1793,43
	optpac5 (S)	+14444000	2,65	3, 4	0,853	364	1	4,79	47,85

Значения h_i, $i=\overline{1,n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ при проведении эксперимента не ограничивались в целях максимально возможного расширения области допустимых решений задачи. В таблицах приняты обозначения: F_0, $F_{\min}$ — соответственно значения функции цели (3.18) (в условных единицах приведенных затрат) до и после оптимизации; $\Delta F_{\min}$ — отклонение целевой функции от лучшего значения (в процентах); $u_0 = \sum_{ij \in S} \left\lceil \frac{a_{ij}}{\omega} \right\rceil$, $u_{\min} = \sum_{ij \in S} \left\lceil \frac{x_{ij}}{\omega} \right\rceil$ — количество транспортных блоков до и после оптимизации; ν_{opt}, ν — оптимальное и конечное число объединений корреспонденций (ν определяет число итераций алгоритмов для нахождения локального оптимума); $K_{av} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\delta_i} \sum_{j=1}^{n} x_{ij} / \left(\left\lceil \frac{x_{ij}}{\omega} \right\rceil \omega \right)$, $\delta_i = \sum_{j=1}^{n} \delta_{ij}$ — средний коэффициент загрузки транспортного блока; $N_{av} = \frac{1}{n}\sum_{i=1}^{n} \delta_i$ — среднее число направлений сортировки мелкопартионных корреспонденций в узле сети; C_1 — число вхождений в главный цикл оптимизации для выполнения преобразований (3.22); t_{sr} — среднее время доставки мелкопартионных корреспонденций получателю (в сутках); t_{sol} — время решения задачи (в секундах).

Из табл. 3.4 видно, что на результаты решения задачи существенно влияют границы изменения значений a_{ij}, $i, j = \overline{1,n}$, по сравнению с величиной ω. При малой постоянной нижней границе и увеличении верхней границы значений a_{ij} ($a_{ij} \in$ [1, 5-120]) экономия затрат составляет от 58,74% до 4,62%.

При росте нижней и верхней границ и росте нижней и постоянной высокой верхней границе показатели экономии соответственно снижаются с 42,57% и 4,64% до нуля.

Для интервалов [20, 40], [60, 80], [80, 100], [80, 120], [100, 120] вообще невозможно уменьшить значение целевой функции, так как выполнение преобразований (3.22) может только увеличить количество транспортных блоков. В то же время для интервалов [40, 60], [40, 120], [60, 120], когда количество транс-

портных блоков до и после оптимизации не изменяется (при $\Delta^{u}_{ij,k} = 0$), имеем снижение затрат с 2,82% до 0,04%, что при большой размерности сети может составлять значительную сумму и представлять практический интерес. Несмотря на то, что значения h_i, $i = \overline{1,n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ не ограничивались, среднее время доставки мелкопартионных корреспонденций получателю оставалось достаточно стабильным и не превышало 3,67 суток. Во всех случаях, когда оптимизация была возможна, достигался неплохой коэффициент загрузки транспортных блоков, и уменьшалось количество направлений сортировки мелкопартионных корреспонденций в узлах сети.

Результаты работы алгоритмов Optpac1-Optpac5 при увеличении n от 25 до 500 (приведенные в табл. 3.5), показывают большую вычислительную трудоемкость алгоритмов Optpac1 и Optpac2. Алгоритм Optpac1 дает лучшие результаты, но его «разумное» применение на обычных персональных компьютерах возможно на сетях, содержащих до 100 узлов. Алгоритмы Optpac3-Optpac5 работают значительно быстрее, но и отклонения от лучших результатов алгоритма Optpac1 (на сетях до 100 узлов) составляют более 2%, что в абсолютном выражении по приведенным затратам достигает величины 462000 у.е. Начиная с размерности сети n = 75 алгоритм Optpac3 показывает лучшие результаты, чем алгоритм Optpac2, и при увеличении размерности сети остается лидером среди алгоритмов, не использующих в процессе оптимизации функции затрат.

Особо отметим, что все алгоритмы допускают распараллеливание вычислительного процесса по значениям размера окрестности поиска локального минимума (параметра $\nu = 1,2,..\nu_{\max}$). Поэтому решение задачи упаковки возможно на мощных вычислительных кластерах (High performance computing clusters, HPC) и многопроцессорных компьютерах, что существенно сократит время ее решения на сетях большой размерности.

Из практических соображений укажем, что для транспортных сетей, связанных с перевозкой мелкопартионных грузов обычно характерно, что для большинства корреспонденций выполняется $a_{ij} < \omega$. Для магистральных сетей

передачи данных с волоконно-оптическими линиями связи и технологиями спектрального уплотнения каналов WDM, DWDM, HDWDM емкость транспортного блока (виртуального контейнера) ω кратна пропускной способности линии связи и $a_{ij} << \omega$. Решение задачи упаковки для таких сетей становится особенно актуальным, поскольку позволит получить значительную экономию дефицитных материальных и финансовых ресурсов. Для сетей большой размерности (от n = 200 и выше) даже 1,67%-2,65% отклонения решений от лучшего локального минимума составляют в абсолютном выражении огромные суммы (от +1495500 до +14444000 у.е. для примера из табл. 3.5).

Проведенный эксперимент, безусловно, не претендует на полноту исследования описанных алгоритмов, поскольку такая работа связана со значительными затратами машинного времени и требует проверки стратегий оптимизации на представительном множестве исходных данных разных размерностей (генерируемых, например, различными датчиками случайных чисел). Несмотря на это, в ходе эксперимента подтвердились теоретические положения о сходимости и временной сложности алгоритмов, выявлена аномалия решения задачи для различных границ значений a_{ij} и ω.

Результаты эксперимента позволяют сделать следующие выводы: при неуверенности в адекватности функций затрат или при неизвестных функциях затрат для решения задачи следует использовать алгоритмы Optpac3-Optpac5, причем, последний алгоритм целесообразно применять для решения задач большой размерности, вследствие значительного сокращения времени счета; для адекватных функций затрат наилучшие результаты дает алгоритм Optpac1 и его предпочтительнее использовать для решения задачи упаковки на вычислительных кластерах, жертвуя затраченным временем решения.

Все расчеты получены на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.

3.4. Задача упаковки с ограничениями на пропускные способности дуг

В подразделе рассматривается задача упаковки, в которой наряду с функ-

циями затрат на обработку и транспортировку мелкопартионных корреспонденций и ограничениями на пропускные способности узлов заданы ограничения на пропускные способности дуг сети [21, 121]. Доказано, что при решении задачи упаковки нелинейная функция транспортных затрат может быть заменена на функцию удельной стоимости транспортировки транспортных блоков от расстояния и грузоподъемности транспортных средств или пропускной способности каналов связи, а задача выбора пропускных способностей дуг может решаться независимо при оптимизации распределения сформированных потоков транспортных блоков.

3.4.1. Постановка и математическая модель задачи

Рассмотрим содержательную постановку задачи. Пусть задана простая (без петель и кратных ребер) ориентированная сеть $G(N, E)$ с множеством узлов N, $n = |N|$ и множеством дуг E, $e = |E|$, где n и e соответственно число узлов и дуг сети, а $|\cdot|$ — знак мощности множества. Будем считать, что сеть такова, что для каждой прямой дуги kl, $(k < l)$ существует обратная lk, $(l > k)$. На сети задана целочисленная матрица потоков $A = \| a_{ij} \|_{n \times n}$. Величины a_{ii}, $i = \overline{1, n}$ определяют внутриузловые объемы на обработку потоков. Потоки a_{ij} из источников i в стоки j, $i, j = \overline{1, n}$, $i \neq j$, должны одновременно передаваться в некоторых транспортных блоках (контейнерах) объема $\omega >> a_{ij}$ с заданной периодичностью. При этом предполагается, что каждый поток может быть упакован в транспортный блок только целиком. Пусть S — множество пар индексов потоков (i, j), определенное на декартовом произведении $n \times n$; x_{ij}, u_{ij} — переменные, определяющие соответственно величину потока (количество единиц потока) из i в j в исходных единицах и транспортных блоках. Первоначально все $x_{ij} = a_{ij}$, а $u_{ij} = \left\lceil \dfrac{x_{ij}}{\omega} \right\rceil$, $ij \in S$, где знаки $\lceil \cdot \rceil$ — означают округление числа до большего целого; w_{kl}, $kl \in E$ — искомые пропускные способности дуг сети в

транспортных блоках, $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$, w_i, $i = \overline{1,\alpha}$ — упорядоченные по неубыванию целые положительные числа; d_{kl}, $kl \in E$ — длины дуг; f_{kl}, $kl \in E$ — суммарные многопродуктовые потоки в транспортных блоках, протекающие по дугам сети при заданном способе маршрутизации потоков, $f_{kl} = \sum_{ij \in S} u_{ij}^{kl}$, где u_{ij}^{kl} — поток транспортных блоков из i в j, проходящий по дуге kl; t_{av}, $T_{\max}$ — расчетная средняя и заданная максимальная задержка в передаче потоков в сети; t_{ij}, T_{ij}, $ij \in S$ — расчетное и заданное время на доставку исходных потоков a_{ij} из i в j; h_i — пропускная способность i-го узла в исходных единицах потока; ν_{ij}, $ij \in S$, $\nu_{\max}$ — число объединений (слияний) потока a_{ij} с другими потоками и максимально допустимое число объединений; C_{tr}^{kl}, C_{sort}^{i}, C_{load}^{i} — некоторые нелинейные функции приведенных затрат на транспортировку и обработку потоков. Требуется минимизировать функцию:

$$\sum_{kl \in E} C_{tr}^{kl}(w_{kl}, d_{kl}) + \sum_{i=1}^{n} C_{sort}^{i}(x_i, q_i) + \sum_{i=1}^{n} C_{load}^{i}(u_i), \tag{3.25}$$

при ограничениях баланса потоков

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1, j \neq i}^{n} a_{ij} = \sum_{j=1}^{n} x_{ji} - \sum_{j=1, j \neq i}^{n} a_{ji}, \; i = \overline{1,n},$$

и ограничениях:

$$w_{kl} > f_{kl}, \; w_{kl} \in \{w_1, w_2, ..., w_\alpha\}, \; \forall kl \in E, \tag{3.26}$$

$$t_{av} = 1/U_\Sigma \sum_{kl \in E} f_{kl}/(w_{kl} - f_{kl}) \leq T_{\max}, \; U_\Sigma = \sum_{ij \in S} u_{ij}, \tag{3.27}$$

$$x_i \leq h_i, \; i = \overline{1,n}, \tag{3.28}$$

$$t_{ij} \leq T_{ij}, \; ij \in S, \tag{3.29}$$

$$\nu_{ij} \leq \nu_{\max}, \; ij \in S. \tag{3.30}$$

Где $x_i = a_{ii} + \sum_{j=1, j \neq i}^{n} (a_{ij} + a_{ji}) + \sum_{j=1}^{n} x_{ij} - \sum_{j=1, j \neq i}^{n} a_{ij}$, $i = \overline{1,n}$; $q_i = q_{in}^{i} + \sum_{j=1}^{n} \delta_{ij}$ — количество направлений сортировки потоков, обрабатываемых в узле i ($\delta_{ij} = 1$, если $x_{ij} \neq 0$, и $\delta_{ij} = 0$, если $x_{ij} = 0$, а q_{in}^{i} определяет заданное количество направле-

ний сортировки для обработки потоков a_{ii}), $i=\overline{1,n}$; $u_i=\sum_{j=1}^{n}(u_{ij}+u_{ji})$, $i=\overline{1,n}$.

Первая составляющая функции (3.25) C_{tr}^{kl} определяет транспортные затраты (затраты на передачу потоков), вторая C_{sort}^{i} — затраты на обработку (сортировку) исходящих, входящих и транзитных потоков в исходных единицах, третья C_{load}^{i} — затраты на обработку исходящих и входящих потоков в транспортных блоках (контейнерах). Функции C_{tr}^{kl} имеют дискретный характер и их общий вид в зависимости от f_{kl} и d_{kl} приведен на рис. 3.4. При $f_{kl}>w_{kl}$, $w_{kl}\in\{w_1,w_2,...,w_\alpha\}$, $kl\in E$ необходимо увеличить пропускную способность дуги kl до большего значения из допустимого множества. Более того, при решении задачи пропускную способность дуг нужно увеличивать до тех пор, пока не начнет выполняться ограничение (3.27). Для реальных задач это означает, что на данном участке нужно вводить дополнительные транспортные средства или каналы связи с той же грузоподъемностью или пропускной способностью, либо выбрать новое транспортное средство или новый канал связи с большей грузоподъемностью или пропускной способностью.

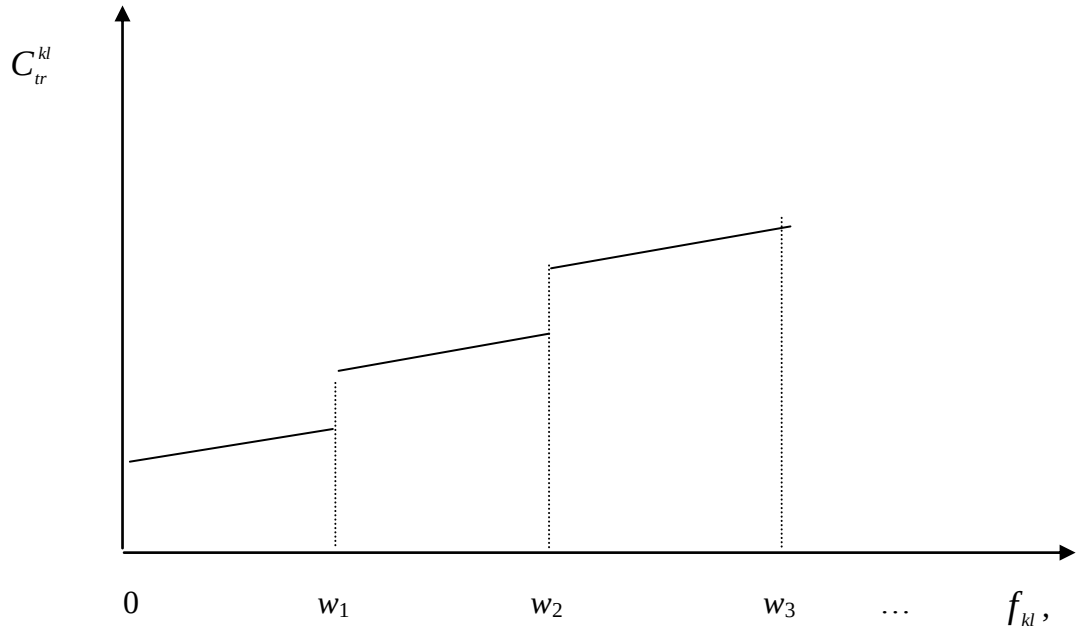


Рис. 3.4

Функции C_{sort}^{i}, C_{load}^{i} могут быть как выпуклыми, так и вогнутыми для различных коммуникационных сетей. Так, например, для решения задач упаковки

и распределения мелкопартионных потоков в транспортных сетях в настоящей работе использован следующий конкретный вид функций:

$$C_{tr}^{kl}(w_{kl},d_{kl}) = k_1^{kl} + k_2^{kl} d_{kl},\ w_{kl} \in \{w_1, w_2, ..., w_\alpha\},\ kl \in E,$$
$$C_{sort}^{i}(x_i,q_i) = k_3^i x_i \exp(-k_4^i x_i /(1+q_i)),\ i = \overline{1,n}, \qquad (3.31)$$
$$C_{load}^{i}(u_i) = \sqrt{k_5^i u_i^2 + k_6^i u_i}\ ,\ i = \overline{1,n},$$

где k_1^{kl}, k_2^{kl}, k_3^i, k_4^i, k_5^i, k_6^i — заданные коэффициенты.

Если C_{sort}^i известны, то при условии их адекватности реальным процессам, проектировщик после решения задачи получит достаточно точную оценку среднегодовых приведенных затрат на обработку (сортировку) всех заданных исходных потоков в сети. Для затрат C_{load}^i на обработку транспортных блоков может быть получена только нижняя оценка, поскольку не будут учтены затраты на обработку транзитных потоков и, для случая транспортной сети, затраты на обработку порожних контейнеров. Относительно транспортных затрат C_{tr}^{kl} следует сделать два важных замечания: во-первых, в процессе решения задачи упаковки невозможно рассчитывать стоимости дуг используя дискретные функции, так как пропускная способность дуг неизвестна; для транспортных сетей в модели не учитываются потоки порожних контейнеров. Поэтому после решения задачи можно получить лишь приблизительную оценку реальных транспортных затрат.

Полные транспортные и погрузочно-разгрузочные затраты (или затраты на передачу и коммутацию) могут быть получены только после решения задачи оптимального распределения потоков транспортных блоков с последующей корректировкой пропускных способностей дуг и пересчетом затрат.

Утверждение 3.1. Решение задачи (3.25)-(3.30) может быть сведено к независимому последовательному решению следующих задач:

Задача 1. Задача упаковки (рассмотренная в подразделе 3.3). Найти минимальное значение функции

$$\sum_{ij \in S} C_{tr}^{ij}(u_{ij},d_{ij}) + \sum_{i=1}^{n} C_{sort}^{i}(x_i,q_i) + \sum_{i=1}^{n} C_{load}^{i}(u_i), \qquad (3.32)$$

при ограничениях (3.28)-(3.30);

Задача 2. Выбор (определение) пропускных способностей дуг (ВПС). Найти минимальное значение функции

$$\sum_{kl\in E} C_{tr}^{kl}(w_{kl}, d_{kl}), \tag{3.33}$$

при ограничениях (3.26), (3.27).

Доказательство. Сразу отметим, что в постановке задачи (3.32), (3.28)-(3.30) не учитываются ограничения на пропускные способности дуг, а первая составляющая функции (3.25), определяющая транспортные затраты, заменена на другую, зависящую от объемов передаваемых потоков u_{ij} и расстояний d_{ij} между i и j для всех $ij \in S$. Для обоснования сделанной подмены рассмотрим сущность итерационных процессов объединения и упаковки потоков при минимизации (3.25) и (3.32). В подразделе 3.3 предложен приближенный метод решения задачи упаковки, идея которого заключается в выборе наилучшей стратегии определения потоков x_{ij}, над которыми итеративно выполняются операции объединения (3.22):

$$x_{ik} \leftarrow x_{ik} + x_{ij},\ x_{kj} \leftarrow x_{kj} + x_{ij},\ c_{ij} \leftarrow k,\ y_k \leftarrow y_k + x_{ij},\ x_{ij} \leftarrow 0,$$

где k — узел, через который выполняется объединение потоков, $k \in K$, K — множество узлов-претендентов, через которые может быть выполнено объединение потока x_{ij} без нарушения ограничений (3.28)-(3.30); c_{ij} — элементы справочной матрицы слияния потоков, определяемые согласно (3.23).

При этом выбор x_{ij} и k для преобразования (3.22) осуществляется по следующей схеме:

1. Для $\forall ij \in S$ определить

$$\Delta_{ij} = \max_{k\in K}(\Delta F_{ij}^{-}(k) - \Delta F_{ij}^{+}(k)),\ c_{ij} = \arg\max_{k\in K}(\Delta F_{ij}^{-}(k) - \Delta F_{ij}^{+}(k)),$$

где $\Delta F_{ij}^{-}(k) = \sum_{ij\in\{ik,kj,ij\}} C_{tr}^{ij}(u_{ij}, d_{ij}) + \sum_{i\in\{i,k\}} C_{sort}^{i}(x_i, q_i) + \sum_{i\in\{i,k,j\}} C_{load}^{i}(u_i),\ u_i = \sum_{l=1}^{n}(u_{il} + u_{lj}),$

$$\Delta F_{ij}^{+}(k) = \sum_{ij\in\{ik,kj\}} C_{tr}^{ij}(u_{ij}^{'}, d_{ij}) + C_{sort}^{i}(x_i, q_i - 1) + C_{sort}^{k}(x_k + x_{ij}, q_k) + \sum_{i\in\{i,k,j\}} C_{load}^{i}(u_i^{'}),$$

$$u_i^{'} = \sum_{l=1}^{n}(u_{il}^{'} + u_{li}^{'}),\ i \in \{i,k,j\},\ u_{ik}^{'} = \left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil,\ u_{kj}^{'} = \left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil,\ u_{ij}^{'} = 0,\ k \in K.$$

2. Найти $ij = \arg\max_{ij \in S} \Delta_{ij}$, $k = c_{ij}$.

Учитывая приведенный способ получения оценок Δ_{ij}, становится очевидным, что транспортные затраты для всех $ij \in S$ можно принять линейными от величины и расстояния и использовать для их расчета любую функцию удельной стоимости транспортировки потока величиной u_{ij} на расстояние d_{ij} от пропускной способности $w_\xi \in \{w_1, w_2, ..., w_\alpha\}$, $\xi = \overline{1,\alpha}$. Так, например, можно принять

$$C_{tr}^{ij}(u_{ij}, d_{ij}) = u_{ij}(k_1^\xi + k_2^\xi d_{ij}) / w_\xi,\ \forall\, ij \in S, \qquad (3.34)$$

где k_1^ξ, k_2^ξ — заданные коэффициенты.

Если выбрать для всех дуг наименьшую пропускную способность (w_1), то можно гарантировать, что значение $\sum_{kl \in E} C_{tr}^{kl}(w_{kl}, d_{kl})$ будет минимальным. Последний факт и то обстоятельство, что функции C_{sort}^i и C_{load}^i явно не зависят от w_{kl} позволяют утверждать о правомерности сделанной подмены первой составляющей функции (3.25) и дают возможность после решения задачи упаковки (3.32), (3.28)-(3.30) перейти к независимому решению задачи выбора пропускных способностей (3.33), (3.26)-(3.27). Утверждение доказано. ■

Утверждение 3.2. При решении задачи (3.32), (3.28)-(3.30) увеличение величины w_ξ приводит к уменьшению транспортных затрат и уменьшению значения t_{av}, т.е. имеется возможность предварительной оценки величины пропускной способности, при которой начинает выполняться ограничение (3.27).

Доказательство. Из (3.34) непосредственно следует, что стоимость транспортировки (передачи) единицы потока с увеличением грузоподъемности транспортных средств или пропускной способности каналов связи уменьшается. При этом уменьшается также значение средней задержки в сети t_{av}, т.е. увеличивая w_ξ можно добиться выполнения условия $t_{av} \le T_{\max}$ и получить верхнюю

оценку пропускной способности для дуг сети. ■

При численном моделировании задачи (3.32), (3.28)-(3.30) проектировщику выводятся до и после оптимизации следующие показатели: полные затраты; транспортные затраты; затраты на сортировку исходных потоков в узлах сети; затраты на погрузку-выгрузку (обработку) транспортных блоков в узлах сети; затраты на приобретение транспортных блоков; общее количество транспортных блоков в сети, необходимое для единовременного отправления всех потоков в сети; средний коэффициент загрузки транспортного блока в сети, среднее число направлений сортировки исходных потоков в узлах сети; суммарный объем транзитной обработки исходных потоков в узлах сети; минимальное, максимальное и среднее время доставки исходных потоков получателю и др.

Кроме того, до и после оптимизации выводится предварительная оценка среднего времени задержки потоков в сети t_{av}. Если t_{av}= infinit, то на одной или более дуг сети суммарная потоковая нагрузка превышает пропускную способность, заданную параметром w_{ξ}. В этом случае, программа моделирования подсказывает, до какого значения нужно увеличить пропускную способность дуг.

Потоковая матрица в транспортных блоках и матрица с предварительными оценками сроков доставки исходных потоков получателю используются как входные данные для задач выбора пропускных способностей и распределения потоков. Справочная матрица объединения (сортировки) потоков является основным результатом решения задачи упаковки (формирования потоков транспортных блоков). Она полностью определяет схему сортировки исходных единичных потоков во всех узлах сети и адресует потоки транспортных блоков, которые будут распределены по маршрутам транспортных средств или по каналам связи при решении задачи распределения потоков. В узлах реальной транспортной сети или сети передачи данных справочная матрица объединения потоков должна использоваться для автоматизированного управления оборудованием, осуществляющим процессы сортировки адресных грузов или как таблица слияния пакетов сообщений в виртуальные контейнеры.

3.4.2. Вычислительный эксперимент

Для практической проверки приведенных утверждений был проведен вычислительный эксперимент на сети передачи данных содержащей n = 12 узлов первого типа и e = 72 ориентированные дуги, сгенерированной датчиком псевдослучайных чисел. Дуга сети представляет скоммутированную линию связи, состоящую из одного или пучка элементарных каналов. Напомним, что элементарный канал это базовая техническая характеристика, представляющая фиксированное значение пропускной способности. В сетях с коммутацией каналов пропускная способность каждой линии связи должна быть равна целому числу элементарных каналов. Отметим, что для транспортной сети при решении задачи выбора пропускных способностей, маршрут транспортного средства представляется одной дугой, узлы которой совпадают с начальным и конечным узлами маршрута (циклические маршруты разбиваются на две дуги). Длины дуг d_{kl} генерировались в пределах от 60 до 800 км; величины исходных потоков a_{ij} — от 1 до 30 пакетов сообщений в секунду (сообщения могут измеряться в пакетах с одинаковой емкостью или непосредственно в Кбит/с, Мбит/с и Гбит/с); объем транспортного блока ω принимался равным 100 пакетам (можно задавать и в непосредственных единицах измерения); скорость передачи данных V_{av} = 25000 км/сек; время на обработку (мультиплексирование) исходных потоков в узлах сети t_a = 1 секунда; время на транзитную коммутацию транспортных блоков в узлах сети t_b = 0,5 секунды; пропускные способности узлов по обработке транзитных пакетов сообщений принимались равными 500 пакетов в секунду для всех узлов сети; пропускные способности дуг в транспортных блоках выбирались из множества {10, 15, 20,…,35}; максимальное время доставки пакетов сообщений получателю T_{ij} = 10 сек. для всех $ij \in S$; максимальное число объединений исходных потоков $\nu_{\max}$ не ограничивалось; максимальная задержка транспортных блоков в сети $T_{\max}$ = 0,05. В качестве функций затрат использовался конкретный вид формул (3.34) и (3.31) в условных единицах стоимости. Для расчета времени доставки пакетов сообщений конечному получателю при-

менялись формулы (3.24).

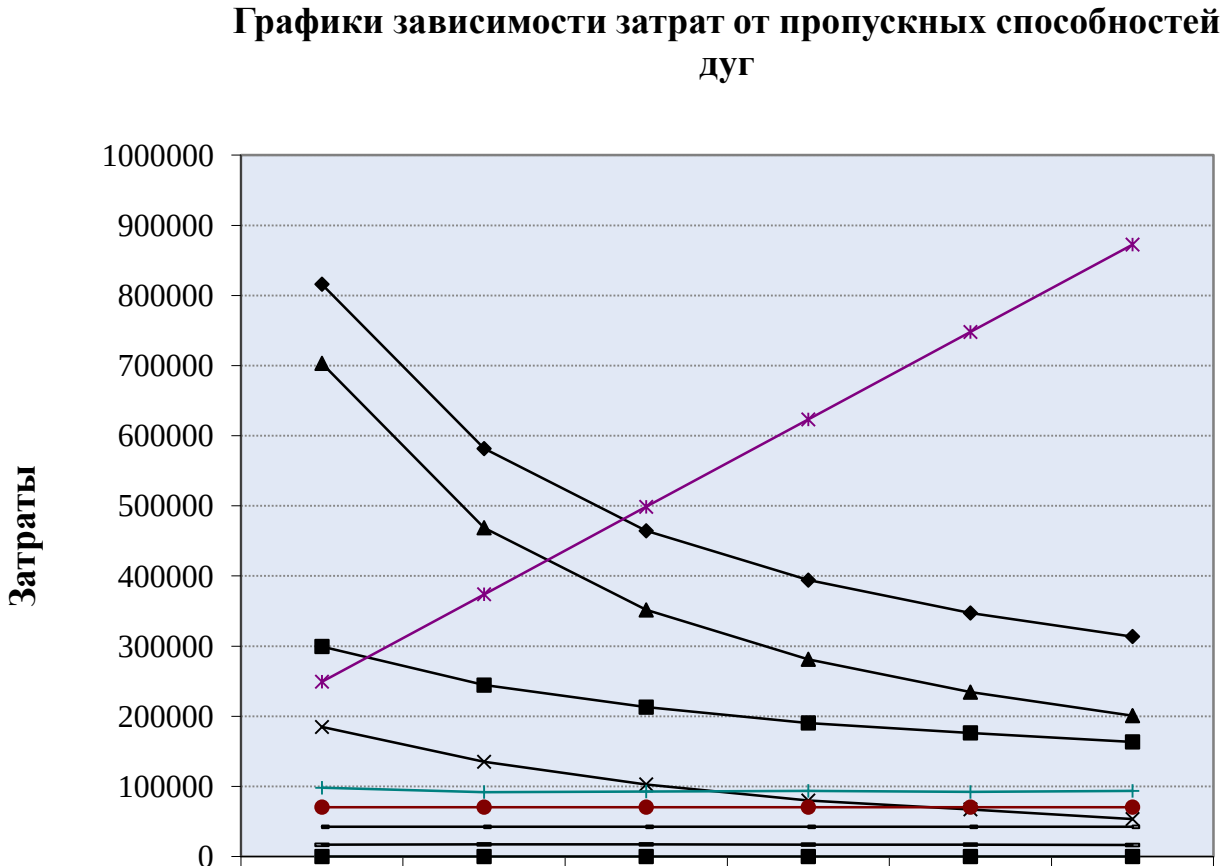


Рис. 3.5

При распределении всех исходных потоков транспортных блоков $u_{ij} = \left\lceil \frac{a_{ij}}{\omega} \right\rceil$ по ступенчатому критерию — минимум транзитных узлов, минимум длины пути без оптимизации упаковок пакетов сообщений и w_{kl} = 10 для всех

$kl \in E$, были нарушены ограничения (3.26) и t_{av} = infinit. Оптимальное решение задачи упаковки для всех w_{kl} = 10 было достигнуто при $\nu_{\max}$ = 5 со значением t_{av} = 0,19315 (см. рис. 3.5). При этом транспортные затраты $C_{tr}^{+} = \sum_{ij \in S} C_{tr}^{ij}(u_{ij}, d_{ij})$ = 184745 были максимальними, а $C_{tr}^{can} = \sum_{kl \in S} C_{tr}^{kl}(w_{kl}, d_{kl})$ = 249300 — минимальными. Далее задача упаковки решалась для w_{kl} = 15, 20,…, 35, пока значение t_{av} = 0,04490 (при w_{kl} = 35) не стало меньше максимально допустимой задержки транспортных блоков в сети $T_{\max}$ = 0,05. При этом транспортные затраты C_{tr}^{+}=53473 стали минимальными, а C_{tr}^{can}=872550 — максимальными.

На рис. 3.5 приняты следующие обозначения: C_{all}^{-}, C_{all}^{+} — полные затраты до и после оптимизации (значения функции (3.32)); C_{tr}^{-}, C_{tr}^{+}, C_{sort}^{-}, C_{sort}^{+}, C_{load}^{-}, C_{load}^{+} — соответственно суммарные транспортные затраты, суммарные затраты на сортировку пакетов сообщений в узлах сети, суммарные затраты на обработку исходящих и входящих транспортных блоков в узлах сети до и после оптимизации из функции (3.32); C_{tr}^{can} — суммарные транспортные затраты, рассчитанные по конкретной формуле вида (3.31) при одинаковой пропускной способности всех дуг сети.

Другие результирующие показатели решения задачи упаковки при изменении w_{kl} от 10 до 35 приведены в табл. 3.6.

Поскольку при решении задачи упаковки затраты на сортировку исходных потоков (пакетов сообщений) и обработку исходящих и входящих потоков транспортных блоков почти не изменяются при изменении значения w_{kl} от минимального до максимального (от 10 до 35 в примере, см. рис. 3.5), то в качестве входных данных для решения задачи выбора пропускных способностей дуг сети можно взять любой из вариантов решения задачи упаковки для w_{kl} от 10 до 35. Рекомендуется выбирать вариант решения, при котором суммарные затраты на сортировку и обработку потоков минимальны. Для задачи выбора пропускных способностей был выбран вариант для w_{kl} = 30 (см. табл. 3.6), в результате решения которой аппроксимационным методом (см. далее по тексту) было по-

лучено значение функции (3.33) $C_{tr}^{can} = \sum_{kl \in S} C_{tr}^{kl}(w_{kl}, d_{kl})$ = 452488,38 при t_{av} = 0,04919762.

Таблица 3.6

Наименования показателей	Значения показателей						
	до опти-ми-зации	после оптимизации					
Сумма затрат на сортировку и обработку потоков (в усл. ед.)	112804	114763	109317	110088	110656	**109098**	109997
Эффект от оптимизации (в усл. ед.)	-	516542	337225	251407	203589	170889	150261
Общее число исходящих транспортных блоков в сети	132	52	54	54	53	53	51
Средний коэффициент загрузки транспортного блока в сети	0,151	0,689	0,593	0,595	0,632	0,612	0,661
Среднее число направлений сортировки пакетов сообщений в узлах сети	11	4	4	4	4	4	4
Суммарный объем транзитной обработки пакетов сообщений в узлах сети	0	1661	1296	1343	1396	1302	1390
Минимальное время доставки пакетов сообщений получателю (сек.)	-	2,00244	2,00244	2,00244	2,00244	2,00244	2,00244
Максимальное время доставки пакетов сообщений получателю (сек.)	-	7,11988	6,61988	7,62088	6,61988	6,61988	6,61948
Среднее время доставки пакетов сообщений получателю (сек.)	-	3,34831	3,23806	3,26042	3,23790	3,24601	3,21024
Максимальное число объединений каждого исходного потока, при котором достигается минимум функции (11)	0	5	5	5	3	3	4
Среднее время задержки до оптимизации	-	infinit	0,29782	0,16703	0,11697	0,09016	0,07339
Среднее время задержки после оптимизации	-	0,19315	0,12753	0,08849	0,06720	0,05698	0,04490
Величина пропускной способности дуг в транспортных блоках в секунду	10	10	15	20	25	30	35

Результаты проведенного эксперимента подтверждают возможность замены функций $C_{tr}^{kl}(w_{kl}, d_{kl})$ на $C_{tr}^{ij}(u_{ij}, d_{ij})$ и сведения решения общей задачи (3.25)-(3.30) к последовательному независимому решению задач упаковки (3.32),(3.28)-(3.30) и выбора пропускных способностей дуг сети (3.33), (3.26)-(3.27). Аналогичные по характеру поведения численные результаты были получены при решении задачи на сетях большой размерности (до 1000 узлов и 200000 ориентированных дуг).

3.4.3. Задача выбора пропускных способностей дуг

Перейдем к обсуждению решения задачи выбора пропускных способностей (3.33), (3.26)-(3.27) [123]. Для прояснения алгоритмической сложности решения задачи, запишем ее в следующем виде:

$$\min \sum_{i=1}^{e}\sum_{j=k}^{\alpha} c_{ij}x_{ij},\ k = \mathrm{The}\ \underset{\xi=1,\alpha}{\mathrm{first}}\ \mathrm{index}\,(w_{i\xi} > f_i),\ i=\overline{1,e}, \tag{3.35}$$

при ограничениях

$$t_{av} = \sum_{i=1}^{e}\sum_{j=k}^{\alpha} t_{ij}^{av}x_{ij} \le T_{\max}u_{\Sigma}, \tag{3.36}$$

$$\sum_{j=k}^{\alpha} x_{ij} = 1,\ i=\overline{1,e}, \tag{3.37}$$

$$x_{ij} \in \{0,1\}, \tag{3.38}$$

где стоимости дуг c_{ij} вычислены, например, как $c_{ij} = k_{ij}^1 + k_{ij}^2 d_i,\ j=\overline{k,\alpha},\ i=\overline{1,e}$, а $k_{ij}^1,\ k_{ij}^2$ — заданные стоимостные коэффициенты для дуги i в состоянии j, $w_{ij} \in \{w_1, w_2, ..., w_\alpha\}$; d_i — длина дуги i; средние задержки в дугах $t_{ij}^{av} = f_i/(w_{ij} - f_i),\ j=\overline{k,\alpha},\ i=\overline{1,e};\ f_i$ — многопродуктовый поток по дуге i.

Искомые пропускные способности w_{ij} соответствуют оптимальному решению задачи x_{ij}^{opt}. Сформулированная задача известна как задача о ранце с мультивыбором (Multiple-choice Knapsack Problem, MCKP) и булевыми переменными и относится к классу NP-трудных, т.е. в общем случае для ее решения неизвестны точные полиномиально ограниченные по трудоемкости алгоритмы. Нетрудно видеть, что любую индивидуальную задачу вида (3.33), (3.26)-(3.27) можно за время $O(e\alpha)$ преобразовать в соответствующий экземпляр задачи (3.35)-(3.38). Для этого необходимо построить матрицу размером $e \times \alpha$, строки которой соответствуют дугам, столбцы — набору дискретных пропускных способностей, а в качестве элементов матрицы выступают стоимости дуг.

В книге М. Гэри и Д. Джонсона [45, стр. 289] рассматривается NP-полная задача выбора пропускных способностей, сформулированная в следующем виде распознавания. Задано множество линий связи E, множество пропускных

способностей $W \subseteq Z^+$, функция стоимости $c: E \times W \to Z^+$, функция штрафа за задержку $t: E \times W \to Z^+$, такие, что для всех упорядоченных по возрастанию пропускных способностей $w_i \in \{w_1, w_2, ..., w_\alpha\} \in W$ и $i < j$ выполняются неравенства $c_l(w_i) \le c_l(w_j)$ и $t_l(w_i) \ge t_l(w_j)$, $l \in E$. В задаче спрашивается, существует ли такой выбор пропускных способностей w_i, что общая стоимость $\sum_{l \in E} c_l(w) \le C$, а общий штраф за задержку $\sum_{l \in E} t_l(w) \le T$, где C и T — заданные положительные целые числа. Отмечается, что задача разрешима за псевдополиномиальное время.

Утверждение 3.3. Задачи (3.33), (3.26)-(3.27) и (3.35)-(3.38) NP-полны.

Доказательство. Задача (3.33), (3.26)-(3.27), сформулированная как задача распознавания принадлежит к классу NP, так как очевидно, что для нее существует сертификат (удостоверение) длиной e, представляющий собой набор оптимальных пропускных способностей линий связи (дуг), который может быть проверен за линейное время. Нетрудно видеть, что любая сформулированная выше индивидуальная задача за полиномиальное время преобразуется (по Карпу [334]) в задачу (3.33), (3.26)-(3.27), что и доказывает NP-полноту последней. В свою очередь, как было показано, задача (3.33), (3.26)-(3.27) полиномиально преобразуется в задачу (3.35)-(3.38). ■

3.4.3.1. Алгоритмы решения задачи выбора пропускных способностей

Обсудим важность ограничений на среднюю задержку (3.27), (3.36) при решении задач проектирования транспортных сетей и сетей передачи данных. Надо учитывать, что эти формулы были впервые получены Л. Клейнроком [67] для сетей передачи данных с идеализированными условиями. Предполагалось, что все каналы и узлы абсолютно надежны и бесшумны, время обработки пакетов в узлах пренебрежительно мало по сравнению со временем передачи по каналу, внутренние потоки между узлами сети распределены по закону Пуассона, являются независимыми и обрабатываются (обслуживаются) в каналах по показательному закону. Однако в нашем случае для практических инженерных рас-

четов средняя задержка, рассчитанная по упрощенной формуле, дает возможность на этапе решения задачи упаковки оценить требуемую грузоподъемность транспортных средств или пропускную способность линий связи. Фактический выбор грузоподъемности и пропускной способности выполняется при решении задачи маршрутизации, когда ограничения на среднюю задержку могут быть заданы или не заданы. Экспериментируя со средней задержкой при решении задачи маршрутизации в среде АИАС, диспетчер транспортной сети или администратор сети передачи данных может обеспечить требуемый ему резерв грузоподъемности или пропускной способности при прогнозируемых колебаниях величины потоков на заданных промежутках времени. Увеличение резерва приводит к удорожанию сети, но уменьшает вероятность оперативного перераспределения потоков и технического переоснащения оборудования на уровне оперативного управления при возникновении перегрузок в сети. По существу средняя задержка является некоторой количественной мерой загрузки сети. В дальнейшем изложении для единообразия под линией связи будем понимать либо маршрут транспортного средства, либо пучок элементарных каналов связи.

Рассмотрим задачу выбора пропускных способностей в постановке (3.33), (3.26)-(3.27). Требуется найти минимальное значение функции

$$\sum_{kl \in E} C_{tr}^{kl}(w_{kl}, d_{kl}),$$

при ограничениях:

$$w_{kl} > f_{kl},\ w_{kl} \in \{w_1, w_2, ..., w_\alpha\},\ \forall kl \in E,$$

$$t_{av} = 1/U_\Sigma \sum_{kl \in E} f_{kl}/(w_{kl} - f_{kl}) \le T_{\max},\ U_\Sigma = \sum_{ij \in S} u_{ij}.$$

В работе Л. Клейнрока [67] для решения этой задачи было предложено аппроксимировать дискретные стоимости линий связи, например,

$$C_{tr}^{kl}(w_{kl}, d_{kl}) = \begin{cases} C_{kl}^0(d_{kl}) + C_{kl}^1(w_{kl}, d_{kl}), \\ C_{kl}^0(w_{kl}, d_{kl}) + C_{kl}^1(w_{kl}, d_{kl}), \end{cases}$$

где $C_{kl}^0(\cdot)$ и $C_{kl}^1(\cdot)$ — соответственно приведенные эксплуатационные и капита-

льные затраты на линию kl с пропускной способностью $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$ и длиной d_{kl}, непрерывными линейными функциями, например,

$$C_{tr}^{kl}(w_{kl}, d_{kl}) = \begin{cases} c_{kl}^0 + c_{kl}^1 \cdot w_{kl}, \\ c_{kl}^2 \cdot w_{kl}, \end{cases} \tag{3.39}$$

где c_{kl}^0, c_{kl}^1, c_{kl}^2 — полученные коэффициенты аппроксимации. Затем, используя метод неопределенных множителей Лагранжа, аналитически решить релаксированную задачу, получить значения оптимальных пропускных способностей и округлить их до ближайших дискретных значений из заданного ряда $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$. В работе Ю.П. Зайченко [58] предложен конкретный алгоритм решения задачи выбора пропускных способностей линий связи с дискретными стоимостями, основанный на подходе Л. Клейнрока. В алгоритме предполагается, что заданные функции стоимости можно с достаточной степенью адекватности аппроксимировать линейными. Алгоритм позволяет быстро попасть в окрестности точек оптимума при таких допущениях и найти приближенное дискретное решение, используя упорядоченные по невозрастанию отношения известных значений потоков в линиях связи к их удельным дискретным стоимостям в окрестности непрерывных точек оптимума. Получение точного решения в окрестностях точек оптимума требует полного перебора. Этот факт был неоднократно проверен при проведении многочисленных экспериментов на сетях различных размерностей и подтверждает общую экспоненциальную сложность дискретной задачи. Другой алгоритм, основанный на общей схеме метода последовательного анализа и отсеивания вариантов (ПАВ) [86-88] предложен в работах [57, 59]. Алгоритм решает задачу выбора дискретных пропускных способностей линий связи при заданном многопродуктовом потоке и ограничении на среднее время задержки в сети. Его можно применять как к исходной постановке задачи, так и к постановке в виде (3.35)-(3.38). Алгоритм перебирает решения, сужая на каждой итерации область допустимых решений. Алгоритм может использоваться для любых функций стоимости линий связи от пропускных способностей (не обязательно линейных). Это отличает его от апп-

роксимационных алгоритмов Л. Клейнрока и Ю.П. Зайченко. Алгоритм теоретически дает точное решение при полном переборе вариантов на заключительном этапе оптимизации, однако, его временная сложность псевдополиномиальна.

Рассмотрим случай, когда стоимостные функции линейно зависят от пропускной способности. Для аналитического решения задачи с линейной функцией стоимости используется функция Лагранжа [6]

$$L = \sum_{kl \in E} c_{kl}(w_{kl}, d_{kl}) + \frac{\beta}{U_\Sigma} \frac{f_{kl}}{w_{kl} - f_{kl}},$$

где β — множитель Лагранжа. Приравняв частные производные этой функции нулю, получаем $\frac{\partial L}{\partial w_{kl}} = c_{kl}^* - \frac{\beta f_{kl}}{U_\Sigma (w_{kl} - f_{kl})^2} = 0$, где $c_{kl}^* = c_{kl}^1$ или $c_{kl}^* = c_{kl}^2$, откуда $w_{kl} = f_{kl} + \sqrt{\frac{\beta f_{kl}}{U_\Sigma c_{kl}^*}}$. Подставив значения w_{kl} в исходное равенство-ограничение, получим $T_{\max} = 1/U_\Sigma \sum_{kl \in E} f_{kl}/(w_{kl} - f_{kl}) = \sum_{kl \in E} \sqrt{\frac{c_{kl}^* f_{kl}}{\beta U_\Sigma}}$ и $\sqrt{\beta} = 1/T_{\max} \sum_{kl \in E} \sqrt{\frac{c_{kl}^* f_{kl}}{U_\Sigma}}$. Подставляя значение множителя в выражение для w_{kl}, окончательно получаем значения w_{kl}^* для оптимальных пропускных способностей линий связи

$$w_{kl}^* = f_{kl} + \frac{1}{T_{\max}} \sqrt{\frac{f_{kl}}{U_\Sigma c_{kl}^*}} \sum_{rs \in E} \sqrt{\frac{c_{rs}^* f_{rs}}{U_\Sigma}},$$

или после очевидных преобразований

$$w_{kl}^* = f_{kl} + \frac{f_{kl}}{U_\Sigma T_{\max}} \frac{\sum_{rs \in E} \sqrt{c_{rs}^* f_{rs}}}{\sqrt{c_{kl}^* f_{kl}}}. \qquad (3.40)$$

Отметим, что на практике для транспортных сетей и сетей передачи данных пропускная способность, как правило, должна быть одинаковой для прямого kl и обратного lk направлений. Поэтому при практическом решении задачи две ориентированные линии связи kl и lk заменяются на одну неориентированную линию связи $kl, (k < l)$ и выбирается $f_{kl} = \max\{f_{kl}, f_{lk}\}$. Оптимальная стоимость сети определяется как

$$C_{\min} = \sum_{kl \in E} c_{kl}^{*} f_{kl} + \frac{1}{U_{\Sigma} T_{\max}} \left(\sum_{kl \in E} \sqrt{c_{kl}^{*} f_{kl}} \right)^2 . \qquad (3.41)$$

3.4.3.2. Аппроксимационный алгоритм

Приведем схему алгоритма, основанную на подходе «удельных стоимостей», предложенном Ю.П. Зайченко [58]. Идея алгоритма заключается в следующем. Предположим, что для всех линий связи $kl \in E$ известны коэффициенты c_{kl}^0, c_{kl}^1, c_{kl}^2 линейной зависимости (3.39). Такие коэффициенты могут быть получены для каждой линии связи kl длиной d_{kl} линейной аппроксимацией (например, методом наименьших квадратов) для ряда стандартных дискретных пропускных способностей $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$, одинаковых для всех линий связи и разных коэффициентов $C_0^1 \le C_0^2 \le ... \le C_0^\alpha$ и $C_1^1 \le C_1^2 \le ... \le C_1^\alpha$, соответствующих пропускным способностям $w_1 \le w_2 \le ... \le w_\alpha$. Зная коэффициенты $c_{kl}^{*} = c_{kl}^1 = tg\varphi$ или $c_{kl}^{*} = c_{kl}^2 = tg\varphi$ (рис. 3.6) и значения f_{kl}, по формуле (3.40) можно найти пропускные способности w_{kl}^{*}. Далее в окрестности непрерывных оптимумов w_{kl}^{*} по определенной процедуре выбираем подходящие значения пропускных способностей из дискретного ряда. Заметим, что если ряд $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$ задан с единичным шагом и все $w_{kl} = f_{kl} + 1$, то имеется возможность предварительной оценки верхней границы величины $t_{av} \le \sum_{kl \in E} f_{kl} / U_{\Sigma}$.

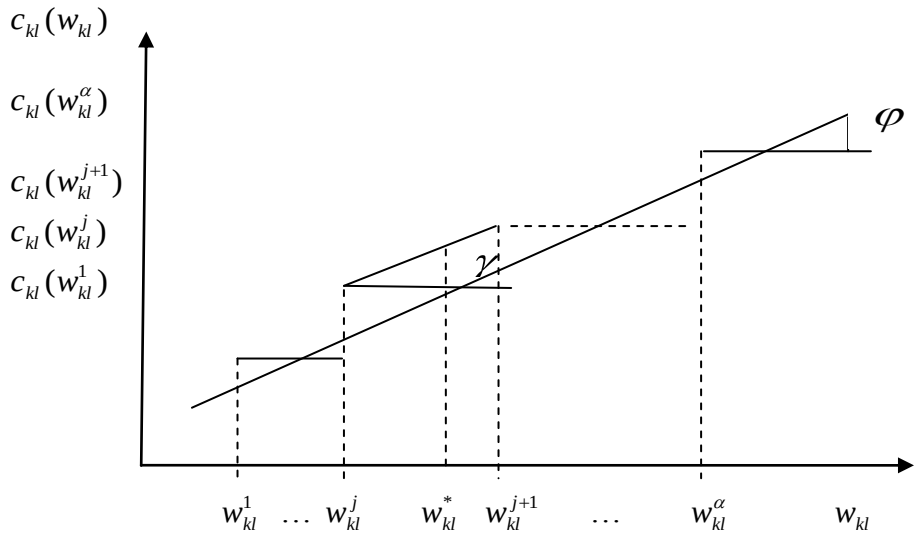


Рис. 3.6

Алгоритм 3.6. А

1. Для каждой линии связи методом наименьших квадратов определяем коэффициенты c_{kl}^{*}. Вычисляем w_{kl}^{*}, $\forall kl \in E$ согласно (3.40).

2. Из $w_{kl} \in \{w_1, w_2, ..., w_\alpha\}$ для каждой линии связи $kl \in E$ выбираем ближайшие к w_{kl}^{*} допустимые значения w_{kl}^{j}, $j = \overline{1,\alpha}$, такие, что $f_{kl} < w_{kl}^{j} \le w_{kl}^{*}$. Если $w_{kl}^{*} > w_\alpha$, то набор $\{w_1, w_2, ..., w_\alpha\}$ необходимо расширить до значения $w_\alpha = \lceil w_{kl}^{*} \rceil$, где $\lceil \cdot \rceil$ — знаки округления до большего целого. В случае если $w_{kl}^{j} \le f_{kl}$, то в качестве w_{kl}^{j} выбираем ближайшее большее значение $w_{kl}^{j} > f_{kl}$ (может оказаться, что $w_{kl}^{j} \ge w_{kl}^{*}$). Если $f_{kl} = 0$, то принимается, что линии связи kl не существует.

3. Линеаризуем зависимость $c_{kl}(w_{kl})$ в окрестности значения w_{kl}^{j} и находим величину удельной стоимости линии связи $c_{kl}^{*} = \dfrac{c_{kl}(w_{kl}^{j+1}) - c_{kl}(w_{kl}^{j})}{w_{kl}^{j+1} - w_{kl}^{j}}(w_{kl}^{*} - w_{kl}^{j}) = tg\gamma(w_{kl}^{*} - w_{kl}^{j})$, $j \in \{1,...,\alpha - 1\}$ в точке w_{kl}^{*} (рис. 3.6).

4. Упорядочим все линии связи $kl \in E$ по невозрастанию величин f_{kl} / c_{kl}^{*} и получим множество $E^{*} = \{(k,l)_1, (k,l)_2, ..., (k,l)_e\}$. Установим начальное значение счетчика линий связи $i = 0$.

5. Полагаем $i \leftarrow i + 1$. Если $i \le e$, то выбираем линию связи $(k,l)_i$ из множества E^{*} и переходим к шагу 6. Иначе переходим к шагу 7.

6. Увеличиваем пропускную способность $w_{(k,l)_i}^{j}$ линии связи $(k,l)_i$ до ближайшего большего значения из дискретного ряда $\{w_1, w_2, ..., w_\alpha\}$, т.е. выбираем $w_{(k,l)_i}^{j+1}$ такую, что $w_{(k,l)_i}^{j+1} > w_{kl}^{*} \ge w_{(k,l)_i}^{j}$, $j \in \{1,...,\alpha - 1\}$. Пересчитываем значение t_{av} с учетом увеличения пропускной способности линии связи $(k,l)_i$. Если $t_{av} \le T_{\max}$, то переходим к шагу 7. Иначе переходим к шагу 5.

7. Конец работы алгоритма. Найденные значения $w_{(k,l)_i}^{j}$, $j \in \{1,...,\alpha\}$, $i \in \{1,...,e\}$, являются приближенным решением задачи.

Временная сложность алгоритма составляет $O(e\alpha^2 + e\alpha + e\log e + K_1 e)$, K_1

— константа, и в основном определяется временной сложностью аппроксимации стоимостей линий связи линейными функциями и выбором алгоритма сортировки.

3.4.3.3. Алгоритм, основанный на методе последовательного анализа и отсеивания вариантов

Метод последовательного анализа вариантов (ПАВ) впервые был предложен В.С. Михалевичем и Н.З. Шором [86] (см. также работы [38, 89, 90]). Приведем общую схему метода для решения задачи выбора пропускных способностей (3.33), (3.26)-(3.27). Метод ПАВ состоит из процедур отсеивания W_1 и W_2.

Алгоритм 3.7. ПАВ

Процедура W_1 — отсеивание по ограничению

1. Пусть λ, $\lambda = 1,2,...$ означает номер подмножества выбранных пропускных способностей для $\forall kl \in E$, а m, $m = 1,2,...,\alpha$ указывает на индекс упорядоченных по возрастанию дискретных пропускных способностей из заданного ряда $\{w_1, w_2, ..., w_\alpha\}$. Определим диапазон возможных значений пропускных способностей $W_{kl}(\lambda) = [\min w_{kl}^m, \max w_{kl}^m] \in \{w_1, w_2, ..., w_\alpha\}$ по каждой линии связи $kl \in E$, используя неравенство $t_{av} \le T_{\max}$. При этом должно выполняться $\min w_{kl}^m \ge w_1$, $\max w_{kl}^m \le w_\alpha$ и хотя бы одно из этих неравенств — строгое. Из множества $\{w_1, w_2, ..., w_\alpha\}$ отсеиваются те значения, которые заведомо не удовлетворяют условию $t_{av} \le T_{\max}$. Первоначально для $\lambda = 1$ и $\forall kl \in E$ выбираем $\min w_{kl}^1 = w_1$, $\max w_{kl}^\alpha = w_\alpha$, т.е. $W_{kl}(1) = [\min w_{kl}^1, \max w_{kl}^\alpha]$.

2. Для отсеивания будем использовать следующее условие

$$a = \frac{f_{kl}}{U_\Sigma (\min w_{kl}^m - f_{kl})} > b = T_{\max} - \frac{1}{U_\Sigma} \sum_{rs \in E, rs \ne kl} \frac{f_{rs}}{\max w_{rs}^{m-1} - f_{rs}}. \qquad (3.42)$$

Последовательно просматриваем слева-направо заданный набор пропускных способностей $\{w_1, w_2, ..., w_\alpha\}$ и выбираем $\min w_{kl}^m = w_1, w_2, ... > f_{kl}$ до тех пор, пока выполняется (3.42). Пусть условие (3.42) выполняется включительно до

$\min w_{kl}^m = w_\beta$ и не выполняется при $\min w_{kl}^m = w_{\beta+1}$, тогда для линии связи kl отсеиваются все $w_\xi \le w_\beta$, $\xi = \overline{1,\beta}$ (см. рис. 3.7) и $\min w_{kl}^{\beta+1} = w_{\beta+1}$. Продолжая эту процедуру для всех линий связи получаем новые границы изменения пропускных способностей $W_{kl}(2) = [\min w_{kl}^{\beta+1}, \max w_{kl}^{\alpha}]$, $\forall kl \in E$. Переходим к процедуре W_2.

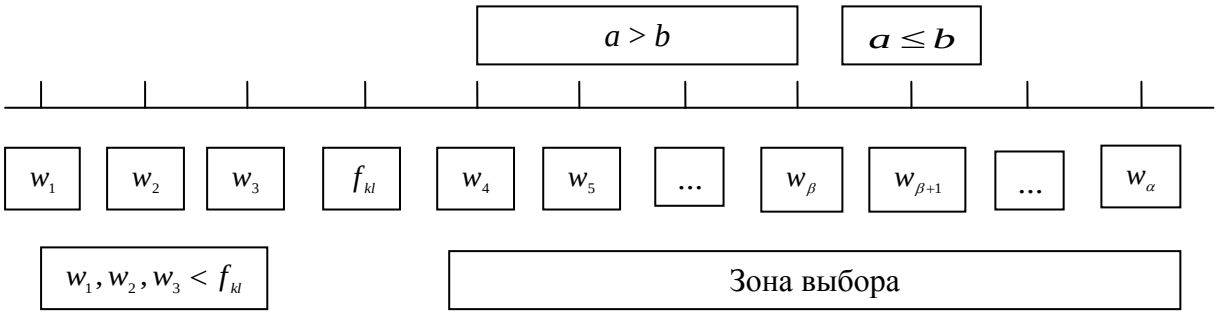


Рис. 3.7

Процедура W_2 — отсеивание по значению целевой функции

1. Задаем начальный порог $C_1^* = \frac{1}{2}(C_{\min} + C_{\max})$, где $C_{\min} = \sum_{kl \in E} c_{kl}(\min w_{kl}^m)$, $C_{\max} = \sum_{kl \in E} c_{kl}(\max w_{kl}^m)$, $m = 1,2,\ldots,\alpha$.

2. Для отсеивания допустимых вариантов используем условие

$$a = c_{kl}(\max w_{kl}^m) > b = C_1^* - \sum_{rs \in E, rs \neq kl} c_{rs}(\min w_{kl}^m) \qquad (3.43)$$

Последовательно просматриваем справа-налево заданный набор пропускных способностей $\{w_1, w_2, \ldots, w_\alpha\}$ и выбираем $\max w_{kl}^m = w_\alpha, \ldots, w_2, w_1 > f_{kl}$ до тех пор, пока выполняется (3.43). Пусть условие (3.43) выполняется включительно до $\max w_{kl}^m = w_\eta$ и не выполняется при $\max w_{kl}^m = w_{\eta-1}$, тогда для линии связи kl отсеиваются все $w_\xi \ge w_\eta$, $\xi = \overline{\eta,\alpha}$ (см. рис. 3.8) и $\max w_{kl}^{\eta-1} = w_{\eta-1}$.

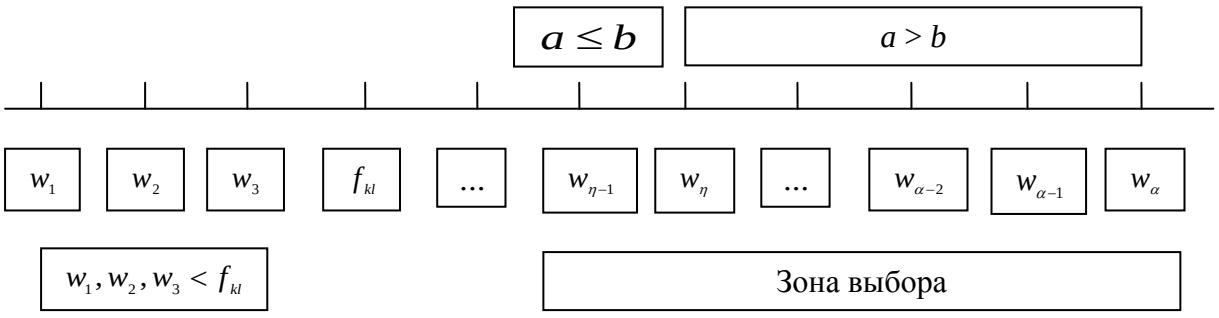


Рис. 3.8

После выполнения процедуры отсеивания вариантов для всех линий связи по условию (3.43) могут быть следующие случаи.

Случай 1. Если отсеивание произошло хотя бы для одной линии связи, то получаем новые границы изменения пропускных способностей $W_{kl}(3)=[\min w_{kl}^{\beta+1}, \max w_{kl}^{\eta-1}],\ \forall kl \in E$. Переходим к процедуре W_1.

Случай 2. Отсеивания нет. Сужаем область поиска, устанавливая новый порог $C_1^* \leftarrow \frac{1}{2}(C_1^* + C_{\min})$, и переходим на продолжение отсеивания по условию (3.43).

Случай 3. Отсеялись все варианты пропускных способностей. Расширяем область поиска, устанавливая новый порог $C_1^* \leftarrow \frac{1}{2}(C_1^* + C_{\max})$, и переходим на продолжение отсеивания по условию (3.43).

Повторяем процедуры W_1 и W_2 до тех пор, пока для $\forall kl \in E$ границы изменения пропускных способностей $W_{kl}(g)=[\min w_{kl}^{\beta}, \max w_{kl}^{\eta}]$ не сократятся до двух значений, т.е. когда $|W_{kl}(g)| = |[\min w_{kl}^{\beta}, \max w_{kl}^{\eta}]| = 2$, при $\eta - \beta = 1$ и для каждой линии связи можно выбрать либо $\min w_{kl}^{\beta}$, либо $\max w_{kl}^{\eta}$. Далее выбор оптимального решения можно выполнить простым перебором, но, как уже отмечалось выше, сложность такого перебора экспоненциальная и оценивается в $O(2^e)$. Для получения приближенного решения в алгоритме ПАВ может использоваться такая же, как и в алгоритме А, процедура упорядочения линий связи во множестве $E^* = \{(k,l)_1, (k,l)_2, ..., (k,l)_e\}$ по невозрастанию величин f_{kl}/c_{kl}^*. При этом удельные стоимости c_{kl}^* определяются по формуле $c_{kl}^* = \frac{c_{kl}(\max w_{kl}^{\eta}) - c_{kl}(\min w_{kl}^{\beta})}{\max w_{kl}^{\eta} - \min w_{kl}^{\beta}},\ \forall kl \in E$.

Сходимость алгоритма ПАВ основана на конечности метода [86], а его временная сложность без полного перебора на заключительном этапе зависит от количества итераций по λ, числа K сужений и расширений области допустимых решений, выбора алгоритма сортировки линий связи и составляет

$O[\lambda(e(e+\alpha)+Ke(e+\alpha))+e\log e+K_1 e]$, где K_1 — некоторая константа.

3.4.3.4. Экспериментальное сравнение алгоритмов

Было проведено сравнение приближенных решений, полученных алгоритмами А и ПАВ (без полного перебора допустимых решений на заключительном этапе оптимизации) для линейных и нелинейных функций затрат. Эксперимент проводился для сетей с размерностью n = 20, 40, 60 и 80 и одинаковой степенью узлов val = 3. Длины линий связи и потоки задавались датчиком псевдослучайных чисел и варьировались для всех размерностей сети в пределах от 50 до 100 км, и от 1 до 5 транспортных блоков. Пропускные способности линий связи задавались набором {5, 10, 15, 20,…}. Для линейных функций коэффициенты $C_0^1 \le C_0^2 \le \ldots \le C_0^\alpha$ и $C_1^1 \le C_1^2 \le \ldots \le C_1^\alpha$, соответствующие пропускным способностям {5, 10, 15, 20,…}, выбирались из наборов {10, 20, 30, 40,…} и {20, 30, 40, 50,…}. Для нелинейных функций значения C_0^1 = 10, C_1^1 = 20, а значения последующих C_0^i и C_1^i, $i = 2,3,\ldots$ возрастали на случайную величину из интервала [5, 10].

В табл. 3.6 показаны результаты решения задачи при ограничении на среднюю задержку $T_{\max}$ = 0,05. Как и следовало ожидать, для линейных функций аппроксимационный алгоритм для всех размерностей сети показывает лучшие результаты. Значения целевой функции, полученные алгоритмом ПАВ, отличаются от «оптимальных» на величину от 1,75% до 3,24%. Для нелинейных функций при n = 40 и n = 60 алгоритмом ПАВ получены лучшие значения целевой функции, а отклонения «оптимумов» составляют от 0,15% до 5,18%.

Таблица 3.6

Алгоритм		Значение целевой функции (в усл. ед.) при			
		n = 20	n = 40	n = 60	n = 80
Линейные функции	А	883160	3784510	9433900	17832150
	ПАВ	898640	3907000	9670130	18344670
Нелинейные функции	А	623244,69	2839533,5	6657285,0	12644782,0
	ПАВ	655501,81	2816475,5	6647623,5	12792124,0

Оба алгоритма без полного перебора вариантов на заключительном этапе не дают точного решения.

На практике, априори трудно судить о достаточной точности аппроксимации дискретных стоимостных функций линий связи линейными и искать решение, отталкиваясь от оптимальных пропускных способностей дуг для линейных функций по Л. Клейнроку. Поэтому, как показали экспериментальные исследования, при выборе алгоритма решения дискретной задачи все таки следует отдать предпочтение алгоритму ПАВ, так как он позволяет работать с произвольными функциями затрат и вводить в общую схему оптимизации вероятностно-временные оценки задержки потоков не только в линиях связи, но и в узлах сети, а также учитывать возможные дополнительные ограничения [56].

В заключение отметим, что для классической задачи о ранце и задачи о ранце с мультивыбором (задачи с булевыми переменными на минимум) известны полностью полиномиальные приближенные схемы решения (FPTAS — Fully Polynomial Time Approximation Scheme) [337, 378]. Это означает, что для них существуют алгоритмы, которые за полиномиальное время от размера входа задач и $1/\varepsilon$, позволяют получить гарантированное приближенное решение с ошибкой не более чем $(1+\varepsilon)$, где ε — сколь угодно малое положительное число. Поэтому на заключительном этапе решения задачи в обоих алгоритмах для получения гарантированного ε-приближенного решения возможно применение алгоритмов, описанных в работах [337, 378]. Эти алгоритмы также можно использовать непосредственно для решения задачи в постановке (3.35)-(3.38).

3.5. Задача выбора структуры сети

В подразделе рассматривается и экспериментально исследуется задача выбора иерархической структуры магистральной сети, состоящей из нескольких типов узлов, отличающихся технологией обработки и распределения мелкопартионных потоков корреспонденций. Предложена общая схема эвристического алгоритма решения задачи, основанная на решении задач упаковки и распределения потоков в зональных сетях [13, 122].

3.5.1. Постановка и общая схема решения задачи

Пусть $G(N,E)$ — иерархическая магистральная сеть с множеством узлов N, $n=\left| N \right|$, $N=N_1 \cup N_2 \cup N_3$, где N_1, N_2, N_3 — множества узлов первого, второго и третьего типа соответственно и множеством неориентированных дуг E, $e=\left| E \right|$. Предполагается, что географические координаты размещения узлов сети известны. Задана целочисленная матрица $A=\left\| a_{ij} \right\|_{n\times n}$ мелкопартионных потоков корреспонденций между всеми узлами сети. Так же как и в предыдущем подразделе, потоки a_{ii}, $i=\overline{1,n}$ в матрице A представляют внутренние потоки между узлами четвертого типа в зоне обслуживания i-го узла.

Требуется определить количественный и качественный состав узлов сети и схему сортировки мелкопартионных корреспонденций в каждом узле, при которых максимально снижаются приведенные затраты на функционирование сети. Формально, необходимо для всех возможных комбинаций типов узлов решить задачу упаковки из подраздела 3.2, т.е. найти минимум функции (3.18) при ограничениях (3.19)-(3.21) и ограничениях на время доставки $t_{ij} \le T_{ij}$ $\forall ij \in S$ и число транзитных объединений $\nu_{ij} \le \nu_{\max}$ $\forall ij \in S$ мелкопартионных корреспонденций при их транспортировке из узлов отправления в узлы назначения. Повторим запись задачи. Найти

$$\min \sum_{ij\in S} C_{tr}^{ij}(u_{ij}, d_{ij}) + \sum_{i=1}^{n} C_{sort}^{i}(x_i, q_i) + \sum_{i=1}^{n} C_{load}^{i}(u_i),$$

при

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1,\, j\neq i}^{n} a_{ij} = \sum_{j=1}^{n} x_{ji} - \sum_{j=1,\, j\neq i}^{n} a_{ji},\ i=\overline{1,n},$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1,\, j\neq i}^{n} a_{ij} \le h_i,\ i=\overline{1,n},$$

$$x_{ij}, u_{ij} \ge 0 \text{ и целые, } \forall ij \in S.$$

Интуитивно ясно, что для выбора оптимальной структуры сети нужно в процессе решения задачи каким-то образом генерировать различные варианты множеств N_1, N_2, N_3, сравнивать значения функции цели для каждого варианта

и таким образом определить искомый минимум. Комбинаторное число вариантов состава множеств N_1, N_2, N_3 растет экспоненциально, поэтому полный перебор вариантов решения просто немыслим. Одним из главных факторов, влияющих на выбор претендентов на узлы второго и третьего типа, является начальный суммарный поток $a_i = a_{ii} + \sum_{j=1, j\neq i}^{n}(a_{ij} + a_{ji})$ в узлах сети, когда $y_i = \sum_{j=1}^{n} x_{ij} - \sum_{j=1, j\neq i}^{n} a_{ij} = 0,\ i = \overline{1,n}$. Очевидно, что чем он меньше, тем у соответствующего узла больше шансов попасть во множество $N_2 \cup N_3$. Напомним, что в узлах второго и третьего типа запрещена сортировка транзитных мелкопартионных корреспонденций и эти узлы отличаются друг от друга только тем, что в узлах второго типа разрешена обработка транзитных транспортных блоков, а в узлах третьего типа — нет. Примем, что изначально во множество $N_2 \cup N_3$ условно выбираются узлы третьего типа. Если в процессе решения задачи будет обнаружено, что в отдельных введенных узлах третьего типа возможно появление транзитных потоков транспортных блоков, то они получают статус узлов второго типа. Другими критериями отбора кандидатов в узлы множества $N_2 \cup N_3$ могут быть данные об отсутствии подъездных путей, площадей для установки сортировочного оборудования и контейнерных терминалов (для транспортных сетей), сетевого оборудования и современных ПЭВМ, квалифицированного обслуживающего персонала и другие трудно-формализуемые факторы. Все такие данные должны присутствовать в базе знаний информационной системы поддержки принятия решений и быть доступными для диспетчеров транспортных систем и администраторов сетей передачи данных.

Резюмируя вышесказанное, можно определить общую стратегию эвристического алгоритма оптимизации структуры сети:

1) Начальная структура сети определяется диспетчерами и администраторами на основании опыта, практических знаний и данных в базе знаний, формируется множество $N_{канд}$ кандидатов (претендентов) в $N_2 \cup N_3$. Всем узлам из $N_{канд}$ присваивается статус узлов третьего типа;

2) Начальные суммарные потоки a_i, $i = \overline{1,n}$ в узлах сети упорядочиваются по неубыванию их величины. Формируется соответствующий список номеров узлов, причем в начале списка находятся только узлы из множества $N_{канд}$. Далее, после последнего узла из $N_{канд}$, следуют узлы первого типа. К множеству узлов из $N_{канд}$ может быть добавлено указанное число узлов первого типа, расположенных после последнего узла из $N_{канд}$ и также являющихся потенциальными кандидатами в $N_2 \cup N_3$;

3) Алгоритм последовательно, с накоплением, выбирает из списка узлы, включает их во множество $N_2 \cup N_3$ и решает задачу (3.18)-(3.21). В процессе решения задачи и исчерпании списка кандидатов в $N_{канд}$, определяется структура сети с минимальным значением функции цели (3.18). После решения задачи множество $N_2 \cup N_3$ может оказаться пустым, т.е. все узлы сети должны быть узлами первого типа. Если по каким-либо причинам множество $N_{канд}$ не было указано, то алгоритм должен сам упорядочить все узлы сети в списке по неубыванию их суммарных потоков, запросить верхнюю границу числа узлов из множества $N_{канд}$ и перейти в режим оптимизации.

Как видно из описания общей схемы алгоритма выбора оптимальной структуры сети, для каждого вновь включаемого узла из $N_{канд}$ во множество $N_2 \cup N_3$ необходимо решать задачу (3.18)-(3.21) и задачу распределения исходящих потоков из узлов второго и третьего типа, что может потребовать значительного времени вычислений.

3.5.2. Экспериментальные исследования решения задачи

Перейдем к описанию и анализу результатов численных экспериментов решения задачи выбора структуры сети. Для проведения экспериментов была выбрана сгенерированная датчиком псевдослучайных чисел транспортная сеть с числом узлов $n = 30$ и числом исходящих дуг из каждого узла (степень узлов) $val = 5$. Длины дуг d_{kl}, $kl \in E$ генерировались в пределах от 80 до 300 км; объ-

ем транспортного блока ω принимался равным 40 единицам мелкопартионных грузов; грузоподъемность транспортных средств W (параметр w_ξ) выбиралась из набора $W \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 65\}$ транспортных блоков; время на обработку (сортировку) грузов в узлах сети t_a = 1 сутки; время на транзитную перегрузку транспортных блоков в узлах сети t_b = 0,5 суток; средняя скорость движения транспортных средств V_{av} = 80 км/час; пропускные способности узлов по обработке транзита h_i, $i = \overline{1,n}$ принимались равными 800 единиц грузов в сутки для всех узлов сети; максимальное время доставки грузов получателю T_{ij} = 15 суток для всех $ij \in S$; максимальное число объединений корреспонденций $\nu_{\max}$ не ограничивалось; средняя задержка транспортных блоков в сети t_{av} рассчитывалась, а их максимальная задержка $T_{\max}$ не ограничивалась. Исходящие потоки из узлов сети также задавались датчиком псевдослучайных чисел для трех интенсивностей роста суммарных потоков в узлах: $A1$ — слабый рост, $A2$ — средний рост, $A3$ — более крутой рост (см. рис. 3.9).

В качестве функций затрат использовался конкретный вид формул (3.34, 3.31) в условных единицах стоимости:

$$C_{tr}^{ij}(u_{ij}, d_{ij}) = \frac{u_{ij}}{W}\left[\frac{13939{,}2(T_{end} + 2d_{ij}/V_{av})}{T_{move}} + 7200 + 116{,}8d_{ij}\right],\ \forall ij \in S,$$ где T_{end} и T_{move} — соответственно время стоянки транспортных средств в конечных пунктах следования и периодичность их движения;

$$C_{sort}^{i}(x_i, q_i) = 16{,}8892975x_i \exp\left(-\frac{17{,}74x_i}{1+q_i}\right),\ i = \overline{1,n};$$

$$C_{load}^{i}(u_i) = \sqrt{25595u_i^2 + 2629u_i},\ i = \overline{1,n}.$$

Для расчета времени доставки грузов получателю применялись формулы (3.24).

Начальная структура сети генерировалась так, чтобы каждый узел третьего типа был связан в своей зоне обслуживания только со смежными узлами первого типа. Это ограничение позволяет для любой сети с заданными параметрами

n и val сгенерировать строго определенное число узлов третьего типа и придать структуре сети более правдоподобный вид.

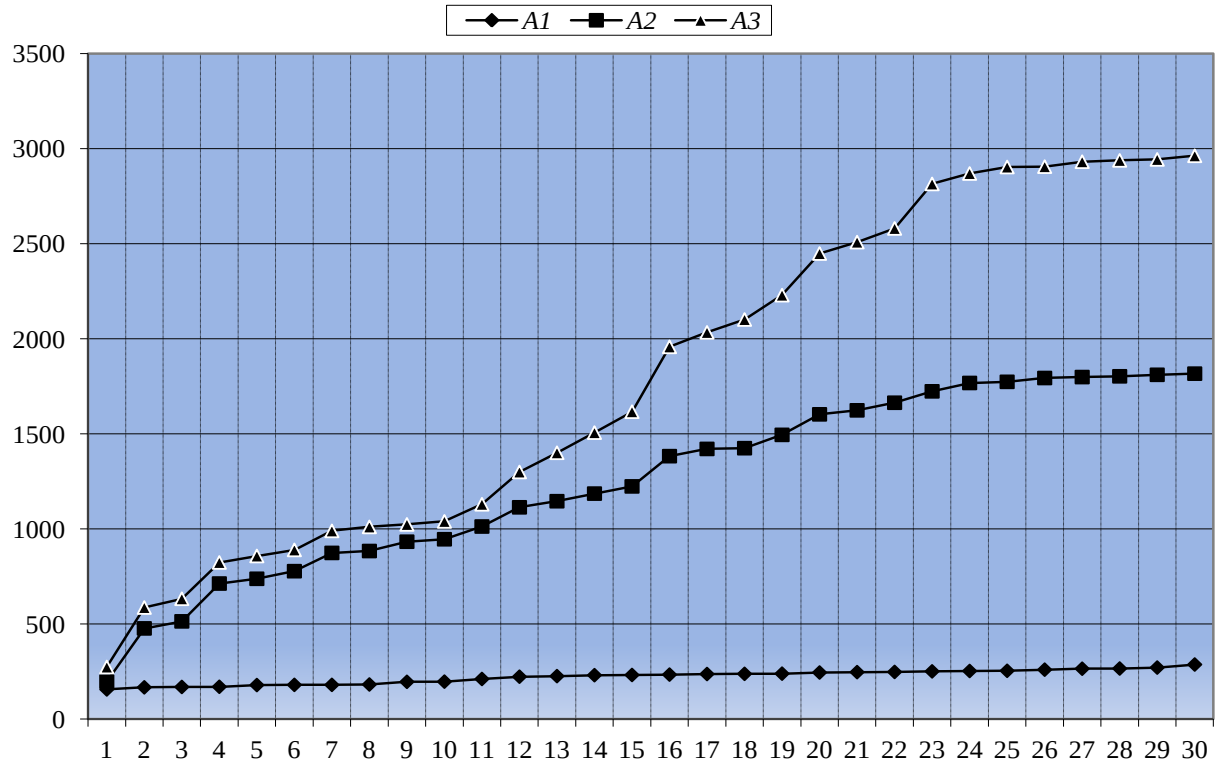
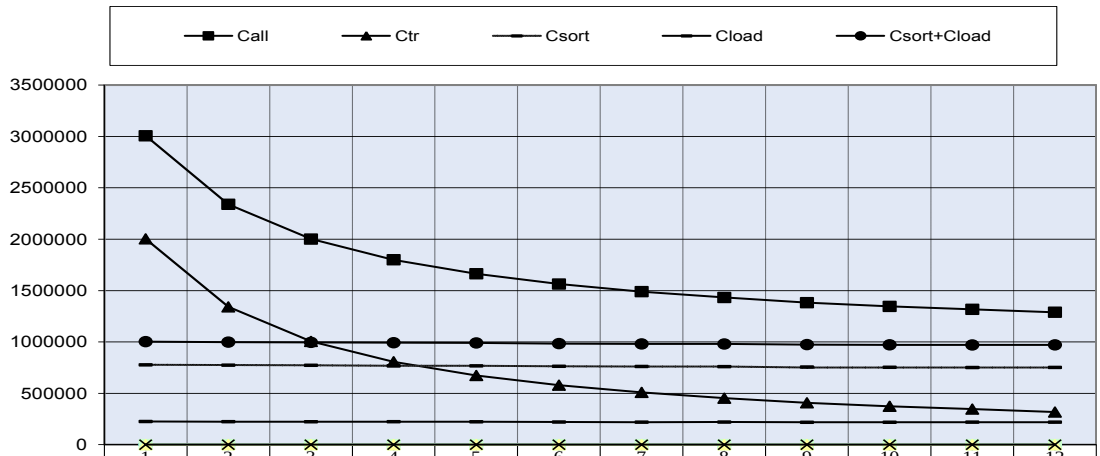


Рис. 3.9. Графики потоков $A1$, $A2$, $A3$. По горизонтали показаны точки, соответствующие номерам узлов сети, упорядоченных по возрастанию их суммарных потоков, а по вертикали — величина потоков

Для сети с $n = 30$ и $val = 5$ было сгенерировано 10 узлов третьего типа. Параметры датчика псевдослучайных чисел для генерации исходящих потоков из узлов третьего и первого типа подбирались так, чтобы в начале списка кандидатов $N_{канд}$ оказались только узлы третьего типа, а далее шли узлы первого типа, причем на границе разрыва узлов разного типа не было бы резких скачков в сторону уменьшения или увеличения суммарных потоков в узлах сети. Для всех вариантов решения задачи выбора структуры сети к начальному множеству $N_{канд}$ узлов третьего типа добавлялось 6 узлов первого типа, расположенных непосредственно за последним узлом третьего типа в списке кандидатов.

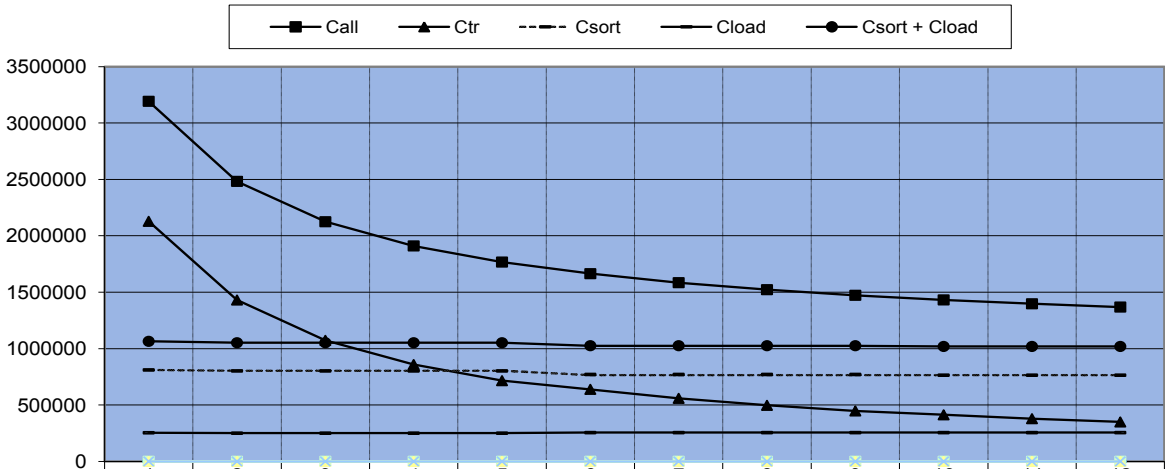
Для практического доказательства утверждения о том, что выбор структуры сети слабо зависит от пропускных способностей ее дуг, была решена задача

упаковки (3.18)-(3.21) для значений $W \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 65\}$ и суммарных потоков узлов $A2$. Задача решалась с помощью алгоритмов Optpac1 и Optpac5, при этом для каждого W фиксировались значения следующих параметров: C_{all} — полные затраты или функция цели; C_{tr} — транспортные затраты; C_{sort} — затраты на сортировку; C_{load} — грузовые затраты на погрузку-выгрузку транспортных блоков; $C_{sort} + C_{load}$ — сумма сортировочных и грузовых затрат; $N3$ — число узлов второго и третьего типа в оптимизированной структуре сети; t_{av} — среднее время задержки в сети; FS — число объединений потоков корреспонденций при котором получено наилучшее решение; K_{av} — средний коэффициент загрузки транспортного блока; $U_{\min}$ — количество транспортных блоков, необходимых для упаковки и транспортировки всех потоков в сети; N_{av} — среднее число магистральных направлений сортировки потоков в узлах сети; t_{sr} — среднее время доставки грузов получателю.



W	10	15	20	25	30	35	40	45	50	55	60	65
Call	3006240	2339650	2001100	1800020	1663750	1563520	1489720	1432650	1382130	1345840	1317500	1289280
Ctr	2003310	1341320	1005100	807566	673630	579648	509033	452625	407855	373885	346789	317896
Csort	777107	774109	772741	768548	767175	763169	761260	759640	756134	753491	750968	751961
Cload	225824	224224	223265	223905	222945	220705	219425	220385	218145	218465	219745	219425
Csort+Cload	1002931	998333	996006	992453	990120	983874	980685	980025	974279	971956	970713	971386
N3	2	2	2	2	2	1	0	1	0	0	0	1
tav	0	0	0	0	0,14645	0,09653	0,08119	0,0654	0,05607	0,05009	0,04478	0,03985
FS	4	5	4	3	6	2	6	4	2	2	5	5
Kav	0,823	0,821	0,822	0,809	0,808	0,808	0,821	0,802	0,813	0,804	0,797	0,792
Umin	705	700	697	699	696	689	685	688	681	682	686	685
Nav	18	18	18	18	18	18	19	18	18	19	19	19
tsr	3,14067	3,16285	3,15749	3,17427	3,15932	3,20034	3,2742	3,20308	3,25391	3,2765	3,26793	3,20916

Рис. 3.10. Результаты решения задачи выбора структуры сети алгоритмом Optpac1



	1	2	3	4	5	6	7	8	9	10	11	12
W	10	15	20	25	30	35	40	45	50	55	60	65
Call	3192730	2482660	2124970	1910360	1767290	1664320	1584480	1522380	1472700	1431760	1397370	1368260
Ctr	2128310	1430730	1073050	858441	715367	638710	558871	496775	447097	412767	378370	349265
Csort	810758	802100	802100	802100	802100	770665	770665	770665	770665	764054	764054	764054
Cload	253662	249822	249822	249822	249822	254941	254941	254941	254941	254941	254941	254941
Csort + Cload	1064420	1051922	1051922	1051922	1051922	1025606	1025606	1025606	1025606	1018995	1018995	1018995
N3	10	8	8	8	8	10	10	10	10	9	9	9
tav	0	0	0	0,21003	0,1482	0,08921	0,06997	0,05799	0,04967	0,04557	0,04046	0,03641
FS	7	1	1	1	1	0	0	0	0	0	0	0
Kav	0,811	0,811	0,811	0,811	0,811	0,732	0,732	0,732	0,732	0,713	0,713	0,713
Umin	792	780	780	780	780	796	796	796	796	796	796	796
Nav	12	12	12	12	12	15	15	15	15	16	16	16
tsr	3,25226	3,22618	3,22618	3,22618	3,22618	3,03609	3,03609	3,03609	3,03609	3,01425	3,01425	3,01425

Рис. 3.11. Результаты решения задачи выбора структуры сети алгоритмом Optpac5

Как видно из графиков на рис. 3.10 и 3.11, при увеличении пропускной способности W всех дуг от 10 до 65, полные и транспортные затраты уменьшаются, а затраты на сортировку, грузовые затраты и их сумма изменяются незначительно. Для алгоритма Optpac1 отклонение значений параметра $C_{sort}+C_{load}$ от минимального = 970713 при W = 60 до максимального = 1002931 при W = 10 составляет 32218 или 3,32 %. Для алгоритма Optpac5 отклонение значений этого параметра от минимального = 1018995 при W = 55 до максимального = 1064420 при W = 10 составляет 45425 или 4,46 %. Минимальные значения параметра $C_{sort}+C_{load}$ для алгоритмов Optpac1 и Optpac5 отличаются на 48282 условных единиц стоимости или на 4,97 %.

Для алгоритма Optpac1 при увеличении значений W от 10 до 30 в структуру сети было введено два узла второго и третьего типа, при дальнейшем увели-

чении W наблюдалось уменьшение числа таких узлов с двух до одного и нуля. Для алгоритма Optpac5 число вводимых в структуру сети узлов второго и третьего типа с увеличением W изменялось в пределах от восьми до десяти. Такое поведение алгоритмов можно объяснить тем, что первый в процессе своей работы использует функции затрат, а второй — нет. При увеличении W до 30 в алгоритме Optpac1 с точки 5 (см. рис. 3.10) транспортные затраты C_{tr} становятся меньше затрат на сортировку C_{sort} и дальнейшее введение узлов второго и третьего типа в структуру сети препятствуется возрастанием удельного веса сортировочных затрат по сравнению с транспортными в значении полных затрат. Алгоритм Optpac5 в процессе работы стремится максимально уменьшить число транспортных блоков $U_{\min}$ и ввести в структуру сети как можно больше узлов второго и третьего типа, а оценивает затраты только после решения задачи. (Как было показано в подразделе 3.3, различные алгоритмы упаковки дают решения, отличающиеся между собой не более чем на 2.65 %.) В точках минимальных значений параметра $C_{sort}+C_{load}$, полные затраты C_{all} составляют 1317500 и 1431760 условных единиц стоимости соответственно для алгоритмов Optpac1 и Optpac5 и отличаются на 8,67 %.

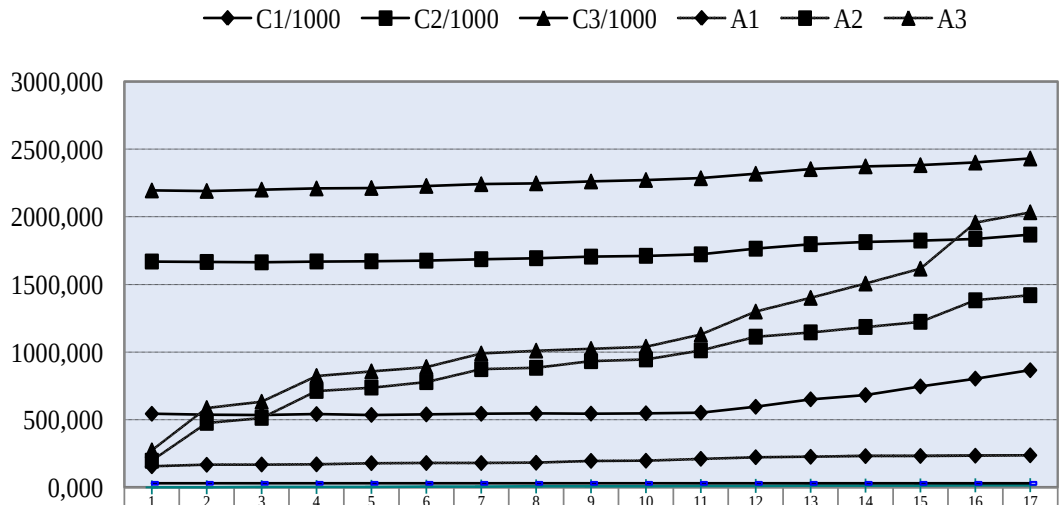
Важным параметром, который нужно учитывать при выборе окончательной структуры сети, является среднее время задержки потоков t_{av}. Параметр t_{av} принимает значение infinit (на рисунках infinit обозначено нулем), когда суммарной пропускной способности дуг сети, заданной параметром W, недостаточно для транспортировки всех транспортных блоков из источников в стоки. Увеличивая значение W, можно предварительно оценить требуемую суммарную пропускную способность дуг, при которой начнут выполняться условия (3.26), (3.27). Поэтому из всех вариантов решения целесообразно отдать предпочтение варианту, при котором t_{av} принимает первое значение, отличное от infinit. Для нашего случая, это значения t_{av} = 0,14645 при W = 30 и $N3$ = 2 и t_{av} = 0,21003 при W = 25 и $N3$ = 8 соответственно для алгоритмов Optpac1 и Optpac5.

Для удобства сравнения решений задачи выбора структуры сети различ-

ными алгоритмами в зависимости от интенсивности роста суммарных потоков узлов — $A1$, $A2$, $A3$ при одинаковой пропускной способности дуг W, примем для алгоритма Optpac5 вариант t_{av} = 0,1482 при W = 30 и $N3$ = 8.

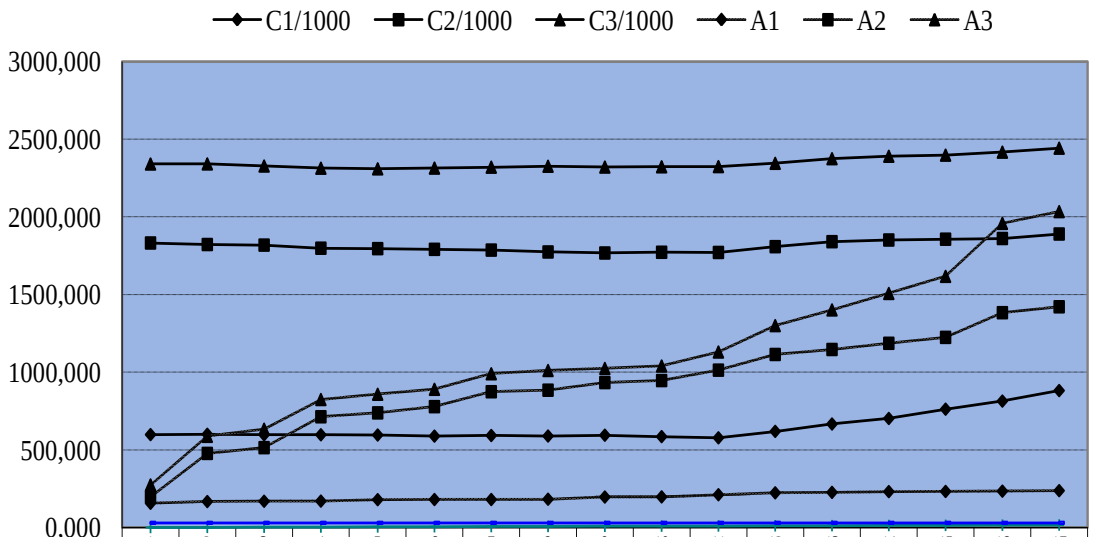
Итак, принимаем структуру сети для алгоритмов Optpac1 и Optpac5, которая была получена при значениях всех параметров в точке 5, когда транспортные затраты становились меньше затрат на сортировку (см. рис. 3.10 и 3.11). В точке 5 для этих алгоритмов значения параметра $C_{sort}+C_{load}$ отличаются на 61802 условных единиц стоимости или на 6,24 %, а значения параметра C_{all} на 103540 единиц или на 6,22 %.

Рассмотрим результаты решения задачи выбора структуры сети, полученные при фиксированной пропускной способности дуг W = 30 и различных интенсивностях роста суммарных потоков узлов $A1$, $A2$, $A3$. На рис. 3.12 и 3.13 приведены графики изменения полных затрат $C1$, $C2$, $C3$ в зависимости от числа узлов второго и третьего типа, включаемых в структуру сети и интенсивности потоков $A1$, $A2$, $A3$ соответственно для алгоритмов Optpac1 и Optpac5.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C1/1000	544,043	537,887	534,570	542,084	535,785	539,411	543,948	546,200	545,332	547,693	551,638	595,846	650,973	682,440	746,248	804,053	865,934
C2/1000	1669,56	1666,45	1663,75	1670,19	1671,52	1676,05	1686,86	1694,24	1705,84	1711,85	1722,77	1764,53	1798,12	1813,24	1824,63	1836,52	1868,48
C3/1000	2195,28	2191,65	2201,45	2210,23	2213,70	2228,73	2241,47	2247,91	2261,58	2272,71	2285,59	2318,87	2352,94	2373,35	2382,37	2401,19	2431,43
A1	156	167	169	170	179	180	180	182	196	197	211	223	226	231	232	234	237
A2	197	477	514	713	738	778	874	884	933	946	1013	1114	1146	1186	1224	1383	1421
A3	274	587	633	824	858	890	990	1011	1024	1040	1130	1300	1401	1507	1617	1958	2034
N3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
W	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30

Рис. 3.12. Графики изменения полных затрат в зависимости от числа узлов второго и третьего типа, включаемых в структуру сети алгоритмом Optpac1



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C1/1000	598,009	600,070	596,967	596,958	595,324	588,393	592,280	588,924	594,093	585,141	577,278	618,819	666,290	701,855	761,812	813,814	881,529
C2/1000	1831,79	1821,81	1818,02	1798,08	1795,32	1790,79	1785,95	1773,99	1767,29	1771,99	1770,77	1808,01	1840,05	1850,98	1856,12	1860,73	1889,56
C3/1000	2340,33	2340,93	2327,60	2313,48	2309,19	2314,02	2320,08	2326,19	2320,77	2322,48	2323,87	2344,88	2374,65	2390,31	2397,48	2416,79	2442,48
A1	156	167	169	170	179	180	180	182	196	197	211	223	226	231	232	234	237
A2	197	477	514	713	738	778	874	884	933	946	1013	1114	1146	1186	1224	1383	1421
A3	274	587	633	824	858	890	990	1011	1024	1040	1130	1300	1401	1507	1617	1958	2034
N3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
W	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30

Рис. 3.13. Графики изменения полных затрат в зависимости от числа узлов второго и третьего типа, включаемых в структуру сети алгоритмом Optpac5

Во множество кандидатов $N_{канд}$ включалось 16 узлов, из которых 6 узлов имели первоначальный статус узлов первого типа. В таблицах значения полных затрат $C1$, $C2$, $C3$ приведены в масштабе 1:1000. Из графиков и таблиц видно, что для алгоритма Optpac1 для потоков $A1$ лучшая структура сети была получена при $N3 = 2$ ($C1 = 534570$), для $A2$ при $N3 = 2$ ($C2 = 1663750$), для $A3$ при $N3 = 1$ ($C3 = 2191650$). Для алгоритма Optpac5 лучшие результаты были получены для $A1$ при $N3 = 10$ ($C1 = 577278$), для $A2$ при $N3 = 8$ ($C2 = 1767290$), для $A3$ при $N3 = 4$ ($C3 = 2309190$). Соответствующие отклонения лучших решений для алгоритмов составили 7,99 %, 6,22 % и 5,36 %, т.е. при увеличении потоков уменьшается разница между результатами полученными алгоритмами Optpac1 и Optpac5. Результаты расчетов показывают и подтверждают интуитивное предположение, что при увеличении суммарных потоков узлов уменьшается общее число узлов второго и третьего типа, включаемых в структуру сети. Отметим также, что для заданной сети и потоков $A1$, $A2$, $A3$

во множество $N_2 \cup N_3$ не было введено ни одного узла первого типа из первоначально включенных в $N_{канд}$.

Для наглядного представления полученных результатов на рис. 3.14 показана гистограмма распределения затрат после решения задачи выбора структуры сети алгоритмами Optpac1 и Optpac5 по данным, приведенным в табл. 3.7.

