

Інститут телекомунікацій і глобального інформаційного простору
Національної академії наук України

Кваліфікаційна наукова
праця на правах рукопису

ГОРБОРУКОВ ВЯЧЕСЛАВ ВІКТОРОВИЧ

УДК 004.62+004.82+303.732.4+ 519.81

ДИСЕРТАЦІЯ

**ТЕХНОЛОГІЧНІ ЗАСОБИ ОНТОЛОГІЧНОГО СУПРОВОДУ
РОЗВ'ЯЗАННЯ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ**

05.13.06 – Інформаційні технології

05 Технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Горборуков В. В.

Науковий керівник:
Франчук Олег Васильович,
кандидат технічних наук,
доцент

Київ – 2018

АННОТАЦІЯ

Горборуков В. В. Технологічні засоби онтологічного супроводу розв'язання задач ранжування альтернатив. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Інститут телекомунікацій і глобального інформаційного простору Національної академії наук України, Київ, 2018.

Дисертаційну роботу присвячено вирішенню актуальної науково-технічної проблеми підвищення ефективності розв'язання багатокритеріальних задач ранжування та вибору альтернатив шляхом розробки та впровадження технологічних засобів, які використовують онтологічне представлення предметної області (ПдО).

У першому розділі «Методи і системи ранжування альтернатив» проведено огляд літератури за темою дисертації, проаналізовані процес розв'язання задач ранжування та MCDA методи, що можуть бути застосовані в дискретному випадку.

Процес проведення ранжування та вибору складається з наступних етапів: визначення проблемної задачі, структурування проблеми, реалізація оптимального вибору, пост-аналіз та отримання результату. Як відомо, помилки на етапі структуризації призводять до утворення хибної моделі задачі, яка швидше за все, призведе до неточних результатів. Отже, якість отриманого розв'язку задачі в першу чергу залежить від вдалого структурування, що вимагає від ОПР скрупульозної деталізації проблемної області для визначення критеріїв, альтернатив та іншої інформації. Саме цей етап може бути ефективно здійснений на основі онтологічного підходу, оскільки онтологія – це детальний опис предметної області за допомогою концептуальної схеми. Така схема складається з ієрархічної структури даних та містить інформацію про властивості об'єктів та відношення між ними.

Після проведення ранжування альтернатив (об'єктів) здійснюється пост-аналіз отриманого розв'язку. На цьому етапі повинна існувати можливість

додаткового дослідження таких об'єктів, які не стали «переможцями», але з огляду на специфіку конкретної задачі можуть представляти інтерес для особи, що приймає рішення (ОПР). В результаті виникає обернена задача ранжування, яка повинна визначити, на скільки тому чи іншому об'єкту необхідно покращити критеріальні значення, щоб у підсумковому рейтинговому списку посісти задане ОПР місце. Розв'язки таких обернених задач породжують додаткові властивості об'єктів дослідження, що може розширювати початкову онтологічну модель. У підсумку це призводить до підвищення ефективності процесу прийняття рішень.

Таким чином, онтологічна модель може бути цінним джерелом надходження інформаційного ресурсу на всіх етапах процесу прийняття рішень.

Досліджено існуючі MCDA програмні засоби, що можуть використовуватись у процесі розв'язання задачі ранжування альтернатив. Перелік проаналізованих засобів складається із 20 систем та включає такі програмно-інформаційні рішення, як: 1000Minds, Analytica, Criterium Decision Plus, DecideIT, D-Sight, GMAA, Logical Decisions, Promax, V.I.P. Analysis та інші.

У результаті проведеного аналізу виявлено, що на теперішній час не існує достатніх технологічних засобів для онтологічного супроводу роботи MCDA-систем. Тільки декілька з усіх програмних засобів містять певні елементи онтологічного супроводу, що здебільшого пов'язано з можливістю візуального формування таксономічної структури. Однак повноцінний онтологічний супровід полягає в застосуванні вже створеної експертом предметної області онтологічної моделі для формування та розв'язання задачі ранжування.

В другому розділі розроблено метод побудови інформаційного середовища задачі ранжування, що здійснює перетворення онтологічної моделі предметної області на основі інтерпретаційних функцій вибору, побудованих за допомогою гіпервідношень над елементами таксономічної структури онтології та властивостями її об'єктів.

Розроблено алгоритм конкурентної нормалізації критеріїв для задач ранжування та рейтингового оцінювання, що враховує конкурентність процесу встановлення ступеня домінування одних альтернатив над іншими в залежності від їх статистичних характеристик.

Сформульована обернена задача ранжування та розроблено алгоритми її розв'язання, що дозволяє на етапі пост-аналізу процесу проведення вибору (ранжування) альтернатив підвищити рівень аналізу отриманих результатів та розширити початкову онтологічну модель предметної області новими властивостями об'єктів дослідження.

В третьому розділі розроблено інформаційну та функціонально-компонентну модель системи розв'язку задач ранжування альтернатив. Інформаційну модель системи представлено сукупністю модулів, що інтегруються в інформаційно-аналітичну систему ТОДОС. Функціонально-компонентна модель програмної системи розв'язку задач ранжування альтернатив включає в себе моделі: поведінки системи, структури системи, структури програмних сутностей. Дані моделі включають в себе множини UML-діаграм варіантів використання, активності, взаємодії та діаграму класів. На основі вищенаведених моделей сформовано архітектуру програмної системи розв'язку задач ранжування альтернатив. Вона показує склад, структуру та особливості функціонування даної системи.

В четвертому розділі описуються інструментальні засоби, в яких програмно реалізована технологія онтологічного супроводу розв'язання задач ранжування.

«Альтернатива» – інструментальний засіб для розв'язку задачі ранжування. Онтологічна модель, що створюється експертом ПдО та/або ОПР є основою на якій формується модель задачі ранжування. В математичне забезпечення цієї системи входять основні методи багатокритеріального прийняття рішення (MCDA методи), а також алгоритм для розв'язку оберненої задачі ранжування.

«Оцінка досягнень» – інструментальний засіб, що дозволяє здійснювати міждисциплінарне рейтингування учасників конкурсних змагань, у тому числі учасників Всеукраїнського конкурсу-захисту науково-дослідницьких робіт учнів – членів Малої академії наук України.

«Технічне обслуговування»

В інструментальному засобі реалізована технологія перетворення онтологічної моделі територіально-розподіленої системи в оптимізаційну модель пошуку оптимальних планів ТО. Виявлення критичних елементів, для яких потрібне проведення ТО здійснюється за допомогою розв’язку задачі ранжування.

Запропоновано інформаційну технологію, що забезпечує підвищення ефективності діяльності ОПР внаслідок здійснення онтологічного супроводу процесів проведення вибору (ранжування) альтернатив.

Розроблені інструментальні засоби створені в рамках дослідних робіт № 0111U002232, № 0113U004981, № 0116U000794 Інституту телекомунікацій і глобального інформаційного простору та дослідних робіт, № 0114U002173, № 0115U002523 Національного центру «Мала академія наук України». Розроблені засоби впроваджені в наукову та прикладну діяльність: Інституту електрозварювання ім. Є. О. Патона НАНУ, Національного центру «Мала академія наук України», Національного університету «Києво-Могилянська академія», кафедри медичної інформатики Національної медичної академії післядипломної освіти ім. П. Л. Шупика, Комітету з питань будівництва, містобудування і житлово-комунального господарства Верховної ради України.

Ключові слова: ранжування альтернатив, багатокритеріальна оптимізація, MCDA-система, онтологія предметної області, таксономія, інформаційна технологія, дискретна оптимізація, динамічне програмування, послідовний аналіз варіантів.

Список опублікованих праць за темою дисертації

Праці, в яких опубліковані основні наукові результати дисертації:

1. Gorborukov V. Researching semistructured problems of multicriteria optimization using the software system / R. Trygub, O. Trygub, V. Gorborukov. // Наукові записки НаУКМА. Комп'ютерні науки. – 2013. – т. 151. – С. 79-88.
2. Gorborukov V. The inverse ranking problem and the algorithm for solving it / V. Gorborukov, O. Franchuk // Information Models and Analyses. – 2018. – Vol. 7, №. 2. – P. 52-62.
3. Горборуков В.В. Використання онтологій у системах підтримки прийняття рішень / В.В. Горборуков, О.Є. Стрижак, О.В. Франчук – Математичне моделювання в економіці: Зб. наук. праць // НАН України Ін-т телекомунікацій і глобал. інформ. простору, Ін-т економіки та прогнозування, Ін-т кібернетики ім. В.М. Глушкова; редкол.: С.О. Довгий (голов. ред.) [та ін.]. – К., 2013. – Вип. 3. – С. 33-40.
4. Горборуков В. В. Задача планування технічного обслуговування складних систем / В. В. Горборуков, О. В. Франчук. // Наукові записки НаУКМА. Комп'ютерні науки. – 2014. – т. 163. – С. 52-58.
5. Горборуков В. В. Застосування задачі вибору в онтографах для дослідження та аналізу стану здоров'я учнів на основі результатів медичної діагностики / В. В. Горборуков // Медична інформатика та інженерія. – 2015. – № 4. – С. 88-91.
6. Горборуков В. В. Основні підходи до створення системи підтримки прийняття рішень для запису на вибіркові навчальні дисципліни / В. В. Горборуков, О. В. Олецький. // Наукові записки НаУКМА. Комп'ютерні науки. – 2017. – Т. 198. – С. 54-58.
7. Горборуков В. В. Про одну задачу багатокритеріального вибору / В. В. Горборуков. // Наукові записки НаУКМА. Комп'ютерні науки. – 2015. – Т. 177. – С. 53-57.
8. Горборуков В.В. Сервіс автоматизованого запису на вибіркові навчальні дисципліни в НаУКМА та можливі напрямки його інтелектуалізації / В.В. Горборуков, О.В. Олецький. // Наукові записки НаУКМА. Комп'ютерні науки. – 2016. т. 190. – С. 52-56.

9. Стрижак О. Є. Онтологія задачі вибору та її застосування при аналізі лімнологічних систем / О. Є. Стрижак, В. В. Горборуков, О. В. Франчук, М. А. Попова // Екологічна безпека та природокористування. – 2014. – Вип. 15. – С. 172-183.

Праці, що засвідчують апробацію матеріалів дисертації:

10. Горборуков В.В. Застосування задачі вибору в онтографах. / В.В. Горборуков // Тези доповідей II Міжнародної науково-практичної конференції «Геоінформаційні технології у територіальному управлінні» (м. Одеса, 17 - 18 вересня 2015 року). – Одеса, 2015. – С. 45-49.

11. Горборуков В.В. Система підтримки прийняття адміністративних та управлінських рішень "VERUM EST" / В.В. Горборуков, О.В. Франчук // Зб. наук. праць за матеріалами VIII-ї Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» – ТАAPSD'2011 (Ялта, 19-23 вересня 2011 року). – Ялта, Україна, 2011. – С. 60-61.

12. Горборуков В. В. База даних «Національний рейтинг інтелектуальних досягнень учнівської молоді» / В. В. Горборуков, О. В. Франчук // Матеріали XIII-ї Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» – ТАAPSD'2016 (Київ, 5-9 грудня 2016 року). – Київ, 2016. – С. 57-61.

13. Горборуков В. В. Розв'язок задачі ранжування альтернатив в СППР «VERUM EST» / В. В. Горборуков // Матеріали 20-го ювілейного Міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXI столітті» (м. Харків, 19-21 квітня 2016 року) – Харків: ХНУРЕ, 2016. – С. 92-93.

14. Горборуков В. В. Система оцінювання інтелектуальних досягнень учнівської молоді. Онтологічний підхід / В.В. Горборуков, В.В. Приходнюк, О. Є. Стрижак, О. В. Франчук // Сборник трудов XVI Международной научной конференции «Интеллектуальный анализ информации (ИАИ-2016)» им. Т.А.Таран. Сборник трудов. – К.: Просвіта, 2016. – С. 36-42.

15. Приходнюк В. В. Методика ідентифікації структури слабоструктурованих документів з допомогою правил, побудованих на основі λ -виразів / В. В.

Приходнюк, В.В. Горборуков // тези III Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами» (м. Київ, 23 листопада 2016 року). – К.: НУХТ, 2016. – С. 256.

16. Франчук О. В. Параметрична оптимізація в задачі ранжування альтернатив / О. В. Франчук, В. В. Горборуков // Зб. наук. праць за матеріали Міжнародної наукової конференції «Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних технологій» (м. Рівне, 2-4 березня 2018 року). – Рівне, 2018. – С. 97–99.

Авторські свідоцтва:

17. Стрижак О. Є. Комп'ютерна програма “Трансдисциплінарні Онтологічні Діалоги Об'єктно-орієнтованих Систем (ТОДОС)” (“ІТ ТОДОС”) / О.Є. Стрижак, В.Ю. Величко, В.В. Приходнюк, С.О. Довгий, О.М. Трофимчук, О.В. Франчук, В.В. Горборуков, М.А. Попова, Л.С. Глоба // Свідоцтво про реєстрацію авторського права на твір № 79827. – Рішення від 15.06.2018 р.

18. Стрижак О. Є. Комп'ютерна програма “Трансдисциплінарна мережецентрична Інформаційно-аналітична система – (ТМІАС)” (“ТМІАС ”) / О.Є. Стрижак, В.Ю. Величко, В.В. Приходнюк, С.О. Довгий, О.В. Лісовий, В.В. Горборуков // Свідоцтво про реєстрацію авторського права на твір № 80835. – Рішення від 08.08.2018 р.

ANNOTATION

Gorborukov Viacheslav. Technological means of ontological support for solving problems of ranking alternatives. – Manuscript.

Dissertation research for degree of PhDs. by specialty 05.13.06 – Information technology. Institute of Telecommunications and Global Information Space of the National Academy of Sciences of Ukraine, Kyiv, 2018.

Thesis of dissertation focuses on solving an actual scientific and technical problem of improving the efficiency of solving multiple criteria problems of ranking and selecting alternatives by developing and introducing technological tools that use the ontological representation of the subject area.

In the first section, the review of the literature on the topic of the thesis was conducted and the process of solving the ranking problems and the MCDA methods, which can be applied in the discrete case, were analyzed.

The process of ranking and selection consists of the following stages: problem identifying, problem structuring, implementation of optimal choice (decision analysis), post-analysis and obtaining results. As known, mistakes at the stage of structuring lead to creating incorrect model of the problem, which is most likely to lead to inaccurate results. Thus, the quality of solution of the problem depends, first of all, on the successful structuring, which requires careful detailing of the problem area for determining the criteria, alternatives, and other information. This stage can be effectively realized on the basis of an ontological approach, since ontology is a detailed description of the subject area using a conceptual scheme. This scheme consists of a hierarchical data structure and contains information about the properties of the objects and relations between them.

After ranking the alternatives (objects) post-analysis of the obtained solution is carried out. At this stage, there should be a possibility for an additional research of such objects which did not become "winners", but taking into account the specifics of the problem these objects may be interesting for the decision maker. As a result, there is an inverse ranking problem, solution of which should determine how much is necessary to improve the criteria values of the specified object, which in the final ranking list would occupy the given place. Solutions of the inverse problems create additional properties of research objects, which can extend the initial ontological model. As a result, this increases the efficiency of the decision-making process.

Thus, the ontological model can be a valuable source of information resources at all stages of the decision-making process.

Existing MCDA software, which can be used for solving the problem of ranking alternatives, were researched. The list of analyzed tools consists of 35 systems and includes software for ranking, such as: 1000Minds, Analytica, Criterium Decision Plus, DecideIT, D-Sight, GMAA, Logical Decisions, Promax, V.I.P. Analysis and others.

As a result of the analysis it is established at present time there are not sufficient technological means for ontological support of the work of MCDA-systems. Only a few of the software contains certain elements of ontological support, which is mainly related with the possibility of visual formation of the taxonomic structure. However, a full ontological support is to apply existing ontological model of the subject area, which is already created by a domain expert, for the formation and solution of the ranking problem.

In the second section the method of constructing the information environment of the multiple criteria ranking problem, which carries out the transformation of the ontological model of the subject area on the basis of the interpretive selection functions constructed using hyper-relation over elements of the taxonomic structure of the ontology and properties of its objects, is developed.

The algorithm of competitive normalization of criteria for ranking problems and rating estimation is developed which takes into account the competitiveness of the process of establishing the degree of dominance of some alternatives over others based on their statistical characteristics.

The inverse ranking problem is formulated and algorithms for solving it are developed, which allow to increase the level of analysis of the obtained results at the post-analysis stage of the process of selection (ranking) alternatives and to extend the initial ontological model of the subject area with new properties of research objects.

In the third section an informational and functional-component model of the system for solving the problem of ranking alternatives is developed. The information model of the system is represented by a set of modules of the system for solving the problem of ranking alternatives, which is integrated in the information-analytical system TODOS. The functional-component model of the system includes models of: the behavior of the system, the structure of the system and the structure of software entities. These models include a set of UML diagrams: use, activity and interaction.

Based on described models, the architecture of the program system for solving the problem of ranking alternatives is created. It shows structure and features of the system.

The fourth section describes the tools in which the technology of ontological support for solving multi-criteria ranking problems is implemented.

«*Alternative*» – an instrumental tool for solving the multi-criteria ranking problem. An ontology model, created by an domain expert or decision maker, is the basis upon which the model of the ranking problem is formed. The mathematical core of this system includes the main methods and techniques of multi-criteria decision analysis (MCDA methods) and also algorithms for solving the inverse ranking problem. The system can display the received rating list of objects, both in tabular form and in the form of diagrams.

«*Achievements Rating*» – the tool that allow to create interdisciplinary rating for participants of competitive events, including participants of the All-Ukrainian competition for the protection of scientific research works of pupils, who are members of the Junior Academy of Sciences of Ukraine. The ontological approach was used for development this tool.

«*Maintenance*».

In this tool the technology of transforming the ontological model of the territorially distributed system into an optimization model for finding optimal maintenance plans is implemented. Identification of critical elements, which require maintenance services, can be carried out by solving the ranking problem. The algorithm for solution the problem of scheduling maintenance, based on the ideology of the method of dynamic programming, has been developed.

The information technology is proposed that allows improving the effectiveness of a decision-maker's activities as a result of the ontological support of processes of selection (ranking) of alternatives.

The developed instrumental tools were created within the scope of research works № 0111U002232, № 0113U004981, № 0116U000794 of the Institute of telecommunications and global information space and research works № 0114U002173, № 0115U002523 of the National center "Minor Academy of Sciences of Ukraine". The developed tools were used by: the Ye. O. Paton Electric Welding Institute of NAS Ukraine, National University of "Kyiv-Mohyla Academy", National

center “Minor Academy of Sciences of Ukraine”, department of medical informatics of the National medical academy of postgraduate Education named after. P. L. Shupyk, Committee on issues of construction, urban development and housing and communal services of the Verkhovna Rada of Ukraine.

Keywords: ranking alternatives, multiple-criteria decision analysis, MCDA software, domain ontology, taxonomy, information technology, discrete programming, dynamic programming, sequential analysis of variants.

List of the candidate’s publications

Articles in scientific and professional publications and foreign magazines:

1. Gorborukov V. Researching semistructured problems of multicriteria optimization using the software system / R. Trygub, O. Trygub, V. Gorborukov. // Наукові записки НаУКМА. Комп’ютерні науки. – 2013. – т. 151. – С. 79-88.

2. Gorborukov V. The inverse ranking problem and the algorithm for solving it / V. Gorborukov, O. Franchuk // Information Models and Analyses, 2018. – Vol. 7, №. 2. – С. 52-62.

3. Горборуков В. В. Використання онтологій у системах підтримки прийняття рішень / В.В. Горборуков, О. Є. Стрижак, О.В. Франчук – Математичне моделювання в економіці: Зб. наук. праць // НАН України Ін-т телекомунікацій і глобал. інформ. простору, Ін-т економіки та прогнозування, Ін-т кібернетики ім. В.М. Глушкова; редкол.:С.О. Довгий (голов. ред.) [та ін.]. – К., 2013. – Вип. 3. – С. 33-40.

4. Горборуков В. В. Задача планування технічного обслуговування складних систем / В.В. Горборуков, О.В. Франчук. // Наукові записки НаУКМА. Комп’ютерні науки. – 2014. – т. 163. – С. 52-58.

5. Горборуков В. В. Застосування задачі вибору в онтографах для дослідження та аналізу стану здоров’я учнів на основі результатів медичної діагностики / В. В. Горборуков // Медична інформатика та інженерія. – 2015. – № 4. – С. 88-91.

6. Горборуков В. В. Основні підходи до створення системи підтримки прийняття рішень для запису на вибіркові навчальні дисципліни / В.В.

Горборуков, О.В. Олецкий. // Наукові записки НаУКМА. Комп'ютерні науки. – 2017. – Т. 198. – С. 54-58.

7. Горборуков В. В. Про одну задачу багатокритеріального вибору / В. В. Горборуков. // Наукові записки НаУКМА. Комп'ютерні науки. – 2015. – Т. 177. – С. 53-57.

8. Горборуков В. В. Сервіс автоматизованого запису на вибіркові навчальні дисципліни в НаУКМА та можливі напрямки його інтелектуалізації / В.В. Горборуков, О.В. Олецкий. // Наукові записки НаУКМА. Комп'ютерні науки. – 2016. т. 190. – С. 52-56.

9. Стрижак О. Є. Онтологія задачі вибору та її застосування при аналізі лімнологічних систем / О. Є. Стрижак, В. В. Горборуков, О. В. Франчук, М. А. Попова // Екологічна безпека та природокористування. – 2014. – Вип. 15. – С. 172-183.

Published probatory works:

10. Горборуков В.В. Застосування задачі вибору в онтографах. / В.В. Горборуков // Тези доповідей II Міжнародної науково-практичної конференції «Геоінформаційні технології у територіальному управлінні» (м. Одеса, 17 - 18 вересня 2015 року). – Одеса, 2015. – С. 45-49.

11. Горборуков В.В. Система підтримки прийняття адміністративних та управлінських рішень "VERUM EST" / В.В. Горборуков, О.В. Франчук // Зб. наук. праць за матеріалами VIII-ї Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» – ТАAPSD'2011 (Ялта, 19-23 вересня 2011 року). – Ялта, Україна, 2011. – С. 60-61.

12. Горборуков В. В. База даних «Національний рейтинг інтелектуальних досягнень учнівської молоді» / В. В. Горборуков, О. В. Франчук // Матеріали XIII-ї Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» – ТАAPSD'2016 (Київ, 5-9 грудня 2016 року). – Київ, 2016. – С. 57-61.

13. Горборуков В. В. Розв'язок задачі ранжування альтернатив в СППР «VERUM EST» / В. В. Горборуков // Матеріали 20-го ювілейного

Міжнародного молодіжного форуму «Радіоелектроніка і молодь в ХХІ столітті» (м. Харків, 19-21 квітня 2016 року). – Харків: ХНУРЕ, 2016. – С. 92-93.

14. Горборуков В. В. Система оцінювання інтелектуальних досягнень учнівської молоді. Онтологічний підхід / В.В. Горборуков, В.В. Приходнюк, О. Є. Стрижак, О. В. Франчук // Сборник трудов XVI Международной научной конференции «Интеллектуальный анализ информации (ИАИ-2016)» им. Т.А.Таран. Сборник трудов. – К.: Просвіта, 2016. – С. 36-42.

15. Приходнюк В. В. Методика ідентифікації структури слабоструктурованих документів з допомогою правил, побудованих на основі λ -виразів / В. В. Приходнюк, В.В. Горборуков // тези III Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем управління організаційно-технічними та технологічними комплексами» (м. Київ, 23 листопада 2016 року). – К.: НУХТ, 2016. – С. 256.

16. Франчук О. В. Параметрична оптимізація в задачі ранжування альтернатив / О. В. Франчук, В. В. Горборуков // Зб. наук. праць за матеріали Міжнародної наукової конференції «Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних технологій» (м. Рівне, 2-4 березня 2018 року). – Рівне, 2018. – С. 97–99.

copyright certificates:

17. Стрижак О. Є. Комп'ютерна програма “Трансдисциплінарні Онтологічні Діалоги Об'єктно-орієнтованих Систем (ТОДОС)” (“ІТ ТОДОС”) / О.Є. Стрижак, В.Ю. Величко, В.В. Приходнюк, С.О. Довгий, О.М. Трофимчук, О.В. Франчук, В.В. Горборуков, М.А. Попова, Л.С. Глоба // Свідоцтво про реєстрацію авторського права на твір № 79827. – Рішення від 15.06.2018 р..

18. Стрижак О. Є. Комп'ютерна програма “Трансдисциплінарна мережецентрична Інформаційно-аналітична система – (ТМІАС)” (“ТМІАС ”) / О.Є. Стрижак, В.Ю. Величко, В.В. Приходнюк, С.О. Довгий, О.В. Лісовий, В.В. Горборуков // Свідоцтво про реєстрацію авторського права на твір № 80835. – Рішення від 08.08.2018 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	17
ВСТУП.....	18
РОЗДІЛ 1. МЕТОДИ І СИСТЕМИ РАНЖУВАННЯ АЛЬТЕРНАТИВ.....	24
1.1. Задачі ранжування альтернатив.....	24
1.1.1. Виникнення і розвиток математичних методів ранжування та вибору альтернатив.....	24
1.1.2. Етапи процесу проведення вибору.....	28
1.1.3. Методи ранжування альтернатив.....	29
1.2. Використання онтологій як основи представлення альтернатив.....	36
1.3. Існуючі MCDA системи	40
1.3.1. Огляд існуючих MCDA систем	41
1.3.2. Порівняння існуючих MCDA систем	44
Висновки за розділом 1	50
РОЗДІЛ 2. МАТЕМАТИЧНІ МОДЕЛІ ТА АЛГОРИТМИ РОЗВ'ЯЗКУ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ.....	51
2.1. Онтологічне представлення задачі ранжування альтернатив	51
2.2. Алгоритм конкурентної нормалізації критеріїв.....	62
2.3. Обернена задача ранжування альтернатив.....	73
2.3.1. Постановка оберненої задачі ранжування альтернатив.....	73
2.3.2. Розв'язання оберненої задачі ранжування альтернатив методом послідовного аналізу варіантів	75
2.3.3. Розв'язання оберненої задачі ранжування альтернатив методом динамічного програмування	86
Висновки за розділом 2	95
РОЗДІЛ 3. АРХІТЕКТУРА СИСТЕМИ РОЗВ'ЯЗКУ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ	96
3.1. Інформаційна модель програмної системи розв'язку задач ранжування альтернатив	96
3.2. Функціонально-компонентна модель програмної системи розв'язку задач ранжування альтернатив	101
3.2.1. Модель поведінки системи	101
3.2.2. Модель структури системи	106
3.2.3. Модель структури програмних сутностей	111
3.3. Трирівнева клієнт-серверна архітектура програмної системи розв'язку задач ранжування альтернатив	113
Висновки за розділом 3	116

РОЗДІЛ 4. ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ФОРМУВАННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СЕРЕДОВИЩ НА ОСНОВІ РОЗВ'ЯЗУВАННЯ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ.....	117
4.1. Інструментальний засіб «АЛЬТЕРНАТИВА»	117
4.1.1. Представлення предметної області у вигляді онтологічної моделі для задачі ранжування у середовищі ТОДОС	118
4.1.2. Створення інформаційного середовища задачі ранжування на основі онтологічної моделі предметної області	122
4.1.3. Отримання розв'язку задачі ранжування альтернатив	124
4.2. Інструментальний засіб «Оцінка досягнень».....	126
4.2.1. Призначення системи та її основні можливості	128
4.2.2. Формування запитів до системи оцінювання досягнень учнів.....	138
4.2.3. Проведення аналітичних досліджень.....	141
4.3. Інструментальний засіб «Технічне обслуговування»	143
4.2.3. Постановка задачі планування технічного обслуговування	144
4.2.3. Алгоритм розв'язання задачі технічного планування методом динамічного програмування	147
4.2.3. Розв'язання задачі планування технічного обслуговування на основі онтологічної моделі територіально-розподіленої системи.....	150
Висновки за розділом 4	153
ВИСНОВКИ.....	155
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	157
Додаток А. Акти впровадження.....	176
Додаток Б. Свідоцтва про реєстрацію авторського права на твори.....	186
Додаток В. Список опублікованих праць за темою дисертації.....	190
Додаток Г. Відомості про апробацію результатів дисертації.....	193
Додаток Д. Методичні рекомендації до формування даних задачі ранжування альтернатив за сукупністю показників	194
Додаток Е. Трансдисциплінарна Мережецентрична Інформаційно-Аналітична Система (ТМІАС).....	198
Додаток Є. Архітектура ІТ-платформи ТОДОС (Трансдисциплінарні Онтологічні Діалоги Об'єктно-орієнтовних Систем)	213

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AHP	–	Analytic Hierarchy Process
CSV	–	Comma Separated Values
ELECTRE	–	ELimination Et Choix Traduisant la REalité (ELimination and Choice Expressing the REality)
JSON	–	JavaScript Object Notation
MA	–	Multi-Attribute
MCDA	–	Multi-Criteria Decision Analysis
MCDM	–	Multiple-Criteria Decision-Making
MOO	–	Multi-Objective Optimization
PROMETHEE	–	Preference Ranking Organization Method for Enrichment Evaluations
TOPSIS	–	Technique for Order Preference by Similarity to Ideal Solution
VIKOR	–	VIseKriterijumska Optimizacija I Kompromisno Resenje (Multicriteria Optimization and Compromise Solution)
XML	–	eXtensible Markup Language
IAC	–	інформаційно-аналітична система
ОПР	–	особа, що приймає рішення;
МАН	–	Мала Академія наук України
ПАВ	–	послідовний аналіз варіантів;
ПдО	–	предметна область
ПЗ	–	програмне забезпечення
СППР	–	система підтримки прийняття рішень;
ТМІАС	–	Трансдисциплінарна Мережецентрична Інформаційно-Аналітична Система
ТО	–	технічне обслуговування.
ТОДОС	–	Трансдисциплінарні Онтологічні Діалоги Об'єктно-орієнтованих Систем

ВСТУП

Актуальність дослідження. Необхідність в ранжуванні та виборі альтернатив виникає в багатьох сферах людської діяльності, пов'язаних з розв'язанням прикладних задач, що сприяють ефективному прийняттю обґрунтованих науково-технічних рішень. Складність і різноманіття ситуацій вибору вимагає врахування великої кількості різних факторів та критеріїв і потребує значного рівня компетенції та об'єму знань від особи, що приймає рішення (ОПР). Методи і підходи, що визначають процеси багатокритеріального вибору і прийняття рішень, розроблені в працях вітчизняних і зарубіжних вчених: Михалевича В.С, Волковича В.Л., Кнопова П.С., Згуровського М.З., Панкратової Н.Д., Волошина О.Ф., Заславського В.А., Айзермана М.А., Ларичева О.І., Малишевського А.В., Подиновського В.В., Ногіна В.Д., Keeney R.L., Raiffa H., Saaty T.L, Hwang C.L., Yoon K.P, Opricovic S., Zadeh L., Roy B., Brans J.P., Pareto V., Von Neumann J та інших.

Вирішення складних прикладних задач, що мають певну практичну значимість, багато в чому залежить від об'єктивності та достовірності інформації, яка найбільш виразно задається за допомогою таксономій. Таксономії здатні відобразити як структурні особливості предметної області задачі, так й відношення між поняттями, що утворюють її описи. Тому ефективним засобом представлення інформації в задачах ранжування є онтологія, яка формується на основі таксономій та функціональних властивостей понять, що їх утворюють.

Більш того, застосування онтологічних систем при розв'язанні задач ранжування, визначає процес їх вирішення як знаннево-орієнтовний, що лежить у напрямках, пов'язаних зі створенням та використанням різноманітних засобів обробки інформації, як пасивної системи мережевих знань. Створення та застосування можливостей складних та знання-орієнтованих інформаційних систем розглядалися у дослідженнях Глушкова В. М., Гладуна В. П., Палагіна О. В., Широкова В. А., Анісімова А. В., Хорошевського В. Ф., Поспєлова Д. А., Овдій О. М., Гаврилової Т. А., Загорулька Ю. А., Марченко О.О., Андона П. І.,

Валькмана Ю. Р., Соловйової К. О., Стрижака О. Є., Яловця А. Л., Gruber T., Guarino N., Noy N., Corcho O., Gomez-Perez A., Gruninger M., Fernandez-Lopez M., Happel H., McGuinness D., Lee J. та інших.

Таким чином, актуальним є розроблення технології розв'язання задач ранжування та вибору, яка б забезпечила можливість використання властивостей онтологічних моделей предметних областей для підвищення ефективності розв'язання задач прийняття рішень.

Зв'язок роботи з науковими програмами, темами, планами. Дисертацію виконано в межах наукових тем Інституту телекомунікацій і глобального інформаційного простору: «Розроблення інформаційно-аналітичного центру місцевих органів виконавчої влади» (2011-2013 рр., ДР № 0111U002232); «Створення інформаційних та алгоритмічних моделей і засобів трансдисциплінарної інтеграції мереж знань», 2014–2016 рр. (ДР № 0113U004981); «Створення програмно-інформаційних засобів інформаційно-аналітичного забезпечення мережецентричних ситуаційних центрів», 2016–2020 рр. (ДР № 0116U000794). Також дисертаційне дослідження є частиною наукових тем Національного центру «Мала академія наук України»: «Створення інформаційно-аналітичної системи та методичних засобів моніторингу та оцінювання навчальних досягнень учнівської молоді на основі онтологій рішення задачі вибору», 2014–2016 рр. (ДР № 0114U002173), «Створення інформаційно-аналітичної системи та методичних засобів моніторингу та оцінювання навчальних досягнень учнівської молоді на основі онтологій рішення задачі вибору», 2015–2017 рр. (ДР № 0115U002523).

Роль автора в науково-дослідних роботах та проектах полягає в розробці методів, засобів та відповідної інформаційної технології розв'язання задач ранжування на основі формування та використання онтологічних систем.

Об'єкт дослідження – багатокритеріальне прийняття рішень, онтологічні моделі і системи.

Предмет дослідження – методи і засоби ранжування альтернатив на основі онтологічного представлення предметної області.

Мета і завдання дослідження – розроблення технології розв’язання задач ранжування та вибору на основі використання онтологічних систем.

Для досягнення поставленої мети необхідно було вирішити такі задачі:

- дослідити існуючі MCDA системи, зокрема, на предмет використання ними онтологічного представлення проблемних задач;
- дослідити способи формального опису онтологічних структур;
- програмно реалізувати основні математичні методи та алгоритми ранжування об’єктів, що характеризуються сукупністю різних за вагомістю показників;
- виробити технологічний підхід для використання онтологічних структур в процесі формування і розв’язання задачі ранжування альтернатив та здійснення постаналізу отриманих результатів на основі переходу до оберненої задачі;
- сформулювати та розробити алгоритм розв’язку оберненої задачі ранжування.

Методи та теоретичні засади дослідження. Розв’язання задач дисертаційного дослідження ґрунтувалось на методах системного аналізу, дискретної та багатокритеріальної оптимізації, теоретичних засадах прийняття рішень, теорії графів, теорії множин; для програмної реалізації інформаційної технології – методології динамічного програмування, послідовного аналізу та відсіву варіантів, шаблони проектування та об’єктно-орієнтований аналіз.

Наукова новизна дослідження визначається тим, що вперше:

- визначено умови застосування онтологій в задачі ранжування;
- створено технологію розв’язання задачі ранжування на основі онтологічного представлення предметної області;
- сформульована обернена задача ранжування та створено її математичну модель;
- розроблені алгоритми для розв’язку оберненої задачі ранжування;

- розроблено алгоритм конкурентної нормалізації критеріїв для задач ранжування та рейтингового оцінювання альтернатив;
- запропонований підхід застосування онтологічних систем для розв'язку оптимізаційних задач.

Удосконалено:

- онтологічна модель задачі вибору;
- алгоритм знаходження оптимального плану технічного обслуговування ієрархічної системи заданої у вигляді онтологічного графу;
- технологія перетворення табличних даних в онтографи.

Таким чином, у дисертаційній роботі вперше розроблена технологія онтологічного супроводу розв'язання задач ранжування альтернатив за сукупністю показників, що дозволяє ефективно використовувати онтологічне представлення предметної області.

Практичне значення дисертаційного дослідження:

- на основі онтологічних графів створена та програмно реалізована технологія розв'язання задачі ранжування, в якій здійснення пост-аналізу відбувається в результаті розгляду оберненої задачі;
- розроблено алгоритмічне забезпечення для розв'язку оберненої задачі ранжування;
- реалізовано алгоритм конкурентної нормалізації у системі рейтингового оцінювання досягнень учнівської молоді в інтелектуальних конкурсах;
- для елементів складної технічної системи на основі її онтологічного представлення розроблено інструментальний засіб знаходження оптимальних планів проведення регулярних регламентно-відновлювальних робіт.

Практичне значення одержаних результатів дослідження підтверджується актами про впровадження в наукову, прикладну та освітню діяльність: Інституту електрозварювання ім. Є. О. Патона НАНУ; Національного центру «Мала академія наук України»; Національного університету «Києво-Могилянська академія» (навчальні курси «Моделювання інформаційних процесів» та «Інноваційні моделі і технології: теорія і практика» магістерської

програми “Інженерія програмного забезпечення”, система автоматизованого запису для обрання студентами вибірових дисциплін в навчальному процесі); кафедри медичної інформатики Національної медичної академії післядипломної освіти ім. П. Л. Шупика.

Особистий внесок здобувача. Основні ідеї та наукові результати дисертаційної роботи отримані автором особисто та висвітлені у шістнадцяти працях, чотири з яких виконані одноосібно: [24, 25, 27, 28].

У наукових працях, опублікованих у співавторстві, автору дисертації належать: онтологічне представлення предметної області в задачах ранжування альтернатив [22, 30, 69]; онтологія задачі вибору та задачі ранжування альтернатив [80]; модель багатокритеріального вибору на основі онтології тем предметів [26, 29]; пост-аналіз в задачі ранжування альтернатив у випадку застосування методу аналізу ієрархій [118]; постановка оберненої задачі ранжування альтернатив та алгоритм її розв’язання [86, 119]; алгоритм конкурентної нормалізації критеріїв в задачах рейтингового оцінювання [21]; алгоритм пошуку оптимальних планів технічного обслуговування елементів складної системи [23]; інструментальні засоби підтримки прийняття рішень в задачах ранжування альтернатив [31].

Апробація результатів. Основні положення та результати дисертаційного дослідження було обговорено та викладено в доповідях на наукових конференціях та науково-практичних заходах: VIII Міжнародна конференція «Теоретичні та прикладні аспекти побудови програмних систем» – TAAPSD’2011, (м. Ялта, 2011 р.); XII міжнародна науково-практична конференція «Розробка і впровадження сучасних інформаційних технологій підтримки рішень по забезпеченню екологічної безпеки регіонів та ефективному використанню їх відновлюваних природних ресурсів» (Кацівелі, 2013 р.); II міжнародна науково-практична конференція «Геоінформаційні технології у територіальному управлінні» (м. Одеса, 2015 р.); I Міжнародний науково-практичний форум «Наука і бізнес» (м. Чернівці, 2015 р.) ; XX-й міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті»

(м. Харків, 2016 р.) ; XVI Міжнародна наукова конференція «Інтелектуальний аналіз інформації» ім. Т. А. Таран (м. Київ, 2016 р.); XIII Міжнародна конференція «Теоретичні та прикладні аспекти побудови програмних систем» – ТАAPSD'2016, (м. Київ, 2016 р.); Міжнародній науковій конференції «Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних технологій (м. Рівне, 2018 р.); спільна нарада представників ДП «КБ «Південне» і Інституту телекомунікацій і глобального інформаційного простору НАН України щодо використання когнітивної ІТ-платформи «ТОДОС» у розробці ракетно-космічної техніки (Дніпро, 2018 р.); нарада у президента НАН України академіка НАН України Б. Є. Патона щодо результатів апробації в Інституті електрозварювання ім. Є. О. Патона та можливості подальшого використання інформаційної ІТ-платформи «ТОДОС» (Київ, 2018 р.).

