

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна

Кваліфікаційна наукова  
праця на правах рукопису

**КУЗЕНКОВ Олександр Олександрович**

УДК 519.8

## **ДИСЕРТАЦІЯ**

# **ДИНАМІЧНІ ЗАДАЧІ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН: МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ, АЛГОРИТМИ, ПРАКТИЧНЕ ЗАСТОСУВАННЯ**

01.05.02 – математичне моделювання та обчислювальні методи

Подається на здобуття наукового ступеня доктора технічних наук

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

\_\_\_\_\_ О.О. Кузенков

Харків 2026

## АНОТАЦІЯ

Кузенков О.О. Динамічні задачі оптимального розбиття множин: математичні моделі, методи, алгоритми, практичне застосування. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 01.05.02 – математичне моделювання та обчислювальні методи. – Харківський національний університет імені В.Н. Каразіна, МОН України, Харків, 2026.

Дисертаційна робота є подальшим розвитком теорії оптимального розбиття множин, що є новим розділом нескінченновимірною математичного програмування з булевими змінними, для динамічних задач оптимального розбиття, у яких для отримання оптимального розбиття, в динамічній постановці, застосовуються сучасні чисельні методи, субградієнтні методи з розтягом простору та інші методи негладкої оптимізації.

Розроблено та теоретично обґрунтовано методи та алгоритми розв’язання динамічних задач оптимального розбиття множин в ускладнених постановках, а саме: з фіксованими центрами, з відшукуванням координат центрів та в умовах невизначеності. Методи та алгоритми застосовані до сформульованих в роботі трьох математичних моделей. Розроблено та теоретично обґрунтовано методи і алгоритми розв’язання динамічних задач у випадках коли параметри задачі змінюються з плином часу. Запропоновані алгоритми засновані на синтезі методів теорії оптимального розбиття множин, методів розв’язання диференціальних рівнянь та їх систем, а також з модифікаціями  $r$ -алгоритму Шора для розв’язання задач недиференційовної оптимізації.

Сформульовані в роботі математичні моделі, методи та алгоритми їх розв’язання застосовані для розв’язання трьох прикладних задач, а саме задачі оптимального розміщення вогневих груп, побудови оптимальних траєкторій для тривимірного друку та оптимального розбиття регіону на

зони забезпечення медичними закладами при спалахах інфекційних захворювань. Розроблено комплекси комп'ютерних програм для реалізації запропонованих алгоритмів розв'язання задач оптимального розбиття множин в тому числі для означених прикладних задач.

**Ключові слова:** нескінченновимірне математичне програмування, теорія оптимального розбиття множин, динамічна задача, чисельні методи, недиференційовна оптимізація,  $r$ -алгоритм Шора.

## ABSTRACT

Kuzenkov O.O. Dynamic problems of optimal set partitioning: mathematical models, methods, algorithms, practical applications. – Qualification scientific work as a manuscript.

Dissertation for the degree of Doctor of Technical Sciences in specialty 01.05.02 – Mathematical Modeling and Computational Methods. – V.N. Karazin Kharkiv national university, Kharkiv, 2026.

The dissertation represents a further development of the theory of optimal set partitioning, which is a new branch of infinite-dimensional mathematical programming with Boolean variables, applied to dynamic optimal partitioning problems. To obtain an optimal partition in a dynamic setting, modern numerical methods are used, including subgradient methods with space stretching and other nonsmooth optimization techniques.

Methods and algorithms for solving dynamic problems of optimal set partitioning in complex formulations have been developed and theoretically substantiated, namely: with fixed centers, with the determination of center coordinates, and under conditions of uncertainty. The methods and algorithms have been applied to the three mathematical models formulated in the work. Methods and algorithms for solving dynamic problems in cases where problem parameters change over time have also been developed and theoretically substantiated. The proposed algorithms are based on a synthesis of methods from the theory of optimal set partitioning, methods for solving differential equations and their

systems, as well as modifications of Shor's r-algorithm for solving non-differentiable optimization problems.

The mathematical models, methods, and algorithms for their solution formulated in this work have been applied to solve three practical problems, namely the problem of optimal placement of fire units, the construction of optimal trajectories for 3D printing, and the optimal division of a region into zones for medical facility coverage during infectious disease outbreaks. Computer program complexes have been developed to implement the proposed algorithms for solving optimal set partitioning problems, including for the specified practical tasks.

**Keywords:** infinite-dimensional mathematical programming, theory of optimal set partitioning, dynamic problem, numerical methods, nonsmooth optimization, Shor's r-algorithm.

#### СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Kiseleva O., Yakovlev S., Prytomanova O., Kuzenkov O. Mathematical Modeling of Regional Infectious Disease Dynamics Based on Extended Compartmental Models. *Computation* (2025), 13(8), 187. ISSN: 2079-3197, DOI: 10.3390/computation13080187 (*Scopus, Q1*)
2. Kiseleva O., Yakovlev S., Chumachenko D., Kuzenkov O. Exploring Bifurcation in the Compartmental Mathematical Model of COVID-19 Transmission. *Computation* 2024, 12, 186. ISSN: 2079-3197, DOI: 10.3390/computation12090186 (*Scopus, Q1*)
3. Busher V., Chorny O., Kuzenkov O., Tryputen M., Kuznetsov V., Rumiantsev V. Investigation of the Bifurcation Properties of the Dynamics of a Biological Population Based on a Logistic Model. *Lecture Notes in Networks and Systems*, vol. 463. Springer, Cham. P. 203–212, ISSN: Print: 2367-3370, Electronic: 2367-3389, DOI: 10.1007/978-3-031-03877-8\_18 (*Scopus, Q4*)
4. Stopkin V., Nikolenko A., Kuznetsov V., Tryputen M., Kuzenkov O. The Urgency of Using Adaptive Observers to Identify the Parameters of the DC. *Smart*

Technologies in Urban Engineering. Vol. 536, P. 213–224. ISSN: Print: 2367-3370, Electronic: 2367-3389. DOI: 10.1007/978-3-031-20141-7\_20 (*Scopus, Q4*)

5. Kuzenkov O., Dubovyk V. Software implementation of the COVID-2019 spread simulation algorithm using the theory of optimal set partitioning.

Актуальні проблеми автоматизації та інформаційних технологій. № 24 (2020). – ДНУ, 2020. С. 96–103. ISSN-print: 2312-119X, DOI: 10.15421/432011

6. Кісельова О. М., Кузенков О. О. Програмна реалізація алгоритму розв'язання динамічної задачі побудови оптимальних траєкторій тривимірного друку. Актуальні проблеми автоматизації та інформаційних технологій. № 25 (2021). – ДНУ, 2021. С. 182–197. ISSN-print: 2312-119X, DOI: 10.15421/432119

7. Kiseleva E., Kuzenkov O., Dubovyk V. Software implementation of the algorithm for solving the dynamic problem of optimal placement of fire groups.

Актуальні проблеми автоматизації та інформаційних технологій. № 26 (2022). – ДНУ, 2022. С. 137–145. ISSN-print: 2312-119X, DOI: 10.15421/432217

8. Kostenko O., Kuzenkov O. Solving the facility location problem by genetic algorithms. Computer-integrated technologies: education, science, production, Lutsk, 2023, Vol. 53, P. 125–131, ISSN: 2524-0552, DOI: 10.36910/6775-2524-0560-2023-53-19

9. Kuzenkov O., Lozovskyi A. Software development for solving the dynamic problem of optimal set partitioning with fixed centers. Актуальні проблеми автоматизації та інформаційних технологій. № 27 (2023). – ДНУ, 2023. С. 212–220. ISSN-print: 2312-119X, DOI: 10.15421/432320

10. Kuzenkov O., Kostenko O. Software development for solving the dynamic problem of optimal set partitioning with determination of the coordinates of their centers. Актуальні проблеми автоматизації та інформаційних технологій. № 28 (2024). – ДНУ, 2024. С. 306–317. ISSN-print: 2312-119X, DOI: 10.15421/432429

11. Kuzenkov O.O. Algorithms and methods in dynamic problems of optimal track placement in three-dimensional printing. System technologies: collection of scientific works. – Dnipro, 2025. – Volume 2 (№ 157). P. 154–169. ISSN: Print: 1562-9945, Online: 2707-7977, DOI: 10.34185/1562-9945-2-157-2025-16
12. Kuzenkov O. Algorithms and methods in dynamic problems of optimal placement of fire groups. Modern problems of metallurgy. № 28 (2025). – Дніпро: УДУНТ, 2025. P. 294–309. ISSN-print 1991-7848, ISSN-online 2707-9457, DOI: 10.34185/1991-7848.2025.01.17
13. Kuzenkov O., Dubovyk V. Development of a dynamic partitioning model and a software system for image segmentation in the implementation of computer vision for robotic systems. Сучасні проблеми моделювання. № 27 (2025). – МДПУ, 2025. С. 123–135. ISSN-print: 2313-125X, DOI: 10.33842/2313-125X-2025-19-123-135
14. Яковлев С. В., Кісельова О. М., Гарт Л. Л., Кузенков О. О., Закутній Д. В. Про динамічну задачу оптимального розбиття множин з фіксованими центрами. Сучасні проблеми моделювання. № 28 (2025). – МДПУ, 2025. С. 190–203. ISSN-print: 2313-125X, DOI: 10.33842/2313-125X-2025-30-190-203
15. Кісельова О., Притоманова О., Кузенков О. Про динамічну задачу оптимального розбиття множин із фіксованими центрами за умов невизначеності. Міжнародний науково-технічний журнал «Проблеми керування та інформатики», 2025, № 4. С. 6–23. ISSN: 2786-6491. URL: <https://jais.net.ua/index.php/files/article/view/525/599>
16. Kiseleva E. M., Prytomanova O. M., Kuzenkov O.O. Software implementation of an algorithm for solving a dynamic problem of optimal set partitioning under uncertainty. Science and transport progress. – Д.: УДУНТ, 2025, 3(111). P. 59–67. DOI: 10.15802/stp2025/342252
17. Kiseleva E., Kuzenkov O. On the dynamic problem of optimal set partitioning with determination of subset center coordinates. Bulletin of V .N. Karazin Kharkiv National University, Series “Mathematical modeling. Information

technology. Automated control systems”, (65). P. 35–47. DOI: 10.26565/2304-6201-2025-65-03

18. Kiseleva E., Hart L., Prytomanova O., Kuzenkov O. An algorithm to construct generalized Voronoi diagrams with fuzzy parameters based on the theory of optimal partitioning and neuro-fuzzy technologies. Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”, MoMLeT&DS-2019, Shatsk, Ukraine, June 2–4, 2019. P. 148–162, ISSN: 1613-0073, URL: <https://ceur-ws.org/Vol-2386/paper12.pdf> (*Scopus, Q4*)

19. Kuzenkov O., Serdiuk T., Kuznetsova A., Tryputen M., Kuznetsov V., Kuznetsova Y. Mathematical model of dynamics of homomorphic objects. Proceedings of the 1st International Workshop on Information-Communication Technologies & Embedded Systems (ICTES 2019), Mykolaiv, Ukraine, November 14–15, 2019. P. 190–205, ISSN: 1613-0073, URL: <https://ceur-ws.org/Vol-2516/paper15.pdf> (*Scopus, Q4*)

20. Kiseleva O., Kuzenkov O., Yakovlev S. Mathematical Modeling of Rhesus Agglutinin Dynamics in the Human Population. Proceedings of the 3rd International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2023), Waterloo, Canada, November 20–22, 2023. P. 270–275, ISSN: 1613-0073, URL: <https://ceur-ws.org/Vol-3641/short8.pdf> (*Scopus, Q4*)

21. Кузенков О. О., Падалко В. Г. Розробка математичної моделі популяційної динаміки резус-аглютиногену. Питання прикладної математики і математичного моделювання. – Д.: ДНУ, 2017. – Вип. 17. С. 125–133. ISSN: 2074-5893. DOI: 10.15421/321714

22. Кісельова О. М., Кузенков О. О., Падалко В. Г. Математичне моделювання розповсюдження COVID-19 у Дніпропетровській області. Питання прикладної математики і математичного моделювання. – Д.: ДНУ, 2020. – Вип. 20. – С. 65–73. ISSN: 2074-5893, DOI: 10.15421/322006

23. Кісельова О. М., Кузенков О. О., Балейко Н. В. Узагальнені підходи до моделювання розповсюдження COVID-19 на закритих та відкритих ареалах.

Питання прикладної математики і математичного моделювання. – Д.: ДНУ, 2021. – Вип. 21. – С. 105–115, ISSN: 2074-5893, DOI: 10.15421/322111

24. Кісельова О. М., Кузенков О. О. Класифікація (не)лінійних біфуркаційних кривих та гіперповерхонь математичної моделі міжгрупових взаємодій у випадку роздільної ніші. Питання прикладної математики і математичного моделювання. – Д.: ДНУ, 2023. – Вип. 23. – С. 83–93. ISSN: 2074-5893. DOI: 10.15421/322333

25. Kiseleva E. M., Hart L. L., Kuzenkov O. O., Zakutnii D. V. On the implementation of algorithms for solving the simplest dynamic problem of optimal set partitioning. Problems of applied mathematics and mathematical modeling: collection of scientific works. – Dnipro, 2024. – Volume 24. P. 66–75. ISSN: 2074-5893, DOI: 10.15421/322407

26. Kiseleva E. M., Prytomanova O. M., Hart L. L., Zaytseva T. A., Kuzenkov O. O. Application of mathematical methods of artificial intelligence to solve problems of optimal set partitioning. Problems of applied mathematics and mathematical modeling: collection of scientific works. – Dnipro, 2024. – Volume 24. P. 89–98. ISSN: 2074-5893. DOI: 10.15421/322409

### **Список опублікованих праць апробаційного характеру**

27. Kuzenkov O., Kuznetsov V., Tryputen N. Analysis of phase trajectories of the third-order dynamic objects. 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2019, pp. 1235–1243. DOI: 10.1109/UKRCON.2019.8879819 (*Scopus*)

28. Кісельова О. М., Падалко В. Г., Кузенков О. О. Математичне моделювання розповсюдження COVID-19 у Дніпропетровській області. Математичне та програмне забезпечення інтелектуальних систем : тези доповідей XVIII Міжнародної науково-практичної конференції, 18–20 листопада. – Д.: 2020. – С. 129–130.

29. Кісельова О. М., Кузенков О. О. Математичне моделювання періодичних процесів розповсюдження COVID-19 у Дніпропетровській

області. Математичне та програмне забезпечення інтелектуальних систем : тези доповідей XIX Міжнародної науково-практичної конференції, 17–19 листопада. – Д.: 2021. – С. 84.

30. Костенко О. В., Кузенков О. О. Алгоритми розв’язання логістичних задач. Математичне та програмне забезпечення інтелектуальних систем : тези доповідей XX Міжнародної науково-практичної конференції, 23–25 листопада. – Д.: 2022. – С. 110.

31. Kuzenkov O., Lozovskyi A. Algorithms for solving logistic problems under conditions of uncertainty. Mathematical Support and Software for Intelligent Systems (MSSIS-2022): Abstracts of the XX International Scientific and Practical Conference. – November 23–25. – Dnipro: DNU, 2022. – P. 118–119.

32. Kuzenkov O., Busher V., Chorny O., Nikolenko A., Kuznetsov V., Savvin O. Nonlinear Analysis of Bifurcatory Properties of Mathematical Model of Subpopulation Dynamics in the Case of a Single Niche for Subpopulation. 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC), 04–07 October 2022. (Scopus). DOI: 10.1109/SAIC57818.2022.9923003

33. Kiseleva O., Kuzenkov O., Lozovskyi A. The risk management in logistics problems. Mathematical Support and Software for Intelligent Systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference. – November 22–24. – Dnipro: DNU, 2023. – P. 14–15.

34. Kiseleva O., Kuzenkov O., Feschenko M. Algorithms and methods for 3D printing trajectories. Mathematical Support and Software for Intelligent Systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference. – November 22–24. – Dnipro: DNU, 2023. – P. 16–17.

35. Kuzenkov O., Masych M., Siryk S. Design and software implementation of the server part of the “Svitlo” – the alternative notification system. Mathematical Support and Software for Intelligent Systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference. – November 22–24. – Dnipro: DNU, 2023. – P. 25–26.

36. Kuzenkov O., Oleshko O. Design and software implementation of the client part of the “Svitlo” – the alternative notification system. *Mathematical Support and Software for Intelligent Systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference.* – November 22–24.–Dnipro: DNU,2023.– P.27–28.
37. Kostenko O., Honcharova Yu., Kuzenkov O. The main transport problems. Сучасні науково-технічні дослідження у контексті мовного простору (англійською мовою): матеріали II Всеукраїнської науково-практичної конференції молодих науковців та студентів. – Дніпро, 11 травня 2023 р. – Дніпро: Видавець Біла К. О., 2023. – С. 271–273.
38. Kiseleva O. M., Kuzenkov O. O. Use of discrete and continuous algorithms for the construction of an optimal trajectory in 3D printing. *Automatics 2024: Abstracts of the XXVII International Conference of Automatic Control, Dnipro, November 20–22, 2024.* – Dnipro: DNU, 2024. – P. 28.
39. Кісельова О. М., Гарт Л. Л., Кузенков О. О., Закутній Д. В. Про найпростішу динамічну задачу оптимального розбиття множин. *Автоматика 2024: тези XXVII Міжнародної конференції з автоматичного керування, Дніпро, 20–22 листопада 2024 р.* – Дніпро: ДНУ, 2024. – С. 121–122.
40. Kiseleva O. M., Yakovlev S. V., Prytomanova O. M., Kuzenkov O. O. Mathematical modeling of spatial dynamics of infectious diseases. *Automatics 2024: Abstracts of the XXVII International Conference of Automatic Control, Dnipro, November 20–22, 2024.* – Dnipro: DNU, 2024. – P. 30–32.
41. Kozakova N. L., Hryhorenko O., Kuzenkov O. O., Baleyko N. Mathematical modeling of the problems of optimal distribution and planning in conditions of uncertainty. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024 / Under the general editorship of E. M. Kiseleva.* – Dnipro: DNU, 2024. – P. 26.
42. Kozakova N. L., Lupynskyi S., Kuzenkov O. O., Baleyko N. Bifurcation analysis and control in natural process models related to distribution and

allocation. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024 / Under the general editorship of E. M. Kiseleva.* – Dnipro: DNU, 2024. – P. 27.