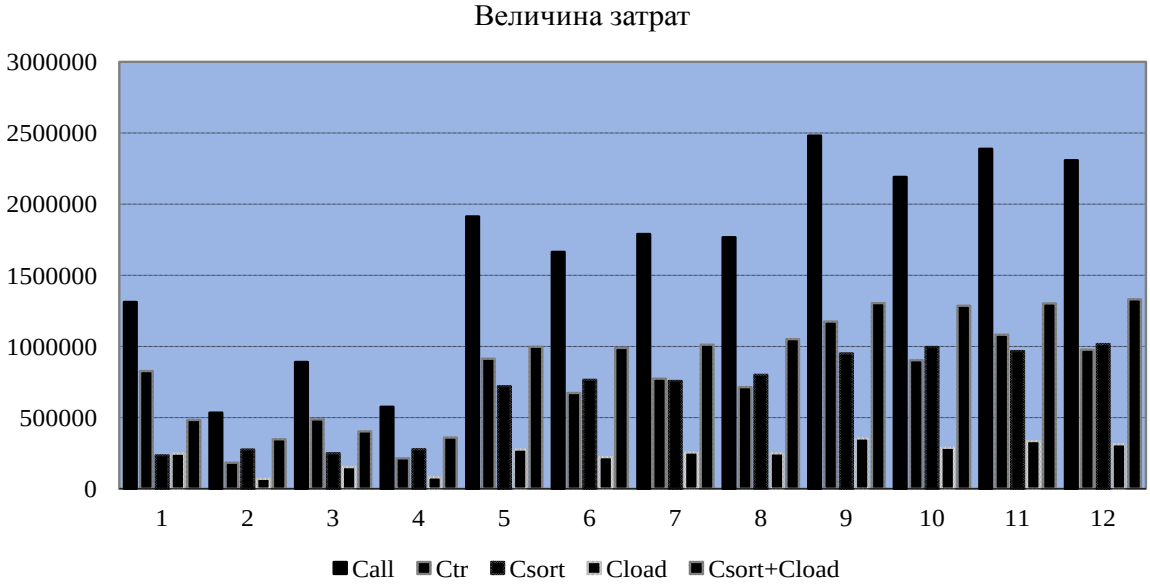


Рис. 3.14. Гистограмма распределения затрат после решения задачи выбора структуры сети

Таблица 3.7

Значения параметров в «оптимальных» точках после решения задачи выбора структуры сети алгоритмами Optpac1 и Optpac5

Параметры	Потоки *A*1, *W*=30				Потоки *A*2, *W*=30				Потоки *A*3, *W*=30			
	Optpac1		Optpac5		Optpac1		Optpac5		Optpac1		Optpac5	
	До птим.	После оптим.	До птим.	После оптим.	До оптим.	После оптим.	До птим.	После оптим.	До оптим.	После оптим.	До птим.	После оптим.
Номер точки	1	2	3	4	5	6	7	8	9	10	11	12
W	30	30	30	30	30	30	30	30	30	30	30	30
*N*3	2	2	10	10	2	2	8	8	1	1	4	4
Call	1312900	534570	891889	577278	1913720	1663750	1789360	1767290	2481560	2191650	2388190	2309190
Ctr	829120	185369	487959	215651	915484	673630	775023	715367	1175490	905065	1083900	977847
Csort	235233	276961	250419	280109	720255	767175	757796	802100	951612	996765	967752	1017530
Cload	248542	72239	153511	81518,1	277979	222945	256541	249822	354452	289818	336534	313816
Csort+Cload	483775	349200	403930	361627,1	998234	990120	1014337	1051922	1306064	1286583	1304286	1331346
Эфф.	-	778330	-	314611	-	249970	-	22070	-	289910	-	79000
tav	0,16159	0,05748	0,08865	0,05863	Infinit	0,14645	Infinit	0,1482	Infinit	Infinit	Infinit	Infinit
*W*max	27	7	14	9	33	27	30	29	41	36	39	41
FS	0	6	0	9	0	6	0	1	0	5	0	1
Kav	0,132	0,683	0,294	0,586	0,553	0,808	0,695	0,811	0,566	0,843	0,651	0,836
*U*min	776	225	479	868	696	689	801	780	1107	905	1051	980
Nav	25	7	15	6	25	18	18	12	27	19	23	15
Y	249	3115	1283	3358	557	4410	3829	7771	228	4599	1852	6901
*Tt*min	15,00000	2,04219	15,00000	2,04219	15,00000	2,04219	15,00000	2,04219	15,00000	2,04219	15,00000	2,04219
*T*max	15,00000	9,82708	15,00000	8,67292	15,00000	6,54063	15,00000	6,06406	15,00000	7,01406	15,00000	6,01615
*T*av	15,00000	3,78798	15,00000	3,95987	15,00000	3,15932	15,00000	3,22618	15,00000	3,15933	15,00000	3,27104

В табл. 3.7 в столбцах с заголовком «До оптим.» приведены значения параметров после решения задачи распределения исходящих потоков из узлов второго и третьего типа в зональных сетях в соответствии с принципами зонально-узловой сортировки для выбранной структуры сети. При этом значение $FS = 0$. В процессе решения задачи упаковки (3.18)-(3.21) параметр FS (ν_{ij} в модели) увеличивается на единицу на каждой итерации алгоритма для поиска наилучшего решения. Алгоритм останавливается, когда не выполняются ограничения $\nu_{ij} \le \nu_{\max} \ \ \forall ij \in S$ или когда значение функции цели (C_{all}) перестает изменяться. В таблице показаны значения параметра FS, при которых был достигнут лучший результат. В строках таблицы с заголовками «Эфф.», «*W*max», «*Y*», «*T*min», «*T*max», «*Tav*», соответственно приведены значения относительного эффекта (C_{all} до оптимизации минус C_{all} после оптимизации), максимальной загрузки дуг сети в транспортных блоках, объемов транзита в узлах сети в единицах грузов, минимального, максимального и среднего времени доставки грузов от отправителей к получателям.

В заключение обсуждения экспериментальных исследований отметим, что аналогичные по характеру поведения численные результаты были получены при решении задачи выбора структуры сети передачи данных, а также при проектировании структуры сетей большой размерности ($n = 300$ и $n = 600$ при $val = 5$). Для сети с $n = 300$ во множество $N_{канд}$ было автоматически введено 75 узлов третьего типа и добавлено 30 узлов первого типа. Потоки генерировались с интенсивностью $A2$, а пропускные способности узлов по обработке транзита h_i, $i = \overline{1,n}$ были увеличены до 1500 единиц. Значения остальных параметров, кроме W, не изменялись.

Решение задачи для $W = 40$ и $W = 6550$ выполнялось алгоритмом Optpac5 с целью максимального сокращения времени счета. Для первого значения параметра W в структуру сети было введено 86 узлов второго и третьего типа и 214 узлов первого типа, для второго — 81 узел второго и третьего типа и 219 узлов первого типа. То есть с увеличением значения W проявляется та же тен-

денция к сокращению числа узлов третьего типа, вводимых в структуру сети.

Для сети с n = 600 в множество $N_{канд}$ было автоматически введено 159 узлов третьего типа и добавлено 30 узлов первого типа. Потоки генерировались с интенсивностью $A2$, а пропускные способности узлов по обработке транзита не ограничивались. В результате решения задачи при заведомо недостаточной пропускной способности дуг (W = 20000, при Wmax = 42363) в структуру сети было включено 189 узлов второго и третьего типа и 411 узлов первого типа, т.е. все дополнительные узлы первого типа потеряли свой первоначальный статус.

Проведенный анализ полученных результатов экспериментальных исследований позволяет сделать следующие выводы.

1. При решении задачи упаковки, выбор структуры сети слабо зависит от величины пропускных способностей ее дуг.

2. Несмотря на то, что алгоритм Optpac1 включает в структуру сети меньшее число узлов второго и третьего типа, чем алгоритм Optpac5, оба алгоритма могут с успехом применяться для решения задачи выбора структуры реальных сетей, функционирующих в условиях воздействия случайных факторов, колебания величины исходящих потоков и изменения текущих ресурсов, так как полученные ими решения с практической точки зрения отличаются незначительно. Алгоритмы Optpac3- Optpac5 работают гораздо быстрее алгоритмов Optpac1- Optpac2 и в ряде случаев, когда нет необходимости в получении наилучшего решения, могут быть более предпочтительными.

3. При выборе окончательной структуры сети и предварительной оценке пропускных способностей W дуг сети следует учитывать значение среднего времени задержки потоков t_{av}, при котором начинают выполняться ограничения (3.26).

Выводы по третьему разделу

1. На основании доказанных утверждений предложены способ построения справочной матрицы объединения мелкопартионных потоков и эффективные алгоритмы, позволяющие определять узлы объединения и объединенные потоки для всех корреспондирующихся пар в многопродуктовой сети. Разработан-

ные алгоритмы могут быть использованы при решении различных задач оптимизации упаковки на транспортных сетях с мелкопартионными потоками грузов, а также при проектировании и анализе передачи пакетов сообщений в виртуальных контейнерах в перспективных опорных сетях на основе сверхширокополосных каналов связи (например, в таких, как Европейская опорная сеть E-bone).

2. Доказана NP-полнота одной задачи оптимизации упаковки, которая используется при сортировке и упаковке мелкопартионных грузов в контейнеры в транспортных сетях или объединении сообщений в виртуальные контейнеры в сетях передачи данных. Приведены данные численного эксперимента с простыми тестовыми алгоритмами упаковки, которые показывают сильную зависимость результатов решения задачи от границ изменения значений входных данных и выбора схем перебора вариантов решения, а также указывают, на что нужно обратить внимание при разработке алгоритмов для решения задач обработки и транспортировки мелкопартионных корреспонденций в реальных коммуникационных сетях.

3. Предложена постановка задачи оптимизации упаковки с функциями затрат и обсуждены особенности ее решения для реальных сетей. Разработан ряд эвристических алгоритмов решения задачи и доказана их сходимость к локальному минимуму за полиномиальное время. Получены оценки временной сложности алгоритмов. Проведены экспериментальные исследования разработанных алгоритмов, показывающие их работоспособность, и выполнен сравнительный анализ быстродействия алгоритмов на сетях, содержащих до 500 узлов. Результаты моделирования позволяют утверждать о применимости алгоритмов для решения практических задач упаковки на сетях большой размерности. Алгоритмы решения задачи допускают распараллеливание процесса нахождения локальных оптимумов и для увеличения их быстродействия могут выполняться на вычислительных кластерах или многопроцессорных компьютерах.

4. Доказано, что на этапе решения задачи упаковки нет необходимости решать дискретную задачу выбора пропускных способностей дуг, а нелинейная

функция транспортных затрат может быть заменена на функцию удельной стоимости транспортировки транспортных блоков от расстояния и грузоподъемности транспортных средств или пропускной способности линий связи. Доказательство подтверждается результатами численного моделирования. Показано, что задача выбора пропускных способностей может быть решена независимо после решения задачи упаковки и использоваться при решении задачи распределения и маршрутизации транспортных блоков.

5. Доказано, что задача выбора пропускных способностей дуг, сформулированная как задача распознавания, является NP-полной. Приведено полиномиальное преобразование исходной постановки задачи к задаче о ранце с мультивыбором и булевыми переменными, которая также принадлежит классу NP-полных задач. Для решения задачи предложено два алгоритма, основанных на аппроксимации дискретных функций затрат линейными, и на методе последовательного анализа вариантов. Проведены экспериментальные исследования алгоритмов, которые показали, что оба алгоритма без полного перебора вариантов на заключительном этапе не дают точного решения. Алгоритмы псевдополиномиальны, так как для полного перебора требуется $O(2^e)$ времени, где e — число дуг в сети.

6. Предложена математическая модель задачи выбора иерархической структуры магистральной сети и разработана общая схема эвристического алгоритма ее решения, основанная на применении алгоритмов упаковки и распределения потоков в зональных сетях. Экспериментально показано, что выбор структуры сети слабо зависит от пропускных способностей ее дуг и их можно не учитывать при решении задачи.

РАЗДЕЛ 4. ЗАДАЧИ ПОСТРОЕНИЯ МАРШРУТОВ В МАГИСТРАЛЬНОЙ СЕТИ

В первом подразделе рассматривается математическая модель нелинейной многоэкстремальной дискретной задачи распределения и маршрутизации транспортных блоков с упакованными в них мелкопартионными потоками грузов или сообщений с различными адресами назначения. Обсуждаются особенности ее решения для транспортной сети и сети передачи данных для неразветвленных и разветвленных потоков при заданных ограничениях на время доставки мелкопартионных потоков корреспонденций получателю и среднее время задержки потоков. Предложен метод сведения решения исходной задачи к решению некоторой совокупности линейных многомерных задач о ранце со связывающими ограничениями.

Во втором подразделе приведена формулировка задачи оптимизации распределения потоков с заданными тарифами на дугах и в узлах на транспортировку и обработку потоков. Доказано, что задача с тарифами в сетевой постановке может быть за полиномиальное время преобразована к задаче целочисленного линейного программирования с блочной структурой и связывающими ограничениями. Отмечаются особенности решения преобразованной задачи при использовании известных методов целочисленного программирования и пакетов прикладных программ.

В третьем подразделе предложен новый алгоритм построения кратчайших путей между всеми парами узлов в неориентированной сети по критерию: минимум дуг в пути; минимум длины пути. Проведен анализ сложности алгоритма и эмпирически показано, что он, по мере увеличения плотности сети, на несколько порядков быстрее алгоритма Флойда, соответствующим образом модифицированного для нахождения кратчайших путей по ступенчатому критерию. В приложении Г приведен обзор и анализ методов и алгоритмов построения кратчайших путей.

В четвертом подразделе рассматриваются две реализации алгоритма Краскала на списочных и древовидных структурах данных. Путем численных экспе-

риментов на связных и несвязных, плотных и разреженных графах большой размерности проведено сравнение вычислительной эффективности предложенного алгоритма на списочных структурах с алгоритмом Тарьяна на древовидных структурах, а также с алгоритмом Прима в классической матричной реализации. Получены оценки временной сложности алгоритмов. Результаты сравнения позволяют утверждать, что для решения практических задач нахождение минимального или максимального остовного дерева (леса) алгоритмы со списочной структурой данных работают не хуже, а в большинстве случаев быстрее, чем алгоритмы с древовидной структурой.

В пятом подразделе предложены алгоритмы решения задачи распределения и маршрутизации транспортных блоков. Получены оценки временной сложности разработанных алгоритмов.

В шестом подразделе рассмотрен численный пример проектирования маршрутов для автотранспортной сети перевозок.

В приложении Д рассматриваются абстрактные типы данных для разработки алгоритмов распределения и маршрутизации потоков. Предложены процедуры редукции, позволяющие значительно сократить требуемые объемы оперативной памяти для представления структур данных при решении задач проектирования новых маршрутов транспортировки потоков, когда в качестве исходных маршрутов выступает их произвольное комбинаторное множество. Приведены оценки сложности алгоритмов. Рассмотрен пример построения структур данных и проведен численный эксперимент для проверки работоспособности и вычислительной эффективности предложенных алгоритмов.

Ключевые слова: многопродуктовая сеть, мелкопартионные потоки корреспонденций, математические модели распределения и маршрутизации потоков, многомерная задача о ранце, NP-трудные задачи, эвристические алгоритмы, сходимость и временная сложность алгоритмов, многокритериальные задачи и алгоритмы построения кратчайших путей, остовные деревья и леса, абстрактные типы данных

4.1. Общая задача распределения и маршрутизации потоков транспортных блоков со смешанными вложениями и нелинейными затратами

Задача распределения и маршрутизации дискретных многопродуктовых потоков транспортных блоков со смешанными вложениями возникает в транспортных сетях с мелкопартионными грузами и в магистральных опорных сетях передачи данных с технологией виртуальных контейнеров [11, 123, 467]. Под смешанными вложениями понимаются объединенные в один транспортный блок (контейнер) мелкопартионные тарно-штучные грузы или сообщения с разными адресами назначения, которые могут не совпадать с адресом назначения транспортного блока. Смешанные потоки образуются для максимального сокращения числа транспортных блоков, необходимых для упаковки и транспортировки мелкопартионных потоков, но при этом увеличивается и время на доставку потребителю тех мелкопартионных потоков, которые проходят дополнительную сортировку в транзитных узлах следования.

Содержательная постановка задачи заключается в выборе такой схемы распределения и маршрутизации потоков транспортных блоков, сформированных при решении задачи выбора иерархической структуры магистральной сети и схемы сортировки мелкопартионных корреспонденций в узлах сети, при которой максимально снижаются приведенные затраты на обработку и транспортировку потоков. Решение задачи должно осуществляться в интерактивном режиме и определять основные технико-экономические показатели функционирования магистральной сети при изменении исходных данных, параметров и ограничений транспортной модели. Поскольку не всегда удается формализовать все факторы, влияющие на выбор наилучшего решения, для окончательного выбора схемы распределения и маршрутизации потоков может использоваться практический опыт диспетчеров транспортных сетей и администраторов сетей передачи данных, а также базы знаний в информационно-аналитической системе поддержки принятия решений (раздел 1).

В данном подразделе исследуется задача распределения и маршрутизации транспортных блоков с упакованными в них мелкопартионными грузами или

сообщениями и предлагается метод сведения ее решения к решению некоторой совокупности линейных многомерных задач о ранце со связывающими ограничениями. Особенность задачи заключается в том, что при ее решении необходимо учитывать заданные ограничения на время доставки получателю мелкопартионных потоков грузов и сообщений.

4.1.1. Математическая модель

В [15] предложены линейные целочисленные модели распределения потоков для задач проектирования и анализа многопродуктовых коммуникационных сетей. Показано, что для задач распределения потоков большой размерности (сотни миллионов и более переменных), возникающих при проектировании и анализе реальных сетевых структур (более 200 узлов и 12 тыс. дуг), целесообразна разработка сетевых математических моделей, наиболее адекватно описывающих процессы обработки и транспортировки мелкопартионных потоков. В настоящем подразделе для задачи распределения и маршрутизации потоков предлагается новая модель в сетевой постановке.

Пусть $G(N,P)$ — иерархическая магистральная сеть с множеством неориентированных дуг P, $p=\left| P \right|$ и множеством узлов $N=N_1 \cup N_2 \cup N_3$, $n=\left| N \right|$, где N_1, N_2, N_3 — множества узлов первого, второго и третьего типа соответственно. Узлы сети соответствуют пунктам сортировки, отправления, назначения и перегрузки потоков, а дуги — участкам дорог для транспортных сетей или каналам связи для сетей передачи данных, связывающим узлы сети. Узлы второго и третьего типа отличаются от узлов первого типа функциональными возможностями, уровнем технической оснащенности, числом обслуживающего персонала (а значит и функциями затрат на обработку потоков) и пр. В узлах третьего типа запрещена обработка транзитных потоков транспортных блоков.

Мелкопартионные потоки корреспонденций заданы исходной целочисленной матрицей $A=\left\| a_{ij} \right\|_{n\times n}$ и преобразованной целочисленной матрицей

$A' = \| a'_{ij} \|_{n \times n}$, полученной после решения задачи упаковки транспортных блоков (раздел 3). Элементы матрицы A' определяются следующим образом:

$$a'_{ij} = \begin{cases} a_{ij} + \sum\limits_{rs \in \Omega_{ij}} a_{rs}, \text{ если потоки } \{a_{rs}\} \text{ объединены с потоком } a_{ij}, \\ 0, \text{ если поток } a_{ij} \text{ объединен с каким-либо другим потоком или } i = j, \end{cases}$$

для $ij \in S$, где S — множество индексов ij потоков, а Ω_{ij} — множество индексов rs потоков $\{a_{rs}\}$, объединенных с потоком a_{ij} (множество Ω_{ij} может быть пустым).

Пусть $\tilde{A} = \| \tilde{a}_{ij} \|_{n \times n}$, $\tilde{a}_{ij} = \left\lceil \dfrac{a'_{ij}}{\omega} \right\rceil$ — матрица потоков транспортных блоков, где ω — объем транспортного блока, $\lceil \cdot \rceil$ — знаки округления числа до большего целого. Потоки $\tilde{a}_{ij}$ из источников i в стоки j должны перевозиться в транспортных средствах или передаваться по каналам связи с заданной периодичностью.

Пусть $\{m_k\}$, $k = \overline{1,l}$, — заданное множество проектируемых маршрутов транспортных средств или каналов связи, каждый из которых состоит из последовательности узлов и дуг сети G, соединяющей начальный и конечный узлы маршрута или канала связи. Предполагается, что множество $\{m_k\}$ для каждой неориентированной дуги сети G содержит прямой и обратный маршруты, и в процессе решения задачи во множество $\{m_k\}$ могут включаться новые маршруты, генерируемые по определенным правилам. Множество $\{m_k\}$ может содержать несколько маршрутов, соединяющих любую пару узлов. С каждым маршрутом транспортной сети связаны его характеристики: функция среднегодовых приведенных затрат на эксплуатацию и содержание маршрута; грузоподъемность и периодичность движения транспортных средств; время прибытия и отправления транспортного средства для каждого узла в маршруте и др. Для каждого маршрута в сети передачи данных заданы функция среднегодовых приведенных затрат на эксплуатацию и содержание канала связи, его длина и пропускная способность.

Построим маршрутную мультисеть $G_M(N, P_M)$ транзитивным замыканием узлов всех маршрутов из $\{m_k\}$, где N — множество узлов сети, P_M — множество ее ориентированных маршрутных дуг. Между любыми узлами α и β сети G_M существует маршрутная дуга, если они связаны хотя бы одним маршрутом транспортного средства или каналом связи из $\{m_k\}$. Введем переменные: $u_{ij,k}^{\alpha\beta}$ — неизвестный поток транспортных блоков из i в j, проходящий по дуге $p_{\alpha\beta} \in P_M$, полученной из маршрута m_k ($u_{ij,k}^{\alpha\beta}$ определяют дуговые потоки в транспортных блоках на маршрутной сети G_M); $u_{ij,k}^{\eta\xi}$ — неизвестный поток транспортных блоков из i в j, проходящий по дуге $p_{\eta\xi} \in P$ на маршруте m_k.

Требуется минимизировать функцию

$$\sum_{k=1}^{l} C_{tr}^{k}\left(\left(\sum_{\eta\xi \in q_k}\sum_{ij \in S} u_{ij,k}^{\eta\xi}\right), d_k\right) + \sum_{\beta=1}^{n} C_{load}^{\beta}\left(\sum_{\alpha=1}^{n}\sum_{k=1}^{l}\sum_{ij \in S}(u_{ij,k}^{\alpha\beta} + u_{ij,k}^{\beta\alpha})\right), \tag{4.1}$$

при ограничениях:

$$\sum_{\beta=1}^{n}\sum_{k=1}^{l} u_{ij,k}^{\alpha\beta} - \sum_{\beta=1}^{n}\sum_{k=1}^{l} u_{ij,k}^{\beta\alpha} = \begin{cases} \tilde{a}_{ij} \text{ при } i = \alpha, \\ 0 \text{ при } i \neq \alpha, j \neq \alpha, \\ -\tilde{a}_{ij} \text{ при } j = \alpha, \text{для } \alpha = \overline{1,n}, ij \in S; \end{cases} \tag{4.2}$$

$$\sum_{\alpha=1}^{n}\sum_{k=1}^{l}\sum_{ij \in S}(u_{ij,k}^{\alpha\beta} + u_{ij,k}^{\beta\alpha}) - \sum_{j=1}^{n}(\tilde{a}_{\beta j} + \tilde{a}_{j\beta}) \leq 2b_\beta, \ \beta = \overline{1,n}; \tag{4.3}$$

$$\sum_{ij \in S} u_{ij,k}^{\eta\xi} \leq W_{\eta\xi}^{k} \text{ для всех } \eta\xi \in q_k, \ k = \overline{1,l}; \tag{4.4}$$

$$\sum_{\beta=1}^{n}\sum_{ij \in S}(u_{ij,k}^{\alpha\beta} + u_{ij,k}^{\beta\alpha}) \leq b_\alpha^k, \ \alpha \in \upsilon_k, \ k = \overline{1,l}; \tag{4.5}$$

$$t_{av} = 1/\tilde{a}_{\Sigma}\sum_{k=1}^{l}\sum_{\eta\xi \in q_k} f_{\eta\xi}^{k}/(W_{\eta\xi}^{k} - f_{\eta\xi}^{k}) \leq T_{\max}, \text{ где } \tilde{a}_{\Sigma} = \sum_{ij \in S}\tilde{a}_{ij}, \ f_{\eta\xi}^{k} = \sum_{ij \in S} u_{ij,k}^{\eta\xi}; \tag{4.6}$$

$$u_{ij,k}^{\alpha\beta} \geq 0, u_{ij,k}^{\eta\xi} \geq 0 \text{ - целые числа.} \tag{4.7}$$

Предполагается, что имеется оператор $\Phi: u_{ij,k}^{\alpha\beta} \Rightarrow \{u_{ij,k}^{\eta\xi}\}$, $p_{\alpha\beta} \in P_M$, $p_{\eta\xi} \in P$, $ij \in S$, $k = \overline{1,l}$, отображающий поток по маршрутной дуге маршрута m_k в сети G_M на соответствующее подмножество дуг маршрута m_k в сети G; учитывают-

ся ограничения на время доставки мелкопартионных потоков получателю $t_{ij} \le T_{ij},\ ij \in S$.

Заметим, что число ограничений (4.2) можно сократить до числа узлов-источников потоков, и записать их в виде

$$\sum_{\beta=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{l} u_{ij,k}^{\alpha\beta} - \sum_{\beta=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{l} u_{ij,k}^{\beta\alpha} = \begin{cases} \sum_{j=1}^{n} \tilde{a}_{ij} \text{ при } i = \alpha, \\ 0 \text{ при } i \ne \alpha,\ j \ne \alpha, \\ -\sum_{i=1}^{n} \tilde{a}_{ij} \text{ при } j = \alpha, \text{ для } \alpha = \overline{1,n},\ j = \overline{1,n} \end{cases} .$$

В конкретных случаях решения задачи к указанным ограничениям могут быть добавлены ограничения на запрет разветвления потоков:

$$u_{ij,k}^{\alpha\beta} = \begin{cases} \tilde{a}_{ij}, \text{если поток проходит по дуге } \alpha\beta \in m_k, \\ 0 \text{ в противном случае.} \end{cases} \qquad (4.8)$$

В формулах (4.1)-(4.8) введены обозначения:

• C_{tr}^{k} — нелинейная функция, определяющая зависимость транспортных затрат от количества транспортных блоков, передаваемых по маршруту m_k, и от длины маршрута d_k;

• q_k — упорядоченное множество дуг из P, составляющих маршрут m_k;

• C_{load}^{β} — нелинейная функция затрат на обработку транспортных блоков в узле β;

• $b_{\beta},\ \beta = \overline{1,n}$ — максимальная пропускная способность β- го узла в транспортных блоках, пропускная способность задается для транзитных потоков, так как исходящие и входящие потоки для каждого узла должны быть обработаны безусловно;

• $W_{\eta\xi}^{k}$ — грузоподъемность транспортного средства или пропускная способность канала связи на маршруте m_k на дуге $\eta\xi \in P$ в транспортных блоках, $W_{\eta\xi}^{k} \in \{w_1, w_2, ..., w_v\}$, где $w_1,\ w_2, \ldots,\ w_v$ — целые, упорядоченные по возрастанию положительные числа;

• b_{α}^{k} — ограничения на максимальное суммарное число транспортных бло-

ков, которое можно обработать в транзитном узле α на маршруте m_k;

• υ_k — упорядоченное множество узлов из N на маршруте m_k;

• t_{av}, $T_{\max}$ — расчетная средняя и заданная максимальная задержка в передаче транспортных блоков в сети;

• t_{ij}, T_{ij}, $ij \in S$, — расчетное и заданное время на доставку мелкопартионных потоков корреспонденций a_{ij} из i в j.

Первая составляющая функции определяет транспортные затраты, вторая — затраты на обработку транспортных блоков. Условия (4.2) обеспечивают неразрывность потока, а условия (4.3)-(4.6) представляют собой соответственно ограничения на пропускные способности узлов; пропускные способности маршрутов; объемы обработки транспортных блоков в узлах следования транспортных средств или в коммутированных узлах сети передачи данных; среднюю задержку в передаче транспортных блоков.

4.1.2. Особенности задачи

Любой поток транспортных блоков может разветвляться, т.е. разбиваться на несколько частей, транспортировка каждой может выполняться по различным путям. В свою очередь, любая часть потока также может разветвляться. Процесс разветвления может продолжаться до тех пор, пока части потоков не будут представлены неделимыми элементами — одним транспортным блоком. Сложная схема разветвления потоков в транспортных сетях при колебании величины потоков может создать серьезные трудности диспетчерам при практическом управлении перевозками и ухудшить качество транспортных сообщений. Поэтому для транспортных сетей учет возможности разветвления потока целесообразен на уровне оперативного управления перевозками при решении задач локальной оптимизации, имеющих значительно меньшую размерность. Для сетей передачи данных разветвление потоков необходимо для получения наилучшего решения задачи. В этом случае усложнение схемы маршрутизации потоков не ухудшает качество функционирования сети, так как управление по-

токами полностью автоматизировано.

Условия $t_{ij} \le T_{ij}$, $ij \in S$, ограничивают время доставки мелкопартионных потоков корреспонденций получателю. Поэтому при решении задачи распределения потоков транспортных блоков $\tilde{a}_{ij}$ необходимо проверять выполнимость этих условий для мелкопартионных потоков a_{ij} и $\{a_{rs}\}$, $rs \in \Omega_{ij}$, упакованных в $\tilde{a}_{ij}$. Множества Ω_{ij} и $\{a_{rs}\}$ определяются из справочной матрицы объединения потоков (подраздел 3.1), из которой можно получить информацию о том, какие мелкопартионные потоки находятся в потоке транспортных блоков $\tilde{a}_{ij}$ из i в j, $ij \in S$, а также определить все узлы дополнительной сортировки любого мелкопартионного потока a_{ij}, $ij \in S$. Это дает возможность с помощью алгоритмов, приведенных в подразделе 3.1, рассчитывать время доставки мелкопартионных потоков получателю в процессе решения задачи распределения и маршрутизации потоков транспортных блоков. При расчете времени доставки используются также параметры, явно не входящие в модель: заданное время на сортировку мелкопартионных потоков в узлах сети, время на транзитную перегрузку транспортных блоков в узлах сети, время стоянки транспортных средств в конечных пунктах следования, периодичность и средняя скорость движения транспортных средств или передачи сообщений и др. (см. подраздел 4.6).

Отметим также, что справочная матрица объединения потоков является главным результатом решения задачи упаковки и полностью определяет схему сортировки мелкопартионных потоков a_{ij}, $ij \in S$, во всех узлах сети, а также адресует потоки транспортных блоков $\tilde{a}_{ij}$, $ij \in S$, которые должны быть распределены по маршрутам транспортных средств или по каналам связи. В узлах реальной транспортной сети или сети передачи данных справочная матрица объединения потоков используется для автоматизированного управления оборудованием, осуществляющим процессы сортировки тарно-штучных грузов, или как таблица слияния сообщений в виртуальные контейнеры.

Еще одной важной особенностью решения поставленной задачи является

учет ограничений на пропускные способности маршрутов (4.4) и среднюю задержку в передаче транспортных блоков (4.6) для транспортных сетей и сетей передачи данных. На практике для транспортных сетей, когда задано множество маршрутов транспортных средств и их грузоподъемность, нет необходимости «жесткого» учета ограничения (4.6). Из множества заданных транспортным диспетчером проектируемых маршрутов и множества новых сгенерированных маршрутов в процессе решения задачи будут выбраны только «наилучшие» в смысле достижения минимума функции (4.1), некоторые «длинные» заданные маршруты могут быть обрезаны или разделены на несколько частей. В процессе функционирования сети при возможных перегрузках на выбранных маршрутах всегда имеется возможность оперативного введения дополнительных транспортных средств. Если же диспетчер планирует обеспечить запас грузоподъемности транспортных средств из-за значительного колебания величины потоков на периодах текущего планирования, то ограничение (4.6) нужно учитывать. В этом случае грузоподъемности транспортных средств на заданных и вновь вводимых маршрутах выступают как искомые величины, и возникает необходимость итеративного решения задач выбора грузоподъемности (подраздел 3.4) и распределения потоков, пока не будет выполняться условие (4.6). Для сетей передачи данных ограничение (4.6), как правило, учитывается всегда. Окончательный выбор остается за диспетчером, который в интерактивном режиме может указать, нужно ли это ограничение.

Учитывая вышесказанное, можно выделить четыре варианта решения задачи (4.1)-(4.8), когда одно или оба ограничения на среднее время задержки потоков (4.6) и разветвление потоков (4.8) учитываются или не учитываются. Для каждого из вариантов решения задачи (4.1)-(4.8) возможно шестнадцать режимов оптимизации, когда ограничения на пропускные способности узлов (4.3), пропускные способности маршрутов (4.4), объемы обработки транспортных блоков в транзитных узлах маршрутов (4.5), время доставки мелкопартионных потоков получателю должны обязательно учитываться или могут ослабляться.

Предположим, имеется алгоритм решения задачи (4.1)-(4.8). Решение бу-

дем считать успешным, если с помощью этого алгоритма удалось распределить все потоки, заданные на сети, и неудачным в противном случае. Ясно, что при некорректном задании ресурсов задача может не иметь решения. В этом случае можно ослабить некоторые ограничения и перераспределить потоки. Выбор тех или иных ослабляемых ограничений может быть выполнен в интерактивном режиме, который должна инициировать программа оптимизации, реализующая алгоритм. Из вышеизложенного следует три важных свойства алгоритма. Во-первых, алгоритм должен определить, что данная задача не имеет решения; во-вторых, обеспечить диалог с пользователем в случае невозможности получения решения при заданных ограничениях; в-третьих, структура алгоритма должна быть такова, чтобы он мог обеспечить решение задачи (4.1)-(4.8) в любом из вариантов и режимов, представляющих практический интерес.

Задача (4.1)-(4.8) относится к классу многоэкстремальных дискретных распределительных задач на сети и является NP-трудной [45]. Сложность решения подобных задач отмечается в работе [160], где рассматривается частично-целочисленная оптимизационная задача составления маршрутов и формирования поездов в транспортной сети. Приведенная в [160] задача во многом сходна с задачей (4.1)-(4.8), в них минимизируются транспортные затраты и затраты на обработку грузов в узлах при заданных объемах перевозимых грузов и маршрутах движения транспортных средств. При этом транспортные затраты имеют дискретный характер и учитываются ограничения на грузоподъемность транспортных средств. Существенное упрощение модели [160] по сравнению с (4.1)-(4.8) заключается в том, что в ней при движении по любому маршруту обмен грузами в промежуточных узлах запрещен, — груз принимается в начальном узле маршрута и сдается в конечном; ограничения на пропускные способности узлов и время доставки грузов не учитываются. В [160] рассматриваются также возможные подходы к построению эвристических алгоритмов, разработка которых, по мнению автора, будет способствовать дальнейшему развитию эффективных методов решения для класса задач подобного типа. Описание различных подходов и методов решения задач о многопродуктовых потоках в сетях и

обширную библиографию по многопродуктовым задачам можно найти в работах [161, 339].

Следует отметить также работы, посвященные исследованию и решению задач оптимального синтеза и развития сетей, формулировка которых во многом аналогична задаче (4.1)-(4.8) при учете отдельных ограничений. Так, например, в [91, 108] рассматриваются задачи синтеза сетей с вогнутой целевой функцией. Указывается на NP-трудность таких задач и предлагается в рамках схемы метода ветвей и границ способ сведения решения исходной задачи к решению последовательности задач линейного программирования. Отмечается эффективность предложенного приближенного алгоритма. О сложности решения задач развития транспортных сетей, характеризующихся невыпуклостью целевого функционала и дискретностью переменных, указывается в работах [10, 77]. Для решения задач рекомендуется использовать различные способы агрегирования и декомпозиции, разработку эвристических алгоритмов и организацию интерактивного режима оптимизации.

Аналогичные результаты получены при изучении распределения и маршрутизации дискретных потоков в сетях передачи данных в работах [6, 37, 60, 66, 67, 217, 268]. Для частных случаев решения задачи предложен ряд эвристических (субоптимальных) алгоритмов и подходы, основанные на методе релаксации Лагранжа и сведения исходной задачи к двойственной, решение которой позволяет получить точную нижнюю оценку.

В работах [1, 71, 82-84, 96, 144] рассматриваются теоретические аспекты решения задач о многопродуктовых потоках в сетях. Исследования, проведенные в этих работах, показали, что сложности, возникающие при решении целого ряда многопродуктовых задач, носят принципиальный характер, и свидетельствуют о NP-трудности рассматриваемого класса задач.

Наиболее современный обзор по многопродуктовым задачам в различных областях применения и методам их решения можно найти в работах [55, 239, 477].

Многоэкстремальность и дискретность сформулированной задачи, а также

наличие ограничений вида (4.5), (4.6) и $t_{ij} \le T_{ij}$, $ij \in S$, не позволяют получить даже ее приближенное решение с гарантированной оценкой точности. Поэтому, в лучшем случае, для решения задачи (4.1)-(4.8) могут быть разработаны субоптимальные методы и алгоритмы, дающие рациональные, с точки зрения проектировщика сети, решения для заданных ограничений задачи при выбранных вариантах и режимах ее решения.

4.1.3. Декомпозиция задачи к совокупности задач о ранце

В связи со сложностью сформулированной задачи рассмотрим метод сведения ее решения к последовательному решению некоторой совокупности более простых линейных многомерных задач о ранце со связывающими ограничениями.

Интуитивно понятно, что можно достичь уменьшения значения функции (4.1) при максимальном увеличении значений $u_{ij,k}^{\eta\xi}$ на всех дугах (участках) $\eta\xi \in P$ маршрутов $\{m_k\}$ при распределении всех потоков в сети G_M по критерию: минимум транзитных перегрузок, при равенстве числа транзитных перегрузок — минимум длины пути. В результате построим некоторый аналог двойственной задачи к задаче (4.1)-(4.8) при исключении из явного рассмотрения нелинейных функций затрат C_{tr}^{k} и C_{load}^{β}. Первоначально предположим, что все потоки $\tilde{a}_{ij}$ распределяются в сети G_M по указанному критерию. Используя справочную матрицу распределения потоков по кратчайшим путям $C = \| c_{ij} \|_{n \times n}$ [28] с элементами

$$c_{ij} = \begin{cases} \mu, \text{если поток } \tilde{a}_{ij} \text{ передается из узла } i \text{ в узел } j \text{ с перегрузкой в узле } \mu, \\ i, \text{если поток } \tilde{a}_{ij} \text{ передается из узла } i \text{ в узел } j \text{ без перегрузок,} \\ 0, \text{если } i = j, \end{cases} \qquad (4.9)$$

и заданное множество $\{m_k\}$, $k = \overline{1,l}$, для каждого маршрута m_k нетрудно найти подмножества $\{u_{ij,k}^{\alpha\beta}\}$ потоков, которые потенциально могут быть распределены на этот маршрут. Обозначим $\left| \{u_{ij,k}^{\alpha\beta}\} \right| = \varepsilon_k$. Поскольку G_M — мультисеть, для

любых m_{k_1}, m_{k_2}, $k_1 \neq k_2$ и фиксированных $\alpha\beta$ и ij, может выполняться $\{u_{ij,k_1}^{\alpha\beta}\} \cap \{u_{ij,k_2}^{\alpha\beta}\} \neq \varnothing$.

Перенумеруем, начиная с единицы, ненулевые потоки $\tilde{a}_{ij}$ в матрице $\tilde{A} = \| \tilde{a}_{ij} \|_{n \times n}$ по строкам слева направо и по столбцам сверху вниз. Построим матрицу $H = \| h_{ij} \|_{n \times n}$, в которой

$$h_{ij} = \begin{cases} \text{номеру } h \text{ потока } \tilde{a}_{ij}, \text{если } \tilde{a}_{ij} \neq 0, \\ 0 \text{ в противном случае}, h = \overline{1, h_{\max}}, \end{cases}$$

где $h_{\max}$ — число ненулевых элементов в матрице $\tilde{A}$, $h_{\max} \leq n^2 - n$.

Пусть $\xi^k = \| \xi_i^k \|_{\eta_k}$, $\eta_k = | \upsilon_k |$, $k = \overline{1, l}$, — векторы, в которых содержатся номера узлов сети (G или G_M) на маршрутах m_k при прохождении их слева направо. Подразумевается, что прямые и обратные маршруты могут как совпадать, так и не совпадать. Запишем оператор отображения любого потока $u_{ij,k}^{\alpha\beta} \in \{u_{ij,k}^{\alpha\beta}\}$ по маршрутной дуге $\alpha\beta$ на m_k на множество дуг этого маршрута в сети G в виде $\Phi : u_{ij,k}^{\alpha\beta} \Rightarrow \{u_{ij,k}^{\xi_\mu \xi_{\mu+1}}, u_{ij,k}^{\xi_{\mu+1} \xi_{\mu+2}}, \ldots, u_{ij,k}^{\xi_{\mu+c-1} \xi_{\mu+c}}\}$, где $\xi_\mu = \alpha$, μ — порядковый номер узла α в векторе ξ^k, $\mu \in [\overline{1, \eta_k})$; $\xi_{\mu+c} = \beta$, $\mu + c$ — порядковый номер узла β в векторе ξ^k, $\mu + c \in (\overline{1, \eta_k}]$.

Построим векторы $f^k = \| f_j^k \|_{\varepsilon_k}$ и матрицы $\dot{A}^k = \| \dot{a}_{ij}^k \|_{(\eta_k - 1) \times \varepsilon_k}$, $\ddot{A}^k = \| \ddot{a}_{ij}^k \|_{\eta_k \times \varepsilon_k}$, $\dddot{A}^k = \| \dddot{a}_{ij}^k \|_{\eta_k \times \varepsilon_k}$, $\ddddot{A}^k = \| \ddddot{a}_{ij}^k \|_{\eta_k \times \varepsilon_k}$, $k = \overline{1, l}$, следующим образом:

$f_j^k = h$, если поток $\tilde{a}_{i'j'}$ с номером h может быть распределен на маршрут m_k. Индексы i' и j' означают индексы i и j переменных $u_{ij,k}^{\alpha\beta} \in \{u_{ij,k}^{\alpha\beta}\}$, где $i, j = \overline{1, n}$, поэтому $\tilde{a}_{i'j'}$ определяют элементы $\tilde{a}_{ij}$ матрицы $\tilde{A}$. Для удобства потоки во множестве $\{u_{ij,k}^{\alpha\beta}\}$ могут быть упорядочены по увеличению нумерации соответствующих им потоков $\tilde{a}_{i'j'}$ в H;

$$\dot{a}_{ij}^{k}=\begin{cases}\widetilde{a}_{i'j'}\ \text{для}\ i=\overline{\mu,\mu+c-1},\\ 0\ \text{для}\ i\notin[\mu,\mu+c-1],\ i=\overline{1,\eta_k-1};\end{cases}\quad \ddot{a}_{ij}^{k}=\begin{cases}\widetilde{a}_{i'j'},\ \text{если}\ (i=\mu)\vee(i=\mu+c),\\ 0\ \text{в противном случае},\ i=\overline{1,\eta_k};\end{cases}$$

$$\dddot{a}_{ij}^{k}=\begin{cases}\widetilde{a}_{i'j'},\ \text{если}\ (i=\mu\wedge i'\neq\alpha)\vee(i=(\mu+c)\wedge j'\neq\beta),\\ 0\ \text{в противном случае},\ i=\overline{1,\eta_k};\end{cases}$$

$$\ddot{\dot{a}}_{ij}^{k}=\begin{cases}\widetilde{a}_{i'j'},\ \text{если}\ (i=\mu\wedge i'=\alpha)\vee(i=(\mu+c)\wedge j'=\beta),\\ 0\ \text{в противном случае},\ i=\overline{1,\eta_k},\end{cases}$$

для $j=\overline{1,\varepsilon_k}$, $k=\overline{1,l}$, где $\vee$, $\wedge$ — знаки логических «или» и «и».

Приведем алгоритм формирования данных.

Алгоритм 4.1. Формирование f^k, $\dot{A}^k$, $\ddot{A}^k$, $\dddot{A}^k$, $\ddot{\dot{A}}^k$

1. Для $\{k \mid k=\overline{1,l}\}$ выполнить пп. 2-14.

2. $\dot{A}^k\leftarrow 0$, $\ddot{A}^k\leftarrow 0$, $\dddot{A}^k\leftarrow 0$, $\ddot{\dot{A}}^k\leftarrow 0$.

3. Для $\{j \mid j=\overline{1,\varepsilon_k}, j\equiv u_{i'j',k}^{\alpha\beta}\}$ выполнить пп. 4-13.

4. $f_j^k\leftarrow h_{i'j'}$.

5. Для $\{\mu \mid \mu=\overline{1,\eta_k-1}\}$, пока $\xi_\mu^k\neq\alpha$, перейти к п. 5.

6. $\ddot{a}_{\mu j}^{k}\leftarrow\widetilde{a}_{i'j'}$.

7. Если $i'\neq\alpha$, то $\dddot{a}_{\mu j}^{k}\leftarrow\widetilde{a}_{i'j'}$.

8. Если $i'=\alpha$, то $\ddot{\dot{a}}_{\mu j}^{k}\leftarrow\widetilde{a}_{i'j'}$.

9. Для $\{i \mid i=\overline{\mu,\eta_k}\}$, пока $\xi_i^k\neq\beta$, выполнить $\dot{a}_{ij}^{k}\leftarrow\widetilde{a}_{i'j'}$, перейти к п. 9.

10. $\ddot{a}_{ij}^{k}\leftarrow\widetilde{a}_{i'j'}$.

11. Если $j'\neq\beta$, то $\dddot{a}_{ij}^{k}\leftarrow\widetilde{a}_{i'j'}$.

12. Если $j'=\beta$, то $\ddot{\dot{a}}_{ij}^{k}\leftarrow\widetilde{a}_{i'j'}$.

13. Перейти к п. 3. ! Конец цикла по j

14. Перейти к п. 1. ! Конец цикла по k

15. Конец алгоритма.

Пусть $W^k = \| w_i^k \|_{\eta_k - 1}$, $B^k = \| b_i^k \|_{\eta_k}$, $B = \| b_i \|_n$, $k = \overline{1,l}$, — соответственно векторы ограничений на грузоподъемность транспортного средства или пропускную способность канала на дугах маршрута m_k; суммарное число транспортных блоков, которое можно обработать в транзитном узле i на маршруте m_k; пропускные способности узлов сети по обработке транзитных транспортных блоков.

Введем переменные x_j^k, $j = \overline{1,\varepsilon_k}$, $k = \overline{1,l}$, где

$$x_j^k = \begin{cases} \text{одному из значений } 1/\tilde{a}_{i'j'}, 2/\tilde{a}_{i'j'}, \ldots, \tilde{a}_{i'j'}/\tilde{a}_{i'j'}, \text{ если поток } \tilde{a}_{i'j'} \\ \text{с номером } j \text{ выбран на маршрут } m_k, \\ 0 \text{ в противном случае.} \end{cases}$$

С учетом вышеизложенного ограничения формулы (4.4), (4.5) запишем в виде

$$\sum_{j=1}^{\varepsilon_k} \dot{a}_{ij}^k x_{f_j^k}^k \le w_i^k, \; i = \overline{1, \eta_k - 1}, k = \overline{1,l}, \tag{4.10}$$

$$\sum_{j=1}^{\varepsilon_k} \ddot{a}_{ij}^k x_{f_j^k}^k \le b_i^k, \; i = \overline{1,\eta_k}, \; k = \overline{1,l}. \tag{4.11}$$

Обозначим $\sum_{j=1}^{\varepsilon_k} \ddot{a}_{ij}^k x_{f_j^k}^k = \psi_{\xi_i^k}^k$, $i = \overline{1,\eta_k}$, $k = \overline{1,l}$, $\sum_{j=1}^{\varepsilon_k} \ddot{a}_{ij}^k x_{f_j^k}^k = \lambda_{\xi_i^k}^k$, $i = \overline{1,\eta_k}$, $k = \overline{1,l}$.

Тогда ограничения (4.2) и (4.3) запишем так:

$$\sum_{k=1}^{l} \lambda_i^k = \sum_{j=1}^{n} (\tilde{a}_{ij} + \tilde{a}_{ji}), \; i = \overline{1,n}, \tag{4.12}$$

$$\frac{1}{2} \sum_{k=1}^{l} \psi_i^k \le b_i, \; i = \overline{1,n}. \tag{4.13}$$

Ограничения на запрет разветвления потоков (4.8) можно записать в виде

$$x_j^k = \begin{cases} 1, \text{если поток } \tilde{a}_{i'j'} \text{ с номером } j \text{ выбран на маршрут } m_k, \\ 0 \text{ в противном случае.} \end{cases} \tag{4.14}$$

Обозначим суммарный поток по дуге маршрута m_k через

$$\gamma_i^k = \sum_{j=1}^{\varepsilon_k} \dot{a}_{ij}^k x_{f_j^k}^k, \; i = \overline{1, \eta_k - 1}, k = \overline{1,l},$$

тогда ограничение (4.6) на среднюю задержку в передаче транспортных блоков

запишем так:

$$t_{av} = 1/\tilde{a}_{\Sigma} \sum_{k=1}^{l} \sum_{i=1}^{\eta_k - 1} \gamma_i^k / (w_i^k - \gamma_i^k) \le T_{\max}. \qquad (4.15)$$

Построим целевую функцию задачи: требуется минимизировать количество l наиболее «дешевых» загруженных маршрутов из множества $\{m_k\}$ при максимальной загрузке дуг (участков) этих маршрутов. Под «дешевыми» маршрутами понимаются такие маршруты, у которых самые низкие среднегодовые приведенные затраты на транспортировку и обработку потоков. Предполагается, что функции приведенных затрат C_{tr}^k заданы для каждого маршрута из $\{m_k\}$. Для транспортных сетей характерна зависимость таких функций от рабочего парка и грузоподъемности транспортных средств на маршруте, а для сетей передачи данных — от пропускной способности и длины каналов связи. Поэтому для реальных сетей приведенные затраты будут одинаковыми как для максимально загруженного маршрута, так и слабо загруженного или вообще не загруженного. Следует отметить, что функции удельной стоимости транспортировки и обработки единицы потока (одного транспортного блока) здесь применять нельзя, так как она меньше у маршрутов с большей пропускной способностью, что неизбежно приведет к загрузке более «дорогих» маршрутов. Кроме того, следует иметь в виду, что верхняя граница числа маршрутов l в множестве $\{m_k\}$ при решении задачи не фиксируется — после решения задачи она может оказаться больше заданной, если в процессе решения будет разрешено генерировать и вводить новые маршруты при невозможности распределения всех потоков.

Запишем целевую функцию

$$\underbrace{\min}_{|\{m_k\}|, C_{tr}^k} \left\{ \{m_k\} \;\middle|\; \sum_{j=1}^{\varepsilon_k} \dot{a}_{ij}^k x_{f_j^k}^k \Rightarrow \max, i = \overline{1, \eta_k - 1}, k = \overline{1, l} \right\}. \qquad (4.16)$$

Итак, в результате преобразования задачи (4.1)-(4.8) ее решение свелось к некоторой последовательности одновременного решения l-многомерных задач о ранце (4.16) с блочными ограничениями (4.10), (4.11) размерностью $(\eta_k - 1) \times \varepsilon_k$ и $\eta_k \times \varepsilon_k$, связывающими ограничениями (4.12), (4.13) и ограниче-

ниями (4.14), (4.15) и $t_{ij} \le T_{ij}$, $ij \in S$. Необходимость неоднократного (итеративного) решения задачи связана с тем, что на одной итерации все потоки, заданные матрицей $\tilde{A}$, могут быть не распределены по кратчайшим путям из-за нарушения каких-либо ограничений. В этом случае для нераспределенных потоков выбираются новые пути, отличные от первоначальных, определяются новые подмножества $\{u_{ij,k}^{\alpha\beta}\}$ и вновь решается ранцевая задача с учетом оставшегося резерва ограничений. Итеративный процесс продолжается до тех пор, пока на сети не будут распределены все потоки. Если все потоки распределить не удается из-за нарушения отдельных ограничений, то, по желанию проектировщика, в интерактивном режиме могут быть введены ослабления для всех или отдельных ограничений.

Полученная комбинаторная задача также является NP-трудной [45], поэтому для ее решения в различных вариантах и режимах предложен ряд алгоритмов, существенно использующих специфику структуры задачи, абстрактные типы данных и приемы, характерные для эвристических алгоритмов решения многомерной задачи о ранце (подразделы 4.3-4.6). Алгоритмы позволяют за приемлемое время получить рациональные, с точки зрения проектировщика сети, решения задачи.

4.2. Задача распределения и маршрутизации потоков с заданными тарифами в узлах и на дугах

Как правило в математических моделях, описывающих процессы обработки и транспортировки многопродуктовых потоков затраты связываются с величиной потока по дугам сети или путям передачи потока. Для сетей передачи данных, где дуги ассоциируются с каналами связи, такие постановки оказываются достаточно приемлемыми. В случае же транспортных сетей очень трудно адекватно определить функции транспортных затрат, а значит, и получить в результате решения задачи достоверный ответ. В математической модели задачи распределения и маршрутизации потоков транспортных блоков, приведенной в

предыдущем подразделе, объемы и пути распределения потоков связываются с множеством искомых «оптимальных» маршрутов транспортных средств или каналов связи. Такая постановка задачи характерна при проектировании новой или реконструкции существующей сети перевозок или сети передачи данных. В этом случае для каждого определенного в результате решения маршрута легко рассчитать требуемые затраты (например, среднегодовые приведенные затраты) и получить более достоверную оценку транспортных затрат для всей сети. Если транспортные услуги предоставляются сторонними транспортными предприятиями (компаниями) или провайдерами сети передачи данных, то задача распределения потоков упрощается, так как в качестве функций транспортных затрат можно использовать тарифы на перевозку или передачу одного транспортного блока от отправителя к получателю.

Целью настоящего подраздела является построение математической модели задачи распределения транспортных блоков со смешанными вложениями при использовании заданных тарифов на транспортировку и обработку потоков. Показано, что такая задача может быть сформулирована в виде задачи целочисленного линейного программирования (ЦЛП) с блочной структурой, связывающими ограничениями и дополнительными ограничениями на время доставки мелкопартионных корреспонденций получателю. Отмечаются особенности решения задачи при использовании известных методов ЦЛП и пакетов прикладных программ.

4.2.1. Постановка и математическая модель задачи

Рассмотрим постановку задачи, когда транспортные услуги предоставляются Q сторонними транспортными предприятиями (компаниями) [26, 124]. В этом случае в качестве функций транспортных затрат C_{tr}^{k} можно использовать тарифы на транспортировку одного транспортного блока (физического или виртуального контейнера). Без потери общности предположим, что все транспортные компании могут оказывать услуги по транспортировке потоков между всеми парами узлов в сети. Тогда задачу (4.1)-(4.8) можно рассматривать на

транспортной мультисети $G(N,P)$ с $|P|=l\le Q(n^2-n)$ ориентированными дугами, когда маршрутная сеть $G(N,P_M)$ полностью совпадает с $G(N,P)$, так как все маршруты представлены одной дугой. Получаем, что $l=|\{m_k\}|\le Q(n^2-n)$, а индексы дуг $\eta\xi\in P$ совпадают с индексами дуг $\alpha\beta\in P_M$. Для всех Q предприятий, занумерованных индексом $k=1,2,\ldots,Q$ заданы тарифы $C_{tr,k}^{\alpha\beta}$ на транспортировку единицы потока и ограничения $W_{\alpha\beta}^{k}$ на провозные возможности из узла α в узел β, $\alpha,\beta=\overline{1,n}$, $\alpha\ne\beta$. Для узлов сети заданы тарифы C_{load}^{α} на обработку единицы потока и ограничения b_α на величину транзитного потока в узле α, $\alpha=\overline{1,n}$. Задача (4.1)-(4.7) упрощается, так как проектировать маршруты не нужно и можно не учитывать ограничения (4.5) и (4.6). Транспортные тарифы $C_{tr,k}^{\alpha\beta}$ и тарифы на обработку транспортных блоков в узлах сети C_{load}^{α} должны быть приведены к одной шкале и определять, например, среднегодовые приведенные затраты. Пусть $u_{ij,k}^{\alpha\beta}$ — неизвестный поток транспортных блоков из i в j, проходящий по дуге $\alpha\beta\in\{q_k\}$, где $\{q_k\}$ — множество дуг k-го предприятия $|\{q_k\}|\le n^2-n$.

Исходная задача преобразуется к следующему виду: требуется найти минимум функции

$$\sum_{k=1}^{Q}\sum_{\alpha\beta\in\{q_k\}}C_{tr,k}^{\alpha\beta}\cdot\left(\sum_{ij\in S}u_{ij,k}^{\alpha\beta}\right)+\sum_{\alpha=1}^{n}C_{load}^{\alpha}\cdot\left(\sum_{k=1}^{Q}\sum_{\substack{\beta=1,\\ \beta\ne\alpha}}^{n}\sum_{ij\in S}(u_{ij,k}^{\alpha\beta}+u_{ij,k}^{\beta\alpha})\right),\qquad(4.17)$$

при ограничениях

$$\sum_{k=1}^{Q}\sum_{\substack{\beta=1,\\ \beta\ne\alpha}}^{n}u_{ij,k}^{\alpha\beta}-\sum_{k=1}^{Q}\sum_{\substack{\beta=1,\\ \beta\ne\alpha}}^{n}u_{ij,k}^{\beta\alpha}=\begin{cases}\tilde a_{ij}\ \text{при}\ i=\alpha,\\ 0\ \text{при}\ i\ne\alpha,\ j\ne\alpha,\\ -\tilde a_{ij}\ \text{при}\ j=\alpha,\ \text{для}\ \alpha=\overline{1,n},\ ij\in S,\end{cases}\qquad(4.18)$$

$$\sum_{k=1}^{Q}\sum_{\substack{\beta=1,\\ \beta\ne\alpha}}^{n}\sum_{ij\in S}(u_{ij,k}^{\alpha\beta}+u_{ij,k}^{\beta\alpha})-\sum_{\substack{j=1,\\ j\ne\alpha}}^{n}(\tilde a_{\alpha j}+\tilde a_{j\alpha})\le 2b_\alpha,\ \alpha=\overline{1,n},\qquad(4.19)$$

$$\sum_{ij\in S}u_{ij,k}^{\alpha\beta}\le W_{\alpha\beta}^{k}\ \forall\,\alpha\beta\in\{q_k\},k=\overline{1,Q},\qquad(4.20)$$

$$u_{ij,k}^{\alpha\beta} \geq 0 \text{ - целые числа,} \qquad (4.21)$$

и ограничениях на время доставки мелкопартионных корреспонденций получателю $t_{ij} \leq T_{ij}$, $ij \in S$.

Дополнительные ограничения (4.8) запишутся в виде

$$u_{ij,k}^{\alpha\beta} = \begin{cases} \tilde{a}_{ij}, \text{если поток проходит по дуге } \alpha\beta \in \{q_k\}, \\ 0 \text{ в противном случае.} \end{cases} \qquad (4.22)$$

Задача (4.17)-(4.21) максимально содержит до $n(n^2-n) + n + Q(n^2-n) + Q(n^2-n)^2 = Q(n^2-n)[(n^2-n)+1] + n(n^2-n) + n$ ограничений (4.18) + (4.19) + (4.20) + (4.21) и до $Q(n^2-n)^2$ переменных. Нетрудно видеть, что ограничения (4.18) можно записать в виде

$$\sum_{k=1}^{Q} \sum_{\substack{\beta=1, \\ \beta\neq\alpha}}^{n} \sum_{ij\in S} u_{ij,k}^{\alpha\beta} - \sum_{\substack{m=1, \\ m\neq\alpha}}^{n} \tilde{a}_{\alpha m} = \sum_{k=1}^{Q} \sum_{\substack{\beta=1, \\ \beta\neq\alpha}}^{n} \sum_{ij\in S} u_{ij,k}^{\beta\alpha} - \sum_{\substack{m=1, \\ m\neq\alpha}}^{n} \tilde{a}_{m\alpha}, \ \alpha = \overline{1,n},$$

а для ограничений (4.19) упростить запись:

$$\sum_{k=1}^{Q} \sum_{\substack{\beta=1, \\ \beta\neq\alpha}}^{n} \sum_{ij\in S} u_{ij,k}^{\alpha\beta} - \sum_{\substack{m=1, \\ m\neq\alpha}}^{n} \tilde{a}_{\alpha m} \leq b_{\alpha}, \ \alpha = \overline{1,n}.$$

Тогда максимальное количество ограничений задачи сократится до $Q(n^2-n)[(n^2-n)+1] + 2n$.

4.2.2. Полиномиальное преобразование задачи с тарифами к задаче целочисленного программирования

Преобразуем (по Карпу) задачу с тарифами (4.17)-(4.21) в целочисленную задачу линейного программирования с блочной структурой и связывающими ограничениями, когда в качестве матрицы коэффициентов ограничений выступает матрица инциденций узлы-дуги ориентированного графа. Для этого рассмотрим расширенную сеть $\tilde{G}(\tilde{N}, \tilde{P})$, в которой каждый узел сети G заменен двумя — левым и правым, причем каждая пара узлов «левый - правый» в сети $\tilde{G}$ имеют номера i и $i+n$, где i — номер узла в сети G (рис. 4.1). Все дуги, входящие в узел сети G, заменяются на дуги, входящие в левый узел сети $\tilde{G}$;

дуги, выходящие из узла сети G, заменяются на дуги, выходящие из правого узла сети $\tilde{G}$.

Каждый левый и правый узлы сети $\tilde{G}$, соединяются фиктивной дугой, стоимость транспортировки единицы потока по которой является стоимостью обработки единицы потока в расщепленном узле. На сети $\tilde{G}$ в качестве источников потоков выступают левые узлы, а в качестве стоков — правые. Пусть $\tilde{n} = n + n$ — число узлов расширенной сети, $\tilde{l} \le Q(n^2 - n) + n$ — число ее дуг.

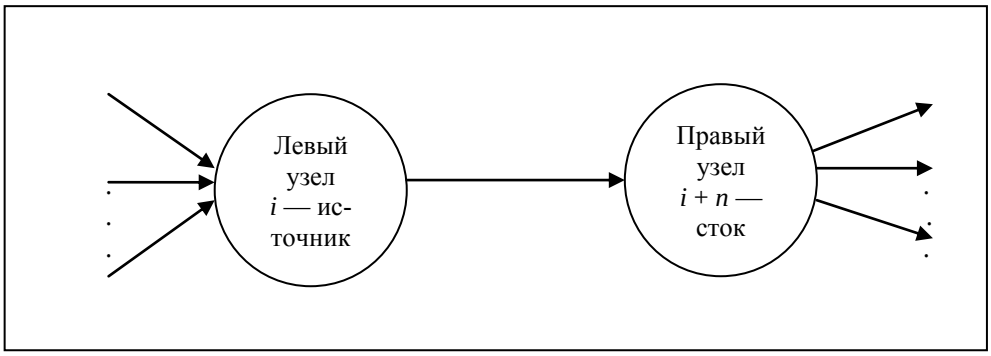


Рис. 4.1. Фрагмент расширенной сети $\tilde{G}$

Для формулировки задачи определим следующие матрицы и векторы. Составим матрицу инциденций $E = \| e_{ij} \|_{\tilde{n} \times \tilde{l}}$ узлы-дуги для расширенной сети таким образом, чтобы первые n ее строк соответствовали номерам узлов источников (левым узлам), а первые n столбцов — фиктивным дугам. Остальные n строк матрицы E соответствуют узлам стокам (правым узлам), а $\tilde{l} - n$ столбцов — заданным дугам. Матрица E имеет вид, показанный на рис. 4.2.

Дуги		1	…	…	n	$n+1$	…	$\tilde{l}$
Узлы источники	1	-1	0	…	0	{1,0}	…	{1,0}
	…	0	-1	…	0	{1,0}	…	{1,0}
	…	0	…	-1	0	{1,0}	…	{1,0}
	n	0	…	0	-1	{1,0}	…	{1,0}
Узлы стоки	$n+1$	1	0	…	0	{-1,0}	…	{-1,0}
	…	0	1	…	0	{-1,0}	…	{-1,0}
	…	0	…	1	0	{-1,0}	…	{-1,0}
	$2n$	0	…	0	1	{-1,0}	…	{-1,0}

Рис. 4.2. Структура матрицы инциденций

В каждом столбце матрицы имеется только два ненулевых элемента с противоположными знаками, а фигурные скобки означают, что элемент матрицы может принимать только одно из двух значений.

Эту матрицу будем использовать для записи условий сохранения потоков в узлах сети. Элементы матрицы

$$e_{ij} = \begin{cases} 1, \text{если дуга } j \text{ направлена к узлу } i, \\ -1, \text{если дуга } j \text{ направлена от узла } i, \\ 0 \text{ в противном случае.} \end{cases}$$

Пусть $V^i = (v_1^i, v_2^i, ..., v_{\tilde{n}}^i)^T$ — векторы-столбцы потребностей узлов в потоках $\tilde{a}_{ij}$, $j = \overline{1,n}$, $i \neq j$, исходящих из узлов $i = \overline{1,n}$, где знак «^{T}» означает транспонирование, а

$$v_\xi^i = \begin{cases} -\sum_{m=1}^{n} \tilde{a}_{im}, \text{если } \xi = i, \\ \tilde{a}_{ij}, \text{если } \xi = j + n, \\ 0 \text{ в противном случае,} \\ i = \overline{1,n},\ j = \overline{1,n}, \xi = \overline{1,\tilde{n}}. \end{cases}$$

Введем переменные $X^i = \| x_\mu^i \|_{\tilde{l}}^T$ — векторы-столбцы неизвестных потоков $x_1^i, x_2^i, ..., x_{\tilde{l}}^i$ по дугам сети от исходящих потоков из узлов i, $i = \overline{1,n}$. Тогда условия сохранения потоков для всех потоков, исходящих из узла i запишутся в виде $E[\tilde{n}, \tilde{l}\,]X^i[\tilde{l}\,] = V^i[\tilde{n}]$, $i = \overline{1,n}$.

Для записи ограничений на пропускные способности узлов и дуг сети определим векторы-столбцы $B = \| b_i \|_{\tilde{l}}^T$ и $F = \| f_i \|_{\tilde{l}}^T$, где b_i и f_i, $i = \overline{1,\tilde{l}}$ — суть целые неотрицательные числа. В векторе B первые n элементов содержат значения ограничений на обработку транзитных потоков в узлах сети, а в остальных $\tilde{l} - n$ элементах записаны значения ограничений на провозные возможности транспортных предприятий. Вектор F формируется следующим образом: $F = (\sum_{j=1}^{n}(\tilde{a}_{1j} + \tilde{a}_{j1}), ..., \sum_{j=1}^{n}(\tilde{a}_{nj} + \tilde{a}_{jn}), 0, ..., 0)^T$ и содержит n первых ненулевых значений и $\tilde{l} - n$ нулей с позиции $n+1$.

Пусть $C^i = \| c_j^i \|_{\tilde{l}}$, $i = \overline{1,n}$ — векторы-строки, в которых первые n элементов $c_1^i,\ldots, c_n^i$ определяют тарифы на обработку единицы потока в узлах сети с номерами $1,\ldots, n$, а остальные элементы $c_{n+1}^i,\ldots, c_{\tilde{l}}^i$ определяют тарифы на транспортировку единицы потока по дугам с номерами от $n+1$ до $\tilde{l}$. Без потери общности предположим, что тарифы на транспортировку одного транспортного блока не зависят от источника потока (все потоки однородны, т.е. все транспортные блоки имеют одинаковый размер) и для данного транспортного предприятия k, $k = \overline{1,Q}$ определяются только узлом отправления i и узлом назначения j. Примем $C^i = C$, $i = \overline{1,n}$ и будем формировать только одну вектор-строку тарифов. Тарифы на обработку одного транспортного блока не связаны с транспортными предприятиями и могут отличаться для разных типов узлов сети.

Окончательно сформулированная задача примет вид: найти

$$\min C[\tilde{l}] \cdot \sum_{i=1}^{n} X^i[\tilde{l}], \tag{4.23}$$

при ограничениях

$$\sum_{i=1}^{n} X^i[\tilde{l}] - F[\tilde{l}] \le B[\tilde{l}], \tag{4.24}$$

$$E[\tilde{n}, \tilde{l}]X^i[\tilde{l}] = V^i[\tilde{n}],\ i = \overline{1,n}, \tag{4.25}$$

$$X^i[\tilde{l}] \ge 0, i = \overline{1,n} \text{ и целые}, \tag{4.26}$$

и дополнительных ограничениях на время доставки мелкопартионных корреспонденций получателю

$$t_{ij} \le T_{ij},\ \forall ij \in S. \tag{4.27}$$

и на запрет разветвления потоков

$$x_\mu^i = \begin{cases} \sum_{ij} \tilde{a}_{ij}, \text{где суммируются потоки из } i \text{ в } j,\ j = \overline{1,n}, \\ \text{которые проходят по дуге } \mu,\ i = \overline{1,n},\ \mu = \overline{1,\tilde{l}}, \\ 0 \text{ в противном случае.} \end{cases} \tag{4.28}$$

Задача (4.23)-(4.26) содержит $2n+1$ векторных ограничений и до

$(Q(n^2-n)+n)\cdot n$ переменных. Векторно-матричная запись задачи позволяет использовать программно-аппаратную архитектуру CUDA (Compute Unified Device Architecture) параллельных вычислений на графических процессорах GPU (Graphics Processing Unit) и значительно (в перспективе в десятки и сотни раз) сократить время ее решения [316]. Задача (4.23)-(4.26) по сравнению с задачей (4.17)-(4.21) максимально содержит до $n+Q(n^2-n)$ + $n(n+Q(n^2-n))$ + $n(n+Q(n^2-n))$ = $Q(n^2-n)(2n+1)+2n^2+n$ << $Q(n^2-n)[(n^2-n)+1]+2n$ невекторных ограничений (4.24) + (4.25) + (4.26). Так, например, при Q = 10 и n = 100 в задаче (4.17)-(4.21) может быть до 9801000000~10x10^9 переменных и до 980199200~ 1x10^9 ограничений, а в задаче (4.23)-(4.26) — до 9910000~10x10^6 переменных и до 19919100~ 20x10^6 ограничений.

Известно, что обе задачи (4.17)-(4.21) и (4.23)-(4.26) являются NP-трудными, так как их соответствующие задачи распознавания NP-полные [45, 249]. Покажем, что любая индивидуальная задача (4.17)-(4.21) может быть преобразована в индивидуальную задачу (4.23)-(4.26) за полиномиальное время. Для этого приведем один из возможных способов формирования по входным данным задачи (4.17)-(4.21) матрицы инциденций $E[\tilde{n},\tilde{l}]$, векторов сохранения потоков $V^i[\tilde{n}]$, $i=\overline{1,n}$, вектора тарифов C и векторов $F[\tilde{l}]$ и $B[\tilde{l}]$ для записи ограничений (4.24).

Пусть $A=\| a_{ij} \|_{n\times n}$ и трехмерная $D=\| d_{ijk} \|_{n\times n\times Q}$ — целочисленные матрицы потоков и тарифов Q предприятий. Подматрица $D=\| d_{ijk} \|_{n\times n\times 1}$ на главной диагонали содержит тарифы на обработку единицы потока в соответствующем узле, главные диагонали подматриц для $k=2,3,\ldots,Q$ содержат нули. Все остальные элементы d_{ijk} матрицы D содержат тарифы на транспортировку единицы потока из i в j предприятия k, $i,j=\overline{1,n}$, $i\neq j$, $k=\overline{1,Q}$. Если предприятие k не оказывает услуги по транспортировке из i в j, то $d_{ijk}=0$, а соответствующая дуга из i в j для предприятия k отсутствует. Задана также трехмерная цело-

численная матрица $W = \| w_{ijk} \|_{n\times n\times Q}$ провозных возможностей Q предприятий в тех же единицах измерения потока — транспортных блоках. Подматрица $W = \| w_{ijk} \|_{n\times n\times 1}$ на главной диагонали содержит заданные значения ограничений на пропускные способности узлов по обработке транзитных потоков, главные диагонали подматриц для $k = 2,3,\ldots,Q$ содержат нули. Остальные элементы w_{ijk} матрицы W содержат заданные значения ограничений на провозные возможности предприятий из i в j, $i, j = \overline{1,n}$, $i \neq j$, $k = \overline{1,Q}$. Все ненулевые и нулевые элементы матрицы W строго соответствуют ненулевым и нулевым элементам матрицы D и наоборот. Таким образом, матрицы D и W полностью определяют структуру ориентированной мультисети $G(N,P)$ с $|N| = n$ узлами и $|P| = l \le Q(n^2 - n)$ дугами.

Приведем алгоритм формирования входных данных для индивидуальной задачи (4.23)-(4.26).

Алгоритм 4.2. Формирование $E[\tilde{n},\tilde{l}]$, $V^i[\tilde{n}]$, $i = \overline{1,n}$, C, $F[\tilde{l}]$, $B[\tilde{l}]$

1. $E \leftarrow 0$; $V \leftarrow 0$; $F \leftarrow 0$; $l \leftarrow 0$.

2. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 3-17.

3. $s \leftarrow 0$.

4. Для $\{ j \mid j = \overline{1,n} \}$ выполнить пп. 5-16.

5. $f_i \leftarrow f_i + a_{ij} + a_{ji}$.

6. Если $i \neq j$, то перейти к п. 7, иначе перейти к п. 12.

7. $v_{j+n,i} \leftarrow a_{ij}$.

8. Для $\{ k \mid k = \overline{1,Q} \}$ выполнить пп. 9-11 .

9. Если $d_{jik} \neq 0$, то перейти к п. 10, иначе перейти к п. 11.

10. $l \leftarrow l+1$; $e_{i,n+l} \leftarrow 1$; $e_{j+n,n+l} \leftarrow -1$; $c_{n+l} \leftarrow d_{jik}$; $b_{n+l} \leftarrow w_{jik}$.

11. Перейти к п. 8. *** Конец цикла по k

12. Для $\{ m \mid m = \overline{1,n} \}$ выполнить пп. 13-14 .

13. $s \leftarrow s + a_{im}$.

14. Перейти к п. 12. *** Конец цикла по m

15. $v_{i,i} \leftarrow -s$; $e_{i,j} \leftarrow -1$; $e_{i+n,j} \leftarrow 1$; $c_i \leftarrow d_{ij1}$; $b_i \leftarrow w_{ij1}$.

16. Перейти к п. 4. *** Конец цикла по j

17. Перейти к п. 2. *** Конец цикла по i

18. Конец алгоритма.

В записи алгоритма векторы $V^i[\tilde{n}]$, $i = \overline{1,n}$ представлены матрицей $V = \| v_{ij} \|_{\tilde{n} \times n}$. Из алгоритма видно, что временная сложность преобразования любой индивидуальной задачи (4.17)-(4.21) в индивидуальную задачу (4.23)-(4.26) составляет в худшем случае $O[n((n-1)Q+n)] = O[(Q+1)n^2 - Qn]$.

Сформулируем основной результат в виде следующей теоремы.

Теорема 4.1. Любая индивидуальная задача вида (4.17)-(4.21) может быть за полиномиальное время $O[(Q+1)n^2 - Qn]$ преобразована в индивидуальную задачу целочисленного линейного программирования вида (4.23)-(4.26).

Доказательство теоремы непосредственно следует из корректности работы алгоритма 4.1 преобразования входных данных экземпляра задачи (4.17)-(4.21) во входные данные задачи (4.23)-(4.26). ■

Следствие 4.1. Если существует допустимое (не обязательно оптимальное) решение исходной задачи (4.17)-(4.21), то оно совпадает с решением преобразованной задачи (4.23)-(4.26). Доказательство очевидно. ■

На основании вышеизложенного можно сделать следующие выводы и определить направления дальнейших исследований.

1. Задача с заданными тарифами на дугах и в узлах может быть за полиномиальное время преобразована к задаче целочисленного линейного программирования. В случае, когда ограничения на время доставки не являются критическими, то ее можно решать с применением известных методов и пакетов программ ЦЛП, однако для установления границ их эффективности и анализа по-

лучаемых решений необходимо проведение обширного вычислительного эксперимента на общедоступных серверах, таких как NEOS, GAMS и др. с программами Gurobi, Linear Program Solver, Simplex OPTIMA, CPLEX, MINTO и пр.

2. При необходимости учета ограничений на время доставки мелкопартионных корреспонденций получателю существующие методы и пакеты программ не могут быть непосредственно применены — требуется проведение дополнительных исследований по возможной их модификации и доработке с учетом новых ограничений.

4.3. Задачи построения кратчайших путей

Задачи построения кратчайших путей (КП) являются одними из фундаментальных задач оптимизации на графах и сетях и часто используется как подзадачи во многих приложениях в различных областях естествознания, а также в обыденной жизни. В большей части работ, посвященных вопросам нахождения КП, авторы рассматривают задачу построения КП по одному критерию — минимум или максимум суммы длин дуг в пути. При этом под длиной дуги понимается некоторый ее параметр, например, расстояние по дуге, затраты при перевозке по дуге, пропускная способность дуги и т.п. Несмотря на многочисленность работ, связанных с проблемой нахождения КП, внимание к разработке эффективных по быстродействию алгоритмов построения КП не ослабевает (см. обзор и анализ методов и алгоритмов построения КП в приложении Г). Это, в первую очередь, объясняется тем, что в подавляющем большинстве случаев, время решения общей оптимизационной задачи в значительной степени определяется временем построения КП. В связи с этим рассмотрим новый двухкритериальный лексикографический алгоритм построения всех кратчайших путей в сети.

4.3.1. Постановка задачи и алгоритмы решения

Рассмотрим алгоритм построения КП, упорядоченных в лексикографиче-

ском порядке по двум критериям. Известно, что такие алгоритмы имеют полиномиальную трудоемкость в отличие от алгоритмов нахождения Парето - оптимальных кратчайших путей. Некоторые алгоритмы построения ступенчатых двухкритериальных путей рассматривались в работах [68, 49]. Идея алгоритма, предложенного в [68], близка к идее алгоритма Мура [394] и отличается особенностью реализации процедуры расстановки «пометок» с помощью использования логических операций над строками матриц, характеризующих сеть. Как показано в [4], асимптотическая трудоемкость алгоритмов такого типа не лучше, чем $O(n^3)$, где n — число узлов в сети. В работе [49], изложен алгоритм, строящий деревья кратчайших путей по двухступенчатому критерию: минимум ранга пути, при равенстве рангов — максимум пропускной способности. Где под рангом пути понимается число транзитных узлов или дуг в пути. Сущность алгоритма заключается в следующем. Пусть от некоторого узла i необходимо построить дерево КП до всех остальных узлов. Поскольку пути с рангом 1 уже построены, на первой итерации алгоритма строятся пути с рангом 2. На каждой последующей итерации строятся пути с рангом на единицу большим, чем на предшествующей. Итерация включает два вложенных цикла по числу узлов в сети, поэтому асимптотическая трудоемкость итерации составляет $O(n^2)$. Для построения дерева КП от узла i потребуется $O((q_i - 1)n^2)$ действий, где q_i — максимальный ранг КП, построенного от узла i. Нахождение деревьев КП от всех узлов потребует сложности порядка $O((q_{\max} - 1)n^3)$ в предположении, что для всех узлов $q_i = q_{\max}$.

Следует особо отметить, что для построения путей по ступенчатому критерию легко модифицировать любой известный алгоритм построения КП по одному критерию, например, алгоритм Флойда [252] или Дейкстры [240]. Как известно, эти алгоритмы имеют асимптотическую трудоемкость порядка $O(n^3)$ для построения КП от всех узлов.

Как правило, при решении большинства практических задач нахождения КП по двум критериям, наибольший интерес представляют алгоритмы, исполь-

зующие в качестве критериев оптимизации минимум транзитных узлов или дуг в пути (ранг пути) и минимум длины пути. В связи с этим, в настоящем подразделе предлагается улучшенная версия двухкритериального алгоритма построения всех КП в сети по указанным критериям, предложенного в работе [27]. Поскольку нас будет интересовать построение КП от всех узлов в сети, трудоемкость предложенного алгоритма будет сравниваться с трудоемкостью измененного для нахождения КП по таким же критериям алгоритма Флойда. Приведенные алгоритмы могут быть легко обобщены и на случай наличия большего числа упорядоченных критериев оптимизации.

Пусть задана простая неориентированная сеть $G(N,P)$ с множеством узлов N, $n=|N|$ и множеством дуг P, $p=|P|$. Каждой дуге $p_{kl}\in P$ поставлено в соответствие некоторое число $r_{kl}>0$, условно называемое ее длиной. Назовем число дуг, составляющих некоторый путь, рангом пути. Требуется построить КП между всеми узлами сети, удовлетворяющие критерию: минимум ранга пути, при равенстве рангов — минимум длины.

Идея предлагаемого алгоритма [28, 29] сходна с идеей алгоритма Беллмана - Шимбела [180, 433] и заключается в том, что на каждой условной l - ой итерации строятся все КП в сети с рангом до 2^l включительно. При этом на каждой последующей итерации используются все пути, построенные на предыдущих, и применяется эффективная техника представления абстрактных типов данных (АТД).

Пусть $R=\| r_{ij} \|_{n\times n}$ — топологическая матрица, r_{ij} — длина дуги p_{ij} (если дуги p_{ij} не существует, $r_{ij}=\infty$); $D=\| d_{ij} \|_{n\times n}$ — матрица длин построенных путей; $C=\| c_{ij} \|_{n\times n}$ — справочная матрица построенных путей, каждый элемент которой c_{ij}, $i\neq j$ определяет номер предпоследнего узла на кратчайшем пути от i до j, $c_{ii}=0$, $i=\overline{1,n}$; $Q=\| q_{ij} \|_{n\times n}$ — матрица рангов построенных путей. Первоначально D совпадает с R, а для путей с рангом 1 соответствующие элементы $q_{ij}=1$, $c_{ij}=i$. Для ненайденных путей $q_{ij}=\infty$, $c_{ij}=0$. (Знаки '$\leftarrow$', '$\wedge$' и

'$\vee$' соответственно означают операции присваивания, конъюнкции (логического «и») и дизъюнкции (логического «или»). Знаком '$***$' отделяются комментарии.)

Приведем алгоритм построения КП по ступенчатому критерию [174].

Алгоритм 4.3. SP1

1. $RMAX \leftarrow 1;\ \ l \leftarrow 0.$

2. $KU \leftarrow 0;\ \ l \leftarrow l+1.$

3. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 4-16.

4. $KS \leftarrow 0.$

5. Для $\{ j \mid j = \overline{1,n} \}$ выполнить пп. 6-14.

6. Если $(i \neq j) \wedge c_{ij} = 0$, то перейти к п. 7; иначе выполнить $KS \leftarrow KS + 1$ и перейти к п. 14.

7. Для $\{ k \mid k = \overline{1,n} \}$ выполнить пп. 8-12.

8. Если $Q(i,k) \le RMAX \wedge Q(k,j) \le RMAX$, то перейти к п. 9; иначе перейти к п. 12.

9. $RT \leftarrow Q(i,k) + Q(k,j).$

10. Если $RT < Q(i,j) \vee (RT = Q(i,j) \wedge D(i,j) > D(i,k) + D(k,j))$, то перейти к п. 11; иначе перейти к п. 12.

11. $D(i,j) \leftarrow D(i,k) + D(k,j);\ Q(i,j) \leftarrow RT;\ C(i,j) \leftarrow C(k,j).$

12. Перейти к п. 7. $***$ Конец цикла по k

13. Если $C(i,j) \neq 0$, то $KS \leftarrow KS + 1.$

14. Перейти к п. 5. $***$ Конец цикла по j

15. Если $KS = n$, то $KU \leftarrow KU + 1.$

16. Перейти к п. 3. $***$ Конец цикла по i

17. $RMAX \leftarrow 2RMAX.$

18. Если $RMAX \le (n-1) \wedge KU < n$, то перейти к п. 2, иначе к п. 19.

19. Если $RMAX > (n-1) \wedge KU < n$, то вывести сообщение о несвязности

сети.

20. Стоп.

Приведем измененный алгоритм Флойда для построения КП по ступенчатому критерию.

Алгоритм 4.4. Floyd

1. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 2-9.

2. Для $\{ j \mid j = \overline{1,n} \}$, если $i \neq j$ выполнить пп. 3-8.

3. Для $\{ k \mid k = \overline{1,n} \}$, если $k \neq j \neq i$ выполнить пп. 4-7.

4. Если $Q(j,i) < \infty \wedge Q(i,k) < \infty$, то перейти к п. 5; иначе перейти к п. 7.

5. Если

$$(Q(j,k) > Q(j,i) + Q(i,k)) \vee (Q(j,k) = Q(j,i) + Q(i,k) \wedge D(j,k) > D(j,i) + D(i,k)),$$

то перейти к п. 6; иначе перейти к п. 7.

6. $Q(j,k) \leftarrow Q(j,i) + Q(i,k);\ D(j,k) \leftarrow D(j,i) + D(i,k);\ C(j,k) \leftarrow C(i,k)$.

7. Перейти к п. 3.

8. Перейти к п. 2.

9. Перейти к п. 1.

10. Стоп.

В алгоритме SP1 в строке 2 начинается цикл, представляющий из себя условную итерацию, общее число которых для построения всех путей в сети накапливается в l. Условная итерация содержит три вложенных цикла по индексам i, j, k.

Пусть каждому номеру l, l = 0, 1, 2, ... условной итерации алгоритма SP1 поставлено во взаимно-однозначное соответствие множество $\{1+2^l,...,2^{l+1}\}$ рангов путей. Назовем это соответствие таблицей рангов. Корректность алгоритма следует из доказанных в [28] следующих утверждений.

Лемма 4.1. Пусть в алгоритме SP1 на l - ой итерации допускается построение путей с рангом большим, чем 2^l, тогда кратчайшие пути будут построены некорректно.

Доказательство. В строке 8 алгоритма во внутреннем цикле по k возможно $Q(i,k) < \infty$ и $Q(k, j) < \infty$. Пусть на некоторой итерации l были построены пути (i,k_1), (i,k_2), причем $2^l < Q(i,k_2) = Q(i,k_1) + Q(k_1,k_2) < \infty$ и номер k_2 больше номера k_1. Тогда, поскольку в цикле по k осуществляется перебор всех номеров узлов в возрастающем порядке, найдется узел с номером $k_3 > k_2 > k_1$, такой, что

$$Q(i,k_2) = Q(i,k_3) + Q(k_3,k_2) \text{ и}$$

$$D(i,k_2) = D(i,k_3) + D(k_3,k_2) < D(i,k_2) = D(i,k_1) + D(k_1,k_2), \quad (4.29)$$

но к моменту определения пути до k_2, $Q(i,k_3) = \infty$ или $Q(k_3,k_2) = \infty$, т.е. путь (i,k_3) или (k_3,k_2) еще не найден. На l - ой итерации путь (i,k_2) получил отметку $C(i,k_2) = k_1$ и на последующих итерациях рассматриваться не будет, но в силу выполнения (4.29) этот путь некорректен. ■

Теорема 4.2. На каждой условной итерации алгоритма SP1 строятся все кратчайшие пути в сети, соответствующие таблице рангов.

Доказательство. Поскольку на каждой l - ой итерации алгоритма нельзя строить пути с рангом, большим 2^l (лемма), используем для построения путей на этой итерации только те пути, которые построены на $(l-1)$ - ой итерации. Фактически это гарантируется использованием в цикле по k в строке 8 следующих операций сравнения: $Q(i,k) \le RMAX$, $Q(k, j) \le RMAX$, $RMAX = 2^l$, l = 0, 1, …, а первой условной итерации соответствует номер l = 0. Очевидно, что на каждой итерации алгоритма по l будут построены все пути в сети с рангами, строго соответствующими таблице рангов. Докажем, что построенные пути являются кратчайшими. Пусть на m - ой итерации получен путь (i, j), не удовлетворяющий критерию: минимум ранга, минимум длины. Значит, существует другой кратчайший путь, который проходит через узел u, причем либо $Q(i,u)$ либо $Q(u, j)$ не определены. Поскольку на m - ой итерации используются все пути, построенные на $(m-1)$ - ой итерации, то во внутреннем цикле по k выполняется исчерпывающий поиск путей между i и j среди всех путей с

множеством рангов $\{1+2^m,\ldots,2^{m+1}\}$ по ступенчатому критерию:

$$t = \arg\min\{Q(i,k)+Q(k,j)\}, k=\overline{1,n}, k\neq u;$$

$$D(i,j)=\min\{D(i,v)+D(v,j)\}, v\in t,$$

где t — множество узлов, пути через которые имеют одинаковый ранг. Следовательно, путь (i,j) через узел u мог быть построен только на последующих итерациях и имел бы больший ранг, что противоречит предположению о существовании другого кратчайшего пути. Теорема доказана. ■

Следствие. Для построения оптимального пути с рангом m $(m\geq 2)$ требуется $\lceil \log_2 m \rceil$ условных итераций алгоритма SP1, где знак $\lceil\cdot\rceil$ означает округление числа до большего целого. Доказательство очевидно. ■

В целях сокращения оперативной памяти, используемой алгоритмом SP1, матрицу рангов путей Q можно не хранить. В этом случае определять значения $Q(i,k)$, $Q(k,j)$, $Q(i,j)$ следует с помощью справочной матрицы C в следующей процедуре-функции, являющейся аналогом процедуры функции в языках программирования высокого уровня.

1. ВХОД $q(C,i,j)$.

2. $q\leftarrow 1$; $k\leftarrow j$.

3. Пока $C(i,k)\neq i \wedge C(i,k)\neq 0$ выполнять п. 4.

4. $k\leftarrow C(i,k)$; $q\leftarrow q+1$; перейти к п. 3.

5. Если $C(i,k)=0$, то $q\leftarrow\infty$.

6. ВОЗВРАТИТЬ (q).

Пусть обращение к процедуре-функции выполняется посредством употребления выражения $q(C,i,j)$, при этом подразумевается, что вместо $q(C,i,j)$ подставляется значение q, возвращаемое процедурой-функцией в шаге 6. Тогда в алгоритме SP1 измененные строки будут такими:

8. Если $q(C,i,k)\leq RMAX \wedge q(C,k,j)\leq RMAX$, то перейти к п. 9; иначе перейти к п. 12.

9. $RT\leftarrow q(C,i,k)+q(C,k,j)$.

10. Если $(RT < q(C,i,j)) \vee (RT = q(C,i,j) \wedge D(i,j) > D(i,k) + D(k,j))$, то перейти к п. 11; иначе перейти к п. 12.

11. $D(i,j) \leftarrow D(i,k) + D(k,j)$; $C(i,j) \leftarrow C(k,j)$.

Очевидно, что сокращение объема требуемой памяти заключается в некотором увеличении времени счета задачи из-за обращения к процедуре-функции.

Пусть $S = \lceil \log_2 m \rceil$, где m — максимальный ранг пути среди всех кратчайших путей, построенных в сети.

Теорема 4.3. Время, необходимое алгоритму SP1 для построения всех кратчайших путей в сети, растет пропорционально функции $Sn^3 - n^2(3S + c) + 2n(S + c)$, где c — некоторая константа.

Оценим трудоемкость алгоритма SP1 без использования процедуры-функции. Обозначим через $h_{k,i}$ — количество путей, построенных от узла i и имеющих ранг k. Тогда для построения всех кратчайших путей в сети число выходов алгоритма на внутренний цикл по k определится следующим образом:

$$\sum_{i=1}^{n} n - (1 + h_{1,i}) + \sum_{i=1}^{n} (n - (1 + h_{1,i} + h_{2,i})) + \ldots + \sum_{i=1}^{n} (n - (1 + h_{1,i} + h_{2,i} + h_{3,i} + \ldots + h_{2^{(S-1)},i})) =$$

$$= \sum_{i=1}^{n} (n - (1 + h_{1,i}) + n - (1 + h_{1,i} + h_{2,i}) + \ldots + n - (1 + h_{1,i} + h_{2,i} + h_{3,i} + \ldots + h_{2^{(S-1)},i})) =$$

$$= \sum_{i=1}^{n} (nS - (S + h_{1,i}S + h_{2,i}(S-1) + h_{3,i}(S-2) + h_{4,i}(S-2) + \ldots + h_{2^{(S-1)},i})) =$$

$$= \sum_{i=1}^{n} [(nS - S) - \sum_{l=0}^{S-1} (S-l) \sum_{k \in \{\lfloor 1+2^{(l-1)} \rfloor, \ldots, 2^l\}} h_{k,i}] = Sn(n-1) - \sum_{i=1}^{n} \sum_{l=0}^{S-1} (S-l) \sum_{k \in \{\lfloor 1+2^{(l-1)} \rfloor, \ldots, 2^l\}} h_{k,i},$$

где знак $\lfloor \cdot \rfloor$ — означает округление числа до меньшего целого.

Число выполнений цикла по k зависит от структуры исходной сети и с увеличением числа условных итераций по l алгоритма стремится к $(n-2)$. Поэтому дадим оценку сверху трудоемкости алгоритма в худшем случае:

$$T_{mp} = Sn(n-1)(n-2) - (n-2) \sum_{i=1}^{n} \sum_{l=0}^{S-1} (S-l) \sum_{k \in \{\lfloor 1+2^{(l-1)} \rfloor, \ldots, 2^l\}} h_{k,i}. \qquad (4.30)$$

Обозначим $\sum_{l=0}^{S-1} (S-l) \sum_{k \in \{\lfloor 1+2^{(l-1)} \rfloor, \ldots, 2^l\}} h_{k,i} = c$ в предположении, что $h_{k,i} = h_k$, для всех

$i \in N$, тогда (4.30) перепишется в виде

$$T_{mp} = Sn(n-1)(n-2) - (n-2)\sum_{i=1}^{n} c = Sn^3 - n^2(3S + c) + 2n(S + c).$$

Теорема доказана. ■

Учитывая, что асимптотическая трудоемкость алгоритма Флойда составляет всегда $O(n^3)$, то можно ожидать, что при увеличении плотности графа или сети быстродействие алгоритма SP1 будет лучше, чем быстродействие алгоритма Флойда.

Из анализа алгоритма SP1 видно, что его трудоемкость можно уменьшить за счет сокращения перебора узлов j в цикле в строке 5 при выборе узлов, до которых еще не найдены пути, и узлов k в цикле в строке 7 при выборе узлов, до которых пути уже найдены. Введем для этого следующие структуры данных. Пусть $A = \| a_{ij} \|_{n\times(n-1)}$ — матрица, содержащая номера узлов, до которых пути найдены, при этом в первой строке матрицы как в стеке находятся номера узлов, до которых найдены пути от первого узла, во второй – от второго узла и так далее. Введем также вектор $T = \| t_i \|_n$ для подсчета числа узлов, до которых найдены пути от узлов i, $i = \overline{1,n}$ и вектор $SP = \| sp_i \|_n$, содержащий указатели sp_i от узлов i, $i = \overline{1,n}$ на линейные односторонние списки, состоящие из элементов E. Каждый элемент E имеет два поля: F и AJ. Поле AJ содержит номер узла, до которого путь не найден, а поле F ссылку на следующий элемент. Последний элемент в списке, так же как и пустой список, имеют нулевые ссылки ($null$ - ссылки).

Приведем улучшенную версию алгоритма SP1 построения КП по ступенчатому критерию.

Алгоритм 4.5. SP2

1. $T \leftarrow 0$.

2. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 3-12.

3. Для $\{ j \mid j = \overline{1,n} \}$, если $j \neq i$ выполнить пп. 4-11.

4. Если $D(i, j) < \infty$, то перейти к п. 5, иначе перейти к п. 6.

5. $T(i) \leftarrow T(i) + 1;\ A(i, T(i)) \leftarrow j$; перейти к п. 11.

6. Образовать новый элемент $E(P)$.

7. Если $SP(i).F \neq null$, то выполнить п. 8, иначе выполнить п. 9.

8. $P2 \Rightarrow P1;\ P1 \Rightarrow P;\ P2.F \Rightarrow P$; перейти к п. 10.

9. $SP(i).F \Rightarrow P;\ P1 \Rightarrow P$; перейти к п. 10.

10. $P.AJ \leftarrow j$.

11. Перейти к п. 3. $***$ Конец цикла по j

12. Перейти к п. 2. $***$ Конец цикла по i

13. $RMAX \leftarrow 1;\ l \leftarrow 0$.

14. $KU \leftarrow 0;\ l \leftarrow l + 1$.

15. Для $\{ i \mid i = \overline{1, n} \}$ выполнить пп. 16-31.

16. Если $SP(i).F \neq null$, то перейти к п. 17, иначе $KU \leftarrow KU + 1$; перейти к п. 31.

17. $P \Rightarrow SP(i).F;\ P1 \Rightarrow SP(i)$.

18. Пока $P.F \neq null$, выполнить пп. 19-30.

19. $j \leftarrow P.AJ$.

20. Для $\{ m \mid m = \overline{1, T(i)} \}$ выполнить пп. 21-26.

21. $k \leftarrow A(i, m)$.

22. Если $Q(i, k) \le RMAX \wedge Q(k, j) \le RMAX$, то перейти к п. 23; иначе перейти к п. 26.

23. $RT \leftarrow Q(i, k) + Q(k, j)$.

24. Если $RT < Q(i, j) \vee (RT = Q(i, j) \wedge D(i, j) > D(i, k) + D(k, j))$, то перейти к п. 25; иначе перейти к п. 26.

25. $D(i, j) \leftarrow D(i, k) + D(k, j);\ Q(i, j) \leftarrow RT;\ C(i, j) \leftarrow C(k, j)$.

26. Перейти к п. 20. $***$ Конец цикла по m

27. Если $C(i, j) \neq 0$, то перейти к п. 28, иначе перейти к п. 29.

28. $T(i) \leftarrow T(i)+1$; $A(i,T(i)) \leftarrow j$; $P1.F \Rightarrow P.F$; $P3 \Rightarrow P$; $P \Rightarrow P.F$; Освободить элемент $E(P3)$. Перейти к п. 30.

29. $P1 \Rightarrow P$; $P \Rightarrow P.F$; Перейти к п. 30.

30. Перейти к п. 18. $***$ Конец цикла по j

31. Перейти к п. 15. $***$ Конец цикла по i

32. $RMAX \leftarrow 2RMAX$.

33. Если $RMAX \le (n-1) \wedge KU < n$, то перейти к п. 14, иначе к п. 34.

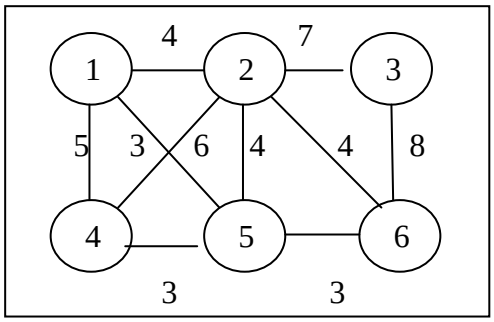
34. Если $RMAX > (n-1) \wedge KU < n$, то вывести сообщение о несвязности сети.

35. Стоп.

В строках 1-12 выполняется формирование начальных значений для T, A и SP, а в строках 13-35 содержится основное тело алгоритма. Вектор SP используется во внешних циклах по i и j для выбора узлов j, до которых пути от i до j еще не найдены (строки 15-31 и 18-30). Матрица A и вектор T используются во внутреннем цикле по m для выбора номеров узлов k, до которых пути от i до k уже найдены (строки 20-26). Если на данной итерации найден КП от i до j через какой либо k, то j вводится в соответствующую i-ю строку A, а из списка sp_i j-й узел выводится. По окончанию работы алгоритма списки sp_i будут пустыми, а строки матрицы A будут полностью заполнены (в случае, если сеть связна). Операции «Образовать новый элемент $E(P)$» в строке 6 и «Освободить элемент $E(P3)$» в строке 28 означают, соответственно, образование элемента E и установку на него указателя P и удаление элемента E на который ссылается указатель $P3$. Операции со знаком «$\Rightarrow$» устанавливают указатель в левой части выражения на элемент в правой части выражения. Выражения с точками, типа $SP(i).F$, $P.F$, $P.AJ$ означают соответствующие поля элементов, на которые ссылаются указатели $SP(i)$ и P. Если в строке 33 окажется, что $RMAX > (n-1)$, то при $KU < n$ исходная сеть несвязна.

На рис. 4.3 приведены пример сети и структуры данных, поясняющие ра-

боту алгоритма.



N	*SP*	*AJ*	*AF*	*AJ*	*AF*	*AJ*	*AF*
1	→	3	→	6			
2							
3	→	1	→	4	→	5	
4	→	3		6			
5	→	3					
6	→	1	→	4			

T
3
5
2
3
4
3

A				
2	4	5		
1	3	4	5	6
2	6			
1	2	5		
1	2	4	6	
2	3	5		

Рис. 4.3. Начальное состояние структур данных SP, T, и A для сети из 6 узлов

N	*SP*
1	
2	
3	
4	
5	
6	

T
5
5
5
5
5
5

A				
2	4	5	3	6
1	3	4	5	6
2	6	1	4	5
1	2	5	3	6
1	2	4	6	3
2	3	5	1	4

D						
N	1	2	3	4	5	6
1	0	4	11	5	3	6
2	4	0	7	6	4	4
3	11	7	0	13	11	8
4	5	6	13	0	3	6
5	3	4	11	3	0	3
6	6	4	8	6	3	0

C						
N	1	2	3	4	5	6
1	0	1	2	1	1	5
2	2	0	2	2	2	2
3	2	3	0	2	2	3
4	4	4	2	0	4	5
5	5	5	2	5	0	5
6	5	6	6	5	6	0

Q						
N	1	2	3	4	5	6
1	0	1	2	1	1	2
2	1	0	1	1	1	1
3	2	1	0	2	2	1
4	1	1	2	0	1	2
5	1	1	2	1	0	1
6	2	1	1	2	1	0

Рис. 4.4. Содержимое структур данных SP, T, A, D, C и Q после окончания работы алгоритма

В результате работы алгоритма за одну условную итерацию будут построены все КП в сети, а структуры SP, T, A, D, C и Q, будут содержать данные, показанные на рис. 4.4.

Для алгоритмов SP1, Floyd и SP2 были разработаны также модифицированные алгоритмы SP1S, FloydS и SP2S для случая, когда матрица R является симметричной, так как свойство симметричности позволяет существенно сократить время нахождения КП. Соответственно измененные строки в алгоритмах будут такими.

Для алгоритма SP1S:

3. Для $\{ i \mid i = \overline{1, n-1} \}$ выполнить пп. 4-16.

5. Для $\{ j \mid j = \overline{i+1, n} \}$ выполнить пп. 6-14.

6. Если $c_{ij} = 0$, то перейти к п. 7; иначе выполнить $KS \leftarrow KS + 1$ и перейти к п. 14.