Публікації. За матеріалами дисертації з викладенням її основних результатів опубліковано 16 наукових праць, серед яких 9 – у фахових наукових виданнях (із них 2 – одноосібно, 1 – в наукових періодичних виданнях іноземних держав, 1 – у виданнях України, що включено до міжнародних наукометричних баз), 7 – в інших виданнях, збірниках і матеріалах наукових конференцій.

Структура роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (156 позицій) та семи додатків. Загальний обсяг дисертації становить 216 сторінок, із них 142 сторінки основного тексту. Робота містить також 14 таблиць та 39 рисунків.

РОЗДІЛ 1. МЕТОДИ І СИСТЕМИ РАНЖУВАННЯ АЛЬТЕРНАТИВ

1.1. Задачі ранжування альтернатив

Необхідність в ранжуванні альтернатив виникає в багатьох сферах людської діяльності, пов'язаних з розв'язанням прикладних задач, що сприяють ефективному прийняттю обґрунтованих науково-технічних рішень. Складність і різноманіття ситуацій вибору вимагає врахування великої кількості різних факторів і критеріїв.

1.1.1. Виникнення і розвиток математичних методів ранжування та вибору альтернатив

Задача вибору обговорювалася багатьма великими стародавніми філософами, включаючи Платона, Аристотеля і Фоми Аквінського [112]. Можна вважати, що саме ця задача фактично була у витоків теорії прийняття рішень. Перший документований підхід, який привів до моделювання проблеми вибору (прийняття рішення), виник з обговорення між Блейз Паскалем та П'єром де Ферма в 1654 році. Обговорення стосувалося азартної гри з двома гравцями, які мають рівні можливості виграти грошову суму. Гравці вносять рівну стартову плату, щоб брати участь в цій грі і погоджуються на її правила (виграшні умови). Проте, математики визначили проблему, яка виникає при розгляді справедливого розподілу грошей, якщо їм потрібно закінчити гру раніше, до того, як будуть досягнуті умови виграшу. Паскаль і Ферма (1654-1660) незалежно розробили рішення, засноване на одному і тому ж фундаментальному принципі. Вони погодилися, що поділ має бути пропорційний шансу кожного гравця на перемогу. Це стало основою теорії очікуваних значень (Expected value theory) [10], яка враховує ймовірність виграшу, помноженого на його вартість.

Однак було встановлено, що поведінка людини може порушувати теорію очікуваних значень, згідно якої особа при прийнятті рішення повинна завжди прагнути до максимального очікуваного значення. Щоб продемонструвати недосконалість цієї теорії, Ніколя Бернуллі у 1713 р. винайшов проблему, яка

отримала назву Санкт-Петербурзький парадокс [130]. Розглядається гра, в якій підкидається монета зі сторонами А і В. Гра триває до тих пір, поки не з'явиться сторона В. Гравець вносить фіксовану плату за вхід у гру і отримує початкову суму, меншу від плати за вхід. При кожному випадінні сторони А сума грошей гравця подвоюється. Основна проблема, пов'язана з цим парадоксом, – це визначення справедливої плати за гру. При розгляді цієї проблеми з використанням теорії очікуваних значень виграшна сума в цій грі завжди сходиться до нескінченності. Тому, за цієї теорією поміркований гравець повинен увійти в гру, заплативши будь-яку плату. Однак насправді кожна розумна людина керується власними міркуваннями, що враховують обмеження особистого багатства і свою толерантність до ризику. Беручи це до уваги, Даніель Бернуллі, двоюрідний брат Ніколаса Бернуллі, запропонував розв'язок парадоксу в Санкт-Петербурзі в 1738 році – використання логарифмічної функції корисності для зміни очікуваного значення в залежності від багатства гравця [95]. Це було першою систематичною появою теорії очікуваної корисності. В сучасному вигляді ця теорія розглядає цінність альтернативи, яка залежить від кардинальних функцій корисності. Цей підхід був запропонований творцями теорії ігор – Джоном Фон Нейманом і Оскаром Моргенштерном [152].

Побудова функції корисності зазвичай вимагає розгляду двох або більше критеріїв. Ці критерії часто взаємозалежні і/або конфліктують один з одним. Таким чином, виникає багатокритеріальна задача, яка вперше була розглянута в роботах Еджворта, а в подальшому – досліджена Парето [67]. Еджворт запропонував термін *optimum*, щоб вказати ідеальну точку між низкою компромісів по багатокритеріальній проблемі. Ця точка є Еджворто-Парето оптимальною, або більш загально – Парето оптимальною (оскільки Парето узагальнив теорію).

Точка називається оптимальною по Парето, якщо не існує іншого допустимого розв'язку багатокритеріальної задачі, який би покращував деякий окремий критерій, не спричиняючи погіршення при цьому щонайменше одного

іншого критерію. Зазвичай існує не тільки одна оптимальна точка Парето, а набір рішень, який називають оптимальною множиною Парето. Якщо оптимальна множина Парето розглядається в дво- або тривимірному просторі, то її називають границею Парето (рис. 1.1). Границя Парето часто використовується для ілюстрації тієї множини альтернатив, саме серед яких особа, що приймає рішення (ОПР) повинна обрати розв'язок, а не розглядати всі можливі альтернативи.

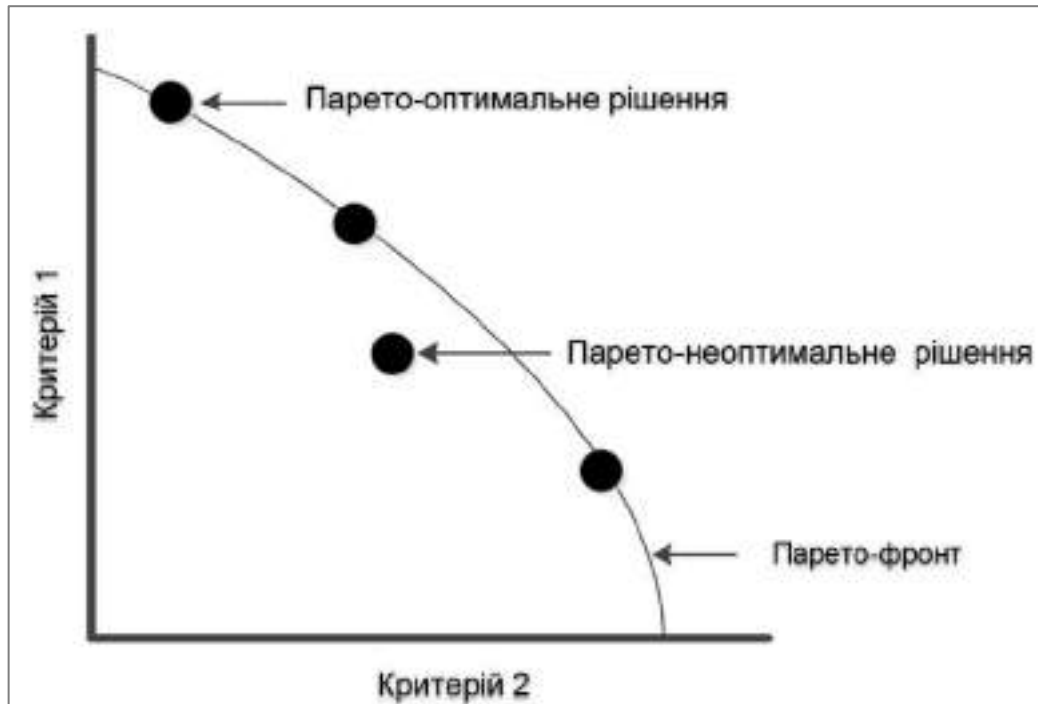


Рис. 1.1. Границя Парето для двокритеріальної задачі

За останні п'ятдесят років з'явилося достатньо багато різних методів прийняття рішень для сортування, ранжування або кількісної оцінки альтернатив на основі оптимального вибору Парето. Ці методи можна розділити на три категорії: Multi-Objective Optimization (MOO), Multi-Attribute (MA), Outranking [112, 115, 123, 133].

Алгоритми MOO засновані на багатокритеріальній оптимізації (максимізації або мінімізації) цільових функцій (критеріїв). Як правило, в цьому підході немає спроби формалізувати функції корисності ОПР. Замість цього алгоритми MOO використовують неявну інформацію про переваги ОПР для здійснення пошуку рішення. Зазвичай методи MOO використовуються, коли є велика або нескінченна кількість можливих рішень.

Методи МА і методи Outranking зазвичай використовуються в дискретних задачах прийняття рішень з невеликим або середнім числом альтернативних рішень. Outranking відрізняються від методів МА, оскільки вони допускають, що альтернатива може мати ступінь домінування над іншою. Це інтерпретується відношеннями переваг кожної можливої пари альтернатив. В результаті визначається часткова або повна впорядкованість множини альтернатив. Навпаки, методи МА агрегують всі критерії в узагальнену функцію, яка оптимізується. Хоча МА і Outranking підходи часто вважаються схожими, застосування методів МА призводить до числових результатів, тоді як методи Outranking продукують порядковий ранг для визначення переваги одного рішення над іншим.

Всі три категорії методів MCDA включають в себе ряд підходів, які мають свої характерні переваги і обмеження. На рис. 1.2 у вигляді деревовидної структури представлена хронологія розвитку теорій та методів багатокритеріального прийняття рішень.

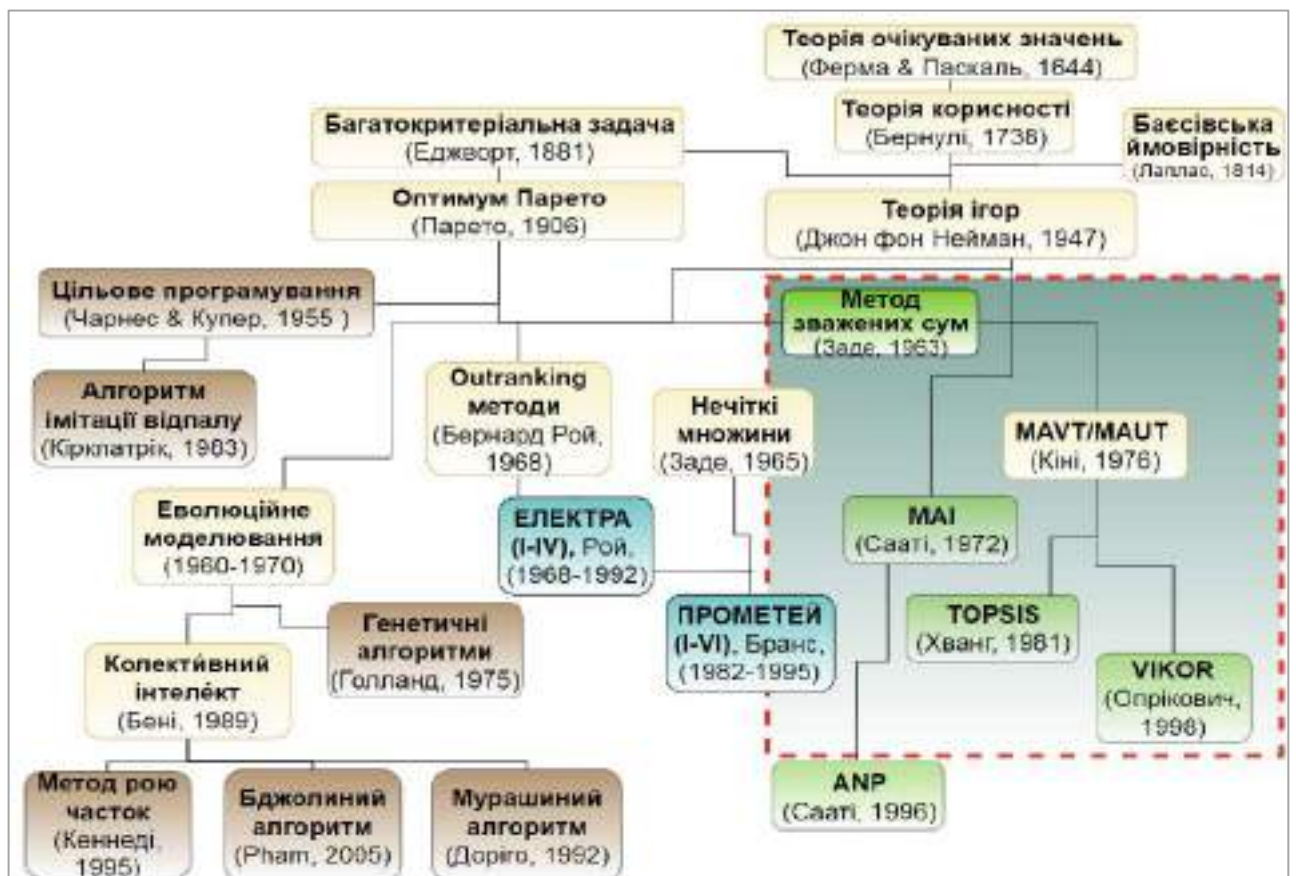


Рис. 1.2. Еволюція методів багатокритеріальної оптимізації

1.1.2. Етапи процесу проведення вибору

Більшість досліджень в прийнятті рішень зосереджено головним чином на розробці методів або застосуванні раніше створених підходів до конкретних задач. При цьому мало уваги приділяється безпосередньо самому процесу прийняття рішень [94]. Цей процес в багатьох випадках зводиться до задачі вибору, яка розв'язується в результаті ранжування об'єктів дослідження.

Процес вибору складається з декількох етапів (рис. 1.3) – структурування

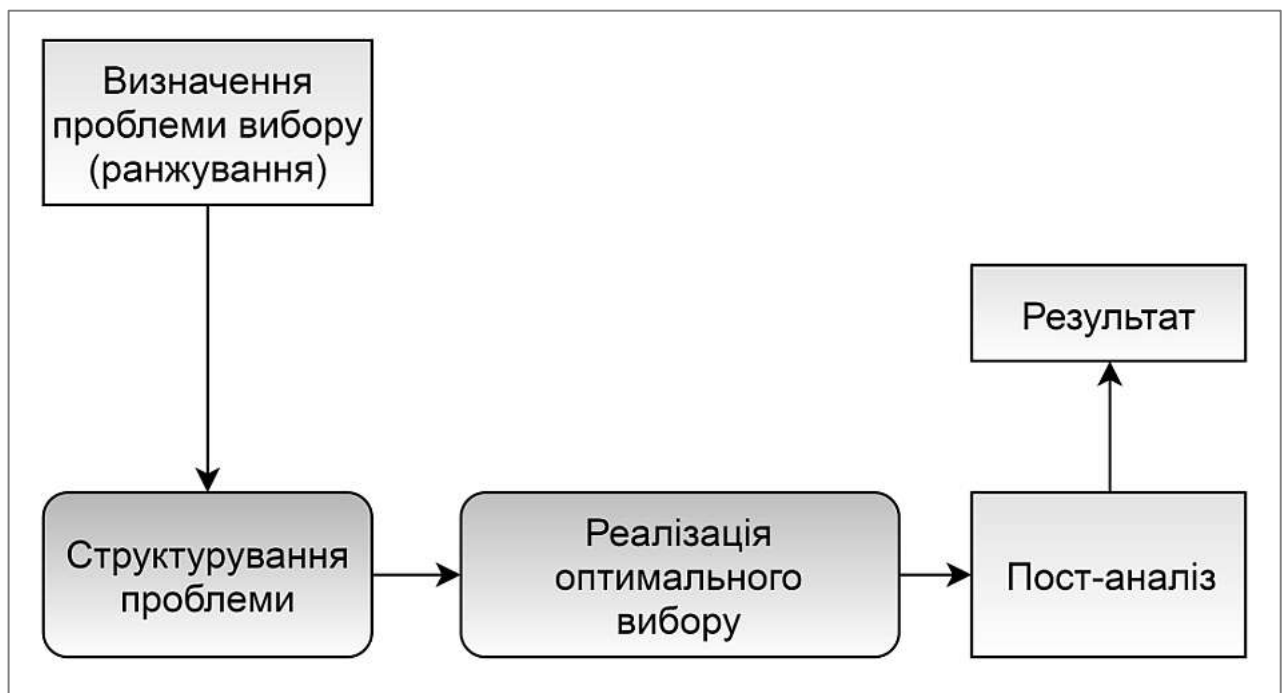


Рис. 1.3. Процес проведення вибору (ранжування)

проблеми, реалізація оптимального вибору і пост-аналіз отриманого розв'язку:

- Структурування проблем
 - визначення цілей / критеріїв
 - визначення альтернатив
- Реалізація оптимального вибору
 - Визначення вагових коефіцієнтів критеріїв
 - Встановлення кількісних і якісних значень критеріїв для альтернативних розв'язків
 - Вибір MCDA методів та алгоритмів

- Пост-аналіз
 - Суб'єктивна оцінка результату
 - Аналіз чутливості розв'язку

В [114] етап структурування проблеми розглядається як найменш важливий аспект процесу проведення вибору. Це пов'язано із загальним припущенням про те, що формальне уточнення добре структурованої проблеми є тривіальним завданням. Багато осіб, які приймають рішення, хочуть швидко перейти до безпосереднього здійснення оптимального вибору, нехтуючи тим, що помилкова модель, швидше за все, призведе до неточних результатів. Проте етап структурування являє собою складну задачу, яка вимагає від ОПР скрупульозної деталізації проблеми з метою визначення критеріїв, альтернатив та іншої ключової інформації [114, 127]. В роботах [94, 108] був запропонований підхід, що базується на ідеях динамічного структурування задачі та ітераційності цього процесу.

- Визначення вагових коефіцієнтів критеріїв
- Встановлення кількісних і якісних значень критеріїв для альтернативних розв'язків
- Вибір MCDA методів та алгоритмів

Етап реалізації оптимального вибору включає встановлення критеріальних значень для альтернатив, визначення важливостей критеріїв, що в більшості випадків ґрунтується на особистісних перевагах (преференціях) ОПР. Способи інтеграції цих значень в розв'язання задачі залежать від обраних MCDA методів.

Останнім етапом процесу вибору є пост-аналітичне дослідження [123]. Тут ОПР при необхідності може поставити під сумнів отриманий розв'язок або переконатись у його достовірності.

1.1.3. Методи ранжування альтернатив

Багатоатрибутні методи створюють один чисельний результат для оцінки кожної доступної альтернативи. Це досягається в два етапи. По-перше, особа,

яка приймає рішення, має визначити і сформувані набори альтернатив, критеріїв з їх значеннями, які описують проблему прийняття рішення. По-друге, для узагальнення цієї інформації в підсумковий бал застосовується метод, який використовує функцію цінності або корисності. Багатоатрибутні методи відносять до Multi-Attribute Value Theory (MAVT) чи Multi-Attribute Utility Theory (MAUT) [123]. Виключенням з цього розподілу можна вважати метод аналізу ієрархії (MAI), оригінальна назва якого є Analytic Hierarchy Process (АНР) [74]. АНР працює з парними порівняннями. Однак в [110], і ряд інших публікацій класифікують АНР як підхід MAUT.

Незважаючи на те, що на створення концепцій MAVT і MAUT претендує цілий ряд авторів, загальноновизнано, що ці концепції вперше були представлені Кіні і Райффа у 1976 р. [126]. Основна відмінність між теоріями MAVT і MAUT полягає в тому, що перша оперує з задачами при повній визначеності критеріїв, в той час як MAUT займається прийняттям рішень в умовах невизначеності. MAUT опрацьовує невизначеність з використанням функцій корисності на відміну від MAVT, яка застосовує функції цінності.

MAVT і MAUT передбачають, що значення критеріїв в задачі повинні належати до спільної шкали. Якщо це не так, то повинні використовуватись процедури нормалізації значень критеріїв в безрозмірний загальний формат [126]. Спосіб, яким це реалізується для певного методу, визначає сам метод.

Метод зваженої суми (WSM – Weighted Sum Method), також відомий як метод простої адитивної маси, був введений Л. Заде у 1963 р. [156]. Він простий для розуміння та застосування і є одним з найбільш широко вживаних багатоатрибутних методів [107].

WSM дозволяє ОПР визначати вагові коефіцієнти критеріїв, як важливість функції. Загальний бал кожної альтернативи дорівнює сумі добутків вагових коефіцієнтів і відповідних критеріальних значень альтернатив:

Незважаючи на те, що WSM широко використовується, він є не допустимим для задач з різними шкалами значень критеріїв [138]. Тому перед використанням WSM необхідне попереднє застосування процедур нормалізації.

Метод WSM заснований на неявному постулаті: "низьке значення по одному критерію може бути компенсовано високим значенням за іншим". Однак, цей постулат вірний далеко не для всіх моделей. Найпростіший приклад – незначний розмір оперативної пам'яті комп'ютера не може бути компенсований великим об'ємом зовнішньої пам'яті. Таким чином, застосування WSM може призвести до результатів, що суперечать здоровому глузду.

Метод аналізу ієрархій був запропонований Сааті [74] як спосіб вирішення задачі прийняття рішення з використанням її природної ієрархічної структури. АНР став одним з найпопулярніших методів прийняття рішень через використання парних порівнянь для отримання кількісної інформації на основі лінгвістичної шкали [92, 93, 139].

Парні порівняння відображаються в зворотньо-симетричній матриці. Числові значення w від ОПР, що відповідають певному рівню в лінгвістичній шкалі, для кожної пари об'єктів, що порівнюються (критерії, альтернативи), вносяться в матрицю. В транспоновані позиції автоматично додаються зворотні значення $1/w$. Локальні пріоритети критеріїв і альтернатив по кожному критерію обраховуються, як компоненти власних векторів відповідних матриць попарних порівнянь. Підсумковий результат отримується шляхом застосування WSM до отриманих локальних пріоритетів.

Сааті [147] визнав, що при використанні парних порівнянь може виникати небажаний ефект транзитивної неузгодженості в оцінках переваг одних об'єктів над іншими. Крім того, ОПР також може допускати неточності у визначенні кількісних характеристик переваг одного об'єкта над іншими з огляду на існуючі, встановлені на попередніх кроках порівняльного аналізу відношення переваг (кардинальна неузгодженість). Для усунення цих негативних наслідків Сааті запропонував використовувати коефіцієнт неузгодженості для оцінки транзитивності попарних порівнянь, який кількісно визначає узгодженість суджень ОПР. Значення цього коефіцієнту при вдалих

діях ОПР повинно знаходитись в межах $[0;0.1]$. В іншому випадку треба відкоригувати проведені попарні порівняння об'єктів.

АНР також був ретельно вивчений для випадків, коли додається нова альтернатива або видаляється існуюча при наявності заданих переваг. Іноді такі зміни можуть привести до порушення встановленого рейтингового порядку. Це відбувається через існуючі взаємозалежності при обчисленні локальних пріоритетів. Для подолання цієї проблеми Сааті запропонував метод ANP (Analytic network process) [146], який відрізняється від АНР тим, що використовує не ієрархічну, а мережеву структуру критеріїв і альтернатив. Проте цей метод є достатньо складним в застосуванні і не гарантовано усуває проблему порушення рейтингового порядку. Найбільш розумний підхід полягає в забезпеченні незмінності множин альтернатив і критеріїв на етапах порівняльного аналізу [147]. Цього можна досягти ретельністю в структуруванні проблеми процесу прийняття рішень.

Методи ідеальної точки оцінюють альтернативи на основі їх відхилення від ідеальних значень. Найбільш відомими з цих методів є TOPSIS (Technique for Order Preference by Similarity to Ideal Solutions) та VIKOR (VIseKriterijumska Optimizacija I Kompromisno Resenje) [125, 136]. TOPSIS був запропонований Хвангом і Юном [124, 125]. Принцип методу полягає в тому, що оптимальна альтернатива повинна мати найкоротшу відстань від найкращого ідеального рішення і найвіддаленішу відстань від найгіршого. Найкраще і найгірше ідеальні рішення досягаються на штучних альтернативах, які висуваються ОПР на основі гіпотетично найкращих і найгірших значень всіх критеріїв. TOPSIS підраховує результати, порівнюючи евклідові відстані між фактичними альтернативами та гіпотетичними.

VIKOR був незалежно розроблений Оприковичем у 1998 р. [136] і є аналогом методу TOPSIS. Відмінність між TOPSIS і VIKOR полягає у підходах до нормалізації значень критеріїв та обрахунках відстаней до ідеальних точок. TOPSIS для кожної альтернативи обраховує показник, що характеризує ступінь її віддаленості від найгіршої точки, а VIKOR – близькість до найкращої. Що

стосується нормалізації, TOPSIS використовує векторне нормування, тоді як VIKOR – лінійне [105, 135].

Аналогічно АНР і ANP, як TOPSIS так і VIKOR не гарантують незмінності рейтингового порядку, оскільки додавання чи видалення лише однієї альтернативи може змінити ідеальні значення критеріїв і, як наслідок – вплинути на результат. Проте, TOPSIS і VIKOR були успішно застосовані до ряду прикладних задач [97, 101, 105, 106, 150, 154].

Методи OUTRANKING є наслідком теорії, представленої французьким професором Бернардом Роем, який є автором сімейства методів ELECTRE.

ELECTRE (ELimination Et Choix Traduisant la REalité) українською мовою з французької буквально означає «Елімінація та вибір, що виражають реальність». На сьогоднішній день існують сім модифікацій цього методу – ELECTRE I, IS, Iv, II, III, IV і Tri.

ELECTRE I [141] – найпростіша форма ELECTRE. Метод використовує індекси конкордації і невідповідності, які розраховуються окремо для кожної пари альтернатив. Індекс конкордації виражає ступінь переваги певної альтернативи над іншою на основі кількості домінуючих критеріїв. Індекс невідповідності враховує кількість критеріїв, за яким альтернатива поступається іншій. Використовуючи порогові значення, встановлені ОПР, для кожної пари альтернатив можна визначити відношення, в якому вона перебуває – перевага (\succ), байдужість (\sim) або незрівнянність. На основі встановлених відношень будується множина альтернатив, які не домінують одна над одною. Серед них ОПР необхідно обрати найкращі.

ELECTRE IS [143] на відміну від ELECTRE I вводить поріг байдужості. Значення, які є нижчими від нього, вказують що пара альтернатив є рівнозначними для ОПР, тобто альтернативи не суттєво відрізняються одна від іншої.

ELECTRE IV [129] вводить поріг вето – значення, при якому ОПР остаточно віддає перевагу одній альтернативі в порівнянні з іншою, і прагне вибрати цю альтернативу з повною упевненістю.

ELECTRE II [144] був представлений Роєм і Берт'є в якості першої модифікації ELECTRE для забезпечення повного ранжування результатів. Щоб досягти цього, індекс конкордації був адаптований для двох рівнів відносин, сильних і слабких. Повний рейтинг розраховується за допомогою процедури дистиляції і двох рівнів ранжування. Кращою вважається альтернатива, яка переважає більшість інших альтернатив, а гіршою – альтернатива, яка переважає найменшу кількість інших.

ELECTRE III [142], на відміну від попередніх версій ELECTRE, для отримання індексів конкордації та невідповідності використовує псевдо-критерії, які дозволяють враховувати невизначену і обмежену інформацію [155]. Ранжування в ELECTRE III відбувається так само, як у ELECTRE II.

ELECTRE IV [145] був запропонований для спрощення процедури ELECTRE II. У всіх попередніх версіях ELECTRE ОПР повинен явно задавати ваги критеріїв. В ELECTRE IV система переваг визначаються за допомогою порогових значень.

ELECTRE Tri [101] є адаптацією ELECTRE III. В ньому було запропоновано класифікувати альтернативи, а не надавати їм рейтинг. ОПР встановлює категорії (групи або класи), які упорядковуються, як правило, в порядку від найгіршої до найкращої. Межі кожної групи визначаються пороговими значеннями, у порівнянні з якими формується відношення переваг і відбувається остаточно категоризація альтернатив.

Таким чином, методи сімейства ELECTRE можна розділити на три групи: *вибір* (I , IS , Iv), *ранжування* (II , III , IV) і *кластеризація* (Tri). З точки зору практичного застосування найбільш вживаними є група методів ранжування, серед яких можна виділити ELECTRE III як найбільш досконалий метод, що дозволяє враховувати невизначеність і надає ОПР контроль по встановленню ваг критеріїв [148].

Метод PROMETHEE (Preference Ranking Organization METHods for Enrichment Evaluation) базується на взаємному порівнянні кожної пари альтернатив за кожним з критеріїв. Як і ELECTRE, PROMETHEE налічує сімейство методів, найбільш вживаними з яких на практиці є PROMETHEE I, II [98].

В даних методах результат порівняння двох варіантів виражається в термінах переваг. Для цього вводиться до розгляду функція переваги, яка описує інтенсивність переваги однієї альтернативи над іншою. Як правило, розглядають шість можливих типів функцій переваги, які у відповідності до методу може обирати ОПР: звичайна (usual), квазі (quasi), V-подібна, багаторівнева, лінійна і функція Гауса (рис. 1.4) [109].

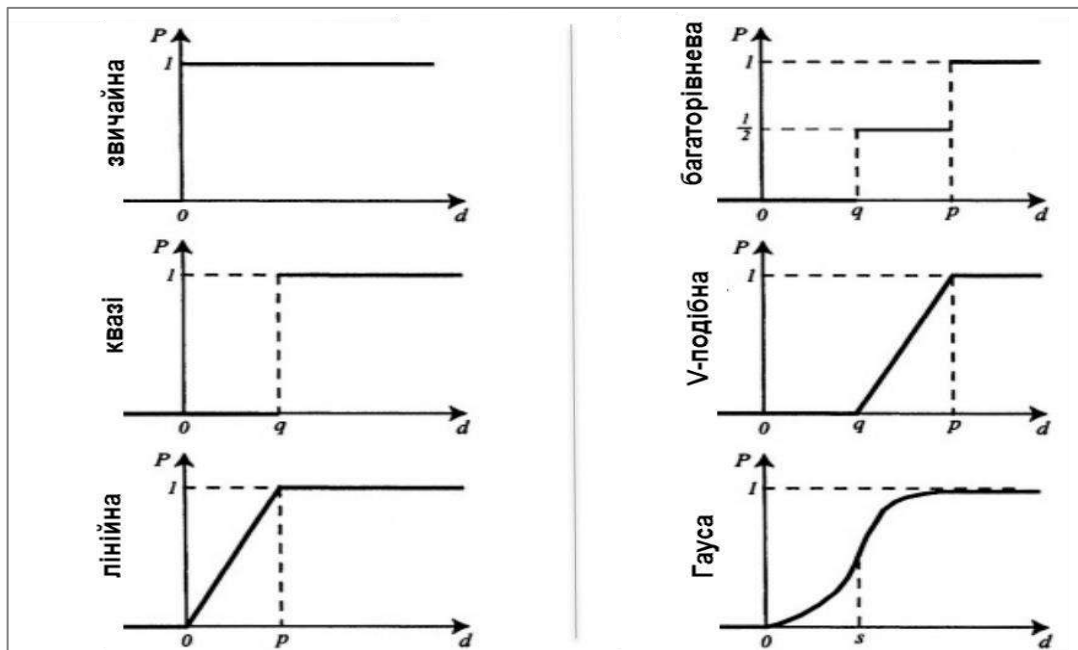


Рис. 1.4. Шість типів функцій переваг

При побудові функції переваги визначають декілька параметрів, які задають поріг строгої переваги, поріг байдужості та параметр, пов'язаний зі стандартним відхиленням нормального розподілу для функції Гауса.

Для кожної пари альтернатив обчислюється індекс переваги, як сума добутків вагових коефіцієнтів критеріїв на числові значення функцій переваг. Індекс переваги відображає ступінь переваги однієї альтернативи над іншою з урахуванням всіх критеріїв. Індeksi переваг утворюють матрицю індєксів, яку

можна представити у вигляді орієнтованого графа зі зваженими дугами. Між кожною парою вершин існують дві дуги з вагами прямої та зворотної переваги.

На наступному етапі методу для кожної вершини (альтернативи) обчислюється сума всіх ваг вихідних дуг – коефіцієнт переваги. Аналогічно обчислюється сума всіх ваг вхідних дуг – коефіцієнт зворотної переваги.

Обчислені коефіцієнти є інформаційною базою для ранжування альтернатив і використовуються по-різному в модифікаціях методу:

- в PROMETHEE I формуються бінарні відношення переваги, еквівалентності та незрівнянності між кожною парою альтернатив; множина встановлених бінарних відношень відображає частковий порядок альтернатив;
- в PROMETHEE II альтернативи впорядковуються за спаданням різниці коефіцієнтів прямої та зворотної переваги.

Слід зазначити, що модифікація 2 не завжди є кращою, так як ОПР більшу інформацію про переваги об'єктів отримує в результаті аналізу попарних відношень переваг. Так як цей метод є евристичним, то отримані результати являються додатковою інформацією для ОПР, якому для збільшення достовірності доцільніше оперувати з множиною бінарних відношень, ніж з агрегованими значеннями в модифікації PROMETHEE 2 [151].

У 1992 р. Дж.-П. Бранс разом з Б. Маршалем доповнили сімейство PROMETHEE методами PROMETHEE III (ранжування на основі інтервалів), PROMETHEE IV і V для повного або часткового ранжування, коли множина можливих рішень є неперервною. [99, 100].

1.2. Використання онтологій як основи представлення альтернатив

Онтологія [4, 6, 7, 11, 12, 18, 22, 32, 34, 35, 38, 40, 43, 44, 47, 50, 54, 57–65, 68, 70–73, 76, 84, 85, 88, 89, 116, 117, 120–122, 131, 132, 134, 137, 149] – спроба всеосяжної і детальної формалізації деякої області знань за допомогою концептуальної схеми. Зазвичай така схема складається з ієрархічної структури даних, що містить всі релевантні класи об'єктів, їх зв'язки і правила, прийняті в

цій галузі. Під концептуальною схемою мається на увазі набір понять (концептів) та інформація про поняття (властивості, відношення обмеження, аксіоми і твердження про поняття, що необхідні для опису процесів вирішення задач в обраній предметній області) [12, 14, 17, 36, 39, 51, 89, 153].

Онтологія є баченням експерта предметної області ПДО, вираженим в деякому формальному поданні, що дає можливість використання цього розуміння структури та значення елементів інформації при її автоматичній обробці. Саме формалізація подання зв'язків між поняттями в онтології робить можливим їх використання в широкому спектрі інформаційно-аналітичних систем (далі – ІАС). В якості онтологій, що різною мірою формалізовані, розглядаються [14, 15, 18, 58, 59, 61–63, 120–122]:

- глосарій – словник з визначеннями, який пояснює маловідомі слова, як правило, в межах однієї тематики;
- проста таксономія – тип керованого словника, що має ієрархічну структуру та описує поняття від простих до складні;
- тезаурус (таксономія з термінами) – словник в якому слова, які відносяться до певних областей знань, розташовані за тематичним принципом і показані семантичні відносини між лексичними одиницями;
 - модель з довільним набором відношень;
 - таксономія і довільний набір відношень;
 - повністю аксіоматизована теорія.

Особливе значення в онтології мають зв'язки між поняттями. За рахунок них онтологія являє собою не просто структуру понять, але і відображає складні відносини між ними та комплексно представляє предметну область. Існує три основних типи зв'язків між поняттями:

Таксономічні зв'язки – виражають відношення «є видом» або відношення «загальне / специфічне», тобто відносини, що ставлять дві сутності в залежність «загальне - часткове»;

Композиційні зв'язки – виражають відношення «є частиною»;

Топологічні зв'язки – відображають як різні складові системи пов'язані один з одним через певні зв'язки, або показують «шляхи» фізичних взаємодій між складовими, а так само забезпечують інформацією про просторове розташування цих складових.

Важливо підкреслити відмінність між фундаментальними онтологіями (fundamental ontologies), які описують предметну область максимально повно з максимальним ступенем формалізації, і прикладними онтологіями (application ontologies), які також називаються «легкими» онтологіями (lightweight ontologies) і які формалізуються настільки, наскільки це необхідно для програми [14, 15, 18, 58, 59, 61–63, 120–122].

Поняття онтології, призначеної для підтримки розв'язання задач інформаційного пошуку, повинно бути акуратно пов'язано зі значеннями термінів предметної області ПдО [14, 15, 58, 59, 61].

Досить багато уваги приділяється методиці створення онтологій в статті Н. Ноя і Д. МакГіннеса [131, 132]. Основоположні правила розробки онтології автори формулюють таким чином:

1) Не існує єдиного правильного способу моделювання предметної області – завжди існують життєздатні альтернативи.

2) Розробка онтології – це обов'язково ітеративний процес. Під ітеративним процесом розуміється неодноразовий прохід по онтології з метою її уточнення: тобто на початковому етапі будується «чорновий» варіант; потім треба перевірити і уточнити складену онтологію, і додаючи деталі, можливо частково або навіть повністю переглянути початкову онтологію.

3) Елементи онтології повинні бути близькі до об'єктів (фізичних або логічних) і відношень у визначеній предметній області ПдО.

При обмеженому розумінні онтологія сама по собі не може і не повинна вирішувати практично значущих інженерних задач. Це мають робити експертні, пошукові, Data Mining системи концептуального доступу до баз даних, інтелектуальні та проблемно-орієнтовані системи. Онтологія повинна лише забезпечувати їм підтримку, пропонуючи точну і програмно-інтерпретовану

відповідь на питання, що ми знаємо про дану проблемну область ПдО [3, 14, 15, 18, 20, 58, 59, 61–63].

У зв'язку з цим представляється можливим розглянути проблеми, пов'язані з побудовою та використанням онтологій [4, 6, 7, 11, 12, 18, 22, 32, 34, 35, 38, 40, 43, 44, 47, 50, 54, 57–65, 68, 70–73, 84, 85, 88, 89, 116, 117, 120–122, 131, 132, 134, 137, 149]:

1) Концептуальна структура одиниці – проблема вибору і рівня деталізації одиниць (понять), статусу службових одиниць, межа між поняттями і лексичними варіантами, розпізнавання одиниць в текстах.

2) Формальна модель – формалізований, за допомогою певної мови представлення знань, опис концептуальної системи, яка специфікує:

- використовувану класифікацію концептів;
- набір допустимих парадигматичних відношень між концептами;
- аксіоми і правила виводу.

3) Обчислювальна функціональність, яка визначається наступним:

- уніфікацією термінології;
- представленням і логічною обробкою таксономічних відношень;
- представленням і логічною обробкою відношень об'ємної сумісності/несумісності;
- представленням і логічною обробкою предметно-асоціативних відношень;
- підтримкою представлення і логічної обробки кількісних даних;
- регламентацією процедур опису об'єктів;
- аксіоматизацією описів процесів, причинних зв'язків і процедур.

Існує проблема створення універсальної онтології – усвідомлений рух до універсальної онтології представляється більш привабливим, ніж розробка та підтримка витончених засобів встановлення відповідностей між незалежними онтологіями, що розробляються. До якогось моменту, очевидно, обидва напрями можуть і повинні співіснувати [81].

Можна сформулювати основні вимоги до побудови онтологій:

1) гнучкість – можливість швидкого і простого оновлення будь-якого з фрагментів онтології, можливість організації децентралізованого «багатоагентного» створення і редагування онтологій;

2) відкритість – для додавання як окремих концептів будь-якого вмісту, так і будь-яких концептуальних підсистем, відкритість для лексики природних мов і додаткових варіантів концептуальної інтерпретації слів, що вже містяться в лексиконі онтології;

3) змістовна масштабованість – можливість оперативно вибирати (підключати/відключати) ті чи інші фрагменти відповідно до задачі, сфери інтересів і точки зору окремих професійних груп;

4) модельна масштабованість – можливість подання концептуальних систем на різних рівнях детальності опису та формалізації відповідних фрагментів дійсності (наприклад, в такій послідовності: проста семантична категоризація лексики – таксономія – повна термінологічна модель – продукційна система – логічна теорія);

5) користувальницька універсальність – придатність для використання в різнотипних додатках;

Враховуючи, що онтологія – це інформаційно-обчислювальний ресурс, який якісно на основі знань експерта представляє предметну область і може бути використаний будь-якими інтелектуальним інформаційним технологіям, зокрема, як джерело інформації. То з неї можуть бути виокремлені об'єкти (альтернативи) та їх властивості для безпосереднього розв'язання багатокритеріальної задачі ранжування.