43. Kozakova N. L., Shvedov V., Kuzenkov O. O., Baleyko N. Algorithmic approaches to solving dynamic problems of optimal placement and distribution. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024 / Under the general editorship of E. M. Kiseleva.* – Dnipro: DNU, 2024.–P.28–29.

44. Kostenko O.V., Kuzenkov O.O. Solution of the facility location problem using the genetic algorithm method. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024 / Under the general editorship of E. M. Kiseleva.* – Dnipro: DNU, 2024. – P. 305–306.

## ЗМІСТ

ВСТУП.....	15
Розділ 1. ОГЛЯД ЛІТЕРАТУРИ. ВИБІР НАПРЯМКУ ДОСЛІДЖЕНЬ.....	23
1.1 Етапи розвитку теорії оптимального розбиття множин.....	23
1.2 Класифікація неперервних задач оптимального розбиття множин $n$ -вимірного евкладового простору $E_n$ на підмножини.....	42
1.3 Задачі оптимального розбиття множин в нечітких постановках....	51
1.4 Біфуркаційні ефекти в динамічних задачах оптимального розбиття множин.....	57
1.5 Висновки до розділу 1.....	61
Розділ 2. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН ІЗ ЗАДАНИМ ПОЛОЖЕННЯМ ЦЕНТРІВ ПІДМНОЖИН ТА МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ.....	64
2.1 Математична постановка однопродуктової динамічної задачі розбиття множин з заданим положенням центрів підмножин.....	68
2.2 Обґрунтування методу розв'язання.....	72
2.3 Алгоритм розв'язання задачі однопродуктової динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин.....	73
2.4 Розв'язання модельних задач та оцінка чисельних результатів.....	74
2.5 Висновки до розділу 2.....	91
Розділ 3. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН З ВІДШУКАННЯМ КООРДИНАТ ЦЕНТРІВ ПІДМНОЖИН ТА МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ..	93
3.1 Математична постановка однопродуктової динамічної задачі оптимального розбиття множин з відшукуванням координат центрів підмножин.....	97
3.2 Обґрунтування методу розв'язання.....	99
3.3 Алгоритм розв'язання однопродуктової динамічної задачі оптимального розбиття множин з розміщенням центрів підмножин.....	101
3.4 Розв'язання модельних задач та оцінка чисельних результатів....	103

3.5 Висновки до розділу 3.....	123
Розділ 4. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН ІЗ ЗАДАНИМ ПОЛОЖЕННЯМ ЦЕНТРІВ ПІДМНОЖИН В УМОВАХ НЕВИЗНАЧЕНОСТІ, МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ ТА БІФУРКАЦІЙНІ ВЛАСТИВОСТІ.....	125
4.1 Математична постановка динамічної задачі оптимального розбиття множин в умовах невизначеності.....	125
4.2 Метод роз'язання динамічної задачі ОРМ з нечітким параметром у цільовому функціоналі.....	129
4.3 Алгоритм роз'язання динамічної задачі ОРМ з нечіткими параметрами у цільовому функціоналі.....	134
4.4 Розв'язання модельних тестових задач.....	135
4.5 Біфуркаційні ефекти в динамічних задачах оптимального розбиття множин та підходи до їх дослідження.....	148
4.6 Висновки до розділу 4.....	153
Розділ 5. РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ВОГНЕВИХ ГРУП З ВИКОРИСТАННЯМ МЕТОДІВ ТЕОРІЇ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН.....	155
5.1 Огляд прикладної проблеми.....	155
5.2 Предметна постановка задачі.....	161
5.3 Математична постановка задачі.....	162
5.4 Результати чисельного експерименту .....	165
5.5 Оцінка ефективності роботи програмного комплексу.....	171
5.6 Висновки до розділу 5.....	173
Розділ 6. РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ПОБУДОВИ ОПТИМАЛЬНИХ ТРАЄКТОРІЙ ДЛЯ ТРИВИМІРНОГО ДРУКУ З ВИКОРИСТАННЯМ МЕТОДІВ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН.....	175
6.1 Огляд прикладної проблеми.....	175
6.2 Предметна постановка задачі.....	186

6.3 Математична постановка задачі.....	187
6.4 Розробка алгоритму побудови оптимальних траєкторії 3D друку.....	190
6.5 Результати чисельного експерименту та опис програмного додатку.....	195
6.6 Висновки до розділу 6.....	204
Розділ 7. РОЗВ’ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ РОЗБИТТЯ ТЕРИТОРІЇ НА ЗОНИ ЗАБЕЗПЕЧЕННЯ МЕДИЧНИМИ ЗАКЛАДАМИ ПРИ СПАЛАХАХ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ.....	206
7.1 Огляд прикладної проблеми.....	206
7.2 Предметна постановка задачі.....	215
7.3 Математична постановка задачі.....	220
7.4 Результати чисельного експерименту.....	226
7.5 Висновки до розділу 7.....	233
ВИСНОВКИ.....	236
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	240
Додаток А. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОЗВ’ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН.....	262
Додаток Б. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛЬТЕРНАТИВНОЇ СИСТЕМИ ОПОВІЩЕННЯ “СВІТЛО”.....	279
Б.1. Компонент відображення мапи з кореспондентами.....	279
Б.2. Компонент відображення мапи з маркерами та траєкторіями.....	283
Б.3. Обробка запитів.....	288
Б.4. Перевірка авторизації користувача.....	293
Б.5. Структура даних маркерів.....	293
Б.6. Структура профілю користувача.....	295
Б.7. Загальний модуль програми.....	296
Додаток В. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОБУДОВИ ОПТИМАЛЬНОЇ ТРАЄКТОРІЇ ДЛЯ ТРИВИМІРНОГО ДРУКУ.....	298

## ВСТУП

Дисертаційна робота є подальшим розвитком теорії оптимального розбиття множин (ОРМ), що є новим розділом нескінченновимірного математичного програмування з булевими змінними. Для динамічних задач оптимального розбиття, застосовуються сучасні чисельні методи, субградієнтні методи з розтягом простору та інші методи негладкої оптимізації. Для отримання розв'язку систем диференціальних рівнянь використовуються аналітичні та чисельні методи, а для якісного дослідження таких систем якісна теорія диференціальних рівнянь.

**Актуальність теми.** Широкий перелік практично важливих задач різного типу та спрямованості, різних галузей та складності можна звести до задач розбиття множин на підмножини з мінімізацією/максимізацією певного критерію або оптимізацією значення цільового функціоналу. До таких задач можна віднести задачі класифікації, кластеризації, розміщення, маршрутизації, розбиття, покриття, розподілу ресурсів, оптимального керування тощо. Умовно такі задачі, залежно від прикладного змісту, можна поділити на неперервні (для континуальної множини, що підлягає розбиттю) та дискретні.

До найпоширеніших задач, що класифікуються як неперервні, можна віднести нескінченновимірні транспортні задачі та нескінченновимірні задачі розміщення підприємств. У таких задачах вважається, що область, яка підлягає розбиттю, неперервно заповнена споживачами. Основною метою розв'язання таких задач є розбиття множини на підмножини, що обслуговуватимуться центрами забезпечення/виробництва для мінімізації витрат на транспортування продукції від центру постачання до споживача.

Широкий перелік задач у різних постановках обумовлений змістовним спектром варіацій таких задач. Зокрема, задачі в постановці можуть відрізнятися складністю (лінійні, нелінійні), кількістю продуктів (однопродуктові, багатопродуктові), стохастичністю (стохастичні, детерміновані), статичністю (статичні, динамічні), розміщенням центрів (з

фіксованими центрами та з їх відшукуванням) та іншими аспектами. Таким чином, задачі оптимального розбиття множин у своїй математичній постановці можуть по-різному комбінувати названі характеристики, змінюючи не тільки математичну постановку задачі, але й застосовні методи, що суттєво залежать від обмежень задачі та прикладного застосування таких постановок. Слід зазначити, що результати застосування сучасних методів розв'язання таких задач показують високу адекватність та є інтуїтивно зрозумілими, проте, на відміну від евристичних рішень, є більш чисельно конкретизованими, що дозволяє отримати рішення з більш оптимальним значенням цільового функціоналу.

Одним із вагомих та малодосліджених у наш час напрямів розвитку теорії оптимального розбиття множин є відповідні задачі в динамічних постановках. Переважна більшість задач, що розглядаються та досліджуються, містять статичні параметри. Проте очевидно, що, принаймні, такі фактори, як вартість перевезення та потужність виробництв, можуть змінюватися з плином часу або залежати, наприклад, від типу продукції, що виготовляється, рівня інфляції, вартості паливно-мастильних матеріалів тощо. Намагання власників виробництв розширити спектр товарів, збільшити обсяги виробництва або амортизувати транспортні засоби для зменшення середньої вартості перевезення одиниці продукції є рушійними для розвитку бізнесу.

Серед вітчизняних та закордонних науковців існує ряд опублікованих робіт з тематики динамічних задач теорії оптимального розбиття множин. Проте в цих роботах під динамічними аспектами, здебільшого, розуміється не стільки динамічність властивостей параметрів моделі, скільки застосування управління до системи, в якій можливі динамічні прояви. Саме тому актуальними у наш час залишаються динамічні задачі оптимального розбиття множин, де під динамічністю розуміється мінливість у часі ключових параметрів системи та залежність оптимального розбиття від фазової траєкторії заданої динамічної системи.

**Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційні дослідження проводилися відповідно до планів наукових досліджень науково-дослідної лабораторії оптимізації складних систем кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара у межах держбюджетних тем: «Математичні моделі і методи оптимізації складних систем на основі нелінійної теорії оптимального розбиття множин» (№ держреєстрації 0112U000193, 2012–2014 рр.), «Математичні моделі та алгоритми розв'язання неперервних задач покриття на основі теорії оптимального розбиття множин» (№ держреєстрації 0115U002392, 2015–2017 рр.), «Розробка математичних моделей та алгоритмів розв'язання прикладних задач класифікації, кластеризації на основі теорії оптимального розбиття множин» (№ держреєстрації 0119U100600, 2019–2021 рр.), а також у рамках наукових тем «Математичні моделі, методи та алгоритми аналізу складних систем» (№ держреєстрації 0113U003558, 2013–2015 рр.), «Теоретичні основи математичних моделей та методів дослідження складних систем» (№ держреєстрації 0116U002214, 2016–2018 рр.), «Математичні моделі, методи та алгоритми розв'язання задач аналізу складних систем» (№ держреєстрації 0119U101302, 2019–2021 рр.) при кафедрі обчислювальної математики та математичної кібернетики відповідно до тематичних планів науково-дослідних робіт Дніпровського національного університету імені Олеся Гончара.

**Мета і завдання дослідження.** *Мета роботи* полягає у створенні теоретичної методології, зокрема, методів і алгоритмів розв'язання динамічних задач оптимального розбиття множин як у класичних, так і в ускладнених постановках. Це включає побудову нових ефективних алгоритмів та методів на основі обґрунтованих підходів та їх числову реалізацію.

Для досягнення означеної мети були поставлені такі *основні завдання дослідження*:

- формалізувати нову динамічну задачу оптимального розбиття множин з фіксованими центрами підмножин;
- розробити новий клас математичних моделей динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин;
- розробити метод та алгоритм розв'язання нового класу математичних моделей динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин;
- формалізувати нову динамічну задачу оптимального розбиття множин з відшукуванням координат центрів підмножин;
- розробити новий клас математичних моделей динамічної задачі оптимального розбиття множин з відшукуванням координат центрів підмножин;
- розробити метод та алгоритм розв'язання нового класу математичних моделей динамічної задачі оптимального розбиття множин з відшукуванням координат центрів підмножин;
- формалізувати нову динамічну задачу оптимального розбиття множин з фіксованими центрами підмножин в умовах невизначеності;
- розробити новий клас математичних моделей динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин в умовах невизначеності;
- розробити метод та алгоритм розв'язання нового класу математичних моделей динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин в умовах невизначеності;
- розробити програмне забезпечення для розв'язання нових класів динамічних задач оптимального розбиття множин з фіксованими центрами та відшукуванням їх координат в тому числі в умовах невизначеності;
- здійснити імплементацію нового класу математичних моделей динамічних задач оптимального розбиття множин в прикладних задачах охорони здоров'я, планування регіональної інфраструктури та використання регіональних військових потужностей;

- з використанням сучасних чисельних методів реалізувати чисельний експеримент для підтвердження адекватності нових математичних моделей динамічних задач оптимального розбиття множин.

*Об'єкт дослідження* — процес математичного моделювання складноорганізованих технічних систем.

*Предмет дослідження* — методи та алгоритми розв'язання динамічних задач оптимального розбиття множин та їх застосування в конкретних прикладних задачах.

*Методи дослідження* включають підходи з теорії неперервних задач оптимального розбиття, якісної теорії диференціальних рівнянь, теорії неklasичних задач оптимізації, теорії двоїстості та методи недиференційованої оптимізації, зокрема  $\gamma$ -алгоритм Н.З.Шора.

**Наукова новизна одержаних результатів** полягає в наступному:

- вперше узагальнено класифікацію динамічних задач оптимального розбиття множин залежно від типів параметрів, що дозволило враховувати їх динамічні зміни;
- вперше запропоновано методологію розв'язання неперервних динамічних задач оптимального розбиття множин на основі інтеграції методів теорії оптимального розбиття множин, якісної теорії диференціальних рівнянь і методів негладкої оптимізації;
- вперше розроблено методи розв'язання динамічних задач оптимального розбиття з відшукуванням центрів підмножин із застосуванням  $\gamma$ -алгоритму Шора, що дозволило враховувати негладкість цільової функції;
- вперше розроблено метод нейро-нечіткої ідентифікації параметрів у динамічних задачах оптимального розбиття множин в умовах невизначеності;
- набули подальшого розвитку математичні моделі задач оптимального розбиття множин, що дозволило враховувати еволюцію параметрів системи;
- набули подальшого розвитку методи розв'язання динамічних задач оптимального розбиття множин шляхом побудови відповідних чисельних

схем з урахуванням аналізу якісних властивостей систем диференціальних рівнянь;

- удосконалено моделі динамічних задач оптимального розбиття множин в умовах невизначеності із застосуванням теорії оптимального розбиття множин та нечіткої логіки;

- удосконалено підходи до моделювання оптимізаційних задач динамічного розбиття множин із формулюванням нових постановок прикладних задач за умов параметричної невизначеності.

**Практичне значення одержаних результатів** дисертаційної роботи полягає в розробці нових ефективних методів і алгоритмів для розв’язання динамічних задач оптимального розбиття множин, які мають теоретичне обґрунтування та програмну реалізацію. Створені програмні комплекси застосовані для розв’язання прикладних оптимізаційних проблем, що у формалізованому вигляді зводяться до неперервних задач оптимального розбиття множин. Серед таких задач:

- зонування територій для забезпечення доступу до медичних установ в умовах поширення інфекцій, що підтверджується довідкою про впровадження результатів дисертаційного дослідження на рівні Дніпропетровської обласної державної адміністрації та Дніпропетровської обласної ради;

- побудова оптимальних траєкторій у процесах тривимірного друку, що підтверджується довідкою про впровадження результатів дисертаційного дослідження на ТОВ “Скайрора” (Skygora UA);

- оцінка ефективності розміщення центрів гірничого обладнання до місць видобутку корисних копалин, що підтверджується актом про впровадження науково-прикладних розробок придніпровського наукового центру НАН України та МОН України;

- динамічне розміщення вогневих груп, що підтверджується пам’ятним нагрудним знаком “За сприяння Повітряним Силам Збройних Сил України”.

Результати дисертаційної роботи впроваджено в навчальний процес Дніпровського національного університету імені Олеся Гончара під час викладання автором дисциплін «Математичні основи інформаційних технологій» та «Основи системного аналізу», а також у процесі керівництва курсовими й дипломними роботами студентів факультету прикладної математики та інформаційних технологій, які навчалися за спеціальностями F1 (113) – прикладна математика та F4 (124) – системний аналіз.

**Особистий внесок здобувача.** Усі основні наукові результати, викладені в дисертаційній роботі, отримані автором особисто, зокрема й визначення загального плану досліджень. У працях, що опубліковані у співавторстві, особистий внесок здобувача полягає в такому: [1] – постановка чисельного експерименту, якісний аналіз математичної моделі; [2], [4] – побудова математичної моделі, якісний аналіз математичної моделі, постановка чисельного експерименту; [3], [5], [6] – побудова математичної моделі, постановка чисельного експерименту, ретроспективний аналіз отриманих результатів; [7], [13] – побудова математичної моделі, якісний аналіз математичної моделі, постановка чисельного експерименту, прикладна інтерпретація отриманих результатів; [8], [9], [10] – побудова математичної моделі, якісний аналіз математичної моделі, постановка чисельного експерименту; [14] – постановка задачі, побудова математичної моделі; [15], [19] – постановка задачі, побудова математичної моделі, постановка чисельного експерименту; [16] – побудова математичної моделі, формалізація методу та алгоритму дослідження; [17], [18], [20] – формалізація задачі, побудова математичної моделі, розробка алгоритму, інтерпретація отриманих результатів.