11. $D(i, j) \leftarrow D(i,k) + D(k, j)$; $D(j,i) \leftarrow D(i, j)$; $Q(i, j) \leftarrow RT$; $Q(j,i) \leftarrow RT$; $C(i, j) \leftarrow C(k, j)$; $C(j,i) \leftarrow C(k,i)$.

15. Если $KS = n - i$, то $KU \leftarrow KU + 1$.

18. Если $RMAX \le (n-1) \wedge KU < (n-1)$, то перейти к п. 2, иначе к п. 19.

19. Если $RMAX > (n-1) \wedge KU < (n-1)$, то вывести сообщение о несвязности сети.

Для алгоритма FloydS:

2. Для $\{ j \mid j = \overline{1, n-1} \}$, если $i \ne j$ выполнить пп. 3-8.

3. Для $\{ k \mid k = \overline{j+1, n} \}$, если $k \ne j \ne i$ выполнить пп. 4-7.

6. $Q(j,k) \leftarrow Q(j,i) + Q(i,k)$; $Q(k, j) \leftarrow Q(j,k)$; $D(j,k) \leftarrow D(j,i) + D(i,k)$; $D(k, j) \leftarrow D(j,k)$; $C(j,k) \leftarrow C(i,k)$; $C(k, j) \leftarrow C(i, j)$.

Для алгоритма SP2S:

4. Если $D(i, j) < \infty$, то перейти к п. 5, иначе перейти к п. 5а.

5. $T(i) \leftarrow T(i) + 1$; $A(i, T(i)) \leftarrow j$; перейти к п. 11.

5а. Если $i < j$, то перейти к п. 6, иначе к п. 11.

15. Для $\{ i \mid i = \overline{1, n-1} \}$ выполнить пп. 16-31.

25. $D(i,j) \leftarrow D(i,k) + D(k,j)$; $D(j,i) \leftarrow D(i,j)$; $Q(i,j) \leftarrow RT$; $Q(j,i) \leftarrow RT$; $C(i,j) \leftarrow C(k,j)$; $C(j,i) \leftarrow C(k,i)$.

28. $T(i) \leftarrow T(i)+1$; $T(j) \leftarrow T(j)+1$; $A(i,T(i)) \leftarrow j$; $A(j,T(j)) \leftarrow i$; $P1.F \Rightarrow P.F$; $P3 \Rightarrow P$; $P \Rightarrow P.F$; Освободить элемент $E(P3)$. Перейти к п. 30.

33. Если $RMAX \le (n-1) \wedge KU < (n-1)$, то перейти к п. 14, иначе к п. 34.

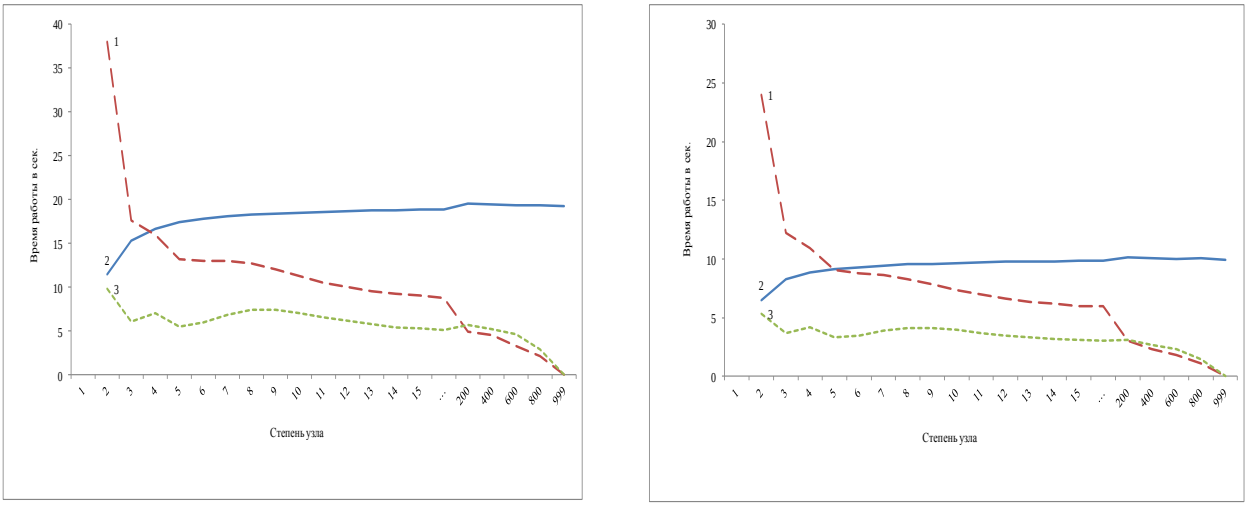
34. Если $RMAX > (n-1) \wedge KU < (n-1)$, то вывести сообщение о несвязности сети.

4.3.2. Экспериментальное исследование алгоритмов

Для проведения эксперимента была составлена тестовая программа [102] на языке Digital Visual Fortran (DVF) компании Digital Equipment Corporation в среде Microsoft (MS) Developer Visual Studio (VS) DVF 6.1. Во внешней программе в режиме диалога вводились: число узлов сети — n; число исходящих из каждого узла дуг (степень узла) — val; границы изменения значений весов дуг от $MINVAL$ до $MAXVAL$; параметр, управляющий выводом входных и выходных данных. Далее с помощью датчика псевдослучайных чисел (встроенной функции языка $RAND()$) генерировались веса дуг от $MINVAL$ до $MAXVAL$, формировались массивы R, D, Q, C и массивы A, T, SP (для алгоритмов SP2, SP2S). Вся оперативная память, необходимая для работы алгоритмов выделялась и освобождалась динамически во внешней программе. Время работы алгоритмов фиксировалось встроенной подпрограммой $cpu\,time()$ непосредственно до входа и после выхода из алгоритмов построения КП. Работа алгоритмов проверялась на ПК с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб под управлением операционной системы Windows Vista. Отдельно проводилось моделирование решения задачи на разреженных сетях с числом узлов n = 1000 при изменении значений параметра val от 2 до 999; при изменении n от 100 до 1000 при val = 5; на сильно разреженных сетях при val = 2 и изменении n от 500 до 1000. Во всех случаях принималось $MINVAL$ = 30, $MAXVAL$ = 120. Все алгоритмы были оформлены в виде внешних подпрограмм с передачей

фактических параметров.

На рис. 4.5а) и рис. 4.5б) приведены соответственно графики изменения времени работы алгоритмов SP1, Floyd, SP2 и SP1S, FloydS, SP2S в зависимости от роста степени узлов *val* от 2 до 999 для $n = 1000$.



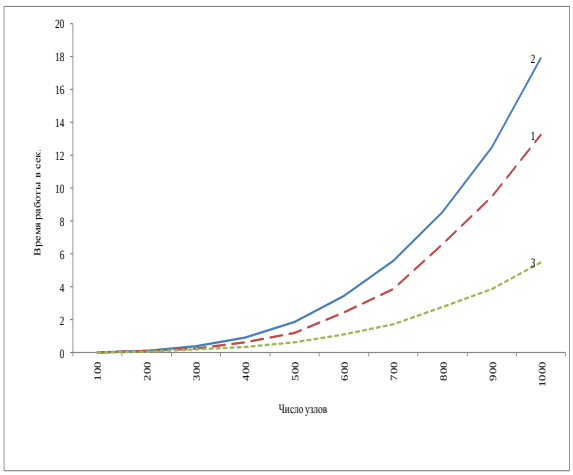
а) б)

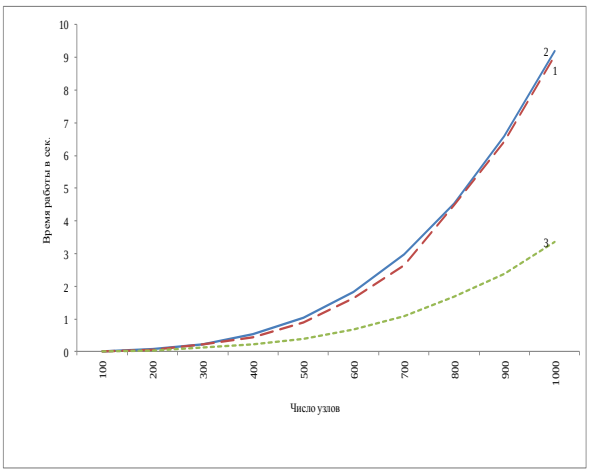
Рис. 4.5. Время работы алгоритмов SP1(1), Floyd(2), SP2(3) — а) и SP1S(1), FloydS(2), SP2S(3) — б) в зависимости от степени узлов

Из графиков видно, что алгоритмы SP1 и SP1S работают быстрее, чем алгоритмы Floyd и FloydS уже при степени узлов $val = 5$, а алгоритмы SP2 и SP2S с ростом степени узлов значительно опережают алгоритмы Floyd и FloydS. Для больших значений параметра *val* (от 200 до 999) алгоритмы SP1, SP1S оказываются лучше, чем SP2, SP2S. Алгоритмы SP2, SP2S всегда лучше алгоритмов Флойда и с ростом степени узлов быстрее последних в несколько раз.

На рис. 4.6а), 4.6б) и 4.7а), 4.7б) приведены соответственно графики изменения времени работы алгоритмов SP1, Floyd, SP2 и SP1S, FloydS, SP2S в зависимости от роста числа узлов от 100 до 1000 при $val = 5$ и при росте числа узлов от 500 до 1000 при $val = 2$.

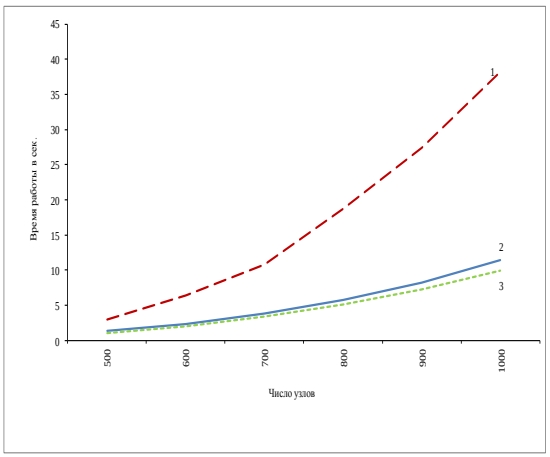
Из рисунков 4.6а) и 4.6б) видно, что для сетей с умеренной разреженностью при росте числа узлов в сети алгоритмы SP1, SP2 и SP1S, SP2S лучше, чем Floyd и FloydS соответственно. Для сильно разреженных сетей алгоритмы Floyd и FloydS с ростом числа узлов остаются быстрее алгоритмов SP1 и SP1S, но всегда хуже, чем алгоритмы SP2 и SP2S (рис. 4.7а) и рис. 4.7б)).

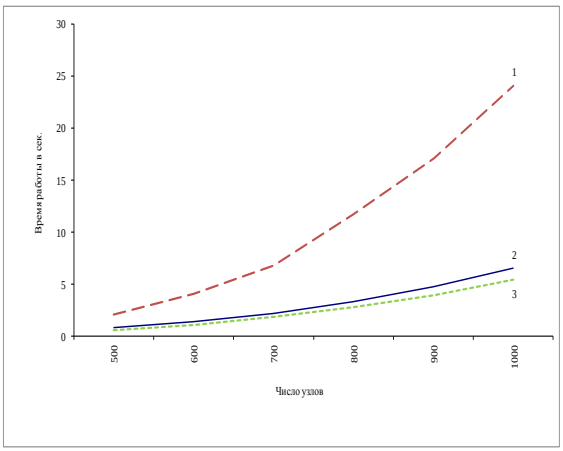




а) б)

Рис. 4.6. Время работы алгоритмов SP1(1), Floyd(2), SP2(3) — а) и SP1S(1), FloydS(2), SP2S(3) — б) в зависимости от роста числа узлов при $val = 5$





а) б)

Рис. 4.7. Время работы алгоритмов SP1(1), Floyd(2), SP2(3) — а) и SP1S(1), FloydS(2), SP2S(3) — б) в зависимости от роста числа узлов при $val = 2$

Отметим интересный факт. При проведении исследований алгоритмов, реализующие их программы оформлялись также как внутренние подпрограммы без передачи фактических параметров. Оказалось, что такие подпрограммы в среде операционной системы Windows XP работают быстрее, чем внешние, а в среде Windows Server-2003 и Vista работают чуть медленнее. Этот факт еще раз подтверждает зависимость скорости выполнения программ, скомпилированных одним и тем же транслятором, от операционной среды их выполнения.

4.4. Задачи построения остовных деревьев

В данном подразделе проведен краткий обзор и анализ методов и алгоритмов построения минимального остовного дерева на взвешенных графах. Рассматривается эффективная реализация «жадного» алгоритма Краскала для нахождения остовного дерева с минимальным или максимальным весом на неориентированном взвешенном графе. Получена практическая оценка трудоемкости алгоритма, которая для связных графов составляет $O(e)$, где e — число ребер графа. Показано, что предложенный алгоритм асимптотически лучше алгоритма Прима для графов с числом ребер не больше, чем $0{,}27v^2$, где v — число вершин графа. Путем численных экспериментов проведено сравнение двух реализаций алгоритма Краскала, основанных на списочных (предложенный алгоритм) и древовидных (алгоритм Тарьяна) структурах данных. Результаты сравнения позволяют утверждать, что для решения практических задач нахождение минимального или максимального остовного дерева (леса) алгоритмы со списочной структурой данных работают не хуже, а в большинстве случаев быстрее, чем алгоритмы с древовидной структурой. Экспериментальное исследование предложенного алгоритма на графах, содержащих от 499500 до 71994000 ребер, показало его высокую вычислительную эффективность, и он может быть рекомендован для решения практических задач на разреженных графах или сетях большой размерности.

4.4.1. Постановка задачи, обзор методов и алгоритмов ее решения

Пусть задан простой (не содержащий петель и параллельных ребер) неориентированный граф $G(V, E)$ с множеством вершин V, $v = \left| V \right|$ и множеством ребер E, $e = \left| E \right|$, где v и e соответственно число вершин и ребер графа, а $\left| \cdot \right|$ — знак мощности множества. Граф может быть неполным и несвязным. Для графа задан вектор весов ребер $C = \left\| c_i \right\|$, $i = \overline{1, e}$, где $c_i \in Z^+$, Z^+ — множество неотрицательных целых чисел.

Требуется найти ациклический подграф (лес) графа G с минимальным или

максимальным общим весом. В литературе принято называть такой ациклический подграф минимальным (максимальным) остовным деревом — minimum (maximum) spanning tree (MST) в случае связного графа и минимальным (максимальным) остовным лесом — minimum (maximum) spanning forest (MSF) в случае несвязного графа. В дальнейшем для упрощения изложения будем использовать для связных и несвязных графов общий термин — минимум (максимум) остовных деревьев — minimum (maximum) spanning trees (MST).

В настоящее время известно достаточно много алгоритмов определения MST- деревьев на взвешенном графе. Наиболее известными из них являются алгоритмы Борувки (1926 г.) [189, 402], Ярника-Прима-Дейкстры (1930, 1956, 1958 гг.) [325, 421, 240] и Краскала-Лобермана-Вайнберга (1956, 1957) [352, 373]. Алгоритм Борувки был много раз переоткрыт другими авторами (Шоке (1938), Флорек, Лукасевич, Перкал, Штейнгауз, Зубжицкий (1951)) и известен также как алгоритм Соллина (1965) [434]. Обширный библиографический обзор по истории развития MST-алгоритмов можно найти в работах Тарьяна [451], Грэхэма и Хелла [295].

Для построения современных эффективных алгоритмов используются абстрактные типы данных (abstract data type) — АТД, которые включают различные комбинации традиционных структур данных, таких как простые переменные и массивы и специальных — указателей, списков, стеков, очередей, деревьев и пр. Для заданных классов АТД определяются функциональные операторы (процедуры) обработки внутренних структур данных, которые реализуются средствами используемого языка программирования. В книгах [2, 4, 5, 36, 69, 72, 78, 107, 144, 381] можно найти подробнейшее описание АТД, применяемых в алгоритмах построения остовных деревьев, оценки трудоемкости алгоритмов для их различных реализаций, а также исчерпывающий обзор и исторические справки по алгоритмам нахождения остовных деревьев. Следует также отметить, что в современных объектно-ориентированных языках программирования, таких как C++ , Java и др. имеются библиотеки со стандартными АТД и программами для решения задач нахождения оптимальных остовных деревьев,

смотри например [188, 292, 362].

Все три основные группы алгоритмов относятся к так называемым «жадным» алгоритмам, так как на каждом шаге наращивания MST-дерева включают в него наиболее подходящие ребра графа (например, ребра с минимальным весом).

Рассмотрим сначала алгоритмы группы Краскала, поскольку для них раньше, чем к другим группам были применены АТД. Алгоритмы Краскала основаны на предварительной сортировке ребер графа по весу и последующем последовательном выборе ребер минимального или максимального веса для наращивания MST-деревьев. Алгоритмы этой группы имеют асимптотическую трудоемкость порядка $O(e\log e)$. Трудоемкость этих алгоритмов зависит от выбора метода сортировки ребер и методов поиска и объединения (слияния) независимых компонент графа (поддеревьев) на каждой итерации алгоритма при добавлении нового ребра. Известны два наиболее эффективных АТД — основанных на списочных структурах, реализованных с помощью указателей и массивов, и древовидных структурах. Для этих АТД определены процедуры начальной установки указателей вершин графа на принадлежность к независимым компонентам — make set, поиска — find set и слияния — merge/union set непересекающихся множеств, которые будем называть FMS-процедурами. В случае использования списочных структур процедуры make set и find set имеют трудоемкость $O(1)$, а процедура merge/union $O(v\log v)$ при условии, что меньшее множество сливается с большим. Общая трудоемкость FMS-процедур для списочных АТД составит $O(v+e'+v\log v)$ для связных графов, где e' — число просмотренных ребер графа для полного построения MST-дерева и $O(v+e+v\log v)$ для несвязных графов.

Интересна история уточнения оценок трудоемкости FMS-процедур для древовидных АТД [101]. В 1964 г. в работе Геллера и Фишера [265] для представления непересекающихся множеств впервые была использована древовидная структура и предложены процедуры merge/union set без балансировки дере-

ва (слияния/объединения по рангу — merge/union by rank) и find set без сжатия пути (find by compression path). В 1972 г. Фишер [251] показал, что для n операций слияния/объединения и n операций поиска непересекающихся множеств с объединением по рангу и сжатием путей требуется порядка $Cn\log\log n$ времени, где C — некоторая константа. В 1973 г. эта оценка была понижена до $Cn\log^* n$ в работе Хопкрофта и Ульмана [309], где $\log^* n$ равна числу итераций двоичной логарифмической функции от n, при котором она станет меньше единицы (ее значение меньше 5, если n меньше 2^{65536}). В 1975 г. Тарьян [450, см. также 452, 453], используя функцию инверсную функции Аккермана, доказал, что время, необходимое для выполнения $n-1$ операций merge/union и $f \geq n$ операций find set с объединением по рангу и сжатием путей не превосходит $Cf\alpha(f,n)$. Где $\alpha(f,n)$ очень медленно растущая функция (для $n \leq 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\}65537\text{двоек}}$, $\alpha(f,n) \leq 3$). Кроме того Тарьян показал, что существует последовательность операций merge/union и find set, для которой требуемое время пропорционально $f\alpha(f,n)$.

Для наших обозначений общая трудоемкость FMS-процедур для древовидных АТД составит $O(v+e'+e'\alpha(e',v))$ для связных графов, где e' — число просмотренных ребер графа для полного построения MST-дерева и $O(v+e+e\alpha(e,v))$ для несвязных графов. Причем процедуры make set и merge/union имеют трудоемкость $O(1)$, а процедура find set $O(e'\alpha(e',v))$ или $O(e\alpha(e,v))$.

В [101] отмечается недостаточность исследований и анализа средней трудоемкости FMS-процедур. Отдельные результаты имеются в работе Яо [482]. В работах Чазелла [202], Петти и Рамахандрана [417] отмечается, что алгоритм Тарьяна с техникой find и merge/union со сжатием путей и объединением по рангу имеет наименьшую асимптотическую сложность для самого худшего случая, и теоретически не улучшаем, однако невозможно дать точную оценку этой сложности через параметры графа.

Ряд других важных результатов о структурах данных для непересекающихся множеств можно найти в работах, приведенных в заключительных замечаниях на стр. 605 в книге Т. Кормена, Ч. Лейзерсона, Р. Ривеста и К. Штайна [72].

Перейдем к рассмотрению группы алгоритмов Прима, которые основаны на методе «ближайшего соседа». Эти алгоритмы не требует предварительной сортировки ребер и проверки на цикличность при наращивании дерева на каждой итерации. Алгоритм начинает работать от любой вершины графа и наращивает только одно дерево, добавляя на каждом шаге ребро с минимальным весом. Для классической матричной реализации алгоритма его трудоемкость составляет $O(v^2)$ операций и он асимптотически оптимален, т.е. для полных графов асимптотически не улучшаем. Для реализации алгоритма Прима для разреженных графов чаще всего используются различные модификации АТД, известные как очередь с приоритетами. Если граф представлен связанными списками смежности, а очередь с приоритетами — полным бинарным деревом (бинарной пирамидой, binary heap), то трудоемкость алгоритма Прима составляет $O(e \log v)$. При использовании фибоначчиевой пирамиды (fibonacci heap) сложность алгоритма можно уменьшить до $O(e + v \log v)$, и в лучших реализациях до $O(e \log^* v)$ [256] (Фридман, Тарьян) и $O(e \log \log^* v)$ [260] (Габов, Галил, Спенсер, Тарьян). Известна также быстрая реализация алгоритма Прима Джонсоном [327, 329], основанная на частично упорядоченных полных d - деревьях. Сложность его алгоритма определяется как $O(e \log_d v)$, где $d = \max\{2, e/v\}$ и он наиболее эффективен для графов со средней плотностью.

В отличие от алгоритмов Краскала и Прима в алгоритме Борувки-Соллина вначале от каждой вершины графа находится наименьшее ребро, и формируются независимые компоненты. Далее полученные компоненты «одновременно» наращиваются путем добавления к ним новых примыкающих минимальных ребер, не образующих циклы, до тех пор, пока не будет построено MST-дерево. Таким образом, этот алгоритм можно рассматривать как симбиоз алгоритмов

Краскала и Прима. Для реализации алгоритма используются АТД, основанные на представлении непересекающихся множеств с операциями find и merge/union и очереди с приоритетами. Простейшие реализации алгоритма имеют трудоемкость $O(e\log v)$. Для разреженных графов независимо Яо [481] и Черитон и Тарьян [207] улучшили эту оценку до $O(e\log\log v)$. Чазелл [202] с использованием идей Борувки-Соллина и новым искусным представлением очереди с приоритетами (soft heap) [203] разработал оригинальный MST-алгоритм со сложностью $O(e\alpha(e,v))$, который не использует «жадную» стратегию. Ключевая идея алгоритма заключается в последовательном образовании, объединении и улучшении кластеризованных субоптимальных (не обязательно с минимальным весом) независимых компонент графа до тех пор, пока не будет получено оптимальное MST-дерево.

Интересны также работы Фридмана, Вилларда [257], Каргера, Клейна, Тарьяна [333] и Кинга [342], где приводятся алгоритмы со сложностью $O(e+v)$. Первый использует для представления входных данных b- битовые целые числа и оперирует с b- битовыми словами ПЭВМ, а второй и третий являются рандомизированными алгоритмами (дают математическое ожидание времени работы) на основе идей Борувки, Комлоса [349] и Диксона, Рауха, Тарьяна [242]. В работе [335] показано, что два последних алгоритма наиболее эффективны для плотных графов.

Перспективной технологией для уменьшения времени работы алгоритма построения минимального остовного дерева является параллельная имплементация алгоритма Борувки на нескольких скоростных процесорах (см. например работу [166]).

Подводя итоги краткому обзору MST-алгоритмов можно сделать следующие выводы: дальнейшее развитие алгоритмов возможно за счет усложнения АТД [203] и уменьшения трудоемкости отдельных операций, используемых при работе с их внутренними структурами данных; увеличение быстродействия алгоритмов может быть достигнуто за счет распараллеливания процесса построения MST, особенно в алгоритмах Борувки-Соллина; в настоящее время

наилучшими по быстродействию для разреженных графов являются алгоритмы Тарьяна [450] (без учета сортировки ребер), Габова, Галила, Спенсера и Тарьяна [260] и Чазелла [202].

В табл. 4.1 приведены основные алгоритмы, использующиеся в настоящее время для построения MST-деревьев.

Таблица 4.1

Алгоритмы построения MST-деревьев и их асимптотическая трудоемкость

Полные графы (верхние оценки)		
$O(e\log v)$ – Boruvka(1926), Sollin(1961) $O(e\log e)$ – Kruskal(1956) $O(v^2)$ – Jarnik(1930), Prim(1961), Dijkstra(1962)		
Разреженные связные графы (лучшие реализации с АТД)		
Kruskal	Prim	Boruvka
$O(e\log e)$ - для всех методов с сортировкой сравнением $O(e'\alpha(e',v))$, без учета сортировки ребер, Tarjan (1984), до сих пор считается самым быстрым	$O(e\log v)$, Binary heap $O(e\log_d v)$, d - heap, $d=e/v$, Johnson (1977) $O(e+v\log v)$, Fibonacci heap $O(e\beta(e,v))=O(e\log^* v)$, Freedman, Tarjan $O(e\log\beta(e,v))=O(e\log\log^* v))$, Gabow, Galil, Spencer and Tarjan (1986) Где $\beta(e,v)=\min\{i:\log^i(e/v)\le 1\}$, $\log^i(e/v)=\underbrace{\log\log\ldots\log(e/v)}_{i}=\log^*(e/v)$, как правило $\beta(e,v)\le 6$	$O(e\log v)$, Sollin (1961) $O(e\log\log v)$, Yao (1975) $O(e\log\log v)$, Cheriton, Tarjan (1976) $O(e\alpha(e,v))$, Chazelle (2000)

Несмотря на существование достаточно эффективных MST- алгоритмов, интерес к разработке еще более быстрых алгоритмов не ослабевает (см., например, работы [202, 419], в которых рассматривается современное положение дел в поиске линейного алгоритма построения MST). Это объясняется тем, что одновременно с постоянным ростом быстродействия и оперативной памяти современных ПЭВМ, появляется возможность решать более сложные оптимизационные задачи для графов или сетей большой размерности (десятки тысяч вершин, сотни тысяч и миллионы ребер), в которых в качестве подзадач нужно

многократно находить MST-деревья. Тогда алгоритм определения MST-деревьев с минимальной трудоемкостью может значительно сократить время решения общей задачи.

4.4.2. Алгоритмы построения MST-деревьев

Рассмотрим возможную реализацию «жадного» алгоритма Краскала, который находит остовное дерево с минимальным или максимальным весом за время $O(e)$. Введем обозначения: $\forall$, $\exists$ — кванторы всеобщности и существования; $\cup$, $\cap$, $\setminus$, $\supseteq$ — знаки объединения, пересечения, вычитания и включения множеств; $\in$, $\notin$ — знаки принадлежности и не принадлежности к множеству; $\varnothing$ — пустое множество; $\{\cdot\}$ — множество подмножеств или элементов множества; $\wedge$, $\vee$ — знаки логического «и», логического «или»; $i=\overline{1,k}$ — означает «для всех i от единицы до k. Для построения алгоритма докажем следующую теорему.

Теорема 4.4. Пусть $\{(V_0,E_0),(V_1,E_1),(V_2,E_2),\ldots,(V_k,E_k)\}$ — частично построенный остовный лес графа $G=(V,E)$, где $(V_0,E_0)=(\varnothing,\varnothing)$, (V_i,E_i) — связные независимые компоненты графа (поддеревья), $\bigcap\limits_{i=0}^{k}\{(V_i,E_i)\}=\varnothing$ и выполняется условие

$$\left|\bigcup_{i=0}^{k}V_i\right|<v\wedge E\setminus\bigcup_{i=0}^{k}E_i\neq\varnothing. \qquad (4.31)$$

Тогда для $\forall e_{lr}\in E\setminus\bigcup\limits_{i=0}^{k}E_i$, включаемого в остовный лес, пока выполняется условие (4.31), справедливо:

1) если $l\notin V_i\wedge r\notin V_i$, $i=\overline{1,k}$, то ребро e_{lr} образует новую $k+1$ компоненту $(\{l,r\},\{e_{lr}\})$, а $\{(V_i,E_i),(\{l,r\},\{e_{lr}\})\}$, $i=\overline{0,k}$ — частичный остовный лес графа G;

2) если

$(l\in V_j,\ j\in\{1,2,\ldots,k\})\wedge(r\notin V_i,i=\overline{1,k})\vee(l\notin V_i,i=\overline{1,k})\wedge(r\in V_j,\ j\in\{1,2,\ldots,k\})$, то

ребро e_{lr} соответственно добавляется к компонентам $(V_j \cup r, E_j \cup e_{lr})$ или $(V_j \cup l, E_j \cup e_{lr})$, а $\{(V_i, E_i), (V_j, E_j)\}$, $i = \overline{0,k}$, $i \neq j$ — частичный остовный лес графа G;

3) если $(l \in V_j, j \in \{1,2,...,k\}) \wedge (r \in V_m, m \in \{1,2,...,k\}) \wedge j \neq m$, то ребро e_{lr} объединяет (сливает) компоненты (V_j, E_j) и (V_m, E_m), а $\{(V_i, E_i), (V_j \cup V_m, E_j \cup E_m \cup e_{lr})\}$, $i = \overline{0,k}$, $i \neq j$, $i \neq m$ — частичный остовный лес графа G;

4) если $l \in V_j \wedge r \in V_j$, $j \in \{1,2,...,k\}$, то ребро e_{lr} образует цикл в компоненте (V_j, E_j).

Если условие (4.31) не выполняется, то при $\left| \bigcup_{i=0}^{k} V_i \right| = v$ остовный лес построен и k — число его компонент. Если $\left| \bigcup_{i=0}^{k} V_i \right| < v$, а $E \setminus \bigcup_{i=0}^{k} E_i = \varnothing$, то остовный лес построен, k — число его компонент и в графе $G = (V, E)$ имеется $v - \left| \bigcup_{i=0}^{k} V_i \right|$ изолированных вершин.

Доказательство. Покажем по индукции, что добавление к остовному лесу любых ребер из $E \setminus \bigcup_{i=0}^{k} E_i$ при выполнении условия (4.31) и пунктов 1), 2), 3) приводит к образованию нового леса на множестве выбранных вершин. Очевидно, что для $\{(V_0, E_0)\}$ добавление любого ребра приводит к образованию первой компоненты с двумя вершинами и одним ребром между ними. Предположим далее, что после выбора j ребер получен $\{(V_0, E_0), (V_1, E_1), (V_2, E_2), ..., (V_k, E_k)\}$ — частично построенный k- компонентный остовный лес графа G на множестве выбранных к данному моменту вершин (нулевая компонента пустая и введена для исключения вырожденного случая, когда $\{(V_i, E_i)\} = \varnothing$). Рассмотрим выполнение $j+1$ шага, т.е. выбор $j+1$ ребра. Пусть (4.31) выполняется, тогда при выборе $j+1$ ребра и выполнении

условий 1), 2), 3) будет соответственно образована новая компонента (V_{k+1}, E_{k+1}), добавлена одна новая вершина и одно новое ребро к существующему лесу, или две независимые компоненты леса сольются вместе с добавлением одного нового ребра. Таким образом, обновленный лес не может содержать цикла ни в одной из своих компонент и остается остовным. Если при выборе $j+1$ ребра не будет выполняться условие (4.31), то очевидно, что на этом шаге при $\left|\bigcup_{i=0}^{k} V_i\right| = v$ будет окончательно построен k- компонентный остовный лес (остовное дерево при $k = 1$), содержащий $v-k$ ребер, а при $\left|\bigcup_{i=0}^{k} V_i\right| < v$ и $E \setminus \bigcup_{i=0}^{k} E_i = \varnothing$ остовный лес будет содержать $v - \left|\bigcup_{i=0}^{k} V_i\right|$ изолированных вершин. Доказательство 4) немедленно следует из того, что каждая компонента (V_j, E_j) является поддеревом и две любые вершины этого поддерева связаны единственным путем. Поэтому добавление нового ребра, соединяющего две любые вершины из (V_j, E_j) приводит к образованию цикла в (V_j, E_j). ■

Ясно, что для любого графа с использованием теоремы 4.4 можно найти множество различных остовных лесов, если определенным образом задавать выбор ребер для построения остовного леса. Так, например, для полного графа с $v = |V|$ вершинами существует v^{v-2} различных остовных деревьев.

Пусть множество ребер $E = \{e_i\}$, $i = \overline{1,e}$ графа $G = (V, E)$ упорядочено по неубыванию или по невозрастанию их весов $\{c_i\}$, $i = \overline{1,e}$, $c_i \le c_{i+1}$ или $c_i \ge c_{i+1}$, $i = \overline{1,e-1}$.

Теорема 4.5. Для нахождения остовного леса графа $G = (V, E)$ с минимальным или максимальным весом необходимо и достаточно при построении остовного леса с использованием условий теоремы 4.4, последовательно выбирать ребра в порядке неубывания или невозрастания их весов.

Доказательство. Пусть $\{(V_j, E_j)\}$, $j = \overline{0,k}$ — частично построенный ос-

товный лес графа $G=(V,E)$, $T=\bigcup_{j=0}^{k}E_j$ и e_i — ребро с наименьшим весом, соединяющее вершину из $\{V_j\}$, $j=\overline{1,k}$ с вершиной из $V\setminus\bigcup_{j=0}^{k}V_j$. Покажем, что существует остовный лес $T'=T\cup\{e_i\}$, такой, что для любого остовного леса $T''\supseteq T$, справедливо $C(T')\le C(T'')$, где: $C(T')=\sum_{e_j\in T'}c(e_j)$; $C(T'')=\sum_{e_j\in T''}c(e_j)$; $c(e_j)$ — вес ребра e_j.

Предположим, что $T''\supseteq T$, не включает e_i и имеет наименьший вес среди всех остовных лесов, содержащих T. Добавим ребро e_i к T'' и получим единственный цикл, содержащий ребро e_i и некоторое другое ребро x, соединяющее вершину из $\{V_j\}$, $j=\overline{1,k}$ с вершиной из $V\setminus\bigcup_{j=0}^{k}V_j$. Согласно предположению $c(e_i)\le c(x)$ и $x\notin T$. Удалим ребро x из T'' и получим новый остовный лес $T'=T''\setminus\{x\}\cup\{e_i\}$, содержащий T, ребро e_i и имеющий вес не больше, чем вес T'', т.е. $C(T')\le C(T'')$.

Очевидно, что все сказанное справедливо для каждого ребра, выбираемого из $\{e_i\}$, $i=\overline{1,e}$ при построении остовного леса с использованием условий теоремы 4.4. Это доказывает необходимость условия теоремы 4.5. Достаточность условия следует из того, что оснований для выбора нового ребра на каждом шаге построения остовного леса достаточно, чтобы гарантировать получение остовного леса с минимальным весом, содержащего выбранное ребро.

Аналогичные рассуждения можно провести для доказательства теоремы для случая построения остовного леса с максимальным весом. ■

Другое доказательство теоремы 4.5 непосредственно вытекает из теоремы Радо-Эдмондса [424, 246], суть которой заключается в следующем. Рассмотрим систему подмножеств $M_G=(E,F)$, где F — множество различных остовных лесов, являющихся независимыми подмножествами E. Система M_G образует графический матроид и задачу определения минимального или максимального

остовного леса на взвешенном графе $G = (V, E)$ можно рассматривать как комбинаторную задачу оптимизации для M_G с использованием «жадного» алгоритма.

Пусть найден остовный лес графа с максимальным весом и множество ребер остовного леса $S = \{e_i\}$, $i = \overline{1, v-k}$ упорядочено в лексикографическом порядке по невозрастанию весов, т.е. $c(e_1) \geq c(e_2) \geq \ldots \geq c(e_{v-k})$. Рассмотрим произвольное независимое множество ребер $T = \{e'_i\}$, $i = \overline{1, m}$, $m \leq v-k$ и $c(e'_1) \geq c(e'_2) \geq \ldots \geq c(e'_m)$. Докажем, что S лексикографически максимальная база матроида M_G, т.е. $C(S) \geq C(T)$ или, что для произвольного $i \leq m$ справедливо $c(e_i) \geq c(e'_i)$. Предположим, что $c(e'_i) > c(e_i)$ и рассмотрим независимые множества $S_{i-1} = \{e_1, \ldots, e_{i-1}\}$, $T_i = \{e'_1, \ldots, e'_{i-1}, e'_i\}$. Согласно аксиомам независимости матроидов существует элемент e'_j, $j \leq i$, такой, что множество $S' = \{e_1, \ldots, e_{i-1}, e'_j\}$ — независимое. Но S' лексикографически больше S, поскольку $c(e'_j) \geq c(e'_i) \geq c(e_i)$, что противоречит лексикографической максимальности S и доказывает, что $c(e_i) \geq c(e'_i)$, $i = \overline{1, m}$.

Доказательство теоремы 4.5 для построения остовного леса с минимальным весом для матроида M_G проводится аналогично. ■

Рассмотрим алгоритм нахождения остовного леса графа $G = (V, E)$ с минимальным или максимальным весом [30, 31]. Прежде всего, необходимо выбрать наиболее быстрый алгоритм сортировки ребер графа, поскольку время сортировки будет существенно влиять на вычислительную эффективность алгоритма в целом. Обширный библиографический обзор по методам и алгоритмам сортировки можно найти в широко известной книге Кнута [69]. В книге проведено тщательное сравнение различных алгоритмов сортировки и изложена история развития методов сортировки. Сразу отметим, что любой алгоритм, основанный на сравнении пар элементов, имеет среднюю трудоемкость и трудоемкость в худшем случае порядка $O(e \log e)$. Поэтому совершенно очевидно,

что использовать такой алгоритм сортировки для решения задачи нахождения минимального или максимального остовного дерева с трудоемкостью ниже, чем $O(e\log e)$ бессмысленно. В связи с этим, для сортировки ребер будем применять алгоритм со сложностью порядка $O(e)$, основанный на методе преобразования ключа (веса ребра) в адрес оперативной памяти компьютера. Такой тривиальный подход к сортировке ребер оправдан низкой трудоемкостью и тем, что для большинства графов (сетей), описывающих реальные объекты, разброс значений весов ребер, как правило, незначителен.

Для описания алгоритма сортировки определим следующие переменные и структуры данных. Обозначим: v, e — число вершин и неориентированных ребер в графе; $MINVAL$, $MAXVAL$ — минимальный и максимальный вес ребра; $IS(e)$, $ST(e)$, $C(e)$ — массивы, содержащие соответственно номера левых и правых вершин ребер и веса ребер. Для удобства эти массивы могут быть упорядочены так, чтобы номер левой вершины был меньше номера правой вершины, а номера левых и правых вершин увеличивались от начала к концу массивов; $U(MMAX,2)$ — массив, содержащий ссылки на списки номеров ребер, упорядоченных по неубыванию весов в массиве $E(e)$; $E(e)$ — массив, содержащий списки номеров ребер, упорядоченных по неубыванию весов.

Алгоритм сортировки не перемещает исходные значения в массивах IS, ST, C. Сортируются только ссылки на номера ребер в исходных массивах в порядке неубывания весов ребер. Для этого строится массив $E(e)$ списков номеров ребер. Поскольку может быть несколько ребер с одинаковым весом, массив содержит списки для ребер с одинаковым весом. В массиве $U(MMAX,2)$ элемент $U(i,1)$ содержит голову списка — номер первого ребра k в массиве $E(e)$ с весом i, а элемент $U(i,2)$ указывает на номер последнего ребра j в массиве $E(e)$ с весом i. При этом элемент $E(k)$ указывает на следующее ребро в $E(e)$ с таким же весом и т.д. Для последнего ребра j значение $E(j)=0$ и означает конец списка ребер с одинаковым весом. Для того чтобы в начале массива U не было пустых элементов, выполняется масштабирование весов ребер. По-

этому значение $MMAX = MAXVAL - MINVAL + 1$ равно максимальному весу ребра после масштабирования плюс 1, а минимальный вес ребра будет всегда равен 1. Если веса ребер являются дробными неотрицательными числами, то с учетом необходимой точности вычислений, можно ввести соответствующие масштабные коэффициенты (например, умножить веса всех ребер на 1000, 10000 и т.д.) для перевода весов ребер в целые числа, а затем определить значение $MMAX$.

Алгоритм 4.6. *SORT*. Сортировка ребер по неубыванию весов с трудоемкостью $O(e)$.

1. $U \leftarrow 0;\ E \leftarrow 0$.

2. Для $\{ i \mid i = \overline{1,e} \}$ выполнить пп. 3-6.

3. $l \leftarrow C(i) - MINVAL + 1$.

4. Если $U(l,1) = 0$, то $U(l,1) \leftarrow i$; иначе $E(U(l,2)) \leftarrow i$.

5. $U(l,2) \leftarrow i$.

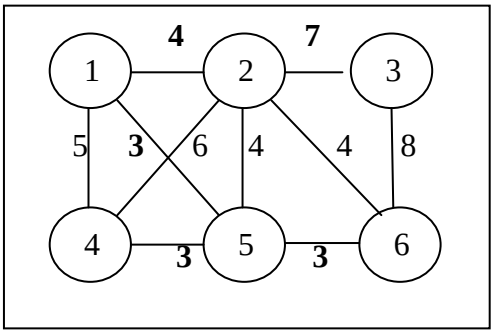
6. Перейти к п. 2. *** Конец цикла по i

7. Стоп.

На рис. 4.8 приведен пример структур $IS(e)$, $ST(e)$, $C(e)$, $U(MMAX,2)$, $E(e)$, полученных после сортировки ребер на графе, содержащем $v = 6$ вершин и $e = 10$ ребер. Значения $MINVAL = 3$, $MAXVAL = 8$, $MMAX = 6$.

Оценим трудоемкость алгоритма. Пусть a, b, c — время на операции присваивания, сравнения и перехода соответственно. Для определения приблизительной оценки, не будем отдельно различать время выполнения простой операции присваивания от операций сложения и операций присваивания со сложным индексом. Тогда трудоемкость шага 1 составит — $a \cdot (MMAX \cdot 2) + a \cdot e$; шага 2 — $a \cdot e + b \cdot e + c \cdot e$; шага 3 — $a \cdot e$; шага 4 — $b \cdot e + a \cdot e/2 + a \cdot e/2$; шага 5 — $a \cdot e$, где «·» — знак умножения. Общее время сортировки $T_{сорт} = e \cdot (5a + 2b + c) + MMAX \cdot 2a$. Ясно, что реальная вычислительная трудоемкость сортировки будет зависеть от конкретного языка кодиро-

вания алгоритма, компилятора и операционной системы, а скорость выполнения конкретной программы сортировки от мощности компьютера. Так как для реальных задач величина $MMAX$ нуль - сравнима по отношению к e, то асимптотическая оценка трудоемкости алгоритма 1 всегда составляет $O(e)$ операций.



№ ребра	*IS*	*ST*	*C*		*U*(*i*,1)	*U*(*i*,2)		*E*(*e*)
1	**1**	**2**	**4**		3	10		6
2	1	4	5		1	7		-
3	**1**	**5**	**3**		2	2		9
4	**2**	**3**	**7**		5	5		-
5	2	4	6		4	4		-
6	2	5	4		8	8		7
7	2	6	4		-	-		0
8	3	6	8		-	-		-
9	**4**	**5**	**3**		-	-		10
10	**5**	**6**	**3**		-	-		0

Рис. 4.8. Структуры данных после сортировки ребер. Жирным шрифтом отмечены ребра, вошедшие в MST

Пусть MK — максимальное число независимых компонент графа, $MK = v/2$; NK — номер очередной порожденной компоненты; KK — количество связных компонент; NY — счетчик количества вершин в строимом минимальном остовном лесе; SV — признак наличия изолированных вершин в графе; $W(MK)$ — массив весов порожденных компонент. Определим следующие массивы: $A(v)$ — массив, содержащий номера связных компонент, в которые входят вершины i, $i = \overline{1,v}$; $H(MK,3)$ — массив, содержащий ссылки на списки номеров вершин компоненты MK в массиве $G(v)$; $G(v)$ — массив, содержащий списки номеров вершин связных компонент графа; $EK(MK,2)$ — массив, содержащий ссылки на списки номеров ребер компоненты MK в массиве $B(e)$; $B(e)$ — массив, содержащий списки номеров ребер связных компонент графа. Массивы $H(MK,3)$, $G(v)$ и $EK(MK,2)$, $B(e)$ устроены аналогично массивам

$U(MMAX,2)$, $E(e)$. Элемент $H(i,3)$ содержит число вершин в i - ой компоненте графа (i - ом поддереве). Введенные структуры данных позволяют за две операции сравнения определять, к какой компоненте принадлежат вершины ребра, включаемого в остовный лес на очередном шаге алгоритма, а также за одну операцию присваивания сливать две независимые компоненты в одну. При этом всегда компонента с меньшим числом вершин или ребер сливается в большую компоненту.

Алгоритм 4.7. *LINK.* Построение остовного леса (дерева) с минимальным весом с использованием списочных структур данных.

1. $A \leftarrow 0$; $NK \leftarrow 1$; $KK \leftarrow 0$; $NY \leftarrow 0$; $G \leftarrow 0$; $B \leftarrow 0$; $W \leftarrow 0$; $SV \leftarrow 0$.

2. $KNOV \leftarrow 0$; $KPRI \leftarrow 0$; $KSLI \leftarrow 0$; $KOST \leftarrow 0$; $KPUS \leftarrow 0$; $KSUZ \leftarrow 0$.

3. Для $\{ i \mid i = \overline{1, MMAX} \}$ выполнить пп. 4-31.

4. $ND \leftarrow U(i,1)$. *** Выбрать номер первого ребра из списка ребер с одинаковым весом

5. Пока $ND \neq 0$ выполнить пп. 6-30.

6. $l \leftarrow IS(ND)$; $k \leftarrow ST(ND)$; $l1 \leftarrow A(l)$; $k1 \leftarrow A(k)$. *** Выбрать левую и правую вершины ребра и номера компонент, в которые эти вершины входят

7. Если $l1 \neq 0 \wedge k1 \neq 0 \wedge l1 = k1$, то $KPUS \leftarrow KPUS + 1$; перейти к п. 28. *** Обнаружено циклическое ребро, перейти на выбор нового ребра

8. Если $l1 = 0 \wedge k1 \neq 0$, то перейти к п. 15. ***! Присоединить к существующей компоненте новую вершину и новое ребро, перейти к присоединению к компоненте

9. Если $l1 \neq 0 \wedge k1 = 0$, то $ll \leftarrow l$; $l11 \leftarrow l1$; $l \leftarrow k$; $l1 \leftarrow k1$; $k \leftarrow ll$; $k1 \leftarrow l11$; перейти к п. 15. *** Присоединить к существующей компоненте новую вершину и новое ребро, перейти к присоединению к компоненте

10. Если $l1 = 0 \wedge k1 = 0$, то перейти к п. 12. *** Образовать новую компоненту, добавить две новые вершины и одно новое ребро, перейти на образование компоненты

11. Перейти к п. 18. *** Слить две независимые компоненты и добавить новое ребро, перейти на слияние компонент

12. $A(l) \leftarrow NK$; $A(k) \leftarrow NK$; $H(NK,1) \leftarrow l$; $G(l) \leftarrow k$; $H(NK,2) \leftarrow k$. *** Образование новой компоненты

13. $H(NK,3) \leftarrow 2$; $EK(NK,1) \leftarrow ND$; $EK(NK,2) \leftarrow ND$; $W(NK) \leftarrow C(ND)$.

14. $KK \leftarrow KK+1$; $NK \leftarrow NK+1$; $NY \leftarrow NY+2$; $KNOV \leftarrow KNOV+1$; $KOST \leftarrow KOST+1$; перейти к п. 28.

15. $A(l) \leftarrow k1$; $G(H(k1,2)) \leftarrow l$; $H(k1,2) \leftarrow l$; $H(k1,3) \leftarrow H(k1,3)+1$. *** Присоединение к компоненте

16. $B(EK(k1,2)) \leftarrow ND$; $EK(k1,2) \leftarrow ND$; $W(k1) \leftarrow W(k1)+C(ND)$.

17. $NY \leftarrow NY+1$; $KPRI \leftarrow KPRI+1$; $KOST \leftarrow KOST+1$; перейти к п. 28.

18. Если $H(l1,3) > H(k1,3)$, то $lp \leftarrow l1$; $l1 \leftarrow k1$; $k1 \leftarrow lp$. *** Слияние компонент

19. $W(k1) \leftarrow W(k1)+W(l1)$; $W(l1) \leftarrow 0$.

20. $KSLI \leftarrow KSLI+1$; $KOST \leftarrow KOST+1$; $KSUZ \leftarrow KSUZ+H(l1,3)$.

21. $G(H(k1,2)) \leftarrow H(l1,1)$; $H(k1,2) \leftarrow H(l1,2)$; $H(k1,3) \leftarrow H(k1,3)+H(l1,3)$.

22. $B(EK(k1,2)) \leftarrow EK(l1,1)$; $EK(k1,2) \leftarrow EK(l1,2)$; $B(EK(k1,2)) \leftarrow ND$.

23. $EK(k1,2) \leftarrow ND$; $W(k1) \leftarrow W(k1)+C(ND)$.

24. $j \leftarrow H(l1,1)$.

25. Пока $j \neq 0$ выполнить п. 26. *** Обновить массив вхождения вершин в компоненты

26. $A(j) \leftarrow k1$; $j \leftarrow G(j)$. *** После слияния двух компонент

27. $KK \leftarrow KK-1$. *** уменьшить число компонент на 1

28. Если $KK=1 \wedge NY=v$, то перейти к п. 33. *** Граф связный и все узлы включены в остовный лес

29. $ND \leftarrow E(ND)$. *** Выбрать номер очередного ребра

30. Перейти к п. 5. *** Конец цикла по ND

31. Перейти к п. 3. *** Конец цикла по i

32. Если $NY < v$, то $SV \leftarrow SV + 1$.

33. Стоп.

Очевидно, что алгоритм может быть использован для построения остовного леса (дерева) с максимальным весом, если в цикле по i ребра просматривать в обратном порядке, т.е. в строке 3 записать: Для $\{i \mid i = \overline{MMAX,1,-1}\}$ выполнить пп. 4-31.

В записи алгоритма 4.7 в строке 2 введены переменные $KNOV$, $KPRI$, $KSLI$, в которых накапливается соответственно число выполнений операторов: «перейти к п. 12» — образование новой компоненты (добавление двух вершин и одного ребра); «перейти к п. 15» — присоединение к существующей компоненте новой вершины и нового ребра; «перейти к п. 18» — слияние двух независимых компонент и добавление нового ребра. В переменных $KOST$, $KPUS$, $KSUZ$ соответственно подсчитывается число ребер в остовном лесе, число пустых проходов (число выполнений оператора «перейти к п. 28» — пропуск ребер, образующих циклы), суммарное число слитых вершин (число выполнений оператора $A(j) \leftarrow k1$ в цикле в строке 25). Все эти переменные введены для проведения дальнейшего экспериментального анализа трудоемкости алгоритма 4.7.

На рис. 4.9 показаны структуры данных $A(v)$, $H(MK,3)$, $G(v)$, $EK(MK,2)$, $B(e)$ и $W(MK)$, полученные после нахождения минимального остовного дерева на графе, изображенном на рис. 4.8. Значение $MK = 3$. В массив $G(v)$ последовательно попали вершины: 5, 4, 6, 2, 3. При этом, $H(1,1)$ содержит номер первой вершины вошедшей в компоненту 1 и указывает на номер позиции в $G(v)$, где находится следующая вершина компоненты 1. В свою очередь в этой позиции в $G(v)$ указан номер следующей позиции, где находится очередной номер вершины и т.д. пока не будет найден признак конца цепочки — 0. Элемент $H(1,2)$ указывает на номер позиции в $G(v)$, где находится последняя вершина вошедшая в компоненту 1. Аналогично устроены массивы $EK(MK,2)$ и $B(e)$.

Ребра, вошедшие в компоненту 1: 3, 9, 10, 1, 4. Первое ребро 3 указано в $EK(1,1)$, а ссылка в $EK(1,2)$ указывает, где находится последнее ребро в $B(e)$.

Размерность массивов	$A(v)$	$H(MK,3)$	$G(v)$	$EK(MK,2)$	$B(e)$	$W(MK)$
1	1	1 2 6	5	3 1	4	20
2	1	- - -	3	- -	-	-
3	1	- - -	0	- -	9	-
4	1		6		0	
5	1		4		-	
6	1		2		-	
7					-	
8					-	
9					10	
10					1	

Рис. 4.9. Структуры данных после нахождения остовного дерева

Все элементы массива $A(v)$ указывают на одну единственную компоненту 1, так как граф связный. Вес остовного дерева находится в $W(1)$.

Приведем приблизительную пооперационную оценку алгоритма по количеству выполняемых операций.

Определим трудоемкость отдельных операций:

строка 1 (операции make set) — $a \cdot v + 4a$, так как массивы G, B, и W можно не обнулять;

строки 3, 4, 31 — $2a \cdot MMAX + b \cdot MMAX + c \cdot MMAX = (2a + b + c) \cdot MMAX$. В строке 3 организован цикл по $\{ i \mid i = \overline{1, MMAX} \}$ для просмотра всех списков ребер с одинаковым весом, который может повторяться $MMAX$ раз в худшем случае, когда граф является несвязным (содержит несвязные компоненты, изолированные вершины);

строки 5, 6, 28, 30 — $a \cdot e + b \cdot e + c \cdot e + 4a \cdot e + b \cdot e = (5a + 2b + c) \cdot e$. В строке 5 организован цикл по ND для просмотра всех ребер с одинаковым весом в каждом из списков. Также как и в строке 3 он может повторяться e раз в худшем случае. Циклы по i и ND независимые, поэтому их трудоемкость складывается;

строка 7 — $(3b+c)\cdot KPUS$;

строки 8, 9, 15-17 — $[(4b+c+6b+6a+c)/2+(8a+c)]\cdot KPRI =$

$=(11a+5b+2c)\cdot KPRI$;

строки 10, 12-14 — $(7b+c+12a+c)\cdot KNOV = (12a+7b+2c)\cdot KNOV$;

строки 11, 18-27 —

$[8b+c+b+3a/2+11a+(2a+b+c)\cdot KSUZ/KSLI+a]\cdot KSLI =$

$=[13{,}5a+9b+c+(2a+b+c)\cdot KSUZ/KSLI]\cdot KSLI$.

Значения $KPUS$, $KPRI$, $KNOV$, $KSLI$ и $KSUZ$ зависят от структуры и размерности графа, их трудно выразить через основные параметры графа, поэтому в дальнейшем изложении для их определения будем использовать эмпирические методы. Очевидно, что $KOST = KPRI + KNOV + KSLI$, а $KOST + KPUS = e'$ или $KOST + KPUS = e$ — числу просмотренных ребер графа для построения остовного дерева или леса соответственно для связного и несвязного графа.

Для полноты описания реализации алгоритма 4.7, приведем псевдокод для вывода результатов его работы. Это необходимо, поскольку в большинстве практических случаев нужно знать не только суммарный вес MST-дерева, но и состав, и вес каждой его компоненты.

Алгоритм 4.8. Вывод результатов работы алгоритма 4.7

1. $SUM \leftarrow 0$.

2. Для всех $\{i \mid i = \overline{1, MK}\}$ выполнить $SUM \leftarrow SUM + W(i)$.

3. Если $SV \neq 0$, то выполнить п. 4, иначе перейти к п. 5.

4. Для всех $\{i \mid i = \overline{1, v}\}$, если $A(i) = 0$, вывести: «Вершина i изолированная».

5. $k \leftarrow 0$.

6. Для $\{i \mid i = \overline{1, MK}\}$, если $W(i) \neq 0$ выполнить пп. 7-14.

7. $k \leftarrow k+1$; Вывести: «Вес компоненты $k = W(i)$».

8. $j \leftarrow H(i,1)$.

9. Для всех j, пока $j \neq 0$, вывести: «Вершина j компоненты k», иначе перейти к п. 11.

10. $j \leftarrow G(j)$; Перейти к п. 9.

11. $j \leftarrow EK(i,1)$.

12. Для всех j, пока $j \neq 0$, вывести: «Ребро j компоненты k с концами $IS(j)$, $ST(j)$ и весом $C(j)$», иначе перейти к п. 14.

13. $j \leftarrow B(j)$; Перейти к п. 12.

14. Перейти к п. 6. *** Конец цикла 6-14 по i

15. Вывести: «Суммарный вес леса = SUM. Стоп.

Очевидно, что трудоемкость алгоритма 4.8 составляет $O(v)$.

Рассмотрим алгоритм Тарьяна [450] построения MST-деревьев с применением древовидной структуры для представления наращиваемого остовного леса с операциями слияния по рангу и сжатия путей, асимптотическая трудоемкость которого составляет $O(e'\alpha(e',v))$ и $O(e\alpha(e,v))$ соответственно для связных и несвязных графов. Будем считать, что ребра графа предварительно отсортированы алгоритмом 4.6.

Определим следующие массивы: $A(i)$, $i=\overline{1,v}$ — массив указателей на вхождение вершин i в независимые компоненты, первоначально каждая вершина указывает сама на себя; $H(i)$, $i=\overline{1,v}$ — массив, содержащий ранги вершин i; $B(i)$, $i=\overline{1,v-1}$ — массив, содержащий номера ребер MST графа.

Алгоритм 4.9. *TREE*. Построение остовного леса (дерева) с минимальным весом с использованием древовидных структур данных

1. $H \leftarrow 0$; $B \leftarrow 0$; $SUM \leftarrow 0$; $LP \leftarrow 0$; $KSLI \leftarrow 0$; $KOST \leftarrow 0$; $KPUS \leftarrow 0$; $KSUZ \leftarrow 0$.

2. Для $\{ i \mid i=\overline{1,v} \}$ выполнить $A(i) \leftarrow i$.

3. Для $\{ i \mid i=\overline{1,MMAX} \}$ выполнить пп. 4-14.

4. $ND \leftarrow U(i,1)$. *** Выбрать номер первого ребра из списка ребер с одинаковым весом

5. Пока $ND \neq 0$ выполнить пп. 6-13.

6. $l \leftarrow IS(ND)$; $k \leftarrow ST(ND)$.

7. $KSLI \leftarrow KSLI + 1$.

8. Вызвать $FND(l)$; Вызвать $FND(k)$.

9. Если $l \neq k$, то перейти к п. 10, иначе $KPUS \leftarrow KPUS + 1$, перейти к п. 11.

10. Вызвать $UN(k,l)$; $KOST \leftarrow KOST + 1$; $LP \leftarrow LP + 1$; $B(LP) \leftarrow ND$; $SUM \leftarrow SUM + C(ND)$.

11. Если $KOST = v - 1$, то перейти к п. 15.

12. $ND \leftarrow E(ND)$.

13. Перейти к п. 5. *** конец цикла по ND

14. Перейти к п. 3. *** конец цикла по i

15. Стоп.

1. Процедура $FND(x)$. *** x, y, z — целые

2. $z \leftarrow x$. *** Ищем корень для x

3. Пока $A(x) \neq x$ выполнить $x \leftarrow A(x)$. *** x — корень, первый проход

4. Пока $A(z) \neq x$ выполнить п. 5.

5. $y \leftarrow z$; $z \leftarrow A(z)$; $A(y) \leftarrow x$; $KSUZ \leftarrow KSUZ + 2$. *** Обновляем указатели на корень, второй проход

6. Возвратить x в точку вызова. *** Конец процедуры $FND(x)$

1. Процедура $UN(x, y)$. *** x, y — целые

2. Если $H(x) < H(y)$, то $A(x) \leftarrow y$, перейти к п. 5, иначе перейти к п. 3.

3. Если $H(x) > H(y)$, то $A(y) \leftarrow x$, перейти к п. 5, иначе перейти к п. 4.

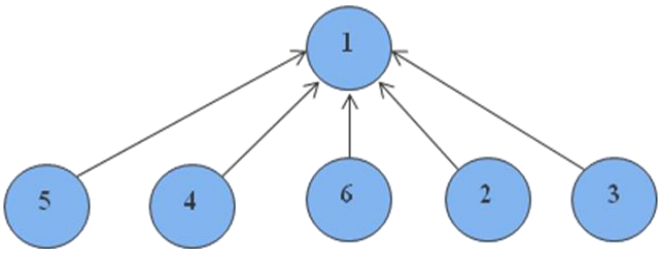
4. $A(x) \leftarrow y$; $H(y) \leftarrow H(y) + 1$.

5. Возврат в точку вызова. *** Конец процедуры $UN(x, y)$

В алгоритме 4.9 переменные $KOST$ и $KPUS$ имеют то же назначение, что и в алгоритме 4.7, а в переменных $KSLI$, $KSUZ$ и SUM накапливается соответ-

ственно число просмотренных ребер, число циклических проходов для нахождения корней деревьев и обновления указателей и суммарный вес остовного дерева или леса. Значение $KSLI = KOST + KPUS$ и равно e' или e для связного и несвязного графа. Чтобы не усложнять алгоритм Тарьяна в нем не формируются массивы подобные $H(MK,3)$, $G(v)$, $EK(MK,2)$, $B(e)$ и $W(MK)$ для хранения вершин, ребер и весов для компонент несвязного графа. Поэтому при дальнейшем экспериментальном сравнении быстродействия алгоритмов 4.7 и 4.9 из алгоритма 4.7 были убраны операторы формирования этих массивов и введены операторы $LP \leftarrow LP+1$, $B(LP) \leftarrow ND$, $SUM \leftarrow SUM + C(ND)$ как в строке 10 алгоритма 4.9.

На рис. 4.10 показано изменение значений массивов $A(i)$ и $H(i)$, $i = \overline{1,6}$, при нахождении минимального остовного дерева алгоритмом 4.9 на графе, изображенном на рис. 4.8. Жирным шрифтом выделены ребра, вошедшие в MST. Пустые клетки означают, что значения $A(i)$ и $H(i)$ не изменяются. Ребра (2,5), (2,6), (1,4) и (2,4) образуют циклы, а ребро (3,6) не просматривается, поскольку MST-дерево уже построено. Вес MST равен 20.



№ вершины	Начальные значения $A(i), H(i)$	Значения $A(i), H(i)$ при просмотре ребер:								
		(1,5)	**(4,5)**	**(5,6)**	**(1,2)**	(2,5)	(2,6)	(1,4)	(2,4)	**(2,3)**
1	1,0	1,1								
2	2,0				1,0					
3	3,0									1,0
4	4,0		1,0							
5	5,0	1,0								
6	6,0			1,0						

Рис. 4.10. Структуры данных после нахождения остовного дерева

Для вывода результатов работы алгоритма 4.9 использован следующий алгоритм.

Алгоритм 4.10. Вывод результатов работы алгоритма 4.9

1. $W \leftarrow 0;\ K \leftarrow 0$.

2. Для $\{ i \mid i = \overline{1,v} \}$ выполнить пп. 3-15.

3. Если $A(i) = i$, то перейти к п. 4, иначе к п. 15.

4. $K \leftarrow K + 1$; Вывести «Компонента K с корнем в вершине i».

5. Для $\{ l \mid l = \overline{1,v} \}$ выполнить пп. 6-13.

6. Если $A(l) = i$, то перейти к п. 7, иначе к п. 13.

7. Вывести «Вершина l».

8. Для $\{ j \mid j = \overline{1,KOST} \}$ выполнить пп. 9-12.

9. Если $IS(B(j)) = l$, то перейти к п. 10, иначе к п. 12.

10. Вывести «Ребро номер $B(j)$, с концами $IS(B(j))$, $ST(B(j))$ и весом $C(B(j))$».

11. $W(K) \leftarrow W(K) + C(B(j))$.

12. Перейти к п. 8. *** Конец цикла по j

13. Перейти к п. 5. *** Конец цикла по l

14. Вывести «Вес компоненты $K = W(K)$». *** Если $W(K) = 0$, то вершина i — изолированная

15. Перейти к п. 2. *** Конец цикла по i

16. Вывести «Суммарный вес = SUM ». Стоп.

Алгоритм 4.10 содержит три вложенных цикла по v и его трудоемкость равна $O(v^3)$. Таким образом, сложность обратного хода — отредактированной распаковки массива $B(i)$, $i = \overline{1,v-1}$, содержащего MST графа оказывается выше, чем сложность прямого хода — построения MST. Конечно же, можно придумать алгоритм распаковки $B(i)$ с меньшей трудоемкостью, например с трудоемкостью $O(v)$. Для этого нужно последовательно просматривать номера ре-

бер в $B(i)$, определять $A(IS(B(i)))$ — отца для каждого ребра i и выводить $A(IS(B(i)))$, номер ребра $B(i)$, номера левой $IS(B(i))$ и правой $ST(B(i))$ вершины, вес ребра $C(B(i))$. При этом нужно дополнительно позаботиться о выводе изолированных вершин (если они могут появляться) и о том, как выводить компоненты леса не вперемешку, а в отредактированном виде.

4.4.3. Экспериментальное исследование алгоритмов

Результаты сравнения алгоритмов опубликованы в [125, 126].

Для проведения эксперимента были составлены главный модуль программы и четыре модуля (подпрограммы), реализующие алгоритм сортировки 4.6 (*SORT*), алгоритм 4.7 (*LINK*) [103], алгоритм 4.9 (*TREE*) и алгоритм Прима в классической матричной форме (*PRIM*). Все тестовые программы написаны на языке Digital Visual Fortran (DVF) компании Digital Equipment Corporation в среде Microsoft (MS) Developer Visual Studio (VS) DVF 6.1. Вычислительный эксперимент проводился на ПК с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб под управлением операционной системы Windows Vista.

В главном модуле программы в режиме диалога вводились: число вершин графа — v; число исходящих из каждой вершины ребер (степень или валентность вершины) — *VAL*; границы изменения значений весов ребер от *MINVAL* до *MAXVAL*; параметр, управляющий выводом входных и выходных данных. Далее с помощью датчика псевдослучайных чисел (встроенной функции языка *RAND*()) генерировались веса ребер от *MINVAL* = 80 до *MAXVAL* = 800, формировались массивы IS, ST, C. Вся оперативная память, необходимая для работы алгоритмов выделялась и освобождалась динамически в главном модуле программы. Время работы алгоритмов фиксировалось встроенной подпрограммой $cpu_time(T)$ непосредственно до входа и после выхода из модулей сортировки и построения минимального остовного дерева. Работа алгоритмов проверялась на неориентированных полных графах с числом вершин v от 1000 (499500 ребер) до 12000 (71994000 ребер), так как для меньших значений v

подпрограмма $cpu_time(T)$ не могла фиксировать время выполнения (подпрограмма возвращает время в секундах с точностью до двух знаков после запятой). Результаты эксперимента приведены в табл. 4.2 и на рис. 4.11.

Таблица 4.2

Время построения минимального остовного дерева полного графа в зависимости от числа вершин в секундах

№	1	2	3	4	5	6	7	8	9	10	11	12	Среднее значение
e (млн.)	0,4995	1,999	4,4985	7,998	12,4975	17,997	24,4965	31,996	40,4955	49,995	60,4945	71,994	
v	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	12000	
e'	3993	8276	15637	24805	23820	38117	39228	38898	48523	53627	65813	77612	
%	0,80	0,41	0,35	0,31	0,19	0,21	0,16	0,12	0,12	0,11	0,11	0,11	0,25
KOST	999	1999	2999	3999	4999	5999	6999	7999	8999	9999	10999	11999	
%	25,02	24,15	19,18	16,12	20,99	15,74	17,84	20,56	18,55	18,65	16,71	15,46	19,08
KPUS	2994	6277	12638	20806	18821	32118	32229	30899	39524	43628	54814	65613	
%	74,98	75,85	80,82	83,88	79,01	84,27	82,16	79,44	81,45	81,35	83,29	84,54	80,92
KPRI	576	1322	2228	3246	4198	5174	6222	7228	8224	9174	10164	11192	
%	14,43	15,97	14,25	13,09	17,63	13,57	15,86	18,58	16,95	17,11	15,44	14,42	15,61
KNOV	212	339	386	377	401	413	389	386	388	413	418	404	
%	5,31	4,10	2,47	1,52	1,68	1,08	0,99	0,99	0,80	0,77	0,64	0,52	1,74
KSLI	211	338	385	376	400	412	388	385	387	412	417	403	
%	5,28	4,08	2,46	1,51	1,68	1,08	0,99	0,99	0,80	0,77	0,63	0,52	1,73
KSUZL	1040	1656	1880	2332	2364	2597	2800	3068	3050	3004	2873	2754	
Число слитых вершин за одну итерацию	5	5	5	6	6	6	7	8	8	7	7	7	6
KSUZT	1288	2240	2642	3180	3346	3636	3818	4158	4164	4396	4324	4236	
SORT	0,02	0,02	0,03	0,06	0,09	0,09	0,16	0,19	0,22	0,28	0,34	0,37	
LINK	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,02	0,02	0,00	0,00	0,02	
TREE	0,00	0,00	0,02	0,00	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,03	
LSUM	0,02	0,02	0,03	0,06	0,09	0,11	0,16	0,21	0,24	0,28	0, 34	0,39	
TSUM	0,02	0,02	0,05	0,06	0,11	0,11	0,18	0,21	0,24	0,30	0,36	0,40	
PRIM	0,00	0,00	0,02	0,03	0,06	0,06	0,09	0,12	0,16	0,19	0,22	0,23	
MST	80351	160083	240013	319950	399930	479925	559921	639920	719920	799920	879920	959920	

Где e' — число ребер, просмотренных для полного построения остовного дерева. Для *KOST*, *KPUS*, *KPRI*, *KNOV*, *KSLI* проценты указаны по отношению к e', а для e' проценты указаны по отношению к e. Для *KSUZ* в алгоритме *LINK* (*KSUZL* в таблице) указано суммарное число слитых вершин (число выполнений оператора $A(J) \leftarrow K1$ в цикле в строке 25) и среднее число слитых вершин за одну итерацию. Для алгоритма *TREE* параметр *KSUZ* обозначен как

KSUZT. Кроме того в таблице приняты следующие обозначения: *SORT* — время сортировки ребер алгоритмом *SORT*; *LINK*, *LSUM* и *TREE*, *TSUM* — соответственно время построения MST-дерева и общее время для предложенного алгоритма и алгоритма Тарьяна; *PRIM* — время работы алгоритма Прима; *MST* — вес *MST*.

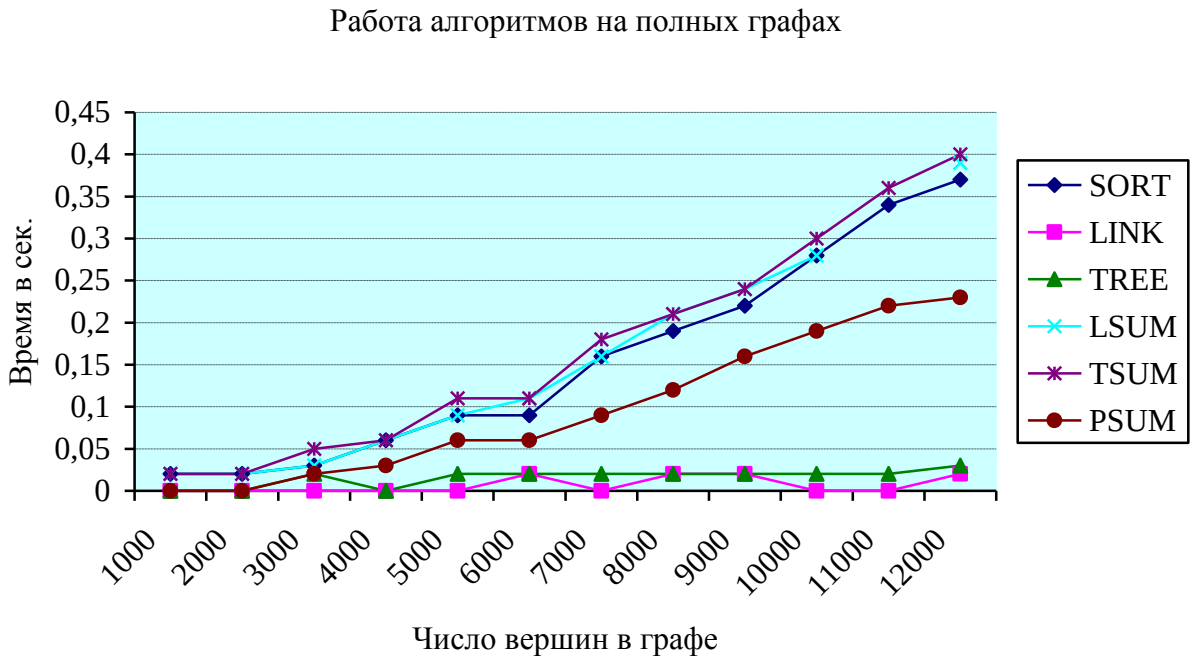


Рис. 4.11. Время построения MST-дерева различными алгоритмами

Из табл. 4.2 видно, что процесс построения минимального остовного дерева для связных графов заканчивается при просмотре в среднем не более 0,25% ребер графа, а средние значения в процентах для *KPUS*, *KPRI*, *KNOV*, *KSLI* приблизительно равны 81%, 15%, 2%, 2%. При этом среднее число слитых вершин за одну операцию слияния равно 6. Время работы алгоритмов *LINK* и *TREE* практически совпадает, растет очень медленно и для полного графа с 12 тыс. вершин не превысило 0,03 с. (см. также рис. 4.11). Общее время построения MST-дерева главным образом определяется временем сортировки ребер в алгоритме *SORT*. Поэтому для связных графов можно практически принять, что общая сложность алгоритмов построения MST-дерева соизмерима со сложностью алгоритма *SORT* для сортировки ребер $O(e)$. Из таблицы и графиков также видно, что на полных графах лучшим остается алгоритм Прима.

При тестировании алгоритмов *LINK* и *TREE* генерировались также разреженные связные и несвязные графы с различным числом вершин и ребер, и результаты вычислений сравнивались с результатами, полученными для этих графов алгоритмом Прима (*PRIM*). Результаты эксперимента для связного и несвязного графа содержащего 11000 вершин при изменении степени вершин (параметра *VAL*) показаны в табл. 4.3, рис. 4.12 и в табл. 4.4, рис. 4.13.

Таблица 4.3

Сравнение алгоритмов *PRIM*, *LINK* и *TREE* для разреженных связных графов при v = 11000; *VAL* = 5, 2000, 3000, 4000,…, 10000, 10999; *MINVAL* = 80; *MAXVAL* = 800

VAL	2000	3000	4000	5000	6000	7000	8000	9000	10000	10999	5
SORT	0,06	0,11	0,14	0,16	0,20	0,22	0,27	0,30	0,31	0,33	0,00
LINK	0,02	0,02	0,02	0,00	0,02	0,02	0,02	0,02	0,02	0,02	0,00
TREE	0,03	0,03	0,02	0,03	0,03	0,02	0,02	0,02	0,02	0,03	0,00
LSUM	0,08	0,13	0,16	0,16	0,22	0,24	0,29	0,32	0,33	0,35	0,00
TSUM	0,09	0,14	0,16	0,19	0,23	0,24	0,29	0,32	0,33	0,36	0,00
PRIM	0,27	0,25	0,20	0,23	0,22	0,23	0,22	0,22	0,20	0,22	0,25
e' (%)	0,56	0,36	0,32	0,31	0,19	0,15	0,12	0,11	0,11	0,11	84,73
KPUS (%)	82,06	81,49	84,51	87,01	82,77	80,82	79,72	79,60	81,89	83,94	52,80
KSUZL	9432 (5)	7413 (5)	6435 (5)	5458 (6)	4537 (6)	4209 (6)	3842 (7)	3731 (7)	3560 (7)	3081 (7)	20067 (6)
KSUZT	12114	9858	8658	7428	6458	5912	5460	5120	5056	4478	21622
MST	880951	880209	879983	879937	879923	879921	879920	879920	879920	879920	2117386

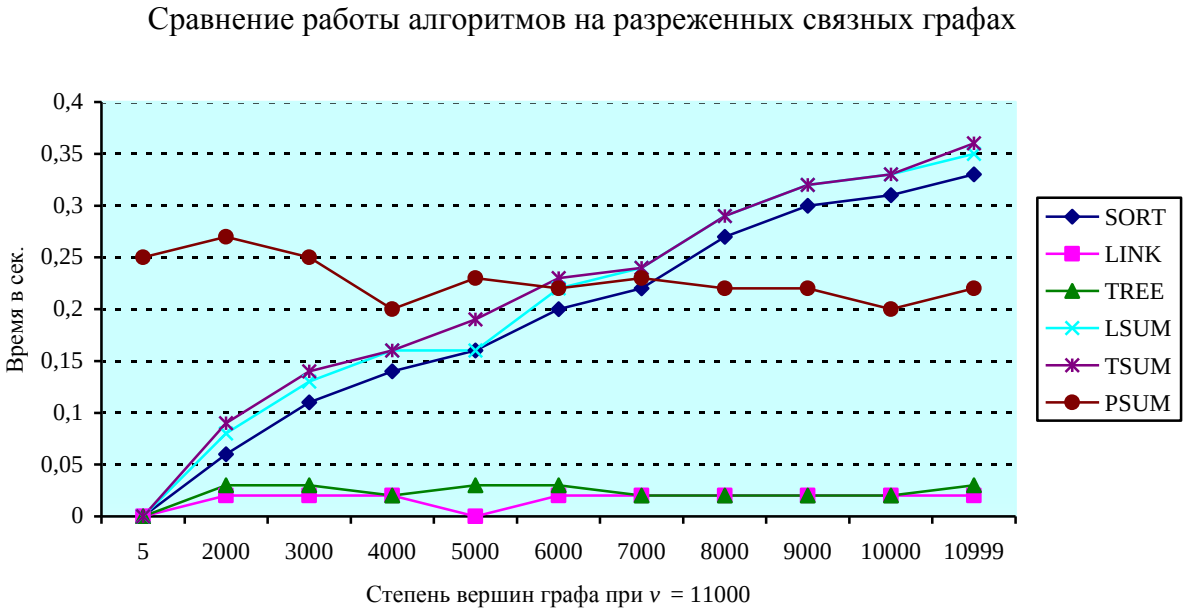


Рис. 4.12. Время построения MST-дерева различными алгоритмами

Таблица 4.4

Сравнение алгоритмов *PRIM*, *LINK* и *TREE* для разреженных несвязных графов при v = 11000; *VAL* = 50, 100, 200,…, 800, 900; *MINVAL* = 80; *MAXVAL* = 800

VAL	50	100	200	300	400	500	600	700	800	900
SORT	0,00	0,02	0,00	0,02	0,02	0,02	0,03	0,03	0,03	0,05
LINK	0,00	0,05	0,12	0,19	0,27	0,36	0,44	0,51	0,61	0,66
TREE	0,03	0,08	0,19	0,33	0,44	0,56	0,67	0,80	0,90	1,01
LSUM	0,00	0,07	0,12	0,21	0,29	0,38	0,47	0,54	0,64	0,71
TSUM	0,03	0,10	0,19	0,35	0,46	0,58	0,70	0,83	0,93	1,06
PRIM	0,27	0,25	0,25	0,23	0,25	0,25	0,27	0,25	0,25	0,25
e' (%)	100	100	100	100	100	100	100	100	100	100
KPUS (%)	96,0	98,0	99,0	99,33	99,5	99,6	99,67	99,71	99,75	99,78
KSUZL	14314(5)	13732(5)	13032(4)	13387(5)	13482(5)	12604(4)	12153(4)	12195(4)	12121(4)	12451(5)
KSUZT	16296	16600	15512	15826	15596	14986	14208	14246	14430	14996
MST	979860	929916	904535	896458	892537	889916	888341	886969	885949	885028

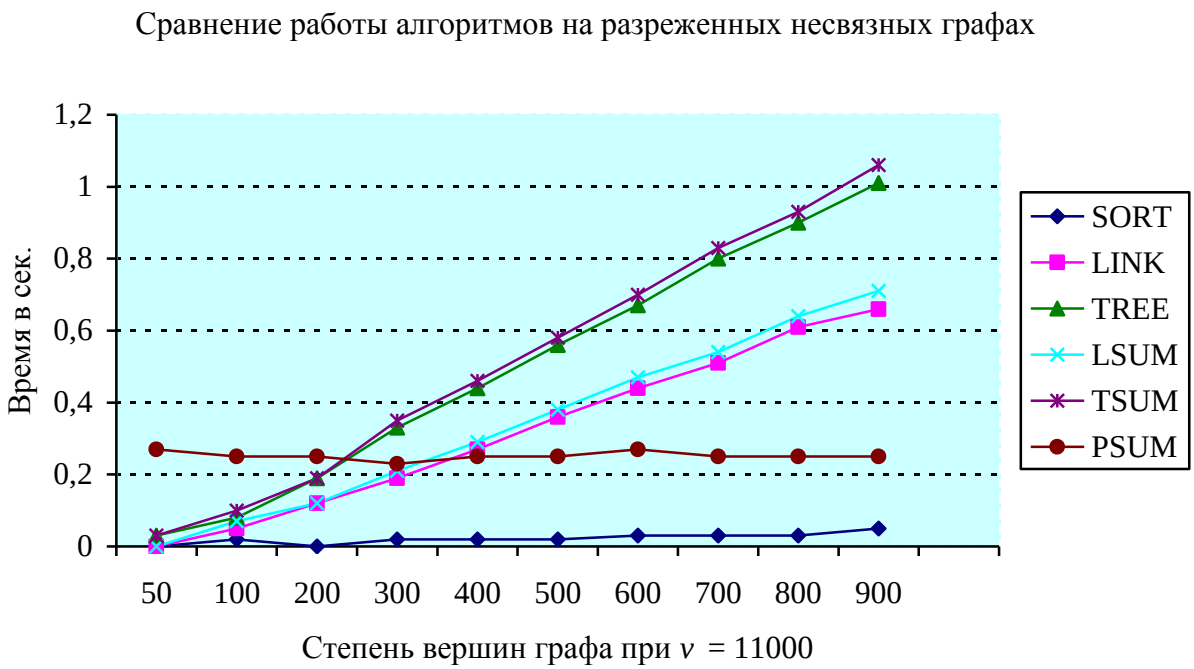


Рис. 4.13. Время построения MST-дерева различными алгоритмами

В табл. 4.3 и 4.4 для общего числа слияний вершин в алгоритме *LINK* (параметр *KSUZL*), в скобках указано среднее число слитых вершин за одну операцию слияния.

Из приведенных данных видно, что алгоритмы *LINK* и *TREE* с учетом времени на сортировку ребер на разреженных связных графах работают быстрее алгоритма Прима (*PRIM*) при $VAL \le 6000$ и числе ребер $e = 0{,}5 \cdot VAL \cdot v$ не

больше, чем $0{,}27v^2$. Полученная оценка уточнена и лучше, чем показанная в работе [30]. Для сильно разреженных графов быстродействие алгоритмов *LINK* и *TREE* с учетом времени сортировки может значительно превышать быстродействие алгоритма Прима — в 15 и более раз. Для несвязных разреженных графов алгоритмы *LINK* и *TREE* с учетом времени сортировки лучше алгоритма Прима при $VAL \le 300$, $e \le 0{,}014v^2$ и $VAL \le 200$, $e \le 0{,}009v^2$ соответственно.

В табл. 4.5, рис. 4.14 показаны результаты сравнения алгоритмов на несвязных графах, содержащих полные несвязные подграфы или изолированные вершины.

Таблица 4.5

Сравнение алгоритмов *PRIM*, *LINK* и *TREE* для несвязных графов при v = 1000, 2000,…, 12000; $VAL = v - 1$; $MINVAL$ = 80; $MAXVAL$ = 800

v	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	12000
SORT	0,00	0,02	0,03	0,05	0,08	0,11	0,16	0,23	0,28	0,33	0,41	0,47
LINK	0,03	0,23	0,59	1,23	1,86	2,50	3,51	4,57	5,77	7,11	8,61	10,26
TREE	0,08	0,37	0,98	1,83	2,75	4,10	5,54	7,30	9,13	11,14	13,65	16,11
PRIM	0,00	0,00	0,02	0,03	0,06	0,06	0,09	0,12	0,16	0,19	0,22	0,23
KPUS (%)	99,8	99,9	99,93	99,95	99,96	99,97	99,97	99,97	99,98	99,98	99,98	99,98
MST	80272	160004	239933	319870	399850	479845	559841	639840	719840	799840	879840	959840

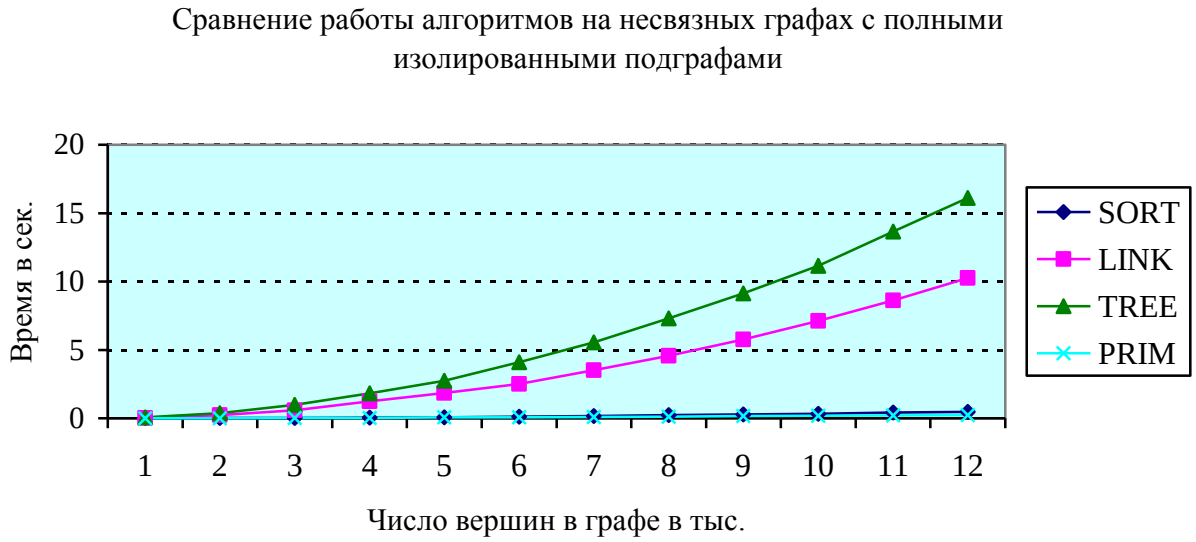


Рис. 4.14. Время построения MST-дерева различными алгоритмами

Из рис. 4.13 и 4.14 видно, что на несвязных графах алгоритм *LINK* при

увеличении степени вершин начинает заметно опережать алгоритм *TREE*.

На связных разреженных графах (табл. 4.3) при увеличении числа ребер (значений *VAL* от 2000 до 10999) процентные значения *KPUS*, *KPRI*, *KNOV*, *KSLI* приближаются к средним значениям для полных графов: 81%, 15%, 2%, 2%, среднее отношение $(e'/e)\cdot 100\%$ к 0,25%, а число слитых вершин за одну операцию слияния к 6.

Проведем пооперационную оценку средней трудоемкости алгоритма *LINK* для связных графов с использованием данных из таблицы 4.2. Примем для средних значений в процентах для *KPUS*, *KPRI*, *KNOV*, *KSLI* соответственно значения: 81%, 15%, 2%, 2%. Тогда общая трудоемкость алгоритма *LINK* для связного графа при среднем числе слитых вершин за одну операцию слияния равным 6 составит

$$T_{MST} = a\cdot v + 4a + (2a+b+c)\cdot MMAX + (5a+2b+c)\cdot e' +$$
$$+ (3b+c)\cdot e'\cdot 0{,}81 + (11a+5b+2c)\cdot e'\cdot 0{,}15 + (12a+7b+2c)\cdot e'\cdot 0{,}02 +$$
$$+ [13{,}5a+9b+c+(2a+b+c)\cdot 6]\cdot e'\cdot 0{,}02 =$$
$$= (7{,}4a+5{,}62b+2{,}29c)\cdot e' + a\cdot v + (2a+b+c)\cdot MMAX + 4a =$$
$$= (0{,}01850a+0{,}01405b+0{,}00573c)\cdot e + a\cdot v + (2a+b+c)\cdot MMAX + 4a.$$

Для несвязного графа число пустых проходов *KPUS* алгоритма *LINK* с увеличением плотности графа увеличивается от 96% до 99,78% (табл. 4.4) и от 99,8% до 99,98% (табл. 4.5). Примем в среднем число пустых проходов *KPUS* = 99,75% в предположении, что остовный лес будет построен при просмотре 0,25% от общего числа ребер в графе при выполнении операций *KPRI*, *KNOV*, *KSLI* и среднем числе слитых вершин за одну операцию слияния равным 6. Тогда для общей трудоемкости алгоритма *LINK* для несвязного графа можно записать:

$$T_{MST} = a\cdot v + 4a + (2a+b+c)\cdot MMAX + (5a+2b+c)\cdot e +$$
$$+ (3b+c)\cdot e\cdot 0{,}9975 + [(11a+5b+2c) + (12a+7b+2c) +$$
$$+ (13{,}5a+9b+c+(2a+b+c)\cdot 6)]\cdot e\cdot 0{,}0025 =$$
$$= (5{,}12125a+3{,}06b+1{,}025c)\cdot e + a\cdot v + (2a+b+c)\cdot MMAX + 4a.$$

Из полученной оценки и графиков на рис. 4.13 и 4.14 следует, что временная сложность построения MST для несвязных графов больше сложности сортировки ребер алгоритмом $SORT$.

Проведем асимптотическую оценку трудоемкости алгоритмов $LINK$ и $TREE$. Из табл. 4.2 видно, что

$$e' = KOST + KPUS = KPRI + KNOV + KSLI + KPUS.$$

Все операции $KPRI, KNOV, KSLI, KPUS$ выполняются в основном цикле по ND при выборе ребер для анализа. Поскольку трудоемкость каждой из этих операций в алгоритме $LINK$ составляет $O(1)$, а всего просматривается e' или e ребер соответственно для связных и несвязных графов, то их общая трудоемкость составит $O(e')$ или $O(e)$. Если учесть, что сложность операций инициализации массива A вхождения вершин в компоненты графа (операции make set) составляет $O(v)$, получим время порядка $O(e' + v)$ или $O(e + v)$. Осталось оценить трудоемкость операций обновления указателей в массиве A при выполнении операций слияния компонент с меньшим числом вершин в компоненты с большим числом вершин. В работах [2, 72, 107] показано, что трудоемкость таких операций в худшем случае составляет $O(v \log v)$. Однако, если проанализировать табл. 4.2, то очевидно, что число вершин, для которых обновляются указатели не соизмеряется с ростом функции $v \log v$. Значения $KSUZL$ в таблице для числа вершин v от 1000 до 12000 изменяются неравномерно от 1040 до 2754, а среднее число обновлений указателей за одну операцию слияния $]KSUZL / KSLI[$ изменяется в пределах от 5 до 8, где $]\cdot[$ — знак округления числа до ближайшего целого. Выразить функцию для числа обновлений указателей через параметры исходного графа v и e затруднительно, поэтому запишем ее в виде некоторой медленно растущей функции вида $\beta(KSUZL / KSLI, v)$. Растет ли эта функция быстрее или медленнее инверсной функции Аккермана $\alpha(m, v)$ (где m — общее число операций найти и объединить) в оценке трудоемкости MST-алгоритма Краскала, реализованного Тарьяном [450] с использованием техники FMS с операциями сжатия пути и объединения по рангу? Как уже от-

мечалось, Тарьян в работах [452, 453] показал, что полученная им оценка $O(m\alpha(m,v))$ асимптотически не улучшаема. Для предложенного алгоритма $LINK$, реализованного на списочных структурах с использованием массивов, экспериментально показано, что в большинстве практических случаев он работает не хуже чем алгоритм $TREE$, основанный на древовидных АТД с использованием операций сжатия и объединения по рангу. Ожидаемую полную трудоемкость алгоритма $LINK$ без трудоемкости сортировки e ребер запишем в виде

$O(e' + v + KSLI\beta(KSUZL / KSLI, v))$ для связных и

$O(e + v + KSLI\beta(KSUZL / KSLI, v))$ для несвязных графов.

Приведенные оценки согласуются с оценками, полученными выше при пооперационном анализе алгоритма. По определению мы приняли

$$[13{,}5a + 9b + c + (2a + b + c) \cdot KSUZ / KSLI] \cdot KSLI \equiv KSLI\beta(KSUZL / KSLI, v),$$

причем $KSLI \cong 0{,}02e' \cong 0{,}02 \cdot 0{,}0025e \cong 0{,}00005e$ является очень малой величиной по сравнению с e.

В древовидной реализации MST-алгоритма Краскала $TREE$ каждая из операций $KPUS, KPRI, KNOV, KSLI$ имеет трудоемкость $O(1)$ и выполняется в процедуре merge/union. Общая трудоемкость этих операций составит $O(e')$ и $O(e)$ для связных и несвязных графов соответственно. Сложность операций make set составляет $O(v)$. Операции find set выполняются для каждого ребра дважды, т.е. имеем $2e'$ или $2e$ входов в процедуру find set. Сложность обновления указателей в древовидной структуре с операциями объединения по рангу и сжатия пути (для сжатия пути применяется двойной проход от листа до корня дерева) определяется как $\alpha(e', v)$ или $\alpha(e, v)$. Поэтому трудоемкость древовидной реализации MST-алгоритма $TREE$ без трудоемкости сортировки ребер составит $O(e' + v + e'\alpha(e', v))$ или $O(e + v + e\alpha(e, v))$. В работах Тарьяна и ван Левувена [452, 453] были рассмотрены также однопроходные варианты эвристик со сжатием пути. Однако, по мнению автора, сложность однопроходных процедур find set на практике не будет сильно отличаться от двухпроходных, так как в

оценках однопроходных процедур скрыты дополнительные константы накладных расходов на их реализацию.

На рис. 4.15 показано число операций производимых для обновления указателей алгоритмами *LINK* и *TREE* для данных из табл. 4.2.

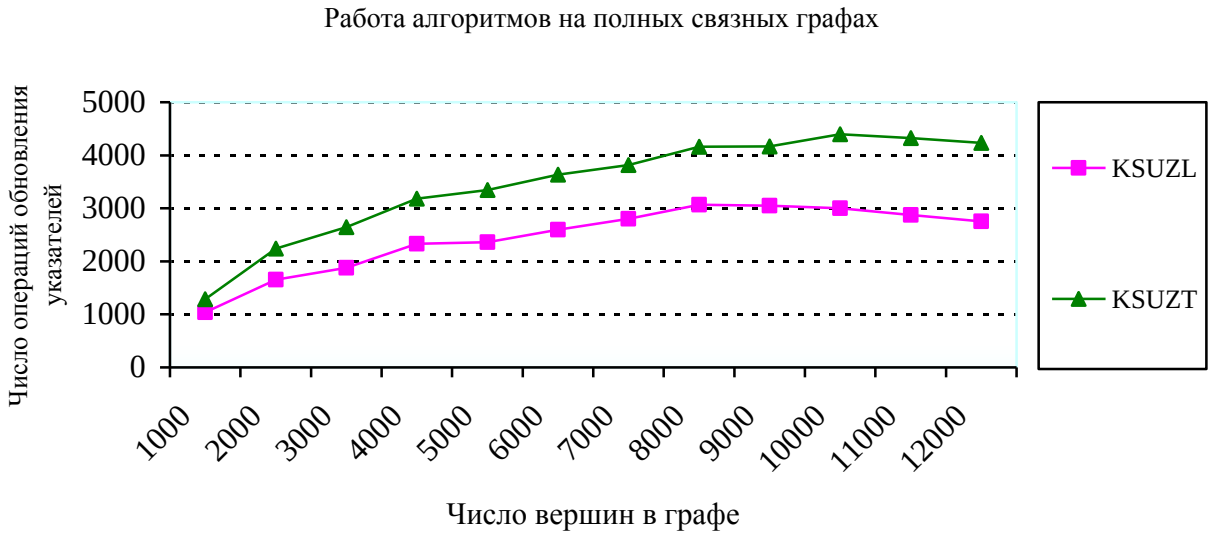


Рис. 4.15. Число операций обновления указателей при построении MST-дерева алгоритмами *LINK* (*KSUZL*) и *TREE* (*KSUZT*) в зависимости от числа вершин в полном графе

Подводя итог по оценке асимптотической сложности алгоритмов *SORT*, *LINK* и *TREE* можно сделать следующие выводы (для наглядности оценки приведены в таблице 4.6.):

1) Для связных графов практическая временная сложность обоих алгоритмов построения MST-дерева с «карманной» сортировкой (bucket sort) весов ребер составляет $O(e)$ операций, так как значения $KSLI\beta(KSUZL/KSLI, v)$, $e'\alpha(e', v)$, $(e' + v)$ нуль сравнимы со значением e, где e' — число просмотренных отсортированных ребер графа для построения MST-дерева;

2) Для несвязных графов сложность построения MST-леса с «карманной» сортировкой и использованием алгоритма *LINK* — $O(e + v + e + KSLI\beta(KSUZL/KSLI, v))$, использованием алгоритма *TREE* — $O(e + v + e + e\alpha(e, v))$. Но $KSLI\beta(KSUZL/KSLI, v)$ на практике растет медленнее чем $e\alpha(e, v)$;

3) Во всех прогонах алгоритм *LINK* оказывался немного быстрее алгоритма *TREE*;

4) Сложность предложенного алгоритма построения MST-дерева для связных графов с использованием алгоритмов *SORT* и *LINK* порядка $O(e)$ получена для практических случаев. В то же время до сих пор не известно о существовании алгоритма построения MST, для которого строго доказано, что его временная сложность составляет $O(e)$ в среднем или в худшем случае для разреженных графов. Поэтому разработка алгоритмов с гарантированной линейной оценкой для разреженных графов все еще остается недостижимой целью [107]. Интенсивному изучению подверглись различные варианты алгоритма Борувки как базиса алгоритмов для вычисления MST на сильно разреженных графах за почти линейное время, а также рандомизированные алгоритмы поиска MST, математическое ожидание времени работы которых сравнимо с $O(e+v)$ [72]. Такие исследования вселяют надежду на успех и будоражат энтузиазм разработчиков более быстрых алгоритмов построения MST;

Таблица 4.6

Асимптотические оценки сложности алгоритма Краскала с использованием АТД		
Для всех методов с сортировкой сравнением	Для сортировки методом преобразования веса ребра в адрес оперативной памяти компьютера (bucket sort) для связных графов	Для сортировки методом преобразования веса ребра в адрес оперативной памяти компьютера (bucket sort) для несвязных графов
$O(e\log e)$	$O(e)$, для алгоритмов со списочной *LINK* и древовидной *TREE* структурой данных, так как $O(KSLI\beta(KSUZL/KSLI,v)) \Rightarrow O(1)$ и $O(e'\alpha(e',v)) \Rightarrow O(1)$, а $e >> (e'+v)$. Но $KSLI\beta(KSUZL/KSLI,v)$ на практике растет медленнее чем $e'\alpha(e',v)$	$O(e+e+v+KSLI\beta(KSUZL/KSLI,v))$ с использованием алгоритма *LINK*; $O(e+e+v+e\alpha(e,v))$ с использованием алгоритма *TREE*. Разница в росте $KSLI\beta(KSUZL/KSLI,v)$ и $e\alpha(e,v)$ очевиднее

5) В целом эксперимент показал высокую вычислительную эффективность алгоритмов, которые могут с успехом применяться при решении практических задач определения оптимальных остовных деревьев (лесов) для графов и сетей большой размерности. В частности, предложенные алгоритмы могут быть

включены в состав типового инструментария разработчика программ и использоваться при решении различных задач анализа и проектирования коммуникационных сетей.

При практическом использовании алгоритмов их быстродействие можно улучшить, если выполнение операций $U \leftarrow 0$, $E \leftarrow 0$, $A \leftarrow 0$, $G \leftarrow 0$, $B \leftarrow 0$, $W \leftarrow 0$ перенести в главный модуль программы.

4.5. Алгоритмы распределения и маршрутизации потоков

В приложении Д рассматриваются вопросы построения справочной структуры H_S, необходимой для описания основных задач и процедур, используемых в алгоритме распределения и маршрутизации потоков.