1.3. Існуючі MCDA системи

Для застосування MCDA методів на практиці було розроблено різноманітне програмне забезпечення, яке отримало назву multi-criteria decision analytical software (MCDA системи). Окрім ранжування об'єктів за сукупністю показників у відповідності до різних методів, програмне забезпечення надає ОПР можливості побудови моделі задачі та аналізу отриманих результатів.

Зокрема, графічні користувацькі інтерфейси систем можуть забезпечувати візуалізацію процесу отримання рішення, а відтак, зробити роботу MCDA системи більш прозорою.

1.3.1. Огляд існуючих MCDA систем

Існує багато різних MCDA систем, як веб-орієнтованих, що доступні безпосередньо для використання в інтернеті у режимі онлайн, так і десктопних версій, які можна завантажити та встановити. Отже цілком доцільним буде не аналізувати кожен окремий програмний продукт, а зосередити увагу переважно на тому програмному забезпеченні, яке активно використовувалося або яке досягло певного статусу серед практиків та спільноти MCDA, що можна розглядати як один із індикаторів корисності програмного забезпечення.

Досліджувалось тільки те програмне забезпечення, яке відноситься до багатоатрибутного прийняття рішень [112]. Таким чином, програмне забезпечення, що не входить в цю категорію, зокрема, наприклад, багатоцільове прийняття рішень (multiple objective decision making) не розглядалось.

Деяке програмне забезпечення в огляді насправді не є єдиним програмним продуктом, а більше схожим на бібліотеку ресурсів. Наприклад, Decision Desk є відкритою колекцією програмного забезпечення, яке в даний час включає в себе кілька безкоштовних модулів підтримки прийняття рішень.

Метою дослідження існуючих MCDA систем був порівняльний аналіз найбільш потужного та варіативного програмного забезпечення, що застосовується в прикладних задачах багатокритеріального прийняття рішень в більшості сфер людської діяльності. Багато таких систем дублюють спільний функціонал і є схожими аналогами.

Таким чином, було обрано 20 програмних засобів та ресурсів для проведення їх аналізу, в основному на базі деякої демонстраційної або пробної версії програмного забезпечення.

Далі наводиться перелік обраного програмного забезпечення з їх коротким описом:

- 1000Minds – Веб-програмне забезпечення з інтерфейсом на базі закладок. Застосунок надає можливість визначення переваг за допомогою попарних запитань (pairwise questions) за критеріями та має у своєму розпорядженні різні способи аналізу результатів. Крім цього, тут існує можливість поширення результатів в глобальній мережі Інтернет та надання користувачам інтерактивних засобів для голосування або опитування. 1000Minds може застосовуватись для розподілення бюджетів або інші обмежені ресурсів. Система застосовує метод PAPRIKA (Potentially all pairwise rankings of all possible alternatives).

- Analytica – має об'єктно-орієнтований візуальний інтерфейс, в який можна імплементувати практично будь-який метод. Присутні можливості побудови різних графових структур. Наявні попередньо визначені модулі. Наприклад, для MAUT, оптимізації та аналізу ризиків. Доступні різні дистрибутиви. Особливості: об'єктно-орієнтований інтерфейс та можливості візуального створення графів.

- Criterium Decision Plus 3.0 – Базове MAVT програмне забезпечення з функціоналом методу аналізу ієрархій (АНР). До його особливостей можна віднести можливість проведення мозкового штурму та представлення критеріїв у вигляді графів.

- DecideIT – MCDA програмне забезпечення, що забезпечує як оцінювання, так і побудову дерев рішень. В цій системі застосовується ймовірнісний аналіз неточних результатів. До особливостей слід віднести моделювання невизначеностей з інтервалами.

- Decision Tools – набір надбудов для Excel, які в себе включають: аналіз ризиків на основі методу симуляції Монте-Карло, дерева рішень та діаграми впливу, прогностичний аналіз на основі нейронних мереж та інше. До особливостей цих модулів можна віднести використання різних розподілів для врахування невизначеності за допомогою методів симуляції. Також варто відмітити застосування Excel-орієнтованого інтерфейсу, що надає гнучкі можливості для конструювання моделі задачі.

- D-Sight – MCDA програмне забезпечення, що базується на методах MAUT та PROMETHEE. Доступні веб- та десктопні версії. Інтерфейс реалізований на базі закладок.
- GMAA – MAUT програмне забезпечення з можливістю використання інтервалів для моделювання нечіткостей.
- Hiview 3 – MCDA програмне забезпечення в якому, використовується метод MACBETH. Система володіє можливістю створення графів та візуального аналізу на основі ху-графіків.
- Logical Decisions – базове MAVT програмне забезпечення з АНР функціональністю.
- M-MACBETH – MAVT програмне забезпечення, яке підтримує метод Macbeth, та включає різні графічні способи оцінки параметрів. До особливостей можна віднести використання різноманітних графіків.
- MakeItRational – АНР система з типовим інтерфейсом на основі закладок. У системі присутня модель групового прийняття рішень.
- MESTA – Програмне забезпечення, засноване на встановленні порогів до критеріїв та аналізі, які альтернативи їм відповідають.
- OnBalance – програмне забезпечення на основі MAVT методів
- Promax – базове MAVT програмне забезпечення з Office подібним інтерфейсом на основі закладок.
- TESLA – програмне забезпечення з застосуванням дерев рішень та використанням досвіду, що формується на ймовірнісних підходах.
- The Decision Deck project – програмне забезпечення з відкритим кодом. Включає в даний час кілька безкоштовних модулів підтримки прийняття рішень.
- V.I.P. Analysis – MAVT програмне забезпечення, яке надає можливість накладати обмеження на вагові коефіцієнти критеріїв. В цій системі присутній аналіз чутливості результатів.

- V.I.S.A. Decisions – базове MAVT програмне забезпечення, для якого наявні веб- та десктопна версії.
- Web-NIPRE – веб-орієнтоване програмне забезпечення для структурування проблеми, багатокритеріального оцінювання та визначення пріоритетів.
- WINPRE – MAVT програмне забезпечення з можливістю використання інтервальних значень критеріїв в моделі задачі. Дозволяє враховувати умови невизначеності.

1.3.2. Порівняння існуючих MCDA систем

Критерії для порівняння MCDA поділялись на категорії у відповідності до основних етапів процесу проведення вибору та ранжування (побудова моделі, оцінка критеріїв, встановлення результатів). Для кожної категорії зафіксований рівень застосовності програмного забезпечення для різних завдань та/або методів цього етапу. Наприклад, на етапі зважування критеріїв перераховано, які методи MCDA підтримуються програмним забезпеченням.

Досліджувались такі категорії: підтримка процесу, конструкція моделі задачі, прикладні методи, аналіз результатів.

Як правило, MCDA аналіз здійснюється на основі одного ОПР чи групи зацікавлених сторін одночасно. У випадку з кількома зацікавленими сторонами, група може визначити загальну оцінку, що представляє певну середню або типову думку кожного представника групи. Проте найчастіше більш доречно дозволяти кожному члену групи надавати власні уподобання. Деякі програмні засоби також надають підтримку для групової роботи.

На практиці, використання методів MCDA вимагає певного досвіду від особи, яка бере участь у процесі побудови моделі та розв'язку задачі, оскільки може виникнути «необ'єктивність» у разі неналежного використання цих методів. Як правило, програмне забезпечення MCDA розробляється для досвідчених ОПР, тому коректне використання цих методів цілком залежить

від користувача. Порівняльний аналіз за категорією підтримки процесу наведений в табл. 1.1.

Таблиця 1.1. Порівняльний аналіз MCDA систем за категорією підтримки процесу

№	Програмне забезпечення	Програмне забезпечення загального призначення	Підтримка процесу	Онлайн підтримка
1	1000Minds	+	+	+
2	Analytica	+	-	-
3	Criterion Decision Plus 3.0	+	-	-
4	DecideIT	+	-	-
5	Decision Tools	+	-	-
6	D-Sight	+	+	-
7	GMAA	+	-	-
8	Hiview 3	+	-	-
9	Logical Decisions	+	-	-
10	M-MACBETH	+	-	-
11	MakeItRational	+	+	-
12	MCDA-Res	+	+	+
13	OnBalance	+	-	-
14	Promax	+	+	-
15	TESLA	+	-	-
16	The Decision Deck project	~	-	-
17	V.I.P. Analysis	+	-	-
18	V.I.S.A. Decisions	+	+	+
19	Web-HIPRE	+	-	-
20	WINPRE	+	-	-

З точки зору побудови моделі, програмне забезпечення, як правило, дуже подібне одне одному. Наприклад, наявність можливості представлення критеріїв ієрархічними структурами.

Практично все програмне забезпечення для введення критеріальних значень альтернатив надає матрично-подібну таблицю. Багато MCDA систем надає візуальні способи введення даних. Порівняльний аналіз за категорією модель задачі наведений в табл. 1.2.

Тільки декілька з усіх програмних засобів містять певні елементи онтологічного супроводу, що здебільшого пов'язано з можливістю візуального

формування таксономічної структури. Наприклад, у системі «Criterium Decision Plus» існує режим мозкового штурму, в якому ОПР може визначати предметну область за допомогою графової структури. Це дозволяє встановлювати елементи проблемної задачі та відношення між ними.

Однак повноцінний онтологічний супровід полягає в застосуванні вже створеної експертом предметної області онтологічної моделі для формування та розв'язання задачі ранжування. Таким чином, знання експерта можуть бути ефективно використані.

Таблиця 1.2. Порівняльний аналіз MCDA систем за категорією моделі задачі

№	Програмне забезпечення	Ієрархічна модель критеріїв	Таблиця значень для альтернатив	Візуальне встановлення оцінок	Елементи онтологічного супроводу (0..10)
1	1000Minds	-	+	-	-
2	Analytica	+	+	+	+
3	Criterium Decision Plus 3.0	+	+	+	+
4	DecideIT	+	-	+	-
5	Decision Tools	+	+	-	-
6	D-Sight	+	+	-	-
7	GMAA	+	+	+	+
8	Hiview 3	+	-	+	-
9	Logical Decisions	+	+	+	-
10	M-MACBETH	+	+	+	-
11	MakeItRational	+	+	+	-
12	MCDA-Res	~	~	~	-
13	OnBalance	+	+	+	-
14	Promax	+	+	+	+
15	TESLA	+	-	+	-
16	The Decision Deck project	~	~	~	~
17	V.I.P. Analysis	-	+	+	-
18	V.I.S.A. Decisions	+	+	+	+
19	Web-HIPRE	+	+	+	-
20	WINPRE	+	+	+	-

Багато MCDA систем розробляються комерційними структурами, які орієнтуються на створення універсальних програмних продуктів. Оскільки, крім того, що вони є загальними з точки зору області застосування, багато з них

можна також розглядати як програмне забезпечення, яке включає різноманітні багатокритеріальні методи із різних розділів MCDA.

Наприклад, майже половина проаналізованих MCDA систем забезпечує підтримку, як АНР (або іншого методу на основі попарних порівнянь), так і MAVT/MAUT методи. Також наявне програмне забезпечення, яке надає підтримку методів Outranking.

За незначним виключенням, все проаналізоване програмне забезпечення надає користувачеві різні способи візуалізації встановлених переваг об'єктів моделі. Порівняльний аналіз за категорією модель задачі наведений в табл. 1.3.

*Таблиця 1.3. Порівняльний аналіз MCDA систем
за категорією прикладні методи*

№	Програмне забезпечення	Візуальне зважування	АНР / Попарне порівняння	MAUT/ MAVT	Outranking	Моделювання невизначеності	дерева рішень
1	1000Minds	-	+	+	-	-	-
2	Analytica	+	-	+	-	+	-
3	Criterion Decision Plus 3.0	+	+	+	-	-	-
4	DecideIT	+	+	+	-	+	+
5	Decision Tools	-	-	-	-	+	+
6	D-Sight	+	-	+	+	-	-
7	GMAA	+	-	+	-	+	-
8	Hiview 3	+	-	+	-	-	-
9	Logical Decisions	+	+	+	-	-	-
10	M-MACBETH	+	+	+	+	-	-
11	MakeItRational	+	+	+	-	-	-
12	MCDA-Res	~	~	~	+	~	~
13	OnBalance	+	-	+	-	-	-
14	Promax	+	+	+	-	-	-
15	TESLA	+	-	-	-	+	+
16	The Decision Deck project	~	~	~	~	~	~
17	V.I.P. Analysis	+	-	+	-	+	-
18	V.I.S.A. Decisions	+	-	+	-	-	-
19	Web-HIPRE	+	+	+	-	-	-
20	WINPRE	+	+	+	-	+	-

Всі MCDA програмні засоби на етапі аналізу результатів (пост-аналізу) надають принаймні якісь візуальні засоби для представлення результатів. Найбільш поширеним підходом є розбиття стовпчиків діаграми на фрагменти, довжини яких представляють рівень впливу різних критеріїв на загальні результати. У дослідженому програмному забезпеченні використовуються також й інші способи візуалізації розв'язків – павутиноподібні діаграми профілю альтернатив, діаграми торнадо, кругові діаграми, ху-графіки та інше.

Аналіз чутливості відбувається найпоширенішим способом – досліджується вплив показника, що параметризується (при фіксованих інших), на вигляд розв'язку задачі. Крім цього, в декількох системах забезпечуються певні статистичні підходи. Вони базуються на застосуванні різних розподілів за параметрами моделі та проведенням її симуляції за ними. Як результат, можна отримати, наприклад, ймовірнісні рейтинги альтернатив або відсотки від того, як часто одна альтернатива домінує на іншою.

Певні програмні засоби надають можливість створювати письмові звіти, в яких з роз'ясненнями представлені основні результати.

В табл. 1.4. наведений порівняльний аналіз за категорією аналіз результатів задачі.

Для проведення більш ґрунтовного аналізу альтернатив на етапі пост-аналізу можна розглянути обернену задачу ранжування – при яких мінімальних змінах критеріальних значень можна покращити рейтинг (ранг) певної обраної альтернативи.

Багато програмного забезпечення надає можливість імпортувати та / або експортувати дані та результати, наприклад, в звичайний текст або в Excel. Крім того, багато програмних продуктів мають "Excel-like" інтерфейси для введення даних.

*Таблиця 1.4. Порівняльний аналіз MCDA систем за категорією
аналіз результатів*

№	Програмне забезпечення	Візуальні графіки	Загальні значення	Аналіз чутливості	х-у графіки	Письмовий звіт	Обернена задача
1	1000Minds	+	+	+	+	+	-
2	Analytica	+	+	+	+	-	-
3	Criterion Decision Plus 3.0	+	+	+	+	-	-
4	DecideIT	+	+	+	+	-	-
5	Decision Tools	+	-	+	+	+	-
6	D-Sight	+	+	+	+	-	-
7	GMAA	+	+	+	+	-	-
8	Hiview 3	+	+	+	+	+	-
9	Logical Decisions	+	+	+	+	+	-
10	M-MACBETH	+	+	+	+	-	-
11	MakeItRational	+	+	+	+	+	-
12	MCDA-Res	~	~	~	~	~	-
13	OnBalance	+	+	+	+	-	-
14	Promax	+	+	+	+	-	-
15	TESLA	+	+	+	+	+	-
16	The Decision Deck project	~	~	~	~	~	-
17	V.I.P. Analysis	+	+	+	+	-	-
18	V.I.S.A. Decisions	+	+	+	+	+	-
19	Web-HIPRE	+	+	+	-	-	-
20	WINPRE	+	+	+	-	-	-

Загалом, проаналізовані програмні засоби багато в чому схожі за своєю функціональністю та підтримують всі етапи MCDA процесу – структуризація проблеми, визначення критеріїв, альтернатив, отримання результатів та їх аналіз на чутливість. Це цілком очікувано, оскільки протягом останніх десятиліть загальна схема процесу MCDA фактично була стандартизована і ретельно досліджена.

На сьогоднішній день можна констатувати, що загальною тенденцією стало створення багатофункціонального MCDA програмного забезпечення, в якому реалізовані різні методи, що використовуються в залежності від специфіки конкретної задачі ранжування.

Висновки за розділом 1

У першому розділі проведено огляд літератури за темою дисертації, проаналізовані процес розв'язання задач ранжування та MCDA методи, що можуть бути застосовані в дискретному випадку.

Процес проведення ранжування та вибору складається з наступних етапів: визначення проблемної задачі, структурування проблеми, реалізація оптимального вибору, пост-аналіз та отримання результату. Помилки на етапі структуризації призводять до утворення хибної моделі задачі, яка швидше за все, призведе до неточних результатів. Отже, якість отриманого розв'язку задачі в першу чергу залежить від вдалого структурування, що вимагає від ОПР скрупульозної деталізації проблемної області для визначення критеріїв, альтернатив та іншої інформації. Саме цей етап може бути ефективно здійснений на основі онтологічного підходу, оскільки онтологія – це детальний опис предметної області за допомогою концептуальної схеми.

Для додаткового дослідження об'єктів, які не стали «переможцями», але з огляду на специфіку конкретної задачі можуть представляти інтерес для ОПР, запропоновано розглянути обернену задачу ранжування.

Таким чином, онтологічна модель може бути цінним джерелом надходження інформаційного ресурсу на всіх етапах процесу прийняття рішень.

Досліджено існуючі MCDA програмні засоби, що можуть використовуватись у процесі розв'язання задачі ранжування альтернатив. У результаті проведеного аналізу було виявлено, що на теперішній час не існує достатніх технологічних засобів для онтологічного супроводу роботи MCDA-систем.

Результати першого розділу дали змогу сформулювати раніше невизначену задачу створення технологічних засобів для онтологічного супроводу процесів ранжування альтернатив за сукупністю показників та сформулювати основні завдання дисертаційного дослідження.

Основні результати розділу опубліковано в [22, 24, 80].

РОЗДІЛ 2. МАТЕМАТИЧНІ МОДЕЛІ ТА АЛГОРИТМИ РОЗВ'ЯЗКУ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ

2.1. Онтологічне представлення задачі ранжування альтернатив

Кожна онтологія містить інформаційні описи, на основі об'єктно-орієнтованої процедури формалізації, а також описи інтерпретаційних функцій, які є функціональним проявом властивостей об'єктів (концептів), що складають онтологію, та які управляють на основі цього процесом поставки інформаційного ресурсу на усіх етапах прийняття рішень. Тому цілком виправданим є представлення інформаційної моделі в середовищі СППР у вигляді певної онтології [15, 42, 59, 120, 121].

Як відомо, в основі онтологічної методології лежить об'єктно-орієнтований підхід, при якому предметна прикладна область представляється у вигляді сукупності об'єктів, взаємодія між якими може бути представлена за допомогою семантичного зв'язування висловлювань, тверджень та суджень [8].

Під об'єктом розуміється деяка сутність (реальна або абстрактна) з притаманними їй станом, поведінкою і індивідуальністю.

- Стан об'єкта характеризується переліком всіх його можливих властивостей – структурою і значеннями кожної з цих властивостей.
- Поведінка об'єкта (або його функціональність) характеризує те, як об'єкт взаємодіє з іншими об'єктами або піддається впливу інших об'єктів, проявляючи свою індивідуальність. Поведінка об'єкта реалізується у вигляді функцій, які називають методами. При цьому структура об'єкта доступна тільки через його методи, які в сукупності формують інтерфейс об'єкта.
- Індивідуальність об'єкта характеризують такі властивості, які відрізняють його від всіх інших об'єктів.

Для формування адекватного операційного середовища СППР особливий інтерес представляють два типи ієрархічних залежностей між об'єктами:

- зв'язки – позначають рівноправні відношення між об'єктами; об'єкт співробітничав з іншими об'єктами через зв'язки, що з'єднують його з ними;

- агрегація – описує відношення цілого і частини, що наводять до відповідної таксономії (ієрархії об'єктів).

Комп'ютерну онтологію певної предметної області (ПрО) будемо розглядати як певну непусту множину об'єктів які задовольняють наступним вимогам:

- 1) об'єкти організовані у вигляді ієрархічної структури скінченної множини понять, що описують задану предметну область;
- 2) структура може бути представлена множиною дводольних графів, вершинами якого є поняття, а дугами – семантичні відношення між ними;
- 3) поняття і відношення інтерпретуються відповідно до загальнозначущих функцій інтерпретації, взятих з електронних джерел знань заданої ПрО;
- 4) визначення понять і відношень виконується на основі аксіом і обмежень їх області дії;
- 5) функції інтерпретації та аксіоми описані мовою формальної теорії.

В загальному випадку онтологія предметної області формально представляється впорядкованою трійкою [19, 59, 77, 81, 120]:

$$O = \langle X, R, F \rangle, \quad (2.1)$$

де X – множина концептів (понять, термінів) предметної області (ПрО), на основі яких формується предметна складова операційного середовища;

R – множина відношень та властивостей між ними (будемо вважати, що властивості є інтерпретацією відношень, тобто існує перетворення, яке кожному відношенню встановлює відповідність певної властивості);

F – функції інтерпретації (визначень) X та/або R , які складають функціональну частину операційного середовища.

Розгляд граничних випадків множин виразу (2.1): $R = \emptyset$; $R \neq \emptyset$; $F = \emptyset$; $F \neq \emptyset$ у всіх чотирьох комбінаціях значень R і F дає різні варіанти онтологічних конструкцій, починаючи від простого словника і таксономії до формальної структури концептуальної бази знань. За процедурою побудови

онтології на основі її певної функціональної повноти і ступені формальності виділяються наступні види онтологій: первинна, кінцева і множина проміжних онтологій.

Тематична онтологія [81] ($R \neq \emptyset; F \neq \emptyset$) – це така онтологія, в якій множини концептів та концептуальних відношень максимально повні, а до функцій інтерпретації додаються аксіоми, визначення та обмеження за тематикою даної ПрО. При цьому опис всіх компонент представлений деякою формальною мовою, яка може інтерпретуватися деякою процедурою (алгоритмом). Схема формальної моделі тематичної онтології описується наступним чином:

$$O = \langle X, R, F, \tilde{A}, D, R_s \rangle \quad (2.2)$$

де X – множина концептів, $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$, ($i = \overline{1, n}, n = \text{Card } X$), кінцева множина концептів (понять) заданої в операційному середовищі СППР;

$R = \{R_1, R_2, \dots, R_k, \dots, R_m\}, R_k \subseteq X \times X, k = \overline{1, m}, m = \text{Card } R$, – кінцева множина семантично значущих властивостей (відношень) між концептами ПрО. Вони визначають тип взаємодії між поняттями.

F – скінченна множина функцій інтерпретації, заданих на концептах і/або відношеннях.

\tilde{A} – скінченна множина аксіом, які використовуються для запису завжди істинних висловлювань (визначень і обмежень) в термінах тематики ПрО;

D – множина додаткових визначень концептів (понять) в термінах тематики ПрО;

R_s – множина обмежень, що визначає область дії понятійних структур визначеної тематики ПрО.

R^+ – множина властивостей, які характеризують концепти ПрО ($R^+ \subseteq R_s$).

Однією з системних компонент онтологічної системи є таксономія [81], що відображає певну ієрархію взаємодії концептів. При цьому власне ієрархія

задається за допомогою бінарних відношень, що визначають характер взаємодії між концептами онтології.

Таксономія – непуста підмножина \tilde{T} множини концептів X онтології O над якими задано відношення упорядкованості $R_t, R_t \subset R | R_t \neq \emptyset$. Якщо для множини концептів X задано множинне бінарне слабе відношення упорядкованості – $\tilde{\rho}$, то формально таксономію можна визначити наступним чином:

$$T = \{t \in X | \exists y (y \in X) : x\tilde{\rho}y \vee y\tilde{\rho}x\} \quad (2.3)$$

де $\tilde{\rho}$ ($\tilde{\rho} \in R_t \subseteq R$) бінарне відношення, що визначає частковий порядок для множини T .

Для формування моделі задачі ранжування на основі онтології виникає необхідність у визначенні певного перетворення $\tilde{\mathcal{H}}_{MR}$.

$$\tilde{\mathcal{H}}_{MR} : \langle X, R, F, \tilde{\mathcal{A}}, D, R_s \rangle \rightarrow M_R \quad (2.4)$$

де M_R – модель задачі ранжування.

Задача ранжування описується набором альтернатив A , для кожної з яких задаються значення певних показників (критеріїв). Розв'язком такої задачі вважається встановлення лінійного порядку над множиною A . Такий порядок, зокрема, дозволяє визначити альтернативи(y), що мають найкращі (за сукупністю) значення критеріїв, які в загальному випадку відрізняються різною важливістю. Формально математична модель задачі ранжування складається з наступних об'єктів:

$$M_R = \langle A, K, \tilde{F}, G \rangle \quad (2.5)$$

де A – множина альтернатив

K – множина критеріїв

$\tilde{F} : A \times K \rightarrow Q$ – функція, що визначає значення альтернатив за певним критерієм, Q – множина значень критеріїв.

G – правило ранжування, яке дозволяє встановити лінійний порядок для множини альтернатив

В залежності від ступеня сформованості моделі задачі ранжування можна виділити наступні випадки:

1. $M_R^{1A}(A \neq \emptyset, K = \emptyset, \tilde{F} = 0, G = 0)$ – визначені тільки альтернативи.
2. $M_R^{1K}(A = \emptyset, K \neq \emptyset, \tilde{F} = 0, G = 0)$ – визначені тільки критерії.
3. $M_R^2(A \neq \emptyset, K \neq \emptyset, \tilde{F} = 0, G = 0)$ – визначено множини альтернатив і критеріїв.
4. $M_R^3(A \neq \emptyset, K \neq \emptyset, \tilde{F} \neq \emptyset, G = \emptyset)$ – для альтернатив встановлені значення за критеріями. При такій моделі вважається, що задача ранжування сформована.
5. $M_R^4(A \neq \emptyset, K \neq \emptyset, \tilde{F} \neq \emptyset, G \neq \emptyset)$ – задано правило ранжування для сформованої моделі ранжування.

Фактично механізм формування моделі задачі ранжування та її розв'язання на основі онтології (2.4) є багатоетапним процесом, кожен з яких потребує використання окремих процедур. Загальну схему виконання такого перетворення можна представити таким процесом:

$$O \rightarrow M_R^{1A} \Leftrightarrow M_R^2 \rightarrow M_R^3 \rightarrow M_R^4 \rightarrow O^* \quad (2.6)$$

де O^* – розширення онтології O в результаті здійснення ранжування, що для концептів онтології (альтернатив) $A \subseteq X$ визначає нові відношення переваг та властивості. Крім того, слід зазначити, що нові властивості можуть бути встановлені і в наслідок пост-аналізу на основі розв'язку оберненої задачі ранжування.

Отже, перший етап формування моделі задачі ранжування полягає в виокремленні з множини концептів X елементів, які можуть розглядатись, як альтернативи. Альтернативами можна вважати однорідні об'єкти, що характеризуються спільними властивостями. Такий відбір може здійснюватись на основі функції вибору

$$F_{sel}(X) = Y, Y \subseteq X \quad (2.7)$$

До класично-раціональних функцій вибору відносяться ті, які мають наступні характеристичні властивості [1, 82]:

1. Умова наслідування (H):

$$X' \subseteq X \Rightarrow F_{sel}(X') \supseteq F_{sel}(X) \cap X' \quad (2.8)$$

2. Умова константності (K), строгого наслідування:

$$(X' \subseteq X) \wedge (X' \cap F_{sel}(X) \neq \emptyset) \Rightarrow F_{sel}(X') = F_{sel}(X) \cap X' \quad (2.9)$$

3. Умова згоди (З):

$$X' \cup X'' = X \Rightarrow F_{sel}(X) \supseteq F_{sel}(X') \cap F_{sel}(X'') \quad (2.10)$$

4. Умова незалежності або відкидання (B):

$$F_{sel}(X) \subseteq X' \subseteq X \Rightarrow F_{sel}(X') = F_{sel}(X) \quad (2.11)$$

Для побудови класу механізмів вибору, що породжують характеристичну область відкидання (B) у просторі функцій застосовується тип структур – гіпервідношення виду $Y\gamma Z$, що зв'язують пари множин [1, 2].

Тоді правило вибору на такій структурі визначається виразом:

$$F_{sel}(X) = Y, Y \subseteq X \Leftrightarrow Y\gamma Z, Z \subseteq X \quad (2.12)$$

Відмітимо, що якщо гіпервідношення γ володіє властивістю гіпертранзитивності, то функція вибору також належить до характеристичної області наслідування (H).

Умова гіпертранзитивності:

$$Y\gamma x, x \in X, X\gamma z \Rightarrow ((Y \cup X)/\{x\})\gamma z \quad (2.13)$$

Визначимо механізми вибору альтернативних варіантів для онтології (2.2) з таксономічною структурою \check{T} , над якою встановлене бінарне відношення слабкої упорядкованості $\check{\rho}$. Частковий порядок $\check{\rho}$ може бути розширеним до відношення транзитивного замикання ρ^+ .

$$x_i^j \check{\rho} x_l^k \xrightarrow{\alpha} x_i^j \rho^+ x_l^k \quad (2.14)$$

Тоді можна визначити функцію вибору на основі гіпервідношення γ_1^+ та γ_1^- між множинами $Y \subseteq \check{T}$ та $Z \subseteq \check{T}$ такі що:

$$Y\gamma_1^- Z, Y = \{y \in \check{T} | \exists z(z \in Z): z\rho^+ y\} .$$

$$Y\gamma_1^+Z, Y = \{y \in Y^* \mid \nexists x(x \in X, x \neq y): y\tilde{\rho}x\}, Y^*\gamma_1^-Z \quad .$$

$$F_{sel}(\check{T}) = Y, Y = \{y \in Y_1/Y_2\}, Y_1\gamma_1^+Z_1, Y_2\gamma_1^-Z_2, Z_1 \subseteq \check{T}, Z_2 \subseteq \check{T} \quad (2.15)$$

Описана функція вибору визначає кінцеві елементи таксономії, що є нащадками класів (категорій), які входять у множину Z_1 та відповідно не є нащадками класів заданих множиною Z_2 . Елементи, які належать до спільних класів (суперкласів) можна розглядати в якості альтернатив, оскільки – це визначає їхню однорідність. Таким чином описана функція для направленої графу у вигляді якого може бути представлена таксономічна структура визначає тисячі вершини, що мають спільних предків Z_1 . Якщо множини Z_1 та Z_2 не визначені, то їх слід задати наступним чином: $Z_1 = \{z \in X \mid \nexists x(x \neq z): x\tilde{\rho}z\}, Z_2 = \emptyset$.

Нехай R^+ – множина всіх властивостей (атрибутів), якими характеризуються об'єкти онтології. За допомогою гіпервідношення γ_2 можна задати множину властивостей для певного об'єкта $P_x = \{P \subseteq R^+ \mid P\gamma_2x\}$. Аналогічно, як і для функції вибору в попередньому випадку можна задати відповідні гіпервідношення γ_3^- та γ_3^+ залишивши правило вибору (2.15):

$$Y\gamma_3^+Z, Y = \{y \in \check{T} \mid \forall u: (Z \subseteq P_y)\} \quad .$$

Гіпервідношення γ_3^+ можна модифікувати до $\tilde{\gamma}_3^+$ увівши порогове значення q .

$$\begin{aligned} Y\tilde{\gamma}_3^+Z, Y &= \{y \in \check{T} \mid (|Z \cap P_y| > q)\} \\ Y\gamma_3^-Z, Y &= \{y \in \check{T} \mid \exists z(z \in Z): z \in P_y\} \end{aligned} \quad (2.16)$$

Механізм вибору, який породжується цим гіпервідношенням, говорить про те, що об'єкти, які розглядаються, мають не обов'язково володіти всіма атрибутами із бажаного списку. Нескладно бачити, що механізми вибору (2.15) та (2.16) є еквівалентними, оскільки приналежність до певного класу можна вважати властивістю об'єкта. Також для елементів, що характеризуються певними властивостями, можна побудувати відповідну таксономію, де на нижньому рівні ієрархії будуть об'єкти, а на верхніх – ієрархічна структура атрибутів.

Описані механізми вибору можуть налаштовуватись за допомогою певних параметрів. Якщо такі параметри не будуть задані, то в якості

альтернатив будуть відібрані всі елементи, що мають тип «об'єкт» для (2.16) та елементи, що знаходяться на нижньому рівні ієрархії для (2.15). Відмітимо, що для вдало структурованих таксономій відсутність параметрів не є проблемою.

Відзначимо, що головною рисою альтернатив є їхня однорідність. Розглянемо гіпердомінатний механізм вибору [2, 49] оптимальних комплектів на основі гіпервідношення γ_4 заданого за допомогою скалярної функції $\Phi(Z)$ (гіпершкала): $Y\gamma_4Z \Leftrightarrow \{\Phi(Y) \geq \Phi(Z)\}$.

Тоді функція вибору буде мати наступний вигляд:

$$F_{sel}(\check{T}) = Y, \quad \Phi(Y) = \max_{Z \subseteq \check{T}} \Phi(Z) \quad (2.17)$$

$$\Phi(Z) = GH_1 \left(|Z|, \left| \bigcap_{z \in Z} P_z \right| - \left| \bigcup_{z \in Z} P_z \right| + \hat{q}, \left| \bigcap_{z \in Z} P_z \right| - \check{q} \right), \quad Z \subseteq \check{T} \quad (2.18)$$

де функція $GH_1(x, y, z) = x \cdot (\text{sign}(y) + \text{sign}(z))$, \hat{q} , \check{q} – цілочисельні параметри ($\hat{q}, \check{q} \in Z$). Відмітимо, що функцію GH_1 можна визначити і в інший спосіб, який би враховував не тільки наявність спільних властивостей, а й також їх кількість. Ця функція вибору орієнтована на знаходження максимальної кількості однорідних об'єктів. Якщо в онтології існує не одна група таких об'єктів, то внаслідок послідовного застосування цього механізму вони теж можуть бути ідентифіковані.

Функція $\Phi(Z)$ може бути визначена також за допомогою безпосередньо попарних порівнянь.

$$\Phi(Z) = \left(\sum_{x \in Z} \sum_{y \in Z, x \neq y} GH_2(x, y, Z) \cdot \text{Pr}(x, y, Z) \right), \quad Z \subseteq \check{T} \quad (2.19)$$

де $\text{Pr}(x, y, Z)$ – функція, яка на основі предикатів визначає, чи володіє комплект об'єктів мінімальними до них вимогами.

$GH_2(x, y, Z)$ – функція, яка визначає міру схожості об'єктів x та y ($x \in Z, y \in Z$) та, яка може враховувати кількість елементів в комплекті, за замовчуванням її можна покласти $GH_2(x, y, Z)=1$.

$$\Pr(x, y, Z) = \begin{cases} 1, & \Pr_1(x, y) \wedge \Pr_2(x, Z) \wedge \Pr_2(y, Z) \\ -\infty, & \neg\Pr_1(x, y) \vee \neg\Pr_2(x, Z) \vee \neg\Pr_2(y, Z) \end{cases}$$

$$\Pr_1(x, y) = (|P_x \cap P_y| \geq \check{q}) \wedge (|P_x \cup P_y| - |P_x \cap P_y| \leq \hat{q})$$

$$\Pr_2(x, Z) = |P_x| - \left| \bigcup_{z \in Z} P_z \right| \geq \check{q}$$

де P_x, P_y – множини властивостей елементів x та y ($x \in Z, y \in Z$), параметри $\check{q}, \hat{q}, \tilde{q}$ – цілочисельні порогові значення.

Функції, що описані (2.15)–(2.16) володіють всіма характеристичними властивостями, що описані в (2.8)–(2.11) за умови, що властивості об'єктів є незмінними. Функція вибору, що описана в (2.17)–(2.19) володіє тільки властивістю відкидання (В). Але слід врахувати, що цей механізм вибору орієнтований на знаходження саме оптимально комплекту на відміну від звичайної ситуації, де об'єкти можуть розглядатися незалежно. Тобто відкидання певних елементів із комплекту об'єктів суттєво змінює його якісні характеристики, тому самі комплекти мають бути цілісними. Якщо накласти цю вимогу, то описану функцію можна розглядати, як раціональну.

Наступний етап, який слідує після визначення множини альтернатив полягає у встановленні множини атрибутів, що можуть розглядатись, як критерії K . Це може бути здійснено наступним чином:

$$K = \bigcup_{z \in A} P_z, \quad F_{sel}(\check{T}) = A, \quad A \subseteq \check{T} \quad (2.20)$$

Таким чином формується множина K , яка може бути використана для до визначення множини альтернатив. Тобто вона може слугувати для визначення множин Z_1 та Z_2 , які застосовуються для вибору альтернатив. Отже ці етапи формування альтернатив та критеріїв взаємопов'язані $M_R^{1A} \Leftrightarrow M_R^2$, оскільки властивості об'єктів дозволяють встановити множину альтернатив, які в свою чергу характеризуються спільними атрибутами.

Наступний крок при формуванні моделі задачі ранжування полягає у встановленні критеріальних значень для альтернатив. Для цього з онтологічної

моделі виокремлюються значення атрибутів із відношення ("об'єкт", "атрибут", "значення").

$$\tilde{F}(x, y) = v, (x, y, v) \in R_v \subseteq R \quad (2.21)$$

де $x \in A$, $y \in K$, $v \in Q$ та R_v – тринарне відношення, що задане для множини X .

Після того, як визначено множину критеріїв і альтернатив, та їх критеріальні значення будемо вважати, що модель задачі ранжування фактично поставлена. Подальше її уточнення та розв'язання залежить від безпосередньої участі ОПР.

Виходячи з даного нами формального визначення онтології (2.1) та категорії вибору, яка представлена виразами 2.7–2.19, представимо онтологію задачі вибору *OTPCh* [22, 80, 81] у вигляді наступного виразу:

$$O_{TPCh} = \langle \check{T}, (\check{\rho} \vee R^+), F_{sel}(\check{T}) \rangle \quad (2.22)$$

де \check{T} – таксономія, $\check{\rho}$ – відношення часткового порядку, R^+ – властивості об'єктів-концептів, F_{sel} – функція вибору.

Отже, об'єктні компоненти, що визначають онтологію задачі вибору, дозволяють інтерпретувати їх як різні інформаційні ресурси, концепти яких пов'язані певними бінарними відношеннями часткового порядку і можуть мати унарні властивості, що характеризують їх у певному якісному вигляді.

Тепер безпосередньо уточнимо математичну модель задачі ранжування (2.5), яка може бути формально описана так. Кожна альтернатива $x \in A = \{A_1, A_2, \dots, A_n\}$ (A – множина альтернатив) характеризується значеннями деякої сукупності показників $f_1(x), f_2(x), \dots, f_m(x)$. Кожна функція $f_j(x)$, $j \in J = \{1 \dots m\}$ задає значення j -го критерія [37, 41, 46, 87, 90], яке належить або до наперед визначеної множини, або обраховується у відповідності з певними математичними правилами [41, 46, 91]. У першому випадку можливі варіанти: множина значень задається бальною чи лінгвістичною шкалою [46, 90] або – у вигляді числового інтервалу $[f_j^{min}, f_j^{max}]$, який утворюється з усіх можливих значень функції (з мінімального до максимального) з урахуванням точності її

обчислення. Прикладом другого випадку є синтез локальних пріоритетів у методі аналізу ієрархій [74]. Отже, можна вважати, що значення j -го критерія завжди є зліченною множиною, позначимо її, як Q_j :

$$Q_j = \{f_j^{(1)}, f_j^{(2)}, f_j^{(3)}, \dots, f_j^{(n_j-1)}, f_j^{(n_j)}\},$$

де $f_j^{(1)} = f_j^{min}$, $f_j^{(n_j)} = f_j^{max}$, $f_j^{(1)} < f_j^{(2)} < \dots < f_j^{(n_j)}$, $n_j = |Q_j|$ – кількість елементів множини Q_j .