**Апробація результатів дисертації.** Основні ідеї, принципи, положення і результати дисертаційного дослідження доповідалися й обговорювалися на наукових конференціях та семінарах різного рівня, серед яких: II Всеукраїнська конференція з електроніки та комп'ютерної інженерії (UKRCON 2019); XVIII, XIX, XX, XXI, XXII Міжнародні науково-практичні

конференції “Математичне та програмне забезпечення інтелектуальних систем” (МПЗІС 2020–2024); III Міжнародна конференція з системного аналізу та інтелектуальних обчислень (SAIC 2022); Міжнародна конференція smart-технологій в міському інжинірингу (STUE 2023); II Всеукраїнська науково-практична конференція молодих науковців та студентів “Сучасні науково-технічні дослідження у контексті мовного простору” (Дніпро, 2023); XXVII Міжнародна конференція з автоматичного керування «Автоматика 2024» (Дніпро, 2024); науковий семінар “Теорія оптимальних рішень”.

**Публікації.** Основні наукові результати дисертаційної роботи повною мірою викладено в 44 наукових публікаціях, з них 26 статей [1–26] та 18 тез доповідей [27–44] у збірниках матеріалів міжнародних наукових конференцій. Серед 26 статей 13 статей [5–17] – у наукових виданнях, включених до Переліку наукових фахових видань України з відповідної дисертаційній роботі наукової галузі (відповідних спеціальностей), 7 статей [1–4, 18–20] – у наукових періодичних виданнях інших держав із напрямку, з якого підготовлено дисертацію, включених до наукометричної бази Scopus, серед яких 2 [1, 2] наукові публікації у виданнях, віднесених до першого квартиля (Q1) відповідно до класифікації SCImago Journal and Country Rank або Journal Citation Reports та 6 [21–26] інших статей за темою дисертаційної роботи.

**Структура та обсяг дисертації.** Дисертаційна робота складається зі вступу, семи розділів, висновків, списку використаних джерел, що містить 200 найменувань на 22 сторінках та додатків на 40 сторінках. Загальний обсяг дисертації – 302 сторінки, обсяг основного тексту – 229 сторінок. Робота містить 53 рисунки та 24 таблиці.

Автор висловлює щирю вдячність члену-кореспонденту НАН України, д.ф.-м.н. наук, професору Кісельовій Олені Михайлівні за консультації та цінні поради під час проведення наукових досліджень.

## **Розділ 1. ОГЛЯД ЛІТЕРАТУРИ. ВИБІР НАПРЯМКУ ДОСЛІДЖЕНЬ**

### **1.1 Етапи розвитку теорії оптимального розбиття множин**

Історія теорії оптимального розбиття множин (ОРМ), яка є новим розділом нескінченновимірного математичного програмування, налічує вже більше 80-ти років. Результати теорії знаходять своє застосування у розв'язку широкого переліку теоретичних та практичних оптимізаційних задач, які можуть бути зведені до неперервних моделей оптимального розбиття множин  $n$ -вимірного евклідового простору та розв'язані відомими методами та алгоритмами, ефективність яких не викликає сумнівів адже підтверджена чисельними, в тому числі натурними експериментами, на практиці та часом.

Зацікавлення міжнародної наукової спільноти неперервними моделями оптимального розбиття множин зумовлене тим, що такі задачі охоплюють широкий спектр теоретичних та прикладних задач оптимізації у математичному формулюванні. Серед класичних прикладів таких задач – нескінченновимірні транспортні задачі, а також узагальнені варіанти задач розміщення підприємств. У цих задачах розглядається регіон, що суцільно заповнений споживачами, який потрібно поділити на зони обслуговування так, щоб кожна з них забезпечувало окреме підприємство. Основна мета таких задач полягає в мінімізації витрат на транспортування та виробництво. Під поняттям "споживачі" можуть розумітися, зокрема, телефонні абоненти, учні, виборці чи пацієнти, які потребують діагностики та/або лікування.

Потреба у використанні нескінченновимірних моделей розміщення виникає у випадках, коли кількість споживачів надзвичайно велика. Це характерно, наприклад, для задач, пов'язаних із телекомунікаціями, шкільним чи виборчим розподілом, де дискретні підходи є надто громіздкими через високу розмірність задачі.

Окрім того, існують задачі, в яких множина, що підлягає розбиттю, із самого початку має неперервну (континуальну) структуру, що також призводить до моделей оптимального розбиття.

Одним з перших кроків становлення теорії ОРМ стала нескінченновимірна транспортна задача аналог якої був розглянутий Л.В. Канторовичем ще у 1942-му році. Спрощений варіант такої моделі нескінченновимірної задачі розміщення виробництва в іноземній літературі названа задачею location-allocation.

У наш час таку задачу, здебільшого можна бачити у кількох різних постановках, а саме:

### **Постановка задачі з фіксованими центрами.**

Нехай споживач деякої однорідної продукції рівномірно розподілений в області  $\Omega \subset E_2$ . Скінченне число  $N$  виробників цієї продукції розташовані в ізольованих точках  $\tau_i = (\tau_i^1, \tau_i^2), i = 1, \dots, N$ , області  $\Omega$ . Вважаються невідомими:  $\rho(x)$  – попит на продукцію споживача з координатами  $x = (x^1, x^2)$ ;  $c(x, \tau_i)$  – вартість транспортування одиниці продукції від виробника  $\tau_i = (\tau_i^1, \tau_i^2), i = 1, \dots, N$  споживачеві з координатами  $x = (x^1, x^2)$ . Передбачається, що прибуток виробника залежить тільки від його витрат, які є сумою виробничих і транспортних витрат.

Для кожного  $i$ -го виробника задана функція  $\varphi_i(Y_i)$ , що описує залежність вартості виробництва від його потужності  $Y_i$ , яка визначається за формулою  $Y_i = \int_{\Omega_i} \rho(x) dx$ , і наведені капітальні витрати на реконструкцію  $i$ -го виробництва для збільшення його потужності від існуючої до проектної  $Y_i$ .

Множину споживачів  $\Omega$  потрібно розбити на зони обслуговування  $\Omega_i$  споживачів  $i$ -м виробником так, щоб

$$\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = 1, \dots, N. \quad (1.1)$$

При цьому потужність  $i$ -го виробника визначається сумарним попитом споживачів, що належать  $\Omega_i$ , і не повинна перевищувати заданих обсягів:

$$\int_{\Omega_i} \rho(x) dx \leq b_i, i = 1, \dots, N. \quad (1.2)$$

Не виключається, що деякі з підмножин  $\Omega_i$  виявляться порожніми.

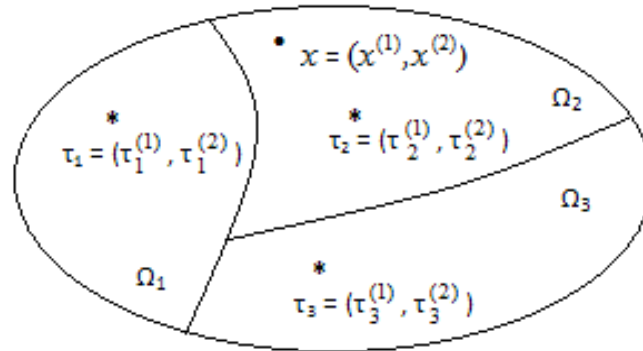


Рисунок 1.1. Схематичний приклад розбиття множини  $\Omega$  на три підмножини  $\Omega_1, \Omega_2, \Omega_3$  з центрами  $\tau_1, \tau_2, \tau_3$ .

На рис. 1.1 зображено приклад розбиття заданої множини  $\Omega \subset E_2$  на три підмножини  $\Omega_1, \Omega_2, \Omega_3$  з центрами  $\tau_1, \tau_2, \tau_3$  цих підмножин відповідно.

### Постановка задачі з відшуканням координат центрів.

Потрібно розбити множину споживачів  $\Omega$  на зони обслуговування їх  $N$  виробниками, тобто на підмножини  $\Omega_i, i = 1, \dots, N$ , і розмістити цих виробників в  $\Omega$  так, щоб мінімізувати функціонал сумарних витрат на виробництво продукції і доставку її до споживача:

$$F(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\}) = \sum_{i=1}^N \left\{ \int_{\Omega_i} c(x, \tau_i) \rho(x) dx + \varphi_i \left( \int_{\Omega_i} \rho(x) dx \right) \right\} \quad (1.3)$$

за додаткових умов (1.1), (1.2).

Задача (1.1)-(1.3) є нескінченновимірною задачею розміщення. У більшості практичних задач вартість виробництва продукції на промисловому підприємстві потужності  $Y_i$  дорівнює добутку собівартості цієї продукції на її кількість. В силу цього маємо

$$\varphi_i(Y_i) = d_i + a_i Y_i, i = 1, \dots, N. \quad (1.4)$$

Підставляючи вираз (1.4) в (1.3) отримаємо

$$F(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\}) = \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx \quad (1.5)$$

Задача (1.1), (1.5) при  $a_i = 0, i = 1, \dots, N$ , є нескінченновимірною транспортною задачею.

Професоркою О.М.Кісельовою були отримані та опубліковані в роботі [21] необхідні умови оптимальності для задачі (1.1), (1.5), що сформульовані у вигляді наступної теореми.

**Теорема.** Якщо можливе розбиття  $(\Omega_{*1}, \dots, \Omega_{*i}, \dots, \Omega_{*N})$  є оптимальним для задачі (1.1), (1.5), то

$$c(x, \tau_i) + a_i \leq c(x, \tau_k) + a_k \text{ майже всюди (м.в.) для } x \in \Omega_{*i}, i, k = 1, \dots, N. \quad (1.6)$$

Ця теорема має важливий наслідок, а саме:

**Наслідок.** В точках  $x$ , які належать оптимальній границі між підмножинами  $\Omega_{*i}$  і  $\Omega_{*k}$ , в нерівності (1.6) досягається знак рівності.

Зазначимо, що на відміну від вітчизняних увага іноземних авторів, здебільшого, зосереджена на прикладних аспектах застосування теорії ОРМ, як наприклад задачі розміщення-розбиття (location-allocation).

Серед робіт в яких викладено результати таких прикладних досліджень можна назвати роботи Murat A., Verter V., Laporte G. A. [161]. Проте українськими науковцями, як правило, робиться акцент на більш фундаментальні дослідження та застосування методів вирішення таких задач, зокрема в теоретичних аспектах теорії ОРМ. Так в роботах Кісельової О.М. [33], [112], Притоманової О.М. [16], Гарт Л.Л., Журавля С.В., Шаравари В.В. [121]-[122], висвітлено напрями, методи та алгоритми розв'язання задач location-allocation [32] з використанням теорії оптимального розбиття множин для теоретичних застосувань, що безперечно дає більш широкий перелік результатів та може бути використаний у більш широкому колі прикладних задач. У названих вище роботах, автори успішно зводять задачі до класичних скінченновимірних та використовують перевірені методи

отримання аналітичних результатів яких, зокрема, підтверджені результатами чисельного експерименту.

В цілому фундатором теорії оптимального розбиття множин, як окремого розділу теорії нескінченновимірного математичного програмування, є професорка О.М.Кісельової. В рамках наукової школи професоркою О.М.Кісельовою та її учнями, запропоновані оригінальні підходи до побудови оптимальних розбиттів у динамічних системах. У роботах [20]-[21] наведено формалізовані постановки задач та їх аналітичні розв'язки. Розглянуті модельні задачі в чітких постановках, до яких наведено алгоритми їх розв'язання та результати багаточисельних комп'ютерних експериментів, які наочно ілюструють та підтверджують отримані авторами аналітичні результати. У монографіях [11]-[13] здійснено аналіз глобальних постановок задач теорії оптимального розбиття множин в яких сформульовано та доведено ряд фундаментальних теорем, які, зокрема, дозволяють перейти від нескінченновимірних задач до скінченновимірних, перейти від задач з обмеженнями до задач без обмежень тощо.

У роботі [11] розглядаються динамічні задачі, особливості їх постановки, алгоритми розв'язання та наведені чисельні результати комп'ютерного експерименту. В роботі розглядається застосування означених методів для прикладних задач, обмеження які при цьому виникають та методи розв'язання задач якщо аналітичний розв'язок, з тих чи інших причин отримати неможливо.

В роботі [12] детально описаний математичний апарат, що застосовується для дослідження і розв'язання неперервних задач оптимального розбиття множин. Особливу увагу авторами приділено сучасним методам недиференційованої оптимізації. Наведено чисельні приклади для розв'язання названих задач з використанням  $r$ -алгоритму Н.З.Шора та його модифікацій. В роботі розглядається широкий перелік неперервних задач розбиття та ряду споріднених до них задач. Особливий акцент авторами зроблено на питанні зведення нескінченновимірних

оптимізаційних задач до задачі оптимізації негладкої функції скінченного числа змінних.

Особливості розв'язання неперервно-дискретних задач, які обумовлені додатковими умовами розглянуті Кісельовою О.М. та її учнями в роботах [29]-[30], [32], [16], [33], [19]. Дискретність задач обумовлено переліком обмежень, як на постановку так і на алгоритми розв'язання таких задач. В роботах показано підходи, як узагальнені так і такі, що мають відношення до конкретних задач. Використовуючи означені підходи авторам вдається отримати чисельні, а у більшості випадків і аналітичні розв'язки задач.

Серед основних праць закордонних авторів, що містять фундаментальні результати теорії оптимального розбиття множин слід виділити роботи L. Briani, G. Buttazzo, F. Prinari [71], F. Maiale [74], D.Maso [73], G.Bourdin, D.Bucur [68], E.Oudet [69]; F.Santambrogio, G.Carlier, C.Jimenez [77], G.De Philippis, A.Mészáros, B.Velichkov [86].

Так в роботі Buttazzo G., Dal Maso G. [73] розглянуто задачі оптимізації форм множин. У наведених задачах функціонал залежить від периметра досліджуваної множини та відповідної міри Лебега. Безрозмірні функціонали витрат представлені у формалізованому вигляді, а область допустимих рішень складається з відкритих двовимірних множин, які задовольняють топологічним обмеженням у вигляді заданої кількості обмежених зв'язних компонент доповнення. Для коректності задачі, застосовується процедура релаксації. Для некоректних задач інфімум наводиться у явному аналітичному вигляді.

В роботі Briani L., Buttazzo G., Prinari F. [71] розглядається задачі оптимізації форм для загальних інтегральних функціоналів варіаційного числення, які можуть містити граничний доданок. Зокрема, цей клас охоплює задачі оптимізації, що описуються еліптичними рівняннями з умовою Робіна на вільній межі. Авторами доведено існування оптимальної області за досить загальних припущень і досліджені випадки, коли оптимальні області є відкритими множинами та мають скінченний периметр.

В роботі Cheramin M. [80] наведено деякі прикладні задачі в яких використовується  $G$ -збіжність та  $\Gamma$ -збіжність для дослідження асимптотичних граничних випадків задач оптимального керування. Тобто маючи послідовність задач оптимального керування  $P_h$  та задачу керування  $P_\infty$  авторами визначено загальні умови, що включають  $G$ -збіжність та  $\Gamma$ -збіжність, за яких послідовність оптимальних пар задач  $P_h$  збігається до оптимальної пари задач  $P_\infty$ .

В роботі Bourdin V., Vucur D. [68] запропоновано новий чисельний метод для апроксимації розбиттів області, що мінімізують суму власних значень оператора Діріхле–Лапласа довільного порядку. Доведено еквівалентність початкової задачі та її релаксованого формулювання на основі мір. Використовуючи цей результат, авторам вдалось побудувати чисельний алгоритм для апроксимації оптимальних конфігурацій. Також в роботі наведено результати чисельних експериментів, що ілюструють асимптотичну поведінку оптимальних розбиттів у випадку великого розміру сітки.

В монографії Drezner Z., Namacher H.W. [89] описані фундаментальні теоретичні аспекти теорії оптимального транспортування, як варіаційної задачі, її застосування для моделювання різноманітних явищ і зв'язок з рівняннями в частинних похідних. В монографії наводяться основні інструменти, необхідні для розуміння сучасних досліджень в галузі оптимального транспортування, а також методи, які є найкориснішими для його практичних застосувань. Вичерпні доведення використовуються для ілюстрації математичних понять, а кінцеві розділи містять прикладні аспекти теорії оптимального транспорту в таких сферах, як економіка, фінанси, ігри, обробка зображень і динаміка рідин. Крім цього в роботі наведені такі теоретичні аспекти, як транспортування Кнота, властивості функціоналів на мірах, потік Дакорогна-Мозера, формулювання через мінімальні потоки з заданою дивергенцією, випадок супремальної вартості та застосування для вирішення названих задач класичних чисельних методів.

В роботі Carlier G., Jimenez C., Santambrogio F. [77] розглядається розширена постановка задачі Монжа–Канторовича. На відміну від класичної постановки в якій транспортні витрати залежать лише від кількості товарів, і не враховують маршрути, якими ці товари перевозяться. В означеній роботі зроблена вдала спроба врахувати означені фактори задачі. Використовуючи поняття інтенсивності трафіку авторами запропонований варіант задачі, який враховує перевантаження. Цей варіант є неперервною версією добре відомої транспортної задачі на мережах, яка вивчається як в економіці, так і в операційному дослідженні. Інтерес до цієї задачі полягає в її зв'язку з рівновагами трафіку типу Вардропа. В роботі доведено існування рівноважних станів у неперервному просторі та подано їх варіаційне формулювання.