Рассмотрим задачу выбора путей для распределения потоков. Предположим, что для первоначального распределения потоков транспортных блоков для всех $\tilde{a}_{ij}$, $ij \in S$ на сети G_M с помощью алгоритма 4.5 (SP2**)**, найдены кратчайшие пути, удовлетворяющие ступенчатому критерию: минимум транзитных перегрузок; при равенстве числа перегрузок — минимум длины пути. Пусть путь π_{ij} задан в виде последовательности узлов $\pi_{ij} = \{\sigma_1, \sigma_2, ..., \sigma_\varphi\}$, $\sigma_1 = i$, $\sigma_\varphi = j$. Любой путь может быть определен из справочной матрицы $C = \| c_{ij} \|_{n \times n}$. В процессе решения задачи может оказаться, что распределить некоторый поток или его часть по кратчайшему пути невозможно из-за нарушения ограничений. В этом случае для распределения всего потока или его частей должны быть найдены новые пути. В связи с этим рассмотрим задачу нахождения для $\tilde{a}_{ij}$, $ij \in S$ множества кратчайших путей $\{\pi_{ij}\}$, упорядоченных по возрастанию числа транзитных узлов и длины. При определении $\{\pi_{ij}\}$ будем отсекать такие пути, транспортировка потока по которым нарушает ограничения $t_{ij} \le T_{ij}$, $ij \in S$ и (4.13). Отметим, что проверку условий $t_{ij} \le T_{ij}$ легко осуществлять, используя формулы (3.24). Для проверки ограничений (4.13) необходимо одновременно решать l - многомерных задач о ранце вида (4.16) с блочными ограничениями

(4.10), (4.11) размерностью $(\eta_k - 1)\times\varepsilon_k$ и $\eta_k \times \varepsilon_k$, связывающими ограничениями (4.12), (4.13) и ограничениями (4.14), (4.15). Учет ограничений (4.13) в рамках «ранцевой» модели представляет сложную задачу. Для ее упрощения предположим, что в процессе решения задачи отдельные потоки распределяются последовательно. При этом имеется алгоритм, позволяющий выбрать оптимальный путь и распределить поток на соответствующие этому пути линии связи (маршруты транспортных средств или коммутированные каналы связи). Пусть $Y = \| y_i \|_n$ — вектор, в котором суммируются транзитные потоки, первоначально $Y = 0$. Тогда можно записать:

$$\pi_{ij} = \{\pi_{ij} \mid t_{ij} \le T_{ij}; \tilde{a}_{ij} + y_\sigma \le b_\sigma\},\ \sigma \in \{\sigma_2,...,\sigma_{\varphi-1}\}. \qquad (4.32)$$

Для нахождения K- кратчайших путей между двумя заданными узлами известно достаточно много алгоритмов [2, 5, 72, 78, 107, 381]. Наиболее эффективными из них являются алгоритм Йена [74] и алгоритм двойного поиска Шиера [134]. Сложность алгоритмов оценивается порядком $O(Kn^3)$, где n — число узлов в сети. Совершенно ясно, что применять эти алгоритмы для нахождения K- кратчайших путей для $(n^2 - n)$ потоков немыслимо из-за большой трудоемкости. Поэтому рассмотрим простой алгоритм, позволяющий получить достаточно хорошие пути за время $O(n)$. Работа алгоритма основана на использовании справочной матрицы путей C и заключается в следующем.

Алгоритм 4.11. Выбор альтернативных путей

1. Для заданных i и j $\{\pi_{ij}\} \leftarrow \pi_{ij}^*$ (где π_{ij}^* — кратчайший путь).

2. Для $\{k \mid k = \overline{1,n}, k \ne i, k \ne j\}$ выполнить п. 3.

3. Используя c_{ik} и c_{kj} получить для π_{ij} последовательность $\{\sigma_1,...,\sigma_\xi,...,\sigma_\varphi\}$, $\sigma_1 = i$, $\sigma_\xi = k$, $\sigma_\varphi = j$. Если для π_{ij} выполняются условия (4.32) и в $\{\pi_{ij}\}$ нет такого же пути, то $\{\pi_{ij}\} \leftarrow \{\pi_{ij}\} \cup \pi_{ij}$.

4. Упорядочить пути в $\{\pi_{ij}\}$ в порядке возрастания числа узлов и длины.

Нетрудно убедиться, что множество $\{\pi_{ij}\}$, сформированное приведенным

алгоритмом, содержит все пути с одним транзитным узлом. Пути с двумя и более транзитными узлами представлены в $\{\pi_{ij}\}$ не полными возможными подмножествами. Учитывая то, что для реальных магистральных сетей число транзитных перегрузок транспортных блоков, как правило, не превышает трех и четырех, можно сделать вывод о практической приемлемости построения $\{\pi_{ij}\}$ указанным способом. В случае необходимости получения для некоторых потоков полного множества $\{\pi_{ij}\}$ можно воспользоваться алгоритмами из [74, 134].

Другой важной задачей, которую необходимо рассмотреть до приведения основного алгоритма, является задача проверки ограничений (4.10) и (4.11) и выбора линии связи для распределения потока. Пусть для каждого потока $\tilde{a}_{ij}$, $ij \in S$ построено множество $\{\pi_{ij}\}$. Для каждого $\pi_{ij} \in \{\pi_{ij}\}$ известна последовательность $\{\sigma_1,\sigma_2,...,\sigma_\varphi\}$ узлов, определяющая путь транспортировки потока. Для любой пары $(\sigma_\psi,\sigma_{\psi+1})$, $\psi = \overline{1,\varphi-1}$ используя справочную H_S нетрудно определить множество линий связи, на которых может быть выполнена транспортировка потока $\tilde{a}_{ij}$ на участке $(\sigma_\psi,\sigma_{\psi+1})$. Пусть для $(\sigma_\psi,\sigma_{\psi+1})$ выделен участок некоторой k-ой линии связи — $(\xi^k_\mu,...,\xi^k_{\mu+c})$, где $(\xi^k_\mu = \sigma_\psi \wedge \xi^k_{\mu+c} = \sigma_{\psi+1}) \vee (\xi^k_\mu = \sigma_{\psi+1} \wedge \xi^k_{\mu+c} = \sigma_\psi)$. Определим для $k = \overline{1,l}$ векторы OF^k_ε, XF^k_ε, OB^k_ε, XB^k_ε, $\varepsilon = \overline{1,\eta_k}$, LF^k_ε, LB^k_ε, $\varepsilon = \overline{1,\eta_k - 1}$. Векторы OF^k_ε, OB^k_ε содержат значения правых частей (4.11), а в векторах XF^k_ε, XB^k_ε суммируются для каждого узла объемы погружаемых и выгружаемых транспортных блоков, распределенных на линию связи k. Обе пары векторов поставлены во взаимно однозначное соответствие υ_k — упорядоченному множеству узлов из N на линии m_k и определяют объемы переработки транспортных блоков при транспортировке потоков в прямом OF^k_ε, XF^k_ε и обратном OB^k_ε, XB^k_ε направлениях. Векторы LF^k_ε и LB^k_ε соответствуют q_k — упорядоченному множеству топологических дуг из P, составляющих линию m_k и определяют их текущую загрузку.

С учетом изложенного, ограничения (4.10) запишем в виде:

$$LF_{\varepsilon}^{k} + \tilde{a}_{ij} \le w_{\varepsilon}^{k},\ LB_{\varepsilon}^{k} + \tilde{a}_{ij} \le w_{\varepsilon}^{k},\ \varepsilon = \overline{\mu, \mu + c - 1}, \tag{4.33}$$

а ограничения (4.11) в виде

$$XF_{\mu}^{k} + \tilde{a}_{ij} \le OF_{\mu}^{k},\ XF_{\mu+c}^{k} + \tilde{a}_{ij} \le OF_{\mu+c}^{k}, \tag{4.34}$$

$$XB_{\mu}^{k} + \tilde{a}_{ij} \le OB_{\mu}^{k},\ XB_{\mu+c}^{k} + \tilde{a}_{ij} \le OB_{\mu+c}^{k}.$$

Определим далее величины:

$$MINT^{k} = \begin{cases} \sum_{\varepsilon=\mu}^{\mu+c-1} w_{\varepsilon}^{k} - (LF_{\varepsilon}^{k} + \tilde{a}_{ij}) \\ \sum_{\varepsilon=\mu}^{\mu+c-1} w_{\varepsilon}^{k} - (LB_{\varepsilon}^{k} + \tilde{a}_{ij}) \end{cases} \tag{4.35}$$

$$MAXT^{k} = \begin{cases} OF_{\mu}^{k} - (XF_{\mu}^{k} + \tilde{a}_{ij}) + OF_{\mu+c}^{k} - (XF_{\mu+c}^{k} + \tilde{a}_{ij}), \\ OB_{\mu}^{k} - (XB_{\mu}^{k} + \tilde{a}_{ij}) + OB_{\mu+c}^{k} - (XB_{\mu+c}^{k} + \tilde{a}_{ij}). \end{cases} \tag{4.36}$$

В выражениях (4.33)-(4.36) первая строка записана для случая, когда $\xi_{\mu}^{k} = \sigma_{\psi}, \xi_{\mu+c}^{k} = \sigma_{\psi+1}$, вторая — для случая, когда $\xi_{\mu}^{k} = \sigma_{\psi+1}, \xi_{\mu+c}^{k} = \sigma_{\psi}$.

Проверка ограничений на пропускную способность участков магистральных линий связи и объемы обработки транспортных блоков в узлах линий связи свелась к проверке условий (4.33) и (4.34). Если эти условия выполняются, линия m_k является допустимой для распределения рассматриваемого потока $\tilde{a}_{ij}$. Очевидно, что, используя справочную H_S, нетрудно определить множество $\{m_k^*\}$ всех допустимых линий связи. Среди множества $\{m_k^*\}$ необходимо выбрать наилучшую. Аналогичная задача, известная в литературе как «задача упаковки контейнера» рассматривалась Грехэмом и Джонсоном [8] и многими другими исследователями [2, 5, 72, 78, 107, 381]. Для ее решения было предложено несколько эвристических алгоритмов с априорными оценками точности получаемого решения в худшем случае. Ключевыми моментами предложенных алгоритмов являются стратегия выбора предмета для загрузки в контейнер и стратегия выбора самого контейнера, обеспечивающие минимизацию потребного числа контейнеров для упаковки данной совокупности предметов. Следуя указанным работам, рассмотрим несколько эвристических стратегий выбора потоков, основанных на процедурах предварительного упорядочения.

Стратегия S1: множество потоков $\tilde{a}_{ij}$, $ij \in S$ ранжируется по убыванию их величины.

Стратегия S2: множество потоков $\tilde{a}_{ij}$, $ij \in S$ ранжируется по убыванию числа транзитных узлов в кратчайших путях π_{ij}, $ij \in S$.

Стратегия S3: множество потоков $\tilde{a}_{ij}$, $ij \in S$ ранжируется по убыванию ступенчатого критерия — наибольшая величина потока; наибольшее число транзитных узлов.

Стратегия S4: выбор потоков $\tilde{a}_{ij}$, $ij \in S$ осуществляется случайно.

В качестве главного критерия выбора линии связи будем использовать минимум значения $MINT^k$, определяемого из (4.35) на множестве $\{m_k^*\}$. Обоснованием выбора такого критерия является стремление к максимальному увеличению загрузки участков линий связи (обеспечению максимума значения функции (4.16)). При выборе линии связи для транспортировки потока могут быть учтены и другие ее характеристики, такие, например, как максимум резерва по возможности погрузки-выгрузки транспортных блоков в узлах линии, определяемого из выражения (4.36); преимущество выбора линии по заданному признаку предпочтения; выбор линии, дающей более короткий по длине путь от i до j и др. Совокупность дополнительных критериев может быть упорядочена таким образом, чтобы максимально учесть практические требования при выборе линии связи.

Следует отметить, что в отличие от процедур упаковки, предложенных в [8], критерий выбора лучшей линии связи в рассматриваемом случае учитывает суммарную неиспользованную грузоподъемность транспортного средства или пропускную способность коммутированного канала на нагружаемых топологических дугах.

Пусть $m_k \in \{m_k^*\}$ — линия, выбранная для распределения потока $\tilde{a}_{ij}$ на участке $(\sigma_\psi, \sigma_{\psi+1})$. Опишем процедуру загрузки m_k, которая состоит из следующих шагов.

Алгоритм 4.12

1. Выделить $(\xi^k_\mu,...,\xi^k_{\mu+c})$.

2. Если $\sigma_{\psi+1} \neq j$, то $y_{\sigma_{\psi+1}} \leftarrow y_{\sigma_{\psi+1}} + \tilde{a}_{ij}$.

3. Выполнить

$$XF^k_\mu \leftarrow XF^k_\mu + \tilde{a}_{ij},\ XF^k_{\mu+c} \leftarrow XF^k_{\mu+c} + \tilde{a}_{ij},\ \xi^k_\mu \leftarrow \sigma_\psi,\ \xi^k_{\mu+c} \leftarrow \sigma_{\psi+1},$$

$$XB^k_\mu \leftarrow XB^k_\mu + \tilde{a}_{ij},\ XB^k_{\mu+c} \leftarrow XB^k_{\mu+c} + \tilde{a}_{ij},\ \xi^k_\mu \leftarrow \sigma_{\psi+1},\ \xi^k_{\mu+c} \leftarrow \sigma_\psi.$$

4. Для $\{\varepsilon \mid \varepsilon = \overline{\mu, \mu+c-1}\}$ выполнить

$$LF^k_\varepsilon \leftarrow LF^k_\varepsilon + \tilde{a}_{ij}, \text{ для } \xi^k_\mu = \sigma_\psi,\ \xi^k_{\mu+c} = \sigma_{\psi+1},$$

$$LB^k_\varepsilon \leftarrow LB^k_\varepsilon + \tilde{a}_{ij}, \text{ для } \xi^k_\mu = \sigma_{\psi+1},\ \xi^k_{\mu+c} \leftarrow \sigma_\psi.$$

5. Стоп.

Кроме рассмотренных задач, для описания основного алгоритма нам понадобится процедура введения новых линий связи. Такая процедура необходима в случае, когда некоторый поток $\tilde{a}_{ij}$ или его часть не могут быть распределены ни на одном из путей $\{\pi_{ij}\}$ из-за нарушения ограничений (4.33) или (4.34).

Алгоритм 4.13

1. Определить последовательность узлов на кратчайшем пути

$\pi^*_{ij} = \{\sigma_1, \sigma_2, ..., \sigma_\varphi\}$, $QS \Rightarrow NULL$, $Q1 \Rightarrow QS$, $l \leftarrow 0$.

2. Для $\{\psi \mid \psi = \overline{1, \varphi-1}\}$ выполнить пп. 3-14 .

3. Для пары $(\sigma_\psi, \sigma_{\psi+1}) \in \pi^*_{ij}$ определить участок $(\xi^k_\mu, \xi^k_{\mu+1}, ..., \xi^k_{\mu+c})$, являющийся прообразом новой линии связи.

4. $l \leftarrow l+1$.

5. Образовать новую линию $m_l = (\xi_1, \xi_2, ..., \xi_{c+1})$, $\xi_1 \leftarrow \xi^k_\mu, ..., \xi_{c+1} \leftarrow \xi^k_{\mu+c}$.

6. Сформировать для m_l векторы OF^l_ε, XF^l_ε, OB^l_ε, XB^l_ε, $\varepsilon = \overline{1, c+1}$, LF^l_ε, LB^l_ε, $\varepsilon = \overline{1, c}$, w^l_ε и другие характеристики линии.

7. Для $\{\varepsilon \mid \varepsilon = \overline{1, c+1}\}$ выполнить $XF^l_\varepsilon \leftarrow 0$, $XB^l_\varepsilon \leftarrow 0$.

8. Для $\{\varepsilon \mid \varepsilon = \overline{1, c}\}$ выполнить $LF^l_\varepsilon \leftarrow 0$, $LB^l_\varepsilon \leftarrow 0$.

9. Скорректировать справочную H_S.

10. Установить указатель $P1$ на элемент E, порожденный в списке для $\Delta_{\sigma_\psi,\sigma_{\psi+1}}$ и указывающий на новую линию m_l.

11. Если $\varphi \neq 2$, перейти к п. 12, иначе перейти к п. 16.

12. Образовать новый $E(Q)$, $Q \Rightarrow E.FWD \Rightarrow NULL$, $Q \Rightarrow E.DM_5 \leftarrow 0$, $Q \Rightarrow E.DM_3 \leftarrow P1 \Rightarrow E.DM_3$, $Q \Rightarrow E.DM_4 \leftarrow P1 \Rightarrow E.DM_4$, $Q1 \Rightarrow E.FWD \Rightarrow Q$, $Q1 \Rightarrow Q$, $Q \Rightarrow E.AY \leftarrow \sigma_{\psi+1}$, $Q \Rightarrow E.AM \leftarrow l$, пометить поле DM_1 или DM_2 элемента $Q \Rightarrow E$.

13. Выполнить процедуру загрузки линии m_l на участке $(\sigma_\psi, \sigma_{\psi+1})$ используя алгоритм 4.12.

14. Перейти к п. 2.

15. $\Delta_{ij} \Rightarrow QS$, перейти к п. 18 .

16 . Пометить поле DM_1 или DM_2 элемента $P1 \Rightarrow E$.

17. Выполнить процедуру загрузки линии m_l на участке $(\sigma_\psi, \sigma_{\psi+1})$ используя алгоритм 4.12.

18. Стоп.

Как видно из приведенного алгоритма новые линии вводятся на кратчайшем пути. Число порождаемых линий определяется величиной $(\varphi - 1)$. В шаге 3, при определении прообраза новой линии, просматривается цепочка элементов E, на которую ссылается указатель $\Delta_{\sigma_\psi,\sigma_{\psi+1}}$. Среди множества линий, определяемых из этой цепочки, выбирается та, которая имеет наилучшие характеристики по заданному критерию. Так, например, в качестве критерия может быть выбран тип линии, максимум грузоподъемности или пропускной способности и другие. Выбор прообраза линии на этом шаге может быть выполнен в интерактивном режиме или по заранее определенному критерию. В шагах 4 и 5 увеличивается счетчик линий и вводится, собственно, сама линия. На следующем шаге для новой линии копируются все характеристики ее прообраза, используемые в процессе решения задачи. При этом в качестве значений для OF_ε^l и

OB^l_ε при $\varepsilon = 1$ и $\varepsilon = c + 1$ принимается грузоподъемность транспортного средства, курсирующего на линии-прообразе, или пропускная способность коммутированного канала связи. В шаге 9 производится корректировка справочной структуры H_S, которая заключается в выполнении действий, подобных действиям алгоритма Д.2. Каждый элемент E, порождаемый для новой линии m_l, включается в соответствующие списки структуры H_S. Включение осуществляется с учетом длины $(c+1)$ новой линии по критерию не убывания их длин. После включения каждого элемента E в список, формируются его поля: AM, DM_3, DM_4 и DM_5.

Оценим временную сложность алгоритма 4.13. Наиболее трудоемкие действия алгоритма выполняются в цикле 2-14. Внутри этого цикла имеется пять шагов: 3, 7, 8, 9 и 13, представляющих интерес для определения оценки сложности, так как реализация этих шагов предполагает выполнение некоторых циклических действий. Предположим, что средняя длина списка элементов в структуре H_S равна k_1, а средняя длина линии — $k_2 = (\sum_{k=1}^{l} \eta_k)/l$. Тогда в шаге 3 в среднем просматривается k_1 линий и для выбранной k - ой линии с трудоемкостью k_2 (в худшем случае) формируется прообраз. Сложность шага оценивается величиной $k_1 + k_2$. В шагах 7 и 8 со сложностью $2k_2$ выполняется циклическое обнуление рабочих векторов, сформированных для новой линии. Анализируя алгоритм Д.2, нетрудно видеть, что корректировка справочной H_S для одной линии потребует $(k_2 - 1)k_2$ действий. В шаге 13 выполняется процедура загрузки новой линии с использованием алгоритма 4.12, сложность которого определяется величиной $2k_2$. Учитывая, что цикл 2-14 выполняется $k_3 = (\varphi - 1)$ раз, запишем общую сложность алгоритма 4.13

$$T_c = k_3(k_1 + k_2 + 2k_2 + (k_2 - 1)k_2 + 2k_2) = k_3(k_2^2 + 4k_2 + k_1).$$

Как показали экспериментальные исследования, для реальных магистральных коммуникационных сетей при распределении всех потоков по кратчайшим путям средняя величина k_3 приближается к двум. Учитывая, что для реальных

сетей, как правило, всегда выполняется неравенство $k_2 > k_1 > k_3$, можно принять, что сложность алгоритма квадратично зависит от средней длины линии связи и составляет $O(k_2^2)$.

Перейдем к описанию основного алгоритма. Пусть $\{\tilde{a}_{ij}\}$ — множество всех потоков, подлежащих распределению на сети; $\{\tilde{a}_{ij}^*\}$ — множество нераспределенных потоков, первоначально $\{\tilde{a}_{ij}^*\} = \varnothing$; π_{ij}^* — кратчайший путь от i до j; c_2, c_3 — переменные, указывающие для каждого потока число линий связи, на которых этот поток может быть распределен без нарушения ограничений (4.34) и (4.33) соответственно; QS — голова списка элементов E для транзитных потоков; Q, P, $Q1$, $P1$ — рабочие указатели; θ — номер линии, выбранной для распределения потока, первоначально θ содержит номер фиктивной линии; c_1 — текущее число порожденных элементов E в списке QS; k — номер текущей рассматриваемой линии; ν_1 — признак ослабления ограничений (4.34), $\nu_1 = 1$, если ограничения (4.34) нарушить можно, $\nu_1 = 0$ в противном случае. Пусть Z_{opt} определяет затраты на транспортировку и обработку транспортных блоков в сети, первоначально $Z_{opt} = \infty$. Для случая разветвления потоков введем в структуру элементов E дополнительные поля: AF и NF. Поле AF содержит величину части распределяемого потока, а поле NF — порядковый номер этой части потока.

До начала работы основного алгоритма проектировщик может в интерактивном режиме осуществить ввод дополнительных линий связи или указать на необходимость автоматической генерации шаблонов линий связи. Для генерации шаблонов используется отдельная программа, в которой реализованы алгоритмы, основанные на построении кратчайших путей.

Алгоритм 4.14. Распределение и маршрутизация потоков

1. Используя одну из стратегий S1-S4 выбрать $\tilde{a}_{ij} \in \{\tilde{a}_{ij}\}$.

2. Если для π_{ij}^{*} ограничения (4.32) выполняются, то перейти к п. 7, иначе перейти к п. 3.

3. Используя алгоритм 4.11 сформировать $\{\pi_{ij}\}$.

4. Если $\{\pi_{ij}\}=\varnothing$, то перейти к п. 5, иначе перейти к п. 33.

5. Поток $\tilde{a}_{ij}$ не может быть распределен из-за нарушения ограничений (4.32).

6. Если ограничения (4.32) нарушить можно, то перейти к п. 7, иначе перейти к п. 40.

7. $\pi_{ij} \leftarrow \pi_{ij}^{*}$.

8. $QS \Rightarrow NULL$, $Q1 \Rightarrow QS$, $c_1 \leftarrow 0$.

9. Определить последовательность $\pi_{ij}=\{\sigma_1,\sigma_2,...,\sigma_\varphi\}$.

10. Для $\{\psi \mid \psi=\overline{1,\varphi-1}\}$ выполнить пп. 11-47.

11. Если $(\sigma_\psi = i \wedge \sigma_{\psi+1} = j)$, то перейти к п. 13, иначе перейти к п. 12.

12. Образовать новый $E(Q)$, $Q \Rightarrow E.FWD \Rightarrow NULL$, $Q \Rightarrow E.DM_5 \leftarrow 0$, $Q1 \Rightarrow E.FWD \Rightarrow Q$, $Q1 \Rightarrow Q$, $c_1 \leftarrow c_1+1$, $Q \Rightarrow E.AY \leftarrow \sigma_{\psi+1}$.

13. $P \Rightarrow \Delta_{\sigma_\psi,\sigma_{\psi+1}}$, $c_2 \leftarrow 0$, $c_3 \leftarrow 0$, $\theta \leftarrow 0$.

14. $k \leftarrow P \Rightarrow E.AM$.

15. Если $\tilde{a}_{\sigma_\psi,\sigma_{\psi+1}}$ может быть распределен на m_k, то перейти к п. 16, иначе перейти к п. 25.

16. Определить участок $m_k=(\xi_\mu^k,\xi_{\mu+1}^k,...,\xi_{\mu+c-1}^k,\xi_{\mu+c}^k)$, где $(\xi_\mu^k=\sigma_\psi \wedge \xi_{\mu+c}^k=\sigma_{\psi+1}) \vee (\xi_\mu^k=\sigma_{\psi+1} \wedge \xi_{\mu+c}^k=\sigma_\psi)$.

17. По формуле (4.36) определить $MAXT^k$.

18. Если $\nu_1 \neq 1$, то перейти к п. 19, иначе перейти к п. 20.

19. Если условия (4.34) выполняются, то перейти к п. 20, иначе перейти к п. 25.

20. $c_2 \leftarrow c_2+1$.

21. Если условия (4.33) выполняются, то перейти к п. 22, иначе перейти к

п. 25.

22. По формуле (4.35) определить $MINT^k$.

23. $c_3 \leftarrow c_3 + 1$.

24. Сравнить m_θ и m_k по критерию сравнения линий связи. Если выбор m_k предпочтительнее чем m_θ, то $\theta \leftarrow k$, $P1 \Rightarrow P$.

25. $P \Rightarrow P \Rightarrow E.FWD$.

26. Если $P \neq NULL \wedge P \Rightarrow E.DM_5 = 1$, то перейти к п. 14, иначе перейти к п. 27.

27. Если $c_3 = 0$, то перейти к п. 28, иначе перейти к п. 43.

28. Если $c_1 = 0$, то перейти к п. 29, иначе перейти к п. 30.

29. Уничтожить порожденные элементы E в списке QS.

30. Если $\{\pi_{ij}\}$ не формировалось, то, используя алгоритм 4.11, найти $\{\pi_{ij}\}$, перейти к п. 31, иначе перейти к п. 31.

31. $\{\pi_{ij}\} \leftarrow \{\pi_{ij}\} \setminus \pi_{ij}$.

32. Если $\{\pi_{ij}\} \neq \varnothing$, то перейти к п. 33, иначе перейти к п. 34.

33. Выбрать очередной $\pi_{ij} \in \{\pi_{ij}\}$. Перейти к п. 8.

34. Если $c_2 = 0$, то перейти к п. 35, иначе перейти к п. 37.

35. Поток $\tilde{a}_{ij}$ не может быть распределен из-за нарушения ограничений (4.34).

36. Если ограничения (4.34) нарушить можно, то перейти к п. 41, иначе перейти к п. 38.

37. Поток $\tilde{a}_{ij}$ не может быть распределен из-за нарушения ограничений (4.33).

38. Если образовать новую линию связи можно, то перейти к п. 39, иначе перейти к п. 40.

39. Выполнить процедуру образования новых линий связи, используя алгоритм 4.13. Перейти к п. 56.

40. Пометить нераспределенный поток $\{\tilde{a}_{ij}^{*}\} \leftarrow \tilde{a}_{ij}$. Перейти к п. 56.

41. $\nu_1 \leftarrow 1$.

42. Восстановить $\{\pi_{ij}\}$. Перейти к п. 33.

43. Если $\varphi = 2$, то перейти к п. 44, иначе перейти к п. 46.

44. Пометить поле DM_1 или DM_2 элемента $P1 \Rightarrow E$. В поля AF и NF занести соответствующие значения.

45. Выполнить процедуру загрузки линии m_θ на участке $(\sigma_\psi, \sigma_{\psi+1})$ с помощью алгоритма 4.12. Перейти к п. 56.

46. $Q \Rightarrow E.AM \leftarrow \theta$, $Q \Rightarrow E.DM_3 \leftarrow P1 \Rightarrow E.DM_3$, $Q \Rightarrow E.DM_4 \leftarrow P1 \Rightarrow E.DM_4$. Пометить поле DM_1 или DM_2 элемента $Q \Rightarrow E$. В поля AF и NF занести соответствующие значения.

47. Перейти к п. 10.

48. Если $\Delta_{ij} = NULL$, то перейти к п. 49, иначе перейти к п. 50.

49. $\Delta_{ij} \Rightarrow QS$. Перейти к п. 52.

50. Найти хвостовой элемент $P \Rightarrow E$ в списке для Δ_{ij}.

51. $P \Rightarrow E.FWD \Rightarrow QS$.

52. Для $\{\psi \mid \psi = \overline{1, \varphi - 1}\}$ выполнить пп. 53-55.

53. $\theta \leftarrow QS \Rightarrow E.AM$.

54. Выполнить процедуру загрузки линии m_θ на участке $(\sigma_\psi, \sigma_{\psi+1})$ с помощью алгоритма 4.12.

55. $QS \Rightarrow QS \Rightarrow E.FWD$. Перейти к п. 52.

56. $\{\tilde{a}_{ij}\} \leftarrow \{\tilde{a}_{ij}\} \setminus \tilde{a}_{ij}$.

57. Если $\{\tilde{a}_{ij}\} \neq \varnothing$, то перейти к п. 1, иначе перейти к п. 58.

58. Определить по формуле (4.15) среднюю задержку t_{av}. Если $t_{av} \le T_{\max}$, то перейти к п. 60, иначе перейти к п. 59.

59. Решить задачу выбора пропускных способностей дуг и определить но-

вые значения w_i^k, $i=\overline{1,\eta_k-1}$, $k=\overline{1,\tilde{l}}$. Восстановить множество $\{\tilde{a}_{ij}\}$ и структуру H_S. Перейти к п. 1.

60. Используя заданные функции стоимости C_{tr}^k, $k=\overline{1,\tilde{l}}$, C_{load}^{β}, $\beta=\overline{1,n}$ рассчитать затраты Z на транспортировку и обработку транспортных блоков. Если $Z \le Z_{opt}$, то $Z_{opt}=Z$, выбрать из S1-S4 очередную стратегию, восстановить множество $\{\tilde{a}_{ij}\}$, структуру H_S и перейти к п. 1. Если все стратегии выбраны, то конец работы алгоритма и Z_{opt} содержит лучший результат.

Рассмотрим работу алгоритма. В цикле 1-57 выполняется выбор потока для распределения. При этом используется одна из стратегий S1-S4. Для стратегий S1-S3 множество $\{\tilde{a}_{ij}\}$ предварительно упорядочивается. Как видно из алгоритма, в шаге 7 в качестве первоначального пути для распределения потока $\tilde{a}_{ij}$ выбирается кратчайший путь. Однако такой путь выбирается только тогда, когда для него выполняются ограничения (4.32). Если эти ограничения нарушаются, то необходимо построить множество $\{\pi_{ij}\}$. Если множество $\{\pi_{ij}\}$ пусто, то поток $\tilde{a}_{ij}$ или его часть распределяются по кратчайшему пути, либо этот поток помечается нераспределенным и осуществляется переход к выбору нового потока. Если только часть потока $\tilde{a}_{ij}$ была распределена, то все равно делается отметка о не распределении этого потока. В противном случае выбирается очередной путь из $\{\pi_{ij}\}$. Описанные действия выполняются в шагах 2-6. В цикле 8-33 производится попытка распределения потока на одном из путей множества $\{\pi_{ij}\}$. Вход в цикл осуществляется только тогда, когда поток $\tilde{a}_{ij}$ не может быть распределен на кратчайшем пути. Для распределения потока по участкам $(\sigma_\psi, \sigma_{\psi+1})$ пути π_{ij} в алгоритме организован цикл 10-47. В цикле 14-26 выполняется просмотр всех линий, на которых может быть распределен поток $\tilde{a}_{\sigma_\psi,\sigma_{\psi+1}}$. В шаге 15 выполняется проверка на возможность распределения потока $\tilde{a}_{\sigma_\psi,\sigma_{\psi+1}}$ на линии m_k. Поток $\tilde{a}_{\sigma_\psi,\sigma_{\psi+1}}$ или его часть не могут быть распределены на линии

m_k в двух случаях: если движение по линии разрешено только в одну сторону ($MV_k = 1$), а для распределения $\tilde{a}_{\sigma_\psi,\sigma_{\psi+1}}$ необходимо двигаться по m_k от $\sigma_{\psi+1}$ к σ_ψ; распределение потоков на этой линии запрещено. После выхода из цикла 14-26 в θ содержится номер линии, выбранной для распределения потока $\tilde{a}_{ij}$ на участке $(\sigma_\psi, \sigma_{\psi+1})$ или значение 0. В случае, если линия не определена ($\theta = 0$ или $c_3 = 0$), выполняется выбор очередного пути π_{ij}. Если множество $\{\pi_{ij}\}$ пусто, выясняется, по какой причине не удалось распределить поток (шаги 34-37) и выполняется одно из трех действий: вводятся новые линии (шаг 39); ослабляются ограничения (4.34) (шаг 41); помечается нераспределенный поток (шаг 40). В шагах 43-45 и 52-55 выполняются процедуры загрузки для распределенных безтранзитных и транзитных потоков соответственно.

Цикл 1-57 заканчивается, если множество $\{\tilde{a}_{ij}\}$ пусто. В этом случае справочная структура H_S полностью сформирована (при условии, что $\{\tilde{a}^*_{ij}\}$ также пусто); вектор Y содержит объемы транзитной обработки; векторы LF^k_ε, LB^k_ε, $\varepsilon = \overline{1, \eta_k - 1}$, $k = \overline{1, \tilde{l}}$ определяют загрузку топологических дуг линий связи, а векторы XF^k_ε, XB^k_ε, $\varepsilon = \overline{1, \eta_k}$, $k = \overline{1, \tilde{l}}$ — объемы обработки транспортных блоков узлах для каждой линии в прямом и обратном направлениях соответственно, где $\tilde{l}$ — число линий связи с учетом вновь введенных в процессе решения задачи. Далее выполняется проверка ограничений (4.15) на среднюю задержку потоков и при необходимости решается задача выбора новых пропускных способностей дуг w^k_i, $i = \overline{1, \eta_k - 1}$, $k = \overline{1, \tilde{l}}$. Для новых значений w^k_i цикл 1-57 повторяется до тех пор, пока не будет выполнено условие (4.15). Если это условие выполняется, то рассчитываются затраты на транспортировку и обработку транспортных блоков для текущей стратегии решения задачи. После выбора всех стратегий решение задачи завершается и определяется лучший вариант решения.

Пользуясь обозначениями, введенными при определении сложности алго-

ритма 4.13, проанализируем общую сложность алгоритма 4.14, которая состоит из сложности процедур автоматической генерации линий связи, формирования справочной H_S, выполнения цикла 1-57 и итеративного решения задачи выбора пропускных способностей дуг. Сначала рассмотрим сложность цикла 1-57 в случае, когда все потоки распределяются по кратчайшим путям без нарушения ограничений (4.32) - (4.34). Внутри цикла 1-57 на одинаковом уровне вложенности имеются два цикла: цикл 10-47 и цикл 52-55. Так как шаги 30, 39 в цикле 10-47 не выполняются, то внутри этого цикла остается только цикл 14-26, сложность которого равна $k_1(k_2+k_2)$. Сложностью процедуры загрузки, выполняемой в шаге 45, в этом случае можно пренебречь, поскольку она мала по сравнению со сложностью цикла 14-26. Сложность цикла 52-55 определяется величиной $2k_2k_3$. Учитывая неравенство $2k_2k_1k_3 > 2k_2k_3$ и то, что цикл 52-55 будет выполняться только для транзитных потоков, легко видеть асимптотическое доминирование цикла 10-47 в общем времени выполнения 1-57. Полагая, что в худшем случае для неразветвленных потоков цикл 1-57 будет выполняться $n(n-1)$ раз, и, записав величину $2k_2k_1k_3$ в виде $2k_2k_1k_3 = n^{\log_{10} 2k_2k_1k_3/\log_{10} n}$, нетрудно видеть, что сложность цикла 1-57 определяется величиной $O(n^{2+\log_{10} 2k_2k_1k_3/\log_{10} n})$. Если при решении задачи было разрешено разветвление потоков и n_g — среднее количество частей в разветвленном потоке, то сложность увеличится до $O(n_g n^{2+\log_{10} 2k_2k_1k_3/\log_{10} n})$. Практические исследования показали, что для реальных сетей величина $\log_{10} 2k_2k_1k_3/\log_{10} n$, как правило, равна $1+\varepsilon$, где $\varepsilon \in [0,0.3]$.

Оценить сложность цикла 1-57 в случае, когда условия (4.32)-(4.34) выполняются не всегда, значительно труднее. Это связано с тем, что, во-первых, неизвестно сколько раз будут выполнены шаги 30 и 39, а, во-вторых, число выполнения цикла 8-33 непредсказуемо. Сложность шага 39 — $O(k_2^2)$ незначительна по сравнению со сложностью шага 30, которая равна $n + n\log_2 n$, где n — сложность формирования $\{\pi_{ij}\}$, $n\log_2 n$ — сложность сортировки $\{\pi_{ij}\}$ одним из лучших методов, основанных на сравнении. Пусть $k_5 = |\{\pi_{ij}\}|$. Тогда в

худшем случае, когда для каждого потока строится множество $\{\pi_{ij}\}$ и просматриваются все пути из $\{\pi_{ij}\}$, грубая оценка сложности цикла 1-57 составит $T_c \approx n(n-1)[n + n\log_2 n + k_5 k_3 k_1 (k_2 + k_2)]$. Для разветвленных потоков $T_c \approx n_g n(n-1)[n + n\log_2 n + k_5 k_3 k_1 (k_2 + k_2)]$. Из алгоритма 4.11 видно, что величина k_5 приближается к n, поэтому следует ожидать, что в асимптотике сложность цикла 1-57 будет никак не лучше, чем $O(n^{3+\log_{10} 2k_2 k_1 k_3 / \log_{10} n})$ или $O(n_g n^{3+\log_{10} 2k_2 k_1 k_3 / \log_{10} n})$. Учитывая полученные ранее оценки сложности формирования справочной H_S $(O((l-lr)\eta_{cp}^2))$ и решения задачи выбора пропускных способностей дуг методом последовательного анализа вариантов $(O[\lambda(e(e+\alpha) + Ke(e+\alpha)) + e\log e + K_1 e])$ определим общую сложность T_o решения задачи распределения и маршрутизации потоков без использования процедуры редукции линий связи:

$$T_o = O(n^3) + O((l-lr)\eta_{cp}^2) +$$

$$+\gamma\,[\,O(n_g n^{4+\varepsilon}) + O[\lambda(e(e+\alpha) + Ke(e+\alpha)) + e\log e + K_1 e]\,].$$

Где $O(n^3)$ — сложность процедуры генерации линий связи, n — количество узлов в сети, l — количество заданных линий связи, η_{cp} — среднее количество узлов в линии связи, lr — количество редуцированных линий связи (lr=0), γ — количество итераций выполнения цикла 1-57 и решения задачи ВПС, n_g — среднее количество частей в разветвленном потоке, e — количество дуг в сети, α — количество различных дискретных пропускных способностей дуг сети, λ, K, K_1 — некоторые константы, $\varepsilon \in [0,0.3]$.

Полученная оценка сложности определена без учета сложности процедуры редукции и операций с редуцированными линиями связи в цикле 1-57. Рассмотрим изменения в алгоритме 4.14, связанные с использованием редукции линий связи. Прежде всего, ясно, что построение цепочек редукции скажется на организации цикла 14-26. Для просмотра всех линий, на которых может быть распределен поток $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$, необходимо внутри цикла 14-26 организовать допол-

нительный цикл. С учетом изменений фрагмент алгоритма 4.14 запишется следующим образом:

... ...

13. $P \Rightarrow \Delta_{\sigma_\psi, \sigma_{\psi+1}}$, $c_2 \leftarrow 0$, $c_3 \leftarrow 0$, $\theta \leftarrow 0$.

14. $k \leftarrow P \Rightarrow E.AM$.

14.1. $k \leftarrow RV_k$.

14.2. Пока $k \neq 0$ выполнить пп. 15-24.1 .

15. Если $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$ может быть распределен на m_k, то перейти к п. 16, иначе перейти к п. 24.1.

... ...

19. Если условие (4.34) выполняется, то перейти к п. 20, иначе перейти к п. 24.1.

... ...

21. Если условие (4.33) выполняется, то перейти к п. 22, иначе перейти к п. 24.1.

24. ... ...

24.1. $k \leftarrow VR_k$, перейти к п. 14.2.

25. $P \Rightarrow P \Rightarrow E.FWD$.

26. Если $P \neq NULL \wedge P \Rightarrow E.DM_5 = 1$, то перейти к п. 14, иначе перейти к п. 27.

Ранее указывалось на две причины, по которым поток $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$ в шаге 15 не может быть распределен на линию m_k. В данном случае появляется третья причина, связанная с тем, что для $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$ и m_k могут не выполняться условия:

$$(\xi_\mu^k = \sigma_\psi \wedge \xi_{\mu+c}^k = \sigma_{\psi+1}) \vee (\xi_\mu^k = \sigma_{\psi+1} \wedge \xi_{\mu+c}^k = \sigma_\psi).$$

Изменению подвергнется также шаг 44, в котором для безтранзитного потока помечается выбранная линия. В этом случае, если номер θ выбранной линии не совпадает с номером линии $P \Rightarrow E.AM$, то генерируется новый элемент E, который вставляется за элементом $P1 \Rightarrow E$. Модифицированный шаг запи-

шется так:

44. Если $\theta = P1 \Rightarrow E.AM$, то пометить поле DM_1 или DM_2 элемента $P1 \Rightarrow E$, в поля AF и NF занести соответствующие значения. В противном случае: Образовать новый $E(P)$, $P \Rightarrow E.FWD \Rightarrow P1 \Rightarrow E.FWD$, $P1 \Rightarrow E.FWD \Rightarrow P$, $P \Rightarrow E.AM \leftarrow \theta$, $P \Rightarrow E.DM_5 \leftarrow 1$, для $P \Rightarrow E$ сформировать поля DM_3 и DM_4, пометить DM_1 или DM_2 и в поля AF и NF занести соответствующие значения.

Поскольку для вновь образованного элемента в поле DM_5 заносится единица, в дальнейшем, в цикле 14-26 этот элемент будет просматриваться. Однако, линия, на которую указывает дополнительно включенный элемент, просматриваться не будет, так как для нее $RV_k = 0$.

Рассмотрим необходимые изменения в процедуре введения новых линий. В шаге 3 алгоритма 4.13 определяется прообраз новой линии. Для этого должны просматриваться все линии, на которых может быть распределен поток $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$. Последовательность действий, выполняемых в шаге 3, не будет отличаться от шагов 13-15, 24.1-26 измененного фрагмента алгоритма 4.14. Предположим, что в шаге 3 установлен указатель $P1$, ссылающийся к элементу E, определяющему линию m_ξ, в которую редуцирован выбранный прообраз m_k и покажем далее, какие изменения необходимо произвести в последующих шагах алгоритма 4.13.

В шаге 9 вместо корректировки справочной H_S необходимо скорректировать векторы VR и RV. При корректировке элемент VR_l для новой линии m_l включается в цепочку редуцированных линий в линию m_ξ в соответствии с длиной $\eta_l = c + 1$ новой линии. Шаг 9 заменяется на последовательность следующих шагов:

... ...

9.1. $k \leftarrow RV_\xi$, $k_1 \leftarrow RV_\xi$, $RV_l \leftarrow 0$, $c_1 \leftarrow 0$.

9.2. Пока $k \neq 0$ выполнить пп. 9.3-9.6.

9.3. Если $\eta_k \geq \eta_l$, то перейти к п. 9.4, иначе перейти к п. 9.6.

9.4. $VR_l \leftarrow k$.

9.5. Если $c_1 \neq 0$, то $RV_\xi \leftarrow l$, иначе $VR_{k_1} \leftarrow l$. Перейти к п. 11.

9.6. $k_1 \leftarrow k$, $c_1 \leftarrow c_1 + 1$, $k \leftarrow VR_k$, перейти к п. 9.2.

Шаг 10 алгоритма 4.13 исключается, поскольку корректировка справочной не производится, а шаг 16 изменяется точно так же, как и шаг 44 алгоритма 4.14. Для включения нового элемента E в список для безтранзитного потока $\tilde{a}_{\sigma_\psi, \sigma_{\psi+1}}$ используется установленный в шаге 3 указатель $P1$.

Необходимо отметить, что изменение алгоритма 4.14 с учетом редукции линий связи несколько увеличивает его сложность, так как добавляется сложность процедуры редукции ($O(l^2(n+\eta_{cp})+l\eta_{cp})$ или $O(l^2\eta_{cp})$) и возрастает число просматриваемых линий в шагах 14-26 (возрастает величина k_1 в выражении для сложности алгоритма) и

$$T_o = O(n^3) + O(l^2(n+\eta_{cp}) + l\eta_{cp}) \text{ (или } O(l^2\eta_{cp}) + O((l-lr)\eta_{cp}^2) +$$

$$+ \gamma\,[O(n_g n^{4+\varepsilon}) + O[\lambda(e(e+\alpha) + Ke(e+\alpha)) + e\log e + K_1 e]].$$

Несмотря на это, редукция оказывается крайне необходимой, если в качестве исходного множества выступают автоматически сгенерированные шаблоны линий связи (см. подраздел 4.6). В этом случае резко возрастает размерность задачи, принимающей характер построения новых, ранее не существующих линий связи.

4.6. Численный пример решения задачи распределения и маршрутизации потоков

Для эксперимента выбрана сгенерированная датчиком псевдослучайных чисел однородная транспортная сеть с числом узлов $n = 30$ и степенью узлов $val = 5$. Маршруты транспортных средств совпадали с дугами сети G, и предполагалось, что движение по маршрутам разрешено в обе стороны. Поэтому заданное число маршрутов на сети равно 75, а ограничения (4.11) не задавались.

Длины дуг генерировались в пределах от 80 до 300 км. Принимались следующие значения параметров: объем транспортного блока ω = 40 единицам тарно-штучных грузов; одинаковая грузоподъемность транспортных средств на всех дугах маршрутов варьировалась в пределах $w^k \in \{20, 10, 5\}$ транспортных блоков; время на сортировку мелкопартионных потоков в узлах сети 24 ч; время на транзитную перегрузку транспортных блоков в узлах сети 12 ч; время стоянки транспортных средств в конечных пунктах следования 22 ч; периодичность движения транспортных средств 24 ч; средняя скорость движения транспортных средств 80 км/час; пропускные способности узлов по обработке транзита b_i, $i = \overline{1,n}$ не ограничивались; максимальное время доставки мелкопартионных потоков конечному потребителю не ограничивалось, т.е. допускалась возможность максимальной загрузки транспортных средств при распределении потоков по «длинным» путям; средняя задержка транспортных блоков в сети t_{av} не ограничивалась; разветвление потоков транспортных блоков было запрещено.

Объемы мелкопартионных исходящих потоков из узлов сети задавались датчиком равномерно распределенных псевдослучайных чисел в пределах от 1 до 9 единиц. Для расчета среднегодовых приведенных затрат на транспортировку и обработку потоков использовались следующие тождественные формулы в условных единицах стоимости:

$$\sum_{k=1}^{l} C_{tr}^{k}\left(\left(\sum_{\eta\xi\in q_k}\sum_{ij\in S} u_{ij,k}^{\eta\xi}\right), d_k\right) \equiv \sum_{k=1}^{l} D_{ts}^{k} w^k (258{,}45 + 2{,}92 d_k) + 108{,}036 D_{mb},$$

где D_{ts}^{k} — рабочий парк транспортных средств на маршруте m_k; D_{mb} — рабочий парк транспортных блоков в сети перевозок;

$$C_{sort}^{i}(a_i, q_i) \equiv 16{,}8892975 a_i \exp\left(-\frac{17{,}74 a_i}{1+q_i}\right), \quad i = \overline{1,n};$$

$$C_{load}^{i}(u_i) \equiv \sqrt{25595 u_i^2 + 2629 u_i}, \quad i = \overline{1,n},$$

где $C_{sort}^{i}(\cdot)$, a_i, q_i и u_i — соответственно затраты на сортировку мелкопартионных потоков, суммарный мелкопартионный поток, число направлений сортировки мелкопартионных потоков на другие узлы сети и суммарный поток

транспортных блоков в узле i.

Результаты решения задачи для двух вариантов, когда не использовалась (вариант А) и использовалась (вариант В) упаковка мелкопартионных грузов в транспортные блоки, при изменении грузоподъемности транспортных средств, приведены в табл. 4.7 и на рис. 4.16, 4.17. Для варианта В предварительно была решена задача оптимизации упаковки транспортных блоков. В таблице $t_{\min}$, $t_{\max}$, t_{mean} — соответственно минимальное, максимальное и среднее время доставки грузов получателю.

Таблица 4.7

Наименования показателей	Грузоподъемность транспортных средств в транспортных блоках					
	w = 20		w = 10		w = 5	
	Результаты решения задачи: А — без упаковки, В — с упаковкой					
	А	В	А	В	А	В
Средний коэффициент загрузки транспортного блока в сети	0,125	0,746	0,125	0,746	0,125	0,746
Среднее число направлений сортировки в узле сети	29	8	29	8	29	8
Заданное число маршрутов	75	75	75	75	75	75
Число введенных новых маршрутов	0	0	71	0	197	4
Общее число загруженных маршрутов	75	72	146	72	272	78
Средний коэффициент загрузки транспортного средства в сети	0,683	0,177	0,873	0,357	0,935	0,733
Исходный рабочий парк транспортных средств	150	150	150	150	150	150
Требуемый рабочий парк транспортных средств	150	144	292	144	544	156
Требуемый рабочий парк транспортных блоков	2968	841	3184	842	3202	873
Максимальная пропускная способность узла	61	22	95	22	90	28
Транспортные затраты (млн. у. е.)	2,602721	2,250343	2,459906	1,170709	2,319991	0,680120
Затраты на сортировку мелкопартионных потоков (млн. у. е.)	0,264038	0,314535	0,264038	0,314535	0,264038	0,314535
Затраты на погрузочно-разгрузочные работы (млн. у. е.)	0,460201	0,112875	0,517636	0,113195	0,529155	0,124074
Общие затраты (млн. у. е.)	3,326960	2,677753	3,241580	1,598439	3,113184	1,118729
$t_{\min}$	2,04	2,04	2,04	2,04	2,04	2,04
$t_{\max}$	3,97	8,07	5,46	8,24	5,57	16,69
t_{mean}	2,84	3,73	3,08	3,74	3,13	4,05

Из приведенных результатов видно, что после упаковки мелкопартионных потоков значительно увеличивается коэффициент загрузки транспортного блока (с 0,125 до 0,746), уменьшается среднее число направлений сортировки (с 29 до 8) и ощутимо сокращается рабочий парк транспортных блоков (что влияет на пропускные способности узлов сети).

При понижении грузоподъемности транспортных средств сильно заметно снижение общих затрат для варианта решения с упаковкой потоков. Однако в варианте B для отдельных потоков значительно увеличивается максимальное время доставки. Среднее время доставки остается достаточно стабильным (рис. 4.17).

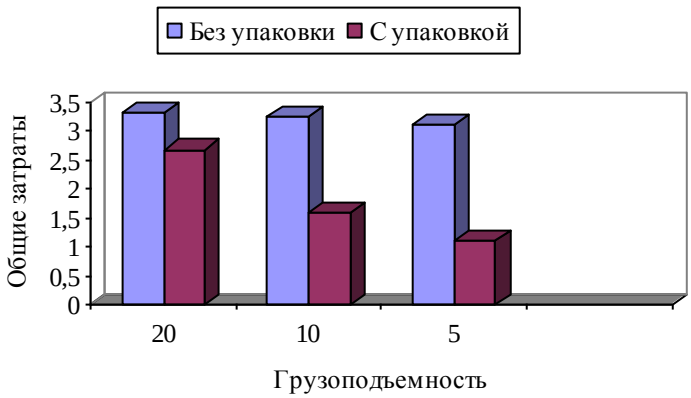


Рис. 4.16

Отметим, что если при решении задачи не было бы разрешено вводить новые маршруты, то в варианте A оказалось бы 43,4 % и 70,34 % нераспределенных потоков для $w = 10$ и $w = 5$, а в варианте B — 1,26 % для $w = 5$.

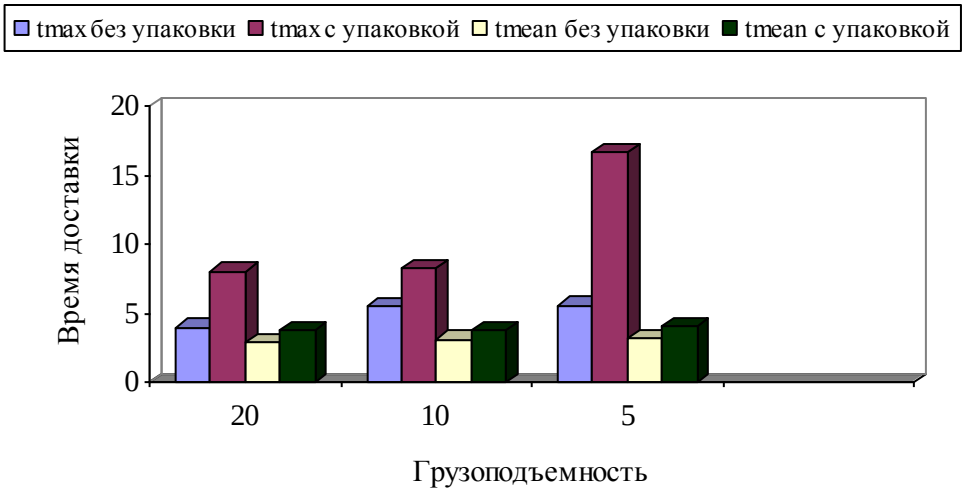


Рис. 4.17

Время расчета задачи в вариантах A и B не превышало долей секунды. Программы составлены на языке *Digital Visual Fortran* и выполнялись под управлением операционной системы *Windows Vista* на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.

Выводы по четвертому разделу

1. Решение нелинейной многоэкстремальной дискретной задачи распределения и маршрутизации потоков транспортных блоков с упакованными в них мелкопартионными грузами или сообщениями с различными адресами назначения может быть сведено к последовательному решению некоторой совокупности линейных многомерных задач о ранце со связывающими ограничениями. При этом исходная задача на минимум целевой функции заменяется на максимум загрузки дуг (участков) маршрутов транспортных средств или каналов связи при всех заданных ограничениях. Алгоритм решения преобразованной задачи должен обеспечивать возможность разветвления потоков транспортных блоков при учете ограничений на время доставки мелкопартионных потоков получателю и среднее время задержки потоков транспортных блоков в сети. Кроме того, решение задачи должно осуществляться в интерактивном режиме и допускать возможность параметрического задания правых частей всех ограничений и их ослабления в процессе решения при невозможности распределения потоков. Для NP-трудной ранцевой задачи в лучшем случае могут быть разработаны эвристические методы и алгоритмы, дающие ее субоптимальное (рациональное, с точки зрения проектировщика сети) решение для заданных ограничений при выбранных вариантах и режимах решения.

2. При формулировке задачи распределения и маршрутизации потоков с заданными тарифами на дугах и в узлах она может быть за полиномиальное время $O[(Q+1)n^2 - Qn]$ преобразована к задаче целочисленного линейного программирования, где n — число узлов в сети. В этом случае при учете ограничений на время доставки мелкопартионных корреспонденций получателю существующие методы и пакеты программ не могут быть непосредственно применены — требуется проведение дополнительных исследований по возможной их модификации и доработке с учетом новых ограничений. Если ограничения на время доставки не являются критическими, то задачу с заданными тарифами на дугах и в узлах в виде задачи ЦЛП можно решать с применением известных методов и пакетов программ, однако для установления границ их

эффективности и анализа получаемых решений необходимо проведение обширного вычислительного эксперимента на общедоступных серверах, таких как NEOS, GAMS и др. с программами Gurobi, Linear Program Solver, Simplex OPTIMA, CPLEX, MINTO и пр.

3. Разработана улучшенная версия алгоритма [27] построения всех КП в сети по критерию: минимум дуг в пути; минимум длины пути. Доказано, что время работы нового алгоритма [28] может быть сокращено за счет применения АТД и уменьшения просмотра узлов во внутренних циклах и растет пропорционально функции $Sn^3 - n^2(3S + c) + 2n(S + c)$, $S = \lceil \log_2 m \rceil$, где m — максимальный ранг пути среди всех кратчайших путей, построенных в сети, а c — некоторая константа. Эмпирически показано, что улучшенный алгоритм на разреженных сетях работает быстрее алгоритма [27] и при увеличении плотности сети на несколько порядков превышает по скорости работы модифицированный алгоритм Флойда. Проведенный эксперимент показал высокую вычислительную эффективность предложенного алгоритма, который может быть включен в состав типового инструментария разработчика и с успехом применяться при решении практических задач нахождения двухкритериальных КП на графах и сетях большой размерности.

4. Предложена эффективная реализация алгоритма Краскала для нахождения остовного дерева минимального (максимального) веса для связного неориентированного графа с практической временной сложностью $O(e)$, где e — число ребер графа. Алгоритм может быть легко приспособлен для ориентированных графов.

5. Показано, что время работы предложенного алгоритма на связных разреженных графах сравнимо со временем сортировки ребер по методу преобразования веса ребра в адрес оперативной памяти компьютера, а время построения минимального остовного дерева практически стремится к $O(1)$.

6. На основании проведенных экспериментальных исследований получена оценка сравнимости быстродействия предложенного алгоритма с алгоритмом Прима, из которой следует, что для связных графов предложенный алгоритм

лучше алгоритма Прима при $e \le 0{,}27v^2$, где v — число вершин графа.

7. Экспериментальное сравнение быстродействия алгоритма Краскала со списочными и древовидными структурами данных показало, что алгоритмы со списочными структурами на практике работают быстрее, чем алгоритмы с древовидными структурами как на связных, так и на несвязных графах.

8. На основе предложенных структур данных и эвристических подходов разработаны полиномиальные по сложности алгоритмы решения задачи распределения и маршрутизации потоков транспортных блоков. Получены оценки временной сложности алгоритмов. Алгоритмы позволяют получить приближенное решение задачи за время

$$T_o = O(n^3) + O(l^2(n+\eta_{cp}) + l\eta_{cp}) \text{ (или } O(l^2\eta_{cp}) + O((l-lr)\eta_{cp}^2) +$$

$$+ \gamma\,[O(n_g n^{4+\varepsilon}) + O[\lambda(e(e+\alpha) + Ke(e+\alpha)) + e\log e + K_1 e]],$$

где n — количество узлов в сети, l — количество заданных линий связи, η_{cp} — среднее количество узлов в линии связи, lr — количество редуцированных линий связи, γ — количество итераций решения задачи выбора пропускных способностей дуг, n_g — среднее количество частей в разветвленном потоке, e — количество дуг в сети, α — количество различных дискретных пропускных способностей дуг сети, λ, K, K_1 — некоторые константы, $\varepsilon \in [0,0.3]$.

РАЗДЕЛ 5. ЗАДАЧИ ТЕКУЩЕГО ПЛАНИРОВАНИЯ КАК МЕТОДОЛОГИЧЕСКАЯ ОСНОВА — ЯДРО ДЛЯ ПЕРСПЕКТИВНОГО РАЗВИТИЯ УЗЛОВ И ТРАНСПОРТНЫХ МАРШРУТОВ И ОПЕРАТИВНОГО УПРАВЛЕНИЯ РАСПРЕДЕЛЕНИЕМ ПОТОКОВ В МАГИСТРАЛЬНОЙ СЕТИ

Важнейшими задачами, входящими в математическое обеспечение информационно-аналитической системы (ИАС) управления процессами обработки и распределения мелкопартионных потоков в многопродуктовой иерархической сети, являются прогнозирование и оптимизация поэтапного развития ее инфраструктуры (раздел 1). При этом необходимо учитывать все возможности качественного организационно-технического совершенствования сети при наличии ограничений на капитальные вложения (инвестиции) и возможности их освоения. По существу, решение этих задач определяет планы ввода в действие производственных мощностей, основных фондов элементов сети (узловых предприятий и линий связи транспортной сети, узлов и каналов связи сети передачи данных) и трудовых ресурсов по этапам развития. Как правило, перспективные планы развития строятся на несколько лет (от трех до десяти), а в качестве этапа планирования (дискреты времени планирования) выступает один год. Результаты решения таких крупноагрегированных задач — распределение капитальных вложений и других материальных ресурсов по этапам внедрения, должны (или могут) быть использованы при построении более детализированных моделей на каждом этапе. Так, например, для транспортных сетей, необходимо разделить всю сумму капитальных вложений этапа на приобретение, модернизацию и ввод в действие сортировочного оборудования (дополнительных линий сортировки), оборудования для погрузки-выгрузки грузов в узловых предприятиях, транспортных средств для существующих и новых маршрутов перевозки. Для сетей передачи данных следует разделить финансовые ресурсы этапа на модернизацию и переоборудование между узлами (мультиплексорами, коммутаторами, маршрутизаторами) и каналами связи.

В каждом конкретном случае выбор длительности этапа перспективного прогнозирования является самостоятельной задачей и зависит от скорости внедрения и освоения новой техники и информационных технологий в сетевых инфраструктурах различных отраслей в хозяйственной сфере. Устойчивость функционирования сетевых структур на протяжении этапов планирования при колебаниях нагрузок в узлах и линиях связи сети, возникновении отказов и воздействии случайных факторов должна обеспечиваться задачами оперативного управления.

В настоящем разделе предлагается методика получения исходных данных для построения динамических моделей поэтапного развития узлов и транспортных маршрутов магистральной сети, в основу которой заложено решение задач оптимизации ее структуры и распределения потоков [127, 128]. Показано также, каким образом можно использовать эти задачи для оперативного перераспределения потоков в случае превышения пропускных способностей узлов (т.е. когда в узлах образуются очереди) и отказов оборудования в узлах и на линиях связи.

5.1. Введение в постановку задачи и основные постулаты

Рассмотрим основные постулаты построения математической модели перспективного развития многопродуктовой коммуникационной сети с дискретными потоками.

1. При прогнозировании исходных данных и параметров сети (исходящих потоков, стоимостных характеристик процессов обработки и транспортировки потоков и др.) для поэтапного ее развития используются современные модели и методы математического моделирования на основе временных рядов данных. К ним относятся метод группового учета аргументов (МГУА), методы авторегрессии, нечеткие нейросети, байесовские модели и сети, позволяющие получить высококачественные прогнозы для линейных, нелинейных, стационарных и нестационарных процессов [50, 51]. Большинство перечисленных моделей и методов реализованы в многочисленных пакетах программ обработки стати-

стических и экспериментальных данных и поддержки принятия решений в технических, социально-экономических, финансовых и других сферах, таких как NeuroShell2, ModelQest, ASPN, KnowledgeMiner, MatLab и др.

2. На каждом этапе развития реализуются все многопродуктовые потоки в сети, т.е. все потоки доставляются от отправителей (источников) до получателей (стоков). Перед отправкой получателю мелкопартионные дискретные потоки упаковываются в транспортные блоки, причем в один транспортный блок могут попасть потоки с разными адресами назначения. В случае возникновения перегрузок в сети и невозможности стабилизации ее функционирования за счет оперативного перераспределения потоков, увеличение производительности узлов и ввод новых линий связи осуществляется за счет внутренних резервов и запасов сетевой инфраструктуры.

3. Структура модели развития сети должна обеспечивать возможность ее декомпозиции на более простые подмодели, с помощью которых можно снизить общую размерность задачи и на каждом этапе развития выполнять поочередную оптимизацию по отдельным группам переменных исходной модели. Здесь подразумевается, что при проектировании развития сети на каждом этапе имеется возможность обоснованного разделения общей задачи на задачу оптимизации узловой структуры сети и задачу оптимизации маршрутов транспортировки потоков. В разделе 3 доказано, что выбор структуры многопродуктовой сети с мелкопартионными потоками не зависит от пропускных способностей ее дуг. Доказательство подтверждено результатами экспериментальных расчетов. Показано, что исходная модель может быть заменена на более простую, в которой не учитываются ограничения на пропускные способности дуг сети и на среднее время задержки потоков. Использование более простой модели позволяет при решении задачи выбора структуры сети отказаться от решения трех других задач: балансировки матрицы межузловых потоков, выбора пропускных способностей дуг и маршрутизации потоков, которые могут быть решены независимо после процесса оптимизации узловой структуры сети. Для задачи перспективного развития это означает, что распределение финансовых и трудо-

вых ресурсов на каждом этапе можно выполнять отдельно для узлов сети и для линий связи, решая соответственно задачи оптимизации структуры сети и маршрутизации потоков.

В случае оптимизации структуры сети передачи данных должны учитываться капитальные затраты на приобретение нового и модернизацию существующего коммутационного оборудования (аппаратных шлюзов, коммутаторов, мультиплексоров, маршрутизаторов, концентраторов, мостов, повторителей, сетевых адаптеров и пр.), а также эксплуатационные расходы, связанные с текущим обслуживанием и ремонтом этого оборудования. При оптимизации структуры транспортной сети необходимо учитывать капитальные затраты на приобретение нового и модернизацию существующего оборудования, устанавливаемого в узлах сети для непрерывной автоматизированной сортировки и накопления грузов по адресам назначения, оборудования для погрузки-выгрузки грузов и эксплуатационные расходы, связанные с обслуживанием и ремонтом.

При маршрутизации потоков в сети передачи данных должны учитываться капитальные затраты на приобретение новых и модернизацию существующих каналов связи и эксплуатационные расходы, связанные с обслуживанием каналов или арендной платой. Для транспортных потоков капитальные затраты зависят от рабочего парка транспортных средств (с различной грузоподъемностью) и контейнеров, необходимых для обеспечения нормального функционирования сети перевозок, а также от рабочего парка автопогрузчиков и, возможно, для крупных узлов — количества автоматизированных контейнерных терминалов, осуществляющих сортировку и группирование исходящих, входящих и транзитных контейнеров по маршрутам следования транспортных средств. Эксплуатационные затраты состоят из нескольких компонентов и включают расходы, связанные с движением (механической работой по транспортировке грузов); объемами обрабатываемых и перевозимых грузов; расстоянием перевозок; временем доставки грузов получателю; обслуживанием и ремонтом контейнерных терминалов, рабочего парка транспортных средств, контейнеров и автопогрузчиков.

В любом случае необходимо учитывать эксплуатационные затраты в узлах сети и на линиях связи на содержание штата административно-управленческого, инженерно-технического и рабочего персонала; внедрение и освоение новых информационных технологий и средств автоматизации и механизации производства; различные организационные мероприятия и др.

4. В процессе развития сети на отдельных этапах допускается переход ее некоторых элементов (узлов и линий связи) на более низкие уровни состояния, т.е. возможно понижение уровня технической оснащенности элементов сети. При этом если сеть выдерживает среднетекущие нагрузки и имеет место экономия от приведенных затрат (относительно эксплуатационных затрат), получаемая за счет оптимизации процессов обработки и транспортировки потоков, то нет необходимости на данном этапе привлекать дополнительные вложения материальных ресурсов в развитие сети. Однако реализация выделенных на данном этапе ресурсов для проведения реконструктивных мероприятий позволяет дополнительно снизить эксплуатационные затраты на обработку и транспортировку потоков. Так, например, увеличение производительности узлов и грузоподъемности транспортных средств приводит к уменьшению удельной стоимости обработки и перевозки единицы потока. Поэтому при наличии нескольких последовательных этапов, на которых сеть выдерживает прогнозируемую нагрузку, необходимо решать дополнительную задачу, что является более выгодным: пропустить вложение материальных ресурсов на этих этапах и сделать это позднее на последующих этапах при увеличении нагрузок и прогнозируемой девальвации ресурсов; использовать доступные ресурсы этапа и обеспечить снижение эксплуатационных затрат на последующих этапах.

5. Капитальные вложения и другие ресурсные ограничения в стоимостной форме должны входить как в целевую функцию, так и в систему ограничений. При этом критерием оптимальности может быть минимум приведенных затрат на развитие сети или максимум экономической эффективности (прибыли) на конец периода планирования. Такие условия необходимы и достаточны (при адекватности используемых функций затрат) для соизмеримости и сравнения

результатов, полученных при численном моделировании задачи перспективного развития и фактических данных о функционировании физической сети на каждом этапе ее развития.

6. Для задач перспективного развития должны быть построены дисконтированные по времени (т.е. с учетом коэффициента приведения разновременных затрат — нормативного коэффициента эффективности капитальных вложений) детерминированные и стохастические модели. В условиях нестабильной экономики Украины в переходной период, обусловленных прежде всего технологической отсталостью, колебанием цен на энергоносители и другие материальные ресурсы, непрерывным изменением налогового законодательства, недостоверными статистическими данными и прочими случайными факторами, особо важное внимание следует уделить стохастическим динамическим моделям развития сети, в которых в явном виде присутствуют одна или несколько случайных переменных, негативно влияющих на протекание процесса развития. Стохастические модели позволяют получить вероятностные распределения капитальных затрат и других ресурсов по этапам развития и оценить степень риска их вложения.

5.2. Методика получения исходных данных для построения динамических моделей поэтапного развития узлов и транспортных маршрутов магистральной сети. Схема взаимодействия задач текущего планирования и оперативного управления

В разделе 3 показано, что при выборе структуры многопродуктовой сети с мелкопартионными потоками можно пренебречь пропускными способностями ее дуг. Приведен алгоритм выбора оптимальной структуры сети, на каждой итерации которого решается задача упаковки. В процессе решения задачи определяется не только структура сети, но и оптимальная схема сортировки исходящих потоков корреспонденций в выбранной структуре. Если при этом используются функции приведенных затрат, адекватные процессам сортировки, то для узлов сети можно получить достаточно реальную оценку их величины.

Поэтому для расчета затрат на функционирование узлов сети, осуществляющих сортировку потоков для каждого периода планирования можно использовать прогнозируемые потоки и математическую модель задачи упаковки (3.18)-(3.21) с дополнительными ограничениями на время доставки $t_{ij} \le T_{ij} \quad \forall ij \in S$ и число транзитных объединений $\nu_{ij} \le \nu_{\max} \quad \forall ij \in S$ мелкопартионных корреспонденций при их транспортировке из узлов отправления в узлы назначения, где T_{ij} и $\nu_{\max}$ — соответственно заданное время доставки мелкопартионных корреспонденций получателю и максимально допустимое число транзитных объединений корреспонденций.

После решения задачи выбора структуры сети необходимо решить задачу распределения и маршрутизации сформированных потоков транспортных блоков (4.16), (4.10)-(4.15) с дополнительными ограничениями $t_{ij} \le T_{ij} \quad \forall ij \in S$. На вход этой задачи поступает исходная $A = \| a_{ij} \|_{n \times n}$ и преобразованная $A' = \| a'_{ij} \|_{n \times n}$ целочисленные матрицы мелкопартионных потоков корреспонденций и матрица потоков транспортных блоков $\tilde{A} = \| \tilde{a}_{ij} \|_{n \times n}$, $\tilde{a}_{ij} = \left\lceil \frac{a'_{ij}}{\omega} \right\rceil$, где ω — объем транспортного блока. Известна также матрица предварительных оценок времени доставки мелкопартионных корреспонденций получателям $T = \| T_{ij} \|_{n \times n}$ и другая информация, полученная после решения задачи выбора структуры сети. Элементы матрицы T выступают в качестве начальных ограничений на время доставки при решении задачи распределения и маршрутизации потоков транспортных блоков.

Приведем методику получения исходных данных для построения динамических моделей поэтапного развития узлов сети и транспортных маршрутов, которая включает следующие шаги.

Шаг 1. Для заданного числа этапов развития сети определить прогнозируемые показатели: величину магистральных межузловых потоков; стоимостные характеристики процессов обработки и транспортировки (передачи) пото-

ков; значения других параметров, влияющих на эксплуатационные затраты в сети.

Шаг 2. Для каждого этапа развития сети по прогнозируемым данным решить задачу оптимизации ее структуры, получить схему сортировки исходящих потоков и распределение по узлам затрат на сортировку потоков и затрат на погрузку-выгрузку исходящих и входящих транспортных блоков. Одновременно определить и необходимые приведенные затраты на функционирование узлов на период завершения планирования (вторая составляющая функции (3.18) и частично третья — без учета затрат на транзитную перегрузку транспортных блоков). При решении задачи выбора структуры сети не следует учитывать ограничения на пропускные способности узлов (3.20). Дополнительные ограничения $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$, $\forall ij \in S$ могут также не учитываться, однако по требованию проектировщика транспортной сети или администратора сети передачи данных все ограничения могут быть заданы директивно. Если задача выбора структуры решалась для транспортной сети, то дополнительно решить задачу развозки порожних контейнеров (см. Приложение В).

Шаг 3. По результатам решения задачи оптимизации структуры сети для каждого этапа ее развития решить задачу распределения и маршрутизации потоков (4.16), (4.10)-(4.15) без учета ограничений (4.11) и (4.13) и получить окончательные технико-экономические показатели функционирования сети для всех этапов. Результаты решения содержат для каждого узла сети схему сортировки исходящих потоков с оценками времени доставки корреспонденций получателям, затраты на сортировку потоков, затраты на погрузку-выгрузку исходящих, входящих и транзитных транспортных блоков, а также затраты на транспортировку (передачу) потоков транспортных блоков на оптимизированных маршрутах и полную информацию о путях их транспортировки.

На рис. 5.1 схематически показана взаимосвязь задач текущего планирования и перспективного развития.

Задачи оперативного управления тесно связаны с исследованием решения задач текущего планирования на устойчивость и проведением постоптимально-

го анализа и реоптимизации этого решения при возмущении входных данных, а также с параметрическим решением задач текущего планирования.

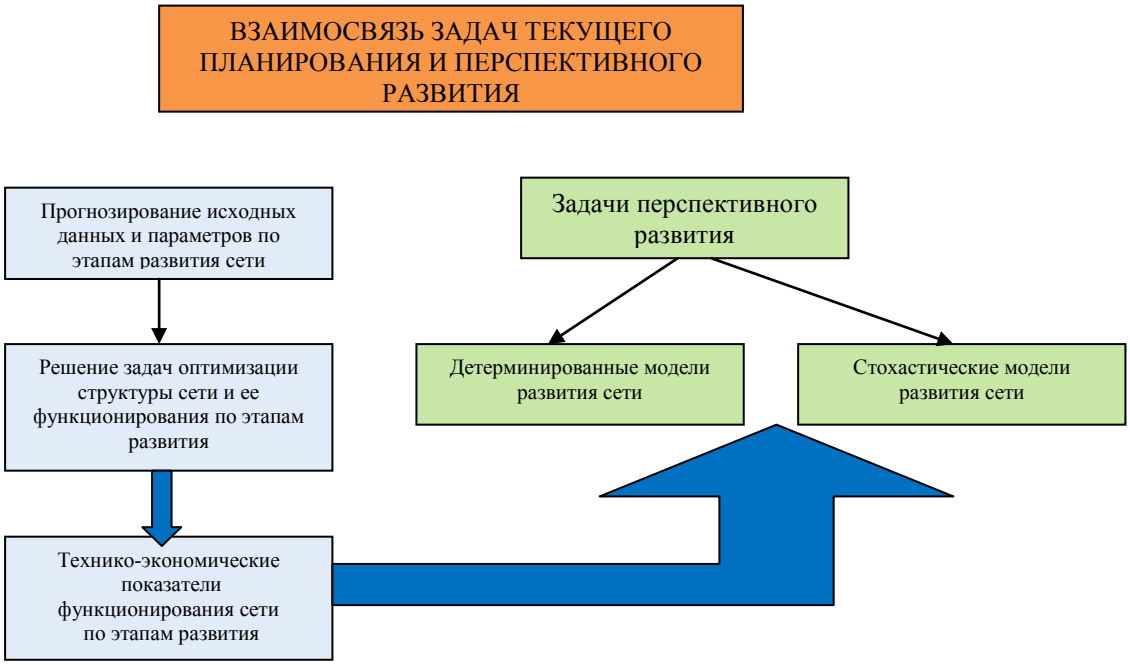


Рис.5.1

В реальных сетях исходные данные и параметры изменяются во времени, а многим факторам, влияющим на процессы обработки и транспортировки потоков, присуща неопределенность. Поэтому возникает необходимость многократного решения задач оптимизации в динамически изменяющихся условиях. Вопросы устойчивости, постоптимального анализа, реоптимизации и параметрического решения задач дискретной оптимизации рассматривались многими учеными [52, 70, 76, 79, 80, 92, 109, 110, 297, 367, 435]. Было показано, что многие из этих задач являются NP-трудными и не менее трудны, чем исходные задачи. В связи со сложностью исследования дискретных задач на устойчивость решения, нас будет интересовать вопрос определения допустимых границ изменения значения целевой функции при малых возмущениях потоков мелкопартионных корреспонденций в узлах сети. Для определения более широких границ изменения значений целевой функции и других контролируемых параметров, полученных в результате решения оптимизационных задач текущего планирования, целесообразно повторное параметрическое решение этих задач.

На рис. 5.2 приведена общая схема взаимодействия задач текущего плани-

рования и оперативного управления на магистральном уровне сети. Если сеть функционирует в штатном режиме, то по заданным границам изменения значений контролируемых параметров осуществляется наблюдение за процессами обработки и распределения потоков в сети. К основным контролируемым параметрам можно отнести: суммарные потоки мелкопартионных корреспонденций и транспортных блоков в узлах сети, объемы необработанных потоков, средние коэффициенты загрузки транспортного блока и линий связи, среднее время доставки корреспонденций получателю. По желанию проектировщика могут быть определены и другие критические параметры состояния сети.

Шкала допустимых значений критических параметров разбита на три зоны — зеленую, желтую и красную. Если все значения параметров находятся в зеленой зоне, то принимается, что сеть функционирует нормально.

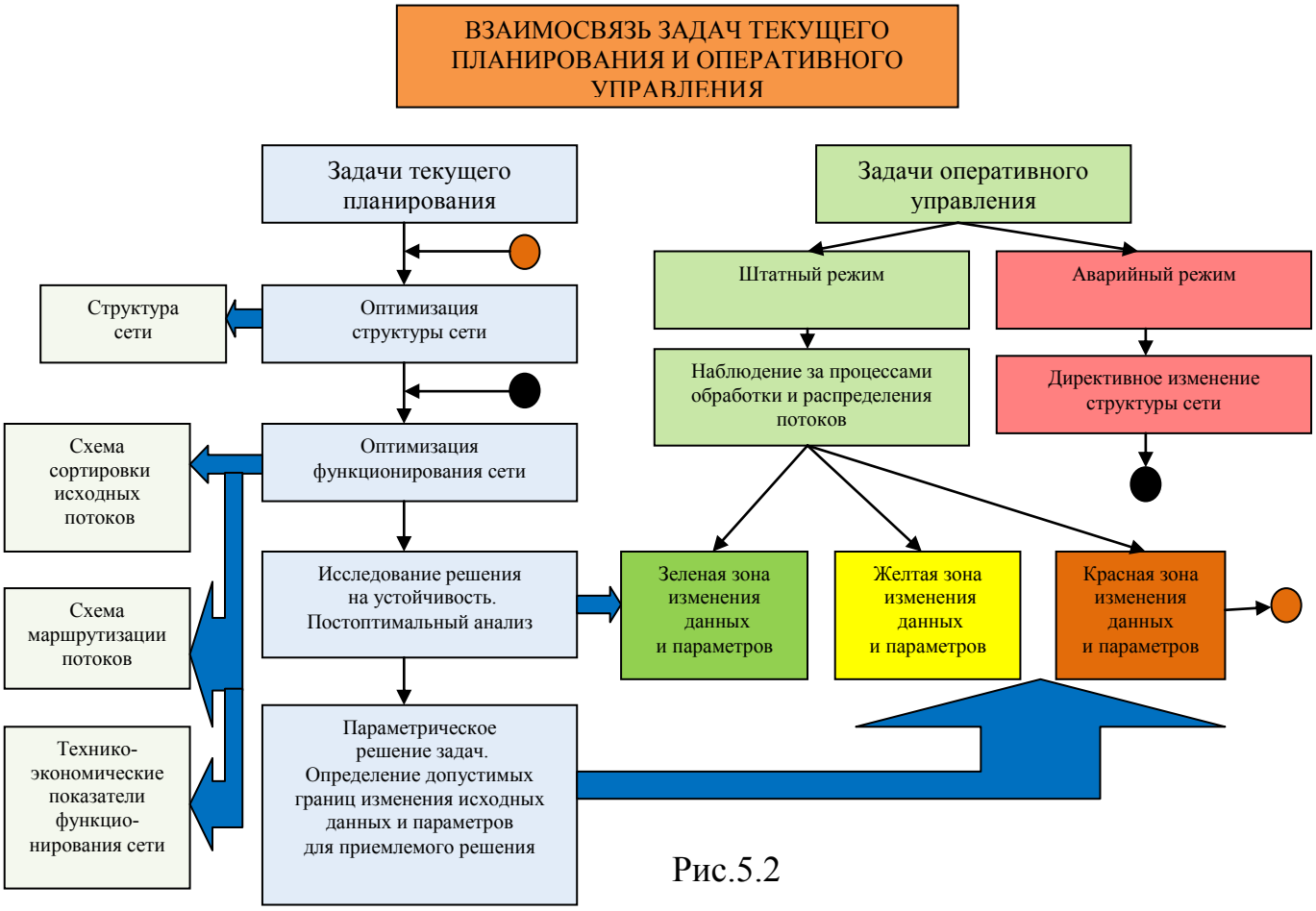


Рис.5.2

Желтая зона определяет предельные значения параметров, при которых еще допускается функционирование сети, но требуется повторное решение задачи выбора структуры сети и определение схем сортировки, распределения и

маршрутизации потоков. По результатам повторного решения задач текущего планирования можно оценить отклонение полных затрат и затрат на обработку и транспортировку потоков от допустимых границ их изменения для работы сети в штатном режиме (в зеленой зоне). Если окажется, что полученные после моделирования затраты превышают допустимые границы, то необходимо фактически перестраивать текущие схемы обработки и транспортировки потоков. Повторное решение задач текущего планирования может потребоваться не только при превышении верхних допустимых границ изменения контролируемых параметров, но и при уменьшении нижних границ. Так, например, падение потоков в узлах сети и уменьшение ниже нормы коэффициентов загрузки транспортных блоков и линий связи, также требует оперативного вмешательства и повторного решения всего комплекса задач.

Допустимые границы изменения значений параметров в зеленой, желтой и красной зоне определяются при исследовании решения задач текущего планирования на устойчивость и проведения постоптимального анализа решения при малых возмущениях потоков в узлах сети, а также при параметрическом решении задач, когда изменяться могут не только потоки, но и функции затрат и значения всех ограничений задачи. Главным критерием для выбора границ в контролируемой трех - зонной шкале параметров является допустимое отклонение значения целевой функции от ее «оптимального» значения при возмущениях входных данных и параметров решаемых задач. Границы допустимого изменения целевой функции устанавливаются проектировщиком сети и могут изменяться, например, в зависимости от текущего финансового состояния транспортного предприятия. Падение нагрузки в транспортных сетях позволяет высвободить часть транспортных средств и сдать их в аренду, поэтому допустимые границы изменения целевой функции могут быть расширены для уменьшения затрат, связанных с перестройкой схем обработки и транспортировки потоков. Постоянный рост потоков приводит к необходимости частого решения задач текущего планирования и перестройке сети, что требует дополнительных инвестиций, которые не всегда удается покрыть за счет имеющейся прибыли.

В случае перехода сети в аварийный режим, когда по различным причинам возникают отказы в узлах и на линиях связи, задачи текущего планирования должны быть решены немедленно и получены новые технико-экономические показатели функционирования сети в аварийном режиме. Если проектировщик принимает решение о директивном определении структуры сети с изменением зон обслуживания и типов магистральных узлов, а также при переводе магистральных узлов в ранг узлов четвертого типа, задача оптимизации структуры сети не решается. В этом случае возрастают затраты во внутренних зонах и для каждого магистрального узла должна быть решена задача маршрутизации внутриузловых потоков и получены суммарные приведенные затраты на обработку и транспортировку потоков во внутренних зонах. Затраты на обработку и транспортировку в магистральной сети определяются при решении задач упаковки, распределения и маршрутизации потоков. Проектировщик может многократно изменять структуру сети, решать соответствующие задачи и выбрать наиболее подходящий вариант. Следует отметить, что если магистральные узлы не переводятся в ранг узлов четвертого типа, а изменяются только их магистральные и внутренние зоны обслуживания, то при решении задачи выбора структуры магистральной сети нет необходимости учитывать затраты во внутренних зонах. Узлы четвертого типа по определению не могут быть магистральными.

В приложении Е приведены результаты вычислительного эксперимента, показывающие как можно использовать решение задач текущего планирования для перспективного развития узлов и транспортных линий связи магистральной сети и оперативного перераспределения потоков при возникновении внештатных ситуаций.

Выводы по пятому разделу

1. Предложенную методику планируется использовать для построения детерминированных и стохастических моделей задач перспективного развития узлов сети при заданных величинах инвестиций по этапам развития и при их ограниченном объеме на конечный период планирования.

2. Задачи оптимизации структуры сети и маршрутизации потоков могут с успехом применяться для получения исходных данных при построении динамических моделей поэтапного развития узлов и транспортных маршрутов сети с мелкопартионными потоками корреспонденций. При этом для этапов развития сети необходимо использовать современные модели и методы математического моделирования и прогнозирования на основе временных рядов данных, такие как метод группового учета аргументов, методы авторегрессии, нечеткие нейросети, байесовские модели и сети.

3. В условиях нестабильной экономики Украины в переходной период для задач перспективного развития многопродуктовых коммуникационных сетей с дискретными мелкопартионными потоками должны быть построены дисконтированные по времени детерминированные и стохастические модели. Особо важное внимание следует уделить построению стохастических динамических моделей развития сети, позволяющих получить вероятностные распределения капитальных затрат и других ресурсов по этапам развития и оценить степень риска их вложения.

4. Оперативное перераспределение потоков в случаях директивного ограничения пропускных способностей узлов, отказа отдельных узлов и линий связи может быть выполнено путем решения задач оптимизации структуры сети и маршрутизации потоков при изменении соответствующих параметров задач — установки новых ограничений и понижения ранга отказавших узлов.

РАЗДЕЛ 6. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ РЕШЕНИЯ ЗАДАЧ ТЕКУЩЕГО ПЛАНИРОВАНИЯ В МАГИСТРАЛЬНОЙ СЕТИ

В первом подразделе предложена методика решения задач текущего планирования в иерархической многопродуктовой сети, которая позволяет в интерактивном режиме осуществлять выбор иерархической структуры сети и определять основные технико-экономические показатели ее функционирования при изменении исходных данных и параметров.

Во втором и третьем подразделах приводится описание компьютерных программ (технологий) для моделирования иерархической структуры, схемы сортировки, распределения и маршрутизации потоков в коммуникационной сети. Рассматриваются числовые примеры решения отдельных задач.

В четвертом подразделе приведены результаты моделирования автотранспортной сети, содержащей 120 магистральных узлов.

Компьютерные программы включены в состав инструментальных программных средств информационно-аналитической системы поддержки принятия решений (ИАС ППР), которая разрабатывается в Институте телекоммуникаций и глобального информационного пространства НАН Украины.

Основные результаты раздела опубликованы в работах [34, 35, 129-131, 462], а на компьютерные программы имеются авторские свидетельства [104-106].

Ключевые слова: магистральные коммуникационные сети, мелкопартионные потоки корреспонденций, задачи обработки и транспортировки потоков, компьютерные технологии, численное моделирование

6.1. Методика решения задач текущего планирования

Перед описанием методики кратко обобщим информацию по решению задач текущего планирования на магистральном уровне сети из предыдущих разделов. Задана иерархическая многопродуктовая магистральная сеть $G(N,P)$ с

множеством неориентированных дуг P, $p = \left| P \right|$ и множеством узлов $N = N_1 \cup N_2 \cup N_3$, $n = \left| N \right|$, где N_1, N_2, N_3 — множества узлов первого, второго и третьего типа соответственно. Предполагается, что известны географические координаты всех узлов, т.е. топология сети задана. Узлы сети соответствуют пунктам сортировки, отправления, назначения и перегрузки потоков, а дуги — участкам дорог для ТС или каналам связи для СПД, связывающим узлы сети. Типы узлов могут быть заданы проектировщиком сети или определяться при решении задачи выбора структуры сети (подраздел 3.5). В последнем случае по критерию минимума приведенных затрат на функционирование сети определяется состав и количество узлов каждого типа в автоматизированном режиме.

В многопродуктовой сети каждый узел может обмениваться корреспонденциями (продуктами, требованиями) со всеми остальными узлами. Корреспонденция характеризуется узлом-источником, узлом-стоком и величиной, которая для ТС задается числом тарно-штучных грузов в упаковке унифицированного размера, а для СПД — числом бит, байт, мегабайт и т.д. в передаваемых данных (сообщениях). Корреспонденции заданы матрицей мелкопартионных дискретных потоков, в которой строки соответствуют узлам-источникам, столбцы — узлам-стокам, а элементы матрицы определяют величину корреспонденций. В магистральной сети все корреспонденции должны транспортироваться в транспортных средствах или передаваться по каналам связи в транспортных блоках заданного размера (емкости, объема). Размер транспортного блока измеряется количеством вмещающихся в него единиц корреспонденций. Все магистральные узлы являются сортировочными центрами, в которых корреспонденции сначала сортируются по адресам (узлам) назначения, а затем упаковываются в транспортные блоки. Поскольку величины некоторых корреспонденций значительно меньше размера транспортного блока, они при сортировке могут несколько раз и в разных узлах объединяться (упаковываться) с корреспонденциями, имеющими другие адреса назначения. При таком объединении корреспонденций в узлах сети уменьшается количество направлений их сортировки и количество транспортных блоков, необходимых для их упаковки,

но в отдельных узлах появляются дополнительные объемы сортировки корреспонденций, не достигших адресов своего назначения. Кроме того, увеличивается время на доставку получателю тех корреспонденций, которые проходят дополнительную сортировку в транзитных узлах следования.

Каждый магистральный узел имеет внутреннюю зону обслуживания, в которую входят только узлы четвертого типа. Узлы четвертого типа явно не входят в математические модели задач текущего планирования на магистральном уровне. Предполагается, что исходящие и входящие потоки этих узлов обрабатываются в главном узле (первого, второго или третьего типа) внутренней сети. Исходящие потоки в узлах четвертого типа не сортируются по адресам назначения, а доставляются в главный узел внутренней сети и сортируются вместе с исходящими потоками из главного узла. Процедура преобразования потоков узлов четвертого типа в потоки на магистральной сети тривиальна, так как все узлы четвертого типа обслуживаются только одним главным узлом внутренней сети. Таким образом, принимается, что потоки узлов четвертого типа учтены в потоках между узлами магистральной сети.

На магистральном уровне узлы первого типа могут отправлять транспортные блоки в любые другие узлы первого типа, а также в узлы второго и третьего типа, находящиеся в магистральной зоне их обслуживания (рис.1.1). Узлы второго и третьего типа могут отправлять транспортные блоки только в ближайшие узлы первого типа и в узлы второго и третьего типа, находящиеся в их магистральной зоне обслуживания. Входящие потоки в транспортных блоках в узлы второго и третьего типа могут поступать только от узлов, расположенных в магистральной зоне обслуживания этих узлов. В узлах второго и третьего типа количество магистральных направлений сортировки мелкопартионных потоков ограничено зоной обслуживания узла, в то время как узлы первого типа могут сортировать мелкопартионные потоки во все магистральные узлы в зоне своего обслуживания и во все другие узлы первого типа в магистральной сети. Узлы второго и третьего типа отличаются функциональными возможностями, уровнем технической оснащенности, числом обслуживающего персонала и пр.,

в этих узлах запрещена сортировка транзитных мелкопартионных потоков (кроме потоков между узлами четвертого типа во внутренней сети). В узлах третьего типа, в отличие от узлов второго типа, запрещена обработка транзитных потоков транспортных блоков.

Управление процессами обработки и транспортировки потоков осуществляется с помощью информационно-аналитической системы (ИАС) поддержки принятия решений.

В основной состав задач текущего планирования входят:

Задача 1. Распределение и объединение мелкопартионных потоков корреспонденций из узлов второго и третьего типа в магистральных зональных сетях. Математическая модель задачи и алгоритмы ее решения приведены в приложении Б. Согласно принципам зонально-узловой сортировки исходящие мелкопартионные потоки корреспонденций из узлов второго и третьего типа рассортировываются в узлы первого типа, лежащие на границе зон их обслуживания. Структурные особенности задачи позволяют использовать для ее решения метод декомпозиции Данцига-Вулфа. Для практического решения задачи предложены приближенные алгоритмы, основанные на методах построения кратчайших путей. В результате решения формируются: преобразованная матрица мелкопартионных потоков корреспонденций $A' = \| a'_{ij} \|_{n \times n}$; элементы справочной матрицы объединения мелкопартионных потоков для узлов второго и третьего типа (матрицы C). В строках и столбцах преобразованной матрицы A' для узлов второго и третьего типа ненулевые значения будут присутствовать только для узлов первого типа, которые находятся в зоне обслуживания узлов второго и третьего типа, а в потоках между узлами первого типа будут учтены все потоки узлов второго и третьего типа.

Задача 2. Распределение и объединение мелкопартионных потоков корреспонденций между узлами первого типа в магистральной сети. На вход задачи после решения задачи 1 поступают частично преобразованная матрица потоков A' и частично сформированная (только для исходящих потоков из узлов второго и третьего типа) справочная матрица объединения потоков C. Матема-

тическая модель задачи и алгоритмы ее решения приведены в подразделе 3.3.

Основные результаты решения задачи следующие: потоковая матрица $\tilde{A} = \| \tilde{a}_{ij} \|_{n \times n}$, $\tilde{a}_{ij} = \left\lceil \frac{a'_{ij}}{\omega} \right\rceil$ в транспортных блоках (обозначаемая также как $U = \| u_{ij} \|_{n \times n}$), где ω — объем транспортного блока; матрица предварительных оценок времени доставки мелкопартионных потоков получателям; окончательно сформированная справочная матрица объединения мелкопартионных потоков C. Эти результаты используются как входные данные для решения задачи распределения и маршрутизации потоков транспортных блоков. Справочная матрица объединения потоков является главным результатом решения задач 1 и 2. Она полностью определяет схему сортировки мелкопартионных потоков корреспонденций во всех узлах сети и адресует потоки транспортных блоков, которые будут распределены по маршрутам транспортных средств или по каналам связи. В узлах реальной транспортной сети или сети передачи данных справочная матрица используется для автоматизированного управления оборудованием, осуществляющим процессы сортировки адресных грузов, или как таблица слияния сообщений в виртуальные контейнеры.

При решении задачи 2 рассчитываются только предварительные оценки транспортных затрат и затрат на обработку транспортных блоков. Реальные оценки этих затрат могут быть получены только после решения задачи распределения и маршрутизации сформированных транспортных блоков на транспортной сети или сети передачи данных (в затратах на обработку транспортных блоков в узлах сети учитываются только исходящие и входящие потоки, — транзитные потоки не учитываются; для транспортных сетей не учитываются затраты на перевозку и обработку порожних контейнеров). Затраты на сортировку мелкопартионных потоков в узлах сети отражают реальные затраты.

Задачи 1 и 2 используются при решении задачи выбора иерархической структуры сети, которая рассмотрена в подразделе 3.5. Если начальная структура сети не задана (не определены узлы первого, второго и третьего типа), то типы и зоны обслуживания узлов определяются при ее решении.

Задача 3. Балансировка матрицы потоков в транспортных блоках (или задача развозки порожних контейнеров). Такая задача возникает для транспортных сетей в случае, если мелкопартионные грузы перевозятся в многооборотной таре — жестких контейнерах. На вход задачи поступает матрица потоков $U = \| u_{ij} \|_{n\times n}$, полученная после решения задачи 2. Матрица U не сбалансирована, поскольку для большинства узлов не выполняется условие $\Delta_i = \sum_{j=1}^{n}(u_{ij} - u_{ji}) = 0, \ \ i = \overline{1,n}.$ Задача сводится к классической транспортной задаче и решается известными методами по критерию минимума расстояния от узлов источников до узлов потребителей порожних контейнеров (приложение В). Для балансировки матрицы U потоки порожних контейнеров добавляются к потокам загруженных контейнеров с такими же адресами назначения.

Задача 4. Распределение и маршрутизация потоков транспортных блоков. На вход задачи поступают: исходная $A = \| a_{ij} \|_{n\times n}$ и преобразованная $A' = \| a'_{ij} \|_{n\times n}$ матрицы мелкопартионных потоков, справочная матрица объединения потоков $C = \| c_{ij} \|_{n\times n}$, несбалансированная или сбалансированная матрица потоков $\tilde{A} = \| \tilde{a}_{ij} \|_{n\times n}$ (или $U = \| u_{ij} \|_{n\times n}$) в транспортных блоках и другие данные, полученные после решения задачи 2.

Приведем общую схему методики решения задач текущего планирования, которая включает следующие основные этапы.

Этап 1. Для заданной топологии сети $G(N,P)$ с потоками $A = \| a_{ij} \|_{n\times n}$: указать тип каждого узла (первый, второй, третий). Типы узлов можно не указывать, тогда они будут определяться автоматически при оптимизации структуры сети. Задать функции затрат на сортировку мелкопартионных потоков C^i_{sort} и обработку транспортных блоков C^i_{load} в узлах сети; для всех узлов в сети указать ограничения на пропускные способности h_i и b_i; установить ограничения на время доставки мелкопартионных потоков между всеми узлами сети T_{ij} и ограничение на среднюю задержку потоков транспортных блоков в сети $T_{\max}$; уста-

новить максимальное нормированное время обработки (сортировки) мелкопартионных потоков T_a и максимальное нормированное время транзитной обработки транспортных блоков T_b в узлах сети. Параметры T_a и T_b явно не входят в модель, но используются при расчете значений t_{ij}; задать размер транспортного блока — ω и максимальное число объединений мелкопартионных потоков корреспонденций — $\nu_{\max}$; задать множество проектируемых маршрутов — $\{m_k\}$, $k = \overline{1,l}$. Для каждого маршрута указать ограничения на грузоподъемность транспортного средства или пропускную способность канала связи — $W^k_{\eta\xi}(w_\xi)$; ограничения на максимальное суммарное число транспортных блоков, которое можно обработать в узле α — b^k_α; среднюю скорость и периодичность движения транспортных средств или передачи информации; функцию затрат — C^k_{tr}. По желанию проектировщика сети может быть указана необходимость автоматической генерации множества маршрутов, из которых будут отобраны наиболее рациональные, соответствующие заданным ограничениям и параметрам задачи.

Этап 2. Для заданной топологии сети, исходных данных и параметров решить задачу выбора структуры сети. Решение может выполняться в двух режимах: при экспертном задании и автоматическом выборе типов узлов. В любом случае будут решаться задачи 1 и 2 для различных составов типов узлов, пока не будет найдена структура сети, при которой минимизируются суммарные приведенные затраты на ее функционирование. Для любого режима оптимизации после получения и анализа результатов решения проектировщик может выборочно последовательно или одновременно изменять параметры T_a, T_b, ω, $\nu_{\max}$, $W^k_{\eta\xi}$, среднюю скорость движения транспортных средств (скорость передачи информации по каналам связи принимается постоянной и равна 25000 км/сек), коэффициент прогнозирования потоков. При решении задачи 2 могут быть ослаблены ограничения на пропускные способности узлов и ограничения на время доставки мелкопартионных потоков корреспонденций получателю

$t_{ij} \le T_{ij}$, $ij \in S$. Учитывая, что значения T_{ij} могут быть изначально заданы некорректно, так как заранее неизвестно, по каким путям будут переданы потоки, осуществляется проверка соответствия заданных значений T_{ij} расчетным T_{ij}^{r}, полученным при условном распределении всех потоков по кратчайшим путям по критерию — минимум транзитных узлов на пути следования потока, минимум длины пути. При некорректном задании отдельных значений T_{ij} необходимо ввести соответствующую поправку для корректировки этих значений. Начальное задание T_{ij} может быть выполнено автоматически путем расчета времени передачи всех потоков по кратчайшим путям по указанному выше критерию и последующим увеличением полученных значений на вводимую константу. Изменение времени доставки мелкопартионных потоков на величину поправки (например, кратную T_a) означает разрешение дополнительных объединений потоков на всех возможных путях их передачи. При решении задачи 2 время доставки рассчитывается предварительно. Окончательный расчет времени доставки выполняется при решении задачи распределения и маршрутизации транспортных блоков, когда определяются фактические пути транспортировки (передачи) потоков. После выбора окончательного варианта решения задачи 2, для транспортной сети может быть решена задача развозки порожних контейнеров.