Отже, задача ранжування альтернатив за сукупністю показників полягає у встановленні певного порядку

$$A_{i_1} \succcurlyeq A_{i_2} \succcurlyeq \dots \succcurlyeq A_{i_n}, \quad i_1, i_2, \dots, i_n \in \{1 \dots n\} \quad (2.23)$$

на основі обчислення значень деякого узагальненого показника $G(x)$ для кожного елемента множини A :

$$\begin{aligned} G(x) &= G(f(x), W) = G((f_1(x), \dots, f_m(x)), (\omega_1, \dots, \omega_m)), \\ x \in A &= \{A_1, \dots, A_n\}, \\ W &= (\omega_1, \dots, \omega_m), \quad \sum_{j=1}^m \omega_j = 1, \quad \omega_j > 0, \end{aligned} \quad (2.24)$$

де значення $G(x)$ обчислюються за певним правилом (алгоритмом), що визначається математичним методом, який використовується в кожному конкретному випадку, причому

$$G(A_{i_1}) \geq G(A_{i_2}) \geq \dots \geq G(A_{i_n}), \quad (2.25)$$

а W – є нормованим вектором вагових коефіцієнтів критеріїв [41, 46, 90].

Найкращою вважається альтернатива A_{i_1} , яка у порядку (2.23) займає перше місце, відповідно, найгіршою – альтернатива A_{i_n} . Далі будемо говорити, що альтернатива A_{i_k} в порядку (2.23) знаходиться на k -му місці, а k будемо називати її рейтингом.

Слід зауважити, що яким би методом не розв'язувалась задача (2.23)-(2.25), достовірність отриманого результату суттєво залежить від способу перетворення значень Q_j , $j \in J$ в єдину шкалу для всіх критеріїв. Так, на практиці достатньо часто застосовують формулу [55, 56, 66, 135]

$$f'_j(x) = q_{min} + (q_{max} - q_{min}) \left(\frac{f_j(x) - \inf Q_j}{\sup Q_j - \inf Q_j} \right), x \in X = \{A_1, \dots, A_n\}, \quad (2.26)$$

проте, вона може не зовсім коректно відображувати «суттєвість» переваги однієї альтернативи над іншою після переведення початкових значень $f_j(x)$ у шкалу $[q_{min}; q_{max}]$. В результаті при використанні отриманих значень $f'_j(x)$ в узагальненому показнику (2.24) розв'язок задачі (2.23)-(2.25) може бути хибним. Тому виникає необхідність у розробленні інших підходів до нормалізації критеріїв, які більш адекватно враховують реальне співвідношення вагомості альтернатив по кожному критерію.

2.2. Алгоритм конкурентної нормалізації критеріїв

Розглядається j -й критерій, за яким необхідно встановити реальний ступінь переваги однієї альтернативи над іншою. Критерій максимізується, причому $f_j(A_1) \leq f_j(A_2) \leq \dots \leq f_j(A_n)$. Нехай кожна альтернатива A_i за цим показником має значення b_i ($f_j(A_i) = b_i$, $i = \overline{1, n}$), а b_{ideal} – максимальне потенційно можливе значення за j -м критерієм. b_{ideal} визначається ОПР і може або співпадати з максимумом шкали значень цього критерія, або обчислюватись в певний спосіб в залежності від «потенційної» здатності альтернативи набути це значення. Обчислимо значення $b_{avg} = \frac{\sum_{i=1}^n b_i}{n}$.

Отже, розробимо відповідне перетворення

$$\Psi: [b_{min}, b_{max}] \rightarrow [0; 1], \quad (2.27)$$

за яким кожне значення b_i , $i = \overline{1, n}$ отримає свій еквівалент у шкалі рейтингових балів $[0; 1]$.

Слід зауважити, що у випадку застосування (2.26) $q_{min} = 0$, $q_{max} = 1$.

В геометричній інтерпретації будемо вважати, що рейтинговий бал кожної альтернативи залежить від довжини ламаної, яка починається з центру координат і проходить через точки, абсиси яких є значеннями j -го критерія.

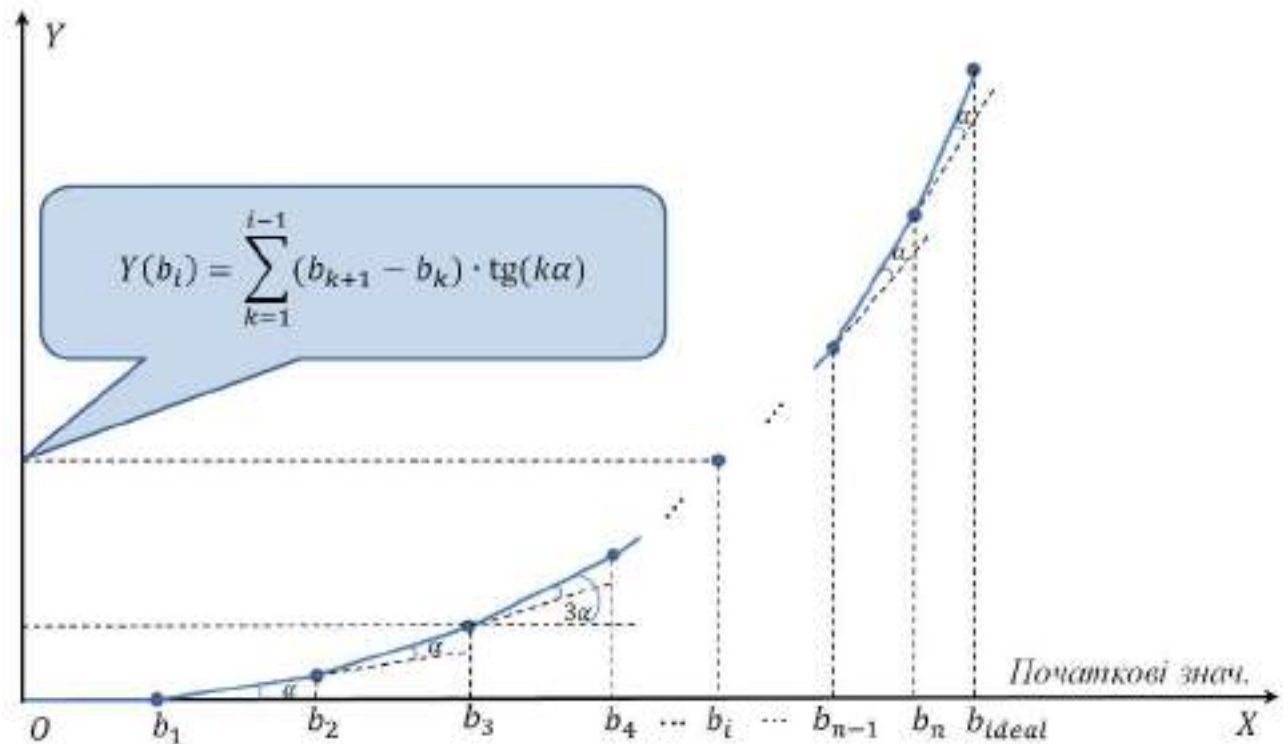


Рис. 2.1. Геометрична інтерпретація перетворення Ψ

Величина кута відхилення α кожної наступної ланки від попередньої вважається не змінною (рис. 2.1) і обраховується з таких міркувань. Введемо позначення:

- $B_i = \frac{P_i}{P_{n+1}}$, $i = \overline{1, n}$ – величина балу i -ї альтернативи у шкалі $[0; 1]$,
- $P_i = P_{i-1} + \frac{(b_i - b_{i-1})}{\cos((i-1)\alpha)}$, $i = \overline{1, n+1}$ – довжина ламаної від центру координат до точки з абсисою b_i ($P_0 = 0$).

Отже, маємо:

$$P_1 = b_1,$$

$$P_2 = P_1 + \frac{(b_2 - b_1)}{\cos(\alpha)},$$

$$P_3 = P_2 + \frac{(b_3 - b_2)}{\cos(2\alpha)},$$

...

$$P_i = P_{i-1} + \frac{(b_i - b_{i-1})}{\cos((i-1)\alpha)},$$

...

$$P_{n+1} = P_n + \frac{(b_{ideal} - b_n)}{\cos(n\alpha)}.$$

Величина P_{n+1} є довжиною всієї ламаної (до точки з абсцисою b_{ideal}) і у шкалі $[0; 1]$ відповідає значенню 1. Тоді $B_i = \frac{P_i}{P_{n+1}}$, $i = \overline{1, n+1}$. Цілком логічним є твердження, що середнє значення величин B_i , $i = \overline{1, n}$ повинно знаходитись на середині інтервалу $[0; 1]$: $\frac{\sum_{i=1}^n B_i}{n} = 0.5$, адже це дозволить коректно зводити початкові критеріальні значення до рейтингових для альтернатив за різнорідними показниками. В цьому випадку середньому критеріальному значенню, яке відрізняється у кожного окремого показника, буде приблизно відповідати один і той самий бал 0.5 у шкалі $[0; 1]$.

Таким чином, отримуємо рівняння для знаходження кута α :

$$\begin{aligned} \frac{\sum_{i=1}^n B_i}{n} &= \frac{\sum_{i=1}^n \frac{P_i}{P_{n+1}}}{n} = \frac{\sum_{i=1}^n P_i}{nP_{n+1}} = \frac{\sum_{i=1}^n \left(\sum_{k=1}^i \frac{(b_k - b_{k-1})}{\cos((k-1)\alpha)} \right)}{n \left(\sum_{k=1}^{n+1} \frac{(b_k - b_{k-1})}{\cos((k-1)\alpha)} \right)} = \\ &= \frac{\sum_{i=1}^n \left(\frac{(b_1 - b_0)}{\cos(0 \cdot \alpha)} + \frac{(b_2 - b_1)}{\cos(1 \cdot \alpha)} + \frac{(b_3 - b_2)}{\cos(2 \cdot \alpha)} + \dots + \frac{(b_i - b_{i-1})}{\cos((i-1)\alpha)} \right)}{n \left(\sum_{k=1}^{n+1} \frac{(b_k - b_{k-1})}{\cos((k-1)\alpha)} \right)} = \\ &= \frac{nb_1 + (n-1) \frac{(b_2 - b_1)}{\cos \alpha} + (n-2) \frac{(b_3 - b_2)}{\cos 2\alpha} + \dots + (1) \frac{(b_n - b_{n-1})}{\cos n\alpha}}{n \left(b_1 + \frac{(b_2 - b_1)}{\cos \alpha} + \dots + \frac{(b_n - b_{n-1})}{\cos n\alpha} + \frac{(b_{n+1} - b_n)}{\cos(n+1)\alpha} \right)} = 0.5, \end{aligned}$$

або

$$\frac{\sum_{k=1}^n \left((n-k+1) \frac{(b_k - b_{k-1})}{\cos((k-1)\alpha)} \right)}{n \left(\sum_{k=1}^{n+1} \frac{(b_k - b_{k-1})}{\cos((k-1)\alpha)} \right)} = 0.5 \quad (2.28)$$

Отримане рівняння будемо розв'язувати чисельно. Проведемо дослідження його лівої частини, як функції $Z(\alpha)$, $\alpha \in [0; \frac{\pi}{2(n+1)})$:

$$Z(\alpha) = \frac{\sum_{k=1}^n (F_k(\alpha))}{n(F_{n+1}(\alpha))} = \frac{F_1(\alpha) + F_2(\alpha) + \dots + F_n(\alpha)}{n(F_{n+1}(\alpha))},$$

де $F_k(\alpha) = \sum_{i=1}^k g_i(\alpha) = \sum_{i=1}^k \frac{(b_i - b_{i-1})}{\cos((i-1)\alpha)}$, $k = \overline{1, n+1}$.

Нехай φ деякий кут ($\varphi > 0, \alpha + \varphi < \frac{\pi}{2(n+1)}$). Покажемо, що виконується нерівність

$$\frac{F_k(\alpha)}{F_{k+1}(\alpha)} > \frac{F_k(\alpha + \varphi)}{F_{k+1}(\alpha + \varphi)}$$

$$\frac{\sum_{i=1}^k g_i(\alpha)}{\sum_{i=1}^k g_i(\alpha) + g_{k+1}(\alpha)} > \frac{\sum_{i=1}^k g_i(\alpha + \varphi)}{\sum_{i=1}^k g_i(\alpha + \varphi) + g_{k+1}(\alpha + \varphi)}$$

$$g_{k+1}(\alpha + \varphi) \sum_{i=1}^k g_i(\alpha) > g_{k+1}(\alpha) \sum_{i=1}^k g_i(\alpha + \varphi)$$

$$\sum_{i=1}^k \frac{(b_i - b_{i-1})}{\cos(k(\alpha + \varphi)) \cos((i-1)\alpha)} > \sum_{i=1}^k \frac{(b_i - b_{i-1})}{\cos(k(\alpha)) \cos((i-1)(\alpha + \varphi))}$$

$$\cos(k(\alpha)) \cos((i-1)(\alpha + \varphi)) > \cos(k(\alpha + \varphi)) \cos((i-1)\alpha)$$

$$\frac{\cos((i-1)(\alpha + \varphi))}{\cos(k(\alpha + \varphi))} > \frac{\cos((i-1)(\alpha))}{\cos(k(\alpha))}, \quad 0 \leq (i-1) < k.$$

Дослідимо функцію $z(x) = \frac{\cos((i-1)x)}{\cos(kx)}$, $0 \leq (i-1)x < kx < \frac{\pi}{2}$

$$z'(x) = \left(\frac{\cos((i-1)x)}{\cos(kx)} \right)' =$$

$$= \frac{-(i-1) \cdot \sin((i-1)x) \cos(kx) + k \cdot \sin(kx) \cos((i-1)x)}{(\sin(kx))^2} > 0$$

$$k \cdot \sin(kx) \cos((i-1)x) > (i-1) \cdot \sin((i-1)x) \cos(kx) \Rightarrow$$

$$\Rightarrow \operatorname{tg}(kx) > \operatorname{tg}((i-1)x), \quad (k > i-1).$$

Оскільки $\operatorname{tg}(x)$ монотонно зростаюча функція на інтервалі $[0; \frac{\pi}{2})$ і $kx > (i-1)x$, то виконується отримана нерівність. Отже, для $\forall k = \overline{1, n}$ маємо

$$\frac{F_k(\alpha)}{F_{k+1}(\alpha)} > \frac{F_k(\alpha + \varphi)}{F_{k+1}(\alpha + \varphi)} \Rightarrow \frac{F_{k+1}(\alpha + \varphi)}{F_{k+1}(\alpha)} > \frac{F_{k+1}(\alpha)}{F_{k+1}(\alpha + \varphi)}$$

$$\frac{F_{n+1}(\alpha + \varphi)}{F_{n+1}(\alpha)} > \frac{F_{k+1}(\alpha + \varphi)}{F_{k+1}(\alpha)} > \frac{F_{k+1}(\alpha)}{F_{k+1}(\alpha + \varphi)} \Rightarrow \frac{F_k(\alpha)}{F_{n+1}(\alpha)} > \frac{F_k(\alpha + \varphi)}{F_{n+1}(\alpha + \varphi)} \Rightarrow$$

$$\Rightarrow \frac{\sum_{k=1}^n (F_k(\alpha))}{n(F_{n+1}(\alpha))} > \frac{\sum_{k=1}^n (F_k(\alpha + \varphi))}{n(F_{n+1}(\alpha + \varphi))} \Rightarrow Z(\alpha) > Z(\alpha + \varphi).$$

Таким чином, $Z(\alpha)$ є монотонно спадною функцією на інтервалі $(0; \frac{\pi}{2(n+1)})$, і задача полягає у пошуку кореня рівняння $Z(\alpha) - 0.5 = 0$.

На інтервалі $[0; \frac{\pi}{2(n+1)})$ розглянемо функцію $U(\alpha) = Z(\alpha) - 0.5$, яка також буде монотонно спадною за властивостями монотонних функцій.

Визначимо знак функції $Z(\alpha)$ в кінці інтервалу $[0; \frac{\pi}{2(n+1)})$.

$$\lim_{\alpha \rightarrow \frac{\pi}{2(n+1)}} Z(\alpha) =$$

$$nb_1 + (n-1) \frac{(b_2 - b_1)}{\cos \frac{\pi}{2(n+1)}} + \dots + \frac{(b_n - b_{n-1})}{\cos \frac{n\pi}{2(n+1)}}$$

$$= \frac{n(b_1 + \frac{(b_2 - b_1)}{\cos \frac{\pi}{2(n+1)}} + \dots + \frac{(b_n - b_{n-1})}{\cos \frac{n\pi}{2(n+1)}}) + \lim_{\alpha \rightarrow \frac{\pi}{2(n+1)}} \frac{(b_{n+1} - b_n)}{\cos(n+1)\alpha}}{1} =$$

$$= \frac{const}{const + \lim_{\alpha \rightarrow \frac{\pi}{2(n+1)}} \frac{(b_{n+1} - b_n)}{\cos(n+1)\alpha}} = \frac{const}{const + (+\infty)} = 0.$$

Таким чином

$$\lim_{\alpha \rightarrow \frac{\pi}{2(n+1)}} U(\alpha) = \lim_{\alpha \rightarrow \frac{\pi}{2(n+1)}} Z(\alpha) - 0.5 = -0.5 < 0.$$

Обчислимо значення $U(\alpha)$ на початку інтервалу при $\alpha = 0$:

$$\begin{aligned} U(0) &= \frac{\sum_{k=1}^n \left((n-k+1) \frac{(B_k - B_{k-1})}{\cos((k-1) \cdot 0)} \right)}{n \left(\sum_{k=1}^{n+1} \frac{(B_k - B_{k-1})}{\cos((k-1) \cdot 0)} \right)} - 0.5 = \\ &= \frac{\sum_{k=1}^n ((n-k+1)(B_k - B_{k-1}))}{n(\sum_{k=1}^{n+1} (B_k - B_{k-1}))} - 0.5 = \frac{\sum_{k=1}^n B_k}{nB_{n+1}} - 0.5 = \frac{B_{avg}}{B_{n+1}} - 0.5. \end{aligned}$$

Отже, функція $U(\alpha)$ є монотонно спадною на інтервалі $\left[0; \frac{\pi}{2(n+1)}\right)$. Якщо виконується умова $\frac{B_{avg}}{B_{n+1}} \geq 0.5$, то функція на кінцях інтервалу приймає різні знаки. Це дозволяє розв'язувати рівняння $U(\alpha) = 0$ методом дихотомії. Наведемо відповідний алгоритм пошуку кута α .

0. Ініціалізація. Кут між останньою ланкою, що з'єднує точки $(B_n; Y(B_n))$ та $(B_{ideal}; Y(B_{ideal}))$, і додатнім напрямом вісі OX не перевищує $\frac{\pi}{2}$. Тому цілком очевидними є початкові значення: $\alpha_{min} = 0$, $\alpha_{max} = \frac{\pi}{2(n+1)}$. Також задається точність, наприклад, $\varepsilon = 10^{-6}$ і номер ітерації $j = 1$.

1. Встановлюємо $\alpha_j = \frac{\alpha_{min} + \alpha_{max}}{2}$.

Для даного α обраховуємо довжини $P_i = P_{i-1} + \frac{(b_i - b_{i-1})}{\cos((i-1)\alpha_j)}$, $i = \overline{1, n+1}$.

2. Обраховуємо віднормовані бали $B_i = \frac{P_i}{P_{n+1}}$, $i = \overline{1, n+1}$.

3. Обчислюємо середнє значення $B_{avg} = \frac{\sum_{i=1}^n B_i}{n}$.
4. Якщо $|B_{avg} - 0.5| < \varepsilon$, переходимо на крок 6, .
5. Якщо $B_{avg} > 0.5$, то $\alpha_{min} = \alpha_j$, інакше $\alpha_{max} = \alpha_j$; $j = j + 1$, переходимо на крок 1.
6. α_j – є розв'язком рівняння (2.28).

Далі ОПР має визначити функцію цінності для інтерпретації отриманих балів, наприклад, за допомогою порогових значень. Подальші дії ОПР залежать від сфери діяльності, для якої розв'язується конкретна задача. В деяких випадках допустимо використання, наприклад, статистичних спостережень, що характеризують властивості об'єктів ранжування.

Відповідно до алгоритму конкурентної нормалізації ОПР повинен визначити віддаленість середнього балу від 0, щоб забезпечити ступінь переваги L очікувано-максимального балу B_* над середнім балом B_{avg} . Слід зауважити, що B_* є параметром, а його значення можна отримати через статистичні спостереження або в інший спосіб, визначений ОПР в наслідок аналізу предметної області і специфіки задачі.

Отже, проводимо стискання в x раз середнього балу 0.5 до такої величини $\frac{0.5}{x}$, яка була б у L разів менша від балу $\frac{0.5}{x} + \Delta$, де $\Delta = B_* - 0.5$. Таким чином можна досягти кардинальної (чисельної) узгодженості між усіма значеннями B_k , $k = \overline{1, n}$.

$$L \cdot \frac{0.5}{x} = \frac{0.5}{x} + \Delta \Rightarrow \frac{0.5}{x} (L - 1) = \Delta \Rightarrow x = \frac{L - 1}{2 \cdot \Delta} = \mu.$$

Тепер для всіх B_i , $i = \overline{1, n}$ отримаємо нові значення \tilde{B}_i :

$$\tilde{B}_i = \begin{cases} \frac{B_i}{\mu}, & B_i \leq 0.5; \\ \frac{0.5}{\mu} + (B_i - 0.5), & B_i > 0.5 \end{cases}$$

Обчислимо значення максимального балу у новій шкалі: $\tilde{B}_{max} = \frac{0.5}{\mu} + (1 - 0.5)$.

Тепер, з огляду на (2.26), можна отримати рейтингові значення об'єктів ранжування для шкали $[q_{min}; q_{max}]$:

$$R_i = q_{min} + (q_{max} - q_{min}) \cdot \frac{\tilde{B}_i}{\tilde{B}_{max}}, \quad i = \overline{1, n},$$

де R_i – остаточні рейтингові бали.

Для прикладу розглянемо обчислення рейтингового балу деяких 23 об'єктів для шкали $[0; 1000]$, тобто $q_{min} = 0$, $q_{max} = 1000$. У табл. 2.1 наведені початкові значення b_i , $i = \overline{1, 23}$ певного показника. Нехай ОПР визначив $b_{ideal} = 98$ і значення параметрів $L = 3$ та $B_* = 0.75$.

Методом дихотомії знаходимо кут α такий, як зображено на рис. 2.1. В нашому випадку це значення є $\alpha = 0,06153$ радіан. Далі обчислюємо бали $P_i, B_i, \tilde{B}_i, R_i$, $i = \overline{1, 23}$ (табл. 2.1).

Таблиця 2.1. Початкові значення для 23 об'єктів

№ п/п	Об'єкти	b
1	Альтернатива 1	91,46
2	Альтернатива 2	89,8
3	Альтернатива 3	86,87
4	Альтернатива 4	84,77
5	Альтернатива 5	84,51
6	Альтернатива 6	80,8
7	Альтернатива 7	80,7
8	Альтернатива 8	79,07
9	Альтернатива 9	78,18
10	Альтернатива 10	75,18
11	Альтернатива 11	71,9
12	Альтернатива 12	67,32
13	Альтернатива 13	67,07
14	Альтернатива 14	66,1
15	Альтернатива 15	65,42
16	Альтернатива 16	62,57
17	Альтернатива 17	62,47
18	Альтернатива 18	60,88
19	Альтернатива 19	58,57
20	Альтернатива 20	18
21	Альтернатива 21	16
22	Альтернатива 22	16
23	Альтернатива 23	16

Довжина ламаної (рис. 2.1) від центру координат до точки з абсцисою b_{ideal} в нашому випадку є сумою довжин відрізків:

$$P_1 = 16,$$

$$P_2 = 16 + \frac{(16 - 16)}{\cos(0,06153)} = 16,$$

$$P_3 = 16 + \frac{(16 - 16)}{\cos(0,12306)} = 16,$$

$$P_4 = 16 + \frac{(18 - 16)}{\cos(0,18459)} = 18.03,$$

$$P_5 = 18.03 + \frac{(58,57 - 18)}{\cos(0,24612)} = 59.87,$$

...

$$P_{23} = 115,69 + \frac{(91,46 - 89,8)}{\cos(1,35367)} = 123.38,$$

$$P_{24} = 123.38 + \frac{(98 - 91,46)}{\cos(1,415199)} = 165.60.$$

Тоді величина балу i -го учасника у шкалі $[0; 1]$ є $B_i = \frac{P_i}{165.60}$, $i = \overline{1,23}$.

Знаходимо такий індекс r , при якому справджується подвійна нерівність

$$B_r < 0.5 \leq B_{r+1} \quad (r = 14).$$

Відповідно до алгоритму $\mu = \frac{L-1}{2(B_*-0.5)} = \frac{3-1}{2(0.75-0.5)} = 4$

$$\tilde{B}_1 = \frac{B_1}{4} = \frac{0.1}{4} = 0.025$$

$$\tilde{B}_2 = \frac{B_2}{4} = \frac{0.1}{4} = 0.025$$

$$\tilde{B}_3 = \frac{B_3}{4} = \frac{0.1}{4} = 0.025$$

$$\tilde{B}_4 = \frac{B_4}{4} = \frac{0.11}{4} = 0.0275$$

...

$$\tilde{B}_{14} = \frac{B_{14}}{4} = \frac{0,49}{4} = 0.1225$$

$$\tilde{B}_{15} = \frac{0,5}{4} + (B_{15} - 0,5) = \frac{0,5}{4} + (0,51 - 0,5) = 0,135$$

$$\tilde{B}_{16} = \frac{0,5}{4} + (B_{16} - 0,5) = \frac{0,5}{4} + (0,52 - 0,5) = 0,145$$

...

$$\tilde{B}_{23} = \frac{0,5}{4} + (B_{23} - 0,5) = \frac{0,5}{4} + (0,75 - 0,5) = 0,375$$

Обчислені значення наведено в табл. 2.2.

Таблиця 2.2. Результати роботи алгоритму обчислення рейтингового балу.

Об'єкти	Індекс	<i>b</i>	<i>P</i>	<i>B</i>	\tilde{B}	<i>R</i>
Альтернатива 1	23	91,46	123,38	0,75	0,375	600
Альтернатива 2	22	89,80	115,69	0,70	0,325	520
Альтернатива 3	21	86,87	105,02	0,63	0,255	408
Альтернатива 4	20	84,77	98,74	0,60	0,225	360
Альтернатива 5	19	84,51	98,07	0,59	0,215	344
Альтернатива 6	18	80,80	89,77	0,54	0,165	264
Альтернатива 7	17	80,70	89,58	0,54	0,165	264
Альтернатива 8	16	79,07	86,63	0,52	0,145	232
Альтернатива 9	15	78,18	85,15	0,51	0,135	216
Альтернатива 10	14	75,18	80,55	0,49	0,1225	196
Альтернатива 11	13	71,90	75,84	0,46	0,115	184
Альтернатива 12	12	67,32	69,64	0,42	0,105	168
Альтернатива 13	11	67,07	69,32	0,42	0,105	168
Альтернатива 14	10	66,10	68,14	0,41	0,1025	164
Альтернатива 15	9	65,42	67,34	0,41	0,1025	164
Альтернатива 16	8	62,57	64,10	0,39	0,0975	156
Альтернатива 17	7	62,47	64,00	0,39	0,0975	156
Альтернатива 18	6	60,88	62,29	0,38	0,095	152
Альтернатива 19	5	58,57	59,87	0,36	0,09	144
Альтернатива 20	4	18,00	18,03	0,11	0,0275	44
Альтернатива 21	3	16,00	16,00	0,10	0,025	40
Альтернатива 22	2	16,00	16,00	0,10	0,025	40
Альтернатива 23	1	16,00	16,00	0,10	0,025	40

Обчислимо $\tilde{B}_{max} = \frac{0.5}{\mu} + (1 - 0.5) = \frac{0.5}{4} + 0.5 = 0.625$.

Тепер, з огляду на (2.26), можна отримати рейтингові значення об'єктів ранжування для шкали $[q_{min}; q_{max}]$:

$$R_i = 0 + (1000 - 0) \cdot \frac{\tilde{B}_i}{0.625} = 1600 \cdot \tilde{B}_i, \quad i = \overline{1, n}$$

Остаточно маємо:

$$R_1 = \tilde{B}_1 \cdot 1600 = 0.0242 \cdot 1600 = 40$$

$$R_2 = \tilde{B}_2 \cdot 1600 = 0.0242 \cdot 1600 = 40$$

$$R_3 = \tilde{B}_3 \cdot 1600 = 0.0242 \cdot 1600 = 40$$

$$R_4 = \tilde{B}_4 \cdot 1600 = 0.0273 \cdot 1600 = 44$$

...

$$R_{23} = \tilde{B}_{23} \cdot 1600 = 0.375 \cdot 1600 = 600$$

На рис. 2.2 у графічному вигляді зображені залежності рейтингових балів у шкалі $[0..1000]$ від початкових.

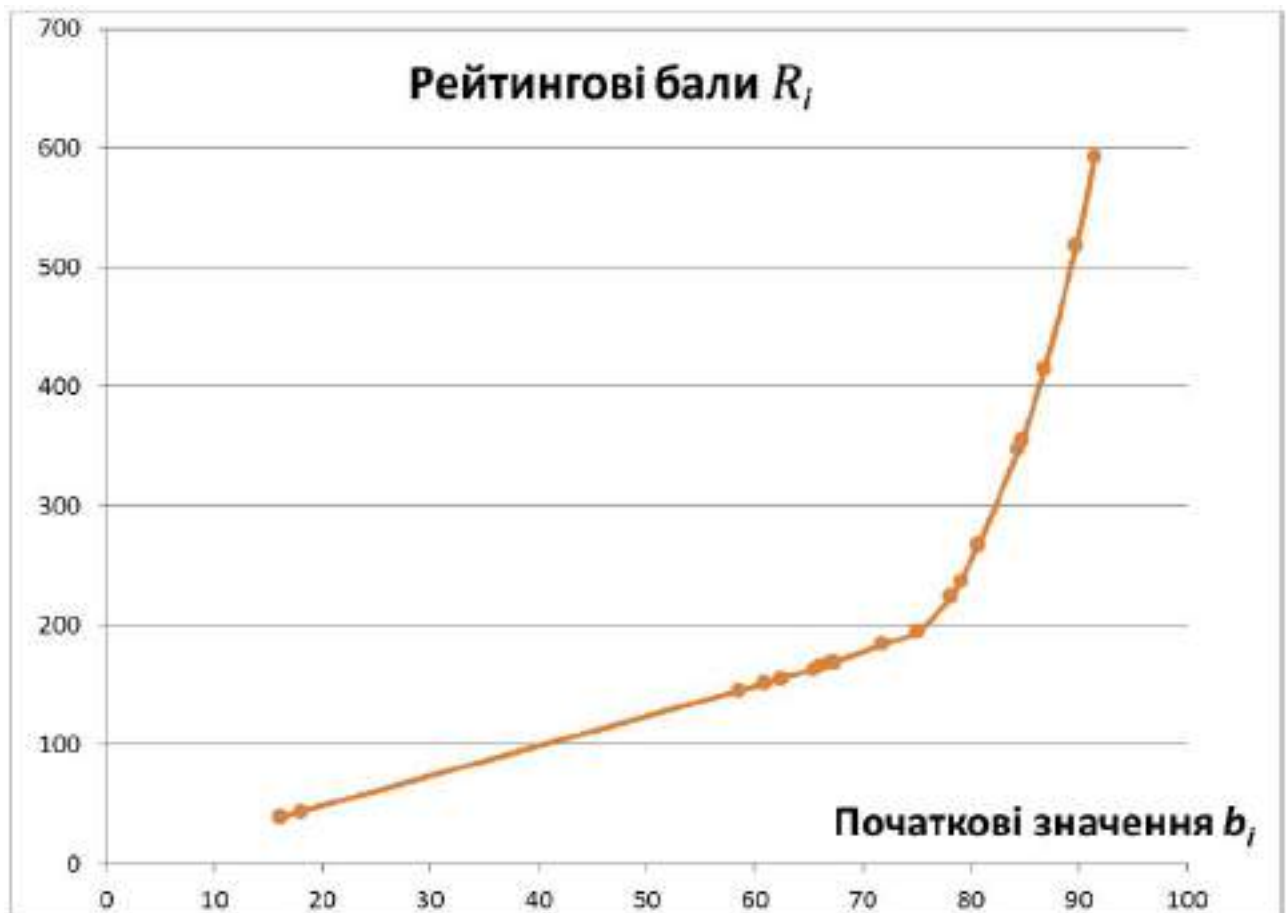


Рис. 2.2. Графічне представлення обчислених алгоритмом рейтингових балів.

Алгоритм конкурентної нормалізації був апробований для розв'язку практичних задач побудови рейтингів об'єктів, зокрема – для учасників республіканських етапів Всеукраїнського конкурсу-захисту науково-дослідницьких робіт учнів – членів Малої академії наук України.

Алгоритм доцільно використовувати у випадках, коли ступінь переваги однієї альтернативи над іншою залежить не від абсолютних чисельних початкових значень показника, а від того, наскільки багато альтернатив мають близькі значення між собою і не досягають або переважають порогові значення задані ОПР в результаті аналізу предметної області.

2.3. Оборнена задача ранжування альтернатив

2.3.1. Постановка оборненої задачі ранжування альтернатив

Після розв'язання задачі (2.23)–(2.25) може виникнути інша задача – при яких мінімальних змінах значень $f_j(a')$, $j \in J$ можна покращити рейтинг обраної альтернативи a' ?

Отже, аналізується певна альтернатива $a' \in A$, яка фігурує у реальній задачі прийняття рішень вигляду (2.23)–(2.25), а бажаним результатом розв'язку нової задачі вважається отримання даною альтернативою рейтингу r , не нижче від наперед заданого значення p ($1 \leq p < n$). Не зменшуючи узагальненості, будемо вважати, що для кожного показника $f_j(a)$, $j \in J$ значення тим краще, чим воно більше.

Нехай після визначення вектору вагових коефіцієнтів W і обрання конкретної функції $G(a)$, $a \in A$ альтернатива a' посіла у порядку (2.23) не бажане місце r , тобто $r > p$. Для кожного критерія $j \in J$ особа, що приймає рішення (ОПР), визначає множину \tilde{Q}_j , утворену такими значеннями показника f_j , які є більшими від $f_j(a')$: $\tilde{Q}_j = \{f_j(a') + \theta_j^0, f_j(a') + \theta_j^1, \dots, f_j(a') + \theta_j^{r_j}\} \subset Q_j$, де $\theta_j^0 < \theta_j^1 < \dots < \theta_j^{r_j}$, $\theta_j^0 = 0$. Верхній індекс r_j , $0 \leq r_j \leq |\tilde{Q}_j| + 1$

відповідає кількості перших таких значень серед усіх можливих, \hat{Q}_j – множина всіх значень критерія f_j більших ніж $f_j(a')$: $\hat{Q}_j = Q_j \setminus \{f_j^{min}, f_j^{(2)}, f_j^{(3)}, \dots, f_j(a')\}$.

Розглянемо впорядковану множину

$$\theta_j = \{\theta_j^0, \theta_j^1, \theta_j^2, \dots, \theta_j^{r_j}\}, j \in J. \quad (2.29)$$

Тоді кожний елемент такої множини слід сприймати, як певну величину покращення значення $f_j(a')$, яке можна досягти, доклавши “певних зусиль” до “збільшення привабливості” альтернативи a' за критерієм j .

ОПР визначає новий вектор переваг $V = (v_1, v_2, \dots, v_m)$, $\sum_{j=1}^m v_j = 1$ з таких міркувань: чим складніше для альтернативи a' досягти покращення значення по j -му показнику, тим більшим повинен бути коефіцієнт v_j , $j \in J$. Зауважимо, що в кожному конкретному випадку може існувати своя специфіка в залежності від вигляду узагальненого показника (2.24) або від способу отримання вагових коефіцієнтів критеріїв початкової задачі ранжування, що розглядається.

Таким чином математична модель задачі має вигляд:

$$h(\theta, V) = \sum_{j=1}^m v_j \theta_j \rightarrow \min \quad (2.30)$$

$$G(f(a', \theta), W) \geq G(f(A_{i_p}), W), \quad (2.31)$$

$$\theta = (\theta_1, \dots, \theta_m) \in \Theta = (\Theta_1 \times \dots \times \Theta_m), \Theta \subset \mathbb{R}^m, \quad (2.32)$$

де $f(a', \theta) = (f_1(a') + \theta_1, f_2(a') + \theta_2, \dots, f_m(a') + \theta_m)$, $\theta \in \Theta$.

У дисертаційній роботі наводиться розв'язання оберненої задачі ранжування альтернатив алгоритмами, що базуються на ідеології методів послідовного аналізу та відсіювання варіантів (ПАВ) [53], динамічного програмування [5].

2.3.2. Розв'язання оберненої задачі ранжування альтернатив методом послідовного аналізу варіантів

Приведемо алгоритм розв'язку задачі (2.30)–(2.32), що базується на ідеології методу ПАВ. У відповідності з цим методом необхідно розробити конструктивні процедури відсіву варіантів [13, 52, 53], що у підсумку впливає на зменшення множини допустимих розв'язків задачі.

Розглянемо випадок, коли для розв'язку задачі (2.23)–(2.25) використовується лінійно-адитивна згортка критеріїв [46, 48, 87, 90]. Тоді обмеження (2.31) буде мати вигляд.

$$\begin{aligned} \sum_{j \in J} \omega_j (f_j(a') + \theta_j) &\geq \sum_{j \in J} \omega_j f(A_{i_p}) \\ \sum_{j \in J} \omega_j \theta_j &\geq \sum_{j \in J} \omega_j f(A_{i_p}) - \sum_{j \in J} \omega_j f_j(a') \\ \sum_{j \in J} \omega_j \theta_j &\geq G(A_{i_p}) - G(a') \\ \sum_{j \in J} \omega_j \theta_j &\geq G^*, \end{aligned}$$

де $G^* = G(A_{i_p}) - G(a') = \text{const}$ – це саме той бар'єр, який треба подолати альтернативі a' , щоб у підсумку посісти місце не нижче ніж p .

Користуючись термінологією методу ПАВ сформулюємо теорему:

Теорема 1. Величина

$$d_j = G^* - \sum_{\substack{k=1 \\ k \neq j}}^m w_k \cdot \max\{\theta_k | \theta_k \in \Theta_k\} \quad (2.33)$$

є допуском для множини Θ_j за обмеженням (2.31).

Доведення. Нехай для деяких s ($0 \leq s < r_j$) маємо $w_j \theta_j^s < d_j$, і s' є максимальним з таких індексів. Тоді

$$\begin{aligned}
G(f(a', \theta), W) &= \sum_{\substack{k=1 \\ k \neq j}}^m w_k \theta_k + w_j \theta_j^s + G(A') < \sum_{\substack{k=1 \\ k \neq j}}^m w_k \theta_k - \\
&- \sum_{\substack{k=1 \\ k \neq j}}^m w_k \cdot \max\{\theta_k | \theta_k \in \Theta_k\} + G(a') - G^* = \sum_{\substack{k=1 \\ k \neq j}}^m w_k \theta_k - \\
&- \sum_{\substack{k=1 \\ k \neq j}}^m w_k \cdot \max\{\theta_k | \theta_k \in \Theta_k\} + G(a') + G(A_{i_p}) - G(a') < G(f(A_{i_p}), W),
\end{aligned}$$

оскільки

$$\sum_{\substack{k=1 \\ k \neq j}}^m w_k \theta_k - \sum_{\substack{k=1 \\ k \neq j}}^m w_k \cdot \max\{\theta_k | \theta_k \in \Theta_k\} < 0.$$

Отже з Θ_j вилучається множина таких елементів $\{\theta_j^0, \theta_j^1, \theta_j^2, \dots, \theta_j^{s'}\}$, для яких $w_j \theta_j^s < d_j$, $s = \overline{0, s'}$. ■

За обмеженням (2.31) і допущами (2.33) відсіювання елементів з множин $\Theta_1, \Theta_2, \dots, \Theta_m$ відбувається знизу. Якщо у підсумку множина допустимих розв'язків суттєво не зменшилась, необхідно розглянути допуски за цільовою функцією (2.30). Нехай на деякому кроці алгоритму розв'язку задачі (2.30) – (2.32) отримано допустимий розв'язок θ^* і $h^* = h(\theta^*, V)$. Має місце твердження.