В роботі De Philippis G., Mészáros A. R., Santambrogio F., Velichkov B. [86] досліджено  $BV$ -регулярність розв'язків деяких варіаційних задач в теорії оптимального транспортування. Доведено, що проєкція за Васерштейном міри з щільністю, яка має обмежену варіацію ( $BV$ ), на множину мір з щільністю, обмеженою заданою функцією  $f \in BV$ , також є функцією обмеженої варіації. До того ж, в роботі наведено точну оцінку  $BV$ -норми. Особливий акцент в роботі зроблено для випадку  $f = 1$ , що відповідає проєкції на множину щільностей з обмеженням у просторі  $L^\infty$ . У цьому випадку авторами доведено, що повна варіація зменшується внаслідок проєкції. Ця оцінка, зокрема її ітерації, має прикладне застосування до деяких еволюційних рівнянь у частинних похідних (PDE), наприклад, таких, що описують рух натовпу. Як застосування результатів роботи [86], отримано  $BV$ -оцінки для розв'язків деяких нелінійних параболічних PDE засобами методів оптимального транспортування. Також авторами встановлено кілька властивостей проєкції за Васерштейном, які є цікавими самі по собі, зокрема, вони дозволяють довести єдиність такої проєкції в доволі загальних умовах.

Наряду з більш прикладним напрямом використання результатів теорії ОРМ серед зарубіжних авторів, вітчизнянні науковці приділяють особливу увагу теоретичним результатам, що можуть бути використані, як для отримання якісних результатів та розуміння загальнодинамічних тенденцій притаманних системі, що досліджується, так і прикладним аспектам таких теоретичних результатів.

Так в роботі Кісельової О.М. [7] розглянуто задачі оптимального розбиття з використанням методів та алгоритмів варіаційного числення та теорії міри. Описано, зокрема, загальні умови існування оптимальних розбиттів у задачах мінімізації енергії функціоналу з урахуванням обмежень на обсяг, міру чи геометрію компонент.

Попри те, що теорія оптимального розбиття множин має в своїй основі глибокі аналітичні основи, існує велика кількість праць, автори яких фокусують свою увагу саме на чисельному розв'язанні задач. Зокрема, в роботах [10]-[11] автори використовують для розв'язання методи рівневих множин, топологічну оптимізацію та методи релаксації та регуляризації.

Підкреслимо, що неперервні задачі розбиття природно вкладаються в рамки нескінченновимірного математичного програмування завдяки: варіаційним формулюванням функціоналів цілі; нескінченновимірним обмеженням (цілісність по мірі, граничні умови тощо); використанню методів субдиференціального аналізу та узагальненої спряженості [12].

Серед сучасних напрямів та відкритих проблем дослідження теорії оптимального розбиття множин слід виділити, зокрема: оптимальні розбиття у випадку нечіткої геометрії або стохастичних обмежень; багатокритеріальні розбиття у задачах екології, біомедицини, кібербезпеки; машинне навчання та оптимізація функціоналів на просторах ймовірнісних мір [8], [13].

Теорія, створена професоркою О.М. Кісельовою [8], [22] та реалізована в рамках її наукової школи, ґрунтується на єдиному підході, що полягає у зведенні нескінченновимірних задач (наприклад, через функціонал Лагранжа) до негладких, як правило, скінченновимірних задач оптимізації, для

чисельного розв'язання яких застосовуються сучасні ефективні методи недиференційовної оптимізації, різні варіанти  $r$ - алгоритму, розроблені [40], [41], [43] в Інституті кібернетики ім. В.М. Глушкова НАН України під керівництвом Н.З. Шора [47], [48], [49].

Одним з етапів стрімкого розвитку теорії ОРМ стали 70-ті роки ХХ-го сторіччя. В означений період теорія ОРМ набула свого стрімкого розвитку, що було зумовлено успішною реалізацією ідеї зведення вихідних нескінченновимірних задач оптимізації типу (1.1)-(1.3) та більш складених її постановок (наприклад, через функціонал Лагранжа за допомогою теорії двоїстості) до скінченновимірних негладких задач максимізації або негладких задач максиміна. Для чисельного розв'язання цих скінченновимірних задач в той час використовувались методи згладжування, які на той час були достатньо робасними.

У 80-их роках ХХ-го сторіччя Н.З. Шор запропонував для чисельного розв'язання негладких скінченновимірних задач застосовувати методи недиференційовної оптимізації – різні варіанти  $r$ -алгоритму, розроблені ним і його учнями в Інституті кібернетики ім. В.М. Глушкова НАН України.

В теорії неперервних задач ОРМ до теперішнього часу сформувався змістовний клас задач, обумовлених як різними типами математичних постановок задач розбиття, так і різними сферами її застосувань, а саме:

- детерміновані, стохастичні, лінійні, нелінійні, однопродуктові, багатодуктові, статичні, динамічні задачі ОРМ при обмеженнях та без обмежень, як із заданим положенням центрів підмножин, так і з відшукуванням оптимального варіанту їх розташування, [6]-[7], [9]-[11];

- задачі оптимального розбиття множин в умовах невизначеності, для зняття невизначеності в яких пропонується застосовувати або математичний апарат стохастичного нескінченновимірного математичного програмування (якщо частина інформації має ймовірнісний характер), або апарати нечітких множин і нечіткої логіки (якщо параметри, що входять до опису моделей, є нечіткими, неточними, недостовірними тощо), [29]-[31],

[14];

– динамічні задачі оптимального розбиття з критерієм оптимальності, що залежить від фазових траєкторій і управління деякої заданої керованої системи, [11], [23], [110], [113]-[114];

– неперервні задачі про кульове покриття, що зводяться до задач ОРМ.

Структуру сформованої до теперішнього часу теорії оптимального розбиття множин можна представити у вигляді блок-схеми наведеної на рис.1.2.

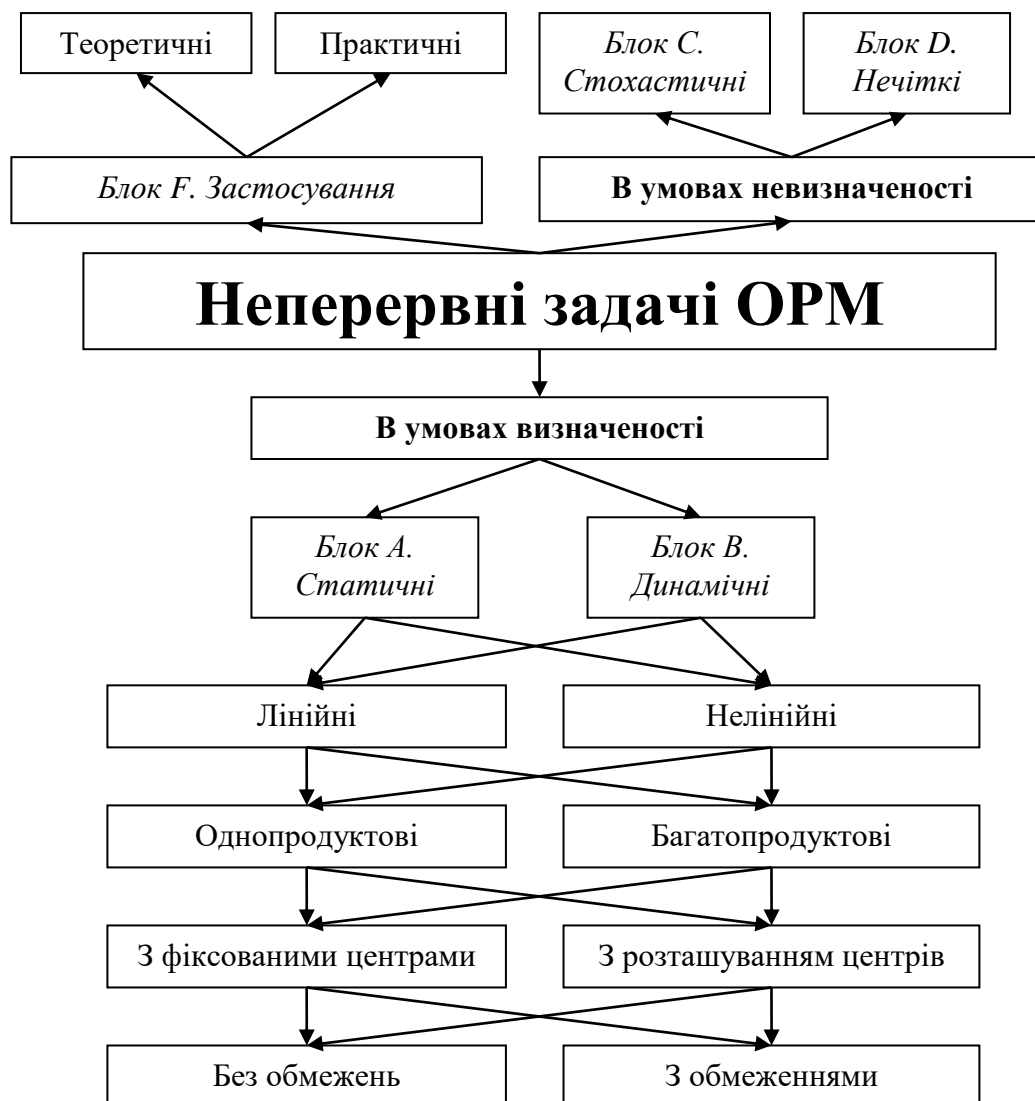


Рисунок 1.2. Схематичне зображення структури теорії оптимального розбиття множин

Найбільш вивченими на сьогоднішній день є статичні задачі

оптимального розбиття множин (Блок А).

У блоці В розглядаються динамічні задачі ОРМ з критерієм оптимальності, що залежать від фазових траєкторій та управління деякої заданої керованої системи. Так, наприклад, у задачі (1.1)-(1.2), (1.5), яка розглядалась як нескінченновимірний аналог транспортної задачі, функції попиту і вартості транспортування одиниці продукції можуть змінюватися протягом деякого проміжку часу в силу дії певних економічних факторів і є невідомими заздалегідь фазовими траєкторіями заданого керованого процесу, який описується деякою системою диференціальних рівнянь з початковими умовами.

Для розв'язання наведених у блоках А і В класів задач ОРМ запропонований, як уже зазначалося вище, єдиний підхід, в основі якого лежить наступна ідея: вихідні задачі ОРМ, що математично сформульовані як нескінченновимірні задачі оптимізації, зводяться певним чином (наприклад, через функціонал Лагранжа) до допоміжних скінченновимірних негладких задач максимізації або негладких задач максиміна, для чисельного розв'язання яких застосовуються сучасні ефективні методи недиференційовної оптимізації, зокрема, різні модифікації  $r$ -алгоритму Н.З. Шора, [20], [47]-[49].

Особливістю такого підходу для лінійних задач ОРМ є той факт, що розв'язання вихідних нескінченновимірних задач оптимізації вдається отримати аналітично в явному вигляді, причому в аналітичний вираз можуть входити параметри, що відшукуються як оптимальний розв'язок вищеназваних допоміжних скінченновимірних задач оптимізації з негладкими цільовими функціями. Розв'язок нелінійних задач ОРМ можна отримати у вигляді операторних рівнянь з параметрами, що відшукуються як оптимальний розв'язок допоміжних скінченновимірних негладких задач оптимізації, для розв'язання яких застосовуються різні модифікації  $r$ -алгоритму Н.З.Шора.

Інший широкий клас важливих прикладних задач зводиться до вирішення задач теорії оптимального розбиття множин в статичній або динамічних постановках. Прикладом таких задач може стати задача зрошування поля. Попри чітке прикладне застосування така задача дуже змістовно відповідає відповідній постановці задачі ОРМ з фіксованими центрами. В задачі вважається заданими розміщення джерел постачання води, яких на полі може бути декілька, та вартість транспортування (перекачування) одиниці об'єму до кожної точки поля. Враховуючи, що для заданої задачі адекватно застосувати евклідову метрику вартість транспортування пропорційне відстані між центром постачання та точкою на полі. Очевидно, що зручно зрошувати ділянки у формі кола з центром в точці постачання води, проте такими колами неможливо покрити все поле без перетину зон зрошування. В результаті поле може бути розбито на ділянки кожна з яких буде зрошуватись із певного центра, мінімізуючи при цьому витрати на транспортування води. Однією з цікавих з прикладної точки зору модифікацій такої задачі може стати задача розбиття поля на зони зрошування від центрів постачання води за умови різної потужності таких центрів. З прикладної точки зору така задача має широкий спектр застосування адже є більш практично орієнтованою. Очевидно, що джерела води мають різну потужність, адже можуть бути природніми де вода безкоштовна для власника, та централізованою, що має фіксовану ціну. В такому випадку оптимальне розбиття буде суттєво відрізнятись від попереднього, адже витрати на транспортування одного кубічного метра води співставно за витратами електроенергії та вартістю отримання такого об'єму води з системи централізованого водопостачання. Ще однією модифікацією наведеної задачі є задача з обмеженнями на пропускну спроможність каналів. Очевидно, що оперативне прийняття рішень, особливо в динамічних постановках, можливе, проте такі фактори як пропускну спроможність прокладених магістралей не дозволяє оперативно адаптувати систему до потрібних вимог, або витрати на таку адаптацію суттєво

збільшують загальний кошторис. Часто пропускна спроможність магістралей є фіксованою та розв'язання задачі суттєвим чином залежить саме від таких статичних факторів.

Ще одним вдалим прикладом використання теорії ОРМ для розв'язання прикладних задач є задача планування ділянки. Вважається, що задана ділянка довільної форми (у загальному вигляді у формі прямокутника) у кожній точці якої таблично чи аналітично задано висоту рельєфу. Задача полягає у побудові такого плану перевезення землі, що б з початкового рельєфу отримати бажаний. Початковими даними такої задачі є вартість транспортування одиниці об'єму ґрунту на одиницю відстані, початковий та кінцевий рельєф ділянки. У спрощеній постановці кінцевий вигляд ділянки вважається таким при якому об'єм ґрунту в кожній точці є однаковим, що суттєвим чином спрощує принаймні постановку, а у більшості випадків і розв'язання задачі. Проте, очевидно, що реальні задачі так не формулюються саме виходячи з прикладних міркувань, адже зумови суттєвих перепадів рельєфу, простіше та дешевше спроектувати різнорівневий рельєф, аніж рівномірний на всій ділянці, враховуючи, що її розмір може бути достатньо великим. Розв'язання таких задач спрощеними методами на практиці, як правило, не розглядається. У наведеній задачі практично обумовленими є також ряд модифікацій. Назвемо кілька з них:

- 1) в задачі може бути використаний не один агрегат для переміщення ґрунту, а декілька;
- 2) агрегати для переміщення ґрунту можуть мати різну потужність та різну продуктивність роботи;
- 3) в залежності від рельєфу (великі чи малі перепади) швидкість та вартість транспортування одиниці об'єму ґрунту може відрізнятись
- 4) суттєвим фактором такої задачі є переміщення агрегатів для транспортування ділянкою. У такій постановці задача може бути зведена до багатоетапної транспортної задачі та до задачі оптимального розбиття множин в динамічній постановці.

Вдалим прикладом задачі оптимального розбиття множин є задачі оптимального розміщення муніципальних установ. У випадку коли розміщення таких установ задано – задача полягає в оптимальному розбитті території населеного пункту на райони обслуговування такими установами. В таких задачах може бути враховано вартість транспортування пасажирів, місткість пасажиропотоків, наявність доріг та їх пропускну спроможність, типи транспорту та відповідну їм вартість транспортування. Проте іноді такі задачі вирішуються для постановки з відшукуванням центрів підмножин. Тобто задача стоїть не тільки у оптимальному розбитті населеного пункту на підмножини, але і у відшуканні оптимального розміщення центрів таких підмножин. Практична актуальність такої задачі полягає у планування розміщення муніципальних установ на моменті планування населеного пункту. Проте у такій постановці, на практиці, задача вирішується достатньо рідко. Здебільшого на території вже існують муніципальні установи сумарна потужність яких менша за запит населення. У такому випадку задача полягає у оптимальному розміщенні ще деякої (однієї, двох) кількості таких установ, для покриття потреби населення та можливої зміни конфігурації зон обслуговування.

Такі задачі знаходять своє застосування в ряді прикладних задач широко описаних в сучасній літературі. Зокрема задача про розбиття множини телефонних абонентів на зони обслуговування різними АТС або оптимального розміщення нових станцій [83]. Такі задачі, як правило зводяться до мінімізації кількості телефонного кабелю, що потрібно прокласти до нових та існуючих абонентів. Також вдалим прикладом прикладних задач, для вирішення яких можна застосувати класичні постановки теорії оптимального розбиття множин є задачі розбиття міста на шкільні регіони в такій задачі мінімізації підлягає вартість транспортування школярів до шкіл; задачі визначення меж виборчих округів та інші.

Вище перераховано ряд конкретних прикладних задач предметні особливості яких досить вдало описують елементи постановки задачі в теорії

ОРМ, проте для більш універсальної постановки задачі всі вони формулюються у вигляді нескінченновимірної задачі розміщення, що дозволяє уникнути ряду обмежень більш конкретизованої постановки, а методи розв'язання при цьому є класичними та зводять такі задачі до задач безумовної оптимізації. Крім того у наведених прикладах важливою прогалиною можна назвати статичність витрат. З прикладної точки зору очевидно, що для різних пунктів з тих чи інших причин вартість транспортування є різною, крім того виходячи з постійної зміни цін в країні такі ціни не тільки різняться, але й набувають змін з плином часу. Виходячи з таких міркувань важливою та актуальною постає задача оптимального розбиття множин в динамічній постановці, тобто з урахуванням можливості змін на транспортування з часом.