Все три задачи объединены в одном инструментальном блоке решения.

Этап 3. Решить задачу 4 для транспортной сети или сети передачи данных. Для каждого типа сети имеется несколько режимов работы блока оптимизации, которые связаны с разрешением или запрещением разветвления потоков транспортных блоков при их распределении по маршрутам транспортных средств или каналам связи; введением новых маршрутов в случае невозможности распределения всех потоков при заданных маршрутах и ограничениях; генерацией множества шаблонов маршрутов и выбором из них наиболее рациональных для распределения всех потоков при заданных ограничениях; выборочным ослаблением ограничений (4.3)-(4.6) и $t_{ij} \le T_{ij}$, $ij \in S$. При распределении и маршру-

тизации потоков с ограничением (4.6) решается задача выбора грузоподъемности транспортных средств или пропускных способностей каналов связи, содержательная постановка и математическая модель которой приведены в подразделе 3.4.3. Использование различных режимов работы дает возможность проектировщику анализировать и оценивать варианты решения задачи на различных стадиях проектирования сети при имеющихся или планируемых ресурсах. При использовании адекватных функций затрат, проектировщик может оперировать реальными среднегодовыми приведенными затратами, необходимыми для приобретения и эксплуатации оборудования при выбранных структуре, исходных данных и параметрах проектируемой сети.

6.2. Компьютерная программа решения задачи упаковки и выбора структуры сети

По существу, моделирование иерархической структуры коммуникационной сети представляет собой компьютерную технологию, состоящую из сценариев действий проектировщика и программной системы при выборе структуры, входных данных и параметров проектируемой сети. В ИАС сценарии реализованы в виде многооконного и многослойного графического интерфейса, позволяющего в картографическом виде просматривать структуру сети и ее отдельные фрагменты; исходящие и входящие потоки в узлах сети; значения всех заданных ограничений и параметров оптимизации; результаты оптимизации для различных вариантов решения задачи и пр. Проектировщик в режиме диалога может изменять значения исходных данных и параметров задачи, получать множество решений и выбирать из них наиболее подходящее. При этом он всегда может сравнить варианты решения по оценке технико-экономических показателей функционирования сети в зависимости от выбранных параметров и критериев предпочтения. Для экспериментального исследования решения задачи выбора структуры сети и схемы сортировки потоков, а также для обучения диспетчеров работе с программой разработана ее демонстрационная версия. Такая программа также включена в состав ИАС и может работать в автономном

режиме, когда все необходимые входные данные генерируются датчиком псевдослучайных чисел.

На рис. 6.1 показана главная форма программы, в которую вводятся входные данные. В ней можно выбрать различные варианты работы программы и указать, нужно ли оптимизировать структуру сети в автоматизированном режиме или вводить типы узлов и зоны их обслуживания вручную, использовать в алгоритмах решения задачи функции затрат или нет.

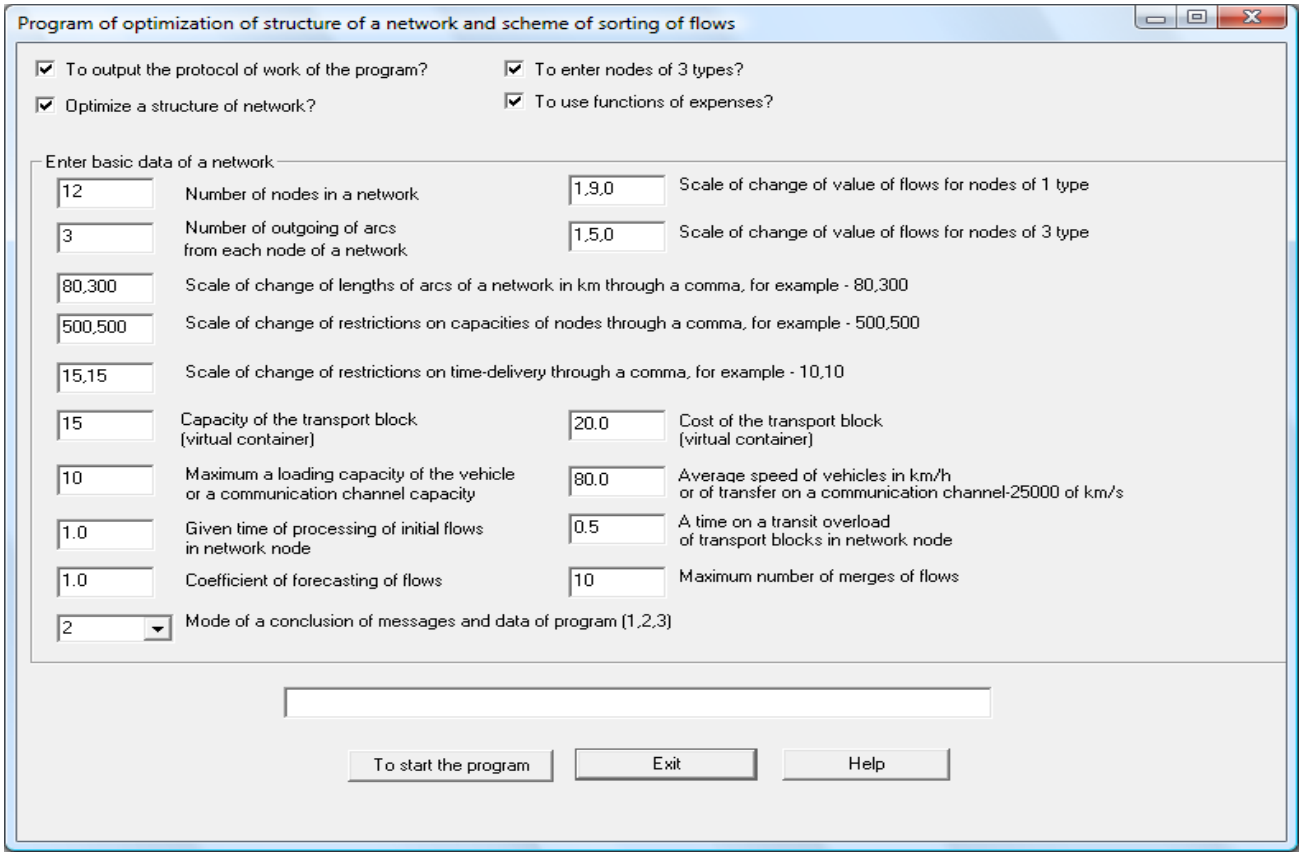


Рис. 6.1

Для изменения начальной величины потоков в процессе решения задачи может использоваться коэффициент прогнозирования потоков (K_p), начальное значение которого по умолчанию установлено равным единице. В нижней части главной формы расположены окно для вывода текущих сообщений программы и кнопки активизации действий.

После старта программы появляется диалоговая форма (рис. 6.2). На форме расположены: окно для вывода информационных сообщений и сообщений, требующих ответа; указатели учета ограничений на пропускные способности

узлов, время доставки корреспонденций получателю, изменения ключевых (критических) параметров и выбора окончательного варианта решения задачи. К ключевым параметрам относятся ω, w_{ξ}, V_{av}, T_a, T_b и K_p. Не ключевыми параметрами являются текущее ν и максимально допустимое $\nu_{\max}$ число транзитных объединений потоков корреспонденций, режим вывода сообщений и результатов решения задачи. Значения всех изменяемых параметров заносятся в соответствующие окна формы. Для выбора режима вывода результатов окончательного варианта решения задачи используется окно в нижней части формы.

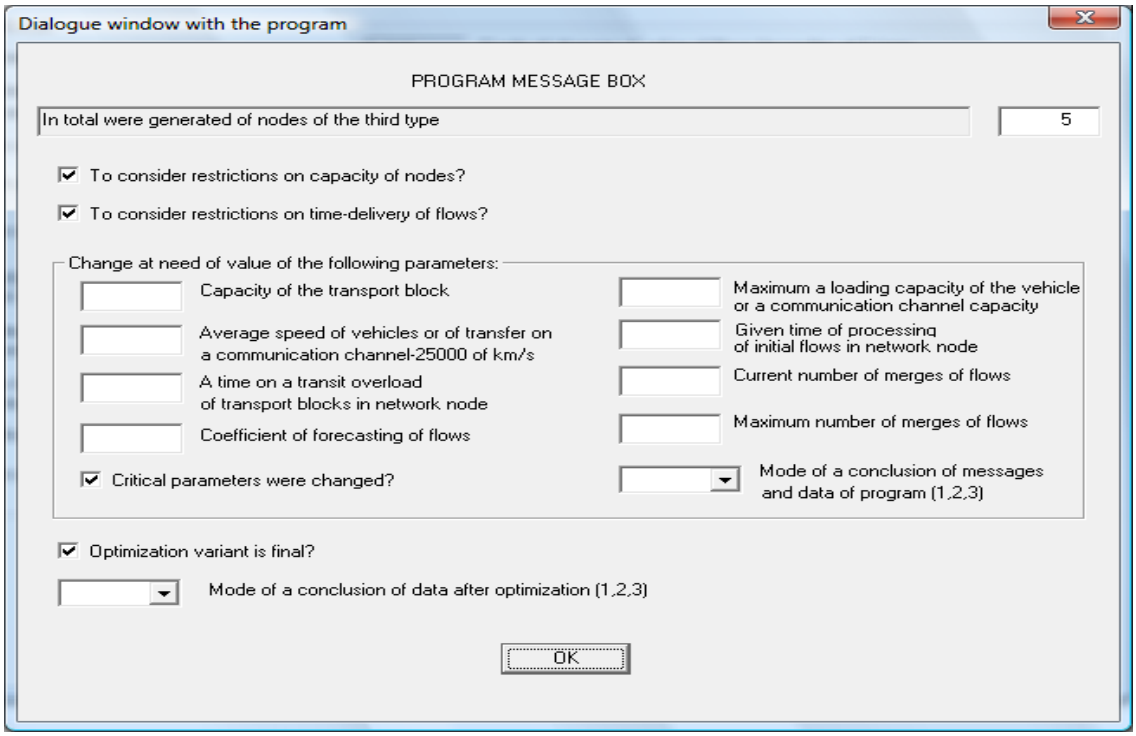


Рис. 6.2

Все выходные данные выводятся в два набора данных out1opt и out1 и отображаются на экране компьютера с помощью системной программы WordPad. В форму out1opt выводятся выбранная структура сети и основные технико-экономические показатели ее функционирования. В форму out1 для каждого узла сети в отредактированном виде выводится схема сортировки потоков корреспонденций и формирования потоков транспортных блоков. Если задача выбора структуры решается для транспортной сети, то в начало формы out1 дополнительно выводятся результаты решения задачи балансировки матрицы контейнерных потоков, возникающей из-за нарушения условий баланса — ра-

венства суммы исходящих и входящих контейнеров в отдельных узлах сети. Балансировка матрицы выполняется при решении задачи развозки порожних контейнеров.

Краткое описание работы программы. После запуска программы в ручном режиме (когда в верхней части главной формы не отмечено поле «оптимизация структуры сети», отмечено поле «вводить узлы третьего типа», поле «использовать функции затрат» отмечено или нет) появляется сообщение о количестве сгенерированных узлов третьего типа для заданной размерности сети и предложение ввести дополнительное число узлов третьего типа. По желанию проектировщика может быть добавлено несколько таких узлов, а затем, в ответ на очередной вопрос программы, указано окончательное число узлов третьего типа, выбранных для потенциального включения в структуру сети. Далее программа запрашивает о необходимости учета ограничений на пропускные способности узлов и время доставки корреспонденций. Проектировщик отмечает флажками нужные поля, и программа начинает работу. Работа программы заключается в итеративном решении задачи (3.18)-(3.21) для нарастающих значений $\nu \leftarrow \nu + 1$, являющихся в данном случае радиусами окрестности поиска экстремума в алгоритмах локального спуска. Значение ν увеличивается до тех пор, пока значения целевой функции (3.18) перестанут изменяться или $\nu \le \nu_{\max}$. В подразделе 3.3 предложены две группы алгоритмов локального спуска для решения задачи (3.18)-(3.21). Первая группа алгоритмов использует заданные функции затрат C_{tr}^{ij}, C_{sort}^{i} и C_{load}^{i}, вторая — нет. Разработка алгоритмов второй группы обоснована тем, что для реальных коммуникационных сетей сложно определить функции, достаточно адекватно характеризующие затраты на процессы обработки и транспортировки потоков корреспонденций. Эти алгоритмы ориентированы на максимальное сокращение транспортных блоков в сети, не используют в своей работе функции затрат, но полученное ими решение все равно оценивается (по заданным функциям затрат). Алгоритмы первой группы показывают лучшие результаты, но имеют высокие оценки временной сложности порядка $O(C_1 \nu_{\max} n^5)$. Сложность алгоритмов второй группы составляет от

$O(C_2 \nu_{\max} n^4)$ до $O(C_3 \nu_{\max} n^2)$. В подразделе 3.3.3 экспериментально показано, что результаты решения задачи (3.18)-(3.21), полученные различными группами алгоритмов на сетях, содержащих до 500 узлов, отличаются не более чем на 2,65%.

После определения наилучшего решения задачи (3.18)-(3.21) программа выводит предложение просмотреть результаты всех решений (при разных значениях ν), и выводит форму out1opt. Проанализировав полученные решения, проектировщик может выбрать наилучший вариант, отметив поле «вариант оптимизации окончательный» или перейти в диалоговое окно программы для изменения ключевых и не ключевых параметров. В случае изменения ключевых параметров необходимо отметить поле «изменение критических параметров». Программа переходит в режим оптимизации, и процесс решения задачи повторяется с новыми значениями измененных параметров. При выборе окончательного решения появляется запрос программы о необходимости решения задачи развозки порожних контейнеров. При положительном ответе эта задача решается. На завершающем этапе программа запрашивает ввести номер режима вывода результатов, после чего выводятся формы out1opt и out1 для принятого варианта решения задачи.

Если программа запускается в автоматизированном режиме выбора структуры сети, когда флажками отмечены поля «оптимизация структуры сети» и «вводить узлы третьего типа», появляется то же сообщение о количестве сгенерированных узлов третьего типа и предложение ввести дополнительное число узлов третьего типа. После ответа проектировщика программа начинает оптимизировать структуру сети, сообщает о числе узлов второго и третьего типа, включенных в сеть и выводит форму out1opt для наилучшего решения задачи. Далее появляется вопрос об окончании оптимизации, и программа выполняет все описанные выше действия.

Числовой пример. Для входных данных (показанных на рис. 6.1) была сгенерирована датчиком псевдослучайных чисел автотранспортная сеть $G(N,P)$ с числом узлов $n = 12$ и степенью узлов $val = 3$. Длины дуг изменяют-

ся в пределах от 80 до 300 км, а величина исходящих мелкопартионных потоков грузов из узлов первого типа задана в пределах от 1 до 9 единиц, из узлов 2 и 3 типа — в пределах от 1 до 5 единиц. Приняты следующие значения ограничений и параметров: пропускные способности узлов по обработке транзита одинаковы и h_i = 500, $i = \overline{1,n}$ единиц мелкопартионных грузов; максимальное время доставки мелкопартионных грузов получателю одинаково для всех корреспонденций и T_{ij} = 15, $ij \in S$ суток; начальный размер контейнера ω = 15 единиц мелкопартионных грузов (в процессе проведения эксперимента варьировался); начальная максимальная грузоподъемность автотранспортных средств w_{ξ} = 10 контейнеров (в процессе проведения эксперимента варьировалась); время на сортировку мелкопартионных грузов в узлах сети T_a = 1 сутки; время на транзитную перегрузку контейнеров в узлах сети T_b = 0,5 суток; стоимость одного контейнера C_{con} = 20 (варьировалась в зависимости от ω); средняя скорость движения транспортных средств V_{av} = 80 км/час; максимальное допустимое число транзитных объединений мелкопартионных грузов $\nu_{\max}$ = 10; режим вывода сообщений и результатов решения задачи $Mode$ = 2.

При решении задачи рассчитывались до и после оптимизации: $U = \sum_{ij \in S} \left\lceil \frac{a_{ij}}{\omega} \right\rceil$, $U = \sum_{ij \in S} \left\lceil \frac{x_{ij}}{\omega} \right\rceil$ — общее количество транспортных блоков в сети; $K_{av} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\delta_i} \sum_{j=1}^{n} x_{ij} / \left(\left\lceil \frac{x_{ij}}{\omega} \right\rceil \omega \right)$, $\delta_i = \sum_{j=1}^{n} \delta_{ij}$ — средний коэффициент загрузки транспортного блока; $N_{av} = \frac{1}{n} \sum_{i=1}^{n} \delta_i$ — среднее число направлений сортировки мелкопартионных корреспонденций в узле сети; значения минимального $\min flow$ и максимального $\max flow$ потока транспортных блоков по дуге сети. Для расчета среднегодовых приведенных затрат на обработку и транспортировку потоков использовались конкретные функции, характерные для автотранспортных предприятий.

Работа программы выполнялась в автоматизированном режиме. Для сети изначально было сгенерировано пять узлов третьего типа и дополнительно добавлено еще три узла. Сначала задача оптимизации структуры сети решалась для $w_\xi \in \{10, 15, 20,\ldots,40, 45\}$ при неизменных значениях потоков мелкопартионных грузов и других заданных параметров. Результаты решения приведены в табл. 6.1.

Таблица 6.1

Значения w_ξ, $\xi = \overline{1,8}$	F, C_{tr}, C_{sort}, C_{load}, (у.е.), U, K_{av}, N_{av}, min *flow*, max *flow*		$F_0 - F_{\min}$, у.е.	ν_{opt}	t_{av}, сут.
	До оптимизации	После оптимизации			
10	374482 279420 66166.2 28895.6 90, 0.476, 7, 2, 8	246705 158215 68233.4 20256.4 63, 0.761, 4, 1, 5	127777	4	3.09
15	281342 186280 66166.2 28895.6 90, 0.476, 7, 2, 8	194687 107035 67715.7 19936.4 62, 0.731, 5, 1, 5	86655	6	3.11
20	234772 139710 66166.2 28895.6 90, 0.476, 7, 2, 8	169084 80910.0 67917.5 20256.4 63, 0.727, 4, 1, 5	65688	2	3.10
25	206830 111768 66166.2 28895.6 90, 0.476, 7, 2, 8	152902 64728.0 67917.5 20256.4 63, 0.727, 4, 1, 5	53928	2	3.10
30	188202 93140.1 66166.2 28895.6 90, 0.476, 7, 2, 8	141038 53291.9 67810.1 19936.4 62, 0.744, 4, 1, 4	47164	4	3.12
35	174896 79834.4 66166.2. 28895.6 90, 0.476, 7, 2, 8	133425 45678.7 67810.1 19936.4 62, 0.744, 4, 1, 4	41471	4	3.12
40	164917 69855.1 66166.2 28895.6 90, 0.476, 7, 2, 8	127715 39968.9 67810.1 19936.4 62, 0.744, 4, 1, 4	37202	4	3.12
45	157155 62093.4 66166.2 28895.6 90, 0.476, 7, 2, 8	123274 35527.9 67810.1 19936.4 62, 0.744, 4, 1, 4	33881	4	3.12

Для всех значений w_ξ была получена одна и та же структура сети, содержащая 9 узлов первого типа (n_1 = 9) и 3 узла второго типа (n_2 = 3), что наряду с полученными результатами в подразделе 3.5.2 еще раз доказывает, что структу-

ра сети не зависит от изменения величины w_ξ.

В табл. 6.2 приведены результаты решения задачи при изменении $\omega_\xi \in \{10, 15, 20,\ldots,40, 45\}$ и постоянных значениях остальных входных данных.

Таблица 6.2

Значения ω_ξ, $\xi = \overline{1,8}$	F, C_{tr}, C_{sort}, C_{load}, (y.e.), u_0, K_{av}, N_{av}, min *flow*, max *flow*		$F_0 - F_{\min}$, y.e.	ν_{opt}	t_{av}, сут.
	До оптимизации	После оптимизации			
10	216071 109421 64315.9 42334.4 132, 0.348, 11, 5, 12	146367 54484.6 67146.1 24736.0 77, 0.834, 6, 1, 7	69704	2	3.18
15	164917 69855.1 66166.2 28895.6 90, 0.476, 7, 2, 8	127715 39968.9 67810.1 19936.4 62, 0.744, 4, 1, 4	37202	4	3.12
20	216071 109421 64315.9 42334.4 132, 0.174, 11, 5, 12	119446 33548.5 69160.8 16736.6 52, 0.732, 4, 1, 5	96625	3	3.48
25	164917 69855.1 66166.2 28895.6 90, 0.285, 7, 2, 8	116444 30881.1 68826.6 16736.7 52, 0.567, 4, 1, 3	48473	2	3.16
30	164917 69855.1 66166.2 28895.6 90, 0.238, 7, 2, 8	112952 28604.9 69210.6 15136.8 47, 0.536, 3, 1, 3	51965	5	3.35
35	164917 69855.1 66166.2 28895.6 90, 0.204, 7, 2, 8	112495 27861.3 69496.7 15136.8 47, 0.473, 3, 1, 3	52422	5	3.31
40	216071 109421 64315.9 42334.4 132, 0.087, 11, 5, 12	110269 25048.3 71684.0 13536.9 42, 0.518, 3, 1, 4	105802	5	3.75
45	195212 93091.1 64905.9 37214.8 116, 0.102, 9, 2, 11	108636 23971.3 71447.3 13216.9 41, 0.483, 3, 1, 3	86576	5	3.55

Для значений ω_ξ = 10, 20, 40 все узлы сети оказались узлами первого типа, для ω_ξ = 15, 25, 30, 35 была получена структура из n_1 = 9 и n_2 = 3 узлов, а для ω_ξ = 45 в сеть было включено n_1 = 11 и n_2 = 1 узлов. Такую нестабильность результатов можно объяснить тем, что при небольшой размерности сети и заданном значении ω_ξ, локальные минимумы F^* для различных структур сети от-

стоят очень близко друг от друга и от $F_{\min}$ в том числе (табл. 6.3). Из табл. 6.3 видно, что отклонения от лучшего локального минимума составляют всего от 0.01% до 1.73%. Следует ожидать, что при увеличении размерности сети локальные оптимумы будут не так близки друг к другу, и структура сети при изменении параметра ω_ξ будет оставаться более стабильной.

Таблица 6.3

Значения ω_ξ, $\xi = \overline{1,8}$	F^*, у.е. v_{opt}, n_1, n_2	$F_{\min}$, у.е. v_{opt}, n_1, n_2	$F^* - F_{\min}$, у.е. (%)
10	146534 5, 11, 1	146367 2, 12, 0	167 (0.11%)
15	129106 3, 10, 2	127715 4, 9, 3	1391 (1.09%)
20	119813 2, 9, 3	119446 3, 12, 0	367 (0.31%)
25	116585 7, 12, 0	116444 2, 9, 3	141 (0.12%)
30	114911 2, 12, 0	112952 5, 9, 3	1959 (1.73%)
35	112509 5, 11, 1	112495 5, 9, 3	14 (0.01%)
40	110488 3, 11, 1	110269 5, 12, 0	219 (0.20%)
45	109041 4, 12, 0	108636 5, 11, 1	405 (0.37%)

В табл. 6.2 и 6.3 во втором столбце приведены результаты, полученные после распределения потоков из узлов второго типа для выбранной структуры сети. Из этих таблиц видно, что при увеличении w_ξ и ω_ξ транспортные затраты и затраты на обработку контейнеров уменьшаются, затраты на сортировку потоков остаются практически постоянными. Адекватно изменяются и значения других параметров. Среднее время доставки грузов не превышает 3.75 суток.

Решение задачи проводилось на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб. Время выполнения программы для всех вариантов решения не превышало четырех секунд.

6.3. Компьютерная программа решения задачи распределения и маршрутизации потоков

Содержательная постановка задачи заключается в выборе такой схемы распределения и маршрутизации потоков транспортных блоков, сформированных при решении задачи выбора иерархической структуры магистральной сети

и схемы сортировки мелкопартионных корреспонденций в узлах сети, при которой максимально снижаются приведенные затраты на обработку и транспортировку потоков. Решение задачи должно осуществляться в интерактивном режиме и определять основные технико-экономические показатели функционирования магистральной сети при изменении исходных данных, параметров и ограничений транспортной модели.

В подразделе 4.1 предложены математическая модель NP-трудной задачи распределения и маршрутизации транспортных блоков с упакованными в них мелкопартионными грузами или сообщениями метод ее преобразования к некоторой совокупности более простых линейных многомерных задач о ранце со связывающими ограничениями. При этом исходная задача на минимум целевой функции заменяется в некотором смысле на двойственную к ней задачу на максимум загрузки участков маршрутов транспортных средств или каналов связи при исключении из явного рассмотрения нелинейных функций затрат. Для решения преобразованной задачи, которая также является NP-трудной, разработаны алгоритмы, существенно использующие специфику ее структуры, абстрактные типы данных и приемы, характерные для эвристических алгоритмов решения многомерной задачи о ранце. Алгоритмы позволяют за приемлемое время получить рациональные, с точки зрения проектировщика сети, решения задачи.

Рассмотрим кратко различные варианты и режимы решения задачи. Из постановки задачи видно, что если ограничения (4.8 или 4.14) не заданы, то любой поток транспортных блоков может разветвляться. При учете ограничений (4.6 или 4.15) пропускные способности маршрутов (грузоподъемность транспортных средств) или каналов связи выступают как искомые величины, и возникает необходимость итеративного решения задач распределения потоков и выбора пропускных способностей. В табл. 6.4 и 6.5 приведены варианты и возможные режимы решения задачи для каждого варианта.

Сформулируем основные требования к программе, реализующей алгоритм решения задачи. Ясно, что при некорректном задании ресурсов и принятых ог-

раничениях, задача может не иметь решения. В этом случае можно ослабить ограничения и попытаться перераспределить потоки. Выбор тех или иных ослабляемых ограничений может быть выполнен в интерактивном режиме. Поэтому программа должна обеспечить диалог с проектировщиком в случае невозможности получения решения при заданных ограничениях.

Таблица 6.4

Ограничения: 0 — не учитываются; 1 — учитываются	Номер варианта			
	1.1	1.2	2.1	2.2
(4.6 или 4.15) — среднее время задержки потоков	0	0	1	1
(4.8 или 4.14) — разветвление потоков	1	0	1	0

Таблица 6.5

Ограничения: 0- не учитываются; 1 - учитываются	Номер режима															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Пропускные способности узлов	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Пропускные способности маршрутов	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
Объемы обработки транспортных блоков в транзитных узлах маршрутов	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Время доставки корреспонденций получателю	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Кроме того структура программы должна быть такова, чтобы она могла обеспечить решение задачи в любом из вариантов и режимов, представляющих практический интерес.

Моделирование процессов распределения и маршрутизации потоков транспортных блоков представляет собой информационную компьютерную технологию, состоящую из сценариев действий проектировщика и программной системы в среде ИАС ППР, что обеспечивает диспетчерским службам на всех уровнях иерархии сети интерактивный картографический доступ к наборам пространственных данных, характеризующих состояние узлов и транспортных магистралей в реальном времени. Информационно-картографическая

система предоставляет возможность работать с многослойной электронной картой узлов и транспортных магистралей коммуникационной сети. На карте можно одновременно или в любой комбинации видеть узлы, их зоны обслуживания, транспортные магистрали, выделенные фрагменты сети (полигоны), маршруты движения транспортных средств и передачи информации, плановую и текущую загрузку узлов, транспортных средств на маршрутах движения и магистральных каналов связи, объемы неотправленных вовремя грузов и информации в узлах сети и множество других данных, характеризующих транспортную сеть. При этом можно изменять масштаб изображения, переключаться на разные слои электронной карты, вызывать различные прикладные программы. Для экспериментального исследования решения задачи распределения и маршрутизации потоков и для обучения диспетчеров работе с программой разработана ее демонстрационная версия. Программа включена в состав программного обеспечения ИАС и может работать в автономном режиме, когда все необходимые входные данные генерируются датчиком псевдослучайных чисел.

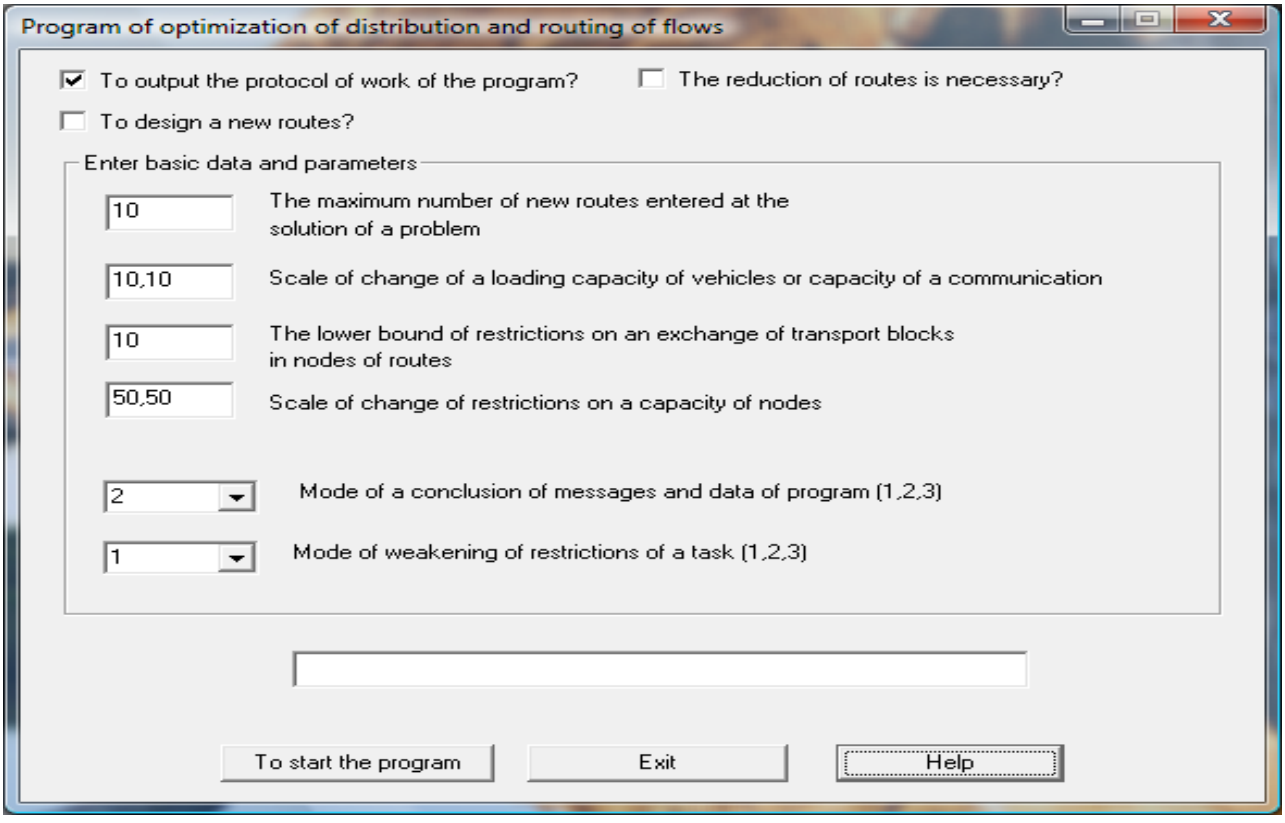


Рис. 6.3

На рис. 6.3 показана главная форма программы. Все входные данные про-

граммы вводятся из наборов данных, хранящих результаты решения задачи выбора структуры сети и схемы сортировки потоков корреспонденций. Вводятся исходная и преобразованная матрицы потоков корреспонденций $A=\| a_{ij} \|_{n\times n}$ и $A'=\| a'_{ij} \|_{n\times n}$, справочная матрица объединения потоков $C=\| c_{ij} \|_{n\times n}$, матрица потоков транспортных блоков $\widetilde{A}=\| \widetilde{a}_{ij} \|_{n\times n}$, матрица предварительных оценок времени доставки $T=\| T_{ij} \|_{n\times n}$ и другая информация, полученная после решения задачи выбора структуры сети и необходимая для решения задачи распределения и маршрутизации потоков. Задача может решаться при ручном проектировании маршрутов и в автоматизированном режиме, когда маршруты генерируются в процессе решения. Для автоматизированного режима (отмечено поле «проектировать новые маршруты») в окнах главной формы можно указать нижнюю и верхнюю границы значений $W^k_{\eta\xi}$, нижнюю границу значений b^k_α (верхняя граница равна $2W^k_{\eta\xi}$) для проектируемых маршрутов и нижнюю и верхнюю границы для b_β. Так как в автоматизированном режиме генерируется большое количество шаблонов маршрутов, то можно отметить поле «нужна ли редукция маршрутов» для экономии используемой оперативной памяти компьютера. При проектировании маршрутов в ручном режиме открывается окно для ввода маршрутов и их характеристик. Для обоих режимов можно указать максимальное количество вводимых маршрутов при невозможности распределения всех потоков при заданных ограничениях, выбрать условия ослабления ограничений и параметр вывода результатов решения задачи. Если условия ослабления равны единице, то все нарушаемые ограничения в процессе решения будут автоматически ослабляться, при равенстве условий двум, программа выводит запрос на разрешение ослабления каждого нарушаемого ограничения, при равенстве условий трем — ослабления ограничений запрещены. В нижней части главной формы расположены окно для вывода текущих сообщений программы и кнопки активизации действий.

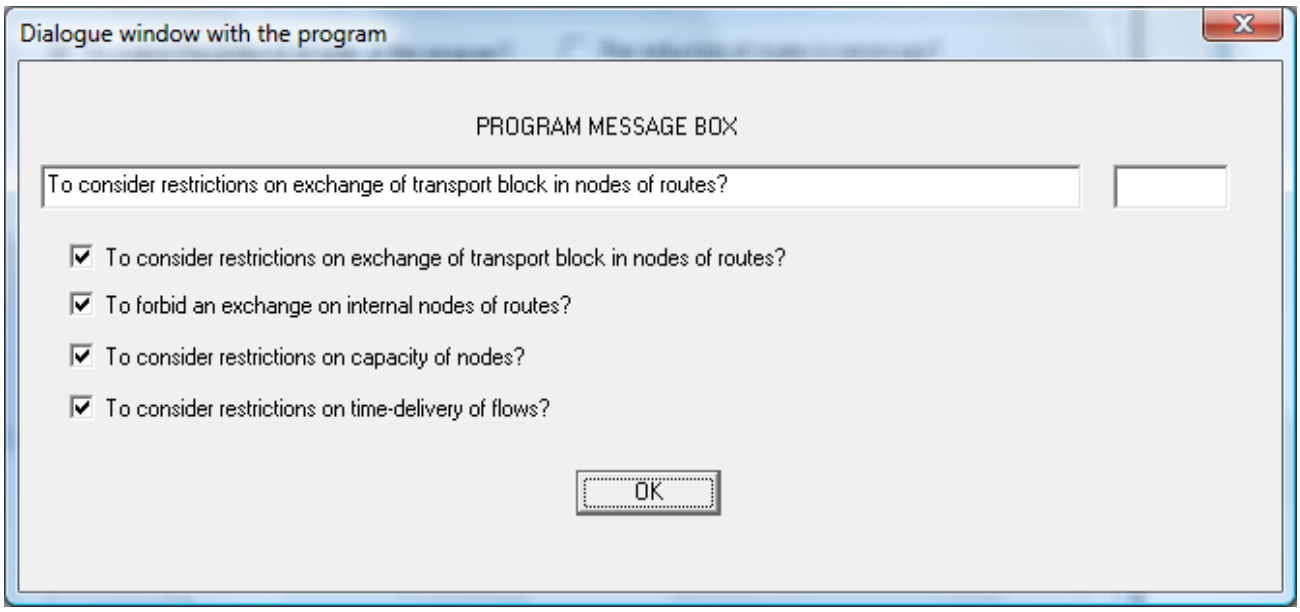


Рис. 6.4

После установки нужных параметров и старта программы появляется диалоговая форма (рис. 6.4). На форме расположены окно для вывода информационных сообщений и сообщений, требующих ответа и поля с флажками указателями учета ограничений. Все выходные данные выводятся в три набора данных out1opt, out1 и out2 и отображаются на экране компьютера с помощью системной программы WordPad. В форму out1opt выводятся основные технико-экономические показатели функционирования сети, полученные в результате решения задачи выбора ее структуры. В форму out1 для каждого узла сети в отредактированном виде выводится схема сортировки потоков корреспонденций и формирования потоков транспортных блоков. Если задача выбора структуры решалась для транспортной сети, то в начало формы out1 дополнительно выводятся результаты решения задачи развозки порожних контейнеров. В форме out2 приводятся схема сортировки потоков корреспонденций с окончательными оценками времени доставки корреспонденций получателям и результаты решения задачи распределения и маршрутизации потоков транспортных блоков. Результаты решения содержат технико-экономические показатели функционирования сети и полную информацию о путях транспортировки потоков на оптимизированных маршрутах.

Краткое описание работы программы. Для транспортной сети или сети передачи данных имеется несколько вариантов работы программы, которые связаны с разрешением или запрещением разветвления потоков транспортных блоков (4.8 или 4.14) и соблюдением ограничений (4.6 или 4.15) (табл. 6.4). Выбор вариантов осуществляется в диалоговой форме программы. Для каждого варианта существует шестнадцать режимов оптимизации (табл. 6.5), при которых одни ограничения учитываются, а другие — ослабляются. Возможно выборочное ослабление ограничений на пропускные способности узлов, маршрутов, объемы обработки транспортных блоков в узлах маршрутов и $t_{ij} \le T_{ij}$, $ij \in S$. Кроме того при решении задачи допускается введение новых маршрутов в случае невозможности распределения всех потоков при заданных маршрутах и ограничениях, генерация множества шаблонов маршрутов и выбор из них наиболее рациональных для распределения всех потоков при заданных ограничениях. Использование различных режимов работы программы дает возможность проектировщику анализировать и оценивать варианты решения задачи на различных стадиях проектирования сети при имеющихся или планируемых ресурсах. При использовании адекватных функций затрат, проектировщик может оперировать реальными среднегодовыми приведенными затратами, необходимыми для приобретения и эксплуатации оборудования при выбранных структуре, исходных данных и параметрах проектируемой сети. Диалог с программой полностью прозрачен и требует выполнения простых действий проектировщика.

Числовой пример. Задача маршрутизации решалась для входных данных, показанных на рис. 6.1. При этом выбор структуры сети (A) выполнялся для $w_\xi \in \{10, 5, 3\}$ при неизменных значениях потоков мелкопартионных грузов и других заданных параметров. Для всех значений w_ξ была получена одна и та же структура сети, содержащая девять узлов первого типа и три узла второго типа. Результаты решения этой задачи использовались как входные данные для решения задачи распределения и маршрутизации потоков контейнеров, которая решалась в ручном (B, C) и автоматизированном (D) режимах.

В ручном режиме в варианте B маршруты транспортных средств совпадали с дугами сети, и предполагалось, что движение по маршрутам разрешено в обе стороны, поэтому заданное количество маршрутов в сети равно 18. В варианте C было введено дополнительно 5 маршрутов с максимальным числом узлов в маршруте равным 5. В варианте D для каждой пары узлов в сети генерировался маршрут транспортного средства, всего сгенерировано $l = n(n-1)/2$ = 12x11/2 = 66 маршрутов, движение по которым разрешалось в обе стороны. В варианте D любой поток контейнеров может быть доставлен получателю без транзитных перегрузок, т.е. в узлах сети нет транзитных потоков контейнеров. Этот крайний случай введен для того, чтобы оценить, как максимальное сокращение времени доставки мелкопартионных грузов получателю t_{ij} отражается на экономических показателях сети.

Принимались следующие значения параметров и ограничений: одинаковая грузоподъемность транспортных средств на всех маршрутах также варьировалась в пределах $w^k \in \{10, 5, 3\}$ контейнеров; в варианте C разрешался транзитный обмен контейнерами в промежуточных узлах маршрутов; пропускные способности узлов одинаковы и равны 50 контейнерам; максимальное время доставки мелкопартионных грузов конечному потребителю не ограничивалось, т.е. допускалась возможность максимальной загрузки транспортных средств при распределении потоков по «длинным» путям; ограничения на среднюю задержку транспортных блоков в сети t_{av} не учитывались; разветвление потоков транспортных блоков запрещено; ослабление ограничений, и ввод новых маршрутов разрешен.

Расчет среднегодовых приведенных затрат на транспортировку и обработку потоков контейнеров и затрат на сортировку выполнялся в условных единицах стоимости по формулам из подраздела 4.7. Рабочий парк транспортных средств (ТС) и рабочий парк контейнеров в сети перевозок рассчитывались для полученной схемы распределения и маршрутизации потоков контейнеров с учетом заданных параметров: времени стоянки транспортных средств в конечных узлах маршрутов, периодичности и средней скорости движения транспорт-

ных средств и длины маршрутов.

Результаты решения задач приведены в таблице 6.6 и на рис. 6.5, 6.6.

Лучший вариант решения (общие затраты составляют 215,2 тыс. у.е.) получен для варианта C при $w_\xi = 3$, когда на сети вводятся более длинные маршруты. При уменьшении значений w_ξ до минимального, общие затраты в вариантах B, C и D отличаются друг от друга в пределах 11,94 % (рис. 6.5), поэтому в случаях, когда проектировщик сети заинтересован в сокращении времени доставки корреспонденций получателям, может быть выбран вариант решения D. Среднее время доставки T_{av} остается для всех вариантов достаточно стабильным (рис. 6.6).

Таблица 6.6

Наименования показателей	Грузоподъемность транспортных средств в контейнерах											
	$w = 10$				$w = 5$				$w = 3$			
	Результаты решения задач оптимизации структуры сети (A), распределения и маршрутизации потоков с заданными маршрутами (B, C) и с генерацией маршрутов (D)											
	A	B	C	D	A	B	C	D	A	B	C	D
Заданное количество маршрутов	-	18	23	66	-	18	23	66	-	18	23	66
Количество введенных маршрутов	-	0	0	0	-	0	1	0	-	4	0	0
Количество загруженных маршрутов	-	18	5	18	-	18	6	18	-	22	14	19
Средний коэфф. загрузки ТС в сети	-	0.29	0.35	0.19	-	0.58	0.68	0.37	-	0.80	0.67	0.58
Исходный рабочий парк ТС	-	36	46	132	-	36	46	132	-	36	46	132
Требуемый рабочий парк ТС	-	36	10	36	-	36	12	36	-	44	28	38
Требуемый рабочий парк контейнеров	-	198	198	198	-	198	203	198	-	202	200	198
Транспортные затраты (тыс. у.е.)	158.2	304.8	227.6	439.5	316.4	163.1	139.3	230.4	533.5	123.1	122.8	151.1
Затраты на сортировку (тыс. у.е)	68.2	68.2	68.2	68.2	68.2	68.2	68.2	68.2	68.6	68.6	68.6	68.6
Затраты на обработку контейнеров (тыс. у.е.)	20.3	25.2	24.3	21.2	20.3	25.2	25.4	21.2	20.6	26.2	23.8	21.2
Общие затраты (тыс. у.е.)	246.7	398.2	320.1	528.9	404.9	256.5	232.9	319.8	622.7	217.9	215.2	240.9
$T_{\min}$	2.04	2.04	2.04	2.04	2.04	2.04	2.04	2.04	2.04	2.04	2.04	2.04
$T_{\max}$	5.39	5.39	5.69	4.40	5.39	5.39	6.23	4.40	8.14	8.14	7.22	6.64
T_{av}	3.09	3.09	3.13	2.81	3.09	3.09	3.25	2.81	3.14	3.21	3.11	2.86

Из табл. 6.6 видно, что при уменьшении w_ξ, адекватно изменяются и значения

других параметров.

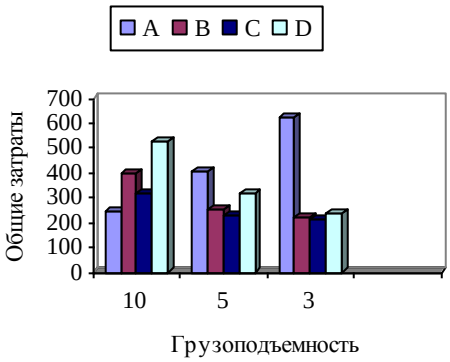


Рис. 6.5

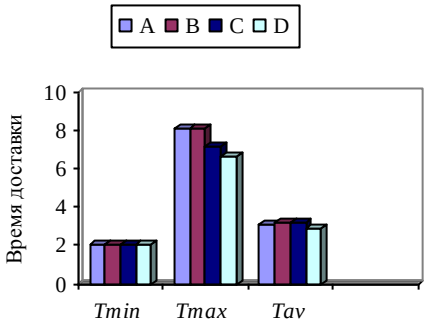


Рис. 6.6

Решение задач проводилось на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб. Время расчета задачи в вариантах B, C и D не превышало долей секунды.

Программное обеспечение разработано в среде *Microsoft Developer Visual Studio* и может быть адаптировано для работы в системе параллельного программирования *Intel® Parallel Studio XE 2016*, в которую вошли последние версии компиляторов C/C++ и Фортран (https://software.intel.com/ru-ru/try-buy-tools).

6.4. Обобщенный пример проектирования автотранспортной сети

В заключение раздела рассмотрим результаты математического моделирования решения задач текущего планирования на автотранспортной сети достаточно большой размерности, которые подтверждают полученные экспериментальные данные для сетей небольшой размерности. Для проведения эксперимента была сгенерирована датчиком псевдослучайных чисел автотранспортная сеть контейнерных перевозок с числом узлов $n = 120$ и степенью узлов $val = 5$. Длины дуг и величина исходящих мелкопартионных потоков грузов из узлов первого и второго типа задавались также как и предыдущих примерах. Для сети было сгенерировано 28 узлов второго типа и 92 узла первого типа. Предпо-

лагалось, что маршруты $\{m_k\}$, $k=\overline{1,l}$ транспортных средств совпадают с дугами сети и движение по маршрутам разрешено в обе стороны, поэтому было сгенерировано l = 120x5/2 = 300 проектируемых маршрутов.

Принимались следующие значения параметров: размер контейнера ω = 40 единицам мелкопартионных грузов; время на обработку (сортировку) мелкопартионных грузов в узлах сети $T_a = 24$ ч; грузоподъемность автотранспортных средств $W_{\eta\xi}^k$ на всех дугах $\eta\xi \in q_k$ всех маршрутов $\{m_k\}$, $k=\overline{1,l}$ принималась одинаковой, равной W, и варьировалась в пределах $W \in \{40, 20, 10\}$ контейнеров; время на транзитную перегрузку контейнеров в узлах сети T_b = 12 ч; время стоянки транспортных средств в конечных узлах маршрутов T_{end} = 22 ч; периодичность движения транспортных средств T_{move} = 24 ч; средняя скорость движения транспортных средств V_{av} = 80 км/час; суточные пропускные способности узлов по обработке транзита h_i и b_i, $i=\overline{1,n}$ принимались равными 5000 единиц мелкопартионных грузов и 500 контейнеров соответственно; время доставки мелкопартионных грузов получателю t_{ij} рассчитывалось в сутках, поэтому максимальное время доставки задавалось в сутках и принималось T_{ij} = 15 сут. для всех $ij \in S$; максимальное число объединений мелкопартионных грузов $\nu_{\max}$ не ограничивалось; средняя задержка контейнеров в сети t_{av} рассчитывалась, а их максимальная задержка $T_{\max}$ не ограничивалась; разветвление потоков контейнеров запрещено.

Для расчета среднегодовых приведенных затрат на транспортировку и обработку потоков использовались формулы из подраздела 4.6.

Задача оптимизации структуры сети решалась для $W \in \{40, 20, 10\}$ при неизменных значениях потоков мелкопартионных грузов и других заданных параметров. Для всех значений W была получена одна и та же структура сети, содержащая 79 узлов первого типа и 41 узел второго типа. Поэтому для решения задач распределения и маршрутизации потоков порожних и груженых контейнеров для $W \in \{40, 20, 10\}$ как входные данные принимались результаты реше-

ния задачи оптимизации структуры сети, полученные при W = 40.

Результаты решения задач на исходной и оптимизированной сети для грузоподъемности транспортных средств W = 40, 20, 10 приведены в табл. 6.7 и 6.8 и на рис. 6.7, 6.8. В таблицах приняты следующие обозначения: C_{all}, C_{tr}, C_{sort}, C_{load} — полные ($C_{all} = C_{tr}+C_{sort}+C_{load}$), транспортные, сортировочные, погрузочно-разгрузочные среднегодовые приведенные затраты в млн. условных единиц; l, l^{+}, l^{0} — количество заданных, введенных и загруженных маршрутов транспортных средств; D_{tr} — рабочий парк транспортных средств; K_{tr} — средний коэффициент загрузки транспортного средства; D_{con} — рабочий парк контейнеров; K_{con} — средний коэффициент загрузки контейнера; N_{sort} — среднее число магистральных направлений сортировки мелкопартионных грузов в узле сети; $t_{\min}$, $t_{\max}$, t_{mean} — минимальное, максимальное и среднее время доставки мелкопартионных грузов получателю в сутках; V_1, V_2, V_3 — варианты решения задачи.

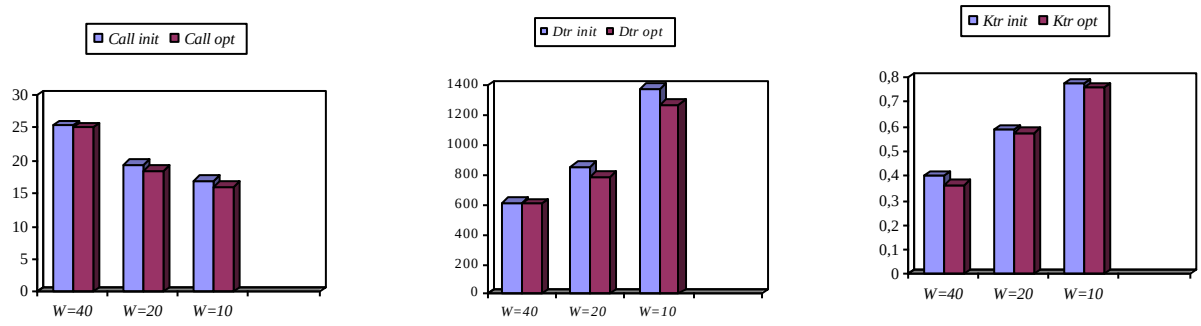
В вариантах V_1 и V_2 распределение потоков контейнеров осуществлялось по заданным маршрутам, совпадающим с дугами сети, и при невозможности распределения всех потоков разрешалось введение новых маршрутов. В варианте V_1 ограничения на время доставки мелкопартионных грузов получателю T_{ij} учитывались, а в варианте V_2 ослаблялись. В варианте V_3 для каждой пары узлов (i, j), $i = \overline{1, n-1}$, $j = \overline{i+1, n}$ в сети генерировался маршрут транспортного средства по критерию: минимум транзитных узлов, при равенстве числа транзитных узлов — минимум длины. Всего сгенерировано $l = n(n-1)/2$ = 120x119/2 = 7140 маршрутов, движение транспортных средств по маршрутам разрешалось в обе стороны. В варианте V_3 любой поток контейнеров может быть доставлен получателю без транзитных перегрузок. Из таблиц видно, что для вариантов V_1 и V_2 при W = 40 и W = 20 полные затраты при ослаблении ограничений на t_{ij} уменьшаются незначительно, а $t_{\max}$ заметно растет (рис. 6.8а).

Таблица 6.7

Результаты решения	Исходная сеть $N_1 = 92,\ N_2 = 28$								
	$W = 40$			$W = 20$			$W = 10$		
	V_1	V_2	V_3	V_1	V_2	V_3	V_1	V_2	V_3
C_{all}	25,396	25,186	152,78	19,464	19,337	79,397	16,975	19,048	42,706
C_{tr}	20,002	19,770	148,17	14,069	13,488	74,791	11,581	13,030	38,100
C_{sort}	3,216	3,216	3,216	3,216	3,216	3,216	3,216	3,216	3,216
C_{load}	2,178	2,200	1,389	2,178	2,632	1,389	2,178	2,802	1,390
l	300	300	7140	300	300	7140	300	300	7140
l^+	5	0	0	126	97	0	388	461	0
l^0	305	300	900	426	397	900	688	761	900
D_{tr}	610	600	1819	852	794	1819	1376	1522	1819
K_{tr}	0,400	0,412	0,055	0,587	0,806	0,109	0,775	0,905	0,221
D_{con}	14573	14638	13041	14573	16093	13041	14573	16715	13041
K_{con}	0,637	0,637	0,637	0,637	0,637	0,637	0,637	0,637	0,637
N_{sort}	21	21	21	21	21	21	21	21	21
$t_{\min}$	2,042	2,042	2,042	2,042	2,042	2,042	2,042	2,042	2,042
$t_{\max}$	6,738	9,359	4,775	6,738	16,999	4,775	6,738	18,302	4,775
t_{mean}	4,205	4,266	3,393	4,205	5,317	3,393	4,205	5,570	3,393

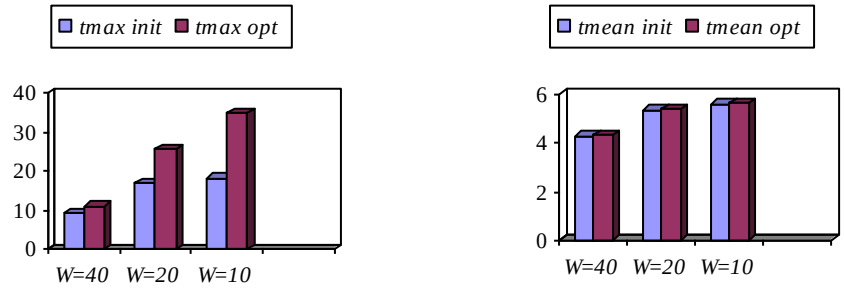
Таблица 6.8

Результаты решения	Оптимизированная сеть $N_1 = 79,\ N_2 = 41$								
	$W = 40$			$W = 20$			$W = 10$		
	V_1	V_2	V_3	V_1	V_2	V_3	V_1	V_2	V_3
C_{all}	25,116	25,030	116,64	18,430	18,415	61,336	15,993	17,679	33,687
C_{tr}	19,798	19,699	111,98	13,111	12,691	56,680	10,674	11,820	29,031
C_{sort}	3,293	3,293	3,293	3,293	3,293	3,293	3,293	3,293	3,293
C_{load}	2,026	2,038	1,363	2,026	2,431	1,363	2,026	2,566	1,363
l	300	300	7140	300	300	7140	300	300	7140
l^+	2	0	0	93	71	0	334	390	0
l^0	302	300	722	393	371	722	633	690	722
D_{tr}	604	600	1455	786	742	1455	1266	1380	1455
K_{tr}	0,364	0,370	0,064	0,575	0,783	0,128	0,758	0,902	0,259
D_{con}	13945	13983	12783	13945	15329	12783	13945	15762	12783
K_{con}	0,643	0,643	0,643	0,643	0,643	0,643	0,643	0,643	0,643
N_{sort}	19	19	19	19	19	19	19	19	19
$t_{\min}$	2,042	2,042	2,042	2,042	2,042	2,042	2,042	2,042	2,042
$t_{\max}$	7,251	11,012	6,554	7,251	25,573	6,554	7,251	34,835	6,554
t_{mean}	4,275	4,311	3,488	4,275	5,367	3,488	4,275	5,594	3,488



а) б) в)

Рис.6.7. Зависимость полных среднегодовых приведенных затрат (а), рабочего парка транспортных средств (б), коэффициента загрузки транспортных средств (в) от грузоподъемности транспортных средств для варианта V_1



а) б)

Рис.6.8. Максимальное (а) и среднее (б) время доставки грузов получателю в сутках при изменении грузоподъемности транспортных средств для варианта V_2

При $W = 10$, когда достигается максимальная загрузка транспортных средств, в варианте V_2 полные затраты не уменьшаются, а увеличиваются за счет распределения потоков по более длинным путям и увеличения рабочего парка контейнеров. Поэтому при уменьшении грузоподъемности лучшее решение может быть получено при заданных ограничениях на t_{ij}. Среднее время доставки t_{mean} остается для всех вариантов достаточно стабильным (рис. 6.8б). В варианте V_3 при уменьшении грузоподъемности транспортных средств наблюдается значительное снижение полных затрат при наименьших значениях $t_{\max}$ и t_{mean}. Это означает, что наилучшее решение задачи может быть получено в варианте V_3 при дальнейшем снижении грузоподъемности и отклонении потоков

с кратчайших путей для увеличения загрузки транспортных средств.

Выводы по шестому разделу

1. Разработанная методика решения задач обработки и распределения мелкопартионных дискретных потоков в иерархической многопродуктовой сети, позволяет в интерактивном режиме осуществлять выбор иерархической структуры сети и определять основные технико-экономические показатели ее функционирования при изменении исходных данных и параметров. Методика может быть использована для текущего (среднесрочного) планирования процессов обработки и транспортировки мелкопартионных потоков корреспонденций и нацелена главным образом на оптимизацию использования имеющихся ресурсов сети.

2. Методика основана на последовательном решении задач текущего планирования, которые представлены комплексом математических моделей, алгоритмов и компьютерных программ (технологий).

3. Разработанные компьютерные технологии и инструментальные программные средства предназначены для математического (компьютерного) моделирования функционирования коммуникационных сетей с дискретными мелкопартионными потоками в инфраструктурных комплексах крупных городов, регионов и страны в целом. Они могут быть использованы для решения задач текущего планирования и управления в транспортных сетях, магистральных сетях передачи данных, сотовой связи и др., а также для решения задач оперативного управления и перспективного развития и позволяют повысить расчетную эффективность функционирования проектируемых коммуникационных сетей.

4. Разработанные компьютерные технологии решения задач выбора иерархической структуры сети, схемы сортировки, распределения и маршрутизации потоков позволяют:

- в интерактивном режиме моделировать различные варианты сети, изменяя топологию, иерархическую структуру, маршруты транспортных средств или передачи информации, потоки, параметры и ограничения модели, и из семейства полученных результатов выбирать наилучший вариант с учетом вы-

бранной функции цели и принятых ограничений;

- повысить эффективность функционирования сети на уровне текущего планирования путем оптимизации использования ее имеющихся ресурсов и снижения эксплуатационных затрат, что дает возможность уменьшать тарифы на перевозку грузов или передачу информации, привлекать дополнительную клиентуру и обеспечивать постоянный прирост прибыли;
- рассчитывать предварительные технико-экономические показатели функционирования сети для заданных и прогнозных значениях потоков, оценивать стоимость дополнительных ресурсов и планировать величину потребных инвестиций на модернизацию и развитие ее структурных элементов, что в конечном итоге дает возможность повысить эффективность функционирования сети за счет снижения дефицитных материальных, сырьевых, энергетических, финансовых и трудовых ресурсов.
- оперативно перераспределять потоки при возникновении отказов оборудования в узлах и на транспортных маршрутах, непредвиденных ситуаций, стихийных бедствий и др.;

5. Экспериментальные исследования показали высокую вычислительную эффективность предложенных алгоритмов и программ, и они могут быть рекомендованы для практического решения задач оптимизации процессов обработки и транспортировки потоков в транспортных сетях и сетях передачи данных (время счета всех задач на транспортной сети, содержащей 120 узлов и 300 неориентированных дуг не превышало 30 с на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.).

6. Разработанный инструментарий может быть также использован для моделирования и оптимизации функционирования традиционных логистических производственных и транспортно-складских систем, включающих узлы поставщиков сырья, производства товаров, склады и конечных потребителей.

7. Кроме того, полученные результаты работы могут быть полезны в учебном процессе при подготовке специалистов в области моделирования и проектирования сложных распределенных многопродуктовых сетей.

ЗАКЛЮЧЕНИЕ

Диссертационная работа является законченным научным исследованием, посвященным решению актуальной научно-прикладной проблемы создания методологии математического моделирования и проектирования многопродуктовых коммуникационных сетей с дискретными потоками. При выполнении исследований в диссертационной работе получены следующие основные научно-технические результаты.

1. Разработаны теоретические и методологические основы построения и анализа функционирования многопродуктовых иерархических сетей с дискретными параметрами и мелкопартионными дискретными потоками корреспонденций, в том числе: концепции, принципы и критерии обработки и распределения потоков, методология синтеза и анализа многоуровневой структуры сети. Определены основные принципы и технология организации сортировки и транспортировки потоков в зонально-узловой структуре сети. В отличие от существующих подходов, предложенная методология основывается на комплексном решении задач долгосрочного, текущего и оперативного планирования и управления и позволяет логически увязать процессы обработки потоков в узлах сети с их последующим распределением на сети и доставкой получателям и при заданных ограничениях повысить экономическую эффективность функционирования и развития транспортной системы в целом за счет снижения приведенных затрат в узлах и на транспортных маршрутах.

2. Для управления нелинейными и нестационарными процессами обработки и распределения потоков в сети предложен концептуальный подход к построению многоуровневой ИАС ППР, которая основана на технологиях обработки пространственных данных и геоинформационных систем и функционирует в режиме реального времени.

3. Определены основные задачи текущего планирования, составляющие методологическую основу — ядро для решения задач оперативного управления процессами обработки и распределения потоков, а также для решения задач перспективного развития сети. Показана взаимосвязь задач текущего планиро-

вания с задачами оперативного управления и перспективного развития.

4. Предложены основные принципы и схемы организации перевозок во внутренних зонах магистральных узлов и определены технические и экономические особенности реальных транспортных процессов обработки и транспортировки мелкопартионных грузов, которые должны быть учтены при формировании целевых функций математических моделей задач маршрутизации на уровнях долгосрочного и текущего планирования и оперативного управления. На основании проведенного обзора и анализа известных математических моделей разработано несколько вариантов математических постановок задачи проектирования доставочных, сборочных и комбинированных маршрутов транспортных средств для перевозки мелкопартионных грузов. Для решения поставленных задач могут быть применены точные, эвристические и метаэвристические методы и алгоритмы, реализованные в многочисленных коммерческих и некоммерческих пакетах программ смешанного и целочисленного программирования.

5. На основании доказанных утверждений предложены способ и эффективные алгоритмы построения справочной матрицы объединения мелкопартионных потоков, что позволяет определять узлы объединения и объединенные потоки для всех корреспондирующихся пар в многопродуктовой сети. Доказана NP-полнота задачи оптимизации упаковки, которая используется при сортировке и упаковке мелкопартионных грузов в контейнеры в транспортных сетях или объединении сообщений в виртуальные контейнеры в сетях передачи данных. Получила дальнейшее развитие математическая модель задачи оптимизации упаковки с функциями затрат, разработаны метод и ряд эвристических алгоритмов ее решения и доказана сходимость алгоритмов к локальному минимуму за полиномиальное время. Получены оценки временной сложности алгоритмов. Проведенные экспериментальные исследования разработанных алгоритмов показали их вычислительную эффективность при решении практических задач на сетях большой размерности.

6. Доказано, что на этапе решения задачи упаковки нет необходимости решать дискретную задачу выбора пропускных способностей дуг, а нелинейная функция транспортных затрат может быть заменена на функцию удельной стоимости транспортировки транспортных блоков от расстояния и грузоподъемности транспортных средств или пропускной способности линий связи. Доказательство подтверждается результатами числового моделирования. Показано, что задача выбора пропускных способностей может быть решена независимо после решения задачи упаковки и использоваться при решении задачи распределения и маршрутизации транспортных блоков.

7. Доказано, что задача выбора пропускных способностей дуг, сформулированная как задача распознавания, является NP-полной. Приведено полиномиальное преобразование исходной постановки задачи к задаче о ранце с мультивыбором и булевыми переменными, для решения которой предложено два псевдополиномиальных алгоритма, основанных на аппроксимации дискретных функций затрат линейными и на методе последовательного анализа вариантов.

8. Предложена математическая модель и эвристический метод решения задачи выбора иерархической структуры магистральной сети, основанный на применении алгоритмов упаковки и распределения потоков в зональных сетях. Экспериментально показано, что выбор структуры сети слабо зависит от пропускных способностей ее дуг и их можно не учитывать при решении задачи.

9. Получила дальнейшее развитие математическая модель задачи распределения и маршрутизации потоков транспортных блоков с упакованными мелкопартионными корреспонденциями. Предложен метод сведения решения исходной нелинейной многоэкстремальной дискретной задачи к последовательному решению некоторой совокупности линейных многомерных задач о ранце со связывающими ограничениями и ограничениями на время доставки мелкопартионных корреспонденций получателю и среднее время задержки потоков. На основе эффективных структур представления данных, разработаны методы и полиномиальные по сложности алгоритмы решения преобразованной задачи, которые могут работать в интерактивном режиме, и допускают возможность

параметрического задания правых частей всех ограничений и их ослабления в процессе решения при невозможности распределения потоков. Получены оценки временной сложности алгоритмов. Доказано, что при формулировке исходной задачи в виде задачи с заданными тарифами на дугах и в узлах, она может быть за полиномиальное время преобразована к задаче целочисленного линейного программирования, для решения которой можно применять известные методы и пакеты программ целочисленного программирования.

10. Разработана улучшенная версия ранее предложенного автором алгоритма построения всех кратчайших путей в сети по лексикографическому критерию: минимум дуг в пути; минимум длины пути. Доказано, что время работы нового алгоритма может быть сокращено за счет применения абстрактных типов данных и уменьшения просмотра узлов во внутренних циклах. Эмпирически показано, что улучшенный алгоритм на разреженных сетях работает быстрее ранее предложенного, и при увеличении плотности сети на несколько порядков превышает по скорости работы алгоритм Флойда, модифицированный для построения двухкритериальных путей.

11. Разработана эффективная реализация алгоритма Краскала для нахождения остовного дерева минимального (максимального) веса для связного неориентированного графа с практической временной сложностью $O(e)$, где e — число ребер графа. На основании проведенных экспериментальных исследований получена оценка сравнимости быстродействия предложенного алгоритма с алгоритмом Прима, из которой следует, что для связных графов предложенный алгоритм лучше алгоритма Прима при $e \le 0{,}27v^2$, где v — число вершин графа. Экспериментальное сравнение быстродействия алгоритма Краскала со списочными и древовидными структурами данных показало, что алгоритмы со списочными структурами на практике работают быстрее, чем алгоритмы с древовидными структурами как на связных, так и на несвязных графах.

12. Предложена методика получения исходных данных для построения динамических моделей поэтапного развития узлов и транспортных маршрутов магистральной сети, которая основана на решении задач текущего планирования.

Методику планируется использовать для построения детерминированных и стохастических моделей задач перспективного развития транспортной системы при заданных величинах инвестиций по этапам развития и на конечный период планирования с учетом степени риска их вложения. Разработана методика и схема принятия оперативных решений в ИАС ППР по стабилизации работы магистральной сети при перераспределении потоков в случаях введения директивных ограничений на иерархическую структуру и ресурсы сети, отказа отдельных узлов и линий связи, непредвиденных ситуаций и стихийных бедствий.

13. Разработанные компьютерные технологии и инструментальные программные средства предназначены для математического (компьютерного) моделирования функционирования коммуникационных сетей с дискретными мелкопартионными потоками в инфраструктурных комплексах крупных городов, регионов и страны в целом. Они могут быть использованы для решения задач текущего планирования и управления в транспортных сетях, магистральных сетях передачи данных, сотовой и почтовой связи и др., а также для решения задач оперативного управления и перспективного развития и позволяют повысить расчетную эффективность функционирования проектируемых коммуникационных сетей. Разработанный инструментарий может быть также использован для моделирования и оптимизации функционирования традиционных логистических производственных и транспортно-складских систем, включающих узлы поставщиков сырья, производства товаров, склады и конечных потребителей.

Кроме того, полученные результаты работы могут быть полезны в учебном процессе при подготовке специалистов в области разработки и проектирования сложных распределенных многопродуктовых сетей с дискретными потоками.

Результаты работы позволяют наметить направления дальнейших исследований: разработку динамических детерминированных и стохастических моделей, методов и алгоритмов для решения задач перспективного развития иерархических многопродуктовых сетей с ограничениями на ресурсы и возможности их освоения с учетом степени риска вложения инвестиций; развитие методологии исследования устойчивости, реоптимизации, постоптимального анализа получаемых решений задач текущего и долгосрочного планирования и параметрического решения этих задач.

СПИСОК ЛИТЕРАТУРЫ

1. Адельсон-Вельский Г.М. Потоковые алгоритмы / Г.М. Адельсон-Вельский, Е.А. Диниц, А.В. Карзанов. — М.: Наука, 1975. — 119 с.

2. Алексеев В.Е. Графы. Модели вычислений. Структуры Данных / В.Е. Алексеев, В.А. Таланов: Учебник. — Нижний Новгород: Изд-во ННГУ, 2005. — 307 с.

3. Андрианов В.Ю. Инфраструктура пространственных данных / В.Ю. Андрианов // ArcReview. — 2006. — № 2. — http://www.datap.ru/Arcrev/Number_37/1_SDI.html.

4. Ахо А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. — М.: Мир, 1979. — 586 с.

5. Ахо А. Структуры данных и алгоритмы / А. Ахо, Дж. Хопхрофт, Дж. Ульман: Пер. с англ.: Уч. пос. — М.: Издательский дом "Вильяме", 2000. — 384 с.

6. Бертсекас Д. Сети передачи данных / Д. Бертсекас, Р. Галлагер: Пер. с англ. — М.: Мир, 1989. — 544 с.

7. Бронштейн Е.М. Детерминированные оптимизационные задачи транспортной логистики / Е.М. Бронштейн, Т.А. Заико // Автоматика и телемеханика. — 2010. — № 10. — С. 133-147.

8. Бруно Дж.П., Грэхем Р.П. и др. Теория расписаний и вычислительные машины / Под ред. Э.Г. Коффмана. — М.: Наука, 1984. — 334 с.

9. Бушуева С.Ф. Результаты конкурса «Транспорт-81» / С.Ф. Бушуева и др. // Экономика и матем. методы. — М., 1983. — Т. 19, вып. 1. — С. 176-181.

10. Васильева Е.М. Нелинейные транспортные задачи на сетях / Е.М. Васильева, Б.Ю. Левит, В.Н. Лившиц. — М.: Финансы и статистика, 1981. — 104 с.

11. Васянин В.А. Задача распределения и маршрутизации транспортных блоков со смешанными вложениями и ее декомпозиция / В.А. Васянин // Проблемы управления и информатики. — 2015. — № 1. — С. 144-156.

12. Васянин В.А. Автоматизация процессов принятия решений в многопродуктовых коммуникационных сетях с мелкопартионными дискретными потоками / В.А. Васянин, А.Н. Трофимчук // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2010. —Вип. 5. — С. 172-213.

13. Васянин В.А. Задача выбора иерархической структуры многопродуктовой коммуникационной сети с мелкопартионными дискретными потоками / В.А. Васянин, А.Н. Трофимчук // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2012. — Вип. 10. — С. 182-204.

14. Васянин В.А. Обобщенная задача упаковки и распределения мелкопартионных потоков в многопродуктовых иерархических сетях и ее последовательная декомпозиция / В.А. Васянин // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2012. — Вип. 11. — С. 136-154.

15. Васянин В.А. Линейные целочисленные модели распределения потоков в задачах проектирования и анализа многопродуктовых коммуникационных сетей / В.А. Васянин, А.Н. Трофимчук // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2013. — Вип. 12. — С. 147-165.

16. Васянин В.А. Задачи построения доставочных и сборочных маршрутов перевозки мелкопартионных грузов во внутренних зонах иерархической автотранспортной сети / В.А. Васянин, Л.П. Ушакова // Математичне моделювання в економіці. — 2016. — № 3-4. — С. 102-131.

17. Васянин В.А. Задачи построения комбинированных и раздельных маршрутов перевозки мелкопартионных грузов во внутренних зонах иерархической автотранспортной сети / В.А. Васянин // Математичне моделювання в економіці. — 2017. — № 1-2. — С. 74-92.

18. Васянин В.А. Справочная матрица слияния потоков в задачах оптимизации упаковок на многопродуктовых сетях / В.А. Васянин // Системні дослідження та інформаційні технології. — 2014. — № 3. — С. 42-49.

19. Васянин В.А. Структуры данных в задачах оптимизации упаковок мелкопартионных потоков в коммуникационных сетях / В.А. Васянин // Тези VII Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології 2014», 29-30 Травня 2014 р., Житомир: тези доповідей - Житомир, Україна: Житомирський державний технологічний університет, 2014. — С. 7-10.

20. Васянин В.А. Сравнительная эффективность алгоритмов оптимизации упаковок в мультипотоковых сетях // Дискретные системы управления: Сб. науч. тр. — Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1988. — С. 36-45.

21. Васянин В.А. Выбор пропускных способностей дуг при оптимизации упаковок мелкопартионных потоков в многопродуктовых коммуникационных сетях / В.А. Васянин, А.Н. Трофимчук // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2012. — Вип. 9. — С. 181-200.

22. Васянин В.А. Модели и алгоритмы распределения дискретных многопродуктовых потоков в зональных сетях иерархических структур / В.А. Васянин // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2011. — Вип. 8. — С. 176-190.

23. Васянин В.А. Зональные модели распределения мелкопартионных потоков в иерархических многопродуктовых сетях / В.А. Васянин // XI міжнародна науково-практична конференція (MPZIS-2013), 20-22 Листопад 2013 р., Дніпропетровськ: тези доповідей - Дніпропетровськ, Україна: Дніпропетровський національний університет ім. Олеся Гончара, 2013. — С. 40-41.

24. Васянин В.А. Балансировка матрицы контейнерных потоков в задаче перевозки мелкопартионных грузов / В.А. Васянин, Л.П. Ушакова // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2015. — Вип. 17. — С. 98-115.

25. Васянин В.А. О балансировке контейнеропотоков в транспортных сетях с мелкопартионными грузами / В.А. Васянин, Л.П. Ушакова // Sisteme de transport şi logistică: Materialele Conferinţei Internaţionale, Chişinău, 27-30 octombrie 2015 / red. resp.: Dumitru Solomon; col. red.: Dumitru Gortolomei [et al.]. — Chişinău: Evrica, ATIC, 2015 (Tipogr. AŞM). — P. 347-359. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

26. Васянин В.А. Экономико-математические модели задачи распределения потоков в многопродуктовой коммуникационной сети / В.А. Васянин, А.Н. Трофимчук, Л.П. Ушакова // Математичне моделювання в економіці. — 2016. — № 2. — С. 5-21.

27. Васянин В.А. Алгоритм построения кратчайших путей на сети по ступенчатому критерию / В.А. Васянин, А.И. Савенков // Дискретные и эргатические системы управления: Сб. науч. тр. — Киев, 1983. — С. 40-49. — В надзаг.: АН УССР, Научный совет по проблеме "Кибернетика", институт кибернетики.

28. Васянин В.А. Двухкритериальный лексикографический алгоритм построения всех кратчайших путей в сети / В.А. Васянин // Кибернетика и системный анализ. — 2014. — № 5 . — С. 122-131.

29. Васянин В.А. О вычислительной эффективности одного двухкритериального алгоритма построения кратчайших путей / В.А. Васянин // Materials of the International Scientific - Practical Conference «Information Control Systems and Technologies» (ICST - ODESSA - 2013), October 8 - 10, 2013, Odessa: Proceedings. — Odessa, Ukraine: ESC “ONMU”, 2013. — P. 307-310.

30. Васянин В.А. О вычислительной эффективности одного алгоритма для нахождения остовного леса графа с минимальным (максимальным) весом / В.А. Васянин // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2009. — Вип. 4. — С. 155-169.

31. Васянин В.А. Алгоритм построения минимального остовного леса графа с e – ребрами за время $O(e)$ / В.А. Васянин, Л.П. Ушакова // Sisteme de transport şi logistică: Materialele Conferinţei Internaţionale, Chişinău, 11-13 decembrie 2013 / red. resp.: Dumitru Solomon; col. red.: Dumitru Gortolomei [et al.]. — Chişinău: Evrica, ATIC, 2013 (Tipogr. AŞM). — P. 310-316. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

32. Васянин В.А. Структуры данных и процедуры редукции маршрутов в задачах распределения потоков в коммуникационных сетях / В.А. Васянин, Л.П. Ушакова // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2014. — Вип. 14. — С. 192-205.

33. Васянин В.А. Редукция маршрутов и структуры даннях в задачах распределения потоков / В.А. Васянин, Л.П. Ушакова // Modelare matematică, optimizare şi tehnologii informaţionale: Materialele Conferinţei Internaţionale, Chişinău, 25-28 martie 2014 / red. resp.: Dumitru Solomon; col. red.: Dumitru Gortolomei [et al.]. — Chişinău: Evrica, ATIC, 2014 (Tipogr. AŞM). — P. 82-91. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

34. Васянин В.А. Компьютерное моделирование распределения и маршрутизации дискретных многопродуктовых потоков в коммуникационной сети / В.А. Васянин // УСиМ. — 2016. — № 3. — С. 43-53.

35. Васянин В.А. Пакет программ для решения задач оптимизации распределения дискретных потоков в многопродуктовых сетях большой размерности / В.А. Васянин, А.И. Савенков // Міжнародна наукова конференція «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту» (ISDMCI’2008), 19-23 Травень 2008 р., Євпаторія: тези доповідей - Євпаторія, Україна: ХНТУ, 2008. — Том 2 (частина 1). — С. 41-44.

36. Вирт Н. Алгоритмы и структуры данных / Н. Вирт: Пер. с англ. — М.: Мир, 1989. — 360 с.

37. Вишневский В.М. Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. — М.: Техносфера, 2003. — 512 с.

38. Волкович В.Л. Об одной общей схеме последовательного анализа и отсеивания вариантов / В.Л. Волкович, А.Ф. Волошин // Кибернетика. — 1978. — № 5. — С. 98-105.

39. Габасов Р. Методы линейного программирования. Часть 2. Транспортные задачи / Р. Габасов, Р.М. Кириллова. — Минск: Изд. — Белорусск. ун-та, 1978. — 236 с.

40. Гергель В.П. Теория и практика параллельных вычислений: учебное пособие / В.П. Гергель. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 423 с.

41. Гиндуллин Р.В. Оптимизация маршрута доставки однородного груза от множества производителей множеству потребителей: автореф. дис. … к-та физ.-мат. наук. — Уфа, 2013. — 16 с.

42. Гольдштейн Е.Г. Задачи линейного программирования транспортного типа / Е.Г. Гольдштейн, Д.Б. Юдин. — М.: Наука, 1969. — 374 с.

43. Грибов А.В. Рекурсивное решение транспортных задач линейного программирования / А.В. Грибов // Математика, Механика. Астрономия: Вестн. ЛГУ. — Л., 1978. — Вып. 4. — № 19. — С. 11-19.

44. Гуляницький Л.Ф. Прикладні методи комбінаторної оптимізації: навч. посіб. / Л.Ф. Гуляницький, О.Ю. Мулеса — К.: Видавничо-поліграфічний центр "Київський університет", 2016. — 142 с.

45. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. — М.: Мир, 1982. — 416 с.

46. Диниц Е.А. Алгоритм поразрядного сокращения невязок и транспортные задачи / Е.А. Диниц // Исследования по дискретной математике. — М.: Наука, 1973. — С. 46-57.

47. Диниц Е.А. Экономные алгоритмы нахождения кратчайших путей в сети / Е.А. Диниц // Тр. Всесоюз. НИИ системных исследований. — М., 1978. — Вып. 4. — С. 36-44.

48. Диниц Е.А. Программа расчета кратчайших путей в сети / Е.А. Диниц, А.В. Карзанов // Тр. Всесоюз. НИИ системных исследований. — М., 1978. — Вып. 4. — С. 44-48.

49. Добролюбов В.В. Сетевой алгоритм построения путей с минимальным числом транзитных узлов и с максимальной пропускной способностью / В.В. Добролюбов, В.А. Педяш // Вычислительные средства в технике и системах связи: Сб. науч. тр. — М.: Связь, 1979. — № 4. — С. 129-132.

50. Довгий С.О. Методи прогнозування в системах підтримки прийняття рішень / С.О. Довгий, П.І. Бідюк, О.М. Трофимчук, О.І. Савенков. — К.: Ази-

мут-Україна, 2011. — 608 с.

51. Довгий С.О. Системи підтримки прийняття рішень на основі статистично-ймовірнісних методів / С.О. Довгий, П.І. Бідюк, О.М. Трофимчук. — Київ: Логос, 2014. — 419 с.

52. Емеличев В.А. Устойчивость и регуляризация векторных задач целочисленного линейного программирования / В.А. Емеличев, Д.П. Подкопаев // Дискрет. анализ и исслед. операций. Сер. 2. — 2001. — Т. 8. — № 1. — С. 47-69.

53. Емельянова Т.С. Об одном генетическом алгоритме решения транспортной задачи / Т.С. Емельянова. — Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». — Таганрог: Изд-во ТРТУ, 2007. — № 1 (73) . — С. 65-70.

54. Еремеев А.В. Исследование эволюционных методов решения задач комбинаторной оптимизации: автореф. дис. … д-ра физ.-мат. наук. — Москва, 2013. — 34 с.

55. Задачі оптимального проектування надійних мереж / Н.З. Шор, І.В. Сергієнко, В.П. Шило, П.І. Стецюк та ін. — Київ: Наук. думка, 2005. — 229 с.

56. Зайченко Е.Ю. Комплекс моделей и алгоритмов оптимизации характеристик сетей с технологией MPLS / Е.Ю. Зайченко // Системні дослідження та інформаційні технології. — 2007. — № 4. — С. 58-71.

57. Зайченко О.Ю. Оптимальний вибір пропускних здатностей каналів зв’язку в мережах АТМ / О.Ю. Зайченко // Наук. вісті НТУУ «КПІ». — 2000. — № 6. — С. 48-53.

58. Зайченко Ю.П. Задача проектирования структуры распределенных вычислительных сетей / Ю.П. Зайченко // Автоматика. — 1981. — № 4. — С. 35-44.

59. Зайченко Ю.П. Оптимальный выбор пропускных способностей каналов связи в сети с технологией MPLS / Ю.П. Зайченко, Хамуди Мухаммед Али-Аззам // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — 2005. — № 43. — С. 196-201.

60. Зайченко Ю.П. Структурная оптимизация сетей ЭВМ / Ю.П. Зайченко, Ю.В. Гонта. — Киев: Техника, 1986. — 168 с.

61. Ивохин Е.В. О подходах к решению транспортной задачи с нечеткими ресурсами / Е.В. Ивохин, А.Б.С. Камл // Проблемы управления и информатики. — 2014. — № 5. — С. 122-133.

62. Итоги конкурса программ построения матрицы кратчайших расстояний «Транспорт-83» // Экономика и математические методы. — М., 1985. — Т. XII. — Вып. 3. — С. 565-567.

63. Канторович Л.В. Применение математических методов в вопросах анализа грузопотоков / Л.В. Канторович, М.К. Гавурин // Проблемы повышения эффективности работы транспорта. — М.: Изд-во АН СССР, 1949. — С. 110-138.

64. Карзанов А.В. Справочная для выборки максимального элемента и ее приложения / А.В. Карзанов // Исследования по дискретной оптимизации / Под ред. А. А. Фридмана. — М.: Наука, 1976. — С. 340-359.

65. Ким И.В. Об эффективности алгоритмов решения двухкомпонентных задач линейного программирования / И.В. Ким // Экономика и матем. методы. — М., 1974. — Т. 10, вып. 3. — С. 621-631.

66. Клейнрок Л. Коммуникационные сети. Стохастические потоки и задержки сообщений / Л. Клейнрок. — М.: Наука, 1970. — 256 с.

67. Клейнрок Л. Вычислительные системы с очередями / Л. Клейнрок. — М.: Мир, 1979. — 600 с.

68. Князева Н.А. Использование логических операций для поиска оптимальных путей в сети / Н.А. Князева // II Всесоюзное совещание «Методы и программы решения оптимизационных задач на графах и сетях», 24-26 Август 1982 г., Улан-Удэ: тезисы докладов — Новосибирск, 1982. — Часть 1. Теория, алгоритмы. — С. 85-86.

69. Кнут Д. Искусство программирования для ЭВМ. Т.3 (Сортировка и поиск) / Д. Кнут. — М.: Мир, 1978. — 800 с.

70. Колоколов А.А. Анализ устойчивости некоторых алгоритмов дискретной оптимизации / А.А. Колоколов, М.В. Девятерикова // Автоматика и телемеханика. — 2004. — № 3. — С. 48-54.

71. Комбинаторные методы в потоковых задачах: Сборник трудов. Выпуск 3 / Под ред. А.В. Карзанова; Всесоюз. НИИ системных исследований. — М., 1979. — 160 с.

72. Кормен Т. Алгоритмы: построение и анализ, 2-е издание / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1296 с.

73. Кочетов Ю.А. Методы локального поиска для дискретных задач размещения: автореф. дис. … д-ра физ.-мат. наук, Новосибирск, 2009. — 30 с.

74. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. — М.: Мир, 1978. — 432 с.

75. Кузовлёв Д.И. Итеративный алгоритм для класса оптимизационных задач транспортного типа: автореф. дис. … к-та физ.-мат. наук, Москва, 2013. — 21 с.

76. Лебедева Т.Т. Разные типы устойчивости векторной задачи целочисленной оптимизации: общий подход / Т.Т ЛебедеваТ.И. Сергиенко // Кибернетика и системный анализ. — 2008. — № 3. — С. 142-148.

77. Левит Б.Ю. Нелинейные сетевые транспортные задачи / Б.Ю. Левит, В.Н. Лившиц. — М.: Транспорт, 1972. — 144 с.

78. Левитин А. Алгоритмы: введение в разработку и анализ / А. Левитин: Пер. с англ. — М.: Издательский дом "Вильямс", 2006. — 576 с.

79. Леонтьев В.К. Устойчивость в линейных дискретных задачах / В.К. Леонтьев. — В кн.: Пробл. кибернетики. Вып. 35. М.: Наука, 1979. — С. 169-184.

80. Леонтьев В.К. Качественное исследование траекторных задач / В.К. Леонтьев, Э.Н. Гордеев // Кибернетика. — 1986. — № 5. — С. 82-90.

81. Макконелл Дж. Основы современных алгоритмов. 2 е дополненное издание / Дж. Макконелл. — Москва: Техносфера, 2004. — 368с.

82. Малашенко Ю.Е. Нормативный подход к анализу многопродуктовых сетей / Ю.Е. Малашенко // Изв. АН СССР. Техн. кибернетика. — 1988. — № 3. — С. 117-122.

83. Малашенко Ю.Е. Обобщенная задача анализа многопродуктовой сети / Ю.Е. Малашенко, Н.М. Новикова // Изв. АН СССР. Техн. кибернетика. — 1989. — № 4. — С. 137-152.

84. Малашенко Ю.Е. Математические модели анализа потоковых сетевых систем / Ю.Е. Малашенко. — М.: ВЦ РАН, 1993. — 56 с.

85. Мальковский С.И. Решение нелинейных транспортных задач методом роя частиц / С.И. Мальковский, В.В. Пересветов // Моделирование систем. — 2012. — № 2, (32) С. — 54-64.

86. Михалевич В.С. Численные решения многовариантных задач по методу последовательного анализа вариантов / В.С. Михалевич, Н.З. Шор. — В кн.: Научно-методические материалы экономико-математического семинара. — М., 1962. — Вып.1. — С. 15-42. — (Ротапринт / АН СССР, ЛЭМИ).

87. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. I. / В.С. Михалевич // Кибернетика. — 1965. — № 1. — С. 45-55.

88. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. II. / В.С. Михалевич. — Там же. — 1965. — № 2. — С. 85-89.

89. Михалевич В.С. Алгоритмы последовательного анализа и отсеивания вариантов в задачах дискретной оптимизации / В.С. Михалевич, В.Л. Волкович, А.Ф. Волошин, Ю.М. Поздняков // Кибернетика. — 1980. — № 3. — С. 76-85.

90. Михалевич В.С. Вычислительные методы исследования и проектирования сложных систем / В.С. Михалевич, В.Л. Волкович. — М.: Наука, 1982. — 286 с.

91. Михалевич В.С. Оптимальные задачи производственно-транспортного планирования. Модели, методы и алгоритмы / В.С. Михалевич, В.А. Трубин, Н.З. Шор. — М.: Наука, 1986. — 264 с.

92. Михайлюк В.О. Оцінки складності постоптимального аналізу та наближені алгоритми реоптимізації для задач дискретного програмування: автореф. дис. ... д-ра фіз.-мат. наук: 01.05.01 / В.О. Михайлюк; НАН України, Ін-т кібернетики ім. В.М. Глушкова. — К., 2014. — 36 с.

93. Неверов Г.А. Эффективный метод решения задачи о кратчайшем пути / Г.А. Неверов // II Всесоюзное совещание «Методы и программы решения оптимизационных задач на графах и сетях», 24-26 Август 1982 г., Улан-Удэ: тезисы докладов. — Новосибирск, 1982. — Часть 2. Теория, алгоритмы. — С.109-110.

94. Панишев А.В. Модели и методы оптимизации замкнутих маршрутов на транспортной сети: монографія / А.В. Панишев, А.В. Морозов. — Житомир: ЖГТУ, 2014. — 316 с.

95. Панюков А.В. Техника программной реализации потоковых алгоритмов / А.В. Панюков, В.А. Телегин // Вестник ЮУрГУ. — 2008. — № 27(127). — С. 78-99.

96. Папернов Б.А. Реализуемость многопродуктовых потоков / Б.А. Папернов // Исследования по дискретной оптимизации. — М.: Наука, 1976. — С. 230-261.

97. Перцовский А.К. Адаптивные модели и алгоритмы маршрутизации: автореф. дис. … к-та физ.-мат. наук, Санкт-Петербург, 2013. — 16 с.

98. Погорілий С.Д. Формування узагальнених паралельних схем алгоритму Флойда-Уоршала / С.Д. Погорілий, В.А. Мар'яновський, Ю.В. Бойко, Д.Ю. Вітель // Системні дослідження та інформаційні технології. — 2010. — №1. — С. 52-68.

99. Пожидаев М.С. Алгоритмы решения задачи маршрутизации транспорта: автореф. дис. … к-та техн. наук, Томск, 2010. — 19 с.

100. Прыгичев А. Н. Эффективная реализация метода потенциалов для транспортной задачи / А.Н. Прыгичев // Применение математики в экономике. — Л.: Изд-во ЛГУ, 1977. — Вып. 12. — С. 125-131.

101. Рейнгольд Э. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део.: Пер. с англ. — М.: Мир, 1980. — 476 с.

102. Свідоцтво про реєстрацію авторського права на твір «Комп'ютерна програма побудови всіх найкоротших шляхів в мережі за двохкрітеріальним лексикографічним упорядкуванням» / заявник і власник В.О. Васянін; А. с. від 20.07.2016 № 66793, Державна служба інтелектуальної власності України; заявка від 25.05.2016 № 67223 про реєстрацію авторського права на твір.

103. Свідоцтво про реєстрацію авторського права на твір «Комп'ютерна програма знаходження остовного лісу графа з мінімальною (максимальною) вагою» / заявник і власник О.М. Трофимчук, В.О. Васянін, Л.П. Ушакова; А. с. від 20.07.2016 № 66792, Державна служба інтелектуальної власності України; заявка від 25.05.2016 № 67222 про реєстрацію авторського права на твір.

104. Свідоцтво про реєстрацію авторського права на твір «Пакет програм для рішення задач оптимізації розподілу дискретних потоків у богатопродуктових мережах» / заявник і власник В.О. Васянін; А. с. від 23.07.2002 № 5978, Україна, Міністерство освіти і науки України, Державний департамент інтелектуальної власності; заявка від 06.06.2002 № 5772 про реєстрацію авторського права на твір.

105. Свідоцтво про реєстрацію авторського права на твір «Комп'ютерна програма оптимізації ієрархічної структури багатопродуктової комунікаційної мережі з дискретними потоками» / заявник і власник О.М. Трофимчук, В.О. Васянін, Л.П. Ушакова; А. с. від 20.07.2016 № 66791, Державна служба інтелекту-

альної власності України; заявка від 25.05.2016 № 67221 про реєстрацію авторського права на твір.

106. Свідоцтво про реєстрацію авторського права на твір «Комп'ютерна програма оптимізації розподілу та маршрутизації дискретних потоків в багатопродуктової комунікаційної мережі» / заявник і власник В.О. Васянін; А. с. від 20.07.2016 № 66794, Державна служба інтелектуальної власності України; заявка від 25.05.2016 № 67224 про реєстрацію авторського права на твір.

107. Седжвик Р. Фундаментальные алгоритмы на С++. Алгоритмы на графах: Пер. с англ. / Р. Седжвик. — СПб: ООО «ДиаСофтЮП», 2002. — 496 с.

108. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации / И.В. Сергиенко. — Киев: Наук. думка,1988. — 472 с.

109. Сергиенко И.В. Исследование устойчивости и параметрический анализ дискретных оптимизационных задач / И.В. Сергиенко, Л.Н. Козерацкая, Т.Т. Лебедева. — Киев: Наук. думка, 1995. — 170 с.

110. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации. Проблемы, методы решения, исследования / И.В. Сергиенко, В.П. Шило. — Киев: Наук. думка, 2003. — 261 с.

111. Соломон Д.И. Дробное программирование и недифференцируемая оптимизация / Д.И. Соломон. — Кишинев: Эврика, 2010. — 556 с.

112. Трофимчук А.Н. Оптимизация схемы сортировки мелкопартионных потоков в узлах многопродуктовой сети [Электронный ресурс]: 7.14 / А.Н. Трофимчук, В.А. Васянин // Информационные технологии в управлении сложными системами - 2013 [Электронный ресурс]: научная конференция (19-20 июня 2013 г.): сборник докладов. Секция 7 / под редакцией академика НАН Украины В.В. Пилипенко. — Днепропетровск: Институт технической механики Национальной академии наук Украины и Государственного космического агентства Украины, 2013. — Режим доступа к сборнику http://www.itm.dp.ua/.

113. Трофимчук А.Н. Организация управления и принятия решений в распределенных иерархических многопродуктовых сетях с мелкопартионными дискретными потоками / А.Н. Трофимчук, В.А. Васянин // XI міжнародна науково-практична конференція (MPZIS-2013), 20-22 Листопад 2013 р., Дніпропетровськ: тези доповідей - Дніпропетровськ, Україна: Дніпропетровський національний університет ім. Олеся Гончара, 2013. — С. 248-249.

114. Трофимчук А.Н. Оптимизация процессов обработки и распределения потоков мелкопартионных грузов в иерархической многопродуктовой транспортной сети / А.Н. Трофимчук, В.А. Васянин // Sisteme de transport şi logistică:

Materialele Conferinţei Internaţionale, Chişinău, 11-13 decembrie 2013 / red. resp.: Dumitru Solomon; col. red.: Dumitru Gortolomei [et al.]. — Chişinău: Evrica, ATIC, 2013 (Tipogr. AŞM). — P. 317-331. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

115. Трофимчук А.Н. Задачи текущего планирования процессов обработки и распределения мелкопартионных потоков в коммуникационных сетях в автоматизированной информационно-аналитической системе поддержки принятия решений / А.Н. Трофимчук, В.А. Васянин // VIII Всероссийская научная конференция с международным участием «Математическое моделирование развивающейся экономики, экологии и технологий (ЭКОМОД-2014)», посвященная 80-летию академика А.А. Петрова и 100-летию академика Г.С. Поспелова, 21-24 октября 2014 г., Москва, ВЦ РАН: тезисы докладов, 2014. — С. 84.

116. Трофимчук А.Н. Информационные технологии поддержки принятия решений в коммуникационных сетях с дискретными потоками / А.Н. Трофимчук, В.А. Васянин // 14 Міжнародна науково-практична конференція «Сучасні інформаційні технології управління екологічною безпекою, природокористуванням, заходами в надзвичайних ситуаціях», Київ, Пуща-Водиця, 5 – 9 жовтня 2015 р. — Київ, 2015. — С. 64-70. [0,35 усл. печ.лист.]

117. Трофимчук А.Н. Линейные модели распределения целочисленных многопродуктовых потоков в коммуникационных сетях / А.Н. Трофимчук, В.А. Васянин // Materials of the International Scientific - Practical Conference «Information Control Systems and Technologies» (ICST-ODESSA-2014), September 23-25, 2014, Odessa: Proceedings. — Odessa, Ukraine: ESC “ONMU”, 2014. — P. 207-210.

118. Трофимчук А.Н. О сложности одной задачи оптимизации упаковок / А.Н. Трофимчук, В.А. Васянин, В.Н. Кузьменко // Кибернетика и системный анализ. — 2016. — Том 52. — № 1. — С. 83-92.

119. Трофимчук А.Н. Алгоритмы оптимизации упаковок мелкопартионных корреспонденций в коммуникационных сетях / А.Н. Трофимчук, В.А. Васянин, В.Н. Кузьменко // Кибернетика и системный анализ. — 2016. — Том 52. — № 2. — С. 93-106.

120. Трофимчук А.Н. Об одном подходе к декомпозиции задачи распределения и маршрутизации мелкопартионных потоков в многопродуктовой сети / А.Н. Трофимчук, В.А. Васянин // 16-th International conference on System Analysis and Information Technologies (SAIT2014), May 26 - 30, 2014, Kyiv: Proceedings. — Kyiv, Ukraine: ESC “IASA” NTUU “KPI”, 2014. — P. 162.

121. Трофимчук А.Н. Задача оптимизации упаковок мелкопартионных грузов в многопродуктовой транспортной сети с дискретными пропускными спо-

собностями дуг / А.Н. Трофимчук, В.А. Васянин // Materials of the International Scientific - Practical Conference «Information Control Systems and Technologies» (ICST - ODESSA - 2013), October 8 - 10, 2013, Odessa: Proceedings. — Odessa, Ukraine: ESC “ONMU”, 2013. — P. 269-272.

122. Трофимчук А.Н. Оптимизация структуры многопродуктовой сети с дискретними мелкопартионными потоками / А.Н. Трофимчук, В.А. Васянин // 15-th International conference on System Analysis and Information Technologies (SAIT2013), May 27-31, 2013, Kyiv: Proceedings. — Kyiv, Ukraine: ESC “IASA” NTUU “KPI”, 2013. — P. 205-206.

123. Трофимчук А.Н. Задача выбора пропускных способностей дуг при распределении потоков в многопродуктовых коммуникационных сетях [Электронный ресурс]: / А.Н. Трофимчук, В.А. Васянин // Відділ методів негладкої оптимізації - 2017: науковий семінар (31 січня 2017 р.). — Київ: Институт кібернетики ім. В.М. Глушкова НАН України, 2017. — Режим доступа http://www.icyb.kiev.ua/s/412/ua/.

124. Трофимчук А.Н. Задача распределения потоков в многопродуктовой сети с заданными тарифами на дугах / А.Н. Трофимчук, Л.П. Ушакова, В.А. Васянин // Тези VIII Міжнародної науково-технічної конференції «Інформаційно-комп’ютерні технології 2016», 22-23 квітня 2016 р., Житомир: тези доповідей - Житомир, Україна: Житомирський державний технологічний університет, 2016. — С.

125. Трофимчук А.Н. Время работы алгоритма Краскала с древовидной и списочной структурой данных / А.Н. Трофимчук, В.А. Васянин // Системні дослідження та інформаційні технології. — 2015. — № 3. — С. 48-61.

126. Трофимчук А.Н. О вычислительной эффективности алгоритма Краскала со списочной и древовидной структурой данных / А. Н. Трофимчук, Л. П. Ушакова, В. А. Васянин // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19-21 берез. 2015 р.). — Полтава: ПУЕТ, 2015. — С. 349-350. http://dspace.puet.edu.ua/handle/123456789/2458

127. Трофимчук А.Н. Методика решения задачи оптимизации упаковок для управления перспективным развитием узлов коммуникационной сети / А.Н. Трофимчук, В.А. Васянин // Кибернетика и системный анализ. — 2014. — № 4. — С. 88-99.

128. Трофимчук А.Н. О перспективном развитии узлов и оперативном перераспределении потоков в многопродуктовой транспортной сети / А.Н. Трофимчук, В.А. Васянин // Modelare matematică, optimizare şi tehnologii informaţionale: Materialele Conferinţei Internaţionale, Chişinău, 25-28 martie 2014 /

red. resp.: Dumitru Solomon; col. red.: Dumitru Gortolomei [et al.]. — Chişinău: Evrica, ATIC, 2014 (Tipogr. AŞM). — P. 472-484. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

129. Трофимчук А.Н. Моделирование упаковки, распределения и маршрутизации мелкопартионных потоков в многопродуктовой сети / А.Н. Трофимчук, В.А. Васянин // Проблемы управления и информатики. — 2015. — № 4. — С. 132-146.