Теорема 2. Величина

$$c_j = h^* - \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \min\{\theta_k | \theta_k \in \Theta_k\} \quad (2.34)$$

є допуском для множини Θ_j за цільовою функцією (2.30).

Доведення. Нехай для деяких s ($0 \leq s \leq r_j$) маємо $v_j \theta_j^s > c_j$, і s' є мінімальним з таких індексів. Тоді

$$\begin{aligned}
h(\theta, V) &= \sum_{\substack{k=1 \\ k \neq j}}^m v_k \theta_k + v_j \theta_j^s > \sum_{\substack{k=1 \\ k \neq j}}^m v_k \theta_k + h^* - \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \min\{\theta_k | \theta_k \in \Theta_k\} > \\
&> \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \min\{\theta_k | \theta_k \in \Theta_k\} + h^* - \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \min\{\theta_k | \theta_k \in \Theta_k\} > h^*
\end{aligned}$$

оскільки

$$\sum_{\substack{k=1 \\ k \neq j}}^m v_k \theta_k - \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \min\{\theta_k | \theta_k \in \Theta_k\} > 0.$$

На відміну від (2.33), за допуском (2.34) відбувається відсіювання елементів з множин $\Theta_1, \Theta_2, \dots, \Theta_m$ зверху, тобто з Θ_j вилучається множина всіх таких елементів $\{\theta_j^{s'}, \dots, \theta_j^{r'j}\}$, для яких $v_j \theta_j^s > c_j$, $s = \overline{0, s'}$. ■

Таким чином, алгоритм розв'язання задачі (2.30)–(2.32) полягає в ітераційному відсіюванні елементів множини $\Theta = (\Theta_1 \times \dots \times \Theta_m)$ за допусками (2.33), (2.34) і формально може бути представлений так.

Крок 0. Якщо при сформованих ОПР значеннях $\theta_1^{l_1}, \theta_2^{l_2}, \dots, \theta_m^{l_m}$ справджується нерівність $\sum_{j \in J} \omega_j \theta_j^{l_j} < G^*$, задача не має розв'язку. В іншому випадку покладемо $\theta_{min}^* = (\theta_1^{l_1}, \theta_2^{l_2}, \dots, \theta_m^{l_m})$, $\theta_j^{max} = \theta_j^{l_j}$, $\theta_j^{min} = \theta_j^0$, $\theta_j^{(0)} = \theta_j, j \in J, t = 0$.

Крок 1. Обчислюємо допуски знизу за обмеженням (2.31):

$$d_j^{(t)} = G^* - \sum_{\substack{k=1 \\ k \neq j}}^m w_k \cdot \theta_j^{max}, \quad j \in J.$$

Визначаємо множину елементів, що залишилась:

$$\tilde{\Theta}_j^{(t+1)} = \{\theta_j \mid w_j \theta_j \geq d_j^{(t)}, \quad \theta_j \in \Theta_j^{(t)}\}, \quad j \in J.$$

Покладемо $\theta_j^{min} = \min\{\theta_j \mid \theta_j \in \tilde{\Theta}_j^{(t+1)}\}$, $j \in J$

Крок 2. Знаходимо поточний допустимий розв'язок θ^* :

$$\theta^* = \operatorname{argmin}_{j \in J} \{v_j \cdot \theta_j^{\min} + \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \theta_j^{\max}\}.$$

Якщо $h(\theta^*, V) < h(\theta_{\min}^*, V)$, покладемо $\theta_{\min}^* = \theta^*$.

Крок 3. Обчислюємо допуски зверху за значенням цільової функції:

$$c_j^{(t)} = h(\theta_{\min}^*, V) - \sum_{\substack{k=1 \\ k \neq j}}^m v_k \cdot \theta_j^{\min}, \quad j \in J.$$

Визначаємо множину елементів, що залишились:

$$\theta_j^{(t+1)} = \{\theta_j \mid v_j \theta_j \leq c_j^{(t)}, \quad \theta_j \in \tilde{\theta}_j^{(t+1)}\}, \quad j \in J.$$

Покладемо $\theta_j^{\max} = \max\{\theta_j \mid \theta_j \in \theta_j^{(t+1)}\}$, $j \in J$

Крок 4. Якщо $\exists j \in J: \theta_j^{(t+1)} \neq \theta_j^{(t)}$, покладемо $t := t + 1$ і переходимо на крок 1, інакше – на крок 5.

Крок 5. Якщо множина $\Theta^{(t)} = (\theta_1^{(t)} \times \theta_2^{(t)} \times \dots \times \theta_m^{(t)})$ не є об'ємною (з міркувань обчислювальної складності здійснення прямого перебору варіантів), знаходимо оптимальний розв'язок задачі, інакше переходимо на крок 6

Крок 6. Вводимо обмеження на цільову функцію і ітераційно змінюючи його методом дихотомії, за допусками типу (2.34) здійснюємо відсів елементів множини $\Theta^{(t)}$ [53].

Якщо множина допустимих розв'язків задачі (2.30) – (2.32) суттєво не зменшилась, оптимальний розв'язок задачі знаходимо направленим перебором варіантів.

Формально описаний алгоритм можна представити так. (Об'єкти, що використовуються в алгоритмі, повністю відповідають визначеним вище структурам даним).

/* k, m, i, j : цілочисельні невід'ємні значення,

$Rez, mRez$ – значення цільової функції: дійсне,

G^* – початкове відхилення A_{i_p} від a' : дійсне,

$g, \omega, v, S, \theta^{\min}$ – масиви розмірності n : дійсні числа,

l, r – масиви розмірності m : цілочисельні невід’ємні значення,
 θ – двомірний масив з m стовпчиків різної висоти: дійсні числа,
 */

For ($j = 1; j \leq n; j++$) $l_i = 1$ **EndFor**

$mRez := MAX_VAL; flag = true;$

// Ініціалізація

Do while ($flag = true$)

$flag := false;$

$sum := 0; sumV := 0$

For ($j = 1; j \leq m; j++$) $sum := sum + \omega_j * \theta_{r_i, j};$ **EndFor**

For ($j = 1; j \leq m; j++$) $sumV := sumV + v_j * \theta_{r_i, j};$ **EndFor**

For ($j = 1; j \leq m; j++$) $d_j := G^* - (sum - \omega_j * \theta_{r_i, j});$ **EndFor**

// Обчислення допусків за обмеженням (2.6)

For ($j = 1; j \leq m; j++$) **For** ($i = l_i; i \leq r_i; i++$)

If ($\theta_{i, j} < d_j$) **Then** $l_i := i + 1;$ **else**

If ($mRez > sumV - v_j * \theta_{r_i, j} + v_j * \theta_{l_i, j}$) **Then**

$mRez := sumV - v_j * \theta_{r_i, j} + v_j * \theta_{l_i, j};$ **EndIf**

$flag := true;$ **Break** **EndIf**

EndFor

If ($l_i > r_i$) **Then** *Задача не має розв'язку* **EndIf**

EndFor

// Відсів варіантів, що не проходять за обмеженням та знаходження допустимого розв'язку.

For ($j = 1; j \leq m; j++$) $sum := sum + v_j * \theta_{l_i, j};$ **EndFor**

For ($j = 1; j \leq m; j++$) $c_j := mRez - (sum - v_j * \theta_{l_i, j});$ **EndFor**

// Обчислення допусків за цільовою функцією (2.30)

```

For ( $j = 1; j \leq m; j++$ ) For ( $i = r_i; i \geq l_i; i--$ )
    If ( $\theta_{i,j} > c_j$ ) Then  $r_i := i - 1$ ; else  $flag := true$ ; Break EndIf
EndFor
    If ( $l_i > r_i$ ) Then Задача не має розв'язку EndIf
EndFor
// Відсів варіантів, що не проходять за цільовою функцією.
EndDo
Perebor ( $\theta$ );

```

// Цілеспрямованим перебором аналізуємо масив θ .

Роботу алгоритму розглянемо на такому прикладі.

Нехай спочатку розглядалася задача ранжування 18 альтернатив за чотирма показниками з ваговими коефіцієнтами $W = (0.5; 0.25; 0.2; 0.05)$. В табл. 2.3 наведений розв'язок цієї задачі.

Таблиця 2.3. Рейтинговий список альтернатив

	K_1	K_2	K_3	K_4	$G(A)$
A_1	98	91	87	83	93,3
A_2	89	94	90	90	90,5
A_3	95	81	79	33	85,2
A_4	95	52	77	80	79,9
A_5	78	75	73	43	74,5
A_6	93	45	47	88	71,55
A_7	70	87	64	17	70,4
A_8	56	97	70	53	68,9
A_9	50	63	93	77	63,2
A_{10}	51	98	51	54	62,9
A_{11}	75	38	33	62	56,7
A_{12}	50	60	35	64	50,2
A_{13}	40	57	56	63	48,6
A_{14}	54	40	37	55	47,15
A_{15}	29	90	41	22	46,3
A_{16}	27	64	26	57	37,55
A_{17}	33	22	67	39	37,35
A_{18}	24	20	42	98	30,3

З другого по четвертий стовпчик таблиці розміщені значення критеріїв K_1, K_2, K_3, K_4 для альтернатив A_1, A_2, \dots, A_{18} . Останній стовпчик містить значення узагальненого показника (2.24). Так, наприклад, $G(A_1) = 98 \cdot 0.5 + 91 \cdot 0.25 + 87 \cdot 0.2 + 83 \cdot 0.05 = 93.3$. Оберемо альтернативу A_{12} і сформулюємо для неї задачу (2.30)–(2.32) наступним чином: за яких мінімальних сумарних відхилень від значень $f_1(A_{12}) = 50, f_2(A_{12}) = 60, f_3(A_{12}) = 35, f_4(A_{12}) = 64$ з новим вектором переваг $V = (0.4; 0.1; 0.45; 0.05)$ можна забезпечити попадання A_{12} у першу трійку рейтингу – розв’язку початкової задачі (2.23)–(2.25)?

Припустимо, що ОПР сформувала множини відхилень $\theta_1, \theta_2, \theta_3, \theta_4$ у вигляді табл. 2.4. Так, наприклад,

$$\theta_3 = \{0; 11; 15; 22; 24; 28; 29; 32; 35; 37; 40; 44; 48; 51; 56; 65\}.$$

Оскільки альтернатива A_{12} повинна випередити A_3 , обчислимо бар’єр G^* , який необхідно подолати, щоб потрапити у першу трійку: $G^* = G(A_3) - G(A_{12}) = 85.2 - 50.2 = 35$.

Таблиця 2.4. Відхилення значень критеріїв для A_{12}

$i \backslash \theta$	θ_1^i	θ_2^i	θ_3^i	θ_4^i
15	46	31	65	36
14	42	30	56	33
13	39	29	51	30
12	35	27	48	29
11	31	25	44	27
10	28	24	40	25
9	26	22	37	23
8	24	20	35	21
7	21	18	32	20
6	17	15	29	17
5	12	13	28	15
4	10	11	24	12
3	7	8	22	8
2	5	7	15	5
1	3	4	11	2
0	0	0	0	0

Отже, в нашому випадку кількість всіх варіантів $|\theta| = |\theta_1| \cdot |\theta_2| \cdot |\theta_3| \cdot |\theta_4| = 16^4 = 65536$, оскільки для кожного з чотирьох критеріїв розглядається 16 можливих значень відхилень (включаючи 0).

Крок 0. При сформованих ОПР значеннях $\theta_1^{15}, \theta_2^{15}, \theta_3^{15}, \theta_4^{15}$ справджується нерівність $\sum_{j=1}^4 \omega_j \theta_j^{15} > G^*$ ($46 \cdot 0.5 + 31 \cdot 0.25 + 65 \cdot 0.2 + 36 \cdot 0.05 = 45.55 > 35$), тому задача має розв'язок. Домножимо кожне відхилення θ_j^l , $j = \overline{1,4}$, $l = \overline{0,15}$ (табл. 2.4) на відповідні компоненти векторів переваг:

$W = (0.5; 0.25; 0.2; 0.05)$ і $V = (0.4; 0.1; 0.45; 0.05)$. В результаті отримаємо табл. 2.5 і табл. 2.6 покладемо $\theta_{min}^* = \theta^{max} = (\theta_1^{15}, \theta_2^{15}, \theta_3^{15}, \theta_4^{15}) = (46, 31, 65, 36)$, $\theta^{min} = (\theta_1^0, \theta_2^0, \theta_3^0, \theta_4^0) = (0, 0, 0, 0)$, $\theta_1^{(0)} = \theta_1$, $\theta_2^{(0)} = \theta_2$, $\theta_3^{(0)} = \theta_3$, $\theta_4^{(0)} = \theta_4$, $t = 0$.

Таблиця 2.5. Зважені відхилення для A_{12} за вектором $W = (0.5; 0.25; 0.2; 0.05)$

$i \backslash \theta$	$w_1 \theta_1^i$	$w_2 \theta_2^i$	$w_3 \theta_3^i$	$w_4 \theta_4^i$
15	23	7,75	13	1,8
14	21	7,5	11,2	1,65
13	19,5	7,25	10,2	1,5
12	17,5	6,75	9,6	1,45
11	15,5	6,25	8,8	1,35
10	14	6	8	1,25
9	13	5,5	7,4	1,15
8	12	5	7	1,05
7	10,5	4,5	6,4	1
6	8,5	3,75	5,8	0,85
5	6	3,25	5,6	0,75
4	5	2,75	4,8	0,6
3	3,5	2	4,4	0,4
2	2,5	1,75	3	0,25
1	1,5	1	2,2	0,1
0	0	0	0	0

Початково покладемо

$\theta_{min}^* = \theta^{max} = (\theta_1^{15}, \theta_2^{15}, \theta_3^{15}, \theta_4^{15}) = (46, 31, 65, 36)$, $\theta^{min} = (\theta_1^0, \theta_2^0, \theta_3^0, \theta_4^0) = (0, 0, 0, 0)$, $\theta_1^{(0)} = \theta_1$, $\theta_2^{(0)} = \theta_2$, $\theta_3^{(0)} = \theta_3$, $\theta_4^{(0)} = \theta_4$, $t = 0$.

Таблиця 2.6. Зважені відхилення для A_{12} за вектором $V = (0.4; 0.1; 0.45; 0.05)$

$i \backslash$	$v_1 \theta_1^i$	$v_2 \theta_2^i$	$v_3 \theta_3^i$	$v_4 \theta_4^i$
15	18,4	3,1	29,25	1,8
14	16,8	3	25,2	1,65
13	15,6	2,9	22,95	1,5
12	14	2,7	21,6	1,45
11	12,4	2,5	19,8	1,35
10	11,2	2,4	18	1,25
9	10,4	2,2	16,65	1,15
8	9,6	2	15,75	1,05
7	8,4	1,8	14,4	1
6	6,8	1,5	13,05	0,85
5	4,8	1,3	12,6	0,75
4	4	1,1	10,8	0,6
3	2,8	0,8	9,9	0,4
2	2	0,7	6,75	0,25
1	1,2	0,4	4,95	0,1
0	0	0	0	0

Крок 2. Обчислимо допуски знизу за обмеженням (2.31):

$$d_j^{(t)} = G^* - \sum_{\substack{k=1 \\ k \neq j}}^4 w_k \cdot \theta_j^{max}, \quad j \in \{1, 2, 3, 4\}.$$

$$d_1^{(0)} = 35 - (31 \cdot 0.25 + 65 \cdot 0.2 + 36 \cdot 0.05) = 12.45$$

$$d_2^{(0)} = 35 - (46 \cdot 0.5 + 65 \cdot 0.2 + 36 \cdot 0.05) = -2.8$$

$$d_3^{(0)} = 35 - (46 \cdot 0.5 + 31 \cdot 0.25 + 36 \cdot 0.05) = 2.45$$

$$d_4^{(0)} = 35 - (46 \cdot 0.5 + 31 \cdot 0.25 + 65 \cdot 0.2) = -8.75$$

Визначимо множини елементів, що залишились:

$$\tilde{\theta}_j^{(t+1)} = \{\theta_j \mid w_j \theta_j \geq d_j^{(t)}, \quad \theta_j \in \Theta_j^{(t)}\}, \quad j \in \{1, 2, 3, 4\}.$$

$$\tilde{\theta}_1^{(1)} = \{26, 28, 31, 35, 39, 42, 46\}$$

$$\tilde{\theta}_2^{(1)} = \{0, 4, 7, 8, 11, 13, 15, 18, 20, 22, 24, 25, 27, 29, 30, 31\}$$

$$\tilde{\theta}_3^{(1)} = \{15, 22, 24, 28, 29, 32, 35, 37, 40, 44, 48, 51, 56, 65\}$$

$$\tilde{\theta}_4^{(1)} = \{0, 2, 5, 8, 12, 15, 17, 20, 21, 23, 25, 27, 29, 30, 33, 36\}$$

В табл. 2.5 відсіяні зважені значення відхилень зображені на сірому фоні.

Покладемо $\theta_j^{min} = \min \{ \theta_j \mid \theta_j \in \tilde{\theta}_j^{(t+1)} \}$, $j \in \{1, 2, 3, 4\}$: $\theta^{min} = (26, 0, 15, 0)$.

Крок 3. Знаходимо поточний допустимий розв'язок θ^* :

$$\theta^* = \operatorname{argmin}_{j \in \{1, 2, 3, 4\}} \{ v_j \cdot \theta_j^{min} + \sum_{\substack{k=1 \\ k \neq j}}^4 v_k \cdot \theta_j^{max} \}.$$

В нашому випадку після проведення відповідних обчислень отримаємо:

$$\begin{aligned} \theta^* &= \operatorname{argmin} \{ 0.4 \cdot 26 + 0.1 \cdot 31 + 0.45 \cdot 65 + 0.05 \cdot 36; 0.1 \cdot 0 + 0.4 \cdot 46 + 0.45 \\ &\quad \cdot 65 + 0.05 \cdot 36; 0.45 \cdot 15 + 0.4 \cdot 46 + 0.1 \cdot 31 + 0.05 \cdot 36; 0.05 \cdot 0 \\ &\quad + 0.4 \cdot 46 + 0.1 \cdot 31 + 0.45 \cdot 65 \} \\ &= \operatorname{argmin} \{ 44.55; 49.45; \mathbf{30.05}; 50.75 \} = (46, 31, 15, 36) \end{aligned}$$

Отже, $h(\theta^*, V) = (0.4 \cdot 46 + 0.1 \cdot 31 + 0.45 \cdot 15 + 0.05 \cdot 36) = 30.05$,

$$h(\theta_{min}^*, V) = (46 \cdot 0.4 + 31 \cdot 0.1 + 65 \cdot 0.45 + 36 \cdot 0.05) = 52.55.$$

Оскільки $h(\theta^*, V) < h(\theta_{min}^*, V)$, покладемо $\theta_{min}^* = \theta^* = (46, 31, 15, 36)$.

Крок 4. Обчислимо допуски зверху за значенням цільової функції:

$$c_j^{(t)} = h(\theta_{min}^*, V) - \sum_{\substack{k=1 \\ k \neq j}}^4 v_k \cdot \theta_j^{min}, \quad j \in \{1, 2, 3, 4\}.$$

$$c_1^{(0)} = 30.05 - (0 \cdot 0.4 + 0 \cdot 0.1 + 15 \cdot 0.45 + 0 \cdot 0.05) = 23.35$$

$$c_2^{(0)} = 30.05 - (26 \cdot 0.4 + 0 \cdot 0.1 + 15 \cdot 0.45 + 0 \cdot 0.05) = 12.9$$

$$c_3^{(0)} = 30.05 - (26 \cdot 0.4 + 0 \cdot 0.1 + 0 \cdot 0.45 + 0 \cdot 0.05) = 19.65$$

$$c_4^{(0)} = 30.05 - (26 \cdot 0.4 + 0 \cdot 0.1 + 15 \cdot 0.45 + 0 \cdot 0.05) = 12.9$$

Визначимо множини елементів, що залишились:

$$\theta_j^{(t+1)} = \{ \theta_j \mid v_j \theta_j \leq c_j^{(t)}, \quad \theta_j \in \tilde{\theta}_j^{(t+1)} \}, \quad j \in \{1, 2, 3, 4\}.$$

$$\theta_1^{(1)} = \{26, 28, 31, 35, 39, 42, 46\}$$

$$\theta_2^{(1)} = \{0, 4, 7, 8, 11, 13, 15, 18, 20, 22, 24, 25, 27, 29, 30, 31\}$$

$$\theta_3^{(1)} = \{15, 22, 24, 28, 29, 32, 35, 37, 40\}$$

$$\theta_4^{(1)} = \{0, 2, 5, 8, 12, 15, 17, 20, 21, 23, 25, 27, 29, 30, 33, 36\}.$$

В табл. 2.6 додалися відсіянні зверху зважені значення відхилень за третім критерієм (зображені на сірому фоні).

Покладемо $\theta_j^{max} = \max \{ \theta_j \mid \theta_j \in \theta_j^{(t+1)} \}$, $j \in \{1, 2, 3, 4\}$: $\theta^{max} = (46, 31, 40, 36)$

Таким чином, після першої ітерації вдалося відсіяти *знизу* значення відхилень по першому та третьому критерію і *зверху* – по третьому (табл. 2.7).

Таблиця 2.7. Відхилення значень критеріїв для A_{12} , що залишились після першої ітерації алгоритму.

$i \backslash \theta$	θ_1^i	θ_2^i	θ_3^i	θ_4^i
15	46	31		36
14	42	30		33
13	39	29		30
12	35	27		29
11	31	25		27
10	28	24	40	25
9	26	22	37	23
8		20	35	21
7		18	32	20
6		15	29	17
5		13	28	15
4		11	24	12
3		8	22	8
2		7	15	5
1		4		2
0		0		0

Крок 5. Оскільки відсіювання відбулось за двома критеріями (достатньо одного) – $\theta_1^{(1)} \neq \theta_1^{(0)}$, $\theta_3^{(1)} \neq \theta_3^{(0)}$, покладемо $t := t + 1$ і переходимо на крок 1 (на наступну ітерацію алгоритму).

В нашому прикладі алгоритм відпрацює шість ітерацій. В результаті отримаємо множину допустимих розв'язків, представлену в табл. 2.8.

Таблиця 2.8. Остаточні значення відхилень критеріїв для A_{12} після завершення алгоритму.

i	θ_1^i	θ_2^i	θ_3^i	θ_4^i
15	46	31		36
14	42	30		33
13		29		30
12		27		29
11		25		27
10		24		25
9		22		23
8				21
7				20
6				17
5				15
4			24	12
3			22	8
2			15	5
1				2
0				0

Якщо спочатку $|\theta| = |\theta_1| \cdot |\theta_2| \cdot |\theta_3| \cdot |\theta_4| = 16^4 = 65536$, то після реалізованого алгоритмом відсіювання безперспективних варіантів отримали $|\theta| = 2 \cdot 7 \cdot 3 \cdot 16 = 672$, отже розмірність задачі (2.30)–(2.32) зменшилася майже на два порядки. Розв'язок $\theta = (46, 31, 15, 27)$, $h(\theta, V) = 29,6$.

2.3.3. Розв'язання оберненої задачі ранжування альтернатив методом динамічного програмування

Якщо в оберненій задачі ранжування альтернатив ОПР не вводить новий вектор переваг $V = (v_1, v_2, \dots, v_m)$, то в задачі (2.30)–(2.32) цільова функція може розглядатись в іншому вигляді:

$$H(a', \theta, p, W) = \left(G(f(a', \theta), W) - G(f(A_{i_p}), W) \right) \rightarrow \min \quad (2.35)$$

Нехай для розв'язку задач (2.23)–(2.25) в якості узагальненого критерію використовується лінійно-адитивна згортка. Тоді функція $G(\cdot)$ в (2.24) буде мати вигляд:

$$G(x) = G(f(x), W) = \sum_{j \in J_1} \omega_j (f_j(x)) - \sum_{j \in J_2} \omega_j (f_j(x)),$$

де J_1 і J_2 – множини індексів критеріїв, що відповідно максимізуються і мінімізуються ($J_1 \cup J_2 = J$), а функція $H(\cdot)$ в (2.35) –

$$\begin{aligned} H(a', \theta, p, W) &= \left(\sum_{j \in J_1} \omega_j (f_j(a') + \theta_j) - \sum_{j \in J_2} \omega_j (f_j(a') + \theta_j) \right) - \\ &\quad - \left(\sum_{j \in J_1} \omega_j (f_j(A_{i_p})) - \sum_{j \in J_2} \omega_j (f_j(A_{i_p})) \right) = \\ &= \sum_{j \in J_1} \omega_j (f_j(A') + \theta_j - f_j(A_{i_p})) - \sum_{j \in J_2} \omega_j (f_j(A') + \theta_j - f_j(A_{i_p})). \end{aligned} \quad (2.36)$$

Для степеневно-адитивної згортки критеріїв маємо:

$$\begin{aligned} G(x) &= G(f(x), W) = \sum_{j \in J_1} (f_j(x))^{\omega_j} - \sum_{j \in J_2} (f_j(x))^{\omega_j}, \\ H(A', \theta, p, W) &= \left(\sum_{j \in J_1} (f_j(A') + \theta_j)^{\omega_j} - \sum_{j \in J_2} (f_j(A') + \theta_j)^{\omega_j} \right) - \\ &\quad - \left(\sum_{j \in J_1} (f_j(A_{i_p}))^{\omega_j} - \sum_{j \in J_2} (f_j(A_{i_p}))^{\omega_j} \right) = \\ &= \sum_{j \in J_1} ((f_j(A') + \theta_j)^{\omega_j} - f_j(A_{i_p})^{\omega_j}) - \sum_{j \in J_2} ((f_j(A') + \theta_j)^{\omega_j} - f_j(A_{i_p})^{\omega_j}). \end{aligned}$$

Якщо задача (2.23)–(2.25) розв'язується методом ідеальної точки, і A^* – ідеальна альтернатива (точка), то

$$G(x) = G(f(x), W) = \left(\sum_{j \in J} \omega_j (f_j(x) - f_j(A^*))^2 \right)^{\frac{1}{2}}.$$

Проте у такому вигляді функція $G(x)$ не відповідає вимогам (2.25), оскільки для даного випадку кращими будуть не більші, а менші значення $G(x)$ (мінімізується відстань до ідеальної точки), тому для коректності використання моделі (2.23)–(2.25) модифікуємо $G(x)$ наступним чином:

$$G(x) = \tilde{G} - \left(\sum_{j \in J} \omega_j (f_j(x) - f_j(A^*))^2 \right)^{\frac{1}{2}},$$

де

$$\tilde{G} = \left(\sum_{j \in J_1} \omega_j (\min_{x \in X} f_j(x) - f_j(A^*))^2 + \sum_{j \in J_2} \omega_j (\max_{x \in X} f_j(x) - f_j(A^*))^2 \right)^{\frac{1}{2}},$$

тобто \tilde{G} – відстань між ідеальною найгіршою і ідеальною найкращою альтернативами. Тоді

$$\begin{aligned} H(A', \theta, p, W) &= G(f(A', \theta), W) - G(f(A_{i_p}), W) = \\ &= \left(\sum_{j \in J} \omega_j (f_j(A_{i_p}) - f_j(A^*))^2 \right)^{\frac{1}{2}} - \left(\sum_{j \in J} \omega_j (f_j(A') + \theta_j - f_j(A^*))^2 \right)^{\frac{1}{2}}. \end{aligned}$$

Розглянемо детально алгоритм розв'язання задачі (2.35) з обмеженнями (2.31), (2.32) для випадку застосування лінійно-адитивної згортки критеріїв в (2.24).

Запишемо (2.35) у вигляді

$$\begin{aligned} H(A', \theta, p, W) &= \sum_{j \in J_1} \omega_j \theta_j - \sum_{j \in J_2} \omega_j \theta_j + \\ &+ \sum_{j \in J_1} \omega_j (f_j(A') - f_j(A_{i_p})) - \sum_{j \in J_2} \omega_j (f_j(A') - f_j(A_{i_p})) = \\ &= \sum_{j \in J_1} \omega_j \theta_j - \sum_{j \in J_2} \omega_j \theta_j - \left(\sum_{j \in J_1} \omega_j (f_j(A_{i_p})) - \sum_{j \in J_2} \omega_j (f_j(A_{i_p})) \right) + \\ &+ \left(\sum_{j \in J_1} \omega_j (f_j(A')) - \sum_{j \in J_2} \omega_j (f_j(A')) \right) = \end{aligned}$$

$$= \sum_{j \in J_1} \omega_j \theta_j - \sum_{j \in J_2} \omega_j \theta_j - \left(G(A_{i_p}) - G(A') \right) = h(\theta_1, \theta_2, \dots, \theta_m) - G^*. \quad (2.37)$$

В отриманій формулі $h(\theta_1, \theta_2, \dots, \theta_m) = \sum_{j \in J_1} \omega_j \theta_j - \sum_{j \in J_2} \omega_j \theta_j$, а G^* – константне значення, яке задає початкову перевагу альтернативи A_{i_p} над A' , тобто $G^* = G(A_{i_p}) - G(A')$. Відповідно до (2.25) G^* є не від'ємним числом.

В множині (2.29) для $j \in J_1$ $0 < \theta_j^1 < \theta_j^2 < \dots < \theta_j^{r_j}$, а для $j \in J_2$ $0 > \theta_j^1 > \theta_j^2 > \dots > \theta_j^{r_j}$.

Мета алгоритму – знайти такий вектор $\theta = (\theta_1, \dots, \theta_m)$, який мінімізує $h(\theta_1, \theta_2, \dots, \theta_m)$ серед всіх можливих наборів θ , для яких (2.37) приймає невід'ємне значення. Для критеріїв, що мінімізуються, елементи множини $\{\theta_j^1, \theta_j^2, \dots, \theta_j^{r_j}\}$ перевизначимо так: $\theta_j^1 := -\theta_j^1, \theta_j^2 := -\theta_j^2, \dots, \theta_j^{r_j} := -\theta_j^{r_j}$, де “:=” розглядається як знак операції переприсвоєння значень. Тоді (2.37) можна переписати у вигляді

$$H(A', \theta, p, W) = h(\theta_1, \theta_2, \dots, \theta_m) - G^* = \sum_{j=1}^m \omega_j \theta_j - G^*. \quad (2.38)$$

Алгоритм базується на ідеології методу динамічного програмування [5]. У відповідності з ним визначимо рекурентне співвідношення, за яким початкова задача зводиться до задач меншої розмірності. Введемо позначення $\theta^k = (\theta_1, \theta_2, \dots, \theta_k)$ – вектор відхилень для перших k критеріїв, $k \in \{1, 2, \dots, m\}$.

Маємо

$$H(A', \theta^m, p, W) = \sum_{j=1}^m \omega_j \theta_j - \left(G(A_{i_p}) - G(A') \right),$$

$$H(A', \theta^m, p, W) = \omega_m \theta_m + H(A', \theta^{m-1}, p, W) - G^*,$$

$$H(A', \theta^{m-1}, p, W) = \omega_{m-1} \theta_{m-1} + H(A', \theta^{m-2}, p, W) - (G^* - \omega_m \theta_m),$$

...

$$H(A', \theta^k, p, W) = \omega_k \theta_k + H(A', \theta^{k-1}, p, W) - \left(G^* - \sum_{j=k+1}^m \omega_j \theta_j \right) \quad (2.39)$$

...

$$H(A', \theta^1, p, W) = \omega_1 \theta_1 - \left(G^* - \sum_{j=2}^m \omega_j \theta_j \right).$$

Початково G^* є значенням наскільки альтернатива A' “гірша” від A_{i_p} при врахуванні всіх m критеріїв.

Опишемо складові частини формули (2.39) для задачі розмірності k . Вираз в дужках $G^* - \sum_{j=k+1}^m \omega_j \theta_j$ означає наскільки вдалося зменшити “відставання” альтернативи A' від A_{i_p} , отримавши значення $\theta_{k+1}, \theta_{k+2}, \dots, \theta_m$ на початкових $m - k$ кроках алгоритму; $H(A', \theta^{k-1}, p, W)$ – мінімальне додатне значення шуканої переваги альтернативи A' над A_{i_p} з врахуванням тільки $k - 1$ перших критеріїв. У доданку $\omega_k \theta_k$ значення θ_k обирається з множини $\{\theta_k^1, \theta_k^2, \dots, \theta_k^{r_k}\}$.

Тепер запишемо рекурентне співвідношення, за яким початкова задача зводиться до задач меншої розмірності

$$\tilde{H}(k, \tilde{G}, \Theta, W) = \begin{cases} \min_{\theta_k \in \Theta_k} \left(\omega_k \theta_k + \tilde{H}(k-1, \tilde{G} - \omega_k \theta_k, \Theta, W) \right), & \tilde{G} \geq 0, k > 0 \\ 0, & \tilde{G} < 0, k = 0 \\ \infty, & \tilde{G} \geq 0, k = 0 \end{cases} \quad (2.40)$$

Отримане співвідношення можна посилити враховуючи верхню оцінку підзадач меншої розмірності.

$$\tilde{H}(k, \tilde{G}, \Theta, W) = \begin{cases} \min_{\theta_k \in \Theta_k} \left(\omega_k \theta_k + \tilde{H}(k-1, \tilde{G} - \omega_k \theta_k, \Theta, W) \right), & \tilde{G} \geq 0, k > 0 \\ 0, & \tilde{G} < 0, k \geq 0 \\ \infty, & \tilde{G} \geq \sum_{i=1}^k \omega_i \max \Theta_i, k \geq 0 \end{cases} \quad (2.41)$$

Рекурентне співвідношення (2.41) дозволяє побудувати алгоритм, який за поліноміальний час знаходить наближений розв’язок \tilde{H}' з заданою точністю ε для обмеження (2.31), тобто $\sum_{i=1}^k \omega_i \theta_i > G^* + \varepsilon$, де θ – знайдений розв’язок.

Далі наводиться алгоритм знаходження точного розв'язку з використанням верхньої та нижньої оцінки підзадач меншої розмірності, що описані в рекурентному співвідношенні (2.41).

Для ефективної роботи алгоритму розв'язання задачі (2.35), (2.31), (2.32) необхідно попередньо виконати декілька процедур.

- По кожному з m критеріїв визначимо середньоарифметичні значення

$$\text{зважених відхилень } \tilde{\theta}_k = \frac{\sum_{i=1}^{r_k} \omega_k \theta_k^i}{r_k}, k = \overline{1, m}.$$

- Впорядкуємо критерії загальної задачі за зростанням отриманих значень $\tilde{\theta}_k, k = \overline{1, m}$. Для запобігання переприсвоєння номерів критеріїв будемо вважати, що у нас початково виконується нерівність $\tilde{\theta}_1 \leq \tilde{\theta}_2 \leq \dots \leq \tilde{\theta}_m$. Це не зменшує узагальненості задачі.
- Створюємо послідовність хешованих сум S_k за наступним правилом:

$$S_1 = 0, S_2 = S_1 + \theta_1^{r_1}, \dots, S_k = S_{k-1} + \theta_{k-1}^{r_{k-1}}, \dots, S_m = S_{m-1} + \theta_{m-1}^{r_{m-1}}.$$

Далі створимо таблицю, що буде складатися з m стовпчиків значень (знизу верх) $0, \omega_k \theta_k^1, \omega_k \theta_k^2, \dots, \omega_k \theta_k^{r_{jk}}$ для кожного критерія і додамо до неї знизу рядок отриманих хешованих сум (табл. 2.9).

Таблиця 2.9. Таблиця параметрів, що ілюструє можливі покращення

	K_1	K_2	...	K_k	...	K_{m-1}	K_m
$\max_{j \in \{1, \dots, m\}} r_j$		$\omega_2 \theta_2^{r_2}$		\vdots		\vdots	
\vdots	\vdots			$\omega_k \theta_k^{r_k}$			\vdots
		\vdots	$\omega_{m-1} \theta_{m-1}^{r_{m-1}}$	
	$\omega_1 \theta_1^{r_1}$			\vdots		\vdots	$\omega_m \theta_m^{r_m}$
	\vdots						\vdots
2	$\omega_1 \theta_1^2$	$\omega_2 \theta_2^2$...	$\omega_k \theta_k^2$...	$\omega_{m-1} \theta_{m-1}^2$	$\omega_m \theta_m^2$
1	$\omega_1 \theta_1^1$	$\omega_2 \theta_2^1$...	$\omega_k \theta_k^1$...	$\omega_{m-1} \theta_{m-1}^1$	$\omega_m \theta_m^1$
0	0	0	...	0	...	0	0
Хеш	S_1	S_2	...	S_k	...	S_{m-1}	S_m

В цій таблиці $r_2 = \max(r_j)$, $j \in \{1, \dots, m\}$, $r_k > r_{m-1} > r_1$, тобто стовпчики мають різну висоту в залежності від кількості можливих відхилень того чи іншого критерію.

Алгоритм дозволяє генерувати допустимі розв'язки задачі (2.31)-(2.32),(2.35) без повного перебору всіх можливих варіантів і словесно описується так.

1. Якщо сума $S_m + \theta_{m,r_m} * \omega_m$ менша від G^* , то задача (2.31)-(2.32),(2.35) не має розв'язку.
2. Ініціалізація. Задаються початкові значення $k = m$, $g_k = G^*$, $l_k = 0$, $\theta = (0, \dots, 0)$, $\theta^{min} = \theta$. Тут і далі в алгоритмі k – номер критерія, що аналізується, g_k – “не подолана” перевага A_{i_p} над A' , що залишилась на момент розгляду підзадачі розмірності k , l_k – індекс зваженого відхилення критерія під номером k ($0 \leq l_k \leq r_k$), $\theta = (\theta_1, \theta_2, \dots, \theta_k, \dots, \theta_m)$ – варіант розв'язку задачі. Початково при ($k = m$) $\theta_1 = 0, \theta_2 = 0, \dots, \theta_m = 0$. В будь-який момент роботи алгоритму вже отриманому мінімальному значенню $H(A', \theta, p, W)$ відповідає розв'язок θ^{min} .
3. Для підзадачі розмірності k розглядається стовпчик зважених відхилень k -го критерія. Індекс l_k збільшується до тих пір, поки зважене відхилення $\omega_k \theta_k^{l_k}$ разом з хешованою сумою S_k не стане більшим від g_k . Слід звернути увагу на те, що S_k – максимально можливе значення, на яке може зменшитись перевага A_{i_p} над A' при вже зафіксованих на попередніх кроках відхиленнях $\theta_{k+1} = \theta_{k+1}^{l_{k+1}}, \theta_{k+2} = \theta_{k+2}^{l_{k+2}}, \dots, \theta_m = \theta_m^{l_m}$. Можливі три випадки.
 - 3.1. Сума максимально можливого зваженого відхилення $\omega_k \theta_k^{r_k} + S_k$ менша від g_k . Переходимо на крок 3.
 - 3.2. Зважене відхилення $\omega_k \theta_k^{l_k}$ без хешованої суми S_k більше від g_k . Це означає, що отримано розв'язок $\theta = (0, \dots, 0, \theta_k^{l_k}, \theta_{k+1}^{l_{k+1}}, \dots, \theta_{m-1}^{l_{m-1}}, \theta_m^{l_m})$. Переходимо на крок 4.

3.3. Зважене відхилення $\omega_k \theta_k^{l_k}$ разом з хешованою сумою S_k більше від g_k .

Переходимо на крок 5.

4. Якщо $k = m$, переходимо на крок 6, інакше – повертаємось до $(k + 1)$ -го критерія – $k = k + 1$. Переходимо на крок 2.
5. Якщо значення $H(A', \theta, p, W)$ для отриманого розв'язку є меншим від $H(A', \theta^{min}, p, W)$, то $\theta^{min} = (0, \dots, 0, \theta_k^{l_k}, \theta_{k+1}^{l_{k+1}}, \dots, \theta_{m-1}^{l_{m-1}}, \theta_m^{l_m})$. Переходимо на крок 3.
6. Розмірність задачі зменшується – $k = k - 1, l_k = 0$. Переходимо на крок 2.
7. θ^{min} – оптимальний розв'язок задачі (2.31)-(2.32),(2.35).

Формально описаний алгоритм можна представити так. (Об'єкти, що використовуються в алгоритмі, повністю відповідають визначеним вище структурам даним).