Нескінченновимірна задача розміщення описана, зокрема, в роботі Vollobás V. [66]. Вважається, що на заданій множині, що рівномірно покрита споживачами, необхідно розташувати пункти постачання/виробництва (із заданою для кожного рівнем потужності) і розбити множину на підмножини (кількість яких не перевищує кількість пунктів) так, щоб сумарні витрати на виробництво та транспортування продукції споживачам була мінімальною. Вважається, що кожен пункт виробляє/поставляє продукт для підмножини своїх споживачів, а зони на які розбита множина не перетинаються та в своїй сукупності представляють задану область.

В наведеній вище постановці виникають значна кількість особливостей, що можуть суттєво ускладнити задачу та вимагати розв'язання задач для постановок яких в наш час методів розв'язання не розроблено. Проте, в залежності від особливостей конкретної задачі загальна постановка може бути спрощена та приведена до більш простих задач, які мають своє як числове так і аналітичне рішення.

Одним з прикладів спрощеної постановки узагальненої задачі є нескінченновимірна транспортна задача, що описана в роботі Ус С. А., Станіної О. Д. [46]. Нескінченновимірна транспортна задача, є окремим,

спрощеним випадком попередньої. Особливість такої задачі полягає у початковому визначенні кількості постачальників та споживачів, потужності постачальників та вартості перевезення одиниці продукції від кожного постачальника до кожного споживача. Вирішення такої задачі здебільшого базується на методі потенціалів, що має ітераційну структуру та ряду методів для пошуку початкового наближення. В залежності від алгоритму обрання початкового наближення на кожній ітерації розв'язання виокремлюється одна з вершин області допустимих значень значення цільового функціоналу в якому не гірше за попередню ітерацію.

До інших прикладів задач спрощених постановок базової задачі є нескінченновимірною задачею про використання обладнання та інші прикладні задачі, що виникають, зокрема, в медицині, сільсько-господарському секторі [7], задачею розпізнавання образів [13] та інші.

Широкий клас таких задач, що у своїй постановці зводяться до розбиття області на підмножини, що не перетинаються можна поділити на дискретні та неперервні. Теоретичні результати дослідження дискретних задач присвячено ряд робіт вітчизняних авторів таких як Сергієнко І. В., Михалевича М. В., Кошляя Л. Б., Стецюка П. І. [38], Ляшко В. І. [45], Ю.Г.Стояна, С.В.Яковлева [193]-[195], А.Г.Сухарева, В.Р.Хачатурова, а також закордонних авторів R.Francis, M.Friedman, H.Corley, S.Roberts [83] та ін.

У випадку коли множина, що підлягає розбиттю є континуальною задачею класифікується, як неперервна. До неперервних задач можна віднести, як нескінченновимірні задачі на розміщення підприємств так і більш узагальнені транспортні задачі. У випадку нескінченновимірних задач розміщення підприємств, множина, що підлягає розбиттю розглядається як неперервно заповнена споживачами та підлягає розбиттю оптимальним чином на підмножини з метою мінімізації цільового функціоналу.

Результати досліджень нескінченновимірних задач можна умовно поділити на теоретичні та практичні. До теоретичних традиційно відносяться

результати що висвітлюють закономірності та причинно-наслідкові зв'язки, а під практичними, як правило розуміють можливість застосування теоретичних результатів в інших теоретичних задачах теорії оптимізації та математики загалом. Слід зазначити, що під практичними результатами також розуміється використання алгоритмів та методів оптимального розбиття направлених для вирішення конкретних прикладних задач, проте прикладні особливості таких задач суттєво звужують застосовність таких результатів для більш широких класів задач.

Основні теоретичні результати теорії нескінченновимірної оптимізації отримані ще у 50-х роках ХХ-го сторіччя Канторовичем Л.В. Результати отримані в тому числі при розв'язанні проблеми Г.Монжа. Деяке розширення результатів міститься в роботах Г.Ш Рубінштейна. Зокрема названими вченими успішно вирішена задачі про покриття, задача Неймана-Пірсона, задача на відшукання областей тяжіння локальних мінімумів багатоекстремальної функції тощо. Значущі результати в цьому питанні серед іноземних авторів належать S.D.Roberts, R.L. Francis, M.Friedman.

Однією з ефективних інструментів розв'язання неперервних задач оптимального розбиття є діаграми Вороного, що був розглянутий зарубіжними авторами, зокрема, А.Murat, V.Verter, G.Laporte [161]. Проте результати узагальненої теорії оптимального розбиття множин має значні переваги в порівнянні з названими вище методами, а саме:

- методи теорії ОРМ можуть бути застосовні для вирішення широкого класу задач оптимального розбиття, зокрема, однопродуктових та багатодуктових задач, задач розміщення-розбиття (location-allocation), задач при наявних обмеженнях у вигляді рівностей та нерівностей та задачі з розміщенням підприємств;

- при розв'язанні багатодуктових задач результатом роботи алгоритмів є розбиття на підмножини за кожним підприємством та кожним продуктом. Крім цього координати розміщення таких підприємств також

можуть бути розраховані оптимальним чином у випадку необхідності з предметної точки зору;

- застосування методів не залежить від розмірності задачі та у випадку програмної реалізації алгоритму багатовимірні результати можуть бути візуалізовані у зручному та наочно зрозумілому вигляді.

- розв'язок суттєвим чином не залежать від форми початкової множини, що підлягає розбиттю;

- аналітичні та числові алгоритми мають невисоку складність, у зв'язку з чим дозволяють розв'язувати задачі великої розмірності.

Створення фундаментальної теоретичної бази неперервної теорії та розв'язанню неперервних задач теорії оптимального розбиття множин з  $n$ -вимірного евклідового простору набуло свого розвитку у 70-х роках ХХ-го сторіччя та змістовно викладено у цілому циклі наукових праць вітчизняних та зарубіжних авторів. Серед зарубіжних авторів, що займалися цим питанням слід назвати Н. Corley та S. Roberts [83], а серед вітчизняних математиків назовемо таких фундаторів названої теорії, як І.В. Бейко та О.М. Кісельова [1]. Слід зазначити, що названим авторам хоч і вдалось закласти фундамент теорії, проте отримані результати окреслювали лише необхідних умов оптимальності розбиття.

Проте українській професорці, доктору фізико-математичних наук Кісельовій О.М. та учням її наукової школи вдалось отримати числові алгоритми та методи відшукування оптимальних рішень. Отримання числових методів стало можливим завдяки використанню методів недиференційованої оптимізації, зокрема декількох варіантів  $r$ -алгоритму, що був формалізований Н.З.Шором. Такі алгоритми можуть бути та використовуються для числового розв'язання скінченновимірних задач оптимізації, до яких зводяться нескінченновимірні задачі. Серед фундаментальних робіт з цієї тематики можна назвати такі монографії як [15], [17], [32], [39].

## 1.2 Класифікація неперервних задач оптимального розбиття множин $n$ -вимірною евклідового простору $E_n$ на підмножини

Для класифікації задач ОРМ наведемо загальну постановку неперервної задачі ОРМ, яка, зокрема, була сформульована у роботі [21].

Нехай  $\Omega$  – безперервна, обмежена, вимірною за Лебегом множина з  $n$ -вимірною евклідового простору  $E_n$ ,  $(\Omega, \Sigma, \mu)$  – вимірний простір,  $\Sigma$  – алгебра множини  $\Omega$ . У якості міри  $\mu$  розглядатимемо міру Лебега.

*Означення 1.1.* Розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин назвемо систему її підмножин  $(\Omega_1, \dots, \Omega_N)$ , щодо яких виконуються умови:

- 1)  $\Omega_i \subseteq \Omega, \forall i = \overline{1, N}$  ;
- 2)  $\bigcup_{i=1}^N \Omega_i = \Omega$  ;
- 3)  $\mu(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = \overline{1, N}$  .

Позначимо через  $\Sigma_{\Omega}^N$  простір всіх можливих розбиттів множини  $\Omega$  на  $N$  підмножин:

$$\Sigma_{\Omega}^N = \left\{ \bar{\omega} = (\Omega_1, \dots, \Omega_N) : \bigcup_{i=1}^N \Omega_i = \Omega, \mu(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = \overline{1, N} \right\}.$$

При цьому розбиття, що відрізняються тільки множиною нульової міри, будемо вважати рівними елементами простору  $\Sigma_{\Omega}^N$  .

Очевидно, що  $\Sigma_{\Omega}^N \subset \Sigma^N$  . Для елементів простору  $\Sigma^N = \underbrace{\Sigma \times \Sigma \times \dots \times \Sigma}_N$  – декартового добутку  $N$  алгебр множини  $\Omega$ , а

відповідно й для елементів  $\Sigma_{\Omega}^N$  , введемо метрику

$$\rho(\{(\Omega_1, \dots, \Omega_N)\}, \{(\bar{\Omega}_1, \dots, \bar{\Omega}_N)\}) = \sqrt{\sum_{i=1}^N \mu^2(\Omega_i \otimes \bar{\Omega}_i)},$$

де під означенням  $\Omega_i \otimes \bar{\Omega}_i$  розуміється симетрична різниця множин.

Метричний простір, побудований на базі  $\sum_{\Omega}^N \subset \sum^N$ , обмежений, щільний в собі, хаусдорфовий, замкнутий.

*Постановка загальної задачі ОРМ.* Нехай функції  $F^j, G_1^j, \dots, G_m^j : \mathfrak{R}_{\Omega}^N \rightarrow R^1, j = 1, \dots, M$  – псевдонеперервні функції розбиття;

$$F(\Omega_1^1, \dots, \Omega_N^1; \dots; \Omega_1^M, \dots, \Omega_N^M) = \sum_{j=1}^M F^j(\Omega_1^j, \dots, \Omega_N^j);$$

$$G(\Omega_1^1, \dots, \Omega_N^1; \dots; \Omega_1^M, \dots, \Omega_N^M) = \sum_{j=1}^M G^j(\Omega_1^j, \dots, \Omega_N^j), i = 1, \dots, m.$$

Потрібно знайти набір підмножин  $(\Omega_1^{1*}, \dots, \Omega_N^{1*}; \dots; \Omega_1^{M*}, \dots, \Omega_N^{M*}) \in (\mathfrak{R}_{\Omega}^N)^M$ , при якому функція розбиття  $F$  досягає свого мінімального значення і виконуються умови:

$$G_i(\Omega_1^1, \dots, \Omega_N^1; \Omega_1^M, \dots, \Omega_N^M) \leq 0, i = 1, \dots, p,$$

$$G_i(\Omega_1^1, \dots, \Omega_N^1; \Omega_1^M, \dots, \Omega_N^M) = 0, i = p + 1, \dots, m.$$

Уведемо наступні позначення:

$$(\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) = (\Omega_1^1, \dots, \Omega_N^1; \Omega_1^2, \dots, \Omega_N^2; \Omega_1^M, \dots, \Omega_N^M) \in (\sum_{\Omega}^N)^M;$$

$$U = \left\{ (\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) \in (\mathfrak{R}_{\Omega}^N)^M : G_i(\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) \leq 0, i = \overline{1, p}; \right. \\ \left. G_i(\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) = 0, i = \overline{p+1, m} \right\}.$$

Тоді, задачу оптимального розбиття множин можна записати так:

$$F(\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) \rightarrow \min_{(\bar{\omega}^{(1)}, \bar{\omega}^{(2)}, \dots, \bar{\omega}^{(M)}) \in U} \quad (1.7)$$

Отже, виходячи з того, що різна постановка оптимізаційних задач та широкий перелік їх застосувань забезпечує широке коло їх застосувань на практиці. До основних неперервних задач оптимального розбиття множин  $n$ -вимірного евкладового простору  $E_n$  на підмножини слід віднести такі:

1) неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  без обмежень із заданим положенням центрів підмножин;

- 2) неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  без обмежень із невідомим положенням центрів підмножин;
- 3) неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  при обмеженнях із заданим положенням центрів підмножин;
- 4) неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  при обмеженнях із невідомим положенням центрів підмножин.

Нижче наведемо постановку та алгоритми розв'язку частини названих задач.

### **Неперервна однопродуктова задача оптимального розбиття множин з $E_n$ без обмежень із заданим положенням центрів підмножин:**

Найпростіша однопродуктова задача ОРМ без обмежень з заданими координатами характерних для кожної підмножини точок (центрами) має такий вигляд:

Нехай  $\Omega$  – обмежена, вимірна за Лебегом множина в  $n$ -вимірному евклидовому просторі  $E_n$ . Сукупність вимірних за лебегом підмножин  $\Omega_1, \dots, \Omega_N$  з  $\Omega \subset E_n$  назвемо можливим розбиттям множини, якщо  $\bigcup_{i=1}^N \Omega_i = \Omega, mes(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = 1, \dots, N$ , де  $mes(\cdot)$  означає міру Лебега.

Позначимо клас всіх можливих розбиттів множини  $\Omega$  як  $\sum_{\Omega}^N$ , тобто

$$\sum_{\Omega}^N = \{(\Omega_1, \dots, \Omega_N) : \bigcup_{i=1}^N \Omega_i = \Omega, mes(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = 1, \dots, N\}.$$

Введемо функціонал

$$F(\{\Omega_1, \dots, \Omega_N\}) = \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx, \quad (1.8)$$

где функції  $c(x, \tau_i)$  – дійсні, обмежені, визначені на  $\Omega \times \Omega$ , вимірні по  $x$  при будь-якому фіксованому  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)})$  з  $\Omega$  для всіх  $i = 1, \dots, N$ ;  $\rho(x)$  – обмежена, вимірна, невід'ємна на  $\Omega$  функція;  $a_1, \dots, a_N$  – задані невід'ємні числа.

Тут і надалі інтеграли розуміються в сенсі Лебега. Будемо вважати, що міра множини граничних точок  $\Omega_i, i=1, \dots, N$ , дорівнює нулю.

Тоді під неперервною одно продуктовою задачею оптимального розбиття множини  $\Omega$  з  $E_n$  на його множини  $\Omega_1, \dots, \Omega_N$ , що не перетинаються (серед яких можуть бути і пусті) без обмежень з заданими координатами центрів  $\tau_1, \dots, \tau_N$  цих підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, будемо розуміти наступну задачу.

**Задача А1.** Знайти

$$\min_{\{\Omega_1, \dots, \Omega_N\} \in \Sigma_{\Omega}^N} F(\{\Omega_1, \dots, \Omega_N\}),$$

де

$$F(\{\Omega_1, \dots, \Omega_N\}) = \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx;$$

$x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ ;  $\tau = (\tau_1, \dots, \tau_N) \in \underbrace{\Omega \dots \Omega}_{\Omega^N}$ ;  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)})$ ;  $a_1, \dots, a_N$  – задані

невід'ємні числа.

Розбиття  $\{\Omega_1^*, \dots, \Omega_N^*\}$ , що є розв'язком задачі, назвемо оптимальним.

На рис.1.3 зображено розбиття множини  $\Omega \subset E_2$  на три підмножини  $\Omega_1, \Omega_2, \Omega_3$  з центрами  $\tau_1, \tau_2, \tau_3$  цих підмножин, відповідно.

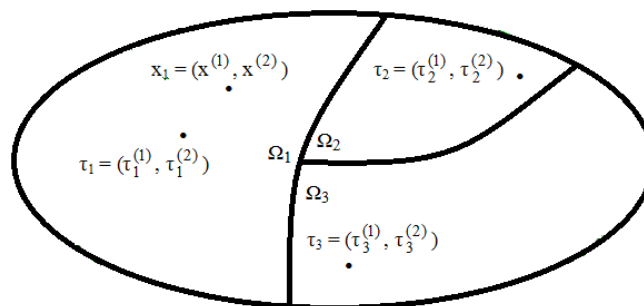


Рис. 1.3. Розбиття множини  $\Omega$  на три підмножини

Введемо характеристичну функцію

$$\lambda_i(x) = \begin{cases} 1, & x \in \Omega_i \\ 0, & x \in \Omega \setminus \Omega_i, i = 1, \dots, N \end{cases}$$

підмножини  $\Omega_i, i = 1, \dots, N$ , та перепишемо задачу А1 в іншому вигляді.

**Задача В1. Знайти**

$$\min_{\lambda(\cdot) \in \Gamma_1} \int \sum_{\Omega, i=1}^N (c(x, \tau_i) + a_i) \rho(x) \lambda_i(x) dx,$$

де

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ майже всюду (м.в.) для } x \in \Omega,$$

$$\lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \};$$

$$\tau = (\tau_1, \dots, \tau_N) - \text{заданий вектор з } \Omega^N.$$

**Неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  без обмежень із невідомим положенням центрів підмножин**

Під неперервною одно продуктовою задачею оптимального розбиття множини  $\Omega$  з  $E_n$  на його підмножини  $\Omega_1, \dots, \Omega_N$ , що не перетинаються (серед яких можуть бути і пусті) без обмежень з вішукуванням координат центрів  $\tau_1, \dots, \tau_N$  підмножин  $\Omega_1, \dots, \Omega_N$ , відповідно, будемо розуміти наступну задачу.

**Задача А2. Знайти**

$$\min_{\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\}} \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx,$$

$$\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N, \tau = (\tau_1, \dots, \tau_N) \in \underbrace{\Omega \dots \Omega}_{\Omega^N},$$

де

$$x = (x^{(1)}, \dots, x^{(n)}) \in \Omega; \tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega.$$

Перепишемо задачу **A2** в термінах характеристичних функцій  $\lambda_i(x)$  підмножин  $\Omega_i, i = 1, \dots, N$ , в наступному вигляді:

**Задача В2. Знайти**

$$\min_{(\lambda(\cdot), \tau) \in \Gamma_1 \times \Omega^N} \int \sum_{\Omega, i=1}^N (c(x, \tau_i) + a_i) \rho(x) \lambda_i dx,$$

де

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega, \\ \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \}; \\ \tau = (\tau_1, \dots, \tau_N) \in \underbrace{\Omega \times \dots \times \Omega}_N = \Omega^N.$$

**Неперервна однопродуктова задача оптимального розбиття множин з  $E_n$  при обмеженнях із заданим положенням центрів підмножин**

Під неперервною одно продуктовою задачею оптимального розбиття множини  $\Omega$  з  $E_n$  на його підмножини  $\Omega_1, \dots, \Omega_N$ , що не перетинаються (серед яких можуть бути і пусті) при обмеженнях у формі рівностей та нерівностей з заданими координатами центрів  $\tau_1, \dots, \tau_N$  підмножин  $\Omega_1, \dots, \Omega_N$ , відповідно, будемо розуміти наступну задачу.