130. Трофимчук А.Н. Компьютерная технология моделирования иерархической структуры сети с дискретными потоками / А.Н. Трофимчук, Л.П. Ушакова, В.А. Васянин // Modelare matematică, optimizare şi tehnologii informaţionale: Materialele Conferinţei Internaţionale, Volumul II, Chişinău, 22-25 martie 2016 / red. resp.: Dumitru Solomon; col. red.: Dumitru Lozovanu [et al.]. — Chişinău: Evrica, ATIC, 2016 (Tipogr. AŞM). — Vol. 2. — P. 354-365. — Antetit.: Academia de Transporturi, Informatică şi Comunicaţii.

131. Трофимчук А.Н. Компьютерное моделирование иерархической структуры коммуникационной сети с дискретными многопродуктовыми потоками / А.Н. Трофимчук, В.А. Васянин // УСиМ. — 2016. — № 2. — С. 48-57.

132. Хмелев А.В. Алгоритмы локального поиска для задач маршрутизации транспортных средств: дис. … к-та физ.-мат. наук, Новосибирск, 2015. — 119 с.

133. Чернышев С.В. Модели, методы и алгоритмы эффективного решения задачи маршрутизации транспорта на графах больших размерностей: автореф. дис. … к-та физ.-мат. наук, Москва, 2011. — 24 с.

134. Филлипс Д. Методы анализа сетей: Пер. с англ. / Д. Филлипс, А. Гарсия-Диас. — М.: Мир, 1984. — 496 с.

135. Филлер М.Ф. Один алгоритм нахождения кратчайших путей / М.Ф. Филлер // Исследования по дискретной оптимизации. — М.: Наука, 1976. — С. 359-364.

136. Aboutahoun A. W. Minimum Cost-Reliability Ratio Path Problem / A. W. Aboutahoun // Life Sci. J. — 2012. — 9(3). — P. 1633-1645.

137. Abraham I. Highway Dimension, Shortest Paths, and Provably Efficient Algorithms / I. Abraham, A. Fiat, A. Goldberg, R. Werneck // in Proceedings of the 21st Annual ACM–SIAM Symposium on Discrete Algorithms (SODA’10), 2010. — P. 782–793.

138. Abraham I. A Hub-Based Labeling Algorithm for Shortest Paths on Road Networks / I. Abraham, D. Delling, A. Goldberg, R. Werneck // Technical Report MSR-TR-2010-165, Microsoft Research, 2010. — 26 p.

139. Abraham I. Highway Dimension, Shortest Paths, and Provably Efficient Algorithms / I. Abraham, A. Fiat, A. Goldberg, R. Werneck // In SODA. — 2010. — P. 782-793.

140. Abraham I. Hierarchical Hub Labelings for Shortest Paths / I. Abraham, D. Delling, A. Goldberg, R. Werneck // In Proc. ESA, LNCS 7501, Springer, 2012. — P. 24-35.

141. Abraham I. Alternative Routes in Road Networks / I. Abraham, D. Delling, A. Goldberg, R. Werneck // In ACM Journal of Experimental Algorithmics. — 2013. — Vol. 18. — N 1. — P. 1–17.

142. Adlakha V. Managing cost uncertainties in transportation and assignment problems / V. Adlakha, H. Arsham // Journal of applied mathematics & decision sciences. — 1998. — Vol. 2. — N 1. — P. 65-104.

143. Aho A. The design and analysis of computer algorithms / A. Aho, J. Hopcroft, J. Ullman. — Addison-Wesley, 1974. — 480 p.

144. Ahuja R.K. Network flows: theory, algorithms, and applications / R.K. Ahuja, T.L. Magnanti, J.B. Orlin. — Prentice-Hall, Inc. Upper Saddle River, New Jersey, 1993. — 846 p.

145. Ahuja R. Faster algorithms for the shortest path problem / R. Ahuja, K. Mehlhorn, J. Orlin, R. Tarjan // Journal of ACM. — 1990. — Vol. 3(2). — N 2. — P. 213-223.

146. Ai T.J. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery / T.J. Ai, V. Kachitvichyanukul // Computers & Operations Research. — 2009. — Vol. 36. — Issue 5. — P. 1693-1702.

147. Aleman R.E. A tabu search with vocabulary building approach for the vehicle routing problem with split demands / R.E. Aleman, R.R. Hill // Int. J. Metaheuristics. — 2010. — Vol. 1. — Issue 1. — P. 55-80.

148. Alon N. On the exponent of the all pairs shortest path problem / N. Alon, Z. Galil, O. Margalit // Journal of Computer and System Sciences. — 1997. — Vol. 54. — P. 255-262.

149. Altiparmak F. An adaptive tabu-simulated annealing for concave cost transportation problems / F. Altiparmak, I. Karaoglan // Journal of the Operational

Research Society. — 2008. — Vol. 59. — N 3. — P. 331-341.

150. Angelelli E. The Vehicle Routing Problem with Time Windows and Simultaneous Pick-up and Delivery / E. Angelelli, R. Mansini // In Quantitative Approaches to Distribution Logistics and Supply Chain Management, Springer, Berlin-Heidelberg, 2002. — Vol. 519. — P. 249-267.

151. Archetti C. Worst-case analysis for split delivery vehicle routing problems / C. Archetti, M. Savelsbergh, M.G. Speranza // Transportation Science. — 2006. — Vol. 40. — Issue 2. — P. 226-234.

152. Archetti C. To split or not to split: That is the question / C. Archetti, M. Savelsbergh, M.G. Speranza // Transportation Research Part E: Logistics and Transportation Review. — 2008. — Vol. 44. — Issue 1. — P. 114-123.

153. Archetti C. The split delivery vehicle routing problem: A survey / C. Archetti, M.G. Speranza // In Golden B.L., Raghavan S., Wasil E.A. The Vehicle Routing Problem: Latest Advances and New Challenges, New York: Springer-Verlag. — 2008. — P. 103-122.

154. Archetti C. A tabu search algorithm for the split delivery vehicle routing problem / C. Archetti, A. Hertz, M.G. Speranza // Transportation Science. — 2006. — Vol. 40. — Issue 1. — P. 64-73.

155. Archetti C. A column generation approach for the split delivery vehicle routing problem / C. Archetti, C. Bianchessi, M.G. Speranza // Networks. — 2011. — Vol. 58. — Issue 4. — P. 241-254.

156. Armacost A. A computational comparison of the network simplex method with the dual affine scaling method / A. Armacost, S. Mehrotra // Opsearh. — 1991. — Vol. 28. — N 1. — P. 18-35.

157. Arora S. Proof verification and the hardness of approximation problems / S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy // Journal of the ACM. — 1998. — **45**, N 3. — P. 501-555.

158. Arora S. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques / S. Arora, K. Jansen, J. Rolim, A. Sahai: 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NY, USA, August 24-26, 2003, Springer, 2003.— 411p.

159. Asano Y. Practical efficiency of the linear-time algorithm for the single source shortest path problem / Y. Asano, H. Imai // Journal of the Operations Re-

search Society of Japan. — 2000. — Vol. 43. — P. 431-447.

160. Assad A.A. Modeling of Rail networks: Toward a routing / makeup model / A.A. Assad // Transportation Research. — 1980. — **148**, N — 1-2. — P. 101-114.

161. Assad A.A. Multicommodity network flows. A survey / A.A. Assad // Networks. — 1978. — N 8. — P. 37-91.

162. Ausiello G. Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties / G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. — Berlin: Springer-Verlag, 1999. — 524 p.

163. Ausiello G. Complexity and approximation in reoptimization / G. Ausiello, V. Bonifaci, B. Escoffier // Computability in Context: Computation and Logic in the Real World / Ed. S. Barry Cooper and Andrea Sorbi. — London: Imperial College Press. — 2011. — P. 101-130.

164. Avci M. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery / M. Avci, S. Topaloglu // Expert Systems with Applications. — 2016. — Volume 53. — Issue 1. — P. 160-171.

165. Backhouse R. A comparison of gaussian and gauss-jordan elimination in regular algebra / R. Backhouse, B. Carre // International Journal of Computer Mathematics. — 1982. — Vol. 10. — N 3-4. — P. 311-325.

166. Bader D.A. Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs / D.A. Bader, G. Cong // Journal of Parallel and Distributed Computing. — 2006. — Vol. 66 (11). — P. 1366-1378.

167. Baldacci R. Routing a Heterogeneous Fleet of Vehicles / R. Baldacci, M. Battarra, D. Vigo // Technical Report DEIS OR. INGCE 2007/1, University Bologna, Italy, 2007. — 25 p.

168. Baldacci R. Routing a heterogeneous fleet of vehicles / R. Baldacci, M. Battarra, D. Vigo // In Golden B.L., Raghavan S., Wasil E.A. (Eds.), The vehicle routing problem: Latest advances and new challenges. New York: Springer, 2008. — P. 1-25.

169. Baldacci R. Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs / R. Baldacci, M. Battarra, D. Vigo // Networks. — 2009. — Vol. 54. — Issue 4. — P. 178-189.

170. Baldacci R. A unified exact method for solving different classes of vehicle routing problems / R. Baldacci, A. Mingozzi // Mathematical Programming, Series A.

— 2009. — Vol. 120. — Issue 2. — P. 347-380.

171. Baldacci R. An exact solution framework for a broad class of vehicle routing problems / R. Baldacci, E. Bartolini, A. Mingozzi, R. Roberti // Computational Management Science. — 2010. — Vol. 7. — Issue 3. — P. 229-268.

172. Baldacci R. Exact algorithms for routing problems under vehicle capacity constraints / R. Baldacci, P. Toth, D. Vigo // Annals of Operations Research. — 2010. — Vol. 175. — Issue 1. — P. 213-245.

173. Balinski M. On an integer program for a delivery problem / M. Balinski, R. Quandt // Operations Research. — 1964. — Vol. 12. — P. 300-304.

174. Barnhart C. Using branch-and-price-and-cut to solve origin destination integer multicommodity flow problems / C. Barnhart, C.A. Hane, P.H. Vange // Operations Research. — 2000. — Vol. 48. — N. 2. — P. 318-326.

175. Barnhart C. Multicommodity Flow Problems / C. Barnhart, N. Krishnan, P.H. Vange: In Encyclopedia of Optimization: Second Edition, C.A. Floudas and P.M. Pardalos (Eds.). — Springer, New York, 2009. — P. 2354-2362.

176. Barr R.S. An improved version of the Out-of-Kilter method and a comparative study of computer codes / R.S. Barr, F. Glover, D. Klingman // Math. Prog. — 1974. — Vol. 7. — N 1. — P. 60-86.

177. Bean J.C. Genetic algorithms and random keys for sequencing and optimization / J.C. Bean // INFORMS Journal on Computing. — 1994. — Vol. 6. — Issue 2. — P. 154-160.

178. Belenguer J. A lower bound for the split delivery vehicle routing problem / J. Belenguer, M. Martinez, E. Mota // Operations Research. — 2000. — Vol. 48. — Issue 5. — P. 801-810.

179. Belfiore P. Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries / P. Belfiore, H.T.Y. Yoshizaki // Computers & Industrial Engineering. — 2013. — Vol. 64. — Issue 2. — P. 589-601.

180. Bellman R. On a routing problem / R. Bellman // Quarterly of Applied Mathematics. — 1958. — Vol. 16. — P. 87-90.

181. Berbeglia G. Static pickup and delivery problems: a classification scheme and survey / G. Berbeglia, J-F. Cordeau, I. Gribkovskaia, G. Laporte // TOP: Operations Research & Decision Theory, Springer-Verlag, 2007. — Vol. 15. — Issue 1. — P. 1-31.

182. Berbotto L. A randomized granular tabu search heurstic for the split delivery vehicle routing problem / L. Berbotto, S. Garcia, F.G. Nogales // Annals of Operations Research. — 2014. — Vol. 222. — P. 153-173.

183. Bertsekas D. A simple and fast label correcting algorithm for shortest paths / D. Bertsekas // Networks. — 1993. — Vol. 23. — P. 703-709.

184. Bertsekas D. Network optimization: continuous and discrete models / D. Bertsekas. — P.O. Box 391, Belmont, MA 02178-9998: Athena Scientific, 1998. — 585 p.

185. Bertsekas D. Polynomial auction algorithms for shortest paths / D. Bertsekas, S. Pallottino, M. Scutellμa // Computational Optimization and Applications. — 1995. — Vol. 4. — P. 99-125.

186. Bianchessi N. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery / N. Bianchessi, G. Righini // Computers & Operations Research. — 2007. — Vol. 34. — Issue 2. — P. 578-594.

187. Blum C. Metaheuristics in combinatorial optimization: Overview and conceptual comparison / C. Blum, A. Roli // ACM Computing Surveys (CSUR). — 2003. — **35**. — N 3. — P. 268-308.

188. Boost.org. Boost C++ Libraries. www.boost.org.

189. Boruvka O. O jistém problému minimálním / O. Boruvka // Pràca Moravské Prirodovedecké Spolecnosti. — 1926. — Vol. 3. — P. 37-58.

190. Boudia M. An effective memetic algorithm with population management for the split delivery vehicle routing problem / M. Boudia, C. Prins, M. Reghioui // Hybrid Metaheuristics, Lecture Notes in Computer Science. — 2007. — Vol. 4771. — P. 16-30.

191. Braekers K. The vehicle routing problem: State of the art classification and review / K. Braekers, K. Ramaekers, I. Van Nieuwenhuyse // Computers & Industrial Engineering. — 2015. — In Press.

192. Brandão J. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem / J. Brandão // European Journal of Operational Research. — 2009. — Vol. 195. — Issue 3. — P. 716-728.

193. Brandão J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem / J. Brandão // Computers & Operations Research. — 2011. — Vol. 38. — Issue 1. — P. 140-151.

194. Bräysy O. An Effective Multi-restart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle Routing Problem with Time Windows / O. Bräysy, W. Dullaert, G. Hasle, D. Mester, M. Gendreau // Transportation Science. — 2008. — Vol. 42. — Issue 3. — P. 371-386.

195. Burke E.K. Search Methodologies. Introductory Tutorials in Optimization and Decision Support Techniques / E.K. Burke, G. Kendall. — New York: Springer, 2014. — 705 p.

196. Burton D. On the inverse shortest path problem / D. Burton // Department de Mathematique, Faculty des Sciences, Faculties Universities Notre-Dame de la Paix de Namur, 1993. — 119 p.

197. Carre B. A matrix factorization method for finding optimal paths through networks / B. Carre // in I.E.E. Conference Publication: Computer-Aided Design, 1969. — N 51. — P. 388-397.

198. Carre B. An algebra for network routing problems / B. Carre // Journal of Institute of Mathematics and Its Applications. — 1971. — Vol. 7. — P. 273-294.

199. Çatay B. A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery / B. Çatay // Expert Systems with Applications. — 2010. —Vol. 37. — Issue 10. — P. 6809-6817.

200. Cerulli R. Graph collapsing in shortest path auction algorithms / R. Cerulli, P. Festa, G. Raiconi // Computational Optimization and Applications. — 2001. — Vol. 18. — N 3. — P. 199-220.

201. Charnes A. Past, present and future of large scale transshipment computer codes and applications / A. Charnes, D. Karney, D. Klingman, J. Stutz, F. Glover // Comput. & Ops. Res. — 1975. — Vol. 2. — P. 71-81.

202. Chazelle B. A minimum spanning tree algorithm with inverse-Ackermann type complexity / B. Chazelle // Journal of the ACM. — 2000. — Vol. 47. — P. 1028-1047.

203. Chazelle B. The Soft Heap: An Approximate Priority Queue with Optimal Error Rate / B. Chazelle // Journal of the ACM. — 2000. — Vol. 47. — P. 1012-1027.

204. Chen J.F. Vehicle routing problem with simultaneous deliveries and pickups / J.F. Chen, T.H. Wu // Journal of the Operational Research Society. — 2006. — Vol. 57. — Issue 5. — P. 579-587.

205. Chen J.F. Approaches for the vehicle routing problem with simultaneous deliveries and pickups / J.F. Chen // Journal of the Chinese Institute of Industrial Engineers. — 2006. — Vol. 23. — Issue 2. — P. 141-150.

206. Chen S. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results / S. Chen, B. Golden, E. Wasil // Networks. — 2007. — Vol. 49. — Issue 4. — P. 318-329.

207. Cheriton D. Finding Minimum Spanning Trees / D. Cheriton, R.E. Tarjan // J. SIAM Comp. — 1976. — Vol. 5. — N 4. — P. 724-742.

208. Cherkassky B. Shorthest paths algorithms: theory and experimental evaluation / B. Cherkassky, A. Goldberg, T. Radzik // Mathematical Programming. — 1996. — Vol. 73. — P. 129-174.

209. Cherkassky B. Shortest-Path Feasibility Algorithms: An Experimental Evaluation / B. Cherkassky, L. Georgiadis, A. Goldberg, R. Tarjan, R. Werneck // In ACM Journal of Experimental Algorithmics. — 2009. — Vol. 14. — N 2. — P. 2.7:1-2.7:37.

210. Choi E. A column generation approach to the heterogeneous fleet vehicle routing problem / E. Choi, D. Tcha // Computers & Operations Research. — 2007. — Vol. 34. — P. 2080-2095.

211. Christofides N. Worst-case analysis of a new heuristic for the travelling salesman problem / N. Christofides // Research Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburg, PA, 1976. — 5 p.

212. Clarke G. Scheduling of vehicles from a central depot to a number of delivery points / G. Clarke, J.W. Wright // Operations Research. — 1964. — Vol. 12. — P. 568-581.

213. Coello C. An updated survey of GA-based multiobjective optimization techniques / C. Coello // ACM Computing Surveys. — 2000. — Vol. 32. — P. 109-143.

214. Cohn A. Integration of the Load-Matching and Routing Problem with Equipment Balancing for Small Package Carrier / A. Cohn, S. Root, A. Wang, D. Mohr // Ttransportation science. — 2007. — V. 41. — N. 2. — P. 238-252.

215. Cornelis C. Shortest Paths in Fuzzy Weighted Graphs / C. Cornelis, P. Kesel, E. Kerre // International journal of intelligent systems. — 2004. — Vol. 19. — P. 1051-1068.

216. Cornuéjols G. Tight bounds for Christofides' traveling salesman heuristic / G. Cornuéjols, G.L. Nemhauser // Math Program. —1978. — N. 14. — P. 116-121.

217. Courtois P.J. An Algorithm for the Optimization of Nonbifurcated Flows in Computer Communication Networks / P.J. Courtois, P. Semal // Performance Evaluation. — 1981. — **1**. — P. 139-152.

218. Coutinho-Rodrigues J. An interactive biobjective shortest path approach: searching for unsupported nondominated solutions / J. Coutinho-Rodrigues, J. Climaco, J. Current // Computers & Operations Research. — 1999. — Vol. 26. — P. 789-798.

219. Coutinho-Rodrigues J. A PC-based interactive decision support system for two objective direct delivery problems / J. Coutinho-Rodrigues, J. Climaco, J. Current // Journal of Business Logistics. — 1994. — Vol. 15. — Issue 1. — P. 305-322.

220. Crispim J. Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls / J. Crispim, J. Brandão // Journal of the Operational Research Society. — 2005. — Vol. 56. — Issue 11. — P. 1296-1302.

221. Dantzig G. On the shortest route through a network / G. Dantzig // Management Science. — 1960. — Vol. 6. — P. 187-190.

222. Dantzig G.B. Decomposition Algorithm for linear programming / G.B. Dantzig, Ph. Wolfe // Econometrica. — 1961. — Vol. 29. — N. 4. — P. 767-778.

223. Dantzig G.B. Minimizing the number of tankers to meet a fixed schedule / G.B. Dantzig, D.R. Fulkerson // Naval Research Logistics Quarterly. — 1954. — N. 1. — P. 217-222.

224. Dantzig G. All shortest routes in a graph / G. Dantzig // in Theory of Graphs (International Symposium, Rome, 1966). — New York: Gordon and Breach, 1967. — P. 91-92.

225. Deb K. Multi-Objective Optimization Using Evolutionary Algorithms / K. Deb. - Chichester, UK: J. Wiley & Sons, 2009. — 544 p.

226. De Jaegere N. The Vehicle Routing Problem: State of the Art Classification and Review / N. De Jaegere, M. Defraeyea, I. Van Nieuwenhuysea // International Journal of Combinatorial Optimization Problems and Informatics. — 2014. — Volume 5. — N. 2. — P. 58-73.

227. Dell'Amico M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection / M. Dell'Amico, G. Righini, M. Salani // Transportation Science. — 2006. — Vol. 40. — Issue 2. — P. 235-247.

228. Delling D. Engineering Route Planning Algorithms / D. Delling, P. Sanders, D. Schultes, D. Wagner // in Algorithmics of Large and Complex Networks, ser. Lecture Notes in Computer Science, J. Lerner, D. Wagner, and K. Zweig, Eds. Springer, 2009. — Vol. 5515. — P. 117-139.

229. Delling D. Customizable Route Planning / D. Delling, A. Goldberg, T. Pajor, R. Werneck // In Proc. SEA, LNCS 6630, Springer, 2011. — P. 376-387.

230. Delling D. A Hub-Based Labeling Algorithm for Shortest Paths in Road Networks / I. Abraham, D. Delling, A. Goldberg, R. Werneck // SEA'11 Proceedings of the 10th international conference on Experimental algorithms, Crete, Greece — May 05 - 07, Springer, LNCS, 2011. — P. 230-241.

231. Delling D. Hub Label Compression / D. Delling, A. Goldberg, R. Werneck // In Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13): Springer Verlag, 2013. — P. 1–12.

232. Delling D. PHAST: Hardware-accelerated shortest path trees / D. Delling, A. Goldberg, A. Nowatzyk, R. Werneck // Journal of Parallel and Distributed Computing. — 2013. — Vol. 73. — Issue 7. — P. 940-952.

233. Delling D. Faster Customization of Road Networks / D. Delling, R. Werneck // In Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13): Springer, 2013. — P. 1–12.

234. Delling D. Computing Multimodal Journeys in Practice / D. Delling, J. Dibbelt, T. Pajor, D. Wagner, R. Werneck // in Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13), Rome, Italy, June 5-7, Springer, 2013. — P. 260-271.

235. Deo N. Shortest path algorithms: Taxonomy and annotation / N. Deo, C. Pang // Networks. — 1984. — N 14. — P. 275-323.

236. Dethloff J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up / J. Dethloff // OR Spektrum. — 2001. — Vol. 23. — Issue 1. — P. 79-96.

237. Dial R. Algorithm 360 shortest path forest with topological ordering / R. Dial // Communications of the ACM. — 1969. — Vol. 12. — P. 632-633.

238. Dial R. A computational analysis of alternative algorithms and labeling techniques for finding shortest path trees / R. Dial, F. Glover, D. Karney, D. Klingman // Networks. — 1979. — Vol. 9. — N. 3. — P. 215-248.

239. Díaz-Parra O. A Survey of Transportation Problems / O. Díaz-Parra, J.A. Ruiz-Vanoye, B.B. Loranca, A. Fuentes-Penna, R.A. Barrera-Cámara // Journal of Applied Mathematics. —2014. — Article ID 848129. — 17 p.

240. Dijkstra F.W. A note on two problems in connection with graphs / F.W. Dijkstra // Numerical mathematic. — 1959. — N 1. — P. 269-271.

241. Divoky J. Performance of shortest path algorithms in network flow problems / J. Divoky, M. Hung // Management Science. — 1990. — Vol. 36. — P. 661-673.

242. Dixon B. Verification and Sensitivity Analysis of Minimum Spanning Trees in Linear Time / B. Dixon, M. Rauch, R.E. Tarjan // SIAM Journal on Computing. — 1992. — Vol. 21(6). — P. 1184-1192.

243. Dreyfus S.E. An appraisal of some shortest path algorithms / S.E. Dreyfus // Oper. Res. — 1969. — Vol. 17. — P. 395-412.

244. Dror M. Savings by split delivery routing / M. Dror, P. Trudeau // Transportation Science. — 1989. — Vol. 23. — Issue 2. — P. 141-145.

245. Dror M. Split delivery routing / M. Dror, P. Trudeau // Naval Research Logistics. — 1990. — Vol. 37. — Issue 3. — P. 383-402.

246. Edmonds J. Matroids and the Greedy Algorithm / J. Edmonds // Math. Programming. — 1971. — Vol. 1. — P. 127-136.

247. Ehrgott M. Multiple Criteria Optimization. State of the Art Annotated Bibliographic Surveys / M. Ehrgott, X. Gandibleux. — Boston, MA: Kluwer, 2002. — 496 p.

248. Eksioglu B. The vehicle routing problem: A taxonomic review / B. Eksioglu, A.V. Vural, A. Reisman // Computers & Industrial Engineering. — 2009. — Vol. 57. — N. 4. — P. 1472-1483.

249. Even S. On the complexity of timetable and multicommodity flow problems / S. Even, A. Itai, A. Shamir // SIAM J: Comput. — 1976. — **5**. — P. 691-703.

250. Farbey B.A. The cascade algorithm for finding all shortest distances in a directed graph / B.A. Farbey, A.H. Land, J.D. Murchland // Management Science. — 1967. — Vol. 14. — P. 19-28.

251. Fischer M.J. Efficiency of Equivalence Algorithms / M.J. Fischer // In Complexity of Computer Computations: Miller R.E. and Thatcher J.W. (Eds.), Plenum Press, New York, 1972. — P. 153-168.

252. Floyd R. Algorithm 97, shortest path / R. Floyd // Comm. ACM. — 1962. — Vol. 5. — P. 345.

253. Ford Jr. L. Network flow theory / Jr. L. Ford // Santa Monica, California: The RAND Corp., 1956. — P. 923.

254. Ford Jr. L. Flows in networks / Jr. L. Ford, D. Fulkerson // Princeton, NJ: Princeton University Press, 1962. — 332 p.

255. Fredman M. New bounds on the complexity of the shortest path problems / M. Fredman // SIAM Journal on Computing. — 1976. — Vol. 5. — P. 83-89.

256. Fredman M. Fibonacci heaps and their uses in improved network optimization algorithms / M. Fredman, R. Tarjan // Journal of the ACM. — 1987. — Vol. 34. — N 3. — P. 596-615.

257. Fredman M.L. Trans-dichotomous Algorithms for Minimum Spanning Trees and Shortest Paths / M.L. Fredman, D.E. Willard // Journal of Computer and System Sciences. — 1994. — Vol. 48(3). — P. 533-551.

258. Fukasawa R. Robust branch-and-cut-and-price for the capacitated vehicle routing problem / R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, R.F. Werneck // Mathematical Programming. — 2006. — Vol. 106. — P. 491-511.

259. Gabow H. Faster Scaling Algorithms for Network Problems / H. Gabow, R. Tarjan // SIAM Journal on Computing. — 1989. — **18**. — P. 1013-1036.

260. Gabow H.N. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs / H.N. Gabow, Z. Galil, T. Spencer, R.E. Tarian // Combinatorica. — 1986. — Vol. 6. — P. 109-122.

261. Gabrel V. Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite / V. Gabrel, D. Vanderpooten // European Journal of Operational Research. — 2002. — 139. — P. 533-542.

262. Gajpal Y. An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup / Y. Gajpal, P. Abad // Computers & Operations Research. — 2009. — Vol. 36. — Issue 12. — P. 3215-3223.

263. Galil Z. All pairs shortest paths for graphs with small integer length edges / Z. Galil, O. Margalit // Journal of Computer and System Sciences. — 1997. — Vol. 54. — P. 243-254.

264. Galil Z. All pairs shortest distances for graphs with small integer length edges / Z. Galil, O. Margalit // Information and Computation. — 1997. — Vol. 134. — P. 103-139.

265. Galler B.A. An Improved Equivalence Algoritm / B.A. Galler, M.J. Fischer // Comm. ACM. — 1964. — Vol. 7. — P. 301-303.

266. Gandibleux X. Martins' algorithm revisited for multi-objective shortest path problems with a MaxMin cost function / X. Gandibleux, F. Beugnies, S. Randriamasy // 4OR A Quarterly Journal of Operations Research. — 2006. — Vol. 4. — P. 47-59.

267. Garey M. Computers and Intractability: A Guide to the Theory of NP Completeness / M. Garey, D. Johnson. — San Francisco, CA: W.H. Freeman, 1979.

268. Gavish B. An Algorithm for Optimal Route Selection in SNA Networks / B. Gavish, S.L. Hantler // IEEE Trans. Commun. — 1983. — **COM-31**, — N 10. — P. 1154-1161.

269. Gen M. Multiobjective genetic algorithm for solving network design problem / M. Gen, L. Lin // presented at the 20th Fuzzy Systems Symposium, Kitakyushu, Japan, June 2004.

270. Gendreau M.G. Metaheuristics for the vehicle routing problem / M.G. Gendreau, G. Laporte, J-V. Potvin // Les Cahiers du GERAD, G98-52, Group for Research in Decision Analysis, Montreal, Canada, 1998. — 27 p.

271. Gendreau M. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography / M. Gendreau, J-Y. Potvin, O. Bräysy, G. Hasle, A. Lökketangen // CIRRELT, 2007. — Vol. 27. — 25 p.

272. Gendreau M. A tabu search heuristic for the heterogeneous fleet vehicle routing problem / M. Gendreau, G. Laporte, C. Musaraganyi, E. Taillard // Computers & Operations Research. — 1999. — Vol. 26. — Issue 12. — P. 1153-1173.

273. Gendreau M. New insertion and postoptimization procedures for the traveling salesman problem / M. Gendreau, A. Hertz, G. Laporte // Operations Research. — 1992. — Vol. 40. — Issue 6. — P. 1086-1094.

274. Gheysens F. A Comparison of Techniques for Solving the Fleet Size and Mix Vehicle Routing Problem / F. Gheysens, B.L. Golden, A. Assad // OR Spectrum. — 1984. — N. 6. — P. 207-216.

275. Glover F. Scatter Search and Path Relinking: Advances and Applications / F. Glover, M. Laguna, R. Marti // In Handbook of Metaheuristics: International Series in Operations Research & Management Science. — Kluwer Academic Publish-

ers, 2003. — Vol. 57. — P. 1-35.

276. Glover F. Implementation and computational comparisons of primal, dual, and primal-dual computer codes for minimum cost network flow problems / F. Glover, D. Karney, D. Klingman // Networks. — 1974. — Vol. 4. — N 3. — P. 191-212.

277. Glover F. A practitioner's guide to the state of large scale network and network-related problems / F. Glover, D. Klingman. — National Computer Conference, 1976. — P. 945-950.

278. Glover F. Computational study of an improved shortest path algorithm / F. Glover, R. Glover, D. Klingman // Networks. — 1984. — Vol. 14. — N 1. — P. 25-36.

279. Glover F. A new polynomially bounded shortest paths algorithm / F. Glover, D. Klingman, N. Phillips // Operations Research. — 1985. — Vol. 33. — N 1. — P. 65-73.

280. Goldberg A.V. Beyond the Flow Decomposition Barrier / A.V. Goldberg , S. Rao // Journal of the ACM. — 1998. — N 45. — P. 783-797.

281. Goldberg A. Scaling Algorithms for the Shortest Paths Problem / A. Goldberg // SIAM Journal on Computing. — 1995. — **24**. — P. 494-504.

282. Goldberg A.V. A simple shortest path algorithm with linear average time / A.V. Goldberg // In 9th European Symposium on Algorithms: of Lecture Notes in Computer Science: Springer, 2001. — Vol. 2161. — P. 230-241.

283. Goldberg A.V. Shortest Path Algorithms: Engineering Aspects / A.V. Goldberg // In Proc. ISAAC ’01, Lecture Notes in Computer Science, Springer-Verlag, 2001. — P. 502-513.

284. Goldberg A. A heuristic improvement of the bellman-ford algorithm / A. Goldberg, T. Radzik // Applied Mathematics Letters. — 1993. — Vol. 6. — N 3. — P. 3-6.

285. Goldberg A. A Practical Shortest Path Algorithm with Linear Expected Time / A. Goldberg // SIAM J. Comp. — 2008. — 37. — P. 1637-1655.

286. Golden B.L. The Vehicle Routing Problem: Latest Advances and New Challenges / B.L. Golden, S. Raghavan, E.A. Wasil. — Springer Science & Business Media, 2008. — 591 p.

287. Golden B.L. The Fleet Size and Mix Vehicle Routing Problem / B.L. Golden, A. Assad, L. Levy, F. Gheysens // Computers & Operations Research. — 1984.

— N. 11. — P. 49-66.

288. Goldfarb D. Shortest path algorithms using dynamic breadth-first search / D. Goldfarb, J. Hao, S. Kai // Networks. — 1991. — Vol. 21. — No 1. — P. 29-50.

289. Goldfarb D. Efficient shortest path simplex algorithms / D. Goldfarb, J. Hao, S. Kai // Operations Research. — 1990. — Vol. 38. — N 4. — P. 624-628.

290. Goldfarb D. An o(nm)-time network simplex algorithm for the shortest path problem / D. Goldfarb, Z. Jin // Operations Research. — 1999. — Vol. 47. — N 3. — P. 445-448.

291. Goksal F.P. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery / F.P. Goksal, I. Karaoglan, F. Altiparmak // Computers & Industrial Engineering. — 2013. — Vol. 65. — Issue 1. — P. 39-53.

292. Goodrich M.T. JDSL – the data structures library in Java / M.T. Goodrich, R. Tamassia // http://www.jdsl.org/.

293. Goto S. Sparse matrix techniques for the shortest path problem / S. Goto, T. Ohtsuki, T. Yoshimura // IEEE Transactions on Circuits and Systems. — 1976. — Vol. CAS-23. — P. 752-758.

294. Gottschlich C. The Shortlist Method for Fast Computation of the Earth Mover's Distance and Finding Optimal Solutions to Transportation Problems / C. Gottschlich, D. Schuhmacher. — PLoS ONE, 2014. — Vol. 9. — N 10. — P. 1-10, www.plosone.org.

295. Graham R.L. On the History of the Minimum Spanning Tree Problem / R.L. Graham, P. Hell // Annals of the History of Computing. — 1985. — Vol. 7(l). — P. 43-57.

296. Granat J. The interactive analysis of the multicriteria shortest path problem by the reference point method / J. Granat, F. Guerriero // European Journal of Operational Research. — 2003. — Vol. 151. — P. 103-111.

297. Greenberg H.J. An annotated bibliography for post-solution analysis in mixed integer and combinatorial optimization / H.J. Greenberg // Advances in computational and stochastic optimization, logic programming and heuristic search. Boston, MA: Kluwer Academic Publishers, 1998. — P. 97-148.

298. Guerriero F. Label correcting methods to solve multicriteria shortest path problems / F. Guerriero, R. Musmanno // Journal of Optimization Theory and Applications. — 2001. — Vol. 111. — P. 589-613.

299. Gulczynski D.J. Recent Developments in Modeling and Solving the Split Delivery Vehicle Routing Problem / D.J. Gulczynski, B. Golden, E. Wasil // Tutorials in Operations Research, INFORMS. — 2008. — P. 170-180.

300. Halse K. Modeling and solving complex vehicle routing problems / K. Halse. — PhD thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Denmark, 1992.

301. Han A. A GIDS metaheuristic approach to the fleet size and mix vehicle routing problem / A. Han, Y.-Y. Cho // In Essays and Surveys in Metaheuristics, C. Ribeiro and P. Hansen, eds., Kluwer, Boston, 2001. — Vol. 15. — P. 399-414.

302. Handbook of approximation algorithms and metaheuristics / Edited by T.F. Gonzales, Chapman & Hall / CRC Computer and Information Science Series, CRC Press, 2007. — 1432 p.

303. Hansen P. Bicriteria path problems / P. Hansen // in Lecture Notes in Economics and Mathematical Systems: G. Fandel & T. Gal Eds. Berlin: Springer Verlag, 1979. — Vol 177. — P. 109-127.

304. Hezer S. Solving vehicle routing problem with simultaneous delivery and pick-up using bacterial foraging optimization algorithm / S. Hezer, Y. Kara // In Proceedings of the 41st International Conference on Computers & Industrial Engineering, University of Southern California, 23-26 October, Los Angeles, California. — 2011. — P. 380-385.

305. Hitchcock F.L. The distribution of a product from several sources to numerous localities / F.L. Hitchcock // Journal Mathematics and Physics. — 1941. — Vol. 20. — N 2. — P. 224-230.

306. Ho S.C. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries / S.C. Ho, D. Haugland // Computers & Operations Research. — 2004. — Vol. 31. — Issue 12. — P. 1947-1964.

307. Hoff A. Industrial Aspects and Literature Survey: Fleet Composition and Routing / A. Hoff, H. Andersson, M. Christiansen, G. Hasle, A. Løkketangen // SINTEF REPORT NO. A7029. — 2008. — 49 p.

308. Hoff A. Industrial aspects and literature survey: Fleet composition and routing / A. Hoff, H. Andersson, M. Christiansen, G. Hasle, A. Løkketangen // Computers & Operations Research. — 2010. — 37. — P. 2041-2061.

309. Hopcroft J.E. Set merging algorithms / J.E. Hopcroft, J.D. Ullman // SIAM J. Computing. — 1973. — Vol. 2. — N. 4. — P. 294-303.

310. Hromkovich J. Design and analysis of randomized algorithms. Introduction to design paradigms / J. Hromkovich. — Berlin, Heidelberg: Springer - Verlag, 2005. — 280 p.

311. http://www.gisa.ru/31553.html.

312. http://www.esri.com.

313. http://www.esri.com/software/arcgis/extensions/networkanalyst.

314. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html, http://www.gams.com/, http://aimms.com/, http://www.gurobi.com/.

315. http://www.informatik.uni-koeln.de/abacus/, http://www.coin-or.org/, http://www.gnu.org/software/glpk/glpk.html, http://groups.yahoo.com/group/lp_solve/info/.

316. http://www.nvidia.ru/, https://developer.nvidia.com/cuda-zone.

317. Hu T. A decomposition algorithm for shortest paths in a network / T. Hu // Operations Research. — 1968. — Vol. 16. — P. 91-102.

318. Hu T. Revised matrix algorithms for shortest paths / T. Hu // SIAM Journal of Applied Mathematics. — 1967. — Vol. 15. — P. 207-218.

319. Huang H. Particle swarm optimization algorithm for transportation problems / H. Huang, Z Hao // Particle swarm optimization, ed. Aleksandar Lazinica, InTech. — 2009. — P. 275-290.

320. Hung M. A computational study of efficient shortest path algorithms / M. Hung, J. Divoky // Computers and Operations Research. — 1988. — Vol. 15. — N 6. — P. 567-576.

321. Ikura Y. Computational experience with a polynomial-time dual simplex algorithm for the transportation problem / Y. Ikura, G.L. Nemhauser // Discrete applied mathematics. — 1986. — Vol. 13. — P. 239-248.

322. I-Lin Wang. Shortest Paths and Multicommodity Network Flows: School of Industrial and Systems Engineering Georgia Institute of Technology: In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy. — 2003. — 227 p.

323. Imran A. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem / A. Imran, S. Salhi, N.A. Wassan // European Journal of Operational Research. — 2009. — Vol. 197. — Issue 2. — P. 509-518.

324. Irnich S. Four variants of the vehicle routing problem / S. Irnich, M. Schneider, D. Vigo // In P. Toth, D. Vigo (Eds.), Vehicle routing: Problems, methods, and applications. In MOS-SIAM series on optimization. Philadelphia. — 2014. — P. 241-271.

325. Jarník V. O jistém problému minimálním / V. Jarník // Práce Moravské Prirodovedecké Spolecnosti. — 1930. — Vol. 6. — P. 57-63.

326. Johnson E. On shortest paths and sorting / E. Johnson // in Proceedings of the ACM 25th annual conference, 1972. — P. 510-517.

327. Johnson D.B. Priority queues with update and finding minimum spanning trees / D.B. Johnson: Inf. Processing Letters. — 1975. — Vol. 4. — N 3. — P. 53-57.

328. Johnson D. Efficient Algorithms for Shortest Paths in Sparse Networks / D. Johnson // Journal of the ACM. — 1977. — 24. — P. 1-13.

329. Johnson D. A priority queue in which initialization and queue operations take o(loglogd) time / D. Johnson // Mathematical Systems Theory. — 1982. — Vol. 15. — N 4. — P. 295-309.

330. Juanjuan H. A Bio-Inspired Algorithm for the Fleet Size and Mix Vehicle Routing Problem / H. Juanjuan, S. Tao // Journal of Computational and Theoretical Nanoscience. — 2014. — Volume 11. — No. 10 — P. 2085-2090.

331. Jun Y. New best solutions to VRPSPD benchmark problems by a perturbation based algorithm / Y. Jun, B.-I. Kim // Expert Systems with Applications. — 2012. — Vol. 39. — Issue 5. — P. 5641-5648.

332. Karagül K. A New Heuristic Routing Algorithm for Fleet Size and Mix Vehicle Routing Problem / K. Karagül, H. Myo // Gazi University Journal of Science. — 2014. — Volume 27. — Issue 3. — P. 979-986.

333. Karger D.R. A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees / D.R. Karger, P.N. Klein, R.E. Tarjan // Journal of the ACM. — 1995. — Vol. 42(2). — P. 321-328.

334. Karp R.M. Reducibility among combinatorial problems / R.M. Karp. — In: Complexity of Computer Computations / Eds. R.E. Miller, J.W. Thatcher. — Proceedings of a symposium on the Complexity of Computer Computations, IBM Thomas J. Watson Research Center, Yorktown Heights, Plenum Press, New York,

1972. — P. 85-103.

335. Katriel I. A practical minimum spanning tree algorithm using the cycle property / I. Katriel, P. Sanders, J.L. Träff. — Technical Report MPI-I-2002-1-003, MPI Informatik, Germany, October, 2002.

336. Kececi B. The heterogeneous vehicle routing problem with simultaneous pickup and delivery: A hybrid heuristic approach based on simulated annealing / B. Kececi, F. Altiparmak, I. Kara // In CIE44 & IMSS'14 Proceedings, 14-16 October, Istanbul / Turkey. — 2014. — P. 412-423.

337. Kellerer H. Knapsack Problems / H. Kellerer, U. Pferschy, D. Pisinger. — Springer-Verlag Berlin Heidelberg, 2004. — 548 p. (The Multiple-Choice Knapsack Problem, P. 317-347).

338. Kennedy J. Particle swarm optimization / J. Kennedy, R. Eberhart // In Proceedings of IEEE international conference on neural networks. — 1995. — Vol. 4. — P. 1942-1948.

339. Kennington J.L. A survey of linear cost multicommodity networks flows / J.L. Kennington // Oper. Res. — 1978. — **26**. — P. 206-236.

340. Khmelev A. A hybrid local search for the split delivery vehicle routing problem / A. Khmelev, Y. Kochetov // International Journal of Artificial Intelligence. — 2015. — Vol. 13. — Issue 1. — P. 147-164.

341. Khmelev A. A hybrid VND method for the split delivery vehicle routing problem / A. Khmelev, Y. Kochetov // Electronic Notes in Discrete Mathematics. — 2015. — Vol. 47. — P. 5-12.

342. King V. A Simpler Minimum Spanning Tree Verification Algorithm / V. King // Algorithmica. — 1997. — Vol. 18(2). — P. 263-270.

343. Kirby D. Is Your Fleet the Right Size? / D. Kirby // Operational Research Quarterly. — 1959. — N. 10. — P. 252.

344. Kleinschmidt P. A Strongly Polynomial Algorithm for the Transportation Problem / P. Kleinschmidt, H. Schannath // Mathematical Programming. — 1995. — Vol. 68. — N 1. — P. 1-13.

345. Koç Ç. Thirty years of heterogeneous vehicle routing / Ç. Koç, T. Bektaş, O. Jabali, G. Laporte // European Journal of Operational Research. — 2015. — Vol. 249. — Issue 1. — P. 1-21.

346. Koç Ç. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems / Ç. Koç, T. Bektaş, O. Jabali, G. Laporte // Computers & Operations Research. — 2015. — Vol. 64. — P. 11-27.

347. Koç Ç. Heterogeneous Location- and Pollution-Routing Problems / Ç. Koç. — Thesis for the degree of Doctor of Philosophy in Management Science, University of Southampton, 2015. — 239 p.

348. Koç Ç. The Fleet Size and Mix Location Routing Problem with Time Windows: Formulations and a Heuristic Algorithm / Ç. Koç, T. Bektaş, O. Jabali, G. Laporte // European Journal of Operational Research. — 2016. — Vol. 248. — Issue 1. — P. 33-51.

349. Komlos J. Linear verification for spanning trees / J. Komlos // Combinatorica. — 1985. — Vol. 5(l). — P. 57-65.

350. Koopmans T.C. Optimum Utilization of the transportation system / T.C. Koopmans // Proceeding of the International Statistical Conferences. — Washington, 1947. — N 5.

351. Korte B. Combinatorial Optimization. Theory and Algorithms, Fourth Edition / B. Korte, J. Vygen. — Berlin Heidelberg: Springer-Verlag, 2008. — 627 p.

352. Kruskal J.B. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem / J.B. Kruskal // Proc. Amer. Math. Soc. — 1965. — Vol. 7. — P. 48-50.

353. Kuhn H.W. The Hungarian method for the assignment problem / H.W. Kuhn // Naval research logistics quaterly. — 1955. — Vol. 2. — N 1, 2. — P. 83-97.

354. Kumar V. Introduction to Parallel Computing. Second Edition / V. Kumar, A. Grama, A. Gupta, G. Karypis. — Addison Wesley, 2003. — 856 p.

355. Kumar R. Multicriteria network design using evolutionary algorithm / R. Kumar, N. Banerjee // in Lecture Notes in Computer Science, Berlin: Springer Verlag, 2003. — Vol. 2724. — P. 2179-2190.

356. Kumar S.N. A Survey on the Vehicle Routing Problem and Its Variants / S.N. Kumar, R. Panneerselvam // Intelligent Information Management. — 2012. — **4**. — P. 66-74.

357. Land A. The extension of the cascade algorithm to large graphs / A. Land, S. Stairs // Management Science. — 1967. — Vol. 14. — P. 29-33.

358. Laporte G. Fifty Years of Vehicle Routing / G. Laporte // Canada Research Chair in Distribution Management, HEC Montr´eal, 2009. — 23 p.

359. Laporte G. Classical Heuristics for the Vehicle Routing Problem / G. Laporte, F. Semet // Les Cahiers du GERAD, G98-54, Group for Research in Decision Analysis, Montreal, Canada, 1998. — 21 p.

360. Laporte G. Classical and modern heuristics for the vehicle routing problem / G. Laporte, M. Gendreau, J.-Y. Potvin, F. Semet // International Transactions in Operational Research. — 2000. — Vol. 7. — Issue 4-5. — P. 285-300.

361. Larsen J. Experiments with the auction algorithm for the shortest path problem / J. Larsen, I. Pedersen // Nordic Journal of Computing. — 1999. — Vol. 6. — N 4. — P. 403-421.

362. LEDA (Library of Efficient Data Types and Algorithms). www.algorithmic-solutions.com.

363. Lee Y.H. A heuristic for vehicle fleet mix problem using tabu search and set partitioning / Y.H. Lee, J.I. Kim, K.H. Kang, K.H. Kim // The Journal of the Operational Research Society. — 2008. — Vol. 59. — N. 6. — P. 833-841.

364. Lenstra J.K. Complexity of vehicle routing and scheduling problems / J.K. Lenstra, A.H.G. Rinnooy Kan // Networks. — 1981. — N. 11. — P. 221-227.

365. Li F. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem / F. Li, B. Golden, E. Wasil // Computers and Operations Research. — 2007. — Vol. 34. — Issue 9. — P. 2734-2742.

366. Li X. An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem / X. Li, P. Tian, Y. Aneja // Transportation Research, Part E: Logistics and Transportation Review. — 2010. — Vol. 46. — Issue 6. — P. 1111-1127.

367. Libura M. Optimality conditions and sensitivity analysis for combinatorial optimization problems / M. Libura // Control and Cybernetics. —1996. — N 25. — P. 1165-1180.

368. Lima C.M.R.R. A memetic algorithm for the heterogeneous fleet vehicle routing problem / C.M.R.R. Lima, M.C. Goldbarg, E.F.G. Goldbarg // Electronic Notes in Discrete Mathematics. — 2004. — Vol. 18. — Issue 1. — P. 171-176.

369. Liu S. An effective genetic algorithm for the fleet size and mix vehicle routing problems / S. Liu, W. Huang, H. Ma // Transportation Research, Part E: Logistics and Transportation Review. — 2009. — Vol. 45. — Issue 3. — P. 434-445.

370. Liu F.-H. The fleet size and mix vehicle routing problem with time windows / F.-H. Liu, S.-Y. Shen // Journal of the Operational Research Society. — 1999. — Vol. 50. — N. 7. — P. 721-732.

371. Liu F.H. A method for vehicle routing problem with multiple vehicle types and times windows / F.-H. Liu, S.-Y. Shen // Proceedings of the National Science Council, Republic of China, Part A: Physical Science and Engineering. — 1999. — Vol. 23. — No 4. — P. 526-536.

372. Liu R. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care / R. Liu, X. Xie, V. Augusto, C. Rodriguez // European Journal of Operational Research. — 2013. — Vol. 230. — Issue 3. — P. 475-486.

373. Loberman H. Formal procedures for connecting terminals with a minimum total wire length / H. Loberman, A. Weinberger // Journal of ACM. — 1957. — Vol. 4. — P. 428-437.

374. Loch G.V. Computational Study of Degeneracy in Initial Basic Feasible Solution for the Transportation Problem / G.V. Loch, A.C. Lindbeck da Silva, T.C. Lindbeck da Silva // International Refereed Journal of Engineering and Science (IRJES). — 2014. — Vol. 3. — N 7. — P. 14-19.

375. Loch G.V. A Computational Study on the Number of Iterations to Solve the Transportation Problem / G.V. Loch, A.C. Lindbeck da Silva // Applied Mathematical Sciences. — 2014. — Vol. 8. — N 92. — P. 4579-4583.

376. Luby M. Randomization and Approximation Techniques in Computer Science / M. Luby, J. Rolim, M. Serna: Second International Workshop, RANDOM'98, Barcelona, Spain, October 8–10, 1998, Proceedings, Springer, 2003. — 385 p.

377. Madduri K. Parallel Shortest Path Algorithms for Solving Large-Scale Instances / K. Madduri, D. Bader, J. Berry, J. Crobak // in The Shortest Path Problem: Ninth DIMACS Implementation Challenge, ser. DIMACS Book, C. Demetrescu, A. V. Goldberg, and D. S. Johnson, Eds. American Mathematical Society, 2009. — Vol. 74. — P. 249-290.

378. Martelo S. Knapsack problems: algorithms and computer implementations / S. Martelo, P. Toth. — Great Britain: Wiley, 1990. — 296 p.

379. Martins E. The labeling algorithm for the multiobjective shortest path problem / E. Martins, J. Santos // Departamento de Matematica, Universidade de Coimbra, Portugal, Tech. Rep. TR-99/005, 1999. — 24 p.

380. Mehlhorn K. The LEDA Platform for Combinatorial and Geometric Computing / K. Mehlhorn, S. Näher // Cambridge University Press, 1999. — 15 p.

381. Mehlhorn K. Algorithms and Data Structures. The Basic Toolbox / K. Mehlhorn, P. Sanders. — Springer, 2008. — 300 p.

382. Meyer U. Average-case complexity of single-source shortest-path algorithms: Lower and upper bounds / U. Meyer // Journal of Algorithms. — 2003. — 48(1). — P. 91-134.

383. Meyer U. Δ-stepping: A parallel shortest path algorithm / U. Meyer, P. Sanders // In 6th European Symposium on Algorithms: of Lecture Notes in Computer Science: Springer, 1998. — Vol. 1461. — P. 393-404.

384. Michalewicz Z. Evolutionary computation techniques for nonlinear programming problems / Z. Michalewicz // International Transactions in Operational Research. — 1994. — Vol. 1. — N 2. — P. 223-240.

385. Michalewicz Z. Genetic algorithms + data structures = evolution programs // Z. Michalewicz. — Berlin, Heidelberg: Springer, 1996. — 387 p.

386. Miller C.E. Integer programming formulations and traveling salesman problems / C.E. Miller, A.W. Tucker, R.A. Zemlin // ACM. — 1960. — Vol. 7. — P. 326-329.

387. Miller D.L. Solution of large dense transportation problems using a parallel primal algorithm / D.L. Miller, J.F. Pekny, G.L. Thompson. — Management Science Research Report No. 546, 1988. — 25 p.

388. Mills G. A decomposition algorithm for the shortest-route problem / G. Mills // Operations Research. — 1966. — Vol. 14. — P. 279-291.

389. Min H. The multiple vehicle routing problem with simultaneous delivery and pickup points / H. Min // Transportation Research Part A: General. — 1989. — Vol. 23. — Issue 5. — P. 377-386.

390. Mingyong L. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows / L. Mingyong, C. Erbao // Engineering Applications of Artificial Intelligence. — 2010. — Vol. 23. — Issue 2. — P. 188-195.

391. Mondou J. Shortest path algorithms: a computational study with the c programming language / J. Mondou, T. Crainic, S. Nguyen // Computers and Operations Research. — 1991. — Vol. 18. — P. 767-786.

392. Montané F.A.T. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service / F.A.T. Montané, R.D. Galvão // European Journal of Operational Research. — 2006. — Vol. 33. — Issue 3. — P. 595-619.

393. Mooney P. An evolutionary algorithm for multicriteria path optimization problems / P. Mooney, A. Winstanley // International Journal of Geographical Information Science. — 2006. — Vol. 20. — P. 401-423.

394. Moore E. The shortest path through a maze / E. Moore // in Proceedings of the International Symposium on theory of Switching: The Annals of the Computation Laboratory of Harvard University, 1957. — Part II. — P. 285-292.

395. Morris J. An escalator process for the solution of linear simultaneous equations / J. Morris // Philos. Mag. — 1946. — 7. — Vol. 37. — P. 106-120.

396. Müller-Hannemann M. Pareto Shortest Paths is Often Feasible in Practice / M. Müller-Hannemann, K. Weihe // in Lecture Notes in Computer Science: G. Brodal, D. Frigioni, A. Marchetti-Spaccamela Eds. Berlin: Springer Verlag, 2001. — Vol. 2141. — P. 185-198.

397. Mulvey J.V. Testing of a large-scale network optimization program / J.V. Mulvey // Math. Prog. — 1978. — Vol. 15. — N 3. — P. 291-314.

398. Murthy I. An interactive procedure using domination cones for bicriterion shortest path problems / I. Murthy, D. Olson // European Journal of Operational Research. — 1994. — 72. — P. 417-431.

399. Mutingi M. A Group Genetic Algorithm for the Fleet Size and Mix Vehicle Routing Problem / M. Mutingi, C. Mbohwa // IEEE. — 2012. — P. 1468-1472.

400. Nagy G. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries / G. Nagy, S. Salh // European Journal of Operational Research. — 2005. — Vol. 162. — Issue 1. — P. 126-141.

401. Nakamori M. A note on the optimality of some all-shortest-path algorithms / M. Nakamori // Journal of the Operations Research Society of Japan. — 1972. — Vol. 15. — P. 201-204.

402. Nešetřil J. Otakar Borůvka on minimum spanning tree problem: translation of both the 1926 papers, comments, history / J. Nešetřil, E. Milková, H. Nešetřilová // Discrete Mathematics. — 2001. — Vol. 233 (1–3). — P. 3-36.

403. Ochi L. An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet / L. Ochi, D. Vianna, L.M.A. Drummond, A.

Victor // Genetic Programming: Lecture notes in computer science. — 1998. — Vol. 1391. — P. 187-195.

404. Ochi L. A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet / L. Ochi, D. Vianna, L.M.A. Drummond, A. Victor // Future Generation Computer Systems: Bio-inspired solutions to parallel processing problems. — 1998. — Vol. 14. — Issue 5-6. — P. 285-292.

405. Ochi L.S. Design and Implementation of a Parallel Genetic Algorithm for the Traveling Purchaser Problem / L.S. Ochi, L.M.A. Drummond, R.M. Figueiredo // ACM Symp. on Applied Computing. — 1997. — P. 257-263.

406. Orlin J. B. Polynomial dual network simplex algorithms / J.B. Orlin, S.A. Plotkin , E. Tardos // Math. Program. — 1993. — Vol. 60A. — N 3. — P. 255-276.

407. Osman I.H. Local Search Strategies for the Vehicle Fleet Mix Problem / I.H. Osman, S. Salhi // In Rayward-Smith V.J., Osman I.H., Reeves C.R., Smith G.D., eds., Modern heuristic search methods., John Wiley & Sons, Chichester, UK, 1996. — P. 131-153.

408. Pallottino S. Shortest-path methods: complexity, interrelations and new propositions / S. Pallottino // Networks. — 1984. — Vol. 14. — P. 257-267.

409. Pallottino S. Dual algorithms for the shortest path tree problem / S. Pallottino, M. Scutellua // Networks. — 1997. — Vol. 29. — N 2. — P. 125-133.

410. Papadimitriou C. Combinatorial optimization: algorithms and complexity / C. Papadimitriou, K. Steiglitz. — Englewood Cliffs, NJ: Prentice-Hall, 1982. — 496 p.

411. Papadimitriou C. On the Approximability of Trade-offs and Optimal Access of Web Sources / C. Papadimitriou, M. Yannakakis // Proc. 41st Symp. Foundations of Computer Science, FOCS, Redondo Beach, CA. — 2000. — P. 86-92.

412. Pape U. Implementation and efficiency of Moore algorithms for the shortest root problem / U. Pape // Mathematical Programming. — 1974. — Vol. 7. — P. 212-222.

413. Paraskevopoulos D.C. A reactive variable neighbourhood tabu search for the heterogeneous fleet vehicle routing problem with time windows / D.C. Paraskevopoulos, P.P. Repoussis, C.D. Tarantilis, G. Ioannou, G.P. Prastacos // Journal of Heuristics. — 2008. — Vol. 14. — Issue 5. — P. 425-455.

414. Parragh S. A survey on pickup and delivery problems. Part I: Transportations between customers and depot / S. Parragh, K. Doerner, R. Hartl // J.

Betriebswirtschaft. — 2008. — Vol. 58. — N 1. — P. 21-51.

415. Parragh S. A survey on pickup and delivery problems. Part II: Transportations between customers and depot / S. Parragh, K. Doerner, R. Hartl // J. Betriebswirtschaft. — 2008. — Vol. 58. — N 2. — P. 81-117.

416. Pessoa A. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem / A. Pessoa, E. Uchoa, M. Poggi de Aragão // Networks. — 2009. — Vol. 54. — Issue 4. — P. 167-177.

417. Pettie S. An optimal minimum spanning tree algorithm / S. Pettie, V. Ramachandran // In 7th International Colloquium on Automata, Languages and Programming: of Lecture Notes in Computer Science, Springer, 2000. — Vol. 1853. — P. 49-60.

418. Pettie S. Computing shortest paths with comparisons and additions / S. Pettie, V. Ramachandran // SODA '02 Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, San Francisco, California — January 06 - 08, 2002. — P. 267-276.

419. Pettie S. Randomized minimum spanning tree algorithms using exponentially fewer random bits / S. Pettie, V. Ramachandran // ACM Transactions on Algorithms. — 2008. — Vol. 4. — N 1. — P. 1-27.

420. Pierce A.R. Bibliography on algorithms for shortest path, shortest spanning tree and related circuit routing problems (1956-1974) / A.R. Pierce // Networks. — 1975. — N 5. — P. 129-149.

421. Prim R.C. Shortest Connection Networks and Some Generalizations / R.C. Prim // Bell Syst. Tech. J. — 1957. — Vol. 36. — P. 1389-1401.

422. Prins C. Two memetic algorithms for heterogeneous fleet vehicle routing problems / C. Prins // Engineering Applications of Artificial Intelligence. — 2009. — Vol. 22. — Issue 6. — P. 916-928.

423. Prins C. Order-first split-second methods for vehicle routing problems: A review / C. Prins, P. Lacomme, C. Prodhon // Transportation Research Part C. — 2014. — 40. — P. 179–200.

424. Rado R. Note on Independence Function / R. Rado // Proc. London Math. Soc., 1957. — **7**. — P. 300-320.

425. Renaud J. A sweep-based algorithm for the fleet size and mix vehicle routing problem / J. Renaud, F. Boctor // European Journal of Operational Research. — 2002. — Vol. 140. — Issue 3. — P. 618-628.

426. Resende M. An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite incapacitated networks / M. Resende, G. Veiga // SIAM Journal on Optimization. — 1992. — P. 1-22.

427. Rodrigues J. An interactive bi-objective shortest path approach: Searching for unsupported nondominated solutions / J. Rodrigues, J. Cl´ımaco, J. Current // Computers & Operations Research. — 1999. — 26. — P. 789-798.

428. Røpke S. A unified heuristic for a large class of vehicle routing problems with backhauls / S. Røpke, D. Pisinger // European Journal of Operational Research. — 2006. — Vol. 171. — Issue 3. — P. 750-775.

429. Sahni S. P-complete approximation problems / S. Sahni, T. Gonzalez // J. Assoc. Comput. Mach. —1976. — N. 23. — P. 555–565.

430. Salhi S. Incorporating vehicle routing into the vehicle fleet composition problem / S. Salhi, G.K. Rand // European Journal of Operational Research. — 1993. — N. 66. — P. 313-330.

431. Salhi S. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling / S. Salhi, G. Nagy // Journal of the Operational Research Society. — 1999. — Vol. 50. — Issue 10. — P. 1034-1042.

432. Seidel R. On the all-pairs-shortest-path problem in unweighted undirected graphs / R. Seidel // Journal of Computer and System Sciences. — 1995. — Vol. 51. — P. 400-403.

433. Shimbel A. Applications of matrix algebra to communication nets / A. Shimbel // Bulletin of Mathematical Biophysics. — 1951. — Vol. 13. — P. 165-178.

434. Sollin M. Le tracé de canalisation / M. Sollin // Programming, Games, and Transportation Networks (in French). — 1965.

435. Sotskov Yu.N. Some concepts of stability analysis in combinatorial optimization / Yu.N. Sotskov, V.K. Leontev, E.N. Gordeev // Discrete Appl. Math. — 1995. — Vol. 58. — N 2. — P. 169-190.

436. Souza M.J.F. A hybrid heuristic, based on iterated local search and GENIUS, for the vehicle routing problem with simultaneous pickup and delivery / M.J.F. Souza, M.T. Mine, M.S.A. Silva, L.S. Ochi, A. Subramanian // International Journal of Logistics Systems Management. — 2010. — Vol. 10. — Issue 2. — P. 142-156.

437. Subramanian A. Metaheurística Iterated Local Search aplicada ao Problema de Rotetamento de Veículos com Coleta e Entrega Simultânea / A. Subramanian. — Master's thesis, Programa de Pós-graduação em Engenharia de Produção, Universidade Federal da Paraíba, João Pessoa, PB, Brazil, 2008. In Portuguese.

438. Subramanian A. An efficient ILS heuristic for the vehicle routing problem with simultaneous pickup and delivery / A. Subramanian, L.S. Ochi, L.A.F. Cabral // Tech. rep. RT 07/08, Universidade Federal Fluminense. — 2008. — 17 p., available at http://www2.ic.uff.br/PosGraduacao/RelTecnicos/400.pdf.

439. Subramanian A. An ILS based heuristic for the vehicle routing problem with simultaneous pickup and delivery and time limit / A. Subramanian, L.A.F. Cabral // Proceedings of the Eighth European Conference on Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science. — 2008. — Vol. 4972. — P. 135-146.

440. Subramanian A. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery / A. Subramanian, L.M.A. Drummond, C. Bentes, L.S. Ochi, R. Farias // Computers & Operations Research. — 2010. — Vol. 37. — Issue 11. — P. 1899-1911.

441. Subramanian A. A hybrid algorithm for the heterogenous fleet vehicle routing problem / A. Subramanian, P.H.V. Penna, E. Uchoa, L.S. Ochi // European Journal of Operational Research. — 2012. — Vol. 221. — Issue 2. — P. 285-295.

442. Subramanian A. Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems / A. Subramanian. — Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science, Niterói, 2012. — 152 p.

443. Taillard E.D. A heuristic column generation method for heterogeneous fleet / E.D. Taillard // RAIRO - Operations Research. — 1999. — Vol. 33. — Issue 01. — P. 1-14.

444. Takaoka T. A new upper bound on the complexity of the all pairs shortest path problem / T. Takaoka // Information Processing Letters. — 1992. — Vol. 43. — N 4. — P. 195-199.

445. Takaoka T. Subcubic cost algorithms for the all pairs shortest path problem / T. Takaoka // Algorithmica. — 1998. — Vol. 20. — N 3. — P. 309-318.

446. Talbi E.G. A taxonomy of hybrid metaheuristics / E.G. Talbi // Journal of Heuristics. — 2002. — **8**. — P. 541-564.

447. Tarantilis C.D. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem / C.D. Tarantilis, C. Kiranoudis, V Vassiliadis // Journal of the Operational Research Society. — 2003. —Vol. 54. — Issue 1. — P. 65-71.

448. Tarantilis C.D. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem / C.D. Tarantilis, C.T. Kiranoudis, V.S. Vassiliadis // European Journal of Operational Research. — 2004. — Vol. 152. — Issue 1. — P. 148-158.

449. Tarantilis C.D. A guided tabu search for the heterogeneous vehicle routing problem / C.D. Tarantilis, E.E. Zachariadis, C.T. Kiranoudis // J. Operational Res. Society. — 2008. — Vol. 59. — Issue 12. — P. 1659-1673.

450. Tarjan R. E. Efficiency of a good but not linear set union algorithm / R.E. Tarjan // Journal of the ACM. — 1975. — Vol. 22. — N. 2. — P. 215-225.

451. Tarjan R.E. Data Structures and Network Algorithms / R.E. Tarjan. — Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

452. Tarjan R.E. Worst-Case Analysis of Set Union Algorithms / R.E. Tarjan, J.V. Leeuwen // J. of the ACM. — 1984. — Vol. 31. — N. 2. — P. 245-281.

453. Tarjan R.E. Amortized computational complexity / R.E. Tarjan // SIAM J. on Algebraic and Discrete Methods. — 1985. — Vol. 6. — N. 2. — P. 306-318.

454. Tasan A.S. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries / A.S. Tasan, M. Gen // Computers & Industrial Engineering. —2012. — Vol. 62. — Issue 3. — P. 755-761.

455. Tavakkoli-Moghaddam R. A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing / R. Tavakkoli-Moghaddam, N. Safaei, M.M.O. Kah, M. Rabbani // Journal of the Franklin Institute. — 2007. — Vol. 344. — Issue 5. — P. 406-425.

456. Thorup M. Undirected single-source shortest paths with positive integer weights in linear time / M. Thorup // Jounral of ACM. — 1999. — Vol. 46. — N 3. — P. 362-394.

457. Thorup M. Floats, integers, and single source shortest paths / M. Thorup // Journal of Algorithms. — 2000. — Vol. 35. — N 2. — P. 189-201.

458. Thorup M. Integer priority queues with decrease key in constant time and the single source shortest paths problem / M. Thorup // Journal of Computer and System Sciences. — 2004. — 69(3). — P. 330-353.

459. Toth P. Vehicle Routing: Problems, Methods, and Applications, Second Edition / P. Toth, D. Vigo, SIAM, 2014. — 463 p.

460. Trofimchuk A.N. Management and decision making in hierarchical communication networks with discrete flows / A.N. Trofimchuk, L.P. Ushakova, V.A. Vasjanin // V International Conference on Optimization Methods and Applications «Optimization and applications» (OPTIMA-2014) held in Petrovac, Montenegro, September 28 - October 4, 2014, Petrovac, Montenegro: Proceedings. — Moscow, Russia: ESC Dorodnicyn Computing Centre of RAS, Russia, 2014. — P. 187-188.

461. Trofymchuk O. On complexity of one variety of the packing problem / O. Trofymchuk, L. Ushakova, V. Vasyanin // VI International Conference on Optimization Methods and Applications «Optimization and applications» (OPTIMA-2015) held in Petrovac, Montenegro, September 27 - October 3, 2015, Petrovac, Montenegro: Proceedings. — Moscow, Russia: ESC Dorodnicyn Computing Centre of RAS, Russia, 2015. — P. 175-176.

462. Trofimchuk A.N. Simulation of Packing, Distribution and Routing of Small-Size Discrete Flows in a Multicommodity Network / A.N. Trofimchuk, V.A. Vasyanin // Journal of Automation and Information Sciences. — 2015. — Vol. — 47. — Issue 7. — P. 15-30.

463. Tsaggouris G. Improved FPTAS for multiobjective shortest paths with applications / G. Tsaggouris, Ch. Zaroliagis // Research Academic Computer Technology Institute, Petras, Greece. — Tech. Rep. No. TR 2005/07/03, 2005.

464. Tsaggouris G. Multi-objective optimization: improved FPTAS for shortest paths and non-linear objectives with applications / G. Tsaggouris, C. Zaroliagis // Proceedings 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006. — P. 389-398.

465. Vanderpooten D. Description and analysis of some representative interactive multicriteria procedures / D. Vanderpooten, Ph. Vincke // Mathematical and Computer Modelling. — 1989. — 12(10/11). — P. 1221-1238.

466. Vassilvitskii S. Efficiently computing sufficient trade-off curves / S. Vassilvitskii, M. Yannakakis // Lect. Notes Comput. Sci. — 2004. — Vol. 3142. — P. 1201-1213.

467. Vasyanin V.A. Problem of Distribution and Routing of Transport Blocks with Mixed Attachments and Its Decomposition / V.A. Vasyanin // Journal of Automation and Information Sciences. — 2015. — Vol. — 47. — Issue 2. — P. 56-69.

468. Vidal T. A unified solution framework for multi-attribute vehicle routing problems / T. Vidal, T.G. Crainic, M. Gendreau, C. Prins // European Journal of Op-

erational Research. — 2014. — Vol. 234. — Issue 3. — P. 658-673.

469. Vural A.V. A GA based meta-heuristic for capacity vehicle routing problem with simultaneous pick-up and deliveries / A.V. Vural. — Master's thesis, Graduate School of Engineering and Natural Sciences, Sabanci University, 2003.

470. Wang H.F. A genetic algorithm for the simultaneous delivery and pickup problems with time window / H.F. Wang, Y.-Y. Chen // Computers & Industrial Engineering. — 2012. — Vol. 62. — Issue 1. — P. 84-95.

471. Wang C. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup & delivery and time windows / C. Wang, D. Mu, F. Zhao, J.W. Sutherland // Computers & Industrial Engineering. — 2015. — Vol. 83. — P. 111-122.

472. Warburton A. Approximation of Pareto optima in multiple-objective, shortest-path problems // A. Warburton // Oper. Res. — 1987. — Vol. 35. — N 1. — P. 70-79.

473. Warshall S. A theorem on boolean matrics / S. Warshall // Journal of ACM. — 1962. — Vol. 9. — P. 11-12.

474. Wassan N.A. Tabu search variants for the mix fleet vehicle routing problem / N.A. Wassan, I.H. Osman // Journal of the Operational Research Society. — 2002. — Vol. 53. — Issue 7. — P. 768-782.

475. Wassan N.A. A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries / N.A. Wassan, A.H. Wassan, G. Nagy // Journal of Combinatorial Optimization. — 2008. — Vol. 15. — Issue 4. — P. 368-386.

476. Wu X. Implementation Issues for the Reliable A Priori Shortest Path Problem / X. Wu, Y. Nie // Transportation Research Record: Journal of the Transportation Research Board. — 2009. — N 2091. — P. 51-60.

477. Yaghini M. Multicommodity Network Design Problem in Rail Freight Transportation Planning / M. Yaghini, R. Akhavan // 8th Intern. Conf. on Traffic and Transportation Studies. Changsha, China, August 1–3. —2012. — N 3. — P. 728-739.

478. Yaman H.D. Formulations and valid inequalities for the heterogeneous vehicle routing problem / H.D. Yaman // Mathematical Programming. — 2006. — Vol. 106. — N. 2. — P. 365-390.

479. Yang L. An extension of the revised matrix algorithm / L. Yang, W. Chen // in IEEE international Symposium on Circuits and Systems, (Portland, Oregon): IEEE, 1989. — P. 1996-1999.

480. Yannakakis M. Computational complexity. Local search in combinatorial optimization / M. Yannakakis. — Chichester: Wiley, 1997. — P. 19-55.

481. Yao A.C.C. An O(|E|loglog|V|) Algorithm for Finding Minimum Spanning Trees / A.C.C. Yao // Info. Proc. Let. — 1975. — Vol. 4. — P. 21-25.

482. Yao A.C.-C. On the Average Behavior of Set Merging Algorithms (extended abstract) / A.C.-C. Yao. — Proceedings of seventh Annual ACM Symposium on Theory of Computing. — 1976. — P. 192-195.

483. Yen J. An algorithm for finding shortest routes from all source nodes to a given destination in general networks / J. Yen // Quarterly of Applied Mathematics. —1970. — Vol. 27. — P. 526-530.

484. Yin C. Mathematical model and algorithm of split load vehicle routing problem with simultaneous delivery and pickup / C. Yin, L. Bu, H. Gong // International Journal of Innovative Computing, Information and Control. — 2013. — Vol. 9. — N. 11. — P. 4497-4508.

485. Zachariadis E.E. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service / E.E. Zachariadis, C.D. Tarantilis, C.T. Kiranoudis // Expert Systems with Applications. — 2009. — Vol. 36. — Issue 2. — P. 1070-1081.

486. Zachariadis E.E. An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries / E.E. Zachariadis, C.D. Tarantilis, C.T. Kiranoudis // European Journal of Operational Research. — 2010. — Vol. 202. — Issue 2. — P. 401-411.

487. Zachariadis E.E. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries / E.E. Zachariadis, C.T. Kiranoudis // Expert Systems with Applications. — 2011. — Vol. 38. — Issue 3. — P. 2717-2726.

488. Zachariadis E.E. The Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries and Two-Dimensional Loading Constraints / E.E. Zachariadis, C.D. Tarantilis, C.T. Kiranoudis // European Journal of Operational Research. — 2016. — Vol. 251. — Issue 2. — P. 369-386.

489. Zadeh N. More pathological examples for network flow problems / N. Zadeh // Math. Prog. — 1973. — Vol. 5. — N 2. — P. 217-224.

490. Zadeh N. A bad network problem for the simplex-method and other minimum cost flow algorithms / N. Zadeh // Math. Prog. — 1973. — Vol. 5. — N 3. — P. 255-266.

491. Zhan F. Shortest path algorithms: an evaluation using real road networks / F. Zhan, C. Noon // Transportation Science. — 1998. — Vol. 32. — P. 65-73.

492. Zitzler E. Multiobjective evolutionary algorithms: a comparative study and the strength Pareto approach / E. Zitzler, L. Thiele // IEEE Transactions on Evolutionary Computation. — 1999. — Vol. 3. — P. 257-271.

493. Zwick U. All pairs shortest paths in weighted directed graphs - exact and almost exact algorithms / U. Zwick // in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, (Palo Alto, California), November, 1998. — P. 310-319.

Приложение А. Типовые задания по разработке информационного обеспечения и автоматизации процессов поддержки принятия решений

В приложении приводятся первоочередные задания по повышению эффективности функционирования сетей перевозок мелкопартионных корреспонденций.

Задание 1. Разработка методов и выбор инструментальных средств для проектирования автоматизированной информационной системы сбора и хранения данных о перевозках мелкопартионных грузов. Задание включает:

разработку структуры и форм документов первичного учета данных о перевозках мелкопартионных грузов;

разработку критериев и алгоритмов проверки данных в документах первичного учета;

проектирование типовой локальной вычислительной сети и локальной базы данных для узлов - источников данных о перевозках;

проектирование типовой локальной вычислительной сети и локальной базы данных для магистральных узловых транспортных предприятий;

проектирование глобальной межузловой сети передачи данных и распределенной централизованной базы данных в главном диспетчерском узле транспортной сети;

разработку методов и информационных процессов обработки данных в местах первичного учета, локальных базах данных, распределенной централизованной базе данных;

разработку схем передачи данных по каналам связи;

разработку методов информационного моделирования данных для решения прикладных задач перспективного развития, текущего планирования и оперативного управления в сети перевозок.

Основными научными и практическими результатами исследований задания 1 являются:

- обобщенное представление об информационном ресурсе, объединяющем как традиционные базы данных, так и документы в их толковании в соответст-

вии с международным стандартом на структуру документов о перевозках грузов;

- методология комплексного проектирования информационного обеспечения перевозок мелкопартионных грузов на основе единой интегрированной концептуальной модели предметной области;

- методы отображения данных как логических единиц документа в концептуальную модель предметной области;

- критерии корректности процессов обмена данными между информационными ресурсами системы и алгоритмы их проверки;

- структура типовой локальной вычислительной сети, типовая локальная база данных и технология обработки данных в узлах-источниках данных о перевозках и в магистральных узловых транспортных предприятиях;

- структура глобальной межузловой сети передачи данных, распределенная централизованная база данных, метаданных и технология обработки и хранения данных в главном диспетчерском узле сети перевозок;

- схема передачи данных по каналам связи между всеми узлами сети перевозок.

Полученные результаты составят методическую основу для разработки и внедрения автоматизированной информационной системы сбора и хранения данных о контейнерных перевозках мелкопартионных грузов.

Задание 2. Разработка комплекса методов и средств автоматизации поддержки процессов принятия оперативных решений в системе контейнерных перевозок мелкопартионных грузов. Задание включает:

- разработку состава комплекса;

- проработку общей методологии процессов принятия оперативных решений на основе баз данных и баз знаний;

- разработку и программную реализацию макетов подсистем управления базами данных и управления базами знаний.

Основными научными и практическими результатами исследований задания 2 являются:

- модель процессов принятия оперативных решений по управлению контейнерными перевозками в экстремальных условиях;

- рабочий макет программных средств анализа альтернативных решений, генерируемых системой при поддержке принятия решений в задачах организационного управления.

Базовый состав комплекса должен включать: подсистему управления базами данных; подсистему управления базами знаний; подсистему ведения каталогов прикладных программ; подсистему описания знаний и модели среды; подсистему настройки на пользователя; подсистему моделирования последствий решений; подсистему обучения.

Задание 3. Провести исследования технологий обработки мелкопартионных грузов в транспортных предприятиях и разработать комплекс методических материалов по проектированию автоматизированных рабочих мест (АРМ) работников транспортных предприятий и действующих макетов АРМ для контейнерной технологии обработки мелкопартионных грузов. Задание включает:

- выбор и обоснование методов проектирования АРМ работников транспортных предприятий для контейнерной технологии обработки грузов;

- анализ и разработку технологий обработки данных первичного учета для контейнерных перевозок грузов;

- разработку методик и инструментария проектирования типовых АРМ работников транспортных предприятий;

- разработку действующих макетов АРМ.

Основными научными и практическими результатами исследований задания 3 являются:

- технология обработки данных первичного учета для контейнерных перевозок мелкопартионных грузов;

- методика и инструментарий проектирования типовых АРМ работников транспортных предприятий;

- действующие макеты АРМ.

Макетные варианты АРМ должны использоваться на тех объектах, где бу-

дет проводиться их экспериментальное внедрение, а также при соответствующей доработке, могут послужить основой для создания соответствующих серийных образцов и их дальнейшего тиражирования.

Задание 4. Разработать экспертно-обучающую систему диспетчеров для автоматизированного составления магистральной схемы транспортировки мелкопартионных грузов в контейнерах. Задание включает:

- проведение технико-экономического обоснования создания и внедрения экспертно-обучающей системы диспетчеров;

- разработку концептуальных подходов к построению экспертно-обучающей системы диспетчеров, позволяющей в интерактивном режиме решать задачи перспективного развития, текущего планирования и оперативного управления контейнерными перевозками;

- разработку проекта технического задания, включающего общую схему экспертно-обучающей системы и схему ее программной реализации;

- разработку экспертно-обучающей системы диспетчеров.

Основными научными и практическими результатами исследований задания 4 являются:

- технико-экономического обоснования создания и внедрения экспертно-обучающей системы диспетчеров;

- первичный состав базовых терминов, понятий и форм представления информации при моделировании задачи составления магистральной схемы транспортировки мелкопартионных грузов в контейнерах;

- техническое задание на разработку экспертно-обучающей системы диспетчеров;

- действующий макет экспертно-обучающей системы диспетчеров.

По результатам исследования возможных подходов к выработке принципов конструирования прототипов экспертно-обучающей системы диспетчеров и к системному представлению знаний в области автоматизации контейнерных перевозок будет сформулирован набор необходимых для последующего использования средств: понятий, терминов, методов и процедур. Эти средства в

соответственно структурированной форме (геоданных) будут скомпонованы для представления на экране персонального компьютера в виде относящихся к отдельным сценариям групп компьютерных «кадров», образующих соответствующий каждому сценарию «виртуальный экран». Программный макет демонстратора разрабатываемой системы, должен включать компьютерные «кадры» и сценарии составления магистральной схемы транспортировки мелкопартионных грузов в контейнерах, а также сценарии для решения задачи выбора структуры магистральной сети.

Задание 5. Провести исследования и разработать математическое обеспечение для решения задач перспективного развития, текущего планирования и оперативного управления контейнерными перевозками мелкопартионных грузов на транспортных магистралях. Задание включает:

- исследование концептуальных моделей сети перевозок мелкопартионных грузов в контейнерах с позиций общей теории систем и программно-целевого подхода к управлению, иерархических теорий многоуровневых систем, теории неформальных структур и связей, теории фреймов и семантических сетей, нейронных сетей, типологии информационных потоков и теории принятия решений;

- исследование и разработку математических моделей, методов, алгоритмов и программ для решения задач перспективного развития, текущего планирования и оперативного управления на всех уровнях иерархии транспортной сети;

- исследование и разработку математических моделей, методов, алгоритмов и программ для решения задач анализа и прогнозирования перевозок грузов в контейнерах.

Основными научными и практическими результатами исследований задания 5 являются:

- модель сети перевозок мелкопартионных грузов в контейнерах;

- математические модели, методы, алгоритмы и программы для решения задач перспективного развития, текущего планирования, оперативного управ-

ления, анализа и прогнозирования перевозок грузов в контейнерах.

Практическая реализация результатов задания 5 возможна после создания и внедрения:

распределенной автоматизированной информационной системы сбора и хранения данных о контейнерных перевозках мелкопартионных грузов;

комплекса средств автоматизации процессов передачи информации и принятия оперативных решений в системе контейнерных перевозок;

автоматизированных рабочих мест (АРМ) работников транспортных предприятий по первичному учету данных о потоках мелкопартионных грузов в сети контейнерных перевозок;

экспертно-обучающей системы диспетчеров для автоматизированного составления магистральной схемы транспортировки грузов в контейнерах.

Приложение Б. Задача распределения и упаковки мелкопартионных потоков корреспонденций в зональных сетях

В первом разделе определены принципы организации сортировки и транспортировки потоков в зонально-узловой структуре сети. Согласно этим принципам узлы второго и третьего типа могут отправлять потоки корреспонденций, адресованные за свою зону обслуживания только через узлы первого типа, лежащие на границе своей зоны. Аналогично в соответствующих узлах первого типа обрабатываются и входящие потоки в узлы второго и третьего типа, поступающие от узлов из других зон обслуживания. Если структура магистральной сети известна (например, изначально задана опытными диспетчерами транспортной сети, администраторами сети передачи данных или определена в результате решения задачи выбора структуры сети, приведенной в подразделе 3.5), то возникает задача оптимального распределения мелкопартионных потоков корреспонденций, исходящих из узлов второго и третьего типа. В настоящем приложении показано, что структурные особенности такой задачи позволяют использовать для получения нижних оценок ее решения метод декомпозиции Данцига-Вулфа [222]. Учитывая, что реальные сетевые структуры всегда

функционируют в условиях неопределенности, воздействия случайных факторов, динамически изменяющихся с течением времени ресурсов, а также при недостаточно точной исходной информации, для практического решения поставленной задачи предложены приближенные алгоритмы, основанные на методах построения кратчайших путей [22, 23].

Б.1. Постановка задачи

Пусть $G(N,P)$ — иерархическая магистральная сеть с множеством неориентированных топологических дуг P, $p=\left|P\right|$, множеством узлов N, $n=\left|N\right|$ и $N=N_1\cup N_2\cup N_3$, где N_1,N_2,N_3 — множества узлов первого, второго и третьего типа соответственно. Под топологической дугой понимается физический отрезок линии связи, соединяющий два любых узла из множества N так, что между рассматриваемыми узлами на данном отрезке нет больше ни одного узла из N. Узлы сети соответствуют пунктам отправления, получения и перегрузки потоков. На сети задана целочисленная матрица потоков $A=\left\| a_{ij} \right\|_{n\times n}$, в которой потоки a_{ii}, $i=\overline{1,n}$ представляют внутренние потоки между узлами четвертого типа в зоне обслуживания i-го узла. Они не подлежат распределению по магистральной сети, но должны учитываться при расчете затрат на сортировку мелкопартионных потоков в узлах первого, второго и третьего типа (предполагается, что число узлов четвертого типа во внутренней зоне каждого i-го узла первого, второго и третьего типа известно). Потоки a_{ij} из источников i в стоки j, $i,j=\overline{1,n}$ должны транспортироваться в транспортных блоках (контейнерах) объема $\omega >> a_{ij}$ и каждый поток может быть упакован в транспортный блок только целиком.

Преобразование потоковой матрицы A должно учитывать принципы направления потоков для узлов второго и третьего типа, согласно которым исходящий поток из этих узлов направляется (рассортировывается) в узлы первого типа, лежащие на границе зон их обслуживания. Пусть $Y=\left\| y_{\xi} \right\|_n$ — вектор, в

котором суммируются дополнительные объемы обработки от преобразованных потоков узлов второго и третьего типов (первоначально Y обнулен); Z_i — зона обслуживания (ЗО), построенная на сети G для i-го узла; l, k — узлы первого типа в ЗО j-го и i-го узлов, обслуживающие соответственно $j \in N_2 \cup N_3$, $i \in N_2 \cup N_3$; S — множество корреспондирующихся пар ij. Тогда содержательная постановка задачи распределения и упаковки потоков для узлов второго и третьего типа в зональных сетях заключается в преобразовании матрицы A, которое выполняется следующим образом:

1. $\forall ij \in S$, $i, j \in N_1$ соответствующие потоки a_{ij} не изменяются;

2. $\forall ij \in S$, $i \in N_1$, $j \in N_2 \cup N_3$, $l \neq i$ выполнить:

$a_{il} \leftarrow a_{il} + a_{ij}$, $a_{lj} \leftarrow a_{lj} + a_{ij}$, $y_l \leftarrow y_l + a_{ij}$, $a_{ij} \leftarrow 0$;

3. $\forall ij \in S$, $i \in N_2 \cup N_3$, $j \in N_1$, $k \neq j$ выполнить:

$a_{ik} \leftarrow a_{ik} + a_{ij}$, $a_{kj} \leftarrow a_{kj} + a_{ij}$, $y_k \leftarrow y_k + a_{ij}$, $a_{ij} \leftarrow 0$;

4. $\forall ij \in S$, $i, j \in N_2 \cup N_3$, $k \neq l$ выполнить:

$a_{ik} \leftarrow a_{ik} + a_{ij}$, $a_{kl} \leftarrow a_{kl} + a_{ij}$, $a_{lj} \leftarrow a_{lj} + a_{ij}$, $y_k \leftarrow y_k + a_{ij}$, $y_l \leftarrow y_l + a_{ij}$,

$a_{ij} \leftarrow 0$;

5. $\forall ij \in S$, $i, j \in N_2 \cup N_3$, $k = l$, $j \notin Z_i$ выполнить п. 3;

6. $\forall ij \in S$, $i, j \in N_2 \cup N_3$, $k = l$, $j \in Z_i$ преобразование a_{ij} не выполняется.

Как видно из постановки задачи для всех корреспонденций, потоки которых подвергаются операции преобразования, необходимо указать узлы k и l. Покажем, что задача определения этих узлов может быть сведена к задаче линейного программирования без учета ограничений на пропускные способности дуг.

Б.2. Математическая модель и алгоритмы решения задачи

Транспортировка или передача потоков из узлов второго и третьего типа до обслуживающих их узлов первого типа и наоборот должна выполняться транспортными средствами магистрального уровня или по магистральным ка-

налам связи, поэтому для таких потоков ограничения на пропускные способности дуг будут учтены в магистральных моделях при распределении потоков между всеми типами узлов в сети. Наиболее существенными факторами, определяющими качество схемы распределения потоков узлов второго и третьего типа, являются затраты на их транспортировку или передачу от узла-источника до узла-потребителя и возможности обработки потоков этих узлов в узлах первого типа. Поскольку в рассматриваемой задаче производится только адресация потоков из узлов второго и третьего типа в некоторые узлы первого типа, лежащие на границе зон обслуживания первых, а фактическое распределение потоков с учетом их объемов и других ограничений осуществляется при решении задачи распределения потоков на магистральном уровне, то транспортные затраты можно считать не зависящими от объемов потоков и принять линейными от расстояния. Примем также, что затраты на дополнительную обработку (пересортировку) единицы потока из узлов второго и третьего типа в обслуживающих их узлах первого типа одинаковы во всех узлах.

Пусть $R = \| r_{ij} \|_{n \times n}$ — топологическая матрица, где

$$r_{ij} = \begin{cases} \text{длине дуги, если } p_{ij} \in P, \\ \infty, \text{ если дуги } p_{ij} \text{ не существует.} \end{cases}$$

Рассмотрим сеть $G_D(N, P_D)$, полученную из $G(N, P)$ следующим образом. Определим на сети G_D, используя матрицу R, кратчайшие пути от всех узлов $i \in N_2 \cup N_3$ до узлов первого типа, находящихся в ЗО i-го узла и кратчайшие пути между всеми узлами первого типа. В результате получим матрицу $D = \| d_{ij} \|_{n \times n}$, в которой элементы не равные бесконечности представляют множество дуг P_D сети G_D.

Лемма Б.1. Пусть выполняется одно из условий:

$$(i, j \in N_1 \wedge d_{ij} \neq \infty) \vee ((i \in N_1, j \in N_2 \cup N_3) \wedge d_{ij} \neq \infty) \vee$$

$$\vee ((i \in N_2 \cup N_3, j \in N_1) \wedge d_{ij} \neq \infty \qquad \text{(Б.1)}$$

$$(i \in N_1, j \in N_2 \cup N_3) \wedge d_{ij} = \infty \qquad \text{(Б.2)}$$

$$(i \in N_2 \cup N_3, j \in N_1) \wedge d_{ij} = \infty \qquad \text{(Б.3)}$$

$$(i, j \in N_2 \cup N_3) \wedge d_{ij} = \infty \wedge j \notin Z_i \qquad \text{(Б.4)}$$

тогда: 1) кратчайший путь между любыми узлами, для которых выполняется условие (Б.1), состоит из одной дуги и совпадает с кратчайшим путем между этими узлами, построенными на сети G; 2) кратчайший путь между узлами, для которых выполняется (Б.2) или (Б.3) состоит из двух дуг; 3) кратчайший путь между узлами, удовлетворяющими условию (Б.4) состоит из двух или трех дуг.

Доказательство. Для доказательства 1) покажем, что длина любого пути, построенного на сети G_D между узлами $i, j \in N_1$ и содержащего более одной дуги не меньше, чем длина пути, состоящего из одной дуги. Предположим, что найден кратчайший путь π_{ij} от i до j, содержащий более одной дуги. Пусть π_{ij} задан перечислением узлов $\pi_{ij} = (i, k_1, k_2, ..., k_\xi, j)$. Тогда длина любой дуги (i, k_1), $(k_1, k_2), \ldots, (k_\xi, j)$ в сети G_D определяется суммой длин некоторого подмножества топологических дуг сети G. Если все топологические дуги, составляющие отрезки $(i, k_1), \ldots, (k_\xi, j)$ являются подмножеством множества топологических дуг, входящих в путь π_{ij}^*, построенный на сети G и представленный в сети G_D одной дугой, то длины путей π_{ij} и π_{ij}^* совпадают. В случае невыполнения последнего условия, длина пути π_{ij} не может быть меньше длины π_{ij}^*, так как тогда путь π_{ij}^* не был бы кратчайшим. Полученное противоречие и тот факт, что в алгоритме построения кратчайших путей, среди нескольких возможных кратчайших путей между заданными узлами, первым всегда будет определен путь, состоящий из меньшего числа дуг, доказывает утверждение 1). Доказательство 2) и 3) следует из правил формирования ЗО для узлов, построения сети G_D, а также из рассуждений, аналогичных приведенным для доказательства утверждения 1). ■

Следует отметить, что для случаев (Б.2)-(Б.4) длины кратчайших путей между узлами i и j, построенных в сети G и G_D могут не совпадать. В случае

(Б.4), когда $j \in Z_i$ имеют место внутри - зональные корреспонденции, потоки которых не подвергаются дополнительной переработке в узлах первого типа, поэтому они не рассматриваются при формировании матрицы A.

Будем далее считать, что каждая неориентированная дуга $p \in P_D$ заменена на две дуги с противоположной ориентацией и p_D означает число всех ориентированных дуг в G_D.

Пусть $X^i = \| x_j^i \|^T$, $j = \overline{1, p_D}$, $i = \overline{1,n}$ — вектор-столбец, определяющий потоки корреспонденций, исходящих из узла i по всем дугам сети G_D; $A = \| a_{ij} \|_{n \times n}$ — матрица потоков, в которой сохранены только те элементы a_{ij}, для индексов которых справедливо одно из условий (Б.2)-(Б.4). Остальные элементы в матрице A заменены на нули. Обозначим через $C^i = \| c_j^i \|$, $j = \overline{1, p_D}$, $i = \overline{1,n}$ вектор-строку затрат на транспортировку единицы потока корреспонденций из i по дугам сети G_D. Построим для G_D матрицу инциденций $E = \| e_{ij} \|_{n \times p_D}$ узлы-дуги и определим матрицы $W^\xi = \| w_{ij}^\xi \|_{n \times p_D}$, $\xi = \overline{1,n}$; векторы-столбцы $V^i = \| v_j^i \|^T$, $j = \overline{1,n}$, $i = \overline{1,n}$; вектор-столбец пропускных способностей узлов $H^W = \| h_i \|^T$, $i = \overline{1,n}$. Знак «^{T}» означает транспонирование. Где соответственно:

$$e_{ij} = \begin{cases} 1, \text{если дуга } j \text{ направлена к узлу } i, \\ -1, \text{если дуга } j \text{ направлена от узла } i, \\ 0 \text{ в противном случае;} \end{cases} \quad w_{ij}^\xi = \begin{cases} 1, \text{если } e_{ij} = 1, \\ 1, \text{если } e_{ij} = -1 \text{ и } i = \xi, \\ 0 \text{ в противном случае;} \end{cases}$$

$$v_j^i = \begin{cases} -\sum_{\xi=1}^{n} a_{ij}, \text{если } j = i, \\ a_{ij}, \text{если } j = \xi. \end{cases}$$

Решение задачи распределения потоков в зональных сетях будем искать в классе задач линейного программирования вида:

$$Min\, Z = C^1[p_D]X^1[p_D] + \ldots + C^n[p_D]X^n[p_D], \qquad (Б.5)$$

$$W^1[n, p_D]X^1[p_D] + \ldots + W^n[n, p_D]X^n[p_D] \le H^w[n], \qquad (Б.6)$$

$$E[n, p_D]X^1[p_D] \qquad = V^1[n],$$

$$\ldots \qquad \text{(Б.7)}$$

$$E[n, p_D]X^n[p_D] = V^n[n],$$

$$X^i[p_D] \geq 0 \text{ и целые, } i = \overline{1,n}. \qquad \text{(Б.8)}$$

Рассмотрим особенности сформулированной задачи. Из утверждения леммы ясно, что для решения блочных задач можно использовать эффективные алгоритмы построения кратчайших путей по одному критерию — минимуму длины пути. Это следует из того, что любой путь для корреспонденций из класса (Б.2) или (Б.3) будет состоять из двух дуг, а для корреспонденций класса (Б.4) — из двух или трех дуг. Поэтому выбор оптимального пути будет всегда производиться среди путей, содержащих одинаковое число дуг, т.е. все пути эквивалентны с точки зрения числа транзитных узлов. Другая особенность задачи (Б.5)-(Б.8) заключается в том, что ограничения на пропускные способности узлов необходимо учитывать только для узлов первого типа, связанных с узлами второго и третьего типов в сети G_D хотя бы одной дугой. Верхняя граница числа учитываемых ограничений (Б.6) равна величине n_1 — числу узлов во множестве N_1. Последнее обстоятельство обеспечивает относительно невысокую размерность задачи (Б.5)-(Б.8) и позволяет использовать для получения нижних оценок ее решения метод декомпозиции Данцига-Вулфа и другие методы целочисленного программирования [222, 175].

Отметим, что если не учитывать ограничения (Б.6), то полученное решение при использовании метода Данцига-Вулфа будет всегда целочисленным, поскольку матрицы ограничений в приведенной модели являются вполне унимодулярными, а правые части ограничений целочисленными векторами и выбор оптимального решения осуществляется на многограннике с целочисленными вершинами.

Поскольку реальные сети функционируют при динамически изменяющихся с течением времени ресурсов и недостаточно точной исходной информации, для практического решения задачи можно отказаться от точных методов и ис-

пользовать алгоритмы, основанные на методах построения кратчайших путей. Приведем алгоритмы, позволяющие решить задачу в случаях заданных и неизвестных зон обслуживания узлов.

Пусть $C^* = \| c_{ij}^* \|_{n \times n}$ — справочная матрица кратчайших путей, построенная для сети G_D, где

$$c_{ij}^* = \begin{cases} 0, \text{если} (i = j) \vee (i, j \in N_2 \cup \mathrm{N}_3 \wedge j \in Z_i), \\ i, \text{если} \, i \text{ и } j \text{ связаны одной дугой}, \\ k, \text{если путь от } i \text{ до } j \text{ содержит более одной дуги}, \end{cases} \qquad (Б.9)$$

где k — предпоследний узел на кратчайшем пути от i до j; B_i^1 — множество узлов первого типа в Z_i; $\{\alpha \,|\, \beta\}$ — множество α, для которых верно утверждение β.

Известно [10], что для построенных кратчайших путей справедливо утверждение $\forall i \in N \, \exists \{\pi_{ij} \,|\, j \in N \setminus i\}$, такое, что в $\forall j$ входит не более одной дуги, принадлежащей любому кратчайшему пути π_{ij}. Таким образом, кратчайшие пути π_{ij}, построенные от узла i представляют собой дерево с корнем в i, которое может быть однозначно описано строкой справочной матрицы C^*.

Алгоритм Б.1. Распределение потоков при заданных ЗОУ

1. Используя матрицу R построить в сети G кратчайшие пути от $\forall i \in N_2 \cup N_3$ до $\forall j \in B_i^1$ и между всеми узлами из множества N_1.

2. Используя матрицу D, построенную в п. 1, найти кратчайшие пути в сети G_D. При построении путей сформировать справочную матрицу C^* в соответствии с (Б.9).

3. $C \leftarrow 0$; $Y \leftarrow 0$. *** C — справочная матрица объединения потоков

4. Для $\{i, j \,|\, (c_{ij}^* = i \vee c_{ij}^* = 0) \wedge i \neq j, \; i, j = \overline{1,n}\}$ выполнить $c_{ij} \leftarrow i$.

5. Для $\{i, j \,|\, c_{ij}^* \neq i \vee c_{ij}^* \neq 0, \; i, j = \overline{1,n}\}$ выполнить пп. 6-12.

6. $\xi \leftarrow j$; $l \leftarrow 0$; $k \leftarrow 0$.

7. Пока $c_{i\xi}^* \neq i$ выполнить пп. 8-10.

8. Если $l = 0$, то $l \leftarrow c^*_{i\xi}$.

9. $k \leftarrow c^*_{i\xi}$.

10. $\xi \leftarrow c^*_{i\xi}$. Перейти к п. 7.