/* k, m, i, j : цілочисельні невід'ємні значення,

Rez – значення цільової функції: дійсне,

G^* – початкове відхилення A_{i_p} від A' : дійсне,

$g, \omega, S, \theta^{min}$ – масиви розмірності m : дійсні числа,

l, r – масиви розмірності m : цілочисельні невід'ємні значення,

θ – двомірний масив з m стовпчиків різної висоти: дійсні числа,

*/

$S_1 = 0$; // Обчислення хешованих сум $S_k, k = \overline{1, m}$

For ($i = 2; i \leq m; i++$) $S_i = S_{i-1} + \theta_{i-1, r_{i-1}} * \omega_{i-1}$ **EndFor**

$Rez := S_m + \theta_{m, r_m} * \omega_m - G^*$; // Початкове значення цільової функції

If ($Rez < 0$) **Then Exit EndIf** // Задача не має розв'язку.

// Ініціалізація

$k := m; g_k := G^*$;

For ($i = 1; i \leq m; i++$) $l_i := 0; \theta_i^{min} := 0$; **EndFor**

```

Do while ( $k \leq m$ )
   $j := r_k + 1$ ;
  For ( $i = l_k; i \leq r_k; i++$ )
    If ( $S_k + \omega_k \theta_{k,i} - g_k > 0$ ) Then  $j := i$ ; Break EndIf
  EndFor
   $l_k := j$ ;
  If ( $l_k \leq r_k$ ) Then
    If ( $(\omega_k \theta_{k,l_k} - g_k > 0)$  Then
      If ( $(\omega_k \theta_{k,l_k} - g_k < Rez)$ ) Then
        For ( $i = 1; i \leq m; i++$ )  $\theta_i^{min} = \theta_{i,l_i}$  EndFor
         $Rez := \omega_k \theta_{k,l_k} - g_k$ ;
      EndIf
     $l_k := 0; k := k + 1; l_k := l_k + 1$ ;
    Else  $g_{k-1} := g_k - \theta_{k,l_k}; k := k - 1; l_k := 0$ ;
    EndIf
  Else  $l_k := 0; k := k + 1; l_k := l_k + 1$ ;
  Endif
EndDo

```

Розроблений алгоритм розв'язання задачі (2.31)-(2.32),(2.35) на основі ідеології динамічного програмування дозволяє знаходити оптимальний набір параметрів (відхилень значень критеріїв), що фактично призводить до формулювання мінімальних вимог для обраної альтернативи з метою досягнення нею кращих результатів ранжування.

Відмітимо, що цей алгоритм може бути використаний не тільки для методу зважених сум, а й також у випадку застосування степеневно-адитивної згортки або методу аналізу ієрархій, оскільки в АНР також застосовується лінійно-адитивна згортка.

Висновки за розділом 2

У другому розділі описано онтологічну модель задачі ранжування, розроблений алгоритм конкурентної нормалізації критеріїв, наведені постановка та алгоритми розв'язання оберненої задачі ранжування.

Застосування онтологій в задачі ранжування забезпечується виконанням умов існування не пустого перетворення таксономії в множину альтернатив та не пустого відображення множини властивостей об'єктів, що складають онтологію у множину критеріїв задачі ранжування. Тоді для формування моделі задачі ранжування на основі онтології стає можливим визначення певного перетворення.

Розроблено метод побудови інформаційного середовища задачі ранжування, що здійснює перетворення онтологічної моделі предметної області на основі інтерпретаційних функцій вибору, побудованих за допомогою гіпервідношень над елементами таксономічної структури онтології та властивостями її об'єктів.

Розроблено алгоритм конкурентної нормалізації критеріїв для задач ранжування та рейтингового оцінювання, що враховує конкурентність процесу встановлення ступеня домінування одних альтернатив над іншими в залежності від їх статистичних характеристик.

Сформульована обернена задача ранжування та розроблено алгоритми її розв'язання, що дозволяє на етапі пост-аналізу процесу проведення вибору (ранжування) альтернатив підвищити рівень аналізу отриманих результатів та розширити початкову онтологічну модель предметної області новими властивостями об'єктів дослідження.

Основні результати розділу опубліковано в [21, 22, 24, 25, 26, 27, 30, 80, 86, 118, 119].

РОЗДІЛ 3. АРХІТЕКТУРА СИСТЕМИ РОЗВ'ЯЗКУ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ

Програмна система розв'язку задач ранжування альтернатив за сукупністю показників є системою, призначеною для вирішення задачі ранжування на основі онтологічної моделі предметної області.

З точки зору програмної інженерії програмна система розглядається у вигляді набору описів, представлених у вигляді математичних моделей, формалізмів і технік моделювання [102, 103].

Структура математичних моделей ПС такого роду включає в себе наступні моделі [102, 103]:

- 1) інформаційна модель;
- 2) функціонально-компонентна модель.

3.1. Інформаційна модель програмної системи розв'язку задач ранжування альтернатив

Інформаційна модель використовується для представлення і опису потоків інформації, структур даних а також програмних модулів в програмній системі.

Узагальнена інформаційна модель програмної системи розв'язку задач ранжування S_A має вигляд (3.1). Вона представляється деякою скінченною сукупністю програмних модулів S_{A_i} , що інтегруються в інформаційно-аналітичну систему ТОДОС.

$$S_A = \sum_{i=1}^n S_{A_i} \quad (3.1)$$

Сукупність модулів можна розглядати як окрему незалежну систему від ТОДОС. Дана сукупність має вигляд (3.2).

$$S_A = \{S_{MO}, S_{MT}, S_{OA}, S_{AL}, S_{CO}\} \quad (3.2)$$

S_{MO} – модуль, що відповідає за інформаційне представлення онтологічного графу та базові операції над ним. Множина функцій даного модуля має вигляд (3.3).

$$F_{MO} = \{F_1^{SMO}, F_2^{SMO}, F_3^{SMO}, F_4^{SMO}, F_5^{SMO}, F_6^{SMO}, F_7^{SMO}, F_8^{SMO}, F_9^{SMO}\} \quad (3.3)$$

До його функцій відносяться:

- 1) F_1^{SMO} – завантаження онтологічної моделі із підтримуваних форматів даних (XML та JSON).
- 2) F_2^{SMO} – знаходження вузла графу за його унікальним ідентифікатором
- 3) F_3^{SMO} – знаходження вузла (*та його унікального ідентифікатора*) за іменем.
- 4) F_4^{SMO} – долучення вузла до графу.
- 5) F_5^{SMO} – вилучення вузла із графу.
- 6) F_6^{SMO} – знаходження дочірніх елементів певного вузла графу
- 7) F_7^{SMO} – знаходження батьківських елементів певного вузла графу
- 8) F_8^{SMO} – читання та редагування властивостей вершини онтологічного графу
- 9) F_9^{SMO} – вивантаження онтологічної моделі в один із підтримуваних форматів (*JSON*)

S_{MT} – модуль, що відповідає за модель інформаційного представлення задачі ранжування альтернатив. Цей модуль складається з S_{MTA} та S_{MTC} (3.4).

$$S_{MT} = \{S_{MTA}, S_{MTC}\} \quad (3.4)$$

S_{MTA} – модуль, що відповідає за модель інформаційного представлення альтернатив у задачі ранжування. Множина функцій даного модуля має вигляд (3.5).

$$F_{MTA} = \{F_1^{SMTA}, F_2^{SMTA}, F_3^{SMTA}, F_4^{SMTA}, F_5^{SMTA}, F_6^{SMTA}\} \quad (3.5)$$

До його функцій відносяться:

- 1) $F_1^{S_{MTA}}$ – долучення альтернативи до даного контейнера
- 2) $F_2^{S_{MTA}}$ – вилучення альтернатив(и)
- 3) $F_3^{S_{MTA}}$ – знаходження альтернативи за її унікальним ідентифікатором
- 4) $F_4^{S_{MTA}}$ – знаходження альтернативи (*та його унікального ідентифікатора*) за іменем.
- 5) $F_5^{S_{MTA}}$ – завантаження та вивантаження об'єктів із/у підтримувані формати даних
- 6) $F_5^{S_{MTA}}$ – читання, встановлення та редагування властивостей об'єктів (альтернатив), що можуть бути задіяні як критеріальні значення чи дані для їх обчислень.

S_{MTC} – модуль, що відповідає за модель інформаційного представлення критеріїв у задачі ранжування. Множина функцій даного модуля має вигляд (3.6).

$$F_{MTC} = \{F_1^{S_{MTC}}, F_2^{S_{MTC}}, F_3^{S_{MTC}}, F_4^{S_{MTC}}, F_5^{S_{MTC}}\} \quad (3.6)$$

До його функцій відносяться:

- 1) $F_1^{S_{MTC}}$ – створення критеріальних об'єктів за допомогою вхідного списку.
- 2) $F_2^{S_{MTC}}$ – долучення критерію до даного контейнера.
- 3) $F_3^{S_{MTC}}$ – вилучення критерія(*ів*).
- 4) $F_4^{S_{MTC}}$ – редагування критеріїв (встановлення ваг, напряму оптимізації та інше).
- 5) $F_5^{S_{MTC}}$ – знаходження критерію за його унікальним ідентифікатором.

S_{OA} – модуль, що відповідає за аналіз онтологічного графу на основі його інформаційної моделі, що представлена модулем S_{MO} . Цей модуль складається з S_{TA} та S_{PI} (3.7).

$$S_{OA} = \{S_{TA}, S_{PI}\} \quad (3.7)$$

S_{TA} – модуль, що відповідає за таксономічний аналіз онтографу та виокремлення підмножини вершин, що можуть бути задіяні, як альтернативи. Множина функцій даного модуля має вигляд (3.8).

$$F_{TA} = \{F_1^{STA}, F_2^{STA}, F_3^{STA}, F_4^{STA}\} \quad (3.8)$$

До його функцій відносяться:

- 1) F_1^{STA} – Визначення альтернатив на основі таксономії. За замовчуванням такими вважаються однорідні об'єкти нижнього рівня ієрархії.
- 2) F_2^{STA} – маркування допустимих вершин на основі початкової множини «білих» вузлів, тобто тільки їх дочірні вузли мають бути включені у розгляд.
- 3) F_3^{STA} – маркування допустимих вершин на основі початкової множини «чорних» вузлів, тобто всі їх дочірні вузли мають бути виключені з розгляду.
- 4) F_4^{STA} – фільтрація вершин на основі вузла(*ів*) більш високого рівня ієрархії.

S_{PI} – модуль, що відповідає за знаходження властивостей об'єктів, які можуть бути застосовані для критеріальних оцінок. Множина функцій даного модуля має вигляд (3.9).

$$F_{PI} = \{F_1^{SPI}, F_2^{SPI}\} \quad (3.9)$$

До його основних функцій відноситься:

- 1) F_1^{SPI} – Виокремлення «критеріальних» властивостей та їх характеристик із вершин онтографу.
- 2) F_2^{SPI} – Створення (або оновлення) критеріїв у моделі задачі ранжування.

S_{AL} – набір модулів, що відповідають за безпосереднє розв'язання задачі ранжування.

Множина функцій даного модуля S_{AL} має вигляд (3.10).

$$F_{AL} = \{F_1^{S_{AL}}, F_2^{S_{AL}}, F_3^{S_{AL}}, F_4^{S_{AL}}, F_5^{S_{AL}}\} \quad (3.10)$$

До його основних функцій відноситься:

- 1) $F_1^{S_{AL}}$ – обчислення вагових коефіцієнтів критеріїв.
- 2) $F_2^{S_{AL}}$ – обчислення критеріальних значень, якщо для критеріїв задано спосіб їх інтерпретації.
- 3) $F_3^{S_{AL}}$ – нормалізація критеріальних значень альтернатив в спільну шкалу.
- 4) $F_4^{S_{AL}}$ – функції обчислення (в залежності від конкретного методу) для обчислення «рейтингового» значення альтернатив, що застосовуються для їх ранжування.
- 5) $F_5^{S_{AL}}$ – ранжування та встановлення певного «рейтингового» значення альтернативам на основі агрегації результату обчислень задіяних для розв'язку математичних методів ранжування альтернатив.

S_{CL} – набір модулів (контролерів), що відповідають за підготовку даних для відображення у веб-інтерфейсі користувача та за обробку його запитів до системи. При необхідності контролери для виконання своїх завдань використовують інші допоміжні модулі-сервіси. Множина функцій даних модулів має вигляд (3.11).

$$F_{CL} = \{F_1^{S_{CL}}, F_2^{S_{CL}}, F_3^{S_{CL}}, F_4^{S_{CL}}, F_5^{S_{CL}}, F_6^{S_{CL}}, F_7^{S_{CL}}, F_8^{S_{CL}}\} \quad (3.11)$$

До їх основних функцій відноситься:

- 1) $F_1^{S_{CL}}$ – визначення множини альтернатив.
- 2) $F_2^{S_{CL}}$ – фіксацію набору критеріїв та задання їх способу інтерпретації.

- 3) F_3^{SCL} – задання діапазону, в якому критерії можуть приймати значення (максимальна та мінімальна оцінка).
- 4) F_4^{SCL} – визначення напрямку оптимізації критеріїв.
- 5) F_5^{SCL} – задання важливостей критеріїв за допомогою передбачених способів.
- 6) F_6^{SCL} – визначення системи переваг.
- 7) F_7^{SCL} – фільтрація та пошук об'єктів.
- 8) F_8^{SCL} – підготовка та шаблонізація даних для їх представлення у веб-інтерфейсі.

3.2. Функціонально-компонентна модель програмної системи розв'язку задач ранжування альтернатив

Функціонально-компонентна модель використовується для представлення взаємодій, відношень і залежностей програмних модулів, а також для детального опису компонентів системи. Узагальнено дану модель для програмного комплексу розв'язку задач ранжування альтернатив можна представити структурою (3.12).

$$S_R = \langle M_D, M_S, M_p \rangle \quad (3.12)$$

Елементи, що входять в дану модель:

- 1) M_D – модель, що задає поведінку системи;
- 2) M_S – модель, що задає структуру системи;
- 3) M_p – модель, що задає структуру програмних сутностей.

3.2.1. Модель поведінки системи

Модель поведінки системи має структуру (3.13).

$$M_D = \langle d_{use}, d_{act}, d_{seq} \rangle \quad (3.13)$$

Вона включає в себе:

1) d_{use} – UML-діаграма варіантів використання системи розв'язку задач ранжування альтернатив.

2) d_{act} – UML-діаграма активності системи розв'язку задач ранжування альтернатив.

3) d_{seq} – UML-діаграма взаємодії системи розв'язку задач ранжування альтернатив.

UML-діаграма варіантів використання [96, 128, 140] програмної системи розв'язку задач ранжування альтернатив представлена на рис. 3.1. Дана діаграма показує основні дії, що можуть здійснюватися в рамках системи, і користувачів, які можуть їх здійснювати.

Основним користувачем системи є ОПР, якому надаються засоби формування та розв'язання задачі ранжування альтернатив.

Формування задачі ранжування в себе включає:

- визначення множини альтернатив;
- визначення критеріїв:
 - визначення множини критеріїв,
 - встановлення важливості критеріїв,
 - визначення інтерпретаційних функцій для критеріїв;
- встановлення критеріальних значень для альтернатив;
- формування системи переваг.

Процес визначення множини альтернатив включає в себе аналіз таксономічної структури онтологічного графу та знаходження однорідних об'єктів, які характеризуються спільними властивостями. ОПР може керувати цим процесом, зокрема за допомогою фільтрації.

Визначення множини критеріїв на основі онтології полягає у встановленні множини властивостей об'єктів, що можуть бути застосовані, як критерії. Також ОПР може обрати та визначити механізм інтерпретації властивостей, як критерії.

За допомогою передбачених способів ОПР може встановити важливість критеріїв та обрати напрям їх оптимізації. Також користувач може сформуванати систему переваг. Цей процес в себе включає вибір способів, за допомогою яких буде встановлюватись важливість критеріїв.

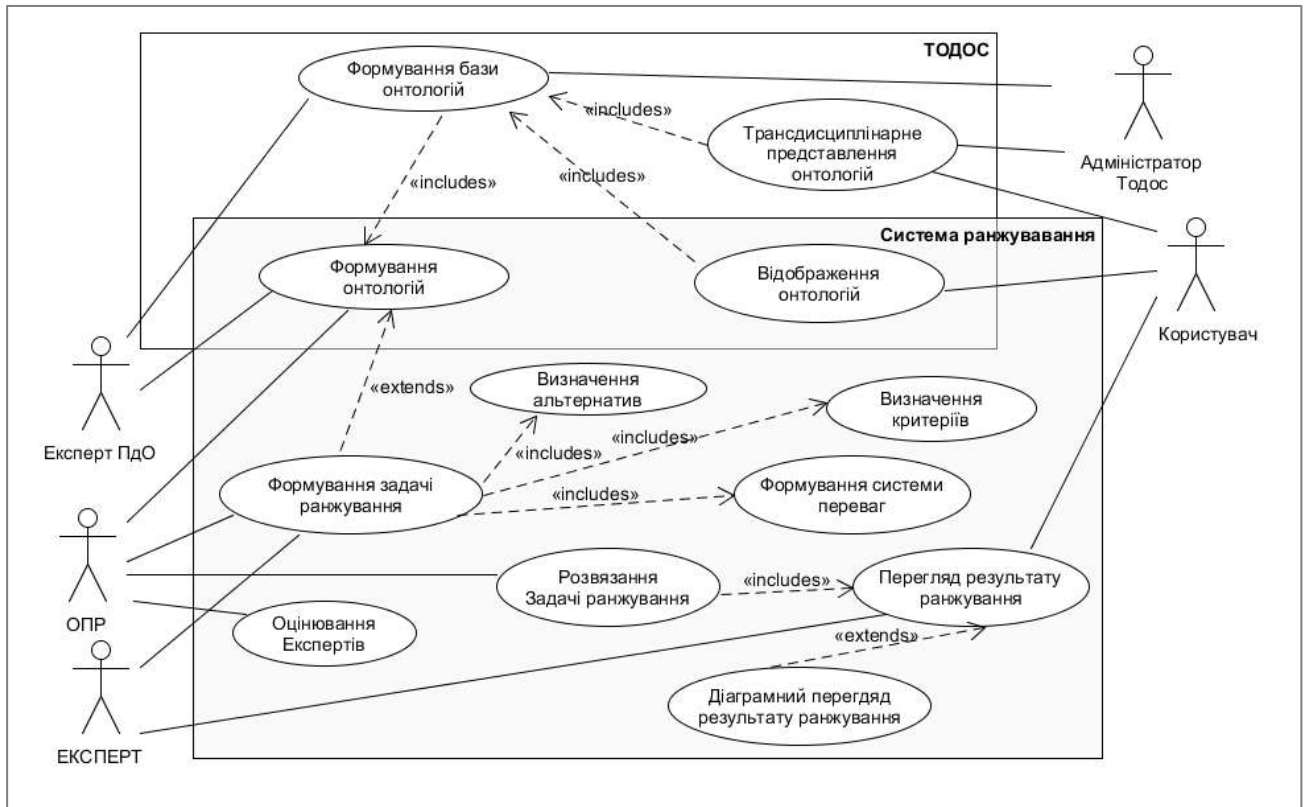


Рис. 3.1. Способи використання системи

Розв'язання задачі ранжування альтернатив в себе включає вибір способу розв'язку задачі та безпосередньо перегляд результату. Результат ранжування може бути провізуалізований за допомогою діаграм, також можуть бути задіяні механізми атрибувної фільтрації.

Експерт ПдО виконує функцію формування тематичних онтологій, що в подальшому можуть бути опубліковані адміністратором в публічній бібліотеці. Формування може бути виконане вручну (за допомогою редактора онтологій) або здійснюватися автоматично – на основі процесу структуризації тексту, що також здійснюється експертом ПдО.

Експерт допомагає ОПР при формуванні задачі ранжування та володіє відповідними всіма засобами ОПР, але участь та роль експерта у вирішенні задачі ранжування визначається ОПР-ом.

Звичайний користувач може проглядати результат ранжування, якщо йому наданий відповідний доступ.

UML-діаграма активності [96, 128, 140] програмної системи розв'язку задач ранжування альтернатив показана на рис. 3.2. Діаграма описує алгоритм роботи з системою розв'язку задач ранжування альтернатив.

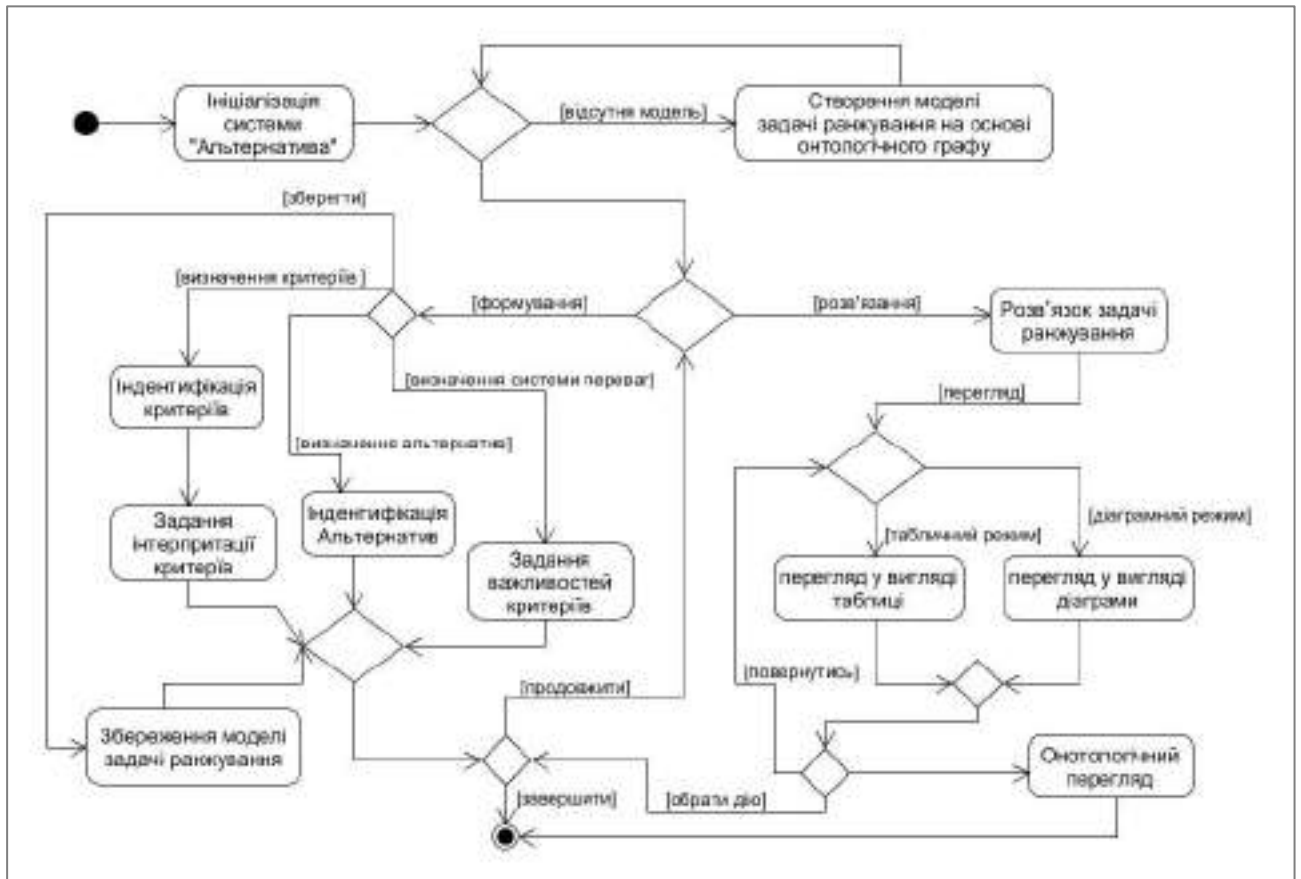


Рис. 3.2. Діаграма активності системи

Після ввімкнення системи відбувається її ініціалізація та формування задачі ранжування альтернатив на основі відповідного онтологічного графу.

Якщо для даного графу у користувача не збережено відповідну модель задачі ранжування, то на основі таксономічного аналізу визначається множина альтернатив. Перелік можливих критеріїв визначається на основі властивостей об'єктів, які були ідентифіковані, як альтернативи.

Після закінчення етапу початкового формування задачі чи її завантаження, користувач системи може за допомогою фільтрації встановити множину альтернатив та за допомогою спеціального інтерфейсу визначити

критерії. Після цього визначається система переваг та задаються важливості критеріїв. Далі або можна зберегти задачу, або її розв'язати. Після її розв'язання відображається рейтингова таблиця об'єктів, для яких відбулось ранжування за сукупністю показників.

UML-діаграма взаємодії програмної системи ранжування альтернатив показана на рис. 3.3. Дана діаграма представляє собою основний процес онтологічного супроводу задачі ранжування.

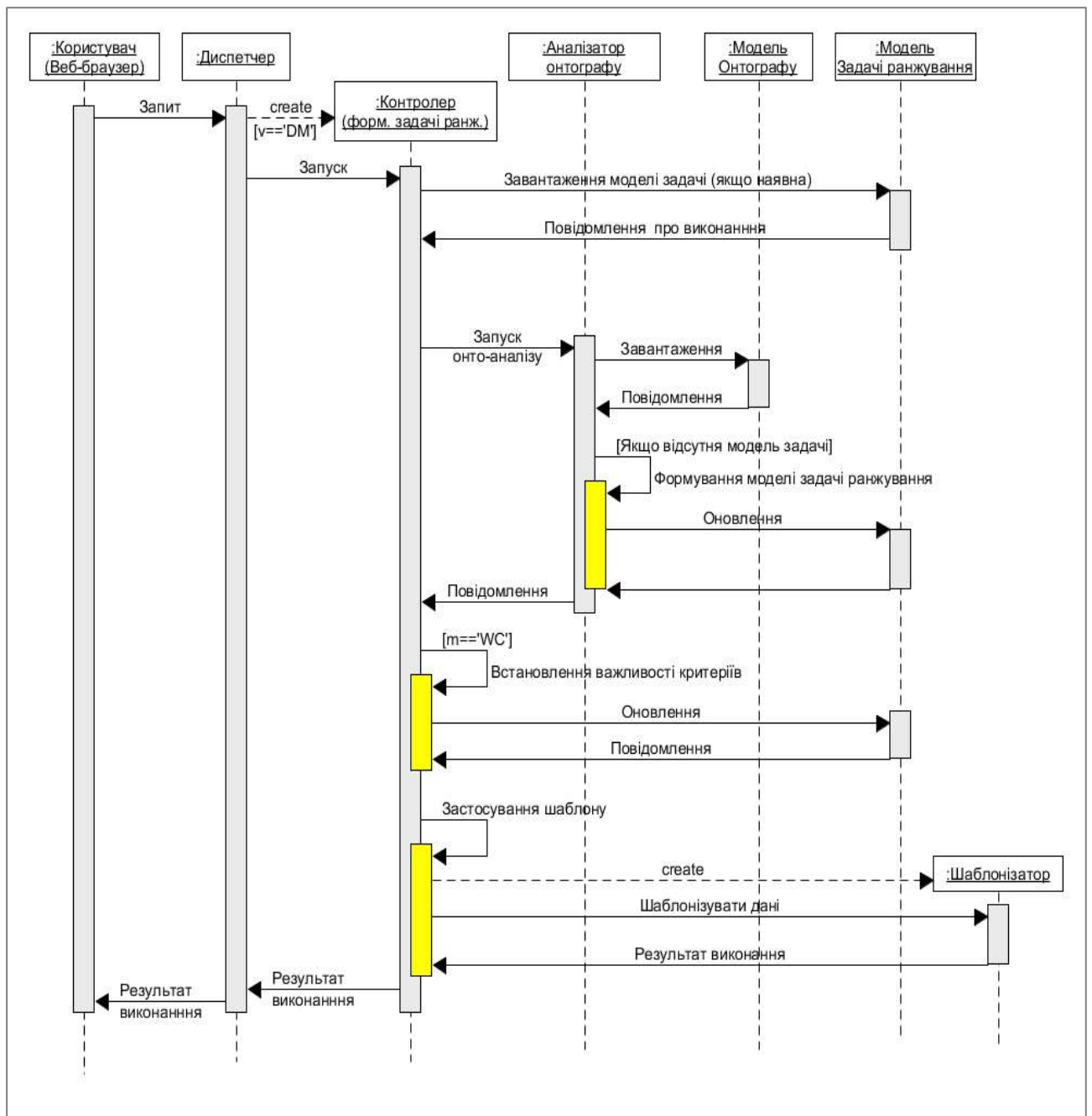


Рис. 3.3. Діаграма взаємодії програмної системи розв'язку задач ранжування

Запит користувача потрапляє на обробку до «Диспетчера», який в залежності від вхідного запиту створює відповідний контролер та передає йому управління на подальше опрацювання. В даному випадку створюється контролер, який відповідає за формування задачі. Далі контролер створює модель задачі ранжування на основі аналізу онтологічного графу, який здійснює «Аналізатор онтографу». Цей процес полягає в тому, щоб із таксономічної структури онтології виокремити множину однорідних об'єктів, які ідентифікуються, як альтернативи, та визначити із цих об'єктів множину властивостей, які можуть бути використані в якості критеріїв. Після завантаження моделі задачі ранжування альтернатив в залежності від вхідного запиту далі може здійснюватись модифікація моделі та її оновлення. Наприклад, встановлення важливостей для критеріїв. Після здійснення обробки вхідного запиту контролер підготовлює та шаблонізує дані і передає утворений результат на сторону клієнта.

3.2.2. Модель структури системи

Модель структури системи має вигляд (3.14).

$$M_S = \langle d_{class}, req \rangle \quad (3.14)$$

Вона включає в себе:

- 1) d_{class} – діаграма класів системи розв'язку задачі ранжування альтернатив за сукупністю показників;
- 2) req – технічне завдання на проектування системи розв'язку задач ранжування альтернатив.

Діаграма класів [8, 9, 45] системи ранжування показана на рис. 3.4.

Класи даної системи умовно можна поділити на контролери, класи даних, класи, що здійснюють онтологічний аналіз та класи, які складають алгоритмічне забезпечення системи розв'язку задачі ранжування.

До, класів які представляють модель даних відносяться:

- 1) `Ontograph` – клас, що зберігає онтологічний граф. Являє собою своєрідний контейнер з набору вершин графу та надає можливість швидкого доступу до кожного вузла за його іменем (імена вершин мають бути унікальними) або унікальним ідентифікатором. Дозволяє отримати доступ до батьківських та дочірніх елементів обраної вершини. Містить деякі базові алгоритми обробки графів. Володіє методами завантаження та збереження графової структури у форматах `xml/json`.
- 2) `Node` – клас, що зберігає інформацію про вершину графу – її ім'я, властивості, дочірні та батьківські вузли. Надає доступ до цих даних.
- 3) `Property` – клас, що зберігає інформацію про властивості певного об'єкта. Надає засоби для їх опрацювання – наповнення, видалення та редагування.
- 4) `DecisionTask` – клас, що призначений для збереження моделі задачі ранжування альтернатив. Містить в собі інформацію про альтернативи та критерії. Надає до них доступ та дозволяє їх долучати, видаляти та редагувати.
- 5) `NodeAlternative` – клас, що призначений для збереження альтернатив. Наслідується від класу `Node`, що дозволяє йому бути сумісним із контейнером `Ontograph`. Містить деякі додаткові атрибути. Надає доступ до своїх властивостей.
- 6) `Criterion` – клас, який містить інформацію про критерій: його назву, діапазон можливих значень, напрям оптимізації та деякі інші характеристики. Володіє відповідними методами для редагування своїх атрибутів.

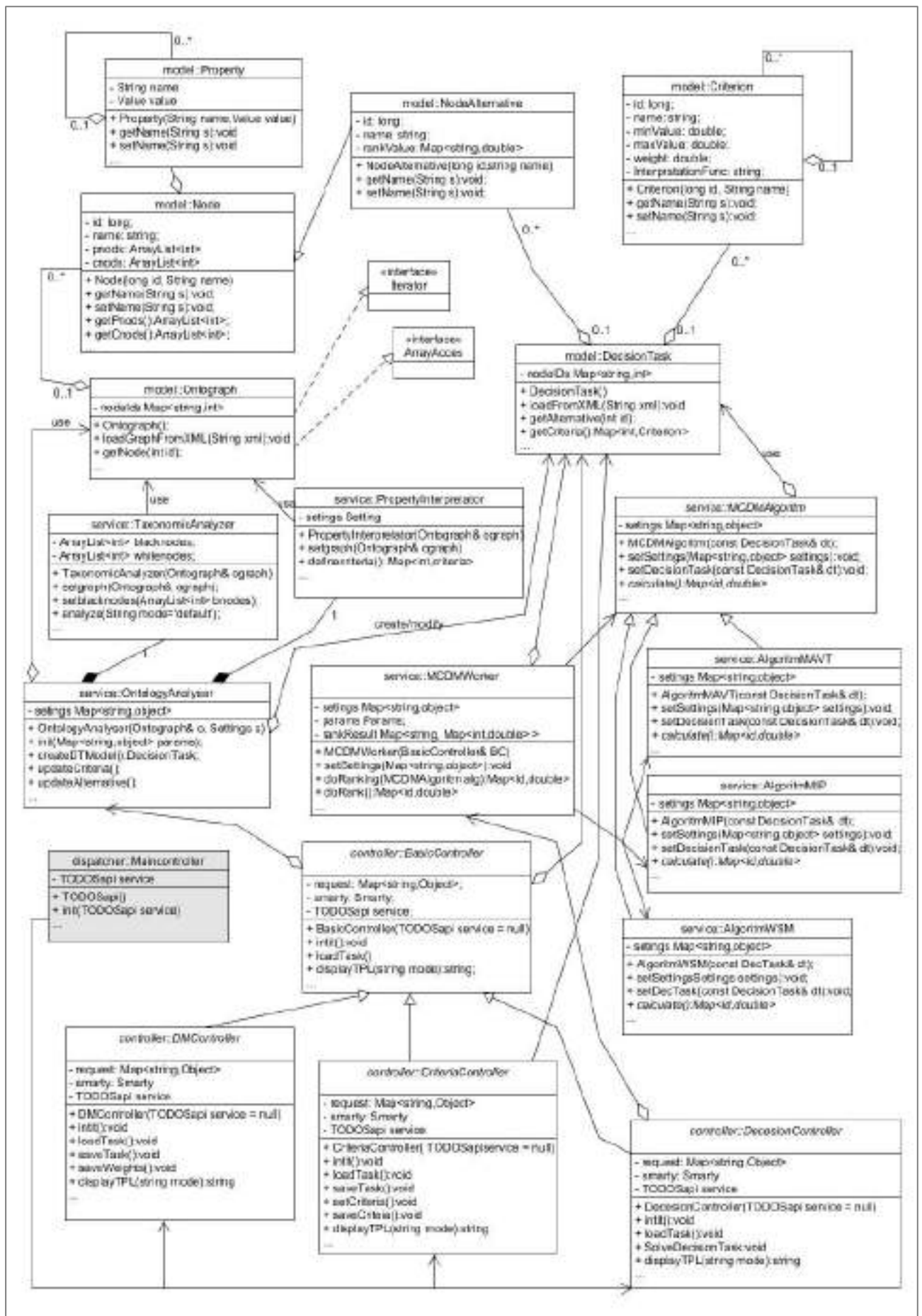


Рис. 3.4. Діаграма класів системи ранжування альтернатив

До класів, які призначені для онтологічного аналізу відносяться:

- 1) `OntologyAnalyser` – головний клас, який власне і здійснює онтологічний супровід та виконує ключову роль при формуванні задачі ранжування альтернатив на основі онтологічної моделі. Він здійснює загальне управління, керує обробкою та аналізом онтологічних графів. Створює та оновлює модель задачі ранжування. Агрегує в собі допоміжні класи `TaxonomicAnalyzer` та `PropertyInterpretator`, які націлені на більш конкретні завдання.
- 2) `TaxonomicAnalyzer` – клас, що призначений для аналізу таксономічної структури графу та виявлення об'єктів онтології, що можуть бути застосовані, як альтернативи. Володіє деякими алгоритмами аналізу графової структури, які дозволяють здійснювати маркування вершин та застосовувати його для фільтрації вершин та альтернатив.
- 3) `PropertyInterpretator` – клас, який призначений для виявлення спільних властивостей однорідних об'єктів-альтернатив, що можуть бути застосовані, як критерії. Відповідно володіє необхідними методами та доступом до онтології для проведення такого аналізу.

До класів, що призначені для знаходження розв'язку задачі ранжування альтернатив відносяться:

- 1) `MCDMWorker` – головний клас, який управляє процесом розв'язання сформованої задачі ранжування альтернатив на основі її моделі та за допомогою класів, які вже реалізують конкретні методи вирішення задачі ранжування та становлять математично ядро системи. Цей клас виконує агрегування отриманих ними розв'язків в остаточний кінцевий результат.
- 2) `MCDMAlgorithm` – абстрактний клас, від якого наслідуються дочірні класи, які безпосередньо вже реалізують свою логіку розв'язання задачі ранжування альтернатив. Цей клас використовує сформовану модель задачі ранжування та володіє загальними методами підготовки даних, які потрібні для його класів-нащадків. Містить методи, що

дозволяють здійснити обрахунок критеріальних значень на основі інтерпретаційних функцій, які ОПР може визначити для критеріїв. Містить методи для зведення критеріальних значень в єдину спільну числову шкалу та методи, що встановлюють вагові коефіцієнти критеріїв на основі обраної системи переваг та заданих значень їх важливостей.

- 3) `AlgorithmWSM` – клас, що реалізує алгоритм розв’язання задачі ранжування альтернатив за допомогою методу зважених сум критеріїв (WSM).
- 4) `AlgorithmMIP` – клас, що реалізує алгоритм розв’язання задачі ранжування альтернатив за допомогою методу ідеальної точки.
- 5) `AlgorithmMAVT` – клас, що містить алгоритмічне забезпечення для розв’язання задачі ранжування альтернатив за допомогою методу багатоатрибутного оцінювання (MAVT).

До рівня контролерів відносяться наступні класи:

- 1) `BasicController` – клас від якого наслідуються всі інші контролери та містить атрибути та методи, що застосовуються дочірніми класами. Визначає базовий механізм підготовки даних із моделі для їх шаблонізації на основі якої і створюється представлення даних у вигляді веб-інтерфейсу.
- 2) `DMController` – клас-контролер, що відповідає за формування задачі ранжування альтернатив за допомогою відповідного веб-інтерфейсу. Використовує класи, що виконують аналіз онтологічного графу. Він оброблює запити, що надходять від користувача та робить необхідні зміни в модель задачі ранжування. Цей клас застосовується для визначення напряму оптимізації і важливостей критеріїв та для встановлення системи переваг, на основі якої буде розв’язуватись задача ранжування. Відповідає за обробку запитів, пов’язаних з формуванням множини альтернатив за допомогою додаткових

обмежень, які може встановлювати ОПР, що накладаються на батьківські вузли таксономічної структури.

- 3) `CriteriaController` – клас-контролер, що відповідає за веб-інтерфейс, який призначений для визначення критеріїв на основі знайдених властивостей об'єктів онтології. Він здійснює обробку запитів, які надходять з цього інтерфейсу та вносить відповідні оновлення в модель задачі ранжування. Зокрема, цей контролер дозволяє користувачеві власноруч встановити діапазонні значення критеріїв, та при необхідності задати функції обчислення критеріальних значень для альтернатив.
- 4) `DecisionController` – клас-контролер, що відповідає за веб-інтерфейс, який призначений для візуалізації рішення задачі ранжування. Використовує клас `MCDMWorker`, якщо виникає необхідність розв'язати задачу ранжування. Здійснює обробку запитів, пов'язаних з пошуком або фільтрацією даних, які викликаються у даному веб-інтерфейсі.