**Задача А3.** Знайти

$$\min_{\{\Omega_1, \dots, \Omega_N\}} \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx$$

при умовах

$$\int_{\Omega_i} \rho(x) dx = b_i, i = 1, \dots, p,$$

$$\int_{\Omega_i} \rho(x) dx \leq b_i, i = p + 1, \dots, N,$$

$$\{\Omega_1, \dots, \Omega_N\} \in \Sigma_{\Omega}^N,$$

де  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega; \tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega; b_1, \dots, b_N$  – задані додатні числа, причому виконуються умови розв'язності задачі

$$S = \int_{\Omega} \rho(x) dx \leq \sum_{i=1}^N b_i, 0 < b_i < S, i = 1, \dots, N.$$

Перепишемо задачу **А3** в термінах характеристичних функцій  $\lambda_i(x)$  підмножин  $\Omega_i, i = 1, \dots, N$ , в наступному вигляді.

**Задача В3.** Знайти

$$\min_{(\lambda(\cdot)) \in \Gamma_2} \int_{\Omega} \sum_{i=1}^N (c(x, \tau_i) + a_i) \rho(x) \lambda_i dx,$$

де

$$\Gamma_2 = \{ \lambda(x) : \lambda(x) \in \Gamma_1 \text{ м.в. для } x \in \Omega,$$

$$\int_{\Omega} \rho(x) \lambda_i(x) dx = b_i, i = 1, \dots, p,$$

$$\int_{\Omega} \rho(x) \lambda_i(x) dx \leq b_i, i = p + 1, \dots, N \};$$

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N,$$

$$\sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega \};$$

$$\tau = (\tau_1, \dots, \tau_N) - \text{заданий вектор з } \Omega^N.$$

### Неперервна однопродуктова задача оптимального розбиття множин з $E_n$ при обмеженнях із невідомим положенням центрів підмножин

Під неперервною одно продуктовою задачею оптимального розбиття множини  $\Omega$  з  $E_n$  на його підмножини  $\Omega_1, \dots, \Omega_N$ , що не перетинаються (серед яких можуть бути і пусті) при обмеженнях у формі рівностей та нерівностей з відшукуванням координат центрів  $\tau_1, \dots, \tau_N$  цих підмножин відповідно будемо розуміти наступну задачу.

**Задача А4.** Знайти

$$\min_{\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\}} \sum_{i=1}^N \int_{\Omega_i} (c(x, \tau_i) + a_i) \rho(x) dx$$

при умовах

$$\int_{\Omega_i} \rho(x) dx = b_i, i = 1, \dots, p,$$

$$\int_{\Omega_i} \rho(x) dx \leq b_i, i = p + 1, \dots, N,$$

$$\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N, \tau = (\tau_1, \dots, \tau_N) \in \Omega^N,$$

де  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega; \tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega; a_1, \dots, a_N, b_1, \dots, b_N$  – задані

невід'ємні числа, причому виконуються умови розв'язності задачі

$$S = \int_{\Omega} \rho(x) dx \leq \sum_{i=1}^N b_i, 0 < b_i < S, i = 1, \dots, N.$$

Перепишемо задачу **A4** в термінах характеристичних функцій  $\lambda_i(x)$  підмножин  $\Omega_i, i = 1, \dots, N$ , в наступному вигляді.

**Задача B4.** Знайти

$$\min_{(\lambda(\cdot)) \in \Gamma_2 \times \Omega^N} \int_{\Omega} \sum_{i=1}^N (c(x, \tau_i) + a_i) \rho(x) \lambda_i dx,$$

де

$$\Gamma_2 = \{\lambda(x) : \lambda(x) \in \Gamma_1 \text{ м.в. для } x \in \Omega,$$

$$\int_{\Omega} \rho(x) \lambda_i(x) dx = b_i, i = 1, \dots, p,$$

$$\int_{\Omega} \rho(x) \lambda_i(x) dx \leq b_i, i = p + 1, \dots, N\};$$

$$\Gamma_1 = \{\lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N,$$

$$\sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega\};$$

$$\tau = (\tau_1, \dots, \tau_N) \in \underbrace{\Omega \times \dots \times \Omega}_N = \Omega^N.$$

Таким чином, виходячи з загальної постановки та наведених прикладів задач неперервні задачі ОРМ можна класифікувати за такими ознаками:

- за видом цільового функціоналу  $F$ ;
- за кількістю продуктів;
- за положенням центрів;
- за наявністю обмежень.

На рис.1.4. наведено графічні результати застосуванні числових алгоритмів розбиття в задачах, для одно- та багатопродуктових задач оптимального розбиття множин.

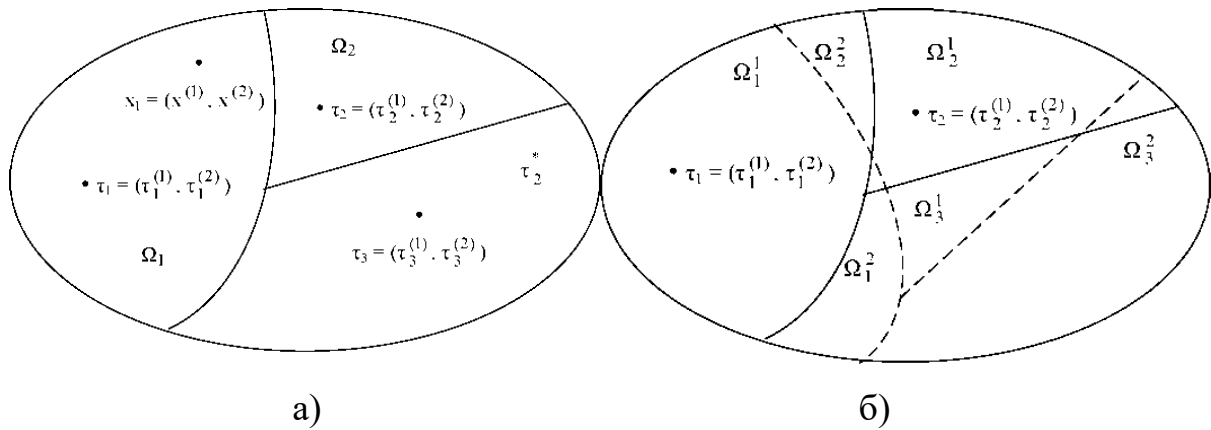


Рисунок 1.4. Приклад результатів застосування чисельного алгоритму розбиття множини на три підмножини для задачі з а) одним та б) двома продуктами

В залежності від постановки задачі та наявності в ній параметрів, що змінюються з плином часу розрізняють задачі:

- статичні задачі ОРМ;
- динамічні задачі ОРМ.

За наявності обмежень у вигляді рівностей та/або нерівностей розрізняють задачі:

- без обмежень:  $U \equiv (\sum_{\Omega}^N)^M$ ;
- з обмеженнями:  $U \subset (\sum_{\Omega}^N)^M$ .

Як було сказано вище, одним з ключових параметрів задачі оптимального розбиття множин на підмножини є центри підмножин, що є результатом розбиття. За цим критерієм задачі ОРМ розрізняють так:

- з фіксованими центрами;
- з шуканими центрами.

За характером параметрів, що входять до задачі розрізняють:

- детерміновані
  - лінійні
  - нелінійні
  - однопродуктові

- багатопродуктові
- з обмеженнями
- без обмежень
- із заданим положенням центрів [10], [15]
- з відшукуванням оптимального розташування центрів [18], [29], [54]-[55].

- задачі ОРМ в умовах невизначеності [14], [29]-[32],;

- динамічні задачі ОРМ [110], [113]-[114];

Для розв'язання статичних та динамічних задач ОРМ існує єдиний підхід, що полягає у зведенні, наприклад, через функціонал Лагранжа, до скінченновимірних задач максимізації/мінімізації або задач максиміна, для розв'язку яких застосовуються існуючі відомі методи недиференційовної оптимізації, зокрема, модифікації  $r$ -алгоритму Н.З.Шора, [180].

### **1.3 Задачі оптимального розбиття множин в нечітких постановках**

В попередніх параграфах роботи були чітко окреслені та сформульовані математичні моделі оптимального розбиття множин в різних проте однопродуктових постановках. Для формалізації задачі в умовах невизначеності наведемо спочатку задачу оптимального розбиття в умовах визначеності для в узагальненій постановці. Зауважимо, що багатопродуктові задачі, що безумовно є практико орієнтованими та більш адекватно відображають взаємодію виробництв між собою та зі споживачами можна вирішити шляхом розбиття багатопродуктової задачі на обмежену кількість однопродуктових, принаймні для спрощення візуального відображення результатів чисельного експерименту. Змістовних та комплексних методів розв'язання таких задач, які б узагальнювали підходи до розв'язання саме багатопродуктових проблем у наш час застосовуються мало, та вони мають достатньо просту структуру.

Однопродуктова адача ОРМ у загальній постановці. Нехай  $\Omega$  – обмежена, вимірنا за Лебегом множина  $n$ -вимірного евклідового простору  $E_n$ .

Сукупність вимірних за Лебегом підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega$  будемо називати можливим розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин, якщо

$$\bigcup_{i=1}^N \Omega_i = \Omega$$

$$mes(\Omega_i \cap \Omega_k) = 0, i \neq k, i, k = 1, \dots, N$$

причому  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)})$  – центр для підмножини  $\Omega_i$ , що належить  $\Omega$ .

Позначимо сукупність всіх можливих розбиттів множини  $\Omega$  на  $N$  підмножин через  $\sum_{\Omega}^N$ , тобто

$$\sum_{\Omega}^N = \left( \{ \Omega_1, \dots, \Omega_N \} : \bigcup_{i=1}^N \Omega_i = \Omega, mes(\Omega_i \cap \Omega_k) = 0, i \neq k, i, k = 1, \dots, N \right)$$

Під неперервною нелінійною одно продуктовою динамічною задачею оптимального розбиття множини  $\Omega \subset E_n$  на її підмножини  $\Omega_1, \dots, \Omega_N$  (серед яких можуть бути порожні) при обмеженнях у формі рівностей і нерівностей з відшукуванням координат центрів підмножин будемо розуміти наступну задачу:

*Задача 1.1.* Знайти

$$\min_{(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\})} \sum_{i=1}^N \left[ \varphi_i \left( \int_{\Omega_i} \rho(x) dx \right) + \int_{\Omega_i} c(x, \tau_i) \rho(x) dx \right]$$

при умовах

$$\{ \Omega_1, \dots, \Omega_N \} \in \sum_{\Omega}^N, \tau = (\tau_1, \dots, \tau_N) \in \Omega^N$$

де  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega; \tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega$

Функції  $\varphi_i(x)$  – дійсні, обмежені, опуклі (або увігнуті), двічі неперервно-диференційовні функції для всіх  $i = 1, \dots, N$ .

Існує великий перелік прикладних задач в яких дана постановка може бути застосована, зокрема, оптимальне розміщення підприємств та віднесення споживачів до таких підприємств, як до центрів постачання. Під підприємствами, у даному випадку можуть розумітись станції стільникового зв'язку, лікарні, школи, ЦНАП-и тощо. Критерієм оптимальності розміщення можуть виступити сумарні витрати на виробництво і доставку продукції (послуги) до споживача. Здебільшого на практиці підприємства виробляють більше однієї одиниці продукції, проте для такої продукції характерна, здебільшого однорідність за переважною більшістю ключових факторів. Так, наприклад, послуги шкіл, або дитячих садочків не стільки обмежені окремими віком учнів, як їх загальною кількістю, послуги ЦНАП-ів є достатньо різноманітними, проте, також, обмежені кількістю звертань, а не їх типом. Виробники продукції, можуть та випускають широкий перелік товарів, проте вартість їх транспортування, здебільшого залежить від кількості, а не найменування і т.д.

Неперервні задачі теорії оптимального розбиття множин в математичній постановці розглядаються для випадків коли множина, що підлягає розбиттю є континуальною, проте переважна більшість реальних задач оперує об'єктами, що мають дискретну структуру. Використання в таких випадках дискретних задач можливо лише коли кількість типових об'єктів є не невеликою. При цьому у випадках коли кількість об'єктів обчислюється десятками математична модель вже стає настільки складною, що обчислення може займати тривалий час. Зважають на відносну рівномірність розміщення споживачів, їх велику кількість до дискретних задач такого роду можна застосовувати неперервні постановки та вважати, що «споживач розподілений неперервно», і, відповідно, розглядати неперервну задачу оптимального розбиття.

У більшості практичних задач крім цільового функціоналу, що окреслює оптимальність шуканого оптимального розбиття виникають обмеження різного роду. Такі обмеження продиктовані виробничими масштабами, потужністю виробництва, пропускною спроможністю тощо. Обмеження можуть бути формалізовані у вигляді рівнянь або нерівностей, проте безпосередні обмеження є достатньо умовними та визначаються експертно. Наприклад, підприємство може виробляти протягом певного періоду часу обмежену кількість товарів типу А, проте підприємство випускає не один тип товарів та за умови скорочення (можливо непропорційного) виробництва інших товарів кількість товарів А може бути збільшено. Можливі випадки оперативних змін у випадку запиту на більшу кількість товарів, тому обмеження в таких постановках є достатньо слабо визначеними та вкладаються у певний інтервал значень.

Наведені вище причини невизначеності не є вичерпними, та здебільшого невизначеність обумовлена недостатньою кількістю інформації, що покладається в основу прийняття рішення. До основних джерел невизначеності, можемо віднести:

- 1) принципова невизначеність, що обумовлена класичними постановками задачі;
- 2) невизначеність, обумовлена кількістю елементів системи кількість яких априорі оцінюється приблизно;
- 3) невизначеність, яка може бути ліквідована проте вартість такого уточнення співставна із значенням цільового функціоналу витрат;
- 4) невизначеність, зумовлена неповнотою вхідної інформації (суперечливість, неповнота, надлишковість, невизначеність, неоднозначність та ін.) або недостовірність в силу різних причин;
- 5) невизначеність, зумовлена зовнішнім впливом, нестійкістю системи та іншими факторами що впливають на процес прийняття рішення.

Виходячи з широкого переліку факторів, що вносять невизначеність в класичні постановки задачі оптимального розбиття множин необхідний

чіткий математичний інструментів, що дозволить описати проблему, отримати рішення, який включав би в себе прояви невизначеності, і враховував би її при отриманні оптимального, в невизначеній постановці, розв'язку.

Найпершим апаратом врахування нечіткості в задачах оптимального розбиття множин був апарат, що базується на базових правилах теорії ймовірностей, де невизначеності ставилась у відповідність нормована міра, що характеризувала імовірність наперед заданих випадкових результатів або їх множин.

Імовірності методи знаходять своє застосування в теорії ігор, теорії статистичних рішень. В певному сенсі дві названі теорії можна вважати крайніми випадками невизначеності.

Оптимізаційні задачі оптимального розбиття множин в умовах невизначеності в залежності від вихідної постановки розглядають в рамках теорії нескінченновимірного математичного програмування при невизначеності або неповноті інформації про вихідні дані. У випадках, коли немає можливості оцінити параметри процесів, що впливають на передбачувані значення параметрів задачі, вважається, що задача вирішується в умовах невизначеності.

З практичної точки зору задачі з невизначеністю є більш застосовними на практиці ніж детерміновані, та більш адекватно у своїй постановці можуть описати проблему. Зауважимо, що характеристика нечіткості з плином часу може змінюватись, адже хоч і не повною мірою визначені параметри можуть, наприклад, змінювати межі значень своїх невизначених параметрів. Певним чином невизначеність може невілювати такі динамічні аспекти, проте навіть найпростіші процеси, що спіткають людину щодня, можуть змінюватись навіть у нечітких постановках. Відповідно нечіткі задачі за змінними характеристиками нечітких параметрів можуть бути зведені у своїй постановці до динамічних задач оптимального розбиття з нечіткими параметрами.

Змістовно задачі з нечіткими параметрами, тобто стохастичні задачі оптимального розбиття множин досліджувалися у роботах О.М.Кісельової, Л.С.Коряшкіна, Н.З.Шор, С.А.Ус [13], [21].

На основі сформульованої вище однопродуктової задачі ОРМ у загальній постановці наведемо одноетапну стохастичну однопродуктову задачу ОРМ з обмеженнями у вигляді рівностей та нерівностей з відшукуванням координат центрів підмножин у наступному вигляді [21].

*Задача 1.2.* Знайти

$$\min_{(\{\Omega_1, \dots, \Omega_N\}, \{\tau_1, \dots, \tau_N\})} \sum_{i=1}^N \int_{\Omega_i} [c(x, \tau_i, \xi_i) + a_i] \rho(x, \xi_0) dx$$

за умов

$$\int_{\Omega_i} \rho(x, \xi_0) dx = b_i, i = 1, \dots, p,$$

$$\int_{\Omega_i} \rho(x, \xi_0) dx \leq b_i, i = p + 1, \dots, N$$

$$\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N, \tau = (\tau_1, \dots, \tau_N) \in \Omega^N, \mathcal{J}$$

де  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ ;  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega$ ;  $a_1, \dots, a_N$ ;  $b_1, \dots, b_N$  – задані невід’ємні числа, причому виконуються умови розв’язності задачі:

$$S = \int_{\Omega} \rho(x, \xi_0) dx \leq \sum_{i=1}^N b_i, 0 \leq b_i \leq S, i = 1, \dots, N;$$

$\xi_i = \xi_i(\theta) : \Theta \rightarrow R (i = 0, \dots, N)$  – випадкові величини на імовірнісному просторі  $(\Theta, \mathcal{J}, P)$  із заданими математичними очікуваннями  $\bar{\xi}_0, \dots, \bar{\xi}_N$  і дисперсіями  $\hat{\xi}_0, \dots, \hat{\xi}_N$ . Такі задачі називають нечіткими задачами оптимального розбиття множин.