11. Если $k = l$, то выполнить преобразование

$a_{ik} \leftarrow a_{ik} + a_{ij}$, $a_{kj} \leftarrow a_{kj} + a_{ij}$, $y_k \leftarrow y_k + a_{ij}$, $a_{ij} \leftarrow 0$, иначе выполнить

преобразование $a_{ik} \leftarrow a_{ik} + a_{ij}$, $a_{kl} \leftarrow a_{kl} + a_{ij}$, $a_{lj} \leftarrow a_{lj} + a_{ij}$, $y_k \leftarrow y_k + a_{ij}$,

$y_l \leftarrow y_l + a_{ij}$, $a_{ij} \leftarrow 0$.

12. $c_{ij} \leftarrow k$. Перейти к п. 5.

13. Стоп.

Запись в строках 4 и 5 выражение $i, j = \overline{1,n}$ означает циклическое изменение индексов, причем второй индекс изменяется быстрее первого. Такая запись аналогична организации внутреннего цикла по j и внешнего по i, когда оба индекса изменяются от единицы до n с единичным шагом.

В результате работы алгоритма Б.1 формируются матрицы A, C и вектор Y. В векторе Y суммируются дополнительные объемы обработки от преобразованных потоков узлов второго и третьего типа (первоначально C и Y обнулены). Элементы справочной матрицы объединения потоков (сортировки) $C = \| c_{ij} \|_{n \times n}$ определяются следующим образом:

$$c_{ij} = \begin{cases} i, \text{если поток } a_{ij} \text{ адресуется в узел } j, \\ k, \text{если поток } a_{ij} \text{ адресуется в узел } k, \\ 0, \text{если } i = j. \end{cases}$$

Матрица C используется для запоминания схемы адресации (сортировки) потоков в узлах второго и третьего типов.

Для работы алгоритма Б.1 необходимо знать множества B^1_i, $i \in N_2 \cup N_3$. Ранее отмечалось, что множества B^1_i определяются методом экспертного анализа с привлечением опытных специалистов или автоматизированным способом после решения задачи выбора структуры сети. В случае если процедура уточ-

нения структуры сети экспертами по каким-либо причинам выполнена не была, может оказаться, что зоны обслуживания для узлов второго и третьего типа точно не известны. Поэтому целесообразно иметь алгоритм распределения потоков в зональных сетях, если задан только вектор $\theta = \|\theta_i\|_n$, определяющий типы узлов, где $\theta_i = 1$, если $i \in N_1$ и $\theta_i = 0$, если $i \in N_2 \cup N_3$.

Алгоритм Б.2. Распределение потоков при неизвестных ЗОУ

1. Используя матрицу R построить на сети G кратчайшие пути между всеми узлами с одновременным формированием справочной матрицы C^* согласно (Б.9).

2. $C \leftarrow 0$; $Y \leftarrow 0$. *** C — справочная матрица объединения потоков

3. Для $\{i \mid i = \overline{1,n}\}$ выполнить пп. 4-23.

4. Для $\{j \mid j = \overline{1,n}\}$ выполнить пп. 5-22.

5. Если $i \neq j$, перейти к п. 6, иначе перейти к п. 22.

6. Если $\theta_i = 1 \wedge \theta_j = 1$, выполнить $c_{ij} \leftarrow i$; перейти к п. 22.

7. $\xi \leftarrow j$; $l \leftarrow 0$; $k \leftarrow 0$.

8. Пока $c^*_{i\xi} \neq i$ выполнить пп. 9-12.

9. Если $\theta_{c^*_{i\xi}} = 1$ выполнить пп. 10-11, иначе перейти к п. 12.

10. Если $l = 0$, то $l \leftarrow c^*_{i\xi}$.

11. $k \leftarrow c^*_{i\xi}$.

12. $\xi \leftarrow c^*_{i\xi}$. Перейти к п. 8.

13. Если $\theta_i = 0$, то перейти к п. 14, иначе к п. 21.

14. Если $\theta_j = 0$, то перейти к п. 15, иначе к п. 19.

15. Если $l \neq 0$, то перейти к п. 16, иначе $c_{ij} \leftarrow i$ и перейти к п. 22.

16. Если $l \neq k$, то выполнить п. 17, иначе перейти к п. 18.

17. $a_{kl} \leftarrow a_{kl} + a_{ij}$, $a_{lj} \leftarrow a_{lj} + a_{ij}$, $y_l \leftarrow y_l + a_{ij}$,

L1: $a_{ik} \leftarrow a_{ik} + a_{ij}$, $y_k \leftarrow y_k + a_{ij}$, $c_{ij} \leftarrow k$,

L2: $a_{ij} \leftarrow 0$, перейти к п. 22.

18. Выполнить

L3: $a_{il} \leftarrow a_{il} + a_{ij}$, $a_{lj} \leftarrow a_{lj} + a_{ij}$, $y_l \leftarrow y_l + a_{ij}$, $c_{ij} \leftarrow l$, перейти к L2.

19. Если $k = 0$, то $c_{ij} \leftarrow i$, перейти к п. 22.

20. $a_{kj} \leftarrow a_{kj} + a_{ij}$. Перейти к L1.

21. Если $l = 0$, то $c_{ij} \leftarrow i$, иначе перейти к L3.

22. Перейти к п. 4. *** конец цикла по j

23. Перейти к п. 3. *** конец цикла по i

24. Стоп.

В алгоритме Б.2 указатели k и l определяют соответственно первый и последний узлы первого типа, находящиеся на кратчайшем пути между i и j. Если между узлами i и j нет узлов первого типа, либо нет вообще никаких узлов, значения $k = l = 0$.

Сравнивая трудоемкость алгоритмов Б.1 и Б.2 с трудоемкостью решения задачи (Б.5)-(Б.8) отметим, что использование простых процедур, основанных на построении кратчайших путей позволяет указать верхнюю границу асимптотической трудоемкости — $O(n^3)$, в то время как точное решение целочисленной задачи (Б.5)-(Б.8) может иметь экспоненциальную оценку [45].

Резюмируя обсуждение задачи распределения потоков в зональных сетях, отметим, что она может быть сформулирована как задача линейного программирования с блочной структурой ограничений и для ее решения применим метод декомпозиции Данцига-Вулфа и другие методы целочисленного программирования. Для решения задачи на реальных сетях предложены приближенные алгоритмы, основанные на построении кратчайших путей. Верхняя граница временной сложности приближенных алгоритмов составляет $O(n^3)$, в то время как точное решение целочисленной задачи может иметь экспоненциальную оценку. Поэтому преимущество при решении поставленной задачи следует отдать приближенным алгоритмам в силу их простоты, небольшой трудоемкости

и, как показали практические исследования, приемлемой точности решения.

Приложение В. Задача развозки порожних контейнеров

При перевозке мелкопартионных грузов в жесткой таре (контейнерах) в многопродуктовой транспортной сети возникает задача балансировки матрицы потоков [24, 25]. Поскольку не для всех узлов сети выполняется условие баланса — равенства суммы исходящих и входящих контейнеров, актуальной становится задача развозки порожних контейнеров (РПК). Часто на практике балансировка матрицы потоков для каждой корреспондирующейся пары узлов осуществляется путем возврата порожних контейнеров в узел с их недостатком, т.е. имеет место симметричная балансировка, при которой учет рабочего парка контейнеров ведется одним из этих узлов. При наличии информационно-аналитической системы (ИАС) управления процессами обработки и перевозки грузов выгоднее вести централизованный учет рабочего парка контейнеров и для балансировки потоков решать задачу оптимизации РПК. При этом появляется возможность значительного снижения приведенных затрат на обработку и перевозку потоков порожних контейнеров и оперативного управления их перераспределением при колебании нагрузок в сети.

Целью настоящего изложения является построение математической модели, выбор методов и алгоритмов для решения задачи оптимизации РПК и экспериментальное сравнение методов симметричной и оптимальной балансировки матрицы потоков на численных примерах.

В.1. Постановка и математическая модель задачи

Пусть $G(N,P)$ — неориентированная транспортная сеть с множеством узлов N, $n = |N|$ и множеством дуг P, $p = |P|$. Узлы сети соответствуют пунктам отправления, получения и перегрузки груженых и порожних контейнеров, а дуги — физическим отрезкам линий связи.

Потоки груженых контейнеров заданы матрицей $U = \| u_{ij} \|_{n \times n}$, полученной

после решения задачи упаковки и выбора структуры сети. Потоки u_{ij} из источников i в стоки j, $i, j = \overline{1,n}$ должны перевозиться в транспортных средствах с заданной периодичностью. Матрица U не сбалансирована, поскольку для большинства узлов не выполняется условие

$$\Delta_i = \sum_{j=1}^{n} (u_{ij} - u_{ji}) = 0,\ i = \overline{1,n}. \qquad \text{(В.1)}$$

Первый способ балансировки предусматривает построение симметричной матрицы $U^c = \| u_{ij}^c \|_{n \times n}$, каждый элемент которой определяется из выражения $u_{ij}^c = \max\{u_{ij}, u_{ji}\}$, $i, j = \overline{1,n}$. Такое построение приводит к дополнительным затратам на перевозку и погрузку/выгрузку порожних контейнеров, однако, упрощает организацию их обработки и отправки. Второй способ балансировки заключается в решении транспортной задачи для развозки порожних контейнеров.

Разделим все множество узлов, для которых не выполняется условие (В.1), на два подмножества A и B так, чтобы выполнялось $\Delta_i < 0$, $i \in \mathrm{A}$, $\Delta_i > 0$, $i \in \mathrm{B}$ и $\mathrm{A} \cup \mathrm{B} \subseteq \mathrm{N}$. Тогда в терминах транспортной задачи множество B является множеством потребителей порожних контейнеров, а A — множеством их поставщиков. Образуем вектор потребителей b_j, $j = \overline{1,k}$ с объемами потребления $b_j = \Delta_\xi$, для $\forall\ \xi \in \mathrm{B}$ и вектор поставщиков a_i, $i = \overline{1,l}$, с объемами предложения $a_i = \left| \Delta_\xi \right|$, для $\forall\ \xi \in \mathrm{A}$. Где k и l получены счетным перечислением множеств B и A соответственно. Векторам a_i и b_j поставим во взаимно однозначное соответствие векторы номеров узлов α_i и β_j из A и B. Сформулируем транспортную задачу. Требуется минимизировать

$$\sum_{i=1}^{l} \sum_{j=1}^{k} f_{ij}(x_{ij}, d_{ij}), \qquad \text{(В.2)}$$

при ограничениях

$$\sum_{j=1}^{k} x_{ij} = a_i,\ i = \overline{1,l}, \qquad \text{(В.3)}$$

$$\sum_{i=1}^{l} x_{ij} = b_j, \ \ j = \overline{1,k}, \tag{B.4}$$

$$\sum_{i=1}^{l} a_i = \sum_{j=1}^{k} b_j \ \ , \tag{B.5}$$

где $x_{ij} \geq 0$ — число порожних контейнеров, отправляемых i-м поставщиком j-му потребителю; d_{ij} — расстояние перевозки от i до j; $f_{ij}(\cdot)$ — кусочно-выпуклые нелинейные функции транспортных затрат, зависящие в общем случае от перевозимого объема, расстояния, типа транспортного средства, выделенного для перевозки, маршрута перевозки, скорости движения и других факторов. Учет всех перечисленных факторов при решении задачи (B.2)-(B.5) не представляется возможным, поскольку перевозка порожних контейнеров должна выполняться вместе с груженными. Маршруты перевозки определяются при решении отдельной задачи распределения и маршрутизации потоков (см. раздел 4). Поэтому предположим, что на этапе балансировки матрицы контейнерных потоков транспортные затраты на перевозку порожних контейнеров пропорциональны объему и расстоянию. Такое допущение позволяет не учитывать нелинейность функций $f_{ij}(\cdot)$ и интерпретировать (B.2)-(B.5) как задачу минимизации «пробега» порожних контейнеров, подставив вместо $f_{ij}(\cdot)$ длины кратчайших путей $\tilde{D}^* = \| \tilde{d}_{ij}^* \|_{l \times k}$ из i в j. Предполагается, что предварительно на сети G рассчитаны кратчайшие пути $D^* = \| d_{ij}^* \|_{n \times n}$, а $\tilde{d}_{ij}^* = d_{\alpha_i \beta_j}^*$, $i = \overline{1,l}$, $j = \overline{1,k}$. Отметим, что в (B.2) расстояния d_{ij} могут отличаться от расстояний по кратчайшим путям d_{ij}^*.

Задача преобразуется к линейному виду: минимизировать

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \tilde{d}_{ij}^* x_{ij} \tag{B.6}$$

при ограничениях (B.3)-(B.5). После решения задачи, балансировка осуществляется следующим образом: $u_{ij}^o = u_{ij}$, $i, j = \overline{1,n}$; $u_{\alpha_i \beta_j}^o = u_{\alpha_i \beta_j} + x_{ij}$, $i = \overline{1,l}$, $j = \overline{1,k}$, где u_{ij}^o — элементы сбалансированной матрицы $U^o = \| u_{ij}^o \|_{n \times n}$.

Пусть имеются операторы $\varphi^o : X = \| x_{ij} \|_{l\times k} \Rightarrow X^o = \| x^o_{ij} \|_{n\times n}$, где $x^o_{ij} = x_{\alpha_i\beta_j}$, $i, j = \overline{1,n}$; $\varphi^c : U = \| u_{ij} \|_{n\times n} \Rightarrow X^c = \| x^c_{ij} \|_{n\times n}$, где x^c_{ij} определяются из условия: если $u_{ij} \geq u_{ji}$, то $x^c_{ji} = u_{ij} - u_{ji}$, иначе $x^c_{ij} = u_{ji} - u_{ij}$, $i = \overline{1,n-1}$, $j = \overline{i+1,n}$.

Сравним способы балансировки с точки зрения затрат на перевозку и обработку порожних контейнеров. Поскольку на этапе решения задачи РПК, в силу сделанных допущений, говорить о реальных затратах не приходится, будем сравнивать два способа балансировки, используя следующие выражения для определения транспортных и погрузочно-разгрузочных затрат:

$$Z^c_{тр} = \sum_{i=1}^{n}\sum_{j=1}^{n} f^*_{ij}(x^c_{ij}, d^*_{ij}),\ Z^c_{погр} = f^*_{конт}\left(2\sum_{i=1}^{n}\sum_{j=1}^{n} x^c_{ij} \right), \qquad \text{(В.7)}$$

$$Z^o_{тр} = \sum_{i=1}^{n}\sum_{j=1}^{n} f^*_{ij}(x^o_{ij}, d^*_{ij}),\ Z^o_{погр} = f^*_{конт}\left(2\sum_{i=1}^{n}\sum_{j=1}^{n} x^o_{ij} \right), \qquad \text{(В.8)}$$

где $Z^c_{тр}$, $Z^c_{погр}$, $Z^o_{тр}$, $Z^o_{погр}$ — соответственно транспортные и погрузочно-разгрузочные затраты при симметричной балансировке и балансировке, основанной на оптимизации развозки порожних контейнеров; $f^*_{ij}(\cdot)$ — некоторые функции транспортных затрат, зависящие от объема и расстояния перевозки; $f^*_{конт}(\cdot)$ — функция затрат от объема погружаемых и выгружаемых контейнеров. В качестве $f^*_{ij}(\cdot)$ и $f^*_{конт}(\cdot)$ могут выступать функции среднегодовых приведенных затрат, учитывающие динамику перевозок и зависящие от рабочего парка транспортных средств и контейнеров, необходимых для нормального функционирования сети при колебаниях нагрузок в сети (см. подраздел 4.6).

Величина снижения затрат

$$\Delta Z = \Delta Z_{тр} + \Delta Z_{погр} = (Z^c_{тр} - Z^o_{тр}) + (Z^c_{погр} - Z^o_{погр}) \qquad \text{(В.9)}$$

может служить оценкой оптимального плана развозки порожних контейнеров.

Отметим несколько особенностей при расчете оценки ΔZ на следующем примере.

Пусть $U = \begin{Vmatrix} 0 & 10 & 10 \\ 5 & 0 & 20 \\ 5 & 10 & 0 \end{Vmatrix}$ — матрица потоков груженых контейнеров, а сеть имеет вид, показанный на рис. В.1. В третьем узле сети образуется избыток порожних контейнеров — 15, а в первом и втором их дефицит — 10 и 5 соответственно. При симметричной балансировке перевозка порожних контейнеров выполняется, как показано на рис. В.2, а при оптимальной — как на рис. В.3.

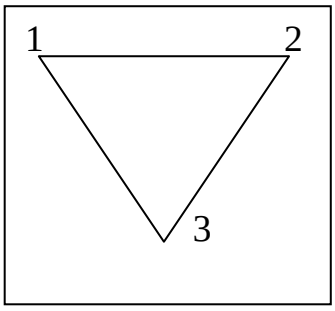


Рис. В.1

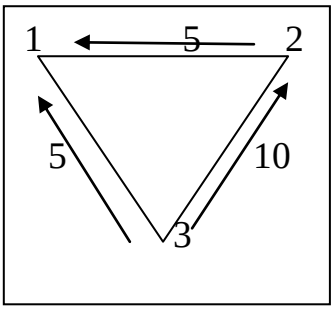


Рис. В.2

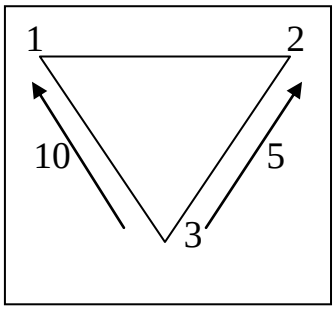


Рис. В.3

Соответственно матрицы X^c, X^o и U^c, U^o будут иметь следующий вид:

$$X^c = \begin{Vmatrix} 0 & 0 & 0 \\ 5 & 0 & 0 \\ 5 & 10 & 0 \end{Vmatrix},\ X^o = \begin{Vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 10 & 5 & 0 \end{Vmatrix},\ U^c = \begin{Vmatrix} 0 & 10 & 10 \\ 10 & 0 & 20 \\ 10 & 20 & 0 \end{Vmatrix},\ U^o = \begin{Vmatrix} 0 & 10 & 10 \\ 5 & 0 & 20 \\ 15 & 15 & 0 \end{Vmatrix}.$$

Пусть π_{21}, π_{31}, π_{32}, — пути перевозки, заданные последовательностью составляющих их дуг, а r_{21}, r_{31}, r_{32} и d_{21}, d_{31}, d_{32}, — соответственно длины дуг и путей в единицах расстояния. Предположим, что затраты на перевозку одного порожнего контейнера на единицу расстояния одинаковы на всех путях и равны $c_{тр}$ у.е., а затраты на погрузку (выгрузку) порожнего контейнера равны $c_{погр}$ у.е. Тогда для нашего примера

$$Z^c_{тр} = c_{тр}(5d_{21} + 5d_{31} + 10d_{32}),\ Z^c_{погр} = 2c_{погр}(5+5+10) = 40c_{погр},$$

$$Z^o_{тр} = c_{тр}(10d_{31} + 5d_{32}),\ Z^o_{погр} = 2c_{погр}(10+5) = 30c_{погр}.$$

Очевидно, что сокращение затрат на погрузочно-разгрузочные работы составит (в %)

$$\Delta Z_{погр} = 100(Z^{c}_{погр} - Z^{o}_{погр}) / Z^{c}_{погр} = 100(40-30)/40 = 25\ \%.$$

Для транспортных затрат запишем

$$\Delta Z^{+}_{тр} = 100(Z^{c}_{тр} - Z^{o}_{тр}) / Z^{c}_{тр} = 100(d_{21} + d_{32} - d_{31})/(d_{21} + d_{31} + 2d_{32}).$$

При $d_{31} \geq d_{21} + d_{32}$ величина $\Delta Z^{+}_{тр}$ может оказаться неположительной. Пусть, например, r_{21}=5, r_{31}=10, r_{32}=5, а длины d_{21}, d_{31}, d_{32} рассчитаны по кратчайшим путям по критерию минимального расстояния: $\pi_{31} = \{r_{31}\} \vee \{r_{32}, r_{21}\}$, d_{31}=10; $\pi_{21} = \{r_{21}\}$, d_{21}=5; $\pi_{32} = \{r_{32}\}$, d_{32}=5. Можно убедиться, что при любом увеличении d_{31} и уменьшении d_{21}, d_{32}, пока $d_{31} \geq d_{21} + d_{32}$, $\Delta Z^{+}_{тр} = 100\% \cdot (d_{21} + d_{32} - d_{31})/(d_{21} + d_{31} + 2d_{32}) = 0$. Если длины путей рассчитываются по критерию минимума дуг в пути, то возможно увеличение транспортных затрат при оптимальной перевозке порожних контейнеров. Пусть $\pi_{31} = \{r_{31}\}, d_{31} = r_{31}$=20; $\pi_{21} = \{r_{21}\}$, $d_{21} = r_{21}$=1; $\pi_{32} = \{r_{32}\}$, $d_{32} = r_{32}$=1. Получим

$$Z^{c}_{тр} = c_{тр}(5d_{21} + 5d_{31} + 10d_{32}) = c_{тр}(5 \cdot 1 + 5 \cdot 20 + 10 \cdot 1) = 115c_{тр},$$

$$Z^{o}_{тр} = c_{тр}(10d_{31} + 5d_{32}) = c_{тр}(10 \cdot 20 + 5 \cdot 1) = 205c_{тр}.$$

Увеличение транспортных затрат составит

$\Delta Z^{-}_{тр} = 100\left(\dfrac{Z^{o}_{тр}}{Z^{c}_{тр}} - 1\right) = 100\left(\dfrac{2d_{31} + d_{32}}{d_{21} + d_{31} + 2d_{32}} - 1\right) = 100\left(\dfrac{41}{23} - 1\right) = 78{,}26\%$, что может превысить экономию на погрузочно-разгрузочных затратах.

На основании изложенного можно сделать следующий вывод: если в планах распределения порожних контейнеров при симметричной и оптимальной балансировке при распределении потоков по кратчайшим путям по критерию минимума длины пути или минимума дуг в пути имеются подобные слагаемые (в примере $5d_{31}$, $10d_{32}$ в $Z^{c}_{тр}$ и $10d_{31}$, $5d_{32}$ в $Z^{o}_{тр}$), то при вычислении (В.9) не для всех слагаемых транспортных затрат может получиться положительный эффект. Такая ситуация возникает когда для некоторых фрагментов транспортной сети не выполняется условие неравенства треугольника. На практике редко случается, что неравенство треугольника нарушается для многих фрагментов

сети, но следует учитывать возможность уменьшения величины снижения затрат ΔZ на перевозку порожних контейнеров при решении самостоятельной задачи распределении и маршрутизации полных потоков в сети, когда будут рассчитываться фактические пути перевозки потоков и определяться значения d_{ij} в (В.2).

В.2. Алгоритм и методы решения задачи

При реализации алгоритма решения задачи нет необходимости создавать и хранить матрицы X^c и X^o, поэтому формулы (В.7) и (В.8) можно записать в виде

$$Z^c_{mp} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} f^*_{ij}(abs(u_{ij} - u_{ji}), d^*_{ij}),\ Z^c_{погр} = f^*_{конт}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} abs(u_{ij} - u_{ji}) \right), \quad (В.10)$$

$$Z^o_{mp} = \sum_{i=1}^{l}\sum_{j=1}^{k} f^*_{ij}(x_{ij}, \tilde{d}^*_{ij}),\ Z^o_{погр} = f^*_{конт}\left(2\sum_{i=1}^{l}\sum_{j=1}^{k} x_{ij} \right), \quad (В.11)$$

где *abs* означает абсолютную величину.

Алгоритм В.1. Алгоритм решения задачи РПК

1. На сети G найти матрицу кратчайших путей $D^* = \| d^*_{ij} \|_{n\times n}$.

2. $k \leftarrow 0;\ l \leftarrow 0$.

3. Для $\{ i \mid i = \overline{1,n} \}$ выполнить пп. 4-11.

4. $SO \leftarrow 0;\ SI \leftarrow 0$.

5. Для $\{ j \mid j = \overline{1,n},\ j \neq i \}$ выполнить пп. 6-7.

6. $SO \leftarrow SO + u_{ij};\ SI \leftarrow SI + u_{ji}$.

7. Перейти к п. 5. *** Конец цикла по j

8. Если $SO > SI$, то перейти к п. 9, иначе перейти к п. 10.

9. $k \leftarrow k+1;\ b_k \leftarrow (SO - SI);\ \beta_k \leftarrow i$; перейти к п. 12.

10. Если $SO < SI$, то перейти к п. 11, иначе перейти к п. 12.

11. $l \leftarrow l+1;\ a_l \leftarrow (SI - SO);\ \alpha_l \leftarrow i$.

12. Перейти к п. 3. *** Конец цикла по i

13. По формулам (В.10) определить Z^c_{mp} и $Z^c_{погр}$.

14. Для $\{i, j \mid i = \overline{1,l},\ j = \overline{1,k}\}$ выполнить пп. 15-16.

15. $\tilde{d}^*_{ij} \leftarrow d^*_{\alpha_i \beta_j}$.

16. Перейти к п. 14. *** Конец циклов по j и i

17. Решить транспортную задачу (В.6), (В.3)-(В.5).

18. По формулам (В.11) определить $Z^o_{тр}$ и $Z^o_{погр}$.

19. Для $\{i, j \mid i = \overline{1,l},\ j = \overline{1,k},\ x_{ij} \neq 0\}$ выполнить пп. 20-21.

20. $u_{\alpha_i \beta_j} \leftarrow u_{\alpha_i \beta_j} + x_{ij}$.

21. Перейти к п. 19. *** Конец циклов по j и i

22. По формуле (В.9) получить оценку ΔZ оптимального плана распределения порожних контейнеров.

23. Конец алгоритма.

Оценивая временную сложность приведенного алгоритма, отметим, что наиболее трудоемкими действиями в нем являются нахождение кратчайших путей и решение транспортной задачи (ТЗ). Для нахождения всех кратчайших путей в сети G например с помощью алгоритма Флойда (Floyd R.W., 1962 г.) потребуется $O(n^3)$ времени. Временная сложность решения транспортной задачи зависит от применяемых методов и алгоритмов.

Впервые транспортная задача в строгой математической постановке была сформулирована Хичкоком [305] и исследована Купмансом [350]. Первым точным методом решения классической транспортной задачи был метод потенциалов, предложенный Л.В. Канторовичем и М.К. Гавуриным [63]. С тех пор появилось множество работ, связанных с постановкой и решением транспортных задач. Среди них можно выделить [39, 42], посвященные классификации и детальному рассмотрению существующих методов и алгоритмов решения транспортной задачи.

Большинство методов решения классической транспортной задачи основано на решении двойственной задачи — найти $\max\left(\sum_{i=1}^{l} a_i u_i + \sum_{j=1}^{k} b_j v_j \right)$ при

$u_i + v_j \le c_{ij}$, $i = \overline{1,l}$, $j = \overline{1,k}$, где u_i, v_j не ограничены в знаке, а c_{ij}, — неотрицательные коэффициенты целевой функции прямой задачи: найти $\min \sum_{i=1}^{l} \sum_{j=1}^{k} c_{ij} x_{ij}$ при ограничениях (В.3)-(В.5).

В методе потенциалов для решения двойственной задачи по существу используется прямой симплекс метод (Dantzig G.B., 1947 г.) с учетом специфики транспортной задачи, а двойственные переменные (симплекс-множители) u_i, v_j являются потенциалами поставщиков и потребителей. Начальное допустимое базисное решение определяется методами «северо-западного угла», «минимального элемента», методом Фогеля и т.п. Для этого решения составляется система из $l + k - 1$ уравнений $u_i + v_j = c_{ij}$ и определяются начальные значения u_i, v_j и оценки для небазисных переменных $\tilde{c}_{ij} = u_i + v_j - c_{ij}$. Далее решение итеративно улучшается обычным путем: введением в базис новой переменной из числа небазисных, имеющих наибольшую положительную оценку $\tilde{c}_{ij}$ (при решении прямой задачи на минимум); выведением из базиса существующей переменной по условиям допустимости в симплекс-методе. Для этого в транспортной таблице для вводимой переменной строится замкнутый цикл из числа допустимых для вывода базисных переменных и выводится одна, имеющая наименьшее значение, которое и присваивается переменной, вводимой в базис. При этом пересчитываются значения всех базисных переменных так, чтобы не нарушались ограничения по a_i и b_j. В условиях допустимости в симплекс-методе знаменатель всегда равен единице, так как все коэффициенты в ограничениях на ресурсы для транспортной задачи равны нулю или единице. Итеративный процесс заканчивается, когда для всех небазисных переменных выполняются условия оптимальности симплекс-метода (для задачи на минимум, — все значения $\tilde{c}_{ij}$ должны быть неположительными, если оценки рассчитываются как $\tilde{c}_{ij} = u_i + v_j - c_{ij}$).

Временная сложность метода потенциалов зависит от эффективности реа-

лизации алгоритмов (матричных или сетевых) выполнения его основных шагов: нахождения начального допустимого базисного решения (начального опорного плана); определения переменной вводимой в базис и проверки на оптимальность текущего решения; определения переменной выводимой из базиса и пересчета новых значений базисных переменных. В алгоритмах могут использоваться различные способы представления данных — абстрактные типы данных (АТД): матричные, списочные, очереди с приоритетами, древовидные и др. Так, например, в сетевой реализации алгоритмов текущее базисное решение представляется остовным деревом, и используются эффективные структуры данных и процедуры для представления и работы с графами. Наиболее трудоемкой операцией в методе потенциалов является определение оценок $\tilde{c}_{ij}$, так как сначала надо выполнить $l+k-1$ операций для вычисления потенциалов базисных переменных, а затем рассчитать потенциалы и оценки $\tilde{c}_{ij}$ для $l \times k-(l+k-1)$ небазисных переменных. При самом простом подходе для этого понадобится просмотр всех элементов транспортной таблицы (одновременно можно найти максимальный $\tilde{c}_{ij}$ для переменной, вводимой в базис или определить оптимальность текущего решения). Трудоемкость шага выведения переменной из базиса зависит от способа представления текущего базисного решения (остовного дерева) в оперативной памяти компьютера и реализации процедур просмотра и пересчета элементов базиса. Эти процедуры могут быть очень эффективно реализованы при использовании современных АТД, поэтому временная сложность выполнения шага для $l+k$ циклически связанных значений невелика и нуль сравнима по сравнению с операциями расчета $\tilde{c}_{ij}$. Число итераций алгоритма для нахождения оптимального решения трудно выразить только через l и k, поскольку оно зависит от величины значений a_i и b_j, вырожденности базисных решений, применяемых структур данных задачи и процедур работы с ними. Однако ясно, что увеличить скорость работы прямого алгоритма симплекс метода можно за счет сокращения операций расчета $\tilde{c}_{ij}$ и проверки условия оптимальности текущего решения.

Широкое применение при решении транспортных задач получил венгерский метод [353]. В его основе лежит построение максимальных потоков через транспортную сеть с частично разрешёнными коммуникациями и последующее сокращение невязок. Венгерский метод решения использует схему прямо-двойственного алгоритма, в которой допустимое двойственное решение итеративно улучшается за счет решения задачи, двойственной к ограниченной прямой задаче.

Всегда можно получить допустимое решение двойственной задачи: $u_i = 0$, $v_j = \min c_{ij}$, $i = \overline{1,l}$, $j = \overline{1,k}$. Пусть пара индексов $ij \in \Omega$, если для них выполняется условие $u_i + v_j = c_{ij}$. Допустимые пары индексов $ij \in \{\Omega\}$ соответствуют допустимым столбцам в транспортной таблице для матричной реализации венгерского метода.

Запишем ограниченную прямую задачу, введя искусственные переменные $x_i \geq 0$, $i = \overline{1,l+k}$: найти

$$\min \sum_{i=1}^{l+k} x_i \quad \text{при} \quad \sum_j x_{ij} + x_i = a_i, \quad i = \overline{1,l}, \quad \sum_i x_{ij} + x_{l+j} = b_j, \quad j = \overline{1,k}, \quad \text{причем}$$

$x_{ij} \geq 0$, если $ij \in \{\Omega\}$ и $x_{ij} = 0$, если $ij \notin \{\Omega\}$. Эту задачу без переменных x_i можно записать как

$$\max \sum_{ij \in \Omega} x_{ij} \text{ при } \sum_j x_{ij} \leq a_i,\ i = \overline{1,l},\ \sum_i x_{ij} \leq b_j,\ j = \overline{1,k} \text{ при тех же ограничениях}$$

на x_{ij}.

Полученную задачу можно интерпретировать как задачу нахождения максимального потока в сети по допустимым парам $ij \in \{\Omega\}$ при оценке минимальной стоимости в ограниченной прямой задаче, равной

$$\Delta = \sum_{i=1}^{l} a_i + \sum_{j=1}^{k} b_j - 2\sum_{ij \in \Omega} x_{ij}$$

и использовать для ее решения эффективные сетевые алгоритмы.

Сетевую постановку транспортной задачи представляют в виде графа $G(s, t, L, K, P)$ с обобщенным узлом отправления s, обобщенным узлом потреб-

ления t, множеством L, $|\mathrm{L}|=l$ узлов отправления, множеством K, $|\mathrm{K}|=k$ узлов потребления и множеством ориентированных дуг P. Узел s связан с узлами отправления исходящими дугами p_{si} с пропускными способностями $w_{si}=a_i$, $i=\overline{1,l}$. Узлы потребления связаны с узлом t входящими дугами p_{jt} с пропускными способностями $w_{jt}=b_j$, $j=\overline{1,k}$. Узлы отправления и потребления соединены направленными дугами с неограниченными пропускными способностями $w_{ij}=\infty$ только для пар индексов узлов $ij\in\{\Omega\}$.

Очевидно, что сложность решения транспортной задачи прямо-двойственным алгоритмом зависит от числа итераций улучшения значений двойственных переменных при решении ограниченной прямой задачи, которую нужно решать до тех пор, пока величина невязки Δ не станет равной нулю и через сеть будет пропущен максимальный поток величины $\sum_{i=1}^{l}a_i=\sum_{j=1}^{k}b_j$. На каждой итерации необходимо находить максимальный поток, а число итераций опять таки трудно представить только через l и k.

Как известно, транспортная задача является частным случаем задачи о потоке минимальной стоимости, в которой все дуги допустимы и имеют стоимости и пропускные способности. Для любой конкретной задачи о потоке минимальной стоимости можно построить соответствующую ей транспортную задачу, решения которых будут совпадать. С этой точки зрения сложность решения транспортной задачи можно сопоставить со сложностью решения задачи о потоке минимальной стоимости.

Учитывая практическую важность транспортной задачи, методы и алгоритмы ее решения активно развивались в прошлом столетии и продолжают совершенствоваться в настоящее время. В работах [43, 46, 65, 100] были предложены эффективные методы и алгоритмы для решения классической транспортной задачи. В частности, в [46] рассматривается эффективный алгоритм, основанный на поразрядном сокращении невязок, с общей временной сложностью $O(kl^2m)$, где $m=\lceil\log_2\min(a_{\max},b_{\max})\rceil$, $a_{\max}=\max a_i$, $i=\overline{1,l}$, $b_{\max}=\max b_j$,

$j = \overline{1,k}$. Работы [65, 100] посвящены эффективным реализациям метода потенциалов для транспортной задачи в сетевой постановке, в которых экономно организованы процедуры вычисления потенциалов и пересчета базиса. Рекурсивный вариант венгерского метода, использующий технику цепных списков, предложен в работе [43]. Обзор с 1966 по 1978 гг., экспериментальное сравнение и практические советы по применению методов и алгоритмов решения транспортных задач можно найти в зарубежных работах [176, 201, 276, 277, 397].

В СССР большинство из указанных алгоритмов прошло экспериментальную апробацию на конкурсе «Транспорт-81», проведенным по инициативе и под руководством Л.В. Канторовича [9]. Как показали экспериментальные исследования, наилучшей оказалась программа, реализующая вариант метода потенциалов, предложенного в [65].

Несмотря на то, что транспортная задача отнесена к классу полиномиально разрешимых, при сравнении различных алгоритмов ее решения в практических случаях приходится полагаться на экспериментальные результаты. Известны патологические случаи, когда для решения транспортной задачи и задачи о потоке минимальной стоимости требуется экспоненциальное число итераций, казалось бы, эффективных алгоритмов [489, 490].

Дальнейшее развитие методов и алгоритмов решения транспортной задачи нашло отражение в работах [95, 142, 321, 344, 406]. В [321] на примере транспортной задачи сравнивается быстродействие теоретически полиномиального двойственного сетевого симплекс алгоритма с прямым сетевым алгоритмом. Приводятся результаты экспериментального сравнения двойственного алгоритма с известными реализациями прямого алгоритма в пакете программ Netflo. Показано, что на сетях до 300 узлов и 10000 дуг алгоритмы сравнимы по быстродействию. При превышении указанных границ прямой алгоритм быстрее двойственного на 30-60%. В работах [344, 406] приводятся эффективные реализации двойственного сетевого симплекс алгоритма и строго полиномиального алгоритма для решения транспортной задачи. Вопросы постоптимального анализа решения транспортной задачи для плавающих цен рассматриваются в ра-

боте [142]. Предлагается универсальный алгоритм «Push-and-Pull», который в отличие от прямого и двойственного симплекс метода, метода «падающих камней» (stepping-stone) не требует введения дополнительных и искусственных переменных для получения начального решения и других операций, связанных с циклическим обновлением базисных решений (не допускает вырожденных решений). Сравнивается вычислительная эффективность предложенного алгоритма с алгоритмами в пакете LINDO. Отмечается, что для проведения анализа на чувствительность и постоптимального анализа не могут быть в полной мере использованы существующие пакеты программ NETFLO, NETSOLVE, GENOS, CPLEX, QSB, LINDO.

В работе [95] рассмотрены способы реализации процедур ведущего преобразования в схеме прямого симплекс-алгоритма для задач транспортного типа, позволяющие осуществлять перестройку базисного дерева за время линейное от числа вершин сети, существенно сократив при этом число проверок условия оптимальности. Основанием для проведения работ явилось то, что многие полиномиальные алгоритмы не могут составить конкуренцию прямому симплекс-алгоритму при решении практических задач. Излагается способ учета дуг, заведомо удовлетворяющих условию оптимальности, основанный на упорядочении изучения базисного дерева. При использовании данного способа из поиска очередного кандидата для ввода в базис исключаются дуги, для которых установлено выполнение условия оптимальности. Предложенные в работе структуры данных и процедуры для них позволяют уменьшить вычислительную сложность всей итерации прямого симплекс-метода, а не только ведущего преобразования как в известных алгоритмах. Приводится техника программной реализации на языке C++ процедур для решения и постоптимизационного анализа транспортных задач соответственно в сетевой и матричной постановках. Отмечается, что предложенный способ дает возможность распараллеливания вычислений при решении многих задач об оптимальном потоке.

В [387] приводятся результаты экспериментальных исследований по решению транспортной задачи прямым симплекс алгоритмом при распараллелива-

нии вычислений на 14 процессорах. Даны расчеты для сетей, содержащих до 3000 узлов поставщиков и потребителей (т.е. $l = k = 3000$). Экспериментально показано, что затраты времени для задач с одинаковым числом поставщиков и потребителей размерностью $n \times n$ растут пропорционально n^a, где $2 \le a \le 2{,}2$. Кроме того в этой работе выполнен прекрасный исторический обзор статей, в которых предлагались новые АТД и процедуры для улучшения программной реализации прямого симплекс алгоритма для решения классической транспортной задачи.

Как уже указывалось, венгерский метод решения транспортной задачи связан с задачей нахождения максимального потока в сети, а сама транспортная задача является частным случаем задачи о потоке минимальной стоимости. Обширную библиографию по алгоритмам нахождения максимального потока и их временной сложности можно найти в [72, стр. 793-794]. Самая быстрая реализация алгоритма нахождения максимального потока разработана Гольдбергом и Рао [280], асимптотическая оценка их алгоритма составляет $O(\min(v^{2/3}, e^{1/2}) e \lg(v^2/e + 2) \lg C)$, где v и e число узлов и дуг сети, а C — максимальная пропускная способность дуги (для транспортной задачи $C = \max(a_i, b_j)$, $i = \overline{1,l}$, $j = \overline{1,k}$). Обзор развития методов и алгоритмов решения задачи о потоке минимальной стоимости и оценки их временной сложности приведены в работе [144, стр. 339-344, 395-397].

Кроме традиционных методов для решения транспортной задачи успешно применяются и другие [156, 426, 111].

В работе [156] сделана попытка приспособления двойственного метода аффинного преобразования (метода внутренней точки Кармаркара) для решения потоковых задач. Сравнивается вычислительная эффективность сетевого симплексного алгоритма с двойственным методом аффинного преобразования на примерах решения задач о потоке минимальной стоимости, транспортной задачи и задачи о назначениях. Показано, что сетевой симплекс алгоритм на сетях большой размерности работает значительно быстрее. Для распараллелива-

ния на кластере лучше второй алгоритм.

В [426] также реализован метод двойственного аффинного преобразования и проводится его сравнение с градиентным методом, сетевым симплекс методом (пакет NETFLO), методами декомпозиции (пакет RELAX). Для задачи о назначениях размерностью 500x500 (37501 дуг) при распараллеливании ее решения на 8 процессорах время работы алгоритма внутренней точки несколько больше чем у градиентного алгоритма. По времени работы на сетях размерностью до 72000 узлов сетевой симплекс алгоритм и декомпозиционные алгоритмы оказались соизмеримыми, а алгоритм внутренней точки в большинстве случаев отставал от лучших результатов.

В недавно вышедшей книге [111], посвященной задачам дробного программирования и недифференцируемой оптимизации, также рассматриваются различные подходы к решению транспортной задачи, основанные на схемах декомпозиции по ограничениям и переменным с применением субградиентных методов. Отмечается, что их применение целесообразно, когда число потребителей намного больше числа поставщиков, а также, если в транспортную задачу включены дополнительные сложные ограничения.

В настоящее время для решения транспортных задач успешно применяются метаэвристические методы и алгоритмы, — генетические и эволюционные [53, 384, 385], имитации отжига [149], роя частиц [319, 85]. Так, например, в работе [53] приводится генетический алгоритм, позволяющий параллельно генерировать множество субоптимальных решений для линейной транспортной задачи. Дана реализация алгоритма на языке C++. Отмечается, что алгоритм может быть легко приспособлен для решения нелинейных задач. В [85] предложен многороевый алгоритм для приближенного решения транспортных задач с нелинейными функциями стоимости произвольного вида. Приведены результаты вычислительных экспериментов по исследованию эффективности решения нелинейных транспортных задач различной сложности в параллельной реализации разработанных алгоритмов.

В заключение обзора приведем несколько последних работ, представляю-

щих интерес для выбора методов решения транспортной задачи [75, 374, 375, 294, 61]. В [75] предлагается метод решения задач транспортного типа, в котором последовательно решаются двумерные задачи с одной связывающей переменной. Для двумерных задач задаются коэффициенты целевой функции для связывающей переменной и формулируются одномерные задачи. Их решения могут дать исходный оптимум или определить систему ограничений на переменные. Допустимые решения построенной системы ограничений дают оптимальное решение исходной задачи. Если систем ограничений не имеет допустимых решений, решается задача о максимальном потоке, находится множество взаимно удовлетворённых пар, формируется множество обобщённых производителей и потребителей и путём суммирования строится новая задача с меньшим числом ограничений. После этого процесс последовательного решения двумерных задач повторяется. Алгоритм строит последовательность псевдорешений с монотонным возрастанием функционала. Отмечается, что метод может быть распространен на широкий класс транспортных и распределительных задач.

В работах [374, 375] соответственно изучаются вопросы, связанные с вырождением базисных решений при построении начального опорного плана методом наименьшего элемента и влиянием выбора метода получения начального опорного плана на число итераций прямого симплекс алгоритма для транспортной задачи (его еще называют MODI или stepping stone методом [276]) для получения оптимального решения. Сравниваются методы северо-западного угла, выбора минимального элемента и метод Фогеля. Как и ожидалось, лучшими оказались метод наименьшего элемента и метод Фогеля. На больших размерностях метод северо-западного угла намного хуже (приблизительно в 3 раза).

В [294] рассматривается Shortlist Method, который является улучшением прямого симплекс метода и позволяет существенно сократить число просмотров небазисных переменных, включаемых в базис и уменьшить объем памяти компьютера для хранения данных. Экспериментальные исследования показали, что предложенный алгоритм позволяет значительно уменьшить время решения

транспортной задачи по сравнению с последней версией (state-of-the-art) симплекс алгоритма, реализованного в пакетах CRAN-Package emdist и LP_Solve. Алгоритм позволяет решать транспортные задачи большой размерности за разумное время (задача с 1000 поставщиками и потребителями решена за 1,2314 с. на компьютере Intel Core i7 CPU, 3.20 GHz). Кроме того в работе приводится новый метод построения начального допустимого решения и экспериментально показано насколько сокращается время решения задачи при его использовании по сравнению с известными методами северо-западного угла, минимального элемента, Фогеля и др.

В работе [61] обсуждается транспортная задача с нечеткими ограничениями на ресурсы, когда значения a_i, $i=\overline{1,l}$ и b_j, $j=\overline{1,k}$ задаются наборами нечетких треугольных чисел. Возвращаясь к решению задачи развозки порожних контейнеров (В.6), (В.3)-(В.5), отметим, что в реальных сетях величина грузопотоков контейнеров всегда изменяется во времени. Как правило, внешние входящие потоки мелкопартионных грузов не подчиняются какому-либо известному вероятностному закону распределения (биномиальному, пуассоновскому и пр.). Поэтому для определения, например, среднесуточных потоков между узлами для различных периодов на протяжении года и в зависимости от сезонных колебаний потоков, в ИАС используются современные методы математического моделирования на основе временных рядов данных [50]. Прежде всего, к ним следует отнести метод группового учета аргументов (МГУА), методы авторегрессии, нечеткие нейросети, байесовские модели и сети, которые позволяют получить высококачественные прогнозы для нелинейных и нестационарных процессов. Большинство из перечисленных моделей и методов реализованы в многочисленных пакетах программ обработки статистических и экспериментальных данных и поддержки принятия решений, таких как NeuroShell2, ModelQest, ASPN, KnowledgeMiner, MatLab и др. Соответственно будут изменяться и грузопотоки в контейнерах, которые заданы матрицей U. Нестационарность входящих потоков и других процессов, происходящих в сети, а также возможность с помощью ИАС оперативно управлять перераспределением по-

токов порожних контейнеров в реальном масштабе времени и послужили отправной точкой для выбора детерминированной постановки задачи (В.6), (В.3)-(В.5). При наличии ИАС всегда известны реальные мелкопартионные потоки грузов и величина грузопотоков в контейнерах. Поэтому всегда можно оперативно решить задачу развозки порожних контейнеров точными методами. При этом небольшие колебания величины потоков будут незначительно отражаться на оптимальном распределении потоков порожних контейнеров и «гаситься» резервом грузоподъемности транспортных средств. Однако нужно оперативно следить, чтобы грузопоток груженых и порожних контейнеров не нарушал схемы распределения и маршрутизации потоков, полученной для периодов текущего планирования. В то же время рассмотренная в [61] транспортная задача с нечеткими ограничениями на ресурсы может представлять интерес для случая, когда по каким-либо причинам ИАС временно неработоспособна.

В.3. Вычислительный эксперимент

Решение задачи развозки порожних контейнеров выполнялось на однородных транспортных сетях с числом узлов n = 100, 250, 500, 1000, 2000, 3000, 4000 и степенью узлов val = 5. Длины дуг сети и элементы матрицы потоков груженых контейнеров $U = \| u_{ij} \|_{n \times n}$ задавались датчиком равномерно распределенных псевдослучайных целых чисел в пределах от 80 до 300 км и от 1 до 20 контейнеров. Принимались следующие значения параметров: грузоподъемность транспортных средств W = 40 контейнеров; периодичность движения транспортных средств $t_{пер}$ = 24 ч; время стоянки транспортных средств в конечных пунктах следования $t_{ст}$ = 22 ч; средняя скорость движения транспортных средств $V_{ср}$ = 70 км/ч. Для расчета среднегодовых приведенных затрат на транспортировку и обработку порожних контейнеров использовались следующие тождественные (В.10) и (В.11) формулы в условных единицах стоимости:

$$Z^{c}_{mp} = \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{abs(u_{ij} - u_{ji})}{W} \left[\frac{13939{,}2(t_{ст} + 2d^{*}_{ij} / V_{ср})}{t_{пер}} + 7200 + 116{,}8 d^{*}_{ij} \right],$$

$$Z^{c}_{погр} = \sqrt{25595\left(\sum_{i=1}^{n}\sum_{j=1}^{n} abs(u_{ij} - u_{ji})\right)^2 + 2629\left(\sum_{i=1}^{n}\sum_{j=1}^{n} abs(u_{ij} - u_{ji})\right)},$$

$$Z^{o}_{mp} = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{k}\frac{x_{ij}}{W}\left[\frac{13939{,}2(t_{cm} + 2\tilde{d}^{*}_{ij}/V_{cp})}{t_{пер}} + 7200 + 116{,}8\tilde{d}^{*}_{ij}\right],$$

$$Z^{o}_{погр} = \sqrt{25595\sum_{i=1}^{l}\sum_{j=1}^{k}(2x_{ij})^2 + 2629\sum_{i=1}^{l}\sum_{j=1}^{k}2x_{ij}}\ .$$

Для нахождения кратчайших путей использовался алгоритм Флойда, а для решения транспортной задачи — венгерский алгоритм в матричной реализации.

Из рис. В.4 видно, что при $n \geq 1000$, $l \geq 503$, $k \geq 496$ начинает значительно возрастать время решения обеих задач, особенно у венгерского алгоритма. Сравнение методов балансировки удобнее проводить не по абсолютным значениям затрат Z^{c}_{mp}, Z^{o}_{mp}, $Z^{c}_{погр}$, $Z^{o}_{погр}$, а по снижению суммарных затрат $Z = (Z^{c}_{mp} + Z^{c}_{погр})/(Z^{o}_{mp} + Z^{o}_{погр})$ и снижению количества порожних контейнеров $U = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} abs(u_{ij} - u_{ji})\right)/\left(2\sum_{i=1}^{l}\sum_{j=1}^{k} x_{ij}\right)$. На рис. В.5 приведены соответствующие гистограммы.

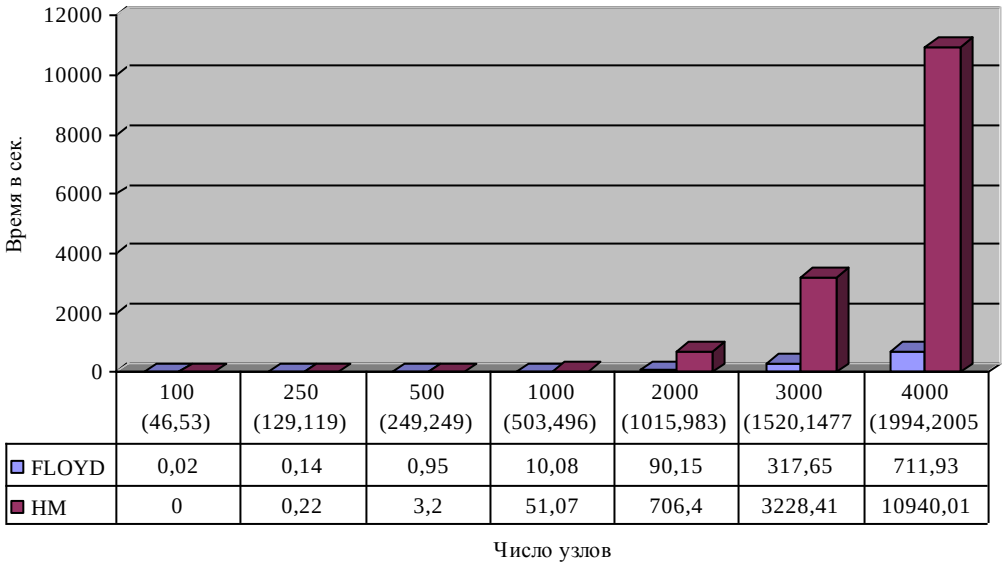
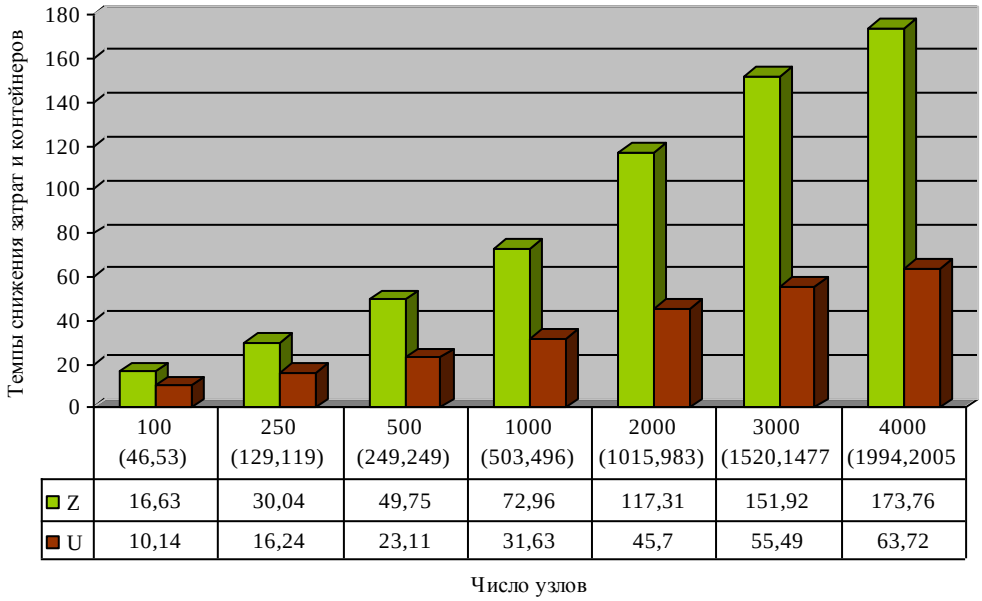


Рис. В.4. Время построения кратчайших путей алгоритмом Флойда (FLOYD) и время решения транспортной задачи венгерским алгоритмом (HM) (в скобках указано соответственно число поставщиков и потребителей)



	100 (46,53)	250 (129,119)	500 (249,249)	1000 (503,496)	2000 (1015,983)	3000 (1520,1477	4000 (1994,2005
Z	16,63	30,04	49,75	72,96	117,31	151,92	173,76
U	10,14	16,24	23,11	31,63	45,7	55,49	63,72

Рис. В.5. Снижение суммарных затрат (Z) и количества порожних контейнеров (U) при применении оптимальной балансировки по сравнению с симметричной (в число раз)

Программы составлены на языке Digital Visual Fortran и выполнялись под управлением операционной системы Windows Vista на ПК с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.

Выводы

Рассмотрены два способа балансировки матрицы контейнерных потоков в транспортной сети — симметричный, часто применяемый на практике, и оптимальный, основанный на решении транспортной задачи. Предложена математическая модель и алгоритм решения задачи развозки порожних контейнеров. Показано, что для решения транспортной задачи при числе узлов в сети, превышающем 1000, необходимо использовать более эффективные алгоритмы, основанные на матричных и сетевых реализациях прямого симплекс алгоритма [65, 95, 75, 294], технике распараллеливания вычислений на многоядерных процессорах и вычислительных кластерах [387], а также приближенные генетические, эволюционные и другие мета-алгоритмы [384, 149, 319, 85]. Экспериментально показано, что оптимальная балансировка по сравнению с симметри-

чной балансировкой позволяет значительно сократить суммарные затраты на транспортировку и обработку порожних контейнеров (на сетях от 100 до 4000 узлов в 17 и 174 раза соответственно).

Приложение Г. Обзор методов построения кратчайших путей

За последние десятилетия было разработано достаточно много эффективных алгоритмов построения КП по одному критерию. Все множество известных алгоритмов построения КП можно условно разбить на три группы: сетевые комбинаторные алгоритмы, основанные на процедурах расстановки и изменения меток в процессе построения КП и использующие эффективные структуры абстрактных типов данных (АТД); алгебраичные или матричные алгоритмы, оперирующие с матричными представлениями данных о структуре и характеристиках графа или сети; алгоритмы, базирующиеся на методах решения задач линейного программирования — различных модификациях симплекс метода.

Для удобства изложения примем единую терминологию — будем говорить о неориентированных сетях с узлами и дугами, всегда подразумевая, что все сказанное в большей или меньшей мере относится и к неориентированным и ориентированным графам.

Первая и третья группы алгоритмов в основном используются для нахождения деревьев КП между заданными подмножествами узлов сети (от одного ко всем и наоборот, между двумя заданными узлами и т.п.), вторая — для построения КП между всеми узлами сети. В зарубежной литературе часто употребляются термины Single Source (or Sink) Shortest Path (SSSP) — алгоритмы для нахождения деревьев кратчайших путей от заданных источников, Shortest Path Tree (SPT) и All Pairs Shortest Paths (APSP) — алгоритмы для нахождения КП между всеми парами узлов. Для краткости, когда это будет удобно, будем пользоваться этой терминологией.

Первые систематизированные библиографические обзоры по алгоритмам построения КП опубликованы в 1969 г. Дрейфусом [243], за период с 1956 по 1974 гг. в 1975 г. Пирсом [420], а также в 1984 г. Део и Пангом [235]. Последу-

ющие обзоры по КП можно найти в книгах [72, 144, 381].

Рассмотрим более подробно развитие комбинаторных методов построения КП. Некоторые ссылки в обзоре взяты из [322]. Основная идея всех комбинаторных методов заключается в создании корневого узла с очередью связанных с ним других узлов с метками расстояний до них, которые на каждой итерации алгоритма обновляются в очереди при наращивании кратчайшего дерева. Процедура наращивания дерева заканчивается, когда очередь узлов оказывается пустой. При этом главной задачей алгоритма является выбор из очереди наиболее подходящих узлов — кандидатов для дальнейшего наращивания кратчайшего дерева за минимальное время. Обычно различают алгоритмы расстановки меток — Label-Setting и алгоритмы изменения (обновления, корректировки) меток — Label-Correcting. Все эти алгоритмы итерационные. Алгоритмы отличаются друг от друга способами обновления меток на последовательных итерациях и выбором направлений для поиска КП. Алгоритмы расстановки меток на каждом шаге назначают узлам постоянные кратчайшие метки, которые на последующих итерациях не изменяются. Алгоритмы изменения меток итеративно обновляют значения меток узлов вплоть до последней итерации, на которой определяются кратчайшие метки. Первые, как правило, применяются для нахождения КП в ациклических сетях с произвольными и неотрицательными длинами дуг, а вторые — для более широкого класса задач, а также в случаях отрицательных весов дуг. Два комбинаторных подхода отличаются алгоритмическими стратегиями и используемыми структурами данных.

Метод расстановки пометок узлов с кратчайшими расстояниями до них впервые был предложен Дейкстрой и Данцигом [240, 221]. Сложность алгоритма Дейкстры складывается из двух основных операций: нахождения узла с наименьшей величиной расстояния и совершения релаксации, т.е. изменения значений меток связанных узлов. При простейшей реализации эти операции потребуют соответственно $O(n)$ и $O(1)$ времени. Учитывая, что первая операция выполняется $O(n)$ раз, а вторая $O(m)$, где n и m число узлов и дуг сети, получим асимптотическую оценку трудоемкости алгоритма $O(n^2 + m)$. Такая асим-

птотика является оптимальной для плотных сетей, когда $m \approx n^2$. Для разреженных сетей ее можно улучшить за счет уменьшения времени выполнения операций первого типа. Ясно, что эффективность алгоритма Дейкстры зависит от структур данных, используемых для реализации очереди и представления входной сети. Как было показано, она равна $O(n^2)$ для сетей, представленных весовыми матрицами и неупорядоченной очередью в виде массива. Для сетей, представленных связанными списками смежности и очередью с приоритетами, реализованной как неубывающая пирамида, эффективность равна $O(m \log n)$. Лучшую верхнюю границу можно получить для алгоритма Дейкстры, если реализовать более сложные структуры данных. Развитие методов с использованием очередей с приоритетами имеет длинную историю, включая исследования бинарных пирамид (binary heaps) — Джонсон [326], пирамид Фибоначчи (Fibonacci heaps) — Фридман и Тарьян [256], структур «bucket heaps» [237] и «radix heaps» [145]. Для целых ключей технология «radix heaps» позволила Торупу получить оценку времени $O(m + n \log\log n)$ [457, 458] и линейное время решения задачи для одного специального случая [456]. Последний алгоритм сканирует узлы один за другим, однако, в другом порядке, чем алгоритм Дейкстры. На основе алгоритма Торупа, Петти и Рамашандран [418], используя технику представления структур данных для построения минимального остовного дерева, разработали новый алгоритм для построения КП от одного источника с неотрицательными длинами дуг. Их алгоритм имеет лучшую оценку, чем $O(m + n \log\log n)$, когда веса дуг мало отличаются друг от друга. Как показано в работе [159], этот алгоритм, несмотря на низкую оценку сложности, для практических целей является малопригодным. Для сетей с отрицательными весами дуг Габовым, Тарьяном и Гольдбергом соответственно предложены алгоритмы с трудоемкостью $O(\sqrt{nm \lg(nw)})$ [259] и $O(\sqrt{nm \lg(w)})$ [281], где w — максимальный вес дуги. Мейер [382] получил первый алгоритм построения КП за линейное время в среднем. Алгоритм ALD был найден Голдбергом [282]. Для графов с ограниченным размером был получен более простой Δ-ступенчатый

метод [383]. Он использует «bucket heaps» и дает хороший параллельный алгоритм для ограниченных графов с малым диаметром. В нем суммирование стоимостей дуг допускается и в случае их отрицательных значений. Если все стоимости дуг являются целыми числами больше $(-n)$, то алгоритм достигает оценки времени $O(m\sqrt{n\log n})$ [283].

Другие подходы к решению задачи построения КП, основанные на корректировке меток (Label-Correcting), были предложены в работах Форда, Мура, Беллмана, Форда и Фалкерсона [253, 394, 180, 254]. В отличие от алгоритма Дейкстры, алгоритм Беллмана-Форда решает задачу SSSP и в случае, когда веса дуг могут быть отрицательными и находит циклы с отрицательным весом, достижимые из истока. Все они используют процедуры включения и исключения из очереди, описанные соответственно в [77, 180, 183, 284, 288, 408, 412]. В работе [483] Йен приводит процедуру просмотра узлов, сокращающую примерно в два раза операции обновления меток узлов по сравнению с алгоритмом Беллмана.

Алгоритмы, приведенные в работах [412, 77, 408, 284] основаны на следующем правиле. Предположим, что узел i в очереди был просмотрен на предыдущей итерации алгоритма и все его кратчайшие предшественники $i_1,\ldots,i_k$ также находятся в очереди. Тогда для дальнейшего наращивания дерева КП лучше выбрать узел i, чем узлы из последовательности $i_1,\ldots,i_k$. Если узел i_l будет выбран перед узлом i, то он после просмотра узла i и изменения меток, должен быть опять просмотрен с соответствующим изменением меток. Таким образом, выбор узла i, а не его предшественников, дает возможность сохранить метки обновления узлов предшественников. Понятно, что такое правило не гарантирует полиномиальное время выполнения операций. Например, операции исключения из очереди, реализованные в [412, 77] очень эффективны на практике, но имеют псевдополиномиальную оценку времени выполнения. Другие алгоритмы, такие как [408, 284], гарантируют полиномиальное время выполнения операций, однако на практике ведут себя хуже.

В алгоритме, предложенном Гловером в [278], комбинируется техника расстановки и обновления меток. Алгоритм поддерживает две очереди: *NEXT* и *NOW*. Узлы вводятся в *NOW* из *NEXT*, если их метки расстояний меньше некоторого порогового значения, далее вновь просматриваются узлы из *NOW* и включаются новые узлы в *NEXT*. После того, когда очередь *NOW* станет пустой, устанавливается новый порог значений и операции обновления продолжаются вплоть до опустошения очереди *NEXT*. Эффективность этого алгоритма сильно зависит от способов выбора пороговых значений и поэтому в каждом конкретном случае требуется тщательная настройка алгоритма.

Несмотря на то, что алгоритмы расстановки меток имеют лучшие теоретические оценки трудоемкости, чем алгоритмы обновления меток, их накладные расходы на сортировку узлов-кандидатов могут значительно ухудшить быстродействие на практике и, особенно для разреженных сетей.

Одними из наилучших известных результатов, полученных в настоящее время для решения задачи SSSP комбинаторными методами, являются $O(\min\{m + n\log n, m\log\log C, m + n\sqrt{\log C)}\})$ для алгоритмов [256], [326] и [145] для случая неотрицательных длин дуг, а также $O(mn)$ для большинства алгоритмов обновления меток в случае сетей с отрицательными дугами не содержащих циклов отрицательной длины [144].

Обширные экспериментальные исследования по сравнению эффективности различных алгоритмов построения КП между заданными подмножествами узлов сети комбинаторными методами были проведены много раз различными авторами (см., например, работы [208, 241, 278, 279, 320, 380, 391, 408, 491]. В работе Черкасского, Гольдберга и Радзика [208] указывается, что не существует наилучшего единого алгоритма для решения всего класса задач нахождения КП. Авторы работы исследовали робастность алгоритма Дейкстры, реализованного с применением технологии «double bucket», названного DIKB и метода обновления меток, использующего идею топологической сортировки, названного GOR1, при многочисленных тестах на сетях различной конфигурации и размерности. Знан и Нун [491], пользуясь этими же программами, исследовали

поведение алгоритмов DIKB и GOR1 на реальных дорожных сетях. Эксперименты показали, что на сетях с небольшими длинами дуг лучше работает алгоритм Дейкстры с технологией «bucket», а для сетей с более длинными дугами алгоритм DIKB, использующий «double bucket».

Рассмотрим кратко матричные APSP — алгоритмы Флойда-Уоршалла [252, 473] и Карре [197, 198], которые чаще всего используются для построения КП между всеми узлами сети.

Известно [198], что для решения задачи нахождения всех КП, сформулированной в виде алгебраичного (матричного) уравнения Беллмана, может быть использована техника, аналогичная схемам прямого или итеративного метода решения системы линейных уравнений. В частности, использование схем методов Жордана-Гаусса и исключения Гаусса приводит к алгоритму Флойда-Уоршалла и алгоритму Карре соответственно.

Итеративные методы Якоби и Гаусса-Зейделя [198] дают схему алгоритмов Беллмана [180] и Форда [254]. Метод релаксации Бертсекаса [184] также может быть интерпретирован схемой вычислений Гаусса-Зейделя [409].

Если матричное уравнение Беллмана решать с помощью вычисления обратной матрицы, то схема Морриса [395] для обращения матрицы приводит к индуктивному алгоритму Данцига [224]. Декомпозиционные алгоритмы предложены в работах Миллса [388] и Ху [317]. В этих алгоритмах сеть разбивается на части, для каждой части находится дерево КП, а затем все деревья объединяются.

Все перечисленные алгоритмы APSP (исключая алгоритмы Беллмана и Форда для нахождения КП между подмножествами узлов) имеют оценку сложности порядка $O(n^3)$ и весьма эффективны для плотных сетей. Для разреженных сетей известен алгоритм Джонсона [328], который использует списки смежных узлов и находит все пути за время $O(n^2 \lg n + nm)$. Он также работает с отрицательными весами дуг и определяет наличие циклов с отрицательной стоимостью.

Рассмотрим более подробно алгоритм Карре, поскольку он имеет некото-

рые преимущества по сравнению с другими матричными алгоритмами. Алгоритм включает *LU*-процедуру декомпозиции, n последовательных операций «вперед» и «назад» и находит все кратчайшие деревья от всех узлов в сети к одному i, $i=\overline{1,n}$. Такой подход аналогичен схеме решения системы линейных уравнений методом исключения Гаусса и столбец i в матрице расстояний будет соответствовать кратчайшему дереву к узлу i. В [197] Карре дал графическую интерпретацию работы своего алгоритма и алгебраически доказал его корректность. Преимущество алгоритма заключается в том, что, во-первых, для полных сетей он выполняет меньшее число сравнений из всех возможных — $n(n-1)(n-2)$ [401], в то время как алгоритм Флойда-Уоршалла, выбирая узлы в другом порядке, выполняет все операции сравнения. Во-вторых, поиск всех КП распадается на поиск n кратчайших деревьев к каждому узлу, что дает возможность избежать многих излишних операций при построении КП от заданного числа узлов.

Из линейной алгебры известно, что проблема решения системы линейных уравнений связана с размерностью решаемых задач и точностью получаемых решений. Итеративные методы не гарантируют получение точного решения, но сходятся гораздо быстрее прямых методов. Поскольку при решении матричного уравнения Беллмана не возникает аналогичных проблем, на практике наиболее популярными остаются итеративные алгоритмы построения КП (например алгоритмы расстановки и обновления меток), чем матричные (алгоритмы Флойда-Уоршалла и Карре).

Известны также алгоритмы, основанные на операциях перемножения матриц [250, 318, 357, 433, 479], но все они хуже, чем алгоритм Флойда-Уоршалла. В работе Ахо и др. [143] показано, что использование технологии перемножения матриц для нахождения всех КП, так же сложно, как и нахождение КП от одного источника и требует $O(n^3)$ времени. Фридман [255] предложил $O(n^{2,5})$ алгоритм для построения КП от одного источника, требующий экспоненциальной памяти. Улучшенный алгоритм Фридмана со сложностью

$O(n^3((\log\log n)/\log n)^{1/2})$ предложен в работе Такаока [444].

Много алгоритмов было разработано с использованием блочной декомпозиции и техники быстрого перемножения матриц. Все эти алгоритмы имеют улучшенные оценки сложности, однако требуют либо целочисленности, либо небольших абсолютных значений длин дуг сети [148, 445, 493] или могут быть применены только для не взвешенных или неориентированных сетей [263, 264, 432]. Все алгоритмы, основанные на технике перемножении матриц, наиболее применимы к плотным сетям и не используют возможностей улучшения быстродействия за счет учета степени разреженности.

До сих пор в литературе не встречается работ, посвященных тщательному анализу экспериментальных исследований эффективности алгоритмов построения всех КП в сети. Принято считать, что матричные алгоритмы требуют большей памяти, применяются к плотным сетям и не подходят для решения задач построения КП для разреженных сетей большой размерности.

Алгоритм Флойда-Уоршалла является лучшим матричным алгоритмом, однако его применение для сильно разреженных сетей мало оправдано. С другой стороны, матричный алгоритм Карре может быть с успехом применен и для разреженных сетей и его дальнейшее развитие могло бы составить конкуренцию лучшим комбинаторным алгоритмам [165]. Возможность развития алгоритма Карре показана в работах [293] и [322] и связана с тем, что для сетей с фиксированной топологией, КП между всеми узлами сети могут многократно пересчитываться при изменении длин дуг. Поэтому, если соответствующим образом использовать неизменность структуры сети, можно в алгоритме Карре уменьшить трудоемкость операции *LU*-декомпозиции, и на стадии предварительной обработки узлов создать схему построения КП, не требующую полного перебора узлов. Экспериментальные исследования в [165, 322] показали, что реализованный на основе такого подхода алгоритм построения всех КП оказался в несколько раз лучше некоторых комбинаторных алгоритмов, используемых для разреженных сетей.

Рассмотрим алгоритмы построения кратчайших путей, основанные на ме-

тодах решения задач линейного программирования. Чаще всего они применяются для нахождения деревьев кратчайших путей. Если проанализировать работу алгоритма Дейкстры, то можно увидеть сходство алгоритмов расстановки и обновления меток с прямым сетевым симплекс-методом [238]. Такой симплексный алгоритм с трудоемкостью $O(n^3)$ был предложен в работе Гольдфарба [289]. Другой алгоритм для нахождения КП от одного источника, сравнимый с лучшими алгоритмами корректировки меток и сложностью $O(mn)$, можно найти в [290]. Был предложен также ряд двойственных симплекс алгоритмов [184, 185, 200, 409]. Алгоритм Дейкстры может быть представлен и как прямо-двойственный алгоритм [410]. Практические исследования [196, 238, 361] показали, что для решения реальных задач комбинаторные алгоритмы все-таки оказываются лучше, чем алгоритмы, основанные на методах линейного программирования, включая прямой, двойственный и прямо-двойственный симплекс методы.

Среди отечественных разработок 1970-1980-ых годов следует отметить алгоритм двусторонней очереди Левита [77], работы Филлера [135], Диница и Карзанова [47, 48], посвященные разработке целого ряда новых эффективных реализаций алгоритма Дейкстры, в основе которых заложена техника использования справочных для представления структур данных [64] (по этому вопросу см. также обзоры в [10, 77]). Известна также работа Неверова [93], в которой предложен новый метод построения КП со сложностью в два раза меньше, чем у алгоритма Дейкстры.

Для сравнения вычислительной эффективности различных алгоритмов построения КП в период с 1983 по 1984 гг. был проведен всесоюзный конкурс «Транспорт-83» [62], на котором были представлены программы, реализующие в основном все известные лучшие алгоритмы, в том числе и программа автора диссертации. Результаты конкурса показали, что наилучшим, в смысле скорости получения решения, оказался метод двусторонней очереди [77]. Весьма эффективной оказалась также модификация алгоритма Флойда, составившая конкуренцию лучшей реализации алгоритма Дейкстры [47, 48]. Отмечается, что с

увеличением размеров сетей разница во времени счета между алгоритмами двусторонней очереди и Дейкстры сокращается, поэтому для сетей, содержащих более 4-5 тысяч узлов, делать выводы о преимуществах первого алгоритма преждевременно. Неплохо зарекомендовал себя также метод Неверова, однако, с ростом размеров сетей этот метод начинает существенно отставать от лучших. Для нахождения КП от одного источника к одному стоку, наилучшим оказался алгоритм Дейкстры.

За последние годы эволюция методов решения задачи нахождения КП была связана с разработкой и дальнейшим усовершенствованием специальных структур данных (АТД) для представления объектов задачи, позволяющих максимально сократить время построения КП и созданием параллельных алгоритмов для многопроцессорного решения задачи. Так, например, под руководством Б. Черкасского (B. Cherkassky), А. Фиата (A. Fiat), К. Харрелсона (C. Harrelson), Х. Каплана (H. Kaplan), Т. Паера (T. Pajor), И. Резенштейна (I. Razenshteyn) группой разработчиков И. Абрахамом (I. Abraham), Д. Деллингом (D. Delling), А. Гольдбергом (A. Goldberg) и Р. Верником (R. Werneck) интенсивно разрабатывался проект, посвященный теоретическому и экспериментальному изучению алгоритмов КП и связанных с ними задач http://research.microsoft.com/en-us/projects/SPA/. Было показано, что для решения отдельных SSSP задач существуют «почти оптимальные» алгоритмы в теории и на практике, в то же время для решения более широкого класса задач, включая и APSP-проблему, имеются предпосылки для улучшения уже существующих алгоритмов. Основные усилия проекта были направлены на решение задачи нахождения КП между заданной парой вершин при ограничениях на доступный объем оперативной памяти для проведения процедуры препроцессинга очень большого графа. Такая задача характерна для вычисления расстояний на интернет картах, таких как Google Maps, Yahoo! Maps, Bing Maps. Были также разработаны методы ускорения классических алгоритмов построения КП, включая методы *landmark-based A* search* и *reach-based pruning* и их комбинации. Полученные алгоритмы оказались очень практичными и могут успешно

использоваться на серверах, настольных и портативных компьютерах, включая планшеты, смартфоны, автомобильные системы навигации. В рамках проекта также были рассмотрены задачи более гибкого планирования и выбора разумных альтернативных маршрутов передвижения в масштабе реального времени, в случаях затора на дорогах, а также выбора индивидуальных маршрутов по заданным критериям и предпочтениям. Другие задачи проекта были связаны с нахождением более дешевых путей передвижения, минимальных циклов, аппаратным ускорением построения дерева кратчайших путей (PHAST) с использованием многоядерных процессоров и новейших графических адаптеров. Всю информацию о результатах выполнения проекта можно найти в Интернете, а также в многочисленных работах руководителей и исполнителей проекта (см., например, [137-141, 209, 228-234, 285, 377]). По вопросам параллельных вычислений см. также [40, 81, 98, 354].

Наряду с задачами построения КП по одному критерию, значительный интерес представляют многокритериальные задачи о путях (multicriteria (multiobjective) shortest path problems, MSPP). Такие задачи естественным образом возникают во многих приложениях, когда имеется несколько различных параметров, характеризующих узлы и дуги сети и интенсивно исследуются при проектировании транспортных, информационных и телекоммуникационных сетей. В большинстве случаев выбор «оптимальных» кратчайших путей по экстремальным значениям этих параметров представляет сложную задачу, так как параметры могут быть противоречивыми и конкурировать между собой. Основываясь на интуиции, лицу, принимающему решение (ЛПП), трудно отдать предпочтение какому-либо конкретному варианту КП по нескольким критериям. Поэтому возникает необходимость в применении некоторой методологии выбора наиболее приемлемых вариантов, базирующейся на строгих математических методах и здравой интуиции ЛПР.

Для решения многокритериальных задач КП как начальное ядро используются различные модификации уже рассмотренных алгоритмов построения КП по одному критерию (Label-Setting, Label-Correcting комбинаторные, мат-

ричные и LP-алгоритмы, основанные на современных АТД, таких как пирамиды Фибоначчи, d-арные кучи и т.п.). Сначала находится множество допустимых вариантов решения по критериям Парето, Слейтера, вероятностным, лексикографическим и др., и далее с использованием множества определенных методик отбираются наиболее приемлемые.

Известно, что в общем случае проблема MSPP является NP-трудной [267]. Одной из первых работ, посвященной систематическому исследованию многокритериальных задач о путях является обзорная статья Хансена (1979г.) [303], в которой определяется класс задач с экспоненциальными оценками сложности в худшем случае и приведена полностью полиномиальная схема решения (fully polynomial time approximation schemes, FPTAS) для двухкритериальных задач. Для многокритериальных задач на ориентированных ациклических сетях также известна полиномиальная схема решения [472]. Хороший обзор по решению двухкритериальных задач приведен в [247]. Для случаев, когда число критериев оптимизации превышает два, пока не достигнуто значительных результатов. В работах [247, 463, 466] отмечается, что использование на практике ε- приближенных алгоритмов для решения задач с двумя, тремя критериями показало перспективные пути для нахождения более эффективных алгоритмов.

Одним из наиболее популярных методов решения MSPP задач является метод построения аппроксимационных поверхностей Парето (оптимальных поверхностей аппроксимации по Парето) [411]. Неформально $(1+\varepsilon)$- Парето поверхность P_ε представляет собой подмножество возможных решений, такое, что для любого другого Парето - оптимального решения в P_ε существует решение не худшее, чем отличающееся на величину $(1+\varepsilon)$ по всем критериям. По-существу, каждой точке этой поверхности соответствует потенциально достижимое значение одного из критериев при фиксированных, соответствующих этой точке, значениях остальных критериев. Парето - оптимальная поверхность может быть представлена многомерными диаграммами связи между критериями, показывающими, как потенциально достижимое значение одного критерия

зависит от потенциально достижимых значений других критериев. В этом случае дальнейшее улучшение значения каждого из критериев оптимизации может быть достигнуто только за счет ухудшения значений остальных критериев.

В [411] показано, что для любой многокритериальной задачи существует $(1+\varepsilon)$- Парето поверхность P_ε полиномиального размера $\left|P_\varepsilon\right| = O((4B/\varepsilon)^{N-1})$, где B — число бит, необходимых для представления полиномиальной от входного размера критериальной функции, которая может быть построена за $O((4B/\varepsilon)^d)$ вызовов программы «gap», решающей за полиномиальное время относительно входного размера и величины $1/\varepsilon$ следующую задачу. Для заданного вектора значений a найти решение доминирующее над a или выдать ответ, что не существует лучшего решения чем a по крайней мере отличающегося на величину $(1+\varepsilon)$ по всем критериям. Развитие такого подхода к получению наименьшей возможной $(1+\varepsilon)$- Парето поверхности для решения задачи с двумя и тремя критериями представлено в работе [466].

Для решения MSPP задач с линейными критериями Пападимитриу и Яннакакис показали как можно использовать программу «gap», основанную на псевдополиномиальном алгоритме построения однокритериальных путей, для построения минимальных поверхностей Парето и предложили полностью полиномиальную схему аппроксимации. Наилучшими работами, в которых приведены полиномиальные технологии решения многокритериальных задач, в настоящее время следует признать работы [411, 463, 472].

В [266, 396] оспаривается экспоненциальность некоторых алгоритмов решения MSPP задачи. В частности показано, что на практике, некоторые комбинаторные алгоритмы, основанные на современных методах построения структур данных, не являются экспоненциальными. Другие авторы опровергают NP-полноту задач и предлагают методы решения за полиномиальное время. В [218, 296] приводятся интерактивные процедуры, позволяющие избежать генерации полного множества Парето - оптимальных решений и сосредоточить выбор только на самых перспективных решениях. Мартинс и Сантос [379] указывают

границы применения комбинаторных алгоритмов для решения MSPP задач и показывают, что они могут быть эффективны только для ациклических сетей. В [393] утверждается, что алгоритмы Мартинса и Сантоса хороши в теоретическом плане, однако не пригодны для решения практических задач из-за непомерных требований к оперативной памяти. Улучшенный MSPP алгоритм Мартинса и Сантоса был предложен в работе [266], экспериментальные исследования которого показали, что для определенных линейных критериальных функций он не ведет себя экспоненциально независимо от размерности сети.

Другие независимые исследования также подтверждают неэкспоненциальность двухкритериальных алгоритмов [396]. В [298] сравнивается эффективность комбинаторных и матричных алгоритмов для построения Парето - оптимальных решений. Отмечается, что комбинаторные алгоритмы работают всегда быстрее матричных, однако для большей эффективности алгоритмов требуется организация их параллельного многопроцессорного выполнения.

В работе [136] рассматривается двухкритериальная задача нахождения путей с минимальным соотношением стоимости и надежности. Для построения Парето - оптимальных решений предлагается параметрический сетевой симплекс алгоритм. В [476] описан алгоритм поиска многокритериальных кратчайших путей в стохастической сети.

В [464] предлагается полностью полиномиальная схема решения и новый метод генерации множества решений для MSPP задач с нелинейными критериальными функциями. Авторы показывают, как их результаты могут быть использованы для получения эффективных приближенных решений для задач с ограничениями и не аддитивными функциями. Алгоритм сходен с алгоритмом Беллмана-Форда и реализует очередь с приоритетами как полиномиальный массив Парето - оптимальных решений.

В работе [296] предложена интерактивная процедура, сводящая решение MSPP задачи к параметрическим однокритериальным задачам с последующим выбором наилучших решений. Авторы провели многочисленные эксперименты, и пришли к выводу о полезности такого подхода, который не требует гене-

рации полного множества Парето - оптимальных решений. Другой интерактивный подход предложен в [219]. Он основан на построении k - кратчайших путей для определения Парето - оптимальных решений для двухкритериальных задач. Отмечается, что предложенный алгоритм при проведении моделирования на большом количестве тестовых сетей работал быстрее по сравнению с другими известными аналогичными алгоритмами. Интерактивным алгоритмам посвящены также работы [261, 296, 398, 427, 465].

Однако, судя по литературным источникам и публикациям в Интернете, особых сдвигов в решении MSPP задач с использованием FPTAS-схем до сих пор не наблюдается. Поэтому заслуживает внимания другой подход, базирующийся на использовании эволюционных алгоритмов (evolutionary algorithm, EA) применительно к решению MSPP задач. Эволюционные алгоритмы представляют собой интеллектуальные адаптивные эвристические алгоритмы поиска, основанные на идеях естественного отбора и генетики и применяются во многих оптимизационных задачах. В своей работе они используют специальные выборки данных для обучения и при генерации новых поколений данных учитывают всю историю генетического развития процесса поиска решения. В результате эволюции и генетического отбора полученное решение наиболее полно удовлетворяет заданным внешним критериям оптимизации.

Как известно, эволюционные EA- алгоритмы широко используются для решения однокритериальных задач и лишь немногие исследователи применяли их для решения многокритериальных задач. Так, например, в работе [269] использован многокритериальный гибридный генетический алгоритм (GA) для решения задачи проектирования сети по двум конфликтующим критериям — минимум стоимости, максимум пропускной способности. В статье показано, как с помощью такого алгоритма можно за счет гибридизации с нечеткой логикой и локального поиска ускорить построение поверхностей Парето - оптимальных решений.

Кумар и Банерджи [355] представили алгоритм оптимизации кратчайших путей и остовных деревьев по двум критериям — сетевой задержки и стоимос-

ти с учетом ограничений на надежность и пропускные способности дуг. Они протестировали эволюционный подход, используя Pareto Converging Genetic Algorithm (PCGA), для построения по таким критериям сетей различной размерности и пришли к выводу о том, что для больших сетей ЕА- алгоритм лучше, чем использование традиционных схем Парето - аппроксимации с эвристическим и полным перебором решений за счет полиномиального времени генерации множества решений.

В работе [393] обсуждаются преимущества эволюционных генетических алгоритмов для решения задач в геоинформационных системах и проводится экспериментальное сравнение ЕА-алгоритмов с адаптированными алгоритмами Дейкстры на реальных сетях дорог. Отмечается, что по времени обе группы алгоритмов конкурируют друг с другом и ЕА- алгоритмы быстро сходятся к Парето - оптимальным путям.

Достаточно полное обсуждение многокритериальных эволюционных алгоритмов (MOEA) можно найти в [213, 225]. В работе [492] по мнению ее авторов, рассматривается один из наиболее эффективных современных MOEA- алгоритмов — алгоритм SPEA2. Отмечается, что эволюционные алгоритмы являются хорошей альтернативой для поиска множества Парето - оптимальных кратчайших путей и особенно привлекательны в случае, когда рассматриваются сети большой размерности и имеются ограничения на используемый объем оперативной памяти.

В работе [215] приводится обзор методов нахождения многокритериальных КП на нечетких графах с расплывчивыми значениями параметров ребер графа (fuzzy weighted graphs), которые часто используются для анализа сложных систем с неточно заданными и трудно - определяемыми параметрами. Предлагается схема формализации алгоритмов нахождения КП в нечетких условиях, основанная на методах расстановки и обновления меток и диффузификации весов и излагаются необходимые и достаточные условия для построения корректных алгоритмов. Показано, что метод диффузификации по центрам тяжести весов непригоден для построения нечетких КП.

Заключая обзор методов и алгоритмов нахождения КП, отметим, что для решения задач распределения и маршрутизации потоков, рассматриваемых в настоящей работе, наиболее востребованными являются алгоритмы лексикографической оптимизации построения кратчайших путей, удовлетворяющих требованиям упорядоченной последовательности критериев по нескольким параметрам.

Выводы

Проведенный анализ алгоритмов построения однокритериальных кратчайших путей показал, что в настоящее время одними из наилучших являются алгоритмы с оценками временной сложности $O(\min\{m + n\log n, m\log\log C, m + n\sqrt{\log C})\})$ для случая неотрицательных длин дуг, предложенные в [326], [256] и [145] и $O(mn)$ с отрицательными дугами для сетей, не содержащих циклов отрицательной длины [144], где n и m — число узлов и дуг в сети, а C — некоторая константа. В [208] указывается, что не существует наилучшего единого алгоритма для решения всего класса задач нахождения КП. За последние годы эволюция методов решения задачи нахождения КП была связана с разработкой и дальнейшим усовершенствованием специальных структур данных (АТД) для представления объектов задачи, позволяющих максимально сократить время построения КП [137-141, 209, 228-234, 285, 377] и созданием параллельных алгоритмов для многопроцессорного решения задачи [40, 81, 98, 354]. Было показано, что для решения отдельных SSSP задач существуют «почти оптимальные» алгоритмы в теории и на практике, в то же время для решения более широкого класса задач, включая и APSP-проблему, имеются предпосылки для улучшения уже существующих алгоритмов за счет аппаратного ускорения построения дерева кратчайших путей (PHAST) с использованием многоядерных процессоров и новейших графических адаптеров. Для многокритериальных задач о путях известны полностью полиномиальные схемы решения только для двухкритериальных задач [247]. Для случаев, когда число критериев оптимизации превышает два, пока не достигнуто значительных результатов.

Приложение Д. Структуры данных и процедуры редукции маршрутов в задачах распределения потоков

В задачах оптимизации распределения и маршрутизации потоков в коммуникационных сетях в качестве входных данных выступают маршруты транспортировки потоков (подраздел 4.1). В терминах теории графов маршрут представляет собой простой путь, заданный перечислением узлов и дуг сети. Допускаются кольцевые маршруты, у которых начальный и конечный узлы совпадают. Множество маршрутов может быть изначально задано или генерироваться по определенным правилам в процессе решения задачи. Поскольку маршруты имеют разную длину по числу входящих в них узлов (или дуг), более короткие маршруты могут совпадать с более длинными маршрутами, как по узлам, так и по дугам. Для распределения потоков в алгоритмах оптимизации используются специальные структуры данных — абстрактные типы данных (АТД) [72, 107], с помощью которых описывается связь между распределяемыми потоками и маршрутами. Процедуры редукции маршрутов позволяют значительно сократить число таких связей в случае, когда количество исходных или сгенерированных маршрутов в комбинаторных задачах оптимизации очень велико [32, 33].

Д.1. Постановка задачи и алгоритм решения

Пусть $G(N,P)$ — многопродуктовая сеть с множеством неориентированных топологических дуг P, $p=\left| P \right|$, множеством узлов N, $n=\left| N \right|$, где $\left|\cdot\right|$ — знак мощности множества. Под топологической дугой, как и ранее, будем понимать физический отрезок линии связи. Узлы сети соответствуют пунктам отправления, получения, сортировки и перегрузки (коммутации) потоков.

На сети задана матрица потоков $U=\left\| u_{ij} \right\|_{n\times n}$, $i,j=\overline{1,n}$. Потоки u_{ij} из источников i в стоки j, $i,j=\overline{1,n}$ должны перевозиться в транспортных средствах или передаваться по каналам связи с заданной периодичностью.

Пусть далее на сети задано множество маршрутов транспортных средств

или каналов связи $\{m_k\}$, $k=\overline{1,l}$, каждый из которых состоит из последовательности узлов и топологических дуг сети G, соединяющих начальный и конечный узлы маршрута или канала связи. В конкретной задаче оптимизации множество маршрутов может быть не заданным, а искомым. Для сетей передачи данных маршруты могут быть представлены простыми каналами связи, соединяющими смежные узлы или коммутируемыми каналами связи, соединяющими любую последовательность узлов (выделенными каналами). В частном случае, все заданные маршруты могут совпадать с топологическими дугами сети. Множество $\{m_k\}$ может содержать несколько маршрутов, соединяющих любую пару узлов. Определим также маршрутную мультисеть $G_M(N,P_M)$, где N — множество узлов сети, P_M — множество ее ориентированных маршрутных дуг. Между любыми узлами i и j сети G_M существует маршрутная дуга, если они связаны хотя бы одним маршрутом транспортного средства или каналом связи из $\{m_k\}$.

Как известно, вычислительная эффективность алгоритмов во многом зависит от удачного представления структур данных, используемых в процессе решения задачи. При решении задач распределения и маршрутизации потоков необходимо иметь такую структуру данных, которая, с одной стороны, позволяла бы для каждого потока u_{ij} быстро находить множество маршрутов, на которые может быть распределен данный поток, а также определять множество потоков, которые могут быть реализованы на данном маршруте. С другой стороны, структура данных должна обеспечивать хранение результатов решения задачи — схему распределения потоков, в которой для каждого u_{ij} определен путь следования от начального i-го узла до конечного j-го узла с указанием маршрутов и транзитных узлов. Другим важным требованием, предъявляемым к организации структуры данных, является минимум информации, хранимой в процессе решения задачи.

Пусть $\upsilon_k=\{\xi_1,\xi_2,...,\xi_{\eta_k}\}$ — упорядоченное множество узлов из N на мар-

шруте m_k, $|\upsilon_k| = \eta_k$, $q_k = \{(\xi_1, \xi_2), (\xi_2, \xi_3), ..., (\xi_{\eta_k - 1}, \xi_{\eta_k})\}$, $|q_k| = \eta_k - 1$ — упорядоченное множество дуг из P, составляющих маршрут m_k.

Поскольку в большинстве случаев прямые и обратные маршруты, как правило, совпадают, в целях экономии оперативной памяти компьютера, для совпадающих маршрутов представляется разумным хранить только прямые маршруты, подразумевая, что транспортировка потока по такому маршруту разрешена в обе стороны. Если прямой и обратный маршруты не совпадают, то каждый из них должен быть представлен отдельно.

Для различия маршрутов будем задавать признак направления движения по маршруту. Пусть $\{m_k\}$, $k = \overline{1,l}$ — множество маршрутов, полученное с учетом сделанных замечаний, а MV_k — признак направления движения по маршруту, где

$$MV_k = \begin{cases} 0, \text{если движение разрешено в обоих направлениях,} \\ 1, \text{если движение разрешено в одну сторону.} \end{cases}$$

Введем также дополнительные признаки маршрутов: $MT_k = \lambda$, где λ — целое число, определяющее тип маршрута;

$$MZ_k = \begin{cases} 0, \text{если движение по маршруту разрешено,} \\ 1, \text{если движение запрещено.} \end{cases}$$

Дополнительные признаки будут использоваться в алгоритмах редукции маршрутов и формирования справочной структуры данных.

Определим справочную структуру H_S, состоящую из матрицы указателей $\Lambda = \| \Delta_{ij} \|_{n \times n}$ и элементов E, связанных в однонаправленные линейные списки.

Указатель Δ_{ij} соответствует потоку u_{ij} и указывает на первый элемент E, называемый головой списка. Каждый элемент E может иметь ссылку на следующий элемент. Последний элемент в списке имеет нулевую ссылку (*NULL*-ссылку) и называется хвостовым. Каждый элемент E состоит из поля FWD, содержащего ссылку на следующий элемент или значение *NULL*; поля AM, указывающего адрес маршрута, на который может быть распределен данный

поток и поля признаков DM. Поле признаков состоит из пяти подполей, которые принимают следующие значения:

$$DM_1 = \begin{cases} 1, \text{если поток } u_{ij}, i < j \text{ выбран на маршрут } AM, \\ 0 \text{ в противном случае;} \end{cases}$$

$$DM_2 = \begin{cases} 1, \text{если поток } u_{ij}, i > j \text{ выбран на маршрут } AM, \\ 0 \text{ в противном случае;} \end{cases}$$

$$DM_3 = \begin{cases} 1, \text{если для } u_{ij} \text{ выполняется } (i < j) \wedge (i \prec j), \\ 0, \text{если для } u_{ij} \text{ выполняется } (i < j) \wedge (i \succ j); \end{cases}$$

$$DM_4 = \begin{cases} 1, \text{если для } u_{ij} \text{ выполняется } (i > j) \wedge (i \prec j), \\ 0, \text{если для } u_{ij} \text{ выполняется } (i > j) \wedge (i \succ j); \end{cases}$$

$$DM_5 = \begin{cases} 0, \text{если достигнут конец цепочки элементов } E, \text{определяющих маршруты,} \\ \text{на которых поток } u_{ij} \text{ может быть распределен без перегрузок,} \\ 1 \text{ в противном случае.} \end{cases}$$

Знаки $i \prec j$, $i \succ j$ соответственно означают, что узел i предшествует узлу j или следует за узлом j при движении по маршруту AM в порядке прохождения узлов из υ_{AM}.

Поток u_{ij} назовем безтранзитным, если для него выполняются условия: $\exists m_k \in \{m_k\}$, $k = \overline{1,l}$, для которого справедливо

$$(i, j \in \upsilon_k \wedge MV_k = 0) \vee (i, j \in \upsilon_k \wedge MV_k \neq 0 \wedge i \prec j).$$

В противном случае, поток будем называть транзитным. При начальном формировании структуры H_S, количество элементов E в списке для безтранзитного потока u_{ij} определяется числом маршрутов, на которые может быть распределен поток, причем, если симметричный поток u_{ji} также безтранзитный, то соответствующие указатели Δ_{ij} и Δ_{ji} получают ссылки на один список. Для транзитных потоков, которые должны быть один или более раз перегружены с одного маршрута на другой, соответствующие указатели из Λ первоначально получают нулевые ссылки. Построение цепочек элементов E для транзитных потоков u_{ij} осуществляется в процессе решения задачи и заключается в следу-

ющем: определяются маршруты, на которые может быть распределен поток; для найденных маршрутов создаются элементы E, которые связываются в список в соответствии с порядком следования по маршрутам от i до j. В справочной H_S для каждого транзитного потока запоминается только один список, определяющий путь, который был выбран для распределения данного потока в процессе оптимизации, промежуточные списки не запоминаются. Если в процессе решения задачи безтранзитный поток переводится в ранг транзитных, то выбранный для него список связывается с первоначальным. Для разграничения списков в этом случае используется подполе DM_5. Для транзитных потоков каждый элемент E содержит дополнительное поле AY, в которое записывается узел маршрута AM, в котором перегружается поток. Последний элемент E в цепочке для транзитного потока u_{ij} в поле AY будет содержать узел j. В случае, когда при решении задачи разрешено разветвление (дробление) потоков в структуру элементов E могут быть введены дополнительные поля для хранения части распределенного потока.

На рис. Д.1 показан пример структуры H_S, поясняющий введенные понятия.

Из правил начального формирования H_S, ясно, что число элементов E в списках H_S зависит от числа маршрутов и определяется следующим образом:

$$E_{HS} = \sum_{k=1}^{l} \eta_k (\eta_k - 1)/2.$$

Величина E_{HS} для реальных сетей может быть весьма значительной. Так, например, для сети содержащей 300 узлов при генерации для всех симметричных $n(n-1)/2$ потоков кратчайших путей распределения потоков (отдельных маршрутов) будет сгенерировано 44850 маршрутов. При средней длине маршрута $\eta_{ср}$ = 15 в структуру H_S будет включено $44850\eta_{ср}(\eta_{ср}-1)/2 = 44850 \times 105 = 4709250$ элементов, для размещения которых потребуется порядка 45 Мб оперативной памяти (учитывая, что для размещения одного элемента E требуется 10 байтов).

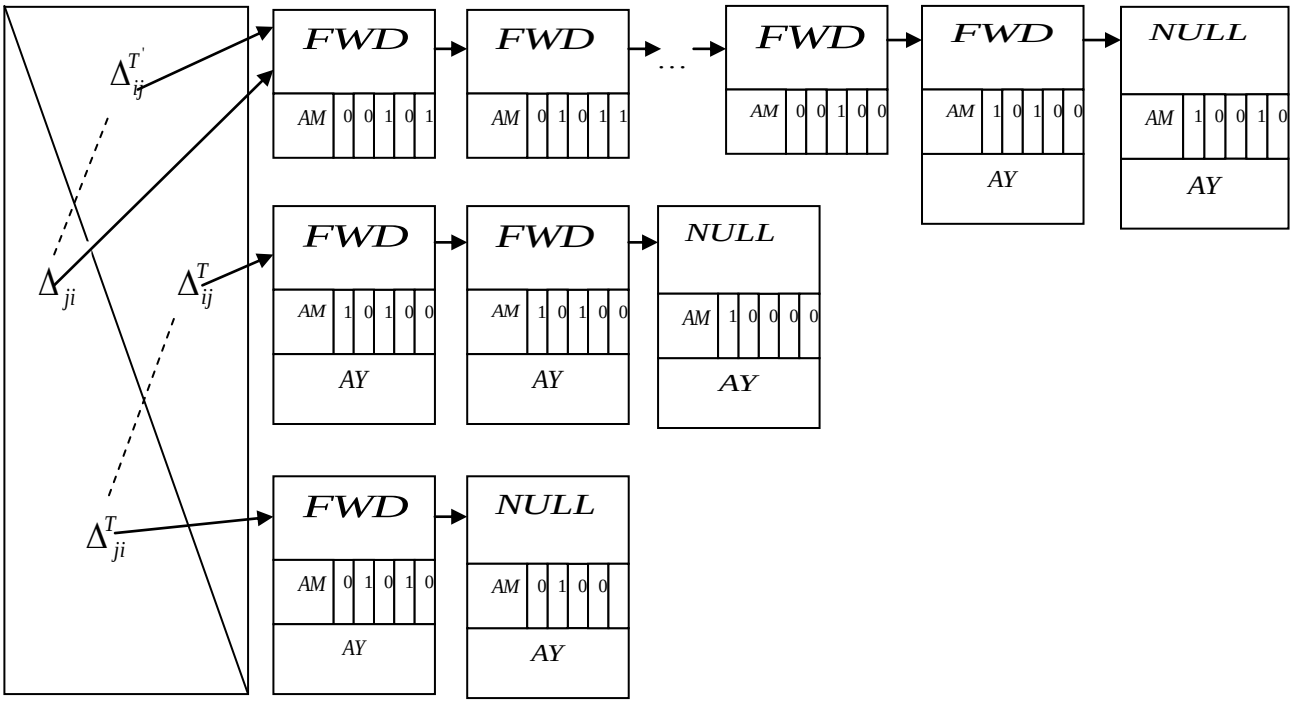


Рис. Д.1. Где: $\Delta_{ij}^{T'}$ — безтранзитный поток, переведенный в ранг транзитных; Δ_{ji} — безтранзитный поток; Δ_{ij}^{T}, Δ_{ji}^{T} — симметричные транзитные потоки

В связи с этим, рассмотрим процедуры редукции, позволяющие сократить размерность структуры H_S. Пусть маршрут m_α редуцируется в маршрут m_β по узлам, если выполняется условие:

$$\upsilon_\alpha \cap \upsilon_\beta = \upsilon_\alpha,$$

и редуцируется по топологическим дугам, если

$$q_\alpha \cap q_\beta = q_\alpha.$$

Очевидно, что редукция маршрутов сокращает количество связей в сети G_M и число элементов в структуре H_S на величину:

$$\sum_{\alpha \in \{k^*\}} \eta_\alpha (\eta_\alpha - 1)/2,$$

где $\{k^*\}$ — множество номеров редуцированных маршрутов.