Центральне управління здійснює клас `dispatcher::Maincontroller`, який в залежності від вхідного запиту створює та активує потрібний контролер, який безпосередньо і здійснює його обробку. Цей клас може створювати контролер ТОДОС-а, який викликає його метод ініціалізації, де вхідним параметром передаються сервіси ТОДОС-а, які безпосередньо можуть бути застосовані в програмних модулях. Таким чином і відбувається інтеграція системи ранжування альтернати у ІТ-платформу ТОДОС.

3.2.3. Модель структури програмних сутностей

Модель структури програмних сутностей розглядається як файли вихідного коду, бібліотеки, виконувані файли і відношення між ними. Дана модель описується за допомогою UML Package Structure і показана на рис. 3.5.

Система має два блоки сутностей:

- веб-інтерфейс, що безпосередньо відображується користувачу і надає йому керуючі елементи для запуску виконання тих чи інших дій;
- серверна інфраструктура, що приймає запити від веб-інтерфейсу і виконує задані запитом дії.

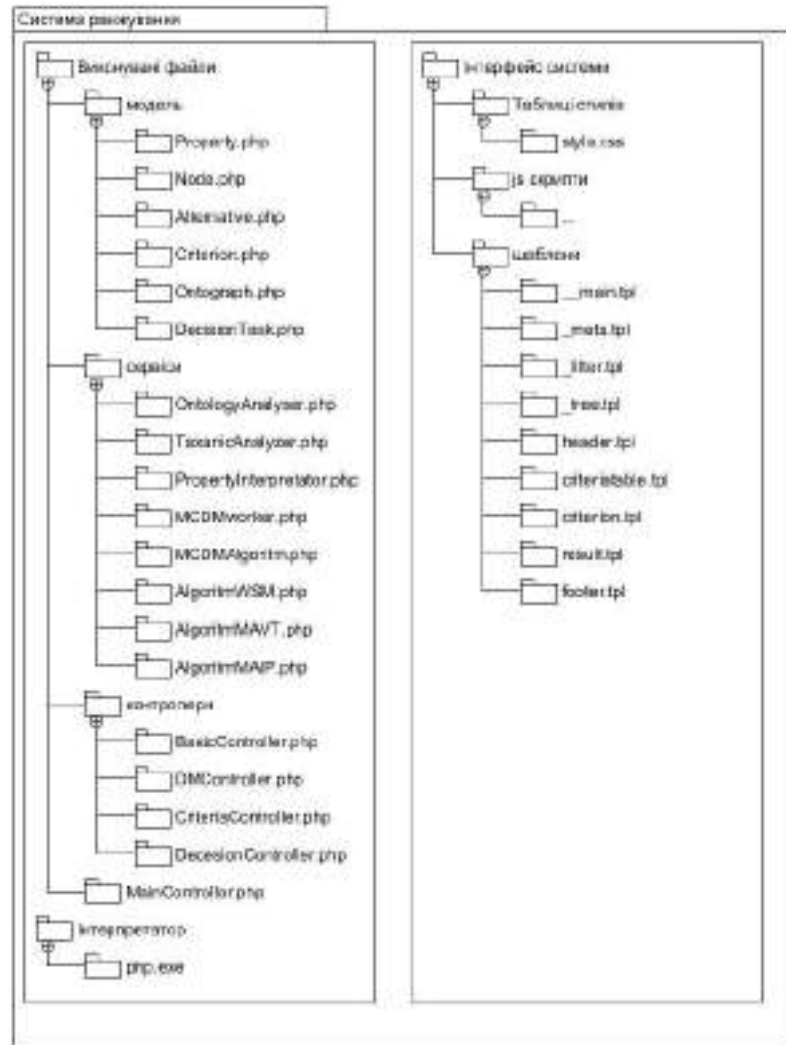


Рис. 3.5. Структура програмних сутностей системи розв'язку задачі ранжування альтернатив

Серверна інфраструктура системи ранжування альтернатив представлена програмними модулями, які відповідають її структурі класів. Отже кожний модуль – це певний клас, який призначений для вирішення своїх завдань.

Веб-інтерфейс системи реалізується за допомогою веб-шаблонів та складається з наступних основних форм представлення:

- «формування задачі ранжування»;

- «визначення та редагування критеріїв»;
- «перегляд результату ранжування».

Для кожної форми представлення існує свій шаблон, який під'єднується при необхідності у головний шаблон системи, який містить загальні і спільні елементи. Компоненти форм представлення описуються своїми шаблонами.

3.3. Трирівнева клієнт-серверна архітектура програмної системи розв'язку задач ранжування альтернатив

Для ефективної розробки Web-орієнтованих застосувань та їх подальшої експлуатації в програмній інженерії використовується трирівнева архітектура, на основі якої було спроектовано програмну систему розв'язку задач ранжування альтернатив [104, 111, 113].

Багаторівнева архітектура дозволяє інтегрувати нові технології та долучати інші компоненти без необхідності перепроєктовувати все програмне забезпечення та переписувати цілу програму, весь її програмний код та існуючі компоненти, що в свою чергу полегшує вирішення проблем масштабування та підтримки. З точки зору безпеки, ця архітектура дозволяє зберігати приватну та конфіденційну інформацію на рівні логіки та відділяти її від рівня представлення, що робить її більш захищеною. Таким чином до головних переваг багаторівневої архітектури відноситься:

- 1) безпека – можливість захистити кожний рівень незалежно використовуючи різні методи;
- 2) легкість в управлінні – можливість керувати та модифікувати кожний рівень окремо не впливаючи на інші рівні;
- 3) масштабованість – долучення ресурсів може бути здійснено безпосередньо на своєму рівні, незалежно від інших рівнів;
- 4) гнучкість – крім ізольованої масштабованості, наявна можливість розширити кожен рівень у будь-який спосіб, що відповідає цілям розробки.

Крім цього, слід відмітити і іншу користь, яку надає ця архітектура:

- 1) більш ефективна розробка – багаторівнева архітектура дуже сприятлива для розробки, оскільки різні команди можуть працювати на кожному рівні незалежно;
- 2) легко додати нові функції – кожна нова функція може бути додана до її відповідного рівня без впливу на інші рівні;
- 3) легке повторне використання – оскільки програма ділиться на незалежні рівні, то наявна можливість повторно використовувати кожен рівень та їх компоненти для інших програмних проектів.

Трирівнева архітектура складається з рівня представлення, рівня логіки та рівня даних і передбачає наявність клієнтської програми, сервера додатків і сховища даних.

- 1) Рівень представлення – рівень з яким безпосередньо взаємодіє користувач системи. Він включає компоненти користувацького інтерфейсу та механізм отримання вхідних даних.
- 2) Рівень логіки – містить набір компонентів, які відповідають за обробку отриманих від рівня представлення даних, реалізує всю необхідну логіку застосунку, всі обчислення, взаємодіє з рівнем даних і передає рівню представлення результат своєї обробки.
- 3) Рівень даних – визначає місце і спосіб збереження та доступу до даних, які використовуються у програмній системі.

Архітектура програмної системи розв'язку задач ранжування альтернатив базується на описаних в попередніх розділах моделях і являє собою набір програмних засобів. Деякі компоненти програмної системи використовують API системи ТОДОС, який надає сервіси для отримання даних. Однак всі компоненти реалізовані у вигляді окремих модулів, що можуть виконуватись як в складі ТОДОС-а, так і незалежно.

Основна архітектура програмної системи розв'язку задач ранжування альтернатив, представлена на рис. 3.6.

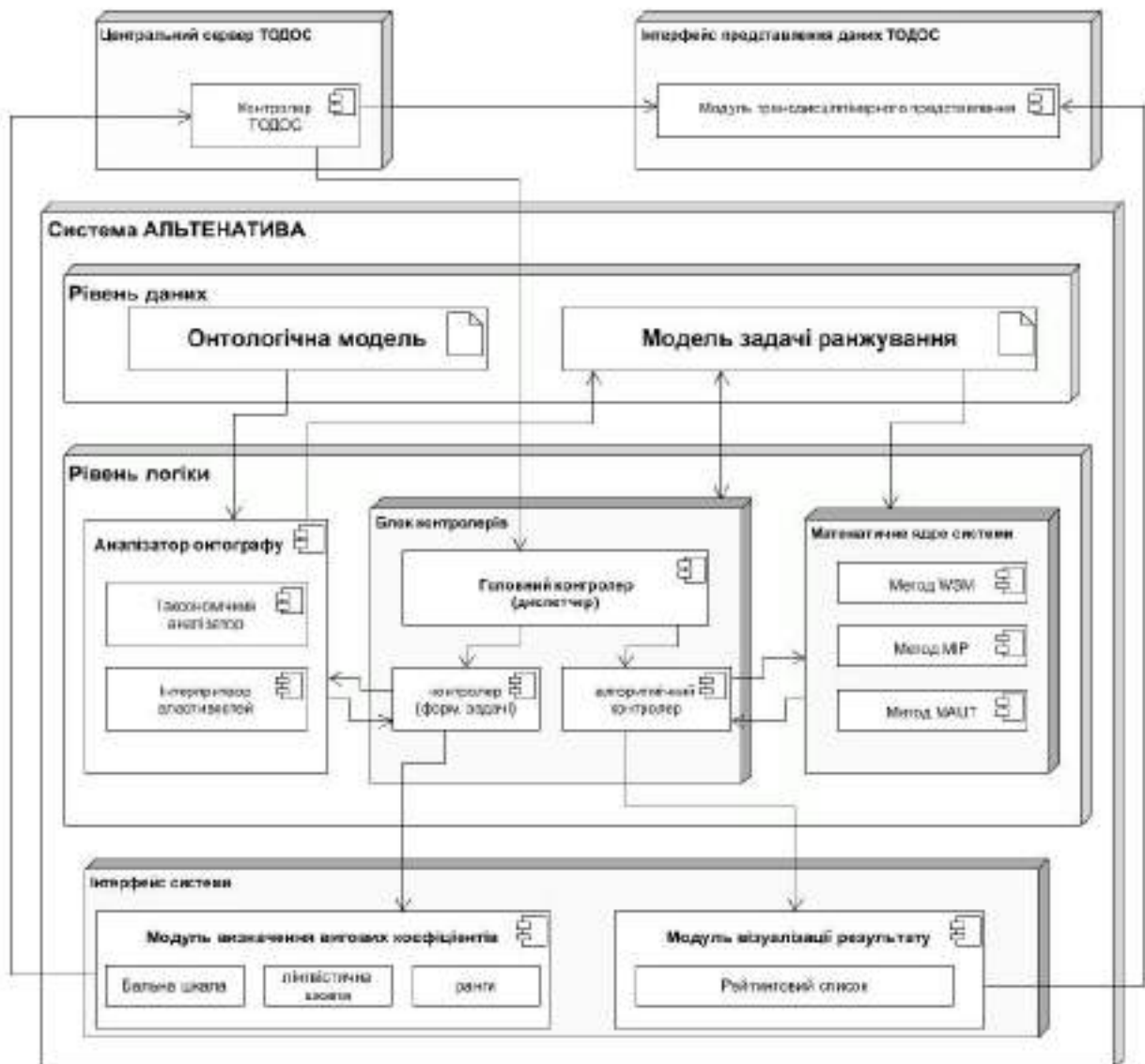


Рис. 3.6. Архітектура системи розв'язку задач ранжування альтернатив

До основних програмних компонентів системи розв'язання задачі ранжування альтернатив (що забезпечують логіку її роботи) відноситься:

- 1) блок контролерів – містить в собі всі контролери системи, яким на обробку надходять запити від веб-інтерфейсу системи;
- 2) аналізатор онтографу – здійснює онтологічний супровід системи на основі онтологічного графу, який є основним елементом при формуванні задачі ранжування. Проводить розбір та аналіз онтографу, виокремлює множину альтернатив та властивості, що можуть бути використані, як критерії для ранжування;

- 3) математичне ядро системи – містить в собі MCDA алгоритмічне забезпечення, яке виконує ранжування альтернатив на основі сформованої моделі задачі.

Інтерфейс системи складається з наступних основних модулів:

- 1) формування задачі – дозволяє користувачеві специфікувати альтернативи та критерії, що були встановлені внаслідок розбору онтологічного графу. Також він дозволяє визначити важливості критеріїв за допомогою бальної або лінгвістичної шкали чи методу ранжування та обрати систему переваг, на основі якої буде проводитись безпосередньо розв'язок задачі;
- 2) візуалізація результату – відповідає за відображення результатів ранжування альтернатив у табличному режимі. Також він підтримує представлення у вигляді стовпчикової діаграми.

Висновки за розділом 3

Розроблено інформаційну модель програмної системи розв'язку задач ранжування альтернатив, що може бути інтегрована в ТОДОС.

Розроблено функціонально-компонентну модель програмної системи розв'язку задач ранжування альтернатив, що включає в себе модель поведінки системи, модель структури системи та модель структури програмних сутностей.

На основі інформаційної та функціонально-компонентної моделей програмної системи розв'язку задач ранжування альтернатив розроблено архітектуру даної системи, що визначає структуру, склад і особливості функціонування розробленого програмного засобу.

Результати досліджень, наведені в третьому розділі, опубліковано в роботах [21, 26, 28, 30, 31].

РОЗДІЛ 4. ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ФОРМУВАННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СЕРЕДОВИЩ НА ОСНОВІ РОЗВ'ЯЗУВАННЯ ЗАДАЧ РАНЖУВАННЯ АЛЬТЕРНАТИВ

4.1. Інструментальний засіб «АЛЬТЕРНАТИВА»

До складу ІТ-платформи ТОДОС входить система підтримки прийняття рішень – «Альтернатива» [83], яка призначена для розв'язку прикладних задач ранжування та рейтингування для прийняття відповідальних та науково обґрунтованих рішень. До таких задач в першу чергу можна віднести вибір найкращого варіанту рішення (з переліку наперед відомих можливих варіантів) та подолання тієї чи іншої конкретної проблеми. Система «Альтернатива» – забезпечує упорядкування об'єктів-концептів онтології на основі інтегрованої обробки властивостей, що їх характеризують. У загальному випадку властивості-критерії характеризуються різними ступенями важливості, що ускладнює прийняття ефективних рішень без застосування відповідних математичних методів.

В математичне забезпечення розробленої системи входять основні методи багатокритеріального прийняття рішення (MCDA методи), а також алгоритм для розв'язку оберненої задачі ранжування. Система дозволяє відображувати отриманий рейтинговий список об'єктів як у табличному вигляді, так і у вигляді діаграм.

Систему доцільно використовувати під час проведення нарад, засідань, конкурсів і т.п., коли необхідно оперативно отримувати обґрунтований результат, враховуючи думки присутніх фахівців, експертів та маючи можливість візуально спостерігати за процесом прийняття рішення.

Дана система для розв'язання конкретної прикладної задачі дозволяє крім знань ОПР та задіяних експертів ефективно також застосовувати закладені знання в онтології експертами предметних областей, що в свою чергу суттєво підвищує якість розгляду проблемної задачі та веде до збільшення ефективності прийняття рішень.

4.1.1. Представлення предметної області у вигляді онтологічної моделі для задачі ранжування у середовищі ТОДОС

Процес розв'язання задачі ранжування за допомогою системи «Альтернатива» має два етапи [24, 25, 80]. На першому етапі будується та наповнюється експертом ПдО або ОПР онтологічний граф. Цей процес може здійснюватись у середовищі ТОДОС. Приклад такого графу наведений на рис. 4.1.

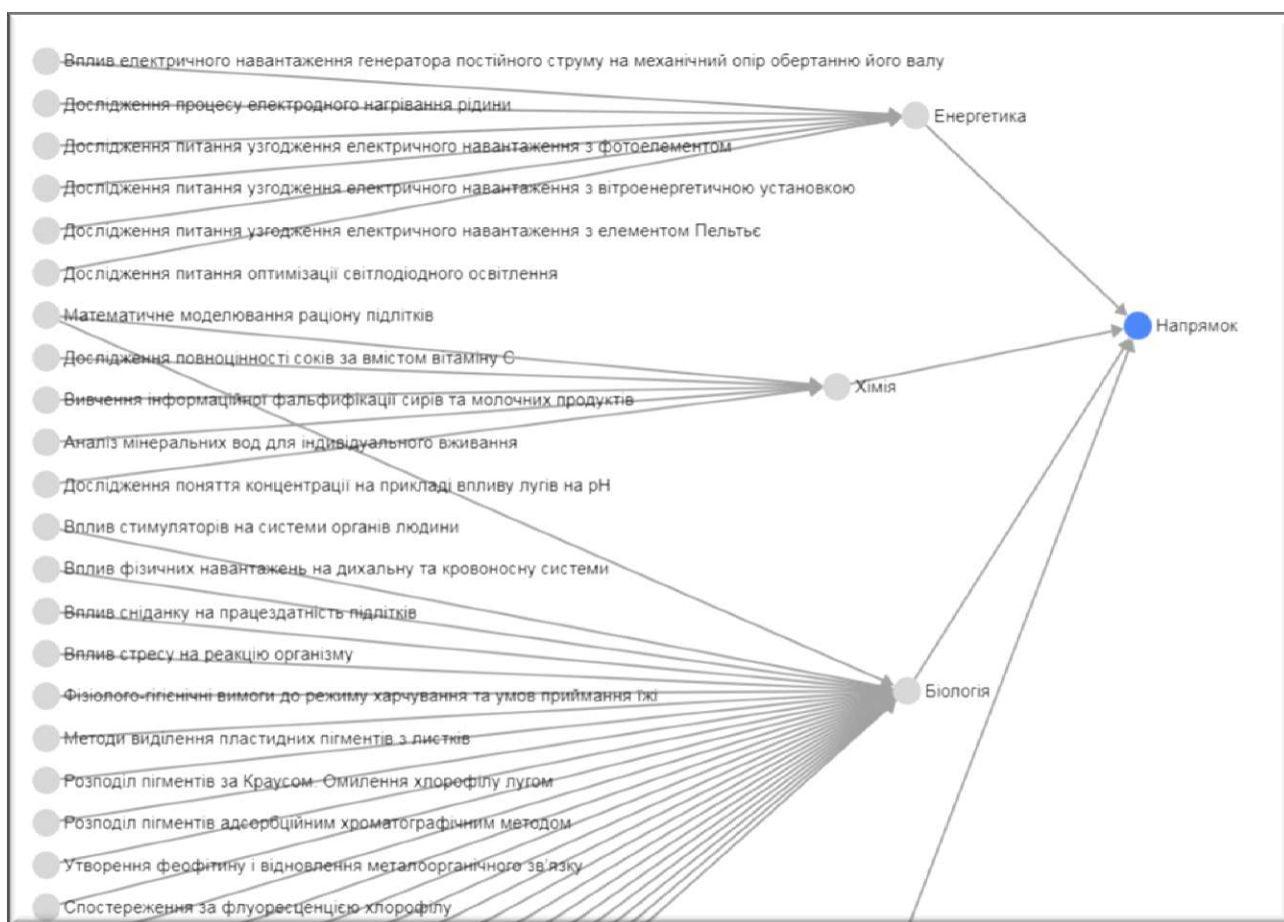


Рис. 4.1. Таксономія навчальних проєктів (Stem-освіта)

У системі «Альтернатива» таксономічна структура є основою для виокремлення альтернатив із множини об'єктів-концептів онтології. В даному випадку при використанні цієї системи для онтологічного графу, зображеного на рис. 4.1, будуть виокремлені елементи – листові вершини направленої графу.

Об'єктне представлення онтологічного графу наведено на рис. 4.2. Власне об'єкти нижнього рівня ієрархії і виступають в якості альтернатив.



Рис. 4.2. Об'єктне представлення онтології

Також відмітимо, що об'єкти онтології можуть характеризуватись багаторівневою ієрархічною структурою властивостей. Приклад такого графу наведений на рис. 4.3, де представлена внутрішня таксономічна структура властивостей онтології аудиту підручників (рис. 4.4). В даному випадку вершинами графу є певні сутності, притаманні навчальним матеріалам, а дугами – існуючі зв'язки між ними.

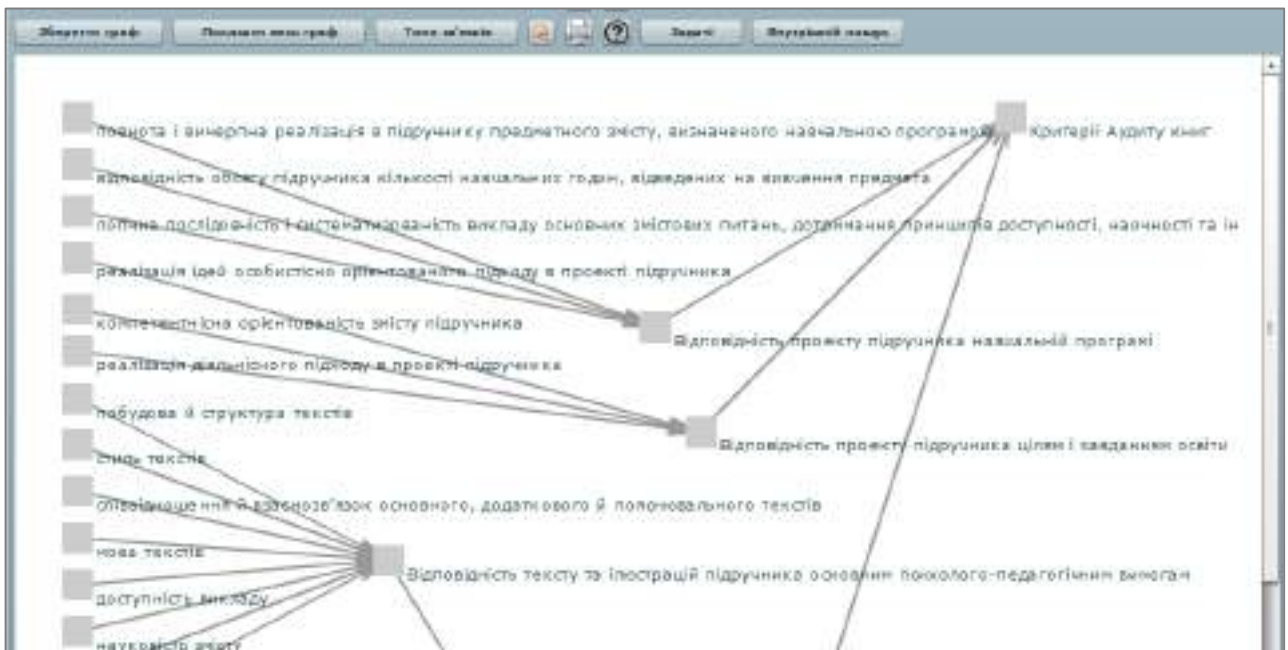


Рис. 4.3. Приклад онтологічного графу для критеріїв задачі аудиту підручників

Що стосується самих безпосередніх значень критеріїв, то крім наявних властивостей об'єктів, які були сформовані експертом ПдО при побудові графу, ОПР (та задіяні експерти) на цьому етапі мають можливість доповнювати та редагувати значення властивостей об'єктів у своєму особистому кабінеті за допомогою відповідних інструментаріїв (рис. 4.4) та вносити їх значення у зручній для себе числовій шкалі.

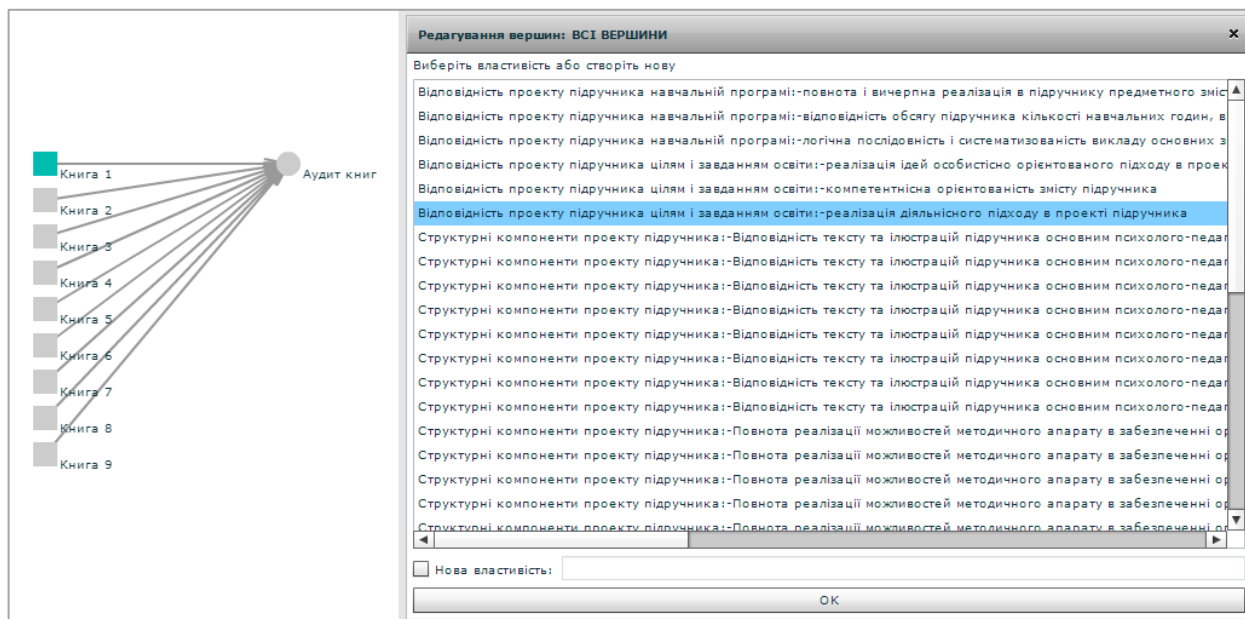


Рис. 4.4. Форма множинного редагування вершин

Дана форма дозволяє редагувати вибрану властивість для всіх вершин з деякої підмножини зручним чином у вигляді таблиці. На формі присутній список, з якого потрібно вибрати властивість. Можна також ввести нову властивість, яка буде додана до всіх вершин. Після того, як обрано певний показник, користувач може проглянути існуючі значення та їх наповнити або оновити в залежності від своїх потреб (рис. 4.5). Відмітимо, що система формування онтологічних моделей ІТ-платформи ТОДОС-а дозволяє задавати значення атрибутів (властивостей) об'єктів і в інший спосіб. Наприклад, за допомогою безпосереднього редагування вершини онтології (картки об'єкта) або за допомогою процедури імпорту даних із форматів, що підтримує система, зокрема – це Ексель файли. Крім встановлення атрибутивних значень, ОПР може створювати та видаляти вершини онтологічного графу та встановлювати їх зв'язки.

Редагування вершин: ВСІ ВЕРШИНИ; властивість: Відповідність проекту підручника цілям і завданням освіти: -ре... x	
Вершина	Значення властивості
Книга 1	67
Книга 2	54
Книга 3	60
Книга 4	43
Книга 5	71
Книга 6	73
Книга 7	68
Книга 8	65
Книга 9	73
Аудит книг	
ОК	

Рис. 4.5. Таблиця значень критерію

Також ОПР має можливість створювати та долучати до онтологічної моделі «штучні» альтернативи, які можуть виступати свого роду певними індикаторами. Значення властивостей (критеріїв) для таких альтернатив ОПР може задавати власноруч для проведення аналітичних досліджень. Такі альтернативи зазвичай будуються на основі вже існуючих, і в них здебільшого змінюються ті показники, на які можна впливати, або ті, які досліджуються. Далі на основі розв'язку задачі (рейтинговий список альтернатив), отриманого на відповідному етапі процесу проведення вибору (ранжування) з'являється додаткова можливість виявити найбільш критичні показники, значення яких суттєво впливають на остаточний результат. Це дозволить у підсумку виробити дієвий механізм впливу на проблему, що досліджується, шляхом визначення можливих напрямів ефективного керування.

Введення до розгляду штучно створених альтернатив додатково дає можливість більш повного дослідження проблемної задачі. Тут мається на увазі підхід, пов'язаний з формуванням так званих «еталонних» альтернатив. Ці альтернативи реально на практиці чи в природі не існують, проте надаючи їм «ідеальні» значення критеріїв, ОПР може провести достатньо ґрунтовний аналіз, порівнюючи ці альтернативи з існуючими.

4.1.2. Створення інформаційного середовища задачі ранжування на основі онтологічної моделі предметної області

Для існуючого онтологічного графу можливе безпосереднє застосування системи «Альтернатива». Після її активації здійснюється аналіз об'єктів онтографу, з яких виокремлюються спільні однорідні об'єкти, їх властивості та створюється експертне середовище для ОПР (рис. 4.6).

Вибір	Ім'я	Ваг. Коеф.	Опт (max/min)	Способи задання вагових коефіцієнтів		
				Бальна шкала (10)	Лінгвістична шкала	Ранжування
<input checked="" type="checkbox"/>	Відповідність проекту підручника навчальній програмі (+)	0.333	max	5	Середня важливість	1
<input checked="" type="checkbox"/>	Відповідність проекту підручника цілям і завданням освіти (+)	0.333	max	5	Середня важливість	1
<input checked="" type="checkbox"/>	Структурні компоненти проекту підручника (+)	0.333	max	5	Середня важливість	1
3/3			Система переваг	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 4.6. Верхній рівень ієрархії задачі оцінювання якості підручника

На верхньому рівні ієрархії присутні наступні критерії:

- 1) Відповідність проекту підручника навчальній програмі;
- 2) Відповідність проекту підручника цілям і завданням освіти;
- 3) Структурні компоненти проекту підручника.

Однак кожен з представлених критеріїв містить в собі набір критеріїв нижчого рівня, що представлено на рис. 4.7 – 4.9.

ОПР у відповідності до своїх вимог задає важливості показників (критеріїв) обраної задачі ранжування альтернатив. Для цього йому надається можливість обрати один або одразу декілька найбільш поширених методів обчислення вагових коефіцієнтів критеріїв:

- бальне оцінювання;
- використання лінгвістичної шкали;
- ранжування критеріїв.

Бальна шкала – користувач проводить оцінювання важливостей критеріїв за допомогою бальної шкали (рис. 4.7). Максимальне значення бала (10, 20, 100 і т.п.) користувач задає у відповідному полі, що з'являється одразу після натискання лівої клавіші миші над числом у дужках в області заголовку стовпчика вибору даного способу.

Вибір	Ім'я	Ваг. Коеф.	Опт (max/min)	Способи задання вагових коефіцієнтів		
				Бальна шкала (10)	Лінгвістична шкала	Ранжування
<input checked="" type="checkbox"/>	реалізація ідей особистісно орієнтованого підходу в проєкті підручника	0.375	max	6	Середня важливість	1
<input checked="" type="checkbox"/>	компетентнісна орієнтованість змісту підручника	0.313	max		Середня важливість	1
<input checked="" type="checkbox"/>	реалізація діяльнісного підходу в проєкті підручника	0.313	max		Середня важливість	1
3/3			Система переваг			

Рис. 4.7. Другий рівень ієрархії для критерія «Відповідність проєкту підручника цілям і завданням освіти»

Лінгвістична шкала (рис. 4.8) – оцінювання важливості критеріїв проводиться за допомогою “словесної” шкали («Мала важливість», «Слабка важливість», «Середня важливість», «Значна важливість», «Абсолютна важливість»).

Вибір	Ім'я	Ваг. Коеф.	Опт (max/min)	Способи задання вагових коефіцієнтів		
				Бальна шкала (10)	Лінгвістична шкала	Ранжування
<input checked="" type="checkbox"/>	повнота і вичерпна реалізація в підручнику предметного змісту, визначеного навчальною програмою	0.389	max	8	Середня важливість	1
<input checked="" type="checkbox"/>	відповідність обсягу підручника кількості навчальних годин, відведених на вивчення предмета	0.289	max	2	Значна важливість	1
<input checked="" type="checkbox"/>	логічна послідовність і систематизованість викладу основних змістових питань, дотримання принципів доступності, научності та ін	0.322	max	6	Значна важливість	1
3/3			Система переваг	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 4.8. Другий рівень ієрархії для критерія «Відповідність проєкту підручника навчальній програмі»

Ранжування критеріїв (рис. 4.9) – критерії впорядковуються користувачем за спаданням своєї важливості (спочатку під номером 1 – найвагоміший критерій, в кінці – самий не суттєвий з існуючих).

Вибір	Назва	Важ. Коэф.	Важ. (max/min)	Способи задання вагових коефіцієнтів		
				Базисна шкала (1/0)	Експліцитна шкала	Результат
<input checked="" type="checkbox"/>	Відомість: часту та пострадій підручників (основні показники/подальші критерії (+))	0.2	max	5	Середня важливість	1
<input checked="" type="checkbox"/>	Кількість рецензій: можливість методичного апарату в забезпеченні обсягів навчально-педагогічної діяльності учнів (+)	0.2	max	5	Середня важливість	1
<input checked="" type="checkbox"/>	Доступність і ефективність апарату освітнього в підручнику (+)	0.2	max	5	Особливу важливість	1
<input checked="" type="checkbox"/>	Забезпечення високого розквітання та здоров'я обсягів навчальної діяльності (+)	0.2	max	5	Середня важливість	1
<input checked="" type="checkbox"/>	Відомість: педархіву підручків в освіті (+)	0.2	max	5	Середня важливість	1
5/5		Система переваг		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 4.9. Другий рівень ієрархії для критерія «Структурні компоненти проекту підручника»

У стовпчику з заголовком «*Важ. Коэф.*» система відображує математично обраховані вагові коефіцієнти у відповідності до обраних користувачем способів задання важливостей критеріїв. Включення того чи іншого способу відбувається встановленням відміток у нижній області вікна з назвою «*Система переваг*».

Коли множина критеріїв сформована і встановлені для них вагові коефіцієнти, можливе розв'язання задачі ранжування, де результатом буде певний рейтинговий список.

4.1.3. Отримання розв'язку задачі ранжування альтернатив

Остаточний результат, який фактично є розв'язком задачі з використанням MCDA методів, оформлюється системою у табличному вигляді, де зокрема, наводяться як результуючі значення по кожному підручнику-альтернативі, так і значення кожного критерія верхнього рівня ієрархії. Останні

значення генеруються за каскадним принципом з початкових значень показників нижнього рівня ієрархії (рис. 4.10).

Результати					
#	Елементи	Значення	Критерії		
			Відповідність проекту підручника навчальній програмі	Відповідність проекту підручника цілям і завданням освіти	Структурні компоненти проекту підручника
1	Книга 6	0,851	0,874	0,712	0,847
2	Книга 0	0,787	0,479	0,999	0,847
3	Книга 4	0,603	0,975	0,14	0,664
4	Книга 8	0,559	0,602	0,617	0,258
5	Книга 1	0,558	0,769	0,026	0,879
6	Книга 3	0,556	0,594	0,09	0,936
7	Книга 2	0,53	0,624	0,941	0,039
8	Книга 5	0,5	0,996	0,319	0,136
9	Книга 7	0,433	0,469	0,684	0,059

Рис. 4.10. Результати ранжування підручників

Крім представлення результату в табличному вигляді наявна можливість його перегляду у вигляді діаграми рис. 4.11.

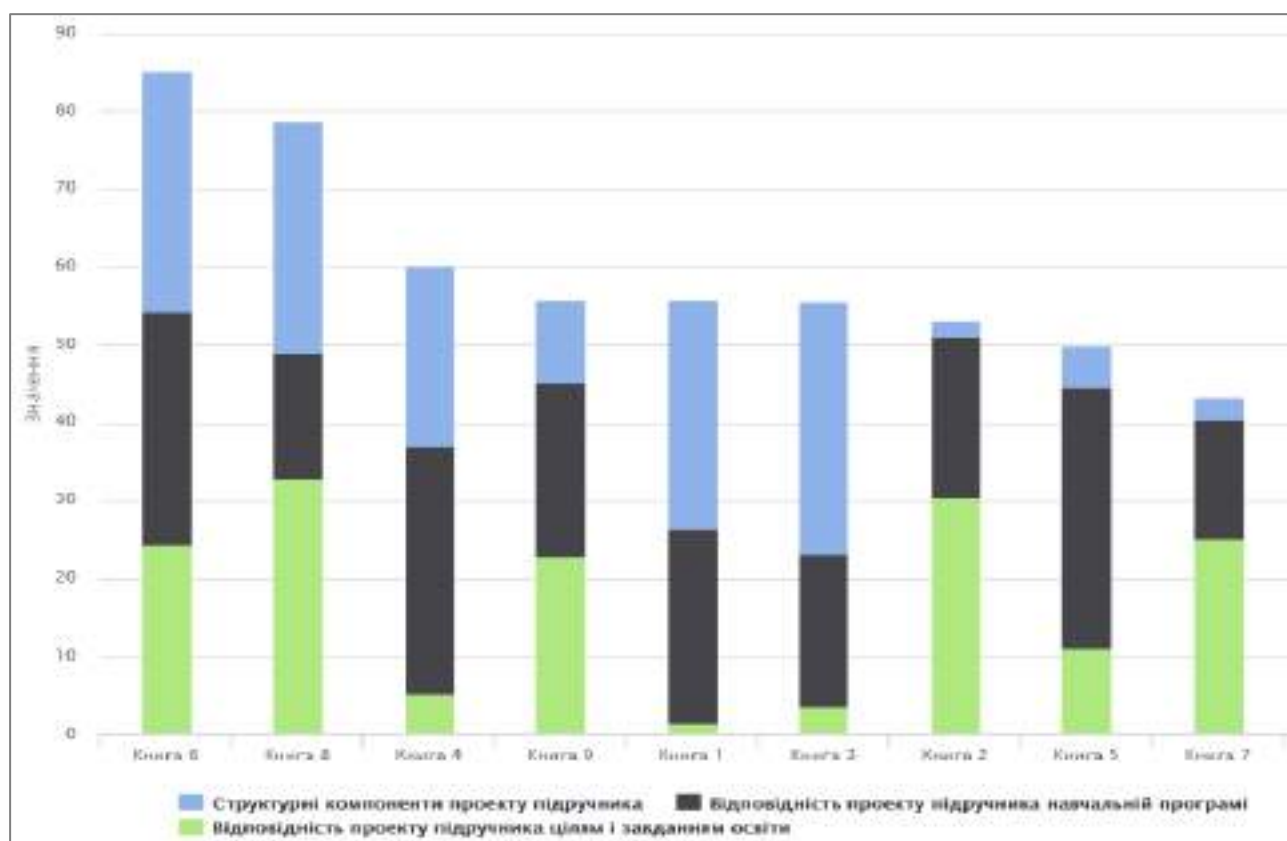


Рис. 4.11. Результати ранжування підручників

Для всебічного дослідження проблеми ОПР може виключати з розгляду ті чи інші показники, знімаючи відповідні відмітки у стовпчику з заголовком «Вибір», та корегувати значення важливостей цих показників (рис. 4.6 – 4.9). На цьому етапі може бути розв’язана обернена задача ранжування.

Відзначимо, що система «Альтернатива» має різні інтерфейси відображення експертного середовища ОПР, які можуть налаштовуватись за допомогою відповідних шаблонів. Результат ранжування для навчальних проектів «Stem-освіти» представлений на рис. 4.12.

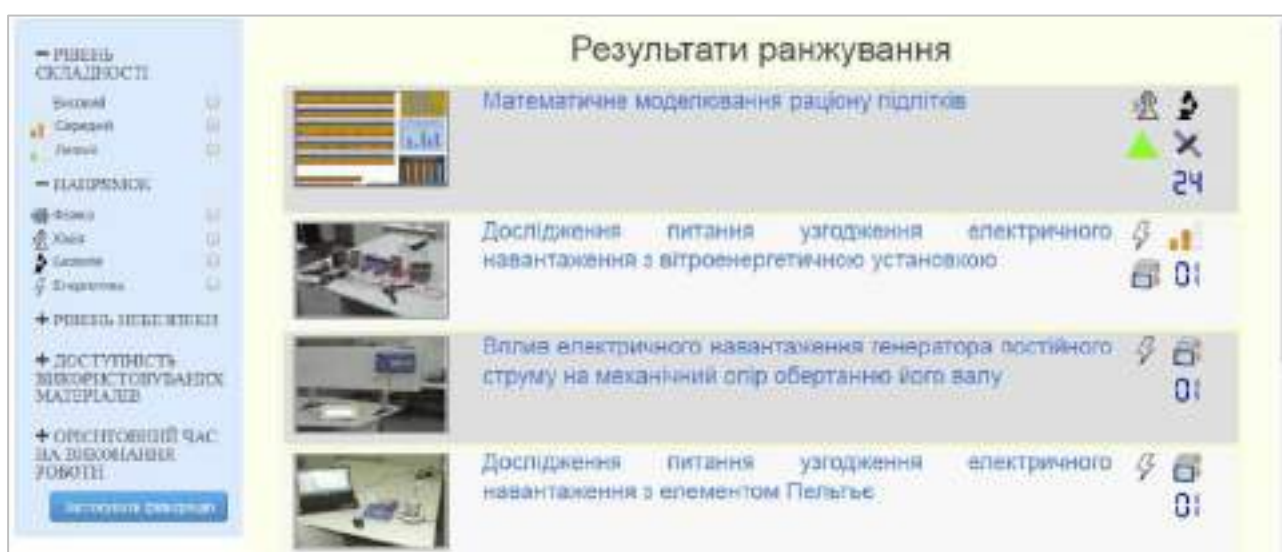


Рис. 4.12. Результати ранжування навчальних проектів

Крім цього у системі наявні механізми фільтрації альтернатив, які є елементами таксономії. Це дозволяє отримати рейтингові списки саме для тих підмножин об’єктів, які цікавлять ОПР.