Найбільш описаним та застосованим наукових напрямів, де активно використовують теорію нечітких множин та інші інтелектуальні технології, є обчислювальний інтелект, в базових підходах якої є можливість об’єднання в єдину нейронечітку технологію управління невизначеністю та оптимізацію.

#### **1.4 Біфуркаційні ефекти в динамічних задачах оптимального розбиття множин**

Знання фундаментальних законів тієї галузі науки, до якої належить природа задачі, відіграє важливу, а подекуди й вирішальну роль у процесі дослідження та математичного моделювання природних явищ. Водночас необхідно мати у розпорядженні методи й прийоми, що дозволяють отримувати уявлення про властивості системи без потреби у знаходженні точного аналітичного розв'язку. Такі інструменти надає якісна теорія диференціальних рівнянь, що спирається на загальні теореми існування й єдиності розв'язку, результати про неперервність, а також положення щодо залежності розв'язків від початкових умов і параметрів.

Безумовно, побудова адекватної математичної моделі реальних процесів — це завдання високої складності. Універсального алгоритму моделювання не існує, тому для кожного типу моделей необхідно формувати специфічні підходи. У багатьох випадках дослідники змушені обмежуватися використанням спрощених, здебільшого лінійних, моделей. Вони можуть бути корисними на початковому етапі аналізу, коли потрібно окреслити загальну структуру системи й виявити базові взаємозв'язки між її параметрами. Втім, дослідження складних систем не може ґрунтуватися лише на лінійних підходах, оскільки останні не враховують важливих нелінійних ефектів. Врахування такого роду ефектів, зокрема, можливо з використанням біфуркаційних властивостей систем, тобто якісні зміни характеру поведінки системи при неперервній (плавній) зміні значень її параметрів.

Однією з ключових задач якісної теорії диференціальних рівнянь є вивчення топологічної структури фазового простору в околах особливих точок. Ще А. Пуанкаре у своїх роботах показав, що хоча для систем другого порядку існує обмежена кількість типів точок рівноваги, їхня класифікація відіграє важливу роль у розумінні динаміки. До таких типів належать: стійкі та нестійкі вузли, сідла, фокуси, вироджені та зіркові вузли, центр.

Подальший внесок у класифікацію точок рівноваги зробили А.Д. Базикін, Ю.А. Кузнецов, А.І. Хибник. Вагому працю у цьому напрямку здійснили Д. Ерроусміт і К. Плейс, які, окрім традиційних означень і теорем теорії лінійних систем, надали класифікацію фазових портретів на площині, розмежувавши їх за алгебраїчними та топологічними ознаками. У цьому ж контексті були проаналізовані локальна й глобальна поведінка динамічних систем поблизу стаціонарних точок, умови виникнення складних режимів та стійкість таких режимів, а також критерії появи біфуркацій.

Вивчення біфуркаційних значень параметрів є невіддільною частиною аналізу нелінійних моделей і дає змогу виявити критичні етапи у зміні поведінки системи. Якісному аналізу диференціальних систем присвячено чимало ґрунтовних праць і монографій, серед яких варто відзначити роботи М. Braun, Р. Hartman, D.W. Jordan, Р. Smith та інших. У цих працях основну увагу зосереджено на побудові нових моделей, дослідженні їхніх властивостей класичними методами, аналізі біфуркаційних явищ, проблемах керування та оптимізації. Значна частина сучасних методів спирається на концепції, запропоновані А. Пуанкаре та А.М. Ляпуновим. Паралельно з цим сучасні дослідники, зокрема В.П. Жуков, розвивають напрями, пов'язані з виявленням інваріантів систем, вивченням умов локальної й глобальної стійкості, розробкою методів керування та побудовою зворотних зв'язків.

У випадку системи з трьох диференціальних рівнянь, її загальний вигляд можна подати у такій формі

$$\begin{cases} \dot{x} = A_{11}x + A_{12}y + A_{13}z \\ \dot{y} = A_{21}x + A_{22}y + A_{23}z \\ \dot{z} = A_{31}x + A_{32}y + A_{33}z \end{cases}$$

Якщо характеристичне рівняння цієї системи має три різні дійсні корені, то можна здійснити таке невироджене перетворення

$$\begin{cases} \zeta = c_{11}x + c_{12}y + c_{13}z \\ \eta = c_{21}x + c_{22}y + c_{23}z, \\ \rho = c_{31}x + c_{32}y + c_{33}z \end{cases}$$

яке дозволяє звести систему до такого вигляду

$$\begin{cases} \dot{\zeta} = \lambda_1 \zeta \\ \dot{\eta} = \lambda_2 \eta, \\ \dot{\rho} = \lambda_3 \rho \end{cases}$$

а відповідна характеристична матриця при цьому набуває такого вигляду

$$G = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \quad (1.9)$$

У результаті інтегрування системи дістаємо

$$\begin{cases} \zeta = C_1 e^{\lambda_1 t} \\ \eta = C_2 e^{\lambda_2 t} . \\ \rho = C_3 e^{\lambda_3 t} \end{cases}$$

Переходячи до границі, отримуємо:

- 1)  $\lambda_1 > 0$ , то  $\lim_{t \rightarrow \infty} \zeta(t) = \infty$ ;  $\lambda_1 < 0$ , то  $\lim_{t \rightarrow \infty} \zeta(t) = 0$ ;
- 2)  $\lambda_2 > 0$ , то  $\lim_{t \rightarrow \infty} \eta(t) = \infty$ ;  $\lambda_2 < 0$ , то  $\lim_{t \rightarrow \infty} \eta(t) = 0$ ;
- 3)  $\lambda_3 > 0$ , то  $\lim_{t \rightarrow \infty} \rho(t) = \infty$ ;  $\lambda_3 < 0$ , то  $\lim_{t \rightarrow \infty} \rho(t) = 0$ .

У разі якщо дискримінант характеристичного рівняння початкової системи дорівнює нулю, то відповідна характеристична матриця матиме іншу, але споріднену з попередньою, структуру. У ситуації, коли власні значення утворюють пару  $\lambda_2$  та  $\lambda_3$  є комплексними, а матриця  $G$  (1.9) набуде комплексного вигляду. Щоб звести систему з комплексними власними значеннями до найпростішої форми, здійснюється перехід за допомогою невиродженого перетворення від  $\eta, \rho$  до  $U, V$ . Внаслідок цього система приймає такий вигляд:

$$\begin{cases} \dot{\zeta} = \zeta \\ \dot{\eta} = U + iV, \\ \dot{\rho} = U - iV \end{cases}$$

звідки

$$\begin{cases} U = (\eta + \rho)/2 \\ V = (\eta - \rho)/2i \end{cases}$$

Спираючись на останню систему і враховуючи її приведену форму, дістанемо  $\dot{U} = (\dot{\eta} + \dot{\rho})/2 = (\lambda_2\eta + \lambda_3\rho)/2 = [(\alpha + i\beta)(u + iV) + (\alpha - i\beta)(u - iV)]/2 = \alpha U + \beta V$  аналогічно  $\dot{V} = \alpha V - \beta U$ . Отже, якщо система має пару комплексно спряжених власних значень, можна виконати невироджене перетворення, яке зводить систему до такого вигляду

$$\begin{cases} \dot{\zeta} = \lambda_1\zeta \\ \dot{U} = \alpha U + \beta V \\ \dot{V} = -\beta U + \alpha V \end{cases}$$

Здійснивши зміну на циліндричну систему координат, приходимо до:

$$\begin{cases} \zeta = \zeta \\ U = r \cos \theta \\ V = r \sin \theta \end{cases}$$

Виконуючи перетворення з урахуванням рівності  $U^2 + V^2 = r^2$ , маємо  $\dot{r} = \alpha \cdot r$ , та відповідно  $\operatorname{tg} \theta = V/U$ . Застосувавши диференціювання до отриманої тригонометричної рівності, дістанемо  $\dot{\theta} = \beta$  та систему у вигляді

$$\begin{cases} \zeta = C_1 e^{\lambda_1 t} \\ r = r_0 e^{\alpha t} \\ \theta = \beta \cdot t + \theta_0 \end{cases}$$

Перейдемо до границі:

$$1) \alpha > 0 \text{ TO } \lim_{t \rightarrow \infty} r(t) = \infty; \alpha < 0 \text{ TO } \lim_{t \rightarrow \infty} r(t) = 0;$$

$$2) \beta > 0 \text{ TO } \lim_{t \rightarrow \infty} \theta(t) = \infty; \beta < 0 \text{ TO } \lim_{t \rightarrow \infty} \theta(t) = -\infty;$$

$$3) \alpha = 0 \text{ TO } \lim_{t \rightarrow \infty} r(t) = \text{const}.$$

Отже, коли два з трьох коренів характеристичного рівняння динамічної системи є чисто уявними, проте  $\operatorname{Det}(J) \neq 0$ , характеристична матриця системи

$$\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & 0 & \beta \\ 0 & -\beta & 0 \end{pmatrix}.$$

Вивчення динамічних систем, як у простих випадках, так і в більш складних постановках, потребує наявності списку топологічно-нееквівалентних фазових портретів для систем із розмірністю більше двох. Запропонована методологія дає змогу виділити топологічно-нееквівалентні фазові портрети навіть для систем високої розмірності. Однак, як показано вище, при  $n = 3$  загальна кількість таких портретів, враховуючи вироджені варіанти, досягає 41.

## 1.5 Висновки до розділу 1

В першому розділі дисертаційної роботи здійснено змістовний огляд сучасного стану теорії оптимального розбиття множин, що є фундаментальною теорією нескінченновимірного математичного програмування. Детально розглянуто складові теорії оптимального розбиття множин, зокрема, статичні та динамічні постановки, лінійні та нелінійні цільові функціонали, одно- та багато продуктові задачі, задачі з фіксованими центрами та з відшукуванням оптимальних координат таких центрів, задачі з обмеженнями та без обмежень, постановки задач в умовах невизначеності та теоретичні і практичні аспекти їх прикладного застосування.

Здійснено критичний огляд сучасних робіт вітчизняних та закордонних авторів з проблематики оптимального розбиття множин, виокремленні задачі, що станом на сьогодні є невивченими. Особливу увагу приділено проблематиці задач в динамічних постановках, тобто таких коли ключові параметри моделі можуть змінюватись з часом відповідно до відомих законів. У наш час, такому класу задач приділено набагато менше уваги, а у деяких напрямках навіть є неоднозначність в розумінні змісту динамічних постановок. Здебільшого під динамічними постановками, в сучасній науковій літературі, розуміється можливість управління такими системами, оцінки впливу на систему зовнішніх збурень та механізмів управління нею. Проте, очевидно, що пропорційна заміна ключових параметрів не вплинула б якісно

на кінцеве рішення, проте динаміка, як правило, є нерівномірною, у зв'язку з чим відбувається якісна зміна у функціонуванні системи загалом.

Наведені вище роздуми ускладнюють дослідження системи, розміщуючи проблему на межі таких наукових напрямів, як теорія оптимального розбиття множин, чисельних методів, теорії біфуркацій та інших, що обумовлені сферою прикладного застосування означеного математичного апарату.

Зауважимо, що на сьогодні існує величезна кількість прикладних задач, що можуть бути розв'язані з використанням постановок, алгоритмів та методів теорії оптимального розбиття множин в динамічних постановках. Серед основних слід виділити такі: розбиття територіальних громад на зони обслуговування муніципальними установами (школи, ЦНАПи тощо), розміщення центрів телефонної комунікації, як стаціонарної так і стільникової, транспортування вантажів, планування територій, зони відповідальності військових формувань, а також в ряді вузькоспеціалізованих задач таких як тривимірний друк прогнозування динаміки інфекційних хвороб тощо.

Очевидно, що в перерахованих вище задачах, існує перелік факторів, що суттєвим чином змінюється з часом, причому рідше вони змінюються випадково, а частіше, закони їх зміни заздалегідь відомі та є результатом спостереження або математичного моделювання. Дослідження на межі означених вище наукових напрямів, вимагає поєднання існуючих методів та алгоритмів та їх вдалої згортки з алгоритмами та методами теорії оптимального розбиття множин, що безперечно є ключовою.

Як було зазначено вище, основним напрямом даного дисертаційного дослідження є розробка алгоритмів та методів розв'язання задач оптимального розбиття в динамічних постановках, тобто в таких постановках при яких ключові параметри системи змінюються з плином часу відповідно до відомих законів. Проте, на нашу думку часткова випадковість таких параметрів не завжди може бути описана у вигляді моделей, проте може бути

вдало розв'язана з використанням теорії нечіткості. В даній дисертаційній роботі буде наведено ряд прикладів використання названої теорії, розроблено методика та алгоритм розв'язання задач розбиття, а результати отримані в чисельних експериментах будуть порівняні з чітким розбиттям та наочно демонструє адекватність результатів задач в нечітких постановках.

Окремо відзначимо, що динамічні постановки задач оптимального розбиття множин знайшли своє змістовне застосування в ряді конкретних прикладних задач. Так оптимальне розбиття множин достатньо рідко застосовувались в задачах військової логістики, проте саме динамічна постановка дозволила адаптувати отримані в роботі результати до розв'язання задачі оптимального розміщення вогневих груп. В прикладній задачі побудови оптимальної траєкторії тривимірного друку задача оптимального розбиття не є основною, проте, зважаючи на те, що основні фактори, що суттєвим чином впливають на якість отриманого рішення, безпосередньо пов'язані з ізольованими точками на зрізі тривимірних деталей, оптимальне розбиття зрізу на окремі частини є важливою підзадачею і попри меншу значимість є більш науково ємною та складною.

В дисертаційній роботі одночасно використовуються, як відомі чисельні методи, що за десятиріччя свого використання довели свою ефективність так і методи теорії біфуркаційного аналізу, методи математичного моделювання, алгоритми та методи теорії оптимального розбиття множин, що зокрема покладені в основу розроблених в роботі методів та алгоритмів, а також дискретні алгоритми розв'язання конкретних прикладних задач.

Результати критичного огляду літератури чітко виокремлюють проблематику оптимального розбиття множин в динамічних постановках, окреслюючи можливі галузі їх застосування та актуальність у наш час, дозволяють вдосконалити існуючий математичний апарат та розв'язати конкретні прикладні задачі, що до сьогодні не мали чіткого математичного розв'язку.

## **Розділ 2. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН ІЗ ЗАДАНИМ ПОЛОЖЕННЯМ ЦЕНТРІВ ПІДМНОЖИН ТА МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ**

Оптимальне розбиття множин є однією з ключових проблем сучасної прикладної математики та теорії оптимізації, яка знаходить широке застосування в логістиці, інформатиці, біоінженерії, моделюванні складних систем та штучному інтелекті. Особливий інтерес становлять динамічні варіанти задач оптимального розбиття, коли умови задачі змінюються у часі, а розбиття має адаптуватися відповідно до динаміки системи. У даному розділі роботи розглянуто модель динамічного розбиття множини з урахуванням змін параметрів системи (зокрема, у термінах нескінченновимірної транспортної задачі, – вартості перевезення одиниці продукції), яка дозволяє побудувати фундаментальні підходи до більш складних задач. Обґрунтовано актуальність застосування нетрадиційних математичних методів для побудови ефективних алгоритмів розв'язання таких задач. Такий підхід є основою для створення нових класів математичних моделей оптимального управління в складних динамічних середовищах.

Особливість розглянутої в цьому розділі задачі полягає у тому, що навіть за найпростіших структурних припущень щодо множини та динаміки зміни її властивостей, оптимальне розбиття може мати складну внутрішню структуру. Це зумовлює необхідність залучення гібридних підходів, що поєднують аналітичні та чисельні методи. У розділі особливу увагу приділено визначенню умов, за яких оптимальне розбиття існує та є стабільним відносно малих збурень параметрів, що моделюють зовнішні впливи. Такий підхід дозволяє не лише теоретично осмислити поведінку систем зі складною структурою, але й забезпечити їхню практичну реалізацію у вигляді алгоритмів з контрольованою складністю. Результати

можуть бути використані для побудови більш складних моделей у задачах кластеризації, управління ресурсами, мережевого планування та адаптивного контролю складних систем.

Динамічні задачі оптимального розбиття множин виникають у контекстах, де об'єкти, які підлягають класифікації або розподілу, змінюють свої властивості з часом, а структура розбиття має оптимізуватись з урахуванням цих змін. У класичному варіанті задача формулюється як знаходження такого розбиття множини на підмножини, яке мінімізує або максимізує задану цільову функцію за умови певних обмежень. У динамічній постановці додатково враховується зміна властивостей елементів множини або зміна самих елементів у часі. Серед найпоширеніших підходів до розв'язання таких задач [22] можна виділити:

1. Математичне програмування: лінійне, цілочислове та нелінійне програмування використовуються для формулювання задач розбиття у вигляді задач оптимізації. Динаміка в таких задачах реалізується через багатокрокові моделі або моделі зі змінним горизонтом планування. Недоліком є складність масштабування таких моделей на великі множини.
2. Методи динамічного програмування: цей клас методів дає змогу поетапно будувати оптимальні розбиття, враховуючи зміну станів у часі. Однак, для задач високої вимірності виникає проблема вимірності, що обмежує їхню практичну застосовність.
3. Графові та мережеві моделі: використовуються для моделювання задач розбиття як задач потоків у мережах або задач розрізу графів. Динаміка вноситься у вигляді змін ваг ребер або топології графа з часом. Такі моделі ефективні для задач маршрутизації, кластеризації або логістики.
4. Евристичні та метаевристичні алгоритми: генетичні алгоритми, мурашині алгоритми тощо активно використовуються у задачах, де точні методи є надто складними з точки зору обчислень. Вони дозволяють шукати наближені розв'язки в умовах складної динаміки та великої кількості

змінних. Окрему нішу займають гібридні алгоритми, які поєднують евристику з локальними точними процедурами оптимізації.