Пусть далее LM_i, $i = \overline{1,l}$ — вектор длин маршрутов по числу узлов, VM_i, $i = \overline{1,l}$ — вектор номеров маршрутов, упорядоченных по убыванию числа узлов в маршруте. В процессе редукции маршрутов будем строить вектор редукции VR_i, $i = \overline{1,l}$ и справочный вектор RV_i, $i = \overline{1,l}$. Элементы векторов определим

следующим образом:

$$VR_i = \begin{cases} 0, \text{если } i\text{-ый маршрут не редуцирован ни в какой другой маршрут,} \\ k, \text{если } i\text{-ый маршрут редуцирован в некоторый маршрут } \xi. \end{cases}$$

Где k — номер маршрута, который был редуцирован в маршрут ξ непосредственно перед маршрутом i. Если i — первый редуцируемый маршрут, то $k = \xi$.

$$RV_i = \begin{cases} 0, \text{ если } i\text{-ый маршрут редуцирован,} \\ \xi, \text{ если в маршрут } i \text{ редуцирован какой-либо маршрут,} \\ \text{но сам он не редуцирован ни в какой другой маршрут,} \\ i, \text{ если маршрут } i \text{ сам не редуцирован ни в какой другой} \\ \text{и в него не редуцирован ни один маршрут.} \end{cases}$$

Где ξ — номер последнего маршрута, редуцированного в маршрут i.

Каждый элемент вектора $VR_i = 0$, соответствующий маршруту m_i определяет цепочку редуцированных в этот маршрут других маршрутов. Упорядочение маршрутов в цепочке обратно порядку их редукции. На первый маршрут в цепочке для VR_i указывает элемент RV_i.

Приведем алгоритм редукции. Пусть $S = \| s_{ij} \|_{l \times \eta_{\max}}$ — матрица маршрутов, заданная перечислением номеров узлов сети, где $\eta_{\max}$ — максимальное число узлов в маршруте. Предполагается, что номера недостающих узлов в более коротких маршрутах заменены на 0.

Обозначим через $BR1$, $BR2$ — рабочие векторы размерностью n. Для большей наглядности в записи псевдокода алгоритма для индексов матриц и векторов будем использовать круглые скобки.

Алгоритм Д.1. Редукция маршрутов

1. Для $\{ i \mid i = \overline{1,l} \}$ выполнить $RV(i) \leftarrow i$, $VR(i) \leftarrow 0$.

2. Если выбрана однородная редукция, то $RT \leftarrow 'O'$, для смешанной редукции $RT \leftarrow 'M'$. *** Для однородной редукции типы маршрутов $MT_k = \lambda$ совпадают

3. Если выбрана редукция по узлам, то $PRT \leftarrow 'U'$, для редукции по дугам $PRT \leftarrow 'A'$.

4. Если $PRT = 'U'$, то перейти к п. 5, иначе перейти к п. 25.

5. Для $\{ i \mid i = \overline{1, l-1} \}$ выполнить пп. 6-24. *** Начало цикла по i, редукция по узлам

6. $K1 \leftarrow VM(i)$.

7. Если $VR(VM(i)) = 0$, то перейти к п. 8, иначе к п. 24.

8. $BR1 \leftarrow 0$.

9. Для $\{ k \mid k = \overline{1, LM(VM(i))} \}$ выполнить $BR1(S(VM(i), k)) \leftarrow 1$.

10. Для $\{ j \mid j = \overline{i+1, l} \}$ выполнить пп. 11-23. *** Начало цикла по j

11. Если $VR(VM(j)) = 0$, то перейти к п. 12, иначе к п. 23.

12. $BR2 \leftarrow 0$.

13. Для $\{ k \mid k = \overline{1, LM(VM(j))} \}$ выполнить $BR2(S(VM(j), k)) \leftarrow 1$.

14. Если $RT = 'M' \vee RT = 'O' \wedge MT(VM(i)) = MT(VM(j))$, то перейти к п. 15, иначе к п. 23.

15. Для $\{ k \mid k = \overline{1, n} \}$ выполнить пп. 16-18. *** Начало цикла по k

16. $BR2(k) = 1$, то перейти к п. 17, иначе к п. 18.

17. Если $BR1(k) = 1$, то перейти к п. 18, иначе к п. 23.

18. Перейти к п. 15. *** Конец цикла по k

19. $RV(VM(j)) \leftarrow 0$.

20. $RV(VM(i)) \leftarrow VM(j)$.

21. $VR(VM(j)) \leftarrow K1$.

22. $K1 \leftarrow VM(j)$.

23. Перейти к п. 10. *** Конец цикла по j

24. Перейти к п. 5. *** Конец цикла по i

25. Для $\{ i \mid i = \overline{1, l-1} \}$ выполнить пп. 26-55. *** Начало цикла по i, редукция по дугам

26. $KS \leftarrow VM(i)$.

27. Если $VR(VM(i)) \neq 0$, то перейти к п. 55, иначе к п. 28.

28. Для $\{ j \mid j = \overline{i+1, l} \}$ выполнить пп. 29-54. *** Начало цикла по j

29. Если $VR(VM(j)) \neq 0$, то перейти к п. 54, иначе к п. 30.

30. $KL \leftarrow 0;\ KF \leftarrow 1$.

31. Для $\{ k1 \mid k1 = \overline{1, LM(VM(i))} \}$ выполнить пп. 32-34.

32. Если $S(VM(j), KF) = S(VM(i), k1)$, то перейти к п. 36.

33. Если $S(VM(j), LM(VM(j))) = S(VM(i), k1)$, то перейти к п. 48.

34. Перейти к п. 31.

35. Перейти к п. 54, продолжить цикл по j.

36. Для $\{ k \mid k = \overline{1, LM(VM(j))} \}$ выполнить пп. 37-40.

37. Если $S(VM(j), k) = S(VM(i), k1)$, то перейти к п. 38, иначе к п. 40.

38. $KL \leftarrow KL + 1;\ k1 \leftarrow k1 + 1$.

39. Если $k1 > LM(VM(i))$, то перейти к п. 41.

40. Перейти к п. 36.

41. Если $RT = 'M' \vee RT = 'O' \wedge MT(VM(i)) = MT(VM(j))$, то перейти к п. 42, иначе к п. 47.

42. Если $KL = LM(VM(j))$, то перейти к п. 43, иначе к п. 47.

43. $RV(VM(j)) \leftarrow 0$.

44. $RV(VM(i)) \leftarrow VM(j)$.

45. $VR(VM(j)) \leftarrow KS$.

46. $KS \leftarrow VM(j)$.

47. Перейти к п. 54, продолжить цикл по j.

48. Для $\{ k \mid k = \overline{LM(VM(j)), 1, -1} \}$ выполнить пп. 49-52.

49. Если $S(VM(j), k) = S(VM(i), k1)$, то перейти к п. 50, иначе к п. 52.

50. $KL \leftarrow KL + 1;\ k1 \leftarrow k1 + 1$.

51. Если $k1 > LM(VM(i))$, то перейти к п. 41.

52. Перейти к п. 48.

53. Перейти к п. 41.

54. Перейти к п. 28. *** Конец цикла по j

55. Перейти к п. 25. *** Конец цикла по i

56. Стоп.

В результате работы приведенного алгоритма цепочки в векторе редукции для $VR_i = 0$, на начало которых указывает RV_i, будут упорядочены по возрастанию числа узлов в редуцированных маршрутах.

После выполнения процедуры редукции строится справочная структура. При этом просматриваются только такие маршруты, у которых $VR_i = 0$. Очевидно, что после выполнения процедуры редукции объем оперативной памяти компьютера, занимаемый справочной H_S, сократится на величину:

$$\Delta = \left(\sum_{k=1}^{l} \eta_k (\eta_k - 1)/2 - \sum_{\alpha \in \{k^*\}} \eta_\alpha (\eta_\alpha - 1)/2 \right) V_E,$$

где V_E — объем памяти, занимаемый одним элементом списка.

Из алгоритма видно, что временная сложность редукции по узлам составляет в среднем $O(l^2(n+\eta_{cp}) + l\eta_{cp})$, а по дугам — $O(l^2\eta_{cp})$ операций, где η_{cp} — средняя длина маршрута.

Необходимо отметить, что редукция маршрутов несколько увеличит трудоемкость основного алгоритма оптимизации, так как возрастет число просматриваемых маршрутов в шагах выбора маршрутов для распределения потоков. Несмотря на это, редукция маршрутов оказывается крайне необходимой, если в качестве исходного множества маршрутов выступают все возможные маршруты. В этом случае резко возрастает размерность задачи, принимающей характер построения новых, ранее не существующих маршрутов.

Рассмотрим алгоритм начального формирования справочной H_S с одновременным построением маршрутных дуг сети G_M.

Пусть $R = \| r_{ij} \|_{n \times n}$ — топологическая матрица сети G, r_{ij} — длина дуги p_{ij}

(если дуги p_{ij} не существует, $r_{ij}=0$); $D=\left\| d_{ij} \right\|_{n\times n}$ — матрица длин маршрутных дуг (первоначально $d_{ij}=\infty$, $i,j=\overline{1,n}$).

Поскольку любые два узла могут быть связаны более чем одним маршрутом, длины дуг d_{ij} будем определять из следующего выражения:

$$d_{ij}=\min_{\{\upsilon_k\}}(r_{\xi_\mu,\xi_{\mu+1}}+r_{\xi_{\mu+1},\xi_{\mu+2}}+\ldots+r_{\xi_{\mu+c-1},\xi_{\mu+c}}),\ \text{при}\ \xi_\mu=i,\ \xi_{\mu+c}=j,\ k=\overline{1,l}.$$

В описании алгоритма запись «Образовать новый элемент $E(P)$» означает генерацию нового элемента E и установку указателя P на адрес элемента. Обращение к полям элемента будем записывать в виде $P.\langle\text{наименование поля}\rangle$, а установку указателя в виде $P\Rightarrow$.

Алгоритм Д.2. Формирование H_S

1. Для $\{i \mid i=\overline{l,1,-1}\}$ выполнить пп. 2-18. *** Начало цикла по i, формирование H_S

2. Если $VR(VM(i))=0\wedge MZ(i)=0$ то перейти к п. 3, иначе к п. 18.

3. Для $\{k \mid k=\overline{1,LM(VM(i))-1}\}$ выполнить пп. 4-17.***Начало цикла по k

4. Для $\{j \mid j=\overline{k+1,LM(VM(i))}\}$ выполнить пп. 5-16.***Начало цикла по j

5. Образовать новый элемент $E(P)$.

6. Если $\Lambda(S(VM(i),k),S(VM(i),j)).FWD\neq NULL$, то перейти к п. 7, иначе к п. 12.

7. $P1\Rightarrow\Lambda(S(VM(i),k),S(VM(i),j)).FWD$.

8. Пока $P1.FWD\neq NULL$, выполнить пп. 9-10.

9. $P1\Rightarrow P1.FWD$.

10 Перейти к п. 8.

11. $P1.FWD\Rightarrow P$. Перейти к п. 14.

12. $\Lambda(S(VM(i),k),S(VM(i),j)).FWD\Rightarrow P$.

13. $\Lambda(S(VM(i),j),S(VM(i),k)).FWD\Rightarrow P$. Перейти к п. 14.

14. $P.FWD\Rightarrow NULL$; $P.AM\leftarrow VM(i)$; $P.DM\leftarrow 0$; $P.DM(5)\leftarrow 1$.

15. Если $S(VM(i),k) < S(VM(i), j))$, то $P.DM(3) \leftarrow 1$, иначе $P.DM(4) \leftarrow 1$.

16. Перейти к п. 4. *** Конец цикла по j

17. Перейти к п. 3. *** Конец цикла по k

18. Перейти к п. 1. *** Конец цикла по i

19. $D \leftarrow \infty$.

20. Для $\{ i \mid i = \overline{1,l} \}$ выполнить пп. 21-30. *** Начало цикла по i, формирование D

21. Для $\{ k \mid k = \overline{1, LM(i) - 1} \}$ выполнить пп. 22-29. *** Начало цикла по k

22. $SUM \leftarrow 0$.

23. Для $\{ j \mid j = \overline{k+1, LM(i)} \}$ выполнить пп. 24-28. *** Начало цикла по j

24. $m \leftarrow j - 1$.

25. $SUM \leftarrow SUM + R(S(i,m), S(i, j))$.

26. Если $D(S(i,k), S(i, j)) > SUM$, то $D(S(i,k), S(i, j)) = SUM$.

27. Если $MV(i) = 0 \wedge D(S(i, j), S(i,k)) > SUM$, то $D(S(i, j), S(i,k)) = SUM$.

28. Перейти к п. 23. *** Конец цикла по j

29. Перейти к п. 21. *** Конец цикла по k

30. Перейти к п. 20. *** Конец цикла по i

31. Стоп.

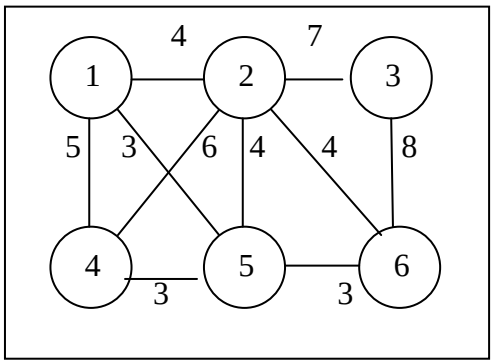
Как нетрудно видеть средняя трудоемкость алгоритма Д.2 составляет $O((l - lr)\eta_{ср}^2)$ операций, где lr — число редуцированных маршрутов, а $\eta_{ср}$ — средняя длина маршрута.

Для начального распределения потоков на сети G_M используется алгоритм построения кратчайших путей по критерию: минимум транзитных перегрузок; при равенстве числа перегрузок — минимум длины пути (подраздел 4.3).

Д.2. Пример формирования справочной структуры

Пусть задана сеть с числом узлов $n = 6$, числом дуг $p = 10$ и матрицей

длин дуг R, приведенная на рис. Д.2. После редукции заданных маршрутов по узлам (U) и дугам (A) были получены результаты, показанные в табл. Д.1 и на рис. Д.3.



R						
N	1	2	3	4	5	6
1	**0**	**4**	**0**	**5**	**3**	**0**
2	**4**	**0**	**7**	**6**	**4**	**4**
3	**0**	**7**	**0**	**0**	**0**	**8**
4	**5**	**6**	**0**	**0**	**3**	**0**
5	**3**	**4**	**0**	**3**	**0**	**3**
6	**0**	**4**	**8**	**0**	**3**	**0**

Рис. Д.2

Таблица Д.1

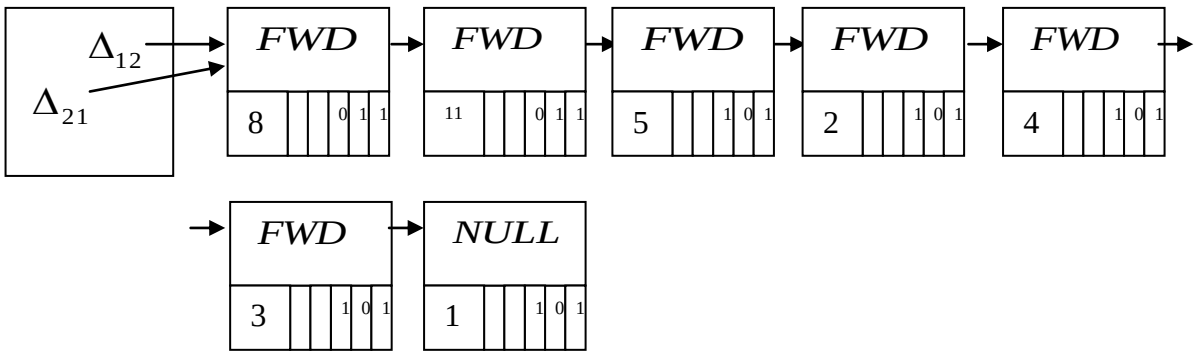
Параметры	Редукция	
	по узлам U	по дугам A
Число узлов в сети	6	6
Число дуг в сети	10	10
Число маршрутов в сети	14	14
Число редуцированных маршрутов, в.т. ч. в %	13, 92%	6, 42%
Число элементов в структуре АТД	72	72
Число редуцированных элементов, в.т. ч. в %	57, 79%	17, 23%

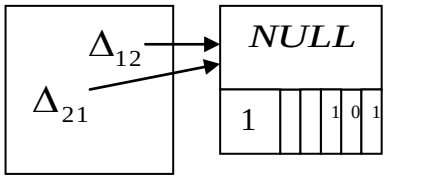
На рис. Д.4 приведены фрагменты справочной структуры H_S для потоков u_{12} и u_{21} без редукции маршрутов и с редукцией по узлам и дугам. При распределении потоков в первом случае маршруты будут просматриваться в следующем порядке — 8, 11, 5, 2, 4, 3, 1; во втором — 6, 13, 8, 7, 11, 10, 5, 14, 12, 2, 4, 3, 9, 1; в третьем — 8, 6, 5, 4, 10, 3, 1, 13, 11, 2, 1. Таким образом, как уже отмечалось, после редукции маршрутов в основном алгоритме оптимизации общее число просмотров маршрутов увеличивается, что соответственно приводит к увеличению его трудоемкости. Кроме того, в случае редукции по дугам просмотр маршрутов по возрастанию числа узлов может быть нарушен, если какой-либо один и тот же поток может проходить по путям с разными дугами. В

нашем примере, это оба потока — u_{12} и u_{21} (маршруты 1, 2, 3, 4, 5, 11 и маршрут 8).

№	$S = \| s_{ij} \|_{14\times6}$						LM_{14}	VM_{14}	VR_{14}	RV_{14}	VR_{14}	RV_{14}
									U		A	
1	1	2	3	6	5	4	6	1	0	6	0	13
2	1	2	3	6	0	0	4	9	4	0	1	0
3	1	2	6	5	0	0	4	3	9	0	0	3
4	1	2	5	6	0	0	4	4	3	0	0	10
5	1	2	4	0	0	0	3	2	14	0	0	5
6	1	4	0	0	0	0	2	12	13	0	8	0
7	2	4	5	0	0	0	3	14	11	0	9	0
8	2	4	1	0	0	0	3	5	7	0	0	6
9	2	4	5	6	3	0	5	10	1	0	0	7
10	2	5	6	0	0	0	3	11	5	0	4	0
11	3	2	1	0	0	0	3	7	10	0	2	0
12	3	2	5	6	0	0	4	8	2	0	0	12
13	3	6	0	0	0	0	2	13	8	0	11	0
14	4	5	2	6	0	0	4	6	12	0	0	14

Рис. Д.3





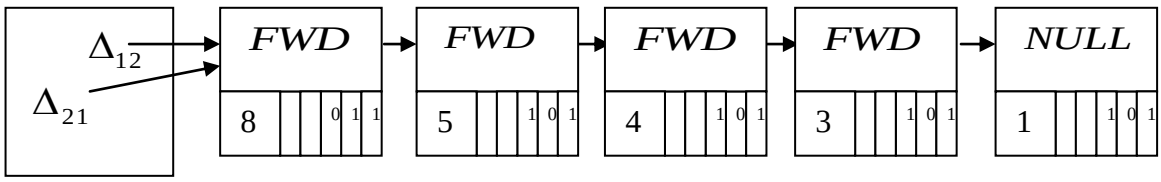


Рис. Д.4

На рис. Д.5 показана матрица длин кратчайших путей D и справочная матрица построенных путей $C = \| c_{ij} \|_{n \times n}$, каждый элемент которой c_{ij}, $i \neq j$ определяет номер предпоследнего узла на кратчайшем пути от i до j, $c_{ii} = 0$, $i = \overline{1,n}$. Матрицы получены для сети G_M алгоритмом из подраздела 4.3.

D						
N	1	2	3	4	5	6
1	**0**	**4**	**11**	**5**	**8**	**8**
2	**4**	**0**	**7**	**6**	**4**	**4**
3	**11**	**7**	**0**	**14**	**11**	**8**
4	**5**	**6**	**14**	**0**	**3**	**6**
5	**8**	**4**	**11**	**3**	**0**	**3**
6	**8**	**4**	**8**	**6**	**3**	**0**

C						
N	1	2	3	4	5	6
1	**0**	**1**	**1**	**1**	**1**	**1**
2	**2**	**0**	**2**	**2**	**2**	**2**
3	**3**	**3**	**0**	**3**	**3**	**3**
4	**4**	**4**	**4**	**0**	**4**	**4**
5	**5**	**5**	**5**	**5**	**0**	**5**
6	**6**	**6**	**6**	**6**	**6**	**0**

Д.3. Результаты вычислительного эксперимента

Для проверки работоспособности и вычислительной эффективности алгоритмов был проведен вычислительный эксперимент на ПК с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб под управлением операционной системы Windows Vista. Проверка алгоритмов осуществлялась на сетях, содержащих от 50 до 500 узлов. Для всех размерностей датчиком псевдослучайных чисел генерировались однородные сети с числом исходящих дуг из каждого узла (степенью узлов) $val = 5$ и длинами топологических дуг в пределах от 80 до 320. Результаты расчетов приведены в табл. Д.2 и на рис. Д.6-Д.9. Где *lr* и *lr*%, E, *Er* и *Er*%, *TFLOYD*, *TRED,U* , *TRED, A*, *TFORM* , *TVVSP* — соответственно число и процент редуцированных маршрутов; число сгенерированных и число и процент редуцированных элементов E в структуре H_S; время построения кратчайших путей по критерию — минимум длины пути на сети G алгоритмом Флойда для формирования маршрутов для всех $n^2 - n$ потоков; время редукции маршрутов по узлам и дугам; время формирования структуры H_S; время построения кратчайших путей по критерию — минимум транзитных перегрузок, при равенстве числа перегрузок — минимум длины пути на сети G_M.

Таблица Д.2

n	50	100	150	200	250	300	400	500
val	5	5	5	5	5	5	5	5
p	125	250	375	500	625	750	1000	1250
l	1225	4950	11175	19900	31125	44850	79800	124750
$\eta_{\max}$	7	9	10	13	11	12	13	14
η_{cp}	3	4	5	5	6	6	6	6
lr $lr\%$	789, 64%	3382, 68%	7764, 69%	14165, 71%	22102, 71%	31336, 69%	55682, 69%	88619, 71%
E	7797	48961	138596	301079	510913	786289	1618257	2822650
Er $Er\%$	3904, 50%	27196, 55%	79606, 57%	181025, 60%	308204, 60%	467208, 59%	967801, 59%	1743565, 61%
$TFLOYD$	0,02	0,02	0,05	0,08	0,16	0,28	0,64	1,20
$TRED,U$ $TRED,A$	0,02 0,00	0,28 0,16	2,12 0,61	6,32 1,81	17,88 4,35	53,38 9,47	184,81 31,47	546,85 85,47
$TFORM$	0,00	0,02	0,03	0,06	0,11	0,27	1,11	2,65
$TVVSP$	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00

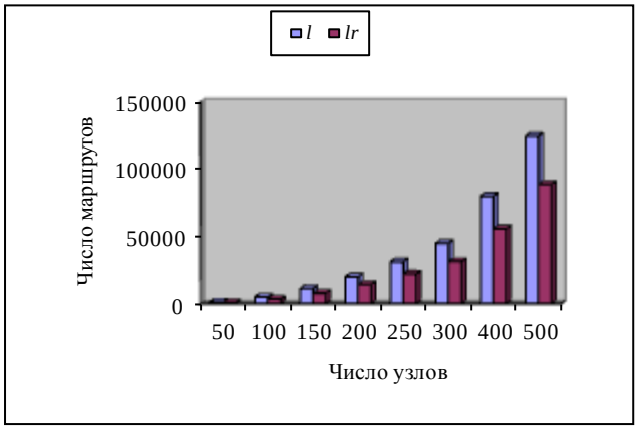


Рис. Д.6

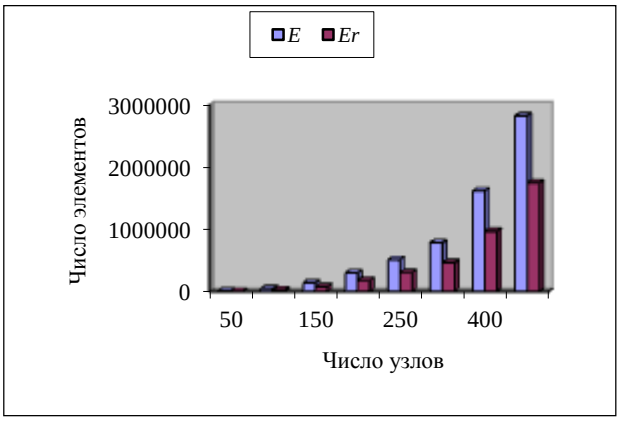


Рис. Д.7

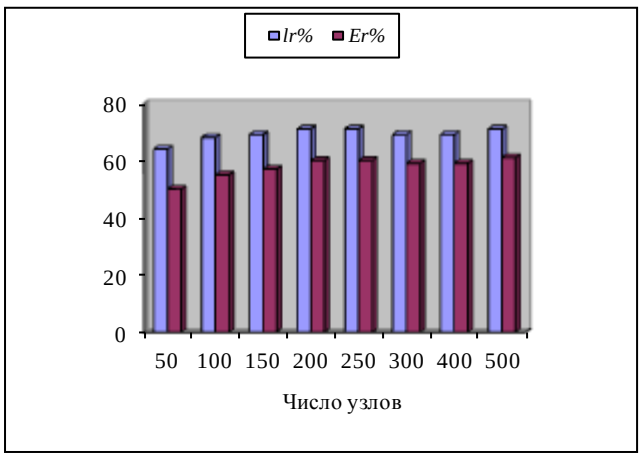


Рис. Д.8

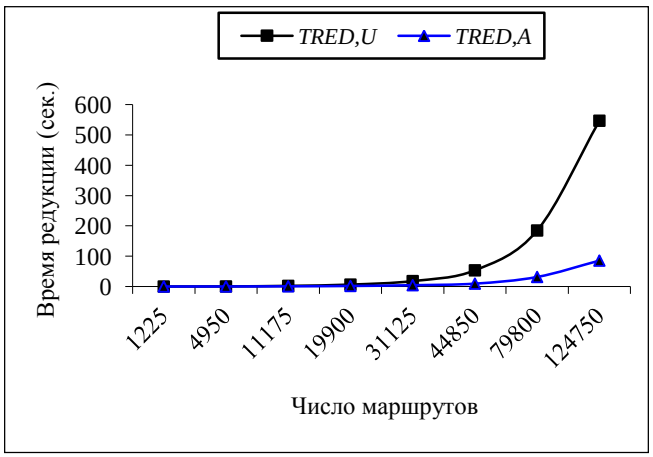


Рис. Д.9

Выводы

Предложены эффективные структуры данных, основанные на линейных динамических списках, с помощью которых можно быстро найти все возможные маршруты для распределения потоков в многопродуктовой сети. Разработаны алгоритмы редукции маршрутов, которые позволяют решать задачи распределения потоков в случае, когда число рассматриваемых маршрутов очень велико, а объем оперативной памяти компьютера ограничен.

Получены оценки трудоемкости алгоритмов редукции маршрутов по узлам — $O(l^2(n+\eta_{ср})+l\eta_{ср})$ и дугам — $O(l^2\eta_{ср})$, а также алгоритма формирования справочной структуры данных — $O((l-lr)\eta_{ср}^2)$, где l и n — число маршрутов и узлов в сети, $\eta_{ср}$ — средняя длина маршрута, lr — число редуцированных маршрутов. Проведено тестирование предложенных алгоритмов на сетях с числом узлов от 50 до 500 и числом маршрутов от 1225 до 124750, которое показало их работоспособность и хорошую вычислительную эффективность и они могут быть использованы в практических задачах распределения и маршрутизации потоков на сетях большой размерности.

Приложение Е. Вычислительный эксперимент к пятому разделу

Цель эксперимента заключается в том, чтобы показать как можно использовать решение задач текущего планирования для получения исходных данных для перспективного развития узлов и транспортных линий магистральной сети, а также для оперативного перераспределения потоков в случае возникновения перегрузок и аварийных ситуаций.

Для проведения экспериментов выбрана сгенерированная датчиком псевдослучайных чисел транспортная сеть с числом узлов $n = 12$ и числом исходящих дуг из каждого узла (степенью или валентностью узлов) $val = 5$. Длины дуг d_{kl}, $kl \in P$ генерировались в пределах от 80 до 300 км. Размер (объем) транспортного блока (ω) принимался равным 40 единицам мелкопартионных грузов, а грузоподъемность транспортных средств (W) (параметры w_ξ и w_i^k в моделях) могла составлять 20 и 5 транспортных блоков и выбиралась одинаковой для всех маршрутов. Предполагалось, что маршруты транспортных средств совпадают с дугами сети, по которым разрешено движение в прямом и обратном направлениях, поэтому изначально было сгенерировано $n*val/2=12*5/2=30$ маршрутов. Время на обработку (сортировку) мелкопартионных грузов в узлах сети составляло 1 сутки, время на транзитную перегрузку транспортных блоков в узлах сети — 0,5 суток, средняя скорость движения транспортных средств принималась равной 80 км/час. Пропускные способности узлов по обработке транзитных потоков мелкопартионных грузов (h_i, $i = \overline{1,n}$) и транспортных блоков (b_i, $i = \overline{1,n}$), максимальное время доставки мелкопартионных грузов получателю ($T_{ij}, \forall ij \in S$), максимальное число объединений (слияний) потоков ($\nu_{\max}$) не ограничивались. Средняя задержка транспортных блоков в сети (t_{av}), рассчитывалась по формуле (4.15), а максимальная задержка $T_{\max}$ не ограничивалась. При распределении потоков транспортных блоков по маршрутам транспортных средств запрещалось их разветвление (дробление). В качестве функций затрат использовались формулы в условных единицах стоимости из подраздела 4.6.

Для расчета времени доставки мелкопартионных грузов конечному получателю применялись формулы (3.24), приведенные в подразделе 3.3.2.

Исходящие потоки из каждого узла сети также задавались датчиком псевдослучайных чисел при изменении величины начальных потоков от одной до девяти единиц для узлов первого типа и от одной до пяти единиц для узлов второго и третьего типов. Для сети с $n = 12$ и $val = 5$ было сгенерировано два узла второго типа и один узел третьего типа. На рис. Е.1 показана начальная структура сети, представленная в виде дерева. Такое отображение сети позволяет легко видеть пути распределения потоков, состоящие из одной, двух и более дуг.

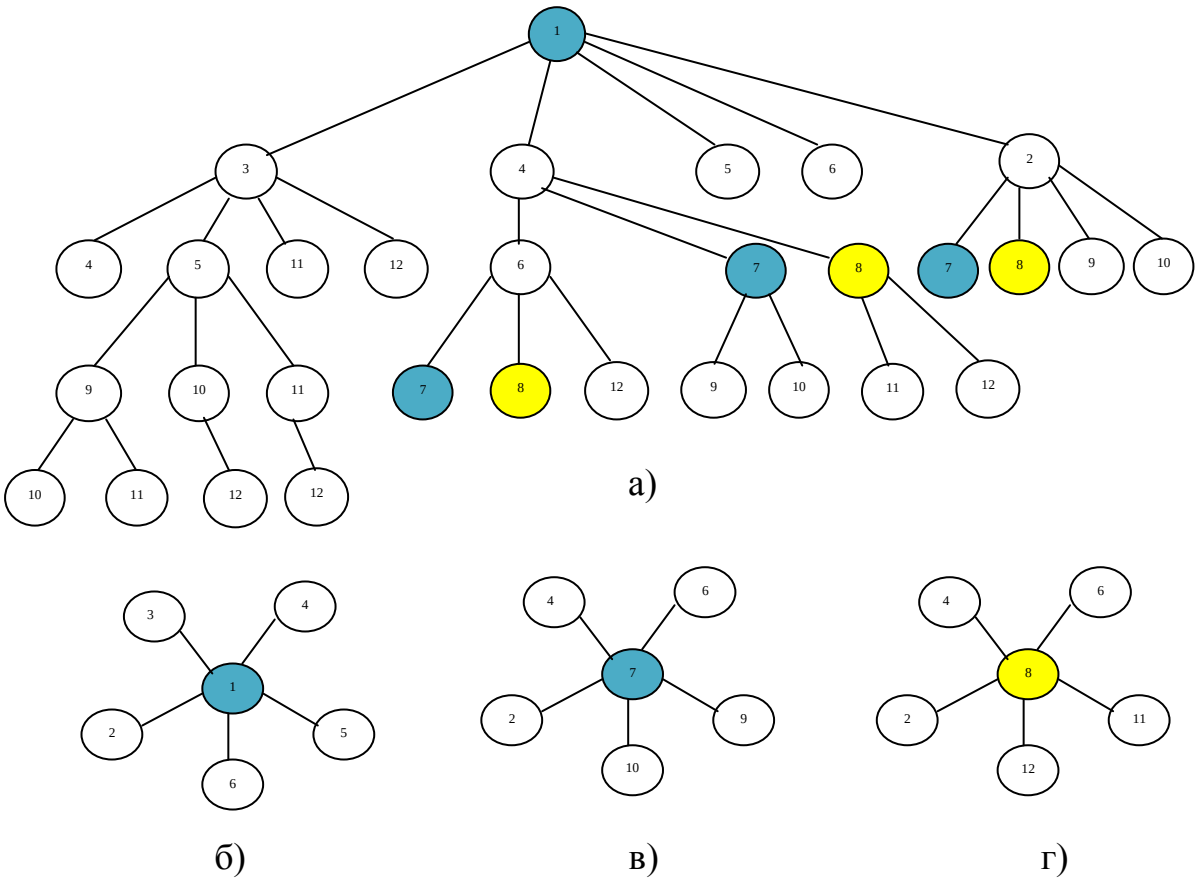


Рис. Е.1. а) сеть с $n = 12$ и $val = 5$, представленная в виде дерева; б), в) — зональные уровни узлов второго типа; г) зональный уровень узла третьего типа

В начале эксперимента была выполнена оптимизация структуры сети и распределены потоки транспортных блоков для пяти этапов ее развития при прогнозируемом пропорциональном изменении исходящих потоков соответственно в 1, 2, 1/2, 10 и 30 раз. Учитывая значительный рост потоков на последних

этапах, на всех маршрутах была выбрана грузоподъемность транспортных средств $W = 20$. На всех этапах развития в структуру сети не было включено ни одного узла второго или третьего типа, т.е. все узлы в сети
получили статус узлов первого типа. Затраты и все параметры до оптимизации получены когда оптимизация структуры и схемы сортировки мелкопартионных грузов для исходной сети, содержащей узлы второго и третьего типов не выполнялась, а оптимизация распределения и маршрутизации потоков проводилась. Задача распределения порожних транспортных блоков (контейнеров) решалась как для исходной, так и для оптимизированной структуры сети. Основные результаты расчетов приведены в табл. Е.1 и на рис. Е.2. Здесь FS — число объединений (слияний) потоков, при котором получено оптимальное решение; KF — коэффициент прогнозирования потоков; K_{av} — средний коэффициент загрузки транспортного блока в сети; Q_{av} — среднее число магистральных направлений сортировки в узле сети; K_{opt} — число транспортных блоков, необходимых для единовременного отправления всех потоков в сети; t_{av} — средняя задержка транспортных блоков в сети; $MINF$, $MAXF$ — минимальный и максимальный поток по дуге сети в транспортных блоках; C_{all}, C_{tr}, C_{load}, C_{sort}, EF — соответственно суммарные среднегодовые приведенные затраты, затраты на транспортировку и погрузку-выгрузку транспортных блоков, затраты на сортировку потоков и эффект от оптимизации для этапов развития сети (в усл. ед.); $t_{\min}$, $t_{\max}$, t_{mean} — минимальное, максимальное и среднее время доставки мелкопартионных грузов получателю (в сутках).

Из табл. Е.1 видно, что оптимальные решения на этапах развития сети достигаются при различных значениях параметра FS. С ростом потоков увеличиваются значения параметров K_{av} и Q_{av}, а эффект от оптимизации EF (разница между полными затратами до и после оптимизации) уменьшается. На первых трех этапах очень низкий средний коэффициент загрузки транспортных средств, имеются незагруженные маршруты, и неиспользованные транспортные средства могут быть сданы в аренду.

Таблица Е.1

Результаты решения для прогнозируемых потоков и этапов развития до и после оптимизации структуры сети, распределения и маршрутизации потоков при $W = 20$

Наименование показателей	Значение показателей до и после оптимизации				
	Этап 1 *FS* =5, *KF* =1	Этап 2 *FS* =2, *KF* =2	Этап 3 *FS* =6, *KF* = 0,5	Этап 4 *FS* =2 *KF* =10	Этап 5 *FS* =1, *KF* =30
Средний коэффициент загрузки контейнера (K_{av})	0,157 0,566	0,315 0,739	0,091 0,563	0,728 0,924	0,887 0,938
Среднее число магистральных направлений сортировки (Q_{av})	8 3	8 4	8 2	8 7	8 9
Количество контейнеров для единовременной отправки потоков (K_{opt})	100 43	100 55	100 34	196 161	499 431
Среднее время задержки потоков (t_{av})	0,08343 0,06027	0,08343 0,06905	0,08343 0,05490	0,10005 0,11480	Infinit Infinit
Минимальный и максимальный поток по дуге (*MINF* - *MAXF*)	1 - 4 1 - 3	1 - 4 1 - 4	1 - 4 1 - 3	1 - 11 1 - 11	1 - 20 1 - 20
Полные затраты (C_{all}), (усл. ед.), в том числе:	1119405 798559	1182822 1001379	1087927 688141	1666215 1642069	2451078 2384971
транспортные затраты (C_{tr}),	1014034 710656	1014034 845146	1014034 634836	1049794 1039206	1222540 1167812
затраты на сортировку (C_{sort}),	66556 72446	129973 135017	35078 41048	537450 531891	1010541 1017240
затраты на погрузку-выгрузку (C_{load})	38815 15457	38815 21216	38815 12257	78971 70972	217997 199919
Эффект от оптимизации (*EF*)	**320846**	**181443**	**399786**	**24146**	**66107**
Число маршрутов	30 23	30 26	30 21	30 30	33 31
Средний коэффициент загрузки маршрута	0,118 0,067	0,118 0,085	0,118 0,062	0,255 0,255	0,686 0,698
Рабочий парк контейнеров	300 138	300 179	300 113	631 533	1651 1445
Рабочий парк транспортных средств	60 46	60 52	60 42	60 60	66 62
Минимальное время доставки ($t_{\min}$)	2,04219 2,04219	2,04219 2,04219	2,04219 2,04219	2,04219 2,04219	2,04219 2,04219
Максимальное время доставки ($t_{\max}$)	3,74531 8,11458	3,74531 5,38646	3,74531 9,04010	3,74531 4,91927	4,33802 4,95885
Среднее время доставки (t_{mean})	2,56401 3,47646	2,56401 3,07575	2,56401 3,78631	2,56401 2,81182	2,64959 2,72448

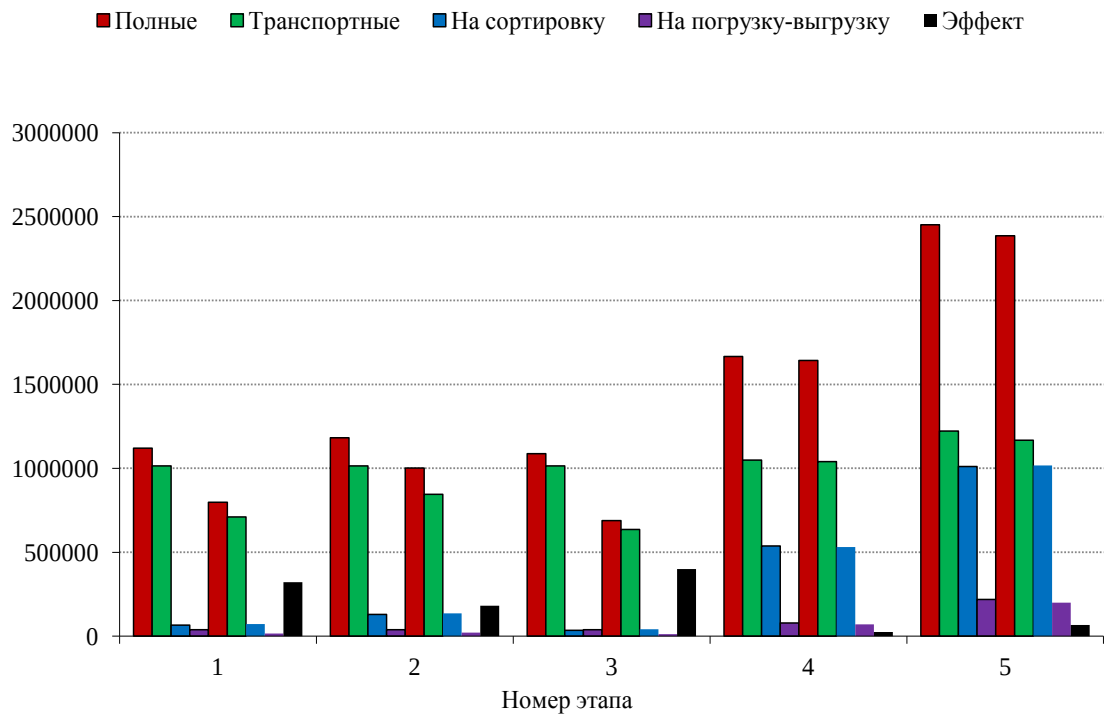


Рис. Е.2. Гистограмма распределения затрат до и после оптимизации по этапам развития сети при $W = 20$

На этапе 5 при максимальных потоках на некоторых маршрутах достигнута полная загрузка транспортных средств — $MAXF = 20$ (поэтому t_{av} = Infinit) и введен один дополнительный маршрут (см. табл. Е.4). На всех этапах среднее время доставки грузов получателю t_{mean} после оптимизации остается достаточно стабильным и изменяется в пределах от 2,72 до 3,79 суток.

В табл. Е.2 приведены суммарная величина потоков мелкопартионных грузов и затраты на их сортировку, а в табл. Е.3 — затраты на погрузку-выгрузку транспортных блоков в узлах сети до и после оптимизации по этапам развития. На рис. Е.3, Е.4 и Е.5 показаны гистограммы изменения потоков и затрат. Как видно из табл. Е.2, Е.3 и рис. Е.3, Е.4 и Е.5 затраты на сортировку и погрузку-выгрузку в узлах сети изменяются не пропорционально обрабатываемым потокам, в отдельных узлах при равенстве или увеличении потоков до и после оптимизации затраты на сортировку уменьшаются (в табл. Е.2 выделено жирным шрифтом).

Таблица Е.2

Номер узла	Потоки и затраты на их сортировку в узлах сети по этапам развития до и после оптимизации (ед., усл. ед.)				
	Этап 1 *FS* = 5 *KF* = 1	Этап 2 *FS* = 2, *KF* = 2	Этап 3 *FS* = 6, *KF* =0,5	Этап 4 *FS* = 2, *KF* = 10	Этап 5 *FS* =1 , *KF* = 30
1	282/314 4695,19 5217,06	564/586 9257,11 9599,10	146/197 2447,66 3292,14	2820/2850 41284,3 41986,7	8460/8520 93058,5 97665,2
2	407/441 6678,26 7147,73	814/776 12976,3 12234,3	214/238 3559,84 3926,46	4070/3580 51496,6 45302,6	12210/10620 86705,2 79066,9
3	352/490 5762,28 7863,23	704/844 11170,3 13174,4	184/311 3057,32 5062,89	3520/3430 43506,8 42055,4	10560/10290 69901,1 69764,8
4	345/355 5700,83 5818,56	690/702 11155,1 11203,8	179/176 2989,09 2926,47	**3450/3490** **46827,6** **46453,7**	10350/10320 90732,9 83802,0
5	342/410 5619,02 6634,59	684/808 10932,4 12654,9	177/206 2947,04 3395,46	3420/3490 43839,9 44485,4	10260/10440 75762,5 75980,8
6	336/342 5581,62 5652,33	672/642 10979,9 10425,4	175/184 2930,25 3070,29	3360/3160 48088,7 45032,2	10080/9510 103598 99186,6
7	**289/289** **4785,70** **4781,93**	**578/578** **9384,52** **9369,73**	**152/152** **2540,69** **2539,63**	2890/3030 40074,9 42516,4	8670/8730 81043,7 89466,5
8	362/386 5945,58 6310,12	724/772 11563,7 12251,6	188/211 3129,48 3503,52	3620/3660 46244,5 48154,7	10860/10920 79370,7 92334,3
9	304/315 5032,81 5180,43	608/634 9866,57 10174,2	158/164 2640,95 2731,74	3040/3070 42047,4 42051,3	9120/9210 84599,8 86786,5
10	296/307 4881,32 5000,13	**592/592** **9532,38** **9322,38**	155/167 2585,33 2761,45	**2960/3020** **39377,2** **39019,6**	**8880/8940** **73290,5** **68322,4**
11	343/336 5672,75 5523,12	686/672 11109,9 10751,0	179/190 2990,26 3157,87	3430/3390 46965,8 45013,2	10290/10080 92609,2 91892,1
12	378/453 6200,71 7316,79	756/884 12045,1 13855,8	196/285 3260,64 4680,13	3780/3950 47696,2 49820,4	11340/11460 79866,6 82968,2
Сумма	**4036/4438** **66556** **72446**	**8072/8490** **129973** **135017**	**2103/2481** **35078** **41048**	**40360/40120** **537450** **531891**	**121080/119040** **1010541** **1017240**

Таблица Е.3

Номер узла	Затраты на погрузку-выгрузку транспортных блоков в узлах сети по этапам развития до и после оптимизации (усл. ед.)				
	Этап 1 *FS* =5, *KF* = 1	Этап 2 *FS* = 2, *KF* = 2	Этап 3 *FS* = 6, *KF* = 0,5	Этап 4 *FS* =2, *KF* = 10	Этап 5 *FS* = 1, *KF* = 30
1	3207,89 1608,04	3207,89 1928,01	3207,89 968,09	6567,57 6087,62	20166,2 17286,5
2	3527,86 1768,03	3527,86 2088,00	3527,86 1288,07	10087,2 7527,48	29125,4 19206,3
3	4167,80 2247,98	4167,80 3527,86	4167,80 1928,01	9127,32 8327,40	23845,9 23365,9
4	3527,86 1288,07	3527,86 1768,03	3527,86 968,09	6087,62 5767,65	15366,7 18566,4
5	3207,89 1288,07	3207,89 2247,98	3207,89 968,09	6247,60 5447,68	15046,7 16006,7
6	3527,86 968,09	3527,86 2247,98	3527,86 968,09	6247,60 4647,76	18886,4 14406,8
7	2887,92 1128,08	2887,92 1448,05	2887,92 968,09	5767,65 5767,65	17926,5 15686,7
8	1608,04 968,09	1608,04 1288,07	1608,04 1288,07	4167,80 3847,83	10247,2 12487,0
9	2887,92 968,09	2887,92 1288,07	2887,92 648,10	5927,63 5127,71	18086,4 15526,7
10	3207,89 648,10	3207,89 648,10	3207,89 648,10	4807,74 4647,76	13126,9 13126,9
11	3207,89 968,09	3207,89 1448,05	3207,89 648,10	7367,49 6567,57	18246,4 17926,5
12	3847,83 1608,04	3847,83 2247,98	3847,83 1288,07	6567,57 7207,51	17926,5 16326,6
Сумма	**38815** **15457**	**38815** **21216**	**38815** **12257**	**78971** **70972**	**217997** **199919**

На третьем этапе при уменьшении потоков нагрузка на сортировочную и погрузочно-разгрузочную работу в узлах сети снижается, и эти узлы должны переходить на более низкий уровень технического оснащения.

В табл. Е.4 для каждого маршрута приведены затраты и максимальная загрузка транспортных средств в транспортных блоках при движении по маршруту в прямом и обратном направлении для грузоподъемности $W = 20$ и $W = 5$. При использовании транспортных средств с грузоподъемностью $W = 20$ на всех маршрутах сеть выдерживает нагрузку на всех этапах развития. При этом

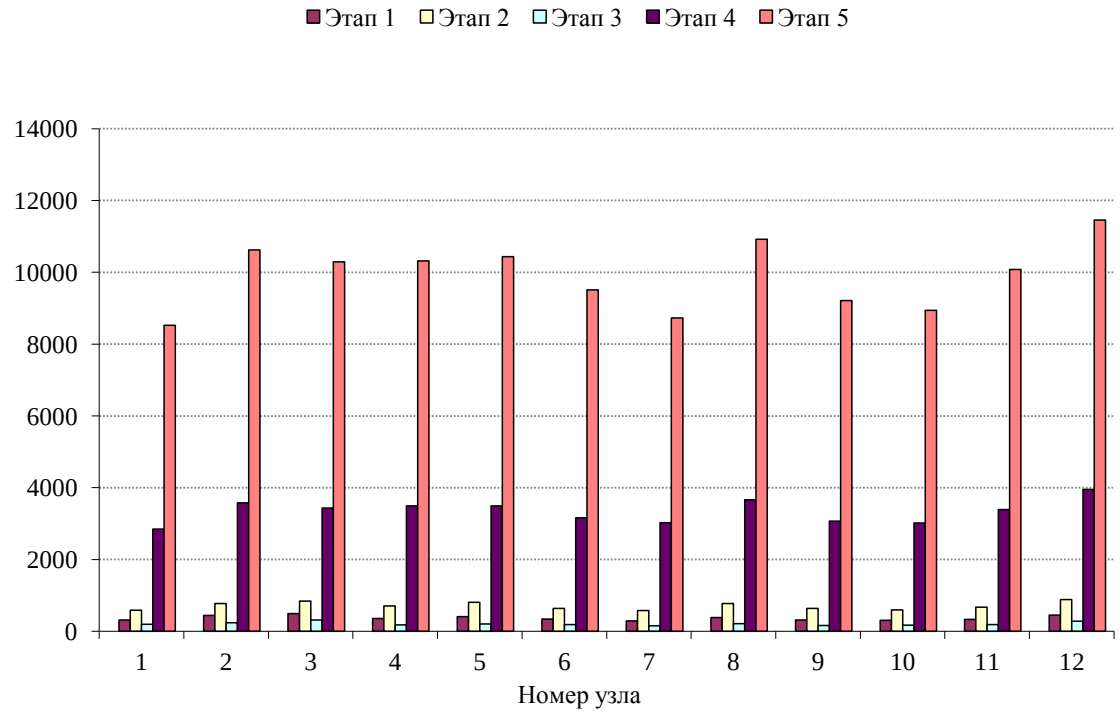


Рис. Е.3. Величина потоков в узлах сети после оптимизации по этапам развития

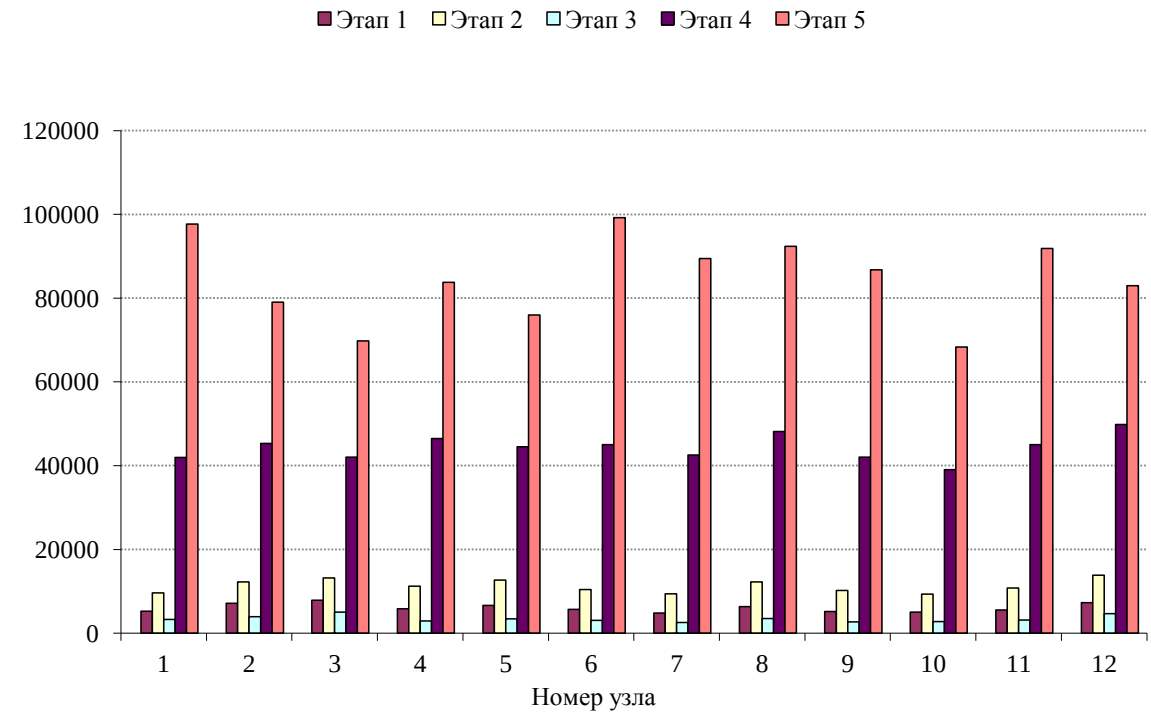


Рис. Е.4. Гистограмма распределения затрат на сортировку потоков в узлах сети после оптимизации по этапам развития

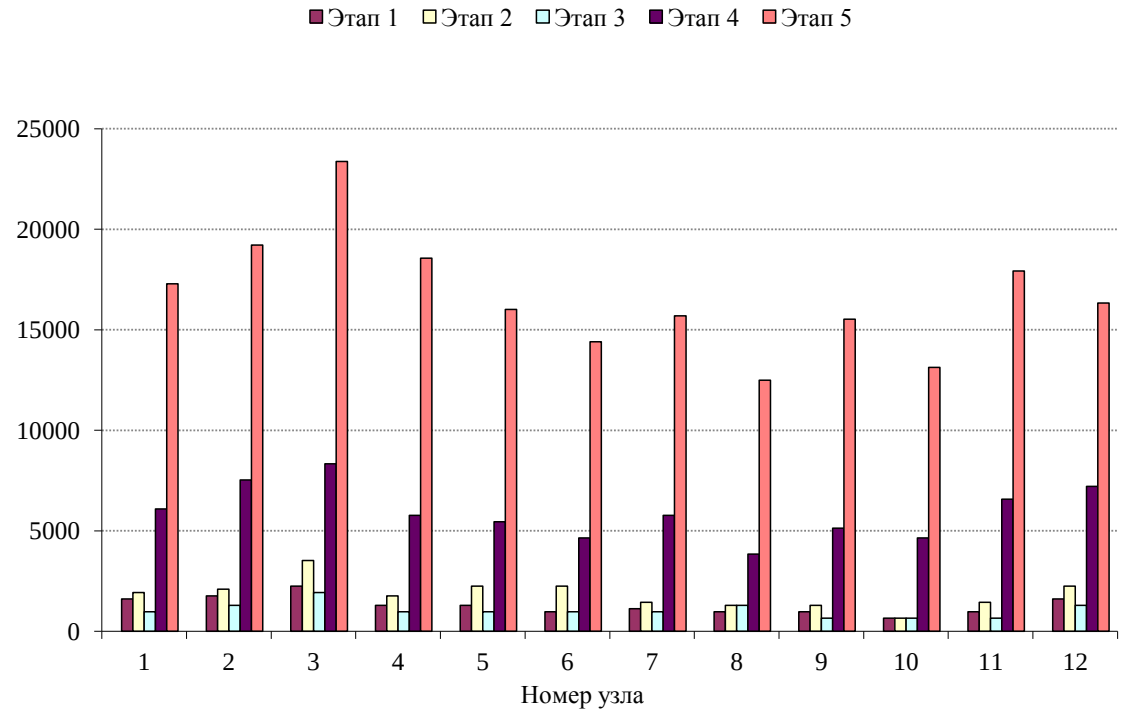


Рис. Е.5. Гистограмма распределения затрат на погрузку-выгрузку транспортных блоков в узлах сети после оптимизации по этапам развития

на первых четырех этапах загрузка всех маршрутов низкая, на отдельных этапах более длинные маршруты с номерами 3, 4, 5, 14, 18, 23, 24, 25, 28 вообще оказались не загруженными. На пятом этапе для распределения всех потоков до оптимизации структуры сети введено три дополнительных маршрута (номера 31, 33 и 34), а после оптимизации — один (номер 37). При $W = 5$ на первых трех этапах не загруженными оказались те же маршруты, что и при $W = 20$. На четвертом этапе до оптимизации введено десять дополнительных маршрутов (номера 31-40), после оптимизации — семь (номера 31, 33, 34, 37, 40-42). На пятом этапе введено до и после оптимизации соответственно тридцать три дополнительных маршрута с минимальной грузоподъемностью $W = 14$ и тридцать шесть маршрутов с минимальной грузоподъемностью $W = 12$.

После оптимизации не загруженным оказался маршрут с номером 4. Грузоподъемность на дополнительных маршрутах была увеличена из-за запрета разветвления потоков при наличии потоков величиной в 14 и 12 транспортных блоков.

Таблица Е.4

Прочерк «-» означает, что маршруты не вводились, а «*», что маршруты вводились, но не перечисляются

Номер, маршрут, длина (км)	Затраты (усл. ед.) при *W* =20/5	Загрузка маршрутов по этапам развития до/после оптимизации				
		Этап 1 *FS* =5, *KF* =1	Этап 2 *FS* =2, *KF* =2	Этап 3 *FS* =6, *KF* =0,5	Этап 4 *FS* =2 , *KF* =10	Этап 5 *FS* =1, *KF* =30
1 (1,2) 81	19798,8 4949,7	4/3 4/3	4/2 4/2	4/1 4/1	11/10 5/5	18/20 4/4
2 (1, 3) 99	21901,2 5475,3	3/2 3/2	3/4 3/4	3/2 3/2	8/7 5/5	15/18 5/3
3 (1,4) 181	31478,8 7869,7	2/1 2/1	2/1 2/1	2/0 2/0	3/4 3/5	18/17 4/4
4 (1,5) 268	41640,4 10410,1	3/ 1 3/1	3/1 3/1	3/0 3/0	5/2 5/5	10/9 5/0
5 (1,6) 291	44326,8 11081,7	3/0 3/0	3/1 3/1	3/0 3/0	5/4 5/5	13/17 5/5
6 (2,7) 122	24587,6 6146,9	2/2 2/2	2/2 2/2	2/1 2/1	7/7 4/5	20/14 4/4
7 (2,8) 155	28442,0 7110,5	1/1 1/1	1/2 1/2	1/1 1/1	6/8 5/5	16/17 1/4
8 (2,9) 120	24354,0 6088,5	2/1 2/1	2/2 2/2	2/1 2/1	6/5 5/5	16/15 3/3
9 (2,10) 138	26456,4 6614,1	2/1 2/1	2/1 2/1	2/1 2/1	5/5 5/5	19/19 4/5
10 (3,4) 169	30077,2 7519,3	4/1 4/1	4/3 4/3	4/1 4/1	10/11 5/5	19/19 5/3
11 (3,5) 175	30778,0 7694,5	4/2 4/2	4/4 4/4	4/1 4/1	7/6 5/5	17/20 5/4
12 (3,11) 224	36501,2 9125,3	2/1 2/1	2/1 2/1	2/2 2/2	7/5 5/5	13/19 5/3
13 (3,12) 100	22018,0 5504,5	4/3 4/3	4/3 4/3	4/3 4/3	7/9 5/5	19/19 3/5
14 (4,6) 243	38720,4 9680,1	1/0 1/0	1/0 1/0	1/0 1/0	1/1 5/1	1/4 3/5
15 (4,7) 178	31128,4 7782,1	3/1 3/1	3/2 3/2	3/1 3/1	4/4 5/5	11/15 5/4
16 (4,8) 154	28325,2 7081,3	1/1 1/1	1/1 1/1	1/1 1/1	2/2 2/3	5/7 5/3
17 (5,9) 263	41056,4 10264,1	2/1 2/1	2/1 2/1	2/1 2/1	5/4 5/5	16/13 4/4
18 (5,10) 299	45261,2 11315,3	1/0 1/0	1/1 1/1	1/0 1/0	1/3 3/5	3/9 4/4
19 (5,11) 218	35800,4 8950,1	1/1 1/1	1/2 1/2	1/1 1/1	4/4 4/4	10/10 5/5

Продолжение таблицы Е.4

20 (6,7) 90	20850,0 5212,5	4/2 4/2	4/2 4/2	4/1 4/1	9/9 5/5	18/19 5/5
21 (6,8) 128	25288,4 6322,1	1/1 1/1	1/1 1/1	1/2 1/2	2/4 5/3	6/7 4/5
22 (6,12) 290	44210,0 11052,5	2/1 2/1	2/1 2/1	2/1 2/1	5/4 5/5	20/11 5/5
23 (7,9) 270	41874,0 10468,5	2/0 2/0	2/0 2/0	2/0 2/0	3/3 5/5	17/12 3/5
24 (7,10) 196	33230,8 8307,7	3/0 3/0	3/0 3/0	3/0 3/0	4/6 5/5	13/10 5/4
25 (8,11) 271	41990,8 10497,7	1/0 1/0	1/0 1/0	1/0 1/0	2/2 2/2	5/14 5/4
26 (8,12) 289	44093,2 11023,3	1/1 1/1	1/1 1/1	1/1 1/1	2/2 2/5	4/12 4/5
27 (9,10) 117	24003,6 6000,9	1/1 1/1	1/1 1/1	1/1 1/1	3/3 5/3	7/8 3/5
28 (9,11) 258	40472,4 10118,1	3/0 3/0	3/1 3/1	3/0 3/0	6/6 5/5	20/14 4/5
29 (10,12) 273	42224,4 10556,1	4/1 4/1	4/1 4/1	4/1 4/1	5/4 5/5	15/10 5/4
30 (11,12) 89	20733,2 5183,3	4/1 4/1	4/2 4/2	4/1 4/1	8/9 5/5	17/19 4/5
31 (1,2) 81	19798,8 4949,7	- -	- -	- -	- 5/1	14/- *
32 (2,7) 122	24587,6 6146,9	- -	- -	- -	- 5/-	- *
33 (6,7) 90	20850,0 5212,5	- -	- -	- -	- 5/5	18/- *
34 (1,3) 99	21901,2 5475,3	- -	- -	- -	- 5/4	20/- *
35 (2,9) 120	24354,0 6088,5	- -	- -	- -	- 5/-	- *
36 (3,11) 224	36501,2 9125,3	- -	- -	- -	- 4/-	- *
37 (3,4) 169	30077,2 7519,3	- -	- -	- -	- 5/5	-/16 *
38 (6,12) 290	44210,0 11052,5	- -	- -	- -	- 3/-	- *
39 (9,11) 258	40472,4 10118,1	- -	- -	- -	- 2/-	- *
40 (11,12) 89	20733,2 5183,3	- -	- -	- -	- 2/5	- *
41 (2,8) 155	28442,0 7110,5	- -	- -	- -	- -/4	- *
42 (3,12) 100	22018,0 5504,5	- -	- -	- -	- -/3	- *
Сумма затрат до и после оптимизации	при W =20 при W =5	**1014034 710656 277816,6 188845**	**1014034 845146 277816,6 225790**	**1014034 634836 277816,6 167865**	**1049794 1039206 385852,4 345132,5**	**1222540 1167812 1117435 1028619**

Из табл. Е.4 видно, что выбор транспортных средств с грузоподъемностью $W = 5$ на всех маршрутах по сравнению с $W = 20$ позволяет значительно уменьшить транспортные затраты на всех этапах развития сети как до, так и после оптимизации (см. последнюю строку таблицы). В эксперименте при решении задачи распределения и маршрутизации потоков не ставилась цель нахождения наилучшего решения при возможности генерации множества шаблонов маршрутов (состоящих из нескольких дуг) и разветвления потоков. Если допустить такие возможности, то транспортные затраты и затраты на погрузку-выгрузку транспортных блоков могут быть уменьшены за счет более полной загрузки транспортных средств и сокращения транзитных перегрузок. Однако следует учитывать, что полная загрузка транспортных средств на большинстве маршрутов при колебании потоков на определенных промежутках времени приведет к необходимости частого решения задачи маршрутизации и может создать серьезные трудности диспетчерам при практическом управлении перевозками и ухудшить качество транспортных сообщений. Поэтому на отдельных этапах планирования при решении задачи маршрутизации можно задавать ограничение на среднее время задержки потоков $T_{\max}$ и тем самым резервировать дополнительную грузоподъемность транспортных средств.

В табл. Е.5 и на рис. Е.6 показано изменение затрат после оптимизации структуры сети, распределения и маршрутизации потоков для $W = 20$ и $W = 5$ на этапах развития сети. Из табл. Е.5 видно, что на отдельных этапах экономия полных затрат составляет от 129594 до 688954 усл. ед., что соответствует 5,43% и 41,96%.

С целью исследования изменения транспортных, погрузочно-разгрузочных затрат и времени доставки мелкопартионных грузов получателям была решена задача маршрутизации для четвертого этапа при $W = 5$ и генерации шаблонов маршрутов по кратчайшим путям между всеми парами узлов сети по критерию — минимум дуг, минимум длины. Всего было сгенерировано $(n^2 - n)/2$=(144-12)/2= 66 маршрутов, из которых только 36 оказались загруженными.

Таблица Е.5

Наименование затрат	Величина затрат после оптимизации для W = 20 и W = 5 (усл. ед.)				
	Этап 1 FS =5, KF =1	Этап 2 FS =2, KF =2	Этап 3 FS =6, KF =0,5	Этап 4 FS =2 KF =10	Этап 5 FS =1, KF =30
Полные затраты (C_{all}), в том числе:	798559 276748	1001379 382023	688141 221170	1642069 953115	2384971 2255377
транспортные (C_{tr}),	710656 188845	845146 225790	634836 167865	1039206 345133	1167812 1028619
на сортировку (C_{sort}),	72446 72446	135017 135017	41048 41048	531891 531891	1017240 1017240
на погрузку-выгрузку (C_{load})	15457 15457	21216 21216	12257 12257	70972 76091	199919 209518
Эффект от оптимизации (EF)	**521811**	**619356**	**466971**	**688954**	**129594**

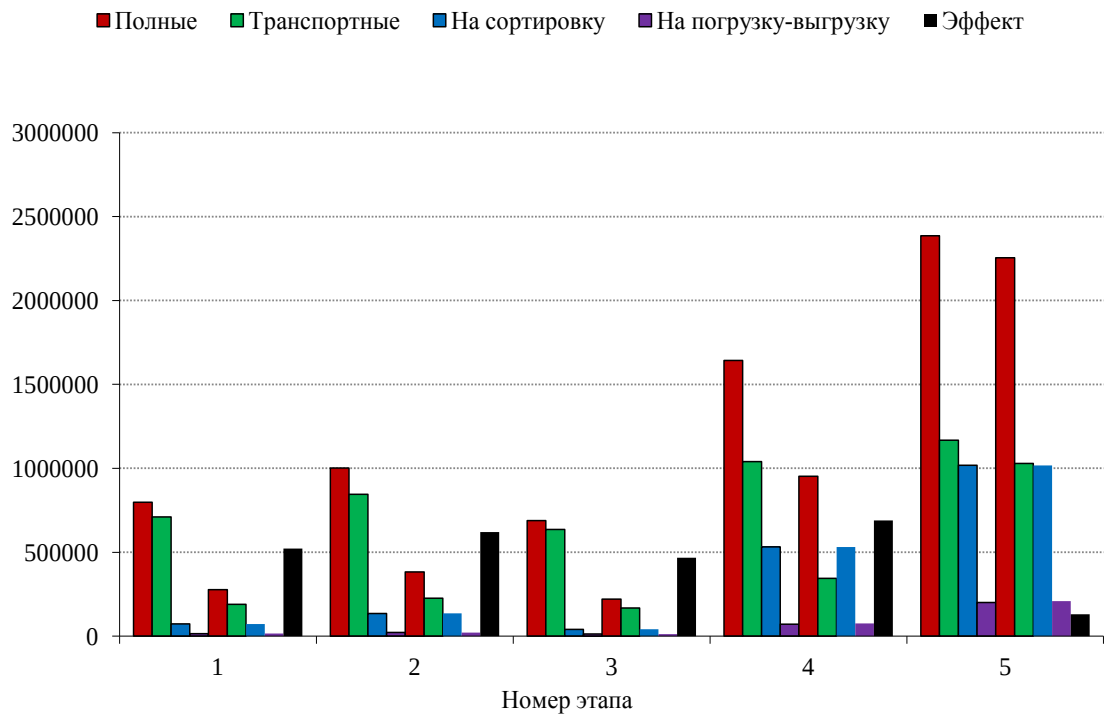


Рис. Е.6. Гистограмма затрат после оптимизации для W = 20 и W = 5

Как и следовало ожидать (так как транспортные затраты на несколько порядков больше, чем затраты на погрузку-выгрузку), значение целевой функции увеличилось с 953115 до 1074922 усл. ед. при возрастании транспортных затрат с 345133 до 486688 усл. ед. и сокращении затрат на погрузку-выгрузку с 76091

до 56413 усл. ед. Затраты на сортировку величиной 531891 усл. ед. остались без изменения. Минимальное время доставки в сутках $t_{\min} = 2{,}04$ не изменилось, а максимальное $t_{\max}$ и среднее t_{mean} — уменьшилось соответственно с 4,92 до 4,42 и с 2,81 до 2,48. Поскольку в каждом сгенерированном маршруте для данной сети не может быть более трех дуг и все потоки транспортных блоков транспортируются без перегрузок, то объяснимо сокращение $t_{\max}$ на 0,5 суток (это заданное время на транзитную перегрузку).

На основании полученных результатов можно сделать следующий вывод: при использовании шаблонов маршрутов для всех корреспондирующихся пар узлов только по кратчайшим путям сокращается максимальное и среднее время доставки, а уменьшение полных затрат возможно только при максимальном снижении грузоподъемности транспортных средств и только на сетях достаточно большой размерности со значительными транзитными потоками транспортных блоков в узлах сети.

Полученные данные в табл. Е.1-Е.5 можно использовать для предварительного пропорционального распределения выделенных капитальных вложений между этапами развития и внутри этапов между узлами сети и маршрутами транспортных средств.

Во второй части эксперимента была исследована возможность использования решения задач оптимизации упаковки, распределения и маршрутизации потоков для оперативного перераспределения потоков в случае вынужденного ограничения пропускной способности узлов, при отказе сортировочного оборудования в отдельных узлах сети и повреждении (ремонте) линий связи (участков дорог). Для большей наглядности результатов эксперимента была выбрана сеть с числом узлов $n = 300$, числом исходящих дуг из каждого узла $val = 3$. Изначально для сети было сгенерировано 107 узлов второго типа и $n*val/2$=300*3/2=450 маршрутов. Объем транспортного блока ω остался прежним, равным 40 единицам грузов, а общая грузоподъемность транспортных средств на дугах сети и максимальное время доставки мелкопартионных грузов получателю при выборе структуры сети были соответственно увеличены

до $W_0 = 6600$ транспортных блоков и $T_{ij} = 15,\ \forall ij \in S$ суток. После оптимизации структуры сети в нее было включено $n_1 = 175$ узлов первого типа и $n_2 = 125$ узлов второго типа. Первоначально пропускные способности узлов по обработке транзита $h_i,\ i = \overline{1,n}$ принимались равными 5000 единиц мелкопартионных грузов в сутки для всех узлов сети. В результате решения задачи было определено, что в 24 узлах сети объемы транзита превышают 3000 единиц — условную границу вынужденного диспетчерского ограничения. Поэтому задача оптимизации структуры сети была решена для ограничений на пропускные способности $h_i = 3000,\ i = \overline{1,n}$ и потоки в сети были перераспределены без нарушения вновь введенных ограничений. В отличие от возможности установки диспетчерского ограничения на допустимый объем транзитных потоков мелкопартионных грузов, установка ограничения на величину транзитных потоков транспортных блоков $b_i,\ i = \overline{1,n}$ в узлах сети может привести к невозможности распределения всех потоков при решении задачи маршрутизации. Такая ситуация возникла при проведении эксперимента, когда были введены ограничения $b_i = 5000,\ i = \overline{1,n}$ и 87 потоков оказались нераспределенными из-за нарушения ограничений в трех узлах. Нарушение ограничений произошло из-за того, что все маршруты транспортных средств совпадали с дугами сети, а генерация шаблонов маршрутов и разветвление потоков были запрещены. В этом случае просто не существовало путей, по которым все потоки могли бы быть распределены без нарушения ограничений в некоторых узлах, т.е. задача маршрутизации при таких ограничениях не имела решения. Поэтому при решении задачи маршрутизации ограничения на пропускные способности узлов $b_i,\ i = \overline{1,n}$ жестко не задавались, что позволяло распределить все потоки и получить решение. Основные результаты решения задач до и после оптимизации структуры сети и маршрутизации потоков при $h_i = 5000,\ h_i = 3000,\ i = \overline{1,n}$ и $n_2 = 125$ приведены во втором и третьем столбцах табл. Е.6. Из таблицы видно, что оптимальные решения в обоих случаях были достигнуты при $FS = 2$, при перераспределе-

Таблица Е.6

Наименование показателей	Значение показателей до оптимизации и после оптимизации при W =40		
	h_i =5000, n_2 =125, FS =2	h_i =3000, n_2 =125, FS =2	h_i =5000, n_2 =125, n_3 =3, FS =2
K_{av}	0,446 0,684	0,446 0,656	0,430 0,680
Q_{av}	125 33	125 38	102 33
K_{opt}	42158 22666	42158 23123	37368 22678
t_{av}	Infinit 0,00085 (при W_0 = 6600)	Infinit 0,00093(при W_0 = 6600)	0,00428 0,00085(при W_0 = 6600)
MINF - MAXF	11 - 7676 4 - 2351	11 - 7676 4 - 2650	10 - 6437 4 - 2356
Полные затраты (C_{all}), (млн.усл. ед.), в том числе:	451,285 176,310	451,285 189,290	393,338 177,930
транспортные затраты (C_{tr}),	363,616 133,649	363,616 144,503	316,099 135,532
затраты на сортировку (C_{sort}),	14,970 16,828	14,970 16,742	15,464 16,844
затраты на погрузку-выгрузку (C_{load})	72,699 25,833	72,699 28,045	61,775 25,554
Эффект от оптимизации (EF)	**274,975**	**261,995**	**215,408**
Число маршрутов	5470 2010	5470 2175	4762 2045
Средний коэффициент загрузки маршрута	0,970 0,912	0,970 0,927	0,952 0,907
Рабочий парк контейнеров	329077 129321	329077 138142	281884 128456
Рабочий парк транспортных средств	10934 4018	10934 4346	9519 4088
$t_{\min}$	2,04 2,04	2,04 2,04	2,04 2,04
$t_{\max}$	(11,91) 35,88 (12,09) 43,01	(11,91) 35,88 (12,09) 37,08	(12,09) 32,38 (12,09) 34,85
t_{mean}	(7,58) 8,48 (7,86) 9,46	(7,58) 8,48 (7,83) 9,38	(7,66) 8,66 (7,87) 9,30

нии потоков несколько уменьшился средний коэффициент загрузки транспортного блока K_{av}, увеличилось среднее число магистральных направлений сорти-

ровки Q_{av} и незначительно снизился эффект от оптимизации EF. При решении задачи выбора структуры сети была сильно завышена общая грузоподъемность транспортных средств на дугах сети (до $W_0 = 6600$), поэтому в процессе решения программой было предварительно определено, что общую грузоподъемность можно уменьшить до значений максимальных потоков по дуге, соответственно до 2351 и 2650.

При решении задачи маршрутизации на всех маршрутах была выбрана грузоподъемность транспортных средств $W = 40$, а максимальное время доставки грузов получателю T_{ij}, $\forall ij \in S$ не ограничивалось с целью максимального увеличения загрузки транспортных средств. Поэтому для распределения всех потоков транспортных блоков были введены дополнительные маршруты, и общее количество загруженных маршрутов соответственно составило 2010 и 2175. Максимальная загрузка транспортных средств (и как следствие уменьшение количества загруженных маршрутов) привела к распределению потоков по более «длинным» путям, что значительно увеличило максимальное время доставки $t_{\max}$ для отдельных потоков грузов с многочисленными транзитными перегрузками. Среднее время доставки изменилось незначительно.

В табл. Е.6 для $t_{\max}$ и t_{mean} приведены значения, полученные после решения задачи оптимизации структуры сети при $T_{ij} = 15$, $\forall ij \in S$ (в скобках) и после решения задачи маршрутизации при $T_{ij} = \infty$, $\forall ij \in S$.

Далее предполагалось, что в сети произошли отказы трех узлов с номерами 243, 277, 287 и четырех линий связи (243, 245), (241, 243), (277, 279), (287, 289). В результате отказов узлы получили статус узлов третьего типа, и перешли в аварийный режим работы, а соответствующие линиям связи маршруты транспортных средств удалены. После решения задачи потоки в сети были полностью перераспределены, и нагрузка на выведенные из строя узлы была сведена к минимуму, так как исходящий поток из внутренней зоны этих узлов нужно сортировать только по пяти и восьми магистральным направлениям. Транзитные потоки мелкопартионных грузов и транспортных блоков (указаны в скобках) в

аварийных узлах отсутствуют. Входящий поток может быть отсортирован на абонентские узлы вышедших из строя узлов в смежных узлах первого типа.

В табл. Е.6 (четвертый столбец) и табл. Е.7 приведены результаты решения задачи при перераспределении потоков.

Таблица Е.7

Наименование показателей и параметров	Значение показателей и параметров в узлах					
	Узел 243		Узел 277		Узел 287	
	исправен	аварийный режим	исправен	аварийный режим	исправен	аварийный режим
Общий поток	3102	2737	3450	2795	3355	2793
Внутриузловой поток	241	241	299	299	296	296
Исходящий поток	1264	1264	1272	1272	1270	1270
Входящий поток	1232	1232	1224	1224	1227	1227
Транзитный поток	365 (90)	0 (0)	655 (87)	0 (0)	562 (140)	0 (0)
Ограничения на пропускную способность узлов - h_i, (b_i)	5000 (∞)	0 (0)	5000 (∞)	0 (0)	5000 (∞)	0 (0)
Коэффициент загрузки транспортного блока	0,630	0,656	0,673	0,343	0,721	0,542
Число магистральных направлений сортировки	38	5	43	8	43	8
Число направлений сортировки на абонентские узлы	27	27	21	21	13	13
Общее число направлений сортировки	65	32	64	29	56	21
Затраты на сортировку по-токов (усл. ед.)	48199,5	39901,3	53032,0	40014,2	51045,4	37659,3
Затраты на погрузку-выгрузку транспортных блоков (усл. ед.)	32965,0	10887,2	35684,7	12807,0	42564,1	12487,0

Оптимальное решение найдено при $FS = 2$. Значения результирующих показателей в сети после оптимизации почти не изменились, а уменьшение коэффициента загрузки транспортного блока в узлах 277 и 287 произошло ввиду то-

го, что во внутреннюю зону этих узлов попали соответственно пять и три магистральных узла второго типа, объемы сортировки потоков на которые незначительны.

УТВЕРЖДАЮ
Генеральный директор
ООО «АС-Транс Брок»
_____ С.В. Шматов
20.03.17

СПРАВКА

об использовании результатов диссертационной работы В.А. Васянина «Методология проектирования многопродуктовых коммуникационных сетей с дискретными потоками»

Разработанное Васяниным В.А. программное обеспечение для решения задач оптимизации иерархической структуры сети, схемы сортировки и маршрутизации дискретных потоков было апробировано ООО «АС-Транс Брок» в период 2015-2016 гг. в инструментальном программном комплексе «Альфа» при решении задач планирования и проектирования системы транспортных перевозок на территории Украины по автотранспортным магистралям. При этом решение оптимизационных задач было основано на методологии математического моделирования проектирования и функционирования многопродуктовых сетей с дискретными потоками мелкопартионных грузов.

Проведенные вычислительные эксперименты и полученные результаты на модели реальной сети автотранспортных магистралей Украины показали эффективность предложенных Васяниным В.А. подходов, методов и алгоритмов для решения задач текущего планирования магистральных перевозок и управления ресурсами ООО «АС-Транс Брок» и позволили улучшить существующие схемы обработки и транспортировки грузов при среднем сокращении годовых приведенных затрат на 5,2%.

Начальник отдела
транспортных перевозок Л.Н. Кармазина

АКТ

о внедрении результатов диссертационной работы В.А. Васянина «Методология проектирования многопродуктовых коммуникационных сетей с дискретными потоками»

Методическое, алгоритмическое и программное обеспечение, разработанное в рамках диссертационной работы Васянина В.А. было использовано ФЛ-П Заяц Юрий Васильевич в информационной системе поддержки принятия решений предприятия для 1) оптимизации структуры сети перевозок мелкопартионных грузов на поддонах (палеттах) автомобильными фургонами большой грузоподъемности (3, 5 и 10 т) на транспортных магистралях Украины и Европы (Польша, Германия, Австрия, Венгрия, Словения, Румыния, Чехия и др.); 2) схемы сортировки грузов в сортировочных центрах предприятия для магистральных и внутренних транспортных сообщений; 3) схемы перевозки грузов с построением доставочных, сборочных и комбинированных маршрутов транспортных средств с возможностью транзитного обмена во всех промежуточных пунктах следования транспортных средств.

В результате математического моделирования и оптимизации сети перевозок и схем обработки и транспортировки грузов предприятию удалось уменьшить фактические транспортные издержки на реальной транспортной сети за период с 01.12.2016 по 28.02.2017 г. на 8,27%. Разработанный Васяниным В.А. комплекс программ для моделирования и проектирования иерархических магистральных сетей перевозки мелкопартионных грузов принят к внедрению в информационной системе поддержки принятия решений предприятия.

Ю.В. Заяц

14.03.2017

УТВЕРЖДАЮ
Физическое лицо-
предприниматель
Н.И. Диптан
23.03.2017

СПРАВКА

об использовании результатов диссертационной работы В.А. Васянина
«Методология проектирования многопродуктовых коммуникационных сетей
с дискретными потоками»

В период с 2015 по 2016 гг. разработанное в диссертационной работе Васянина В.А. программное обеспечение для проектирования и исследования функционирования многопродуктовых коммуникационных сетей с дискретными потоками корреспонденций было использовано ФЛ-П Диптан Н.И. в системе планирования и развития сети перевозок на внутренних и международных транспортных автомагистралях. Программное обеспечение было апробировано как для текущего планирования процессов обработки и транспортировки мелкопартионных грузов, так и для решения задач перспективного развития узлов и магистральных маршрутов предприятия.

Результаты моделирования решения задач текущего планирования и перспективного развития на реальной транспортной сети предприятия показали адекватность предложенных Васяниным В.А. математических моделей процессов обработки и транспортировки мелкопартионных грузов, вычислительную эффективность разработанных алгоритмов и программ. Данные, полученные в результате проведенных экспериментов, подтвердили близкие к фактическим оценки технико-экономических показателей функционирования сети, позволили выявить «узкие места» в сети перевозок, дать рекомендации по их устранению и, как следствие, повысить эффективность и устойчивость работы транспортной сети.

Предприятие ФЛ-П Диптан Н.И. в дальнейшем планирует внедрение основных результатов диссертационной работы Васянина В.А. — комплекса задач текущего планирования и инструментального программного обеспечения для их решения в свою производственную деятельность.

Н.И. Диптан

Republica Moldova
Academia de Transporturi, Informatică şi Comunicaţii
ATIC

or. Chişinău, str. Munceşti 121/A, tel./fax: 022 38-45-30, tel.: 022 38-45-40, e-mail: atic@mtc.md

Nr. 112
17. 05. 2017

Институт телекоммуникаций и глобального информационного пространства НАН Украины

СПРАВКА – ПОДТВЕРЖДЕНИЕ

об использовании результатов диссертационной работы В.А. Васянина «Методология проектирования многопродуктовых коммуникационных сетей с дискретными потоками» в учебном процессе факультета «Инженерии и транспорта» Академии транспорта, информатики и коммуникаций Республики Молдова

Настоящим подтверждаем, что в течение последних шести лет Васянин Владимир Александрович регулярно принимал участие в работе проводимых в АТИК международных конференций «Математическое моделирование, оптимизация и информационные технологии» и «Системы транспорта и логистики». Он неоднократно выступал с докладами по различным вопросам проектирования многопродуктовых коммуникационных сетей.

Материалы диссертационной работы В.А. Васянина «Методология проектирования многопродуктовых коммуникационных сетей с дискретными потоками» используются в учебном процессе факультета «Инженерии и транспорта» АТИК. Полученные им результаты используются при формировании учебных программ по дисциплинам «Моделирование, оптимизация и планирование на автомобильном транспорте» и «Транспортная логистика». Они также используются в практических занятиях и при подготовке методических материалов для курсового и дипломного проектирования по указанным дисциплинам.

Ректор АТИК,
доктор технических наук,
профессор **Д.И. Соломон**

ЗАТВЕРДЖУЮ

Ректор ЖДТУ,
д.е.н., професор

В.В. Євдокимов

2017 р.

АКТ

про використання результатів дисертаційної роботи В.О. Васяніна «Методологія проектування багатопродуктових комунікаційних мереж з дискретними потоками» в навчальному процесі факультету інформаційно-комп'ютерних технологій Житомирського державного технологічного університету

Матеріали дисертаційної роботи Васяніна В.О. «Методологія проектування багато продуктових комунікаційних мереж з дискретними потоками» використовуються в навчальному процесі факультету ІКТ ЖДТУ, починаючи з 2015 р. при формуванні навчальних програм з дисциплін «Комп'ютерна дискретна математика», «Теорія складності екстремальних задач» і викладанні курсу «Математичні методи дослідження операцій», а також в практичних заняттях і при підготовці методичних матеріалів для курсового та дипломного проектування із зазначених дисциплін.

Декан факультету
інформаційно-комп'ютерних
технологій ЖДТУ,
к.т.н., доцент А.В. Морозов

Зав. кафедрою
програмного забезпечення систем
факультету
інформаційно-комп'ютерних
технологій ЖДТУ,
д.т.н., професор А.В. Панішев

УКРАЇНА
Міністерство освіти і науки України
Державний департамент інтелектуальної власності

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 5978

Комп'ютерна програма "Пакет програм для рішення задач оптимізації розподілу дискретних потоків у багатопродуктових мережах"
(вид, назва твору)

Автор(и) Васянін Володимир Олександрович
(повне ім'я, псевдонім (за наявності))

Дата реєстрації "23" липня 2002 р.

Голова Державного департаменту інтелектуальної власності

М.П.

М.В.Паладій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ДЕПАРТАМЕНТ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ

Україна, МСП 04655, Київ, Львівська площа, 8
Тел.: (044) 212-50-82 Факс: (044) 212-34-49

РІШЕННЯ

ПРО РЕЄСТРАЦІЮ АВТОРСЬКОГО ПРАВА НА ТВІР

Державний департамент інтелектуальної власності розглянув заяву
Васянін Володимир Олександрович, м.Київ, вул.Салютна, 4а, кв.39
(повне ім'я автора, спадкоємця, роботодавця, адреса)

заявка від "06" червня 2002 року № 5772

про реєстрацію авторського права на твір і прийняв рішення зареєструвати авторське право на твір Комп'ютерна програма "Пакет програм для рішення задач оптимізації розподілу дискретних потоків у багатопродуктових мережах";
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора (ів), роботодавець (у разі реєстрації службового твору)
Васянін Володимир Олександрович

Внесення відомостей до Державного реєстру свідоцтв про реєстрацію авторського права на твір та видача свідоцтва будуть здійснені за умови сплати збору за оформлення і видачу свідоцтва про реєстрацію авторського права на твір відповідно до п.3 постанови Кабінету Міністрів України від 27 грудня 2001 року № 1756 "Про державну реєстрацію авторського права і договорів, які стосуються права автора на твір".

Якщо протягом трьох місяців від дати одержання заявником рішення про реєстрацію авторського права на твір Державний департамент не одержав документ про сплату збору за оформлення і видачу свідоцтва у розмірі та порядку, визначених законодавством, або копію документа, що підтверджує право на звільнення від сплати зазначеного збору, заявка вважається відхиленою і реєстрація авторського права та публікація відомостей про реєстрацію Державним департаментом не проводиться.

Голова Державного департаменту Інтелектуальної власності **М.В.Паладій**

УКРАЇНА

ДЕРЖАВНА СЛУЖБА ВЛАСНОСТІ УКРАЇНИ ІНТЕЛЕКТУАЛЬНОЇ

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 66791

Компьютерна програма "Компьютерная прорамма оптимизации иерархической структуры многопродуктовой коммуникационной сети с дискретными потоками"

(вид, назва твору)

Автор(и) Трофимчук Олександр Миколайович, Васянін Володимир Олександрович, Ушакова Людмила Павлівна

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 20.07.2016

Державна служба інтелектуальної власності України Ідентифікаційний код 37552556 М.П.

Голова Державної служби інтелектуальної власності України

В.о. Голови А.А. Малиш

ДЕРЖАВНА СЛУЖБА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ УКРАЇНИ
Україна, 03680, МСП, м. Київ-35,
вул. Урицького, 45
Тел. (044) 494-06-06
Факс (044) 494-06-67
E-mail: post@sips.gov.ua

STATE INTELLECTUAL PROPERTY SERVICE OF UKRAINE
Ukraine, 03680, MSP, Kyiv-35,
45, Urytskogo str.
Tel. (044) 494-06-06
Fax (044) 494-06-67
E-mail: post@sips.gov.ua

РІШЕННЯ

ПРО РЕЄСТРАЦІЮ АВТОРСЬКОГО ПРАВА НА ТВІР

Державна служба інтелектуальної власності розглянула заяву

Трофимчук Олександр Миколайович, вул. Єреванська, 18-А, кв. 21, м. Київ, 03186
(повне ім'я автора, адреса)

заявка від **25.05.2016** **№ 67221**

про реєстрацію авторського права на твір і прийняла рішення зареєструвати авторське право на твір **Компьютерна програма "Компьютерная прорамма оптимизации иерархической структуры многопродуктовой коммуникационной сети с дискретными потоками"; Трофимчук Олександр Миколайович, Васянін Володимир Олександрович, Ушакова Людмила Павлівна**
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора (ів))

Внесення відомостей до Державного реєстру свідоцтв про реєстрацію авторського права на твір та видача свідоцтва будуть здійснені за умови сплати збору за оформлення і видачу свідоцтва про реєстрацію авторського права на твір відповідно до п.3 постанови Кабінету Міністрів України від 27 грудня 2001 року № 1756 "Про державну реєстрацію авторського права і договорів, які стосуються права автора на твір".

Якщо протягом трьох місяців від дати одержання заявником рішення про реєстрацію авторського права на твір Державна служба не одержала документ про сплату збору за оформлення і видачу свідоцтва у розмірі та порядку, визначених законодавством, або копію документа, що підтверджує право на звільнення від сплати зазначеного збору, заявка вважається відхиленою і реєстрація авторського права та публікація відомостей про реєстрацію Державною службою не проводиться.

В.о. Голови Державної служби інтелектуальної власності **А.А.Малиш**

УКРАЇНА

ДЕРЖАВНА СЛУЖБА ВЛАСНОСТІ УКРАЇНИ ІНТЕЛЕКТУАЛЬНОЇ

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 66794

Комп'ютерна програма "Компьютерная программа оптимизации распределения и маршрутизации дискретных потоков в многопродуктовой коммуникационной сети"

(вид, назва твору)

Автор(и) Васянін Володимир Олександрович

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 20.07.2016

Державна служба інтелектуальної власності України Ідентифікаційний код 37552556 М.П.

Голова Державної служби інтелектуальної власності України

В.о. Голови А.А. Малиш

ДЕРЖАВНА СЛУЖБА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ УКРАЇНИ
Україна, 03680, МСП, м. Київ-35,
вул. Урицького, 45
Тел. (044) 494-06-06
Факс (044) 494-06-67
E-mail: post@sips.gov.ua

STATE INTELLECTUAL PROPERTY SERVICE OF UKRAINE
Ukraine, 03680, MSP, Kyiv-35,
45, Urytskogo str.
Tel. (044) 494-06-06
Fax (044) 494-06-67
E-mail: post@sips.gov.ua

РІШЕННЯ

ПРО РЕЄСТРАЦІЮ АВТОРСЬКОГО ПРАВА НА ТВІР

Державна служба інтелектуальної власності розглянула заяву

Васянін Володимир Олександрович, вул. Салютна, 4а, кв. 39, м. Київ
(повне ім'я автора, адреса)

заявка від **25.05.2016** **№ 67224**

про реєстрацію авторського права на твір і прийняла рішення зареєструвати авторське право на твір **Комп'ютерна програма "Компьютерная программа оптимизации распределения и маршрутизации дискретных потоков в многопродуктовой коммуникационной сети"; Васянін Володимир Олександрович**
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора (ів))

Внесення відомостей до Державного реєстру свідоцтв про реєстрацію авторського права на твір та видача свідоцтва будуть здійснені за умови сплати збору за оформлення і видачу свідоцтва про реєстрацію авторського права на твір відповідно до п.3 постанови Кабінету Міністрів України від 27 грудня 2001 року № 1756 "Про державну реєстрацію авторського права і договорів, які стосуються права автора на твір".

Якщо протягом трьох місяців від дати одержання заявником рішення про реєстрацію авторського права на твір Державна служба не одержала документ про сплату збору за оформлення і видачу свідоцтва у розмірі та порядку, визначених законодавством, або копію документа, що підтверджує право на звільнення від сплати зазначеного збору, заявка вважається відхиленою і реєстрація авторського права та публікація відомостей про реєстрацію Державною службою не проводиться.

В.о. Голови Державної служби інтелектуальної власності **А.А.Малиш**

УКРАЇНА

ДЕРЖАВНА СЛУЖБА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ УКРАЇНИ

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 66793

Комп'ютерна програма "Компьютерная программа построения всех кратчайших путей в сети по двухкритериальному лексикографическому упорядочению"

(вид, назва твору)

Автор(и) Васянін Володимир Олександрович

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 20.07.2016

Голова Державної служби інтелектуальної власності України

В.о. Голови А.А. Малиш

Державна служба інтелектуальної власності України Ідентифікаційний код 37552556 М.П.

ДЕРЖАВНА СЛУЖБА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ УКРАЇНИ
Україна, 03680, МСП, м. Київ-35,
вул. Урицького, 45
Тел. (044) 494-06-06
Факс (044) 494-06-67
E-mail: post@sips.gov.ua

STATE INTELLECTUAL PROPERTY SERVICE OF UKRAINE
Ukraine, 03680, MSP, Kyiv-35,
45, Urytskogo str.
Tel. (044) 494-06-06
Fax (044) 494-06-67
E-mail: post@sips.gov.ua

РІШЕННЯ

ПРО РЕЄСТРАЦІЮ АВТОРСЬКОГО ПРАВА НА ТВІР

Державна служба інтелектуальної власності розглянула заяву

Васянін Володимир Олександрович, вул. Салютна, 4а, кв. 39, м. Київ
(повне ім'я автора, адреса)

заявка від **25.05.2016** № **67223**

про реєстрацію авторського права на твір і прийняла рішення зареєструвати авторське право на твір **Комп'ютерна програма "Компьютерная программа построения всех кратчайших путей в сети по двухкритериальному лексикографическому упорядочению"; Васянін Володимир Олександрович**
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора (ів))

Внесення відомостей до Державного реєстру свідоцтв про реєстрацію авторського права на твір та видача свідоцтва будуть здійснені за умови сплати збору за оформлення і видачу свідоцтва про реєстрацію авторського права на твір відповідно до п.3 постанови Кабінету Міністрів України від 27 грудня 2001 року № 1756 "Про державну реєстрацію авторського права і договорів, які стосуються права автора на твір".

Якщо протягом трьох місяців від дати одержання заявником рішення про реєстрацію авторського права на твір Державна служба не одержала документ про сплату збору за оформлення і видачу свідоцтва у розмірі та порядку, визначених законодавством, або копію документа, що підтверджує право на звільнення від сплати зазначеного збору, заявка вважається відхиленою і реєстрація авторського права та публікація відомостей про реєстрацію Державною службою не проводиться.

В.о. Голови Державної служби інтелектуальної власності **А.А.Малиш**

УКРАЇНА

ДЕРЖАВНА СЛУЖБА ВЛАСНОСТІ УКРАЇНИ ІНТЕЛЕКТУАЛЬНОЇ

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 66792

Комп'ютерна програма "Компьютерная программа нахождения остового леса графа с минимальным (максимальным) весом"

(вид, назва твору)

Автор(и) Трофимчук Олександр Миколайович, Васянін Володимир Олександрович, Ушакова Людмила Павлівна

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 20.07.2016

Державна служба інтелектуальної власності України Ідентифікаційний код 37552556 М.П.

Голова Державної служби інтелектуальної власності України
В.о. Голови А.А. Малиш

ДЕРЖАВНА СЛУЖБА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ УКРАЇНИ
Україна, 03680, МСП, м. Київ-35,
вул. Урицького, 45
Тел. (044) 494-06-06
Факс (044) 494-06-67
E-mail: post@sips.gov.ua

STATE INTELLECTUAL PROPERTY SERVICE OF UKRAINE
Ukraine, 03680, MSP, Kyiv-35,
45, Urytskogo str.
Tel. (044) 494-06-06
Fax (044) 494-06-67
E-mail: post@sips.gov.ua

РІШЕННЯ

ПРО РЕЄСТРАЦІЮ АВТОРСЬКОГО ПРАВА НА ТВІР

Державна служба інтелектуальної власності розглянула заяву

Трофимчук Олександр Миколайович, вул. Єреванська, 18-А, кв. 21, м. Київ, 03186
(повне ім'я автора, адреса)

заявка від **25.05.2016** **№ 67222**

про реєстрацію авторського права на твір і прийняла рішення зареєструвати авторське право на твір **Комп'ютерна програма "Компьютерная программа нахождения остового леса графа с минимальным (максимальным) весом"; Трофимчук Олександр Миколайович, Васянін Володимир Олександрович, Ушакова Людмила Павлівна**
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора (ів))

Внесення відомостей до Державного реєстру свідоцтв про реєстрацію авторського права на твір та видача свідоцтва будуть здійснені за умови сплати збору за оформлення і видачу свідоцтва про реєстрацію авторського права на твір відповідно до п.3 постанови Кабінету Міністрів України від 27 грудня 2001 року № 1756 "Про державну реєстрацію авторського права і договорів, які стосуються права автора на твір".

Якщо протягом трьох місяців від дати одержання заявником рішення про реєстрацію авторського права на твір Державна служба не одержала документ про сплату збору за оформлення і видачу свідоцтва у розмірі та порядку, визначених законодавством, або копію документа, що підтверджує право на звільнення від сплати зазначеного збору, заявка вважається відхиленою і реєстрація авторського права та публікація відомостей про реєстрацію Державною службою не проводиться.

В.о. Голови Державної служби інтелектуальної власності **А.А.Малиш**

Державна служба інтелектуальної власності України
Ідентифікаційний код 37525529