4.2. Інструментальний засіб «Оцінка досягнень»

Основна мета створення Національного рейтингового списку учнів – учасників та переможців Міжнародних і фінальних етапів Всеукраїнських предметних олімпіад, переможців конкурсів-захистів науково-дослідних робіт учнів – членів Малої академії наук України – виявлення обдарованої молоді, яка своїми інтелектуальними досягненнями заслуговує на увагу та підтримку з боку держави.

Головна проблема полягала у виробленні науково-обґрунтованого підходу для коректного обрахунку рейтингових балів учнів, адже у кожного інтелектуального змагання існує своя числова шкала оцінки результатів учасників. Так, на олімпіаді з інформатики максимально можлива кількість балів, яку може набрати переможець – 800, а наприклад, на олімпіаді з математики – 56. Слід враховувати також, що для різних олімпіадних предметів складність конкурсних завдань є не однаковою, отже не однаковим буде і рівень успішності виконання учнями завдань.

Для конкурсу-захисту, що проводить МАН – своя система оцінювання досягнень. Тут підсумкові результати учнів складаються з суми балів за контрольну роботу з фахової дисципліни, заочне оцінювання самої наукової роботи та бали за виступ. Хоча у будь-якій секції конкурсу МАН теоретичний максимум є 100 балів, але навіть у цьому випадку робити висновок про “інтелектуальну перевагу” учня однієї секції над учнем іншої на основі звичайного порівнювання конкурсних балів є некоректним. Проблема полягає в тому, що (як і в олімпіадах) в різних секціях контрольні роботи можуть відрізнятися складністю. Крім того, “ступінь жорсткості” оцінювання результатів конкурсів залежить від конкретного журі, членами якого є живі люди зі своїми суб’єктивними судженнями.

Таким чином, дійсно виникає необхідність у чіткій формалізації підходів до обчислення рейтингових показників успішності учнів не залежно від типу інтелектуального змагання. Необхідно побудувати таке перетворення протокольних результатів (турнірних балів), яке було б однаковим для всіх конкурсів і олімпіад, і при цьому дозволяло б отримувати підсумковий результат (рейтинговий бал), в якому би успішність того чи іншого учня на пряму залежала від ступеню його переваги над іншими учнями в межах конкурсу. Іншими словами, треба враховувати не просто абсолютний результат кожного учасника, а те – на скільки він є визначним у порівнянні з представниками тієї ж галузі знань. Це дозволить порівнювати в одній шкалі інтелектуальні досягнення учнів в конкурсах та олімпіадах різної тематики

дисциплін (фізико-технічні, природничо-математичні, суспільні, гуманітарні і т.п.).

Вся інформація по учням, що входять до ТМІАС-у, зберігається в системі у вигляді онтологічного графу, вершинами якого є назви регіонів України, відділення та секції конкурсу МАН, предмети, з яких проводиться Всеукраїнська та міжнародні олімпіади, а також назви самих інтелектуальних змагань (рис. 4.13). Кожному листку графа відповідає конкретний учень, інформація про якого (школа та перелік інтелектуальних досягнень) міститься у цьому вузлі.

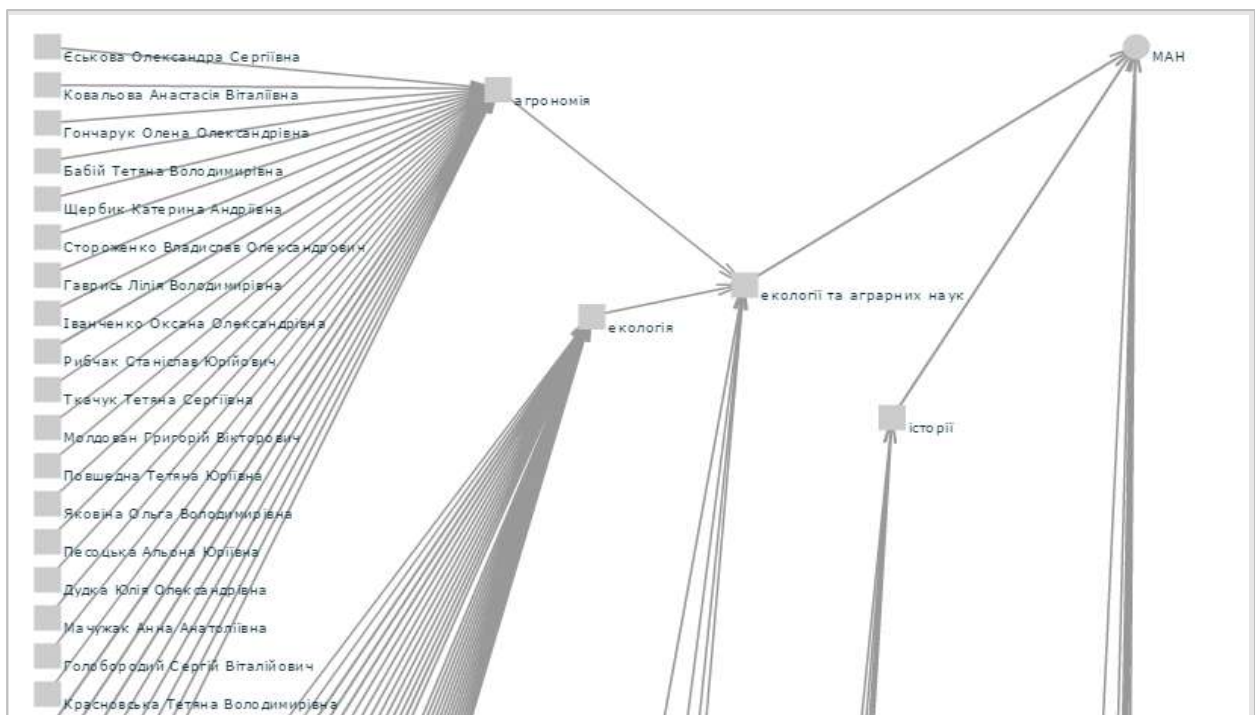


Рис. 4.13. Фрагмент онтологічного графу Національного рейтингу

4.2.1. Призначення системи та її основні можливості

Для отримання рейтингового списку (далі – рейтинг) навчальних досягнень учнів розроблений інструментарій для візуалізації актуальних даних по кожному з учасників, відповідний алгоритм обрахунку їх рейтингових балів та достатньо потужні засоби для проведення аналітичних досліджень по всім даним, що входять до системи: учні, регіони, школи, напрями, секції і т. п. Система апробована і впроваджена у Національному центрі «Мала академія

наук України».

Розроблений підхід та сам алгоритм отримання рейтингових показників учасників конкурсних змагань є науково-обґрунтованим і може бути адаптованим до будь-яких навчальних змагань (олімпіад, конкурсів, турнірів тощо), учасники яких характеризуються атрибутами (табл. 4.1):

- прізвище, ім'я, по-батькові;
- назва конкурсного змагання;
- рік проведення;
- регіон (адміністративна область України);
- навчальний заклад (повна юридична назва);
- клас (група) учасника;
- науковий напрям навчального змагання, в якому учасник приймав участь;
- секція напрямку (у разі існування);
- науковий керівник учасника;
- вчитель (викладач, керівник гуртка, студії тощо), який веде напрям навчально-пізнавальної діяльності (викладає навчальний предмет (дисципліну) за тематикою конкурсного змагання);
- зайняте місце (I, II, III / учасник);
- результат учасника (сумарний конкурсний бал, який в загальному випадку може складатись з різних складових, наприклад: виступ, контрольна робота, експеримент і т.п.

Табл. 4.1 фактично складається з двох частин. Перша частина містить інформацію, яка є незмінною для всіх учасників, результати виступів яких наведені у другій частині. Це – рік проведення конкурсу, назва конкурсу, назва напрямку та секції, в якій змагались учасники. Якщо в певному конкурсі напрями додатково не розділяються на окремі секції, відповідний рядок першої частини не заповнюється.

Таблиця 4.1. Формат даних для побудови підсумкової рейтингової таблиці

1.	Рік проведення конкурсу																						
2.	Конкурс																						
2.1.	Напрямок																						
2.1.1.	Секція																						
2.1.1.1.	Максимальний бал																						
3.	Кількість складових результатів																						
3.1.	Назва складової 1																						
3.1.1.	Максимальний бал																						
...	...																						
3. К.	Назва складової К																						
3. К.1.	Максимальний бал																						
№	Учасник	Клас	Регіон	Школа	Науковий керівник	Посада	Місце	Підсумковий результат	Складова результату 1	Складова результату 2	...	Складова результату К											
1	ПІБ_1	<Клас учасника>	<Регіон учасника>	<Школа учасника>	Секція	Науковий керівник	Секція	Секція	Підсумковий результат	Підсумковий результат	...	Підсумковий результат											
	...																						
N	ПІБ_N	<Клас учасника>	<Регіон учасника>	<Школа учасника>	<ПІБ>	<посада>	<Місце>	<Бал V _N >	V _{N1}	V _{N2}	...	V _{Nk}											

Для коректного обрахування рейтингових балів необхідна обов'язкова інформація по теоретично-максимально можливим значенням як підсумкових результатів учасників кожного конкурсу, так і максимально допустимих значень складових оцінок у разі їх наявності. Інформація про кількість складових підсумкового результату також заноситься до першої частини таблиці.

Заголовки табл. 4.1 обов'язково складаються з 2 рядків. Стовпчики «Учасник», «Клас», «Регіон», «Школа», «Підсумковий результат» – є обов'язковими і саме з такими назвами. Стовпчики, назви яких складаються з 2-х рядків формуються за таким правилом 1-й рядок – назва атрибуту, 2-й рядок – батьківський атрибут. Таким чином, вся інформація по конкурсу зберігається у вигляді ієрархічної структури. Будь-яка заповнена ячійка таблиці відноситься до учасника, прізвище якого розташоване у тому ж рядку, що і дана ячійка.

В загальному випадку передбачається, що будь-яке конкурсне змагання проводиться за певними напрямками (атрибут I-го рівня), а в середині кожного напрямку – за окремими секціями (атрибути II-го рівня).

Підсумковий бал кожного учасника заноситься окремо в таблицю, крім того заносяться результати по кожному різновиду конкурсних завдань (як правило, сума цих результатів і є підсумковим балом учасника конкурсного змагання).

Підсумкові протоколи виступів учасників рейтингу по кожній секції НЦ «МАН України» у вигляді екселівських файлів-таблиць є вхідною інформацією для системи.

Для конкурсу МАН атрибутами I-го рівня (напрямами) будуть назви наукових відділень: комп'ютерні науки, математика, мовознавство і т.д (всього 12); атрибути II-го рівня – назви секцій для кожного напрямку; конкурсні завдання – заочне оцінювання роботи, контрольна робота, виступ.

За кожне з цих завдань учасник отримує певні бали, сума яких і є його підсумковим результатом. Так, <Бал учасника ПБ_1> = $b_{11} + b_{12} + b_{13}$. За наявності підготовлених в наслідок автоматичної обробки вхідних даних

можливим є побудова онтологічного графу. Тобто спочатку обираються атрибути об'єктів, за якими буде проводитись таксономізація.

В даному випадку для НЦ «МАН України» такими атрибутами є «напрями і секції» та «регіони». Після визначення таксономічної сітки майбутнього графу відбувається автоматична конвертація атрибутивних даних таблиці в графову структуру, на основі якої стає можлива побудова графу за допомогою системи «Конфор» [83] або інших засобів.

Після побудови таксономії наступним етапом є безпосереднє завантаження атрибутивних даних як властивостей об'єктів. Після цього граф сформований і може бути відображений різними засобами. Приклад такого графу зображено на рис. 4.13. Слід зазначити, що для графу, представленому у форматі xml, існує його взаємно однозначне відображення у базу даних та навпаки.

Враховуючи значний можливий об'єм онтологічного графу «Рейтинг», для забезпечення оперативної роботи з ним з боку користувача, він безпосередньо зберігається у базі даних.

Підхід, вироблений в процесі побудови Національного рейтингу інтелектуальних досягнень на основі аналізу та обробки значних масивів вхідних даних, структурованих у табличному вигляді, фактично був трансформований у технологію комплексування даних, що зберігаються у файлах форматів MS Excel та MS Word. Така необхідність викликана тим, що переважна кількість актуальної інформації, що слід використовувати при розв'язку різного роду прикладних задач, створюється та супроводжується засобами пакету MS Office.

Для обрахунку рейтингового балу кожного учасника може бути застосований розроблений алгоритм конкурентної нормалізації, що дає змогу порівнювати навчальні досягнення представників різних наукових напрямів та секцій. Цей алгоритм враховує не тільки кількість набраних учасником конкурсних балів, а також й рівень складності самого конкурсу. Визначимо необхідні параметри для його застосування.

Нехай маємо n учасників у певній секції, кожний з яких набрав b_i конкурсних балів ($i = \overline{1, n}$). Як відомо, специфіка турніру полягає в тому, що остаточний бал є сумою трьох складових оцінок – балів за контрольну роботу з фахової дисципліни (c_i), заочне оцінювання роботи (z_i), захист науково-дослідної роботи (v_i). За регламентом максимально можливою сумою є 100 балів. Пропонується такий спосіб обчислення ідеально-потенційно можливого значення балу. Будемо вважати, що «ідеальний» суб'єкт (учень) отримав би такий бал:

$$b_{ideal} = \frac{c_{max} + z_{max} + v_{max} + 100}{2},$$

де c_{max} , z_{max} , v_{max} – максимальні оцінки по секції (серед усіх учасників) з контрольної роботи, заочного оцінювання та захисту відповідно.

Відповідно до алгоритму конкурентної нормалізації ОНР повинен визначити віддаленість середнього балу від 0, щоб забезпечити ступінь переваги L очікувано-максимального балу B_* над середнім балом B_{avg} та визначити коефіцієнт стискання. Далі визначається величина стискання $\mu = (L - 1) / (2B_* - 1)$ і остаточні рейтингові бали.

Будемо вважати, що шуканий рейтинговий бал найкращого учня кожної секції повинен бути утричі більшим від середнього балу по секції ($L=3$), який практично у кожному випадку близький до найменшого балу серед призерів. Такі міркування можна обґрунтувати тим, що за «Положенням про Всеукраїнські учнівські олімпіади...», затвердженого наказом Міністерства освіти і науки № 1099 від 22.09.2011, розподіл призерів відбувається у пропорції 1:2:3, причому загальна кількість призерів повинна обмежуватись половиною від кількості учасників.

За статистичними дослідженнями результатів учасників конкурсу захисту МАН протягом 2013–2017 рр. значення $B_* \approx 0.75$. Тоді $\mu = 3.99$, таким чином всі необхідні параметри для алгоритма конкурентної нормалізації визначені, і він може бути застосований для конкурсу-захисту МАН.

На рис. 4.14 у графічному вигляді зображені залежності рейтингових балів у шкалі [0..1000] від конкурсних за стобальною шкалою для секцій загальної біології, медицини, психології та хімії відділення хімії та біології. З наведеного рисунку видно, що високий конкурсний бал сам по собі зовсім не означає, що буде високим і рейтинговий бал, адже згідно з алгоритмом величина рейтингового балу залежить від того, наскільки впевнено конкурсант випередив своїх безпосередніх суперників. Так конкурсний бал переможця секції психології (91.46) є більшим, ніж переможця секції загальної біології (90.39), проте рейтингові бали цих учасників розмістились в іншому порядку. Це не важко пояснити.

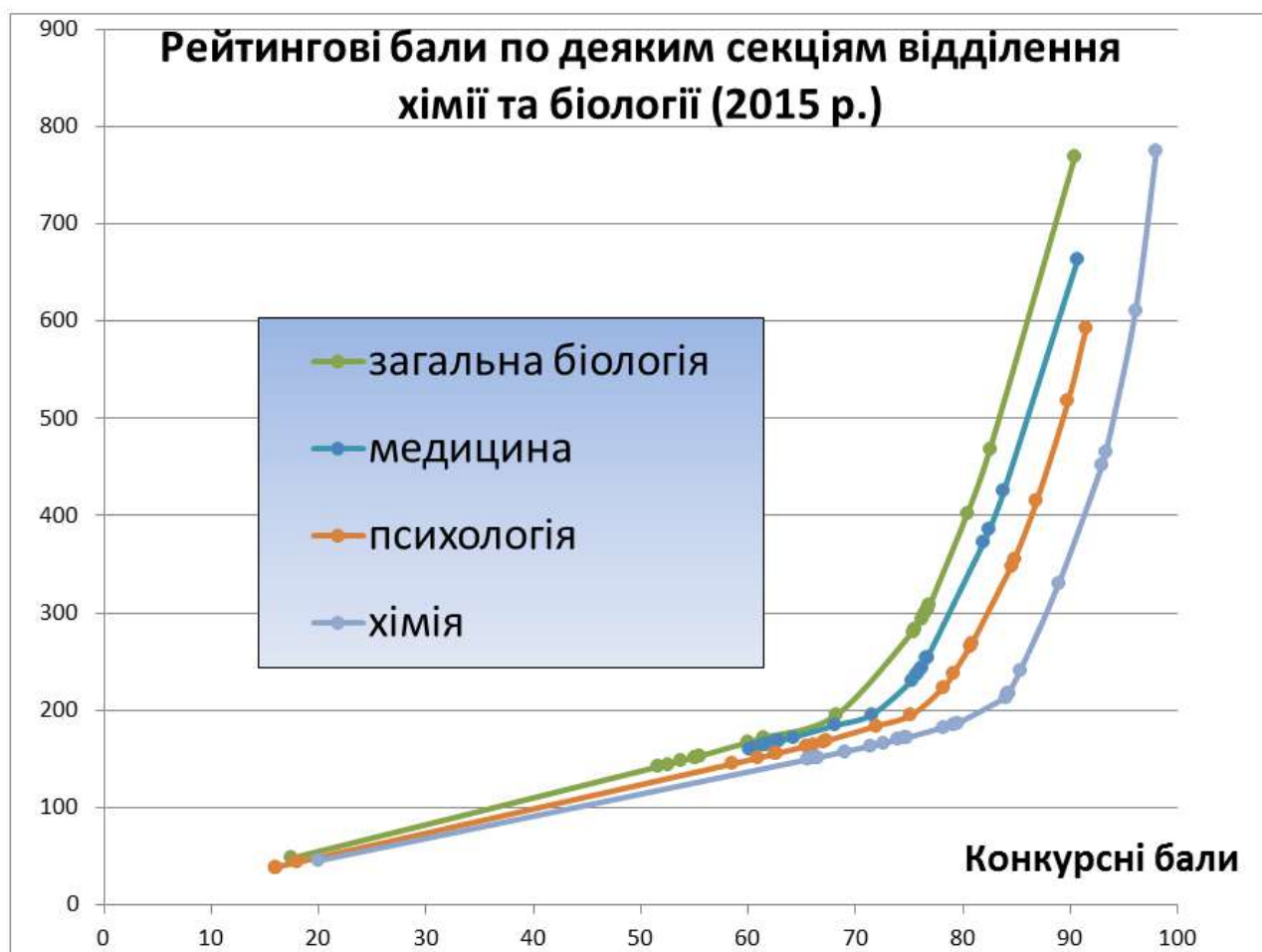


Рис. 4.14. Графічне представлення рейтингових балів відділення хімії та біології

У секції загальної біології лише троє учнів спромоглися набрати бал

вищий ніж 80. В секції психології таких школярів шестеро. Крім того, переможець в секції психології випередив найближчого свого переслідувача на величину значно меншу ($91.46 - 89.8 = 1.66$), ніж переможець секції загальної біології ($90.39 - 82.5 = 7.89$). Іншими словами, перемога у секції загальної біології виявилася суттєвішою, тому що тут значно менша кількість учнів набрала високий бал ніж у секції психології, і хоча результати переможців є величинами майже одного порядку, проте у другого з них значно кращі досягнення у порівнянні зі своїми найближчими конкурентами.

Підсумкова рейтингова таблиця доступна на офіційному сайті НЦ «Мала академія наук України» (рис. 4.15).

№ арт	№	ІПН	РЕЗУЛЬТАТ ОЦІНЮВАННЯ	РЕГІОН (ОБЛАСТЬ)	ІМ'Я	ІНТЕРНАЦІОНАЛЬНА СПЕЦІАЛІЗАЦІЯ	ОЛІМПІАДА	РІВ	СУМАРНА ОЦІНКА
1	1	Король Роман Богданович	2014	Тернопіль	18 фізика	1 інформ	1 матем	2013	3950.65
2	2	Дубня Андрій Іванович	2016	Черкаська	1 інформ	1 інформ	1 матем	2014	3875.61
3	3	Сайкош Діана Степанівна	2015	Черкаська	1 інформ	1 інформ	1 інформ	2015	3675.00
4	4	Шукаєва Олександра Кошовна	2016	Харківська	1 інформ	1 інформ	1 інформ	2014	3557.20
5	5	Мисюк Олександр Сергійович	2016	Харківська	1 інформ	1 інформ	1 інформ	2015	3514.35

Рис. 4.15. Початковий фрагмент рейтингової таблиці

Користувач має можливість конкретизувати свої побажання з відображення певної інформації за допомогою механізму фільтрації даних

(рис. 4.16). Так, наприклад, можна переглядати результати учасників з наперед заданими значеннями полів: рік події, регіон, турнір, напрям, секція і т.п. (рис. 4.17).

ФІЛЬТРУВАТИ •

[Очистити все](#) сортувати

Відображення:

Стовпчикова діаграма ▾

Звичайний режим ▾

Групування: турніри ▾

Рейтинг по рокам:

▾ — ▾

▼ **РЕГІОНИ**

▼ **Властивості**

Міжнародна олімпіада:

▼ Місце

I

II

III

учасник

▼ Предмет

астрономія

біологія

географія

екологія

інформатика

історія

математика

фізика

хімія

Олімпіада:

▼ Місце

I

II

III

учасник

▼ Предмет

англійської мови

астрономії

біології

географії

екології

економіки

інформатики

інформаційних технологій

іспанської мови

історії

математики

німецької мови

педагогіки і психології

правознавства

російської мови і літератури

трудового навчання

української мови і літератури

фізики

фізичної культури і спорту

МАН:

▼ Місце

I

II

III

учасник

▼ Напрямок

екології та аграрних наук

економіки

історії

історія України

історичне краєзнавство

археологія

всесвітня історія

етнологія

комп'ютерних наук

літературознавства, фольклористики та мистецтвознавства

математики

мовознавства

наук про Землю

технічних наук

фізики і астрономії

філософії та суспільствознавства

хімії та біології

Рис. 4.16. Фільтрація даних у системі «Оцінка досягнень»

Щоб перейти в діаграмний режим представлення даних, необхідно в пункті меню у блоці “Відображення” обрати один із наявних видів діаграм,

наприклад *стовпчикова діаграма* (рис. 4.18), де зображений рівень досягнень по регіонах.

№ р.	Рр.	ІМ.	РІВ.	РЕГІОН (ОБЛАСТЬ)	МАП	ОБ'ЄКТ (Д.)	РР.	СУМАРНА ОЦІНКА
1	20	Іваненко Людмила Вікторівна	2010	Дніщина		Фінанси, грошовий обіг і кредит	2013	1367.23
							2014	
							2015	
2	110	Голубничий Станіслав Валентинівич	2010	Львівська		Фінанси, грошовий обіг і кредит	2014	1550.00
							2015	
							2016	
3	223	Халавка Олена Андріївна	2010	Львівська		Фінанси, грошовий обіг і кредит	2014	1210.82
							2015	
4	262	Клець Олена Навродівна	2010	Львівська		Фінанси, грошовий обіг і кредит	2012	1100.64
							2015	

Рис. 4.17. Фрагмент рейтингового списку по секції фінанси, грошовий обіг і кредит

Меню фільтрування (рис. 4.16) дозволяє: представляти дані у вигляді агрегованої таблиці, стовпчикової діаграми, графіка та кругової діаграми; змінювати ряди даних (транспонування); групувати дані по турнірам, рокам, секціям, зайнятим місцям та секціям або предметам; формувати рейтинг тільки за певний часовий інтервал; здійснювати сортування за різними ключами; формувати рейтинги для певних регіонів та ознак; створювати рейтинг тільки за певними турнірами, секціями та предметами або іншими ознаками і т.п.

Створена система орієнтована на не підготовлених користувачів, серед яких окремо можна виділити: учнів, педагогічних працівників та менеджерів різних рівнів у галузі освіти. Система дозволяє створювати різного роду запити для отримання даних по тим ключовим ознакам, які цікавлять користувача, а

також здійснювати аналітичні дослідження за допомогою вбудованого інструментарію графічної візуалізації даних. В першому випадку йдеться про фільтрацію даних, у другому – про їх відображення не в табличному вигляді, а як елементи стовпчикових та кругових діаграм, а також різного роду графіків.

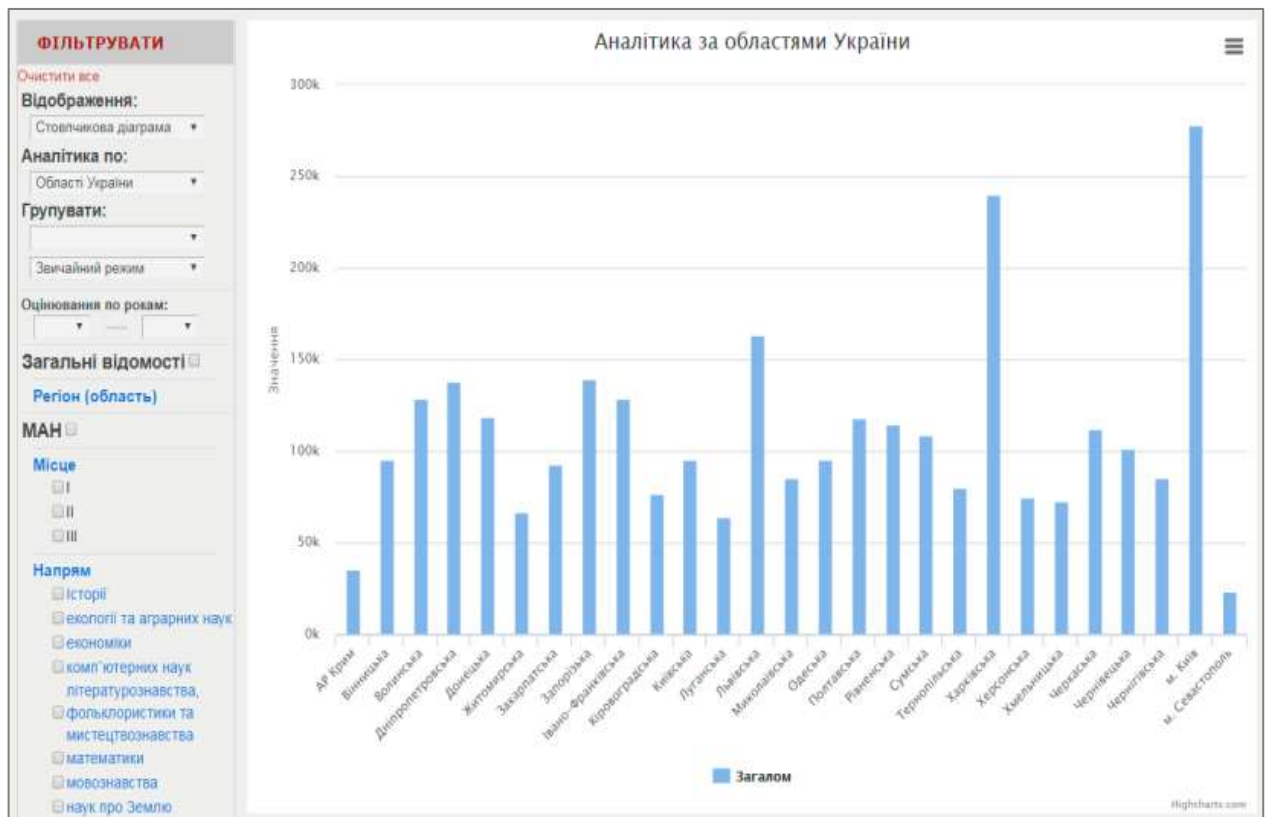


Рис. 4.18. Стовпчикова діаграма успішності учнів за регіонами України.

4.2.2. Формування запитів до системи оцінювання досягнень учнів.

Отже, для фільтрації даних користувач має можливість оперувати з такими атрибутами (рис. 4.15):

- *Рік закінчення навчального закладу.* Користувач має можливість отримувати інформацію по таким учням, які завершують навчання у закладі середньої освіти в заданому році. В таблиці залишаться тільки ті учні, які завершать (завершили) своє навчання в обраному році;
- *Рік події.* Задається рік, в якому відбулося інтелектуальне змагання (МАН, олімпіада), що цікавлять користувача. Сформується вибірка тільки таких учнів, які приймали участь у конкурсі в обраному році.

- *Оцінювання по рокам.* Користувачем обирається період, протягом якого відбувались інтелектуальні змагання. У результуючій таблиці рейтинги учнів будуть рахуватись тільки за цей проміжок часу.
- *Навчальний заклад.* Користувач задає ключове слово з назви навчального закладу або його номер. Після цього з'явиться перелік тих навчальних закладів, у назві яких присутнє задане значення. Користувачеві необхідно обрати саме той навчальний заклад, який його цікавить.
- *Регіон (область).* Надається можливість обрати окремі області України за бажанням користувача. У результуючій таблиці буде виведений рейтинговий список тільки учнів обраних регіонів.
- *МАН.Місце.* Користувач при необхідності може задати конкретне призове місце. Сформується вибірка тільки з тих учнів, які посіли саме таке призове місце.
- *МАН.Напрям.* Користувач має дві можливості: обрати або окремі напрями конкурсу, або окремі секції, що його цікавлять. В першому випадку достатньо поставити відмітку перед назвою конкретного напрямку. У другому – «клікнути» мишею по назві напрямку і з переліку тих секцій, що з'являться обрати необхідні.

Таким чином, наведені атрибути належать до шести груп. При створенні запиту, в якому користувач задав значення атрибутів з різних груп, система формує таку вибірку даних, які задовольняють одночасно кожне задане значення атрибуту. При цьому значення атрибутів *Рік закінчення навчального закладу*, *Рік події*, *Оцінювання по рокам* є однозначними. Значення атрибутів *Навчальний заклад*, *Регіон*, *МАН* формується не однозначно, адже користувач для кожного з цих атрибутів може обирати декілька різних значень (наприклад, декілька різних областей, різні напрями, секції та призові місця). В цьому випадку в межах атрибуту значення формується як будь-яке з заданих користувачем.

Наведений на рис. 4.19 варіант запиту до системи інтерпретується так. Вивести рейтинговий список за 2013-2015 рр. всіх учнів, які задовольняють

таким значенням атрибутів:

- ✓ закінчили навчальний заклад у 2016 р.,
- ✓ є представниками **або** Вінницької, **або** Дніпропетровської, **або** Харківської обл. **або** міста Києва,
- ✓ посіли **або** I **або** II місце,
- ✓ приймаючи участь **або** в секції «математичне моделювання» **або** в секції «прикладна математика» напряму «математика», **або** в будь-якій секції напряму «комп'ютерні науки».

ФІЛЬТРУВАТИ

Очистити все

Рік закінчення навчального закладу: ▼

Рік події: ▼

Оцінювання по рокам: — ▼

Навчальний заклад:

Загальні відомості

Регіон (область)

- АР Крим
- Вінницька
- Волинська
- Дніпропетровська
- Донецька
- Житомирська
- Закарпатська
- Запорізька

- Івано-Франківська
- Кіровоградська
- Київська
- Луганська
- Львівська
- Миколаївська
- Одеська
- Полтавська
- Рівненська
- Сумська
- Тернопільська
- Харківська
- Херсонська
- Хмельницька
- Черкаська
- Чернівецька
- Чернігівська
- м. Київ
- м. Севастополь

МАН

Місце

- I
- II
- III

Напрямок

- Історії
- екології та аграрних наук
- економіки
- комп'ютерних наук літературознавства,
- фольклористики та мистецтвознавства
- математики
 - математичне моделювання
 - прикладна математика
 - математика

Рис. 4.19. Приклад сформованого запиту до системи «Оцінка досягнень».

За цим запитом сформується відповідна вибірка учнів, яка відобразиться у таблиці. При цьому рейтингові бали будуть складатись тільки з тих, які отримали відібрані учні протягом 2013-2015 рр.

4.2.3. Проведення аналітичних досліджень

Для візуалізації результатів запитів не в табличному вигляді, а графічному необхідно обрати значення атрибутів (як описано для формування запитів), і задати конкретні параметри відображення діаграм (рис. 4.20).

Система оперує трьома параметрами, що впливають на графічне відображення та кількість рядів даних (рис. 4.21):

- *Відображення.* Користувач обирає конкретний вигляд графічного представлення даних (стовпчикова чи кругова діаграма або графік).
- *Аналітика по.* Цей параметр фактично задає назви рядів даних для сформованої вибірки, які розміщуються по осі ОХ (області України, школи, учні, роки, секції/предмети).
- *Групувати.* Обрання способу деталізації елементів рядів даних за тим чи іншим показником (турніри, роки, секції/предмети, місця, класи).

The screenshot shows a filter interface titled "ФІЛЬТРУВАТИ" (Filter). It includes several sections for configuring data visualization:

- Очистити все** (Clear all)
- Відображення:** A dropdown menu set to "Стовпчикова діаграма" (Column chart).
- Аналітика по:** A dropdown menu set to "Області України" (Regions of Ukraine).
- Групувати:** Two dropdown menus: "Роки" (Years) and "Звичайний режим" (Normal mode).
- Оцінювання по рокам:** Two dropdown menus for the years "2013" and "2015".
- Навчальний заклад:** An empty text input field.
- Загальні відомості** (General information) section with a checkbox.
- Регіон (область)** (Region) section with a list of regions and checkboxes:
 - АР Крим
 - Вінницька
 - Волинська
 - Дніпропетровська
 - Донецька
 - Житомирська
 - Закарпатська
- A central list of regions with checkboxes:
 - Запорізька
 - Івано-Франківська
 - Кіровоградська
 - Київська
 - Луганська
 - Львівська
 - Миколаївська
 - Одеська
 - Полтавська
 - Рівненська
 - Сумська
 - Тернопільська
 - Харківська
 - Херсонська
 - Хмельницька
 - Черкаська
 - Чернівецька
 - Чернігівська
 - м. Київ
 - м. Севастополь
- МАН** (MAN) section with a checked checkbox.
- Місце** (Place) section with three radio buttons:
 - I
 - II
 - III
- Напрямок** (Direction) section with a list of subjects and checkboxes:
 - Історії
 - екології та аграрних наук
 - економіки
 - комп'ютерних наук
 - літературознавства,
 - фольклористики та мистецтвознавства
 - математики
 - мовознавства
 - наук про Землю
 - технічних наук
 - фізики і астрономії
 - філософії та суспільствознавства
 - хімії та біології

Рис. 4.20. Параметри та атрибути для побудови діаграм та графіків.

Рис. 4.21. Значення параметрів для побудови діаграм

Наведений на рис. 4.20 варіант запити до системи з графічним відображенням отриманої вибірки інтерпретується так. Обчислити сумарні рейтинги по обраним областям, представники яких приймали участь у секціях напрямів: «комп'ютерних наук», «математики», «наук про Землю», «технічних наук», «фізики і астрономії». Результат такого запити представлений на рис. 4.22.

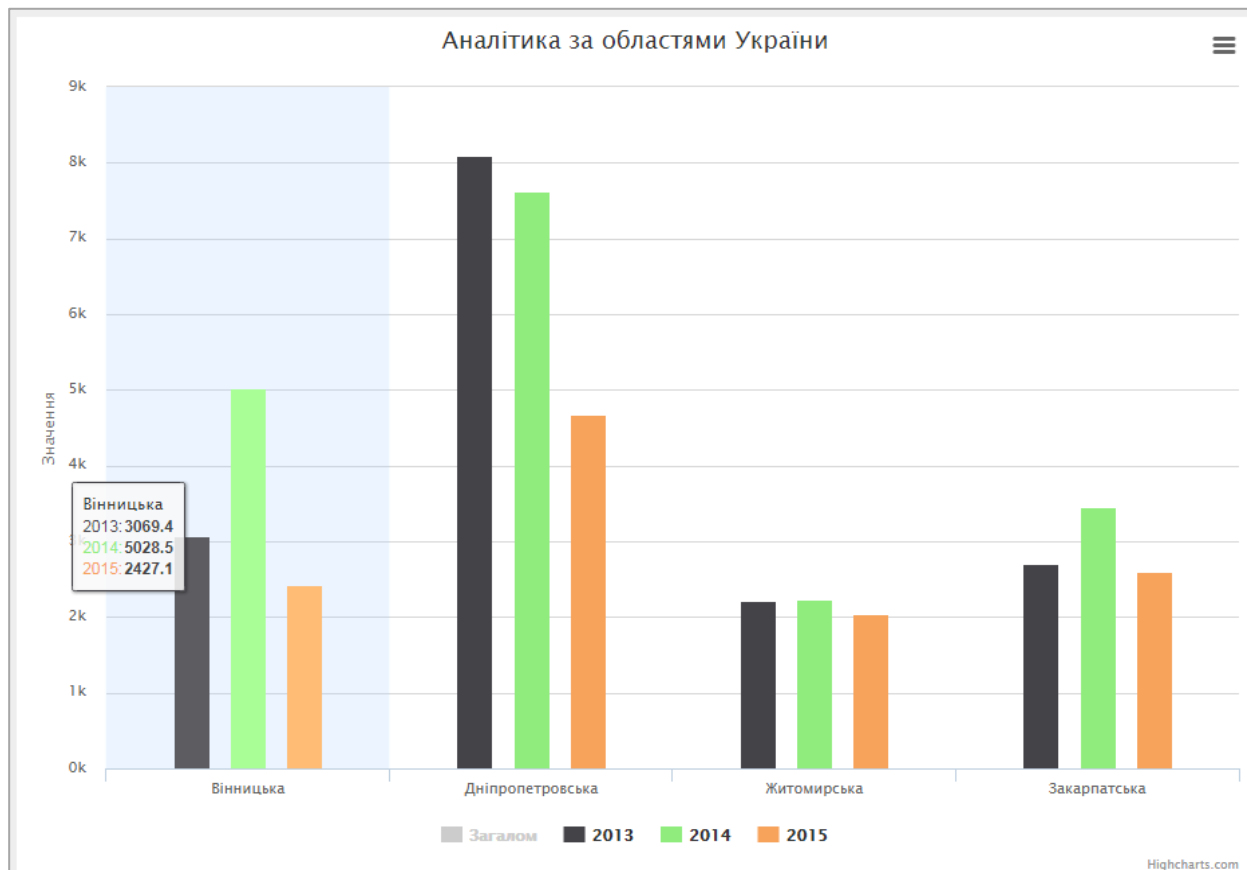


Рис. 4.22. Приклад графічного відображення запити до системи.