5. Машинне навчання та нейромережеві підходи: у задачах, де доступна історична (статистична) інформація про зміни системи, використовуються моделі навчання з підкріпленням, рекурентні нейронні мережі (RNN, LSTM) та графові нейронні мережі для адаптивного перегрупування та передбачення оптимальних структур розбиття у майбутньому.

Динамічні задачі теорії оптимального розбиття множин охоплюють широкий спектр прикладних і теоретичних підходів до кластеризації, покриття, маршрутизації та ресурсного розподілу в умовах змінного середовища. В Україні ця проблематика досліджується як у прикладному, так і в фундаментальному контекстах.

Першість у створенні базової теорії саме неперервних задач оптимального розбиття множин як окремого розділу теорії нескінченновимірного математичного програмування належить науковій школі під керівництвом професорки Кісельової О.М. В роботах Кісельової О.М. та її учнів запропоновані оригінальні підходи до побудови оптимальних розбиттів у динамічних системах. У роботі [22] наведено формалізовані постановки задач та можливі їх аналітичні розв'язки. Розглянуті модельні задачі в чітких постановках, до яких наведено алгоритми їх розв'язання та результати багаточисельних комп'ютерних експериментів, які наочно ілюструють та підтверджують аналітичні результати.

У роботах [22], [114], [120], [21] здійснено аналіз глобальних постановок задач теорії оптимального розбиття множин в яких сформульовано та доведено ряд фундаментальних теорем, які, зокрема, дозволяють перейти від нескінченновимірних задач до скінченновимірних, перейти від задач з обмеженнями до задач без обмежень тощо.

У роботі [114] розглядаються динамічні задачі, особливості їх постановки, алгоритми розв'язання та наведені чисельні результати комп'ютерного експерименту. В роботі розглядається застосування

означених методів для прикладних задач, обмеження, які при цьому виникають та методи розв'язання задач коли аналітичний розв'язок отримати з тих чи інших причин неможливо.

Особливості розв'язання неперервно-дискретних задач, які обумовлені додатковими умовами розглянуті Кісельовою О.М. та її учнями в роботах [120], [118], [119]. Дискретність задач обумовлено переліком обмежень як на постановку так і на алгоритми розв'язання таких задач. В роботі показано підходи, як узагальнені так і такі, що мають відношення до конкретних задач з використанням яких авторам вдається отримати чисельний, а у деяких випадках і аналітичний розв'язок задач.

Серед основних праць закордонних авторів, що містять фундаментальні результати у питаннях теорії оптимального розбиття множин слід виділити роботи, зокрема, G. Buttazzo, G. Dal Maso, F. Santambrogio, B. Bourdin. Так в роботі G. Buttazzo, G. Dal Maso [71], [73], [74] розглянуто задачі оптимального розбиття з використанням методів та алгоритмів варіаційного числення та теорії міри. Так у названій роботі описано, зокрема, загальні умови існування оптимальних розбиттів у задачах мінімізації енергії функціоналу з урахуванням обмежень на обсяг, міру чи геометрію компонент.

В роботах Caffarelli L., Lin F. [75], Conti M., Terracini S., Verzini G. [82] розглядаються задачі розбиття, які мінімізують (максимізують) суму перших власних значень операторів Лапласа на підмножинах, що виникають у контексті дифузійних моделей, фазових переходів та кластеризації.

Попри те, що теорія оптимального розбиття множин має в своїй основі глибокі аналітичні основи, існує велика кількість праць, автори яких фокусують свою увагу саме на чисельному розв'язанні задач. Зокрема, в роботах Amstutz S., Novotny A. A. [54]-[55] автори використовують для розв'язання методи рівневих множин, топологічну оптимізацію та методи релаксації та регуляризації.

Підкреслимо, що неперервні задачі розбиття природно вкладаються в рамки нескінченновимірного математичного програмування завдяки: варіаційним формулюванням функціоналів цілі; нескінченновимірним обмеженням (цілісність по мірі, граничні умови тощо); використанню методів субдиференціального аналізу та узагальненої спряженості.

Серед сучасних напрямів та відкритих проблем дослідження теорії оптимального розбиття множин слід виділити, зокрема: оптимальні розбиття у випадку нечіткої геометрії або стохастичних обмежень; багатокритеріальні розбиття у задачах екології, біомедицини, кібербезпеки; машинне навчання та оптимізація функціоналів на просторах ймовірнісних мір (Annette Dumas, Filippo Santambrogio), але основним напрямом, що підлягає подальшому дослідженню та вдосконаленню, на нашу думку, є саме дослідження задач оптимального розбиття множин в динамічних постановках, тобто таких де ключові параметри системи змінюються з часом. Важливим є дослідження впливу такої динаміки на оптимальне розбиття та його суміжність з зовнішнім впливом на систему в цілому.

Таким чином, сьогодні теорія неперервних задач оптимального розбиття множин є інтегральною складовою нескінченновимірного математичного програмування. Вона поєднує аналітичну глибину, алгоритмічну складність та широку сферу застосування — від механіки до штучного інтелекту. Актуальним є подальший розвиток як фундаментальної теорії (дослідження питань розв'язуваності, стійкості, збіжності, похибки, складності), так і прикладних засобів її реалізації.

## **2.1 Математична постановка однопродуктової динамічної задачі розбиття множин з заданим положенням центрів підмножин**

Розглянемо неперервну однопродуктову задачу оптимального розбиття множин з  $E_n$  без обмежень із заданим положенням центрів підмножин [21] в динамічній постановці у такому вигляді.

Нехай  $\Omega$  – обмежена замкнена вимірنا за Лебегом множина в  $n$ -вимірному евклідовому просторі  $E_n$ .

Сукупність вимірних за Лебегом підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega \subset E_n$  (серед яких можуть бути і порожні) назвемо можливим розбиттям цієї множини, якщо  $\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j)$ , де  $N > 0$  – задане натуральне число;  $\text{mes}(\cdot)$  – міра Лебега.

Позначимо через  $\sum_{\Omega}^N$  клас всіх можливих розбиттів множини  $\Omega \subset E_n$  на задану кількість  $N$  її вимірних за Лебегом підмножин:

$$\sum_{\Omega}^N = \{\bar{\omega} \equiv \{\Omega_1, \dots, \Omega_N\} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j\}$$

Далі, через  $\tau_1, \dots, \tau_N$  позначимо сукупність деяких еталонних точок для підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, які назвемо центрами цих підмножин:  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , і будемо вважати, що координати усіх центрів задані.

**Постановка задачі.** Знайти розбиття  $\bar{\omega} = \{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$  множини  $\Omega \subset E_n$  і векторну функцію  $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$ , визначену м.в. для  $x \in \Omega$  при заданому фіксованому наборі центрів  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та всіх  $t \in [0, T]$ , які забезпечують

$$\inf_{\bar{\omega} \in \sum_{\Omega}^N : c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, c(\cdot, \cdot, \cdot)) \quad (2.1)$$

де

$$F(\bar{\omega}, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad (2.2)$$

за наявності умов

$$\begin{aligned} \frac{dc_i(x, \tau_i, t)}{dt} &= \sum_{j=1}^N A_{ij} \cdot f_j(c_j(x, \tau_j, t)), 0 \leq t \leq T \\ c_i(x, \tau_i, t_0) &= c_{0i}(x, \tau_i), i = 1, \dots, N \end{aligned} \quad (2.3)$$

м.в. для  $x \in \Omega$ , при фіксованих  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , та умови замкненості системи

$$\sum_{i=1}^N A_{ij} = 1, j = 1, \dots, N. \quad (2.4)$$

Тут  $c_i(x, \tau_i, t), i = 1, \dots, N$ , – шукані дійсні функції, визначені на  $\Omega \times \Omega \times [0, T]$ , які при будь-якому фіксованому  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i$  неперервно диференційовані за аргументом  $t$  на відрізку  $[0, T]$  м.в. для  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ , обмежені та вимірні за аргументом  $x$  на  $\Omega$  для всіх  $t \in [0, T]$ ;  $m(x, \tau_i), c_{0i}(x, \tau_i)$  – задані дійсні функції, визначені на  $\Omega \times \Omega$ , обмежені та вимірні за аргументом  $x \in \Omega$  при будь-якому фіксованому  $\tau_i \in \Omega_i$  для всіх  $i = 1, \dots, N$  (зокрема,  $m(x, \tau_i)$  може відігравати роль метрики на  $\Omega \times \Omega$ );  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , – задані дійсні функції, неперервні та ліпшицеві в області свого визначення;  $\rho(x)$  – задана невід’ємна функція, обмежена та вимірна на  $\Omega$ ;  $a_i, i = 1, \dots, N$ , – задані, як правило, невід’ємні числа;  $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$ , – задані числові параметри;  $T > 0$  та  $t_0 \in [0, T]$  задані.

Тут та в подальшому інтеграли розуміються в сенсі Лебега. Будемо вважати, що міра множини межових точок підмножин  $\Omega_1, \dots, \Omega_N$  дорівнює нулю.

Пару  $(\bar{w}^*, c^*(x, \tau, t))$ , що доставляє функціоналу (2.2) мінімальне значення на множині  $\sum_{\Omega}^N \times L_2^N(\Omega \times \Omega \times [0, T])$  за умов (2.3), (2.4), назвемо оптимальним розв’язком задачі (2.1)–(2.4). При цьому розбиття  $\bar{w}^* = \{\Omega_1^*, \dots, \Omega_N^*\} \in \sum_{\Omega}^N$  назвемо оптимальним розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин, а векторну функцію  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)) \in L_2^N(\Omega \times \Omega \times [0, T])$  – оптимальною фазовою траєкторією динамічної системи в задачі (2.1)–(2.4).

З предметної точки зору, незалежна змінна  $t \in [0, T]$  у наведеній математичній постановці динамічної задачі оптимального розбиття може відігравати роль часової змінної, а  $T > 0$  та  $t_0 \in [0, T]$  – задані кінцевий та початковий моменти часу у досліджуваному динамічному процесі відповідно. Так, функції  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , у диференціальних зв’язках (2.3), що відображають динаміку цін на транспортування, можуть мати

різний вигляд (2.5)-(2.8), в залежності від предметного змісту, який в неї вкладається, наприклад:

- модель інфляції/дефляції відображає схильність цін до постійного (експоненціального) зростання/спадання

$$\begin{aligned} f_i(c_i(x, \tau_i, t)) &= d_i \cdot c_i(x, \tau_i, t), \\ x \in \Omega, \tau_i &= (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = \overline{1, N}, 0 \leq t \leq T \end{aligned} \quad (2.5)$$

- модель односторонньої цінової регуляції, коли, наприклад, державою встановлені певні обмеження на ціну перевезення (мінімальна/максимальна).

$$\begin{aligned} f_i(c_i(x, \tau_i, t)) &= d_i \cdot c_i(x, \tau_i, t) \cdot \left(1 - \frac{c_i(x, \tau_i, t)}{K_1}\right), \\ x \in \Omega, \tau_i &= (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = \overline{1, N}, K_1 > 0, 0 \leq t \leq T \end{aligned} \quad (2.6)$$

- модель двосторонньої цінової регуляції, коли, наприклад, державою встановлені певний діапазон на ціну перевезення  $K_1 \leq c_i(x, \tau_i, t) \leq K_2$ .

$$\begin{aligned} f_i(c_i(x, \tau_i, t)) &= d_i \cdot (c_i(x, \tau_i, t) - K_2) \cdot \left(1 - \frac{c_i(x, \tau_i, t)}{K_1}\right), \\ x \in \Omega, \tau_i &= (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = \overline{1, N}, K_1, K_2 > 0, 0 \leq t \leq T \end{aligned} \quad (2.7)$$

- модель дотаційної односторонньої цінової регуляції, коли, наприклад, державою встановлені граничні ціни, що забезпечується дотаціями

$$\begin{aligned} f_i(c_i(x, \tau_i, t)) &= d_i \cdot c_i(x, \tau_i, t) \cdot \left(1 - \frac{c_i(x, \tau_i, t)}{K_1}\right) + Z, \\ x \in \Omega, \tau_i &= (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = \overline{1, N}, K_1, Z > 0, 0 \leq t \leq T \end{aligned} \quad (2.8)$$

У формулах (2.5)–(2.8) позначення слід розуміти у такому змісті:

$d_i, i = 1, \dots, N$  - задані дійсні коефіцієнти, що відображають схильність вартості транспортування одиниці продукції  $i$ -го центру споживачу до зростання/спадання;

$K_1, K_2$  - задані невід'ємні коефіцієнти, що відображають порогові значення ціни (максимальне та мінімальне відповідно) на транспортування одиниці продукції для всіх  $\tau_i, i = 1, \dots, N$ ;

$Z$  - заданий невід'ємний коефіцієнт, що відображає обсяг дотацій, які здійснюються кожному з центрів постачання за одиницю часу.

В якості базової в цьому розділі та для адекватності порівнянь та аналізу далі в роботі буде використана модель інфляції/дефляції (2.5).

## 2.2 Обґрунтування методу розв'язання

Нехай  $\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$  – деяке можливе розбиття вихідної замкненої обмеженої вимірної множини  $\Omega \subset E_n$  на  $N$  її вимірних за Лебегом неперетинних підмножин.

Введемо у розгляд на множині  $\Omega \subset E_n$  характеристичні функції  $\lambda_1(x), \dots, \lambda_N(x)$  підмножин  $\Omega_1, \dots, \Omega_N$  відповідно:

$$\lambda_i(x) = \begin{cases} 1, & x \in \Omega_i, \\ 0, & x \in \Omega \setminus \Omega_i, \end{cases} \quad i = 1, \dots, N \quad (2.9)$$

і переформулюємо задачу (2.1)–(2.4) у термінах характеристичних функцій підмножин у більш зручному вигляді. Отже потрібно знайти векторні функції  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x)), x \in \Omega$  і  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)), x \in \Omega$ , при фіксованих  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та  $t \in [0, T]$ , які забезпечують

$$\inf_{\lambda(\cdot) \in \Gamma; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) \quad (2.10)$$

де функціонал  $J(\lambda(\cdot), c(\cdot, \cdot, \cdot))$  має вигляд

$$J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega} (c_i(x, \tau_i, t) m(x, \tau_i) + a_i) \rho(x) \lambda_i dx dt, \quad (2.11)$$

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega; \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \} \quad (2.12)$$

за умов (2.3)–(2.4). Очевидно, що для функціонала  $F(\bar{\omega}, c(\cdot, \cdot, \cdot))$  виду (2.2) і функціонала  $J(\lambda(\cdot), c(\cdot, \cdot, \cdot))$  виду (2.11) виконується рівність  $J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) = F(\bar{\omega}, c(\cdot, \cdot, \cdot))$ .

Далі від задачі (2.10)–(2.12) за умов (2.3)–(2.4) з булевими значеннями функцій  $\lambda_i(\cdot), i = 1, \dots, N$ , перейдемо до відповідної задачі зі значеннями  $\lambda_i(\cdot), i = 1, \dots, N$ , що змінюються на відрізьку  $[0, 1]$ .

**Задача.** Знайти векторні функції  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x))$ ,  $x \in \Omega$  і  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t))$ ,  $x \in \Omega$  при фіксованих  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та  $t \in [0, T]$ , які забезпечують

$$\inf_{\lambda(\cdot) \in \Gamma; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) \quad (2.13)$$

де функціонал  $J(\lambda(\cdot), c(\cdot, \cdot, \cdot))$  має вигляд

$$J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) = \int_0^T \int_{\Omega} \sum_{i=1}^N (c_i(x, \tau_i, t) m(x, \tau_i) + a_i) \rho(x) \lambda_i(x) dx dt, \quad (2.14)$$

$$\Gamma = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega; \\ 0 \leq \lambda_i(x) \leq 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \} \quad (2.15)$$

за умов (2.3)-(2.4).

Перша компонента  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x))$  оптимального розв'язку такої задачі м.в. для  $x \in \Omega$  і для кожного  $i = 1, \dots, N$  має вигляд

$$\lambda_i^*(x) = \begin{cases} 1, \text{ якщо виконується умова (2.17)} \\ 0, \text{ в іншому випадку} \end{cases} \quad (2.16)$$

$$\int_0^T (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dt = \min_{k=1, N} \left( \int_0^T (c_k(x, \tau_k, t) \cdot m(x, \tau_k) + a_k) \rho(x) dt \right) \quad (2.17)$$

при заданих  $\tau_i \in \Omega, i = 1, \dots, N$

### 2.3 Алгоритм розв'язання задачі однопродуктової динамічної задачі оптимального розбиття множин з фіксованими центрами підмножин

Крок 1: Множину  $\Omega$  включаємо в  $n$ - вимірний паралелепіпед  $\Pi$ , сторони якого паралельні осям декартової системи координат. Паралелепіпед  $\Pi$  покриваємо прямокутною сіткою.

Крок 2: Задаємо значення функцій  $c_{0i}(x, \tau_i), i = 1, \dots, N$ , з початкових умов (2.3) в вузлах сітки.

Крок 3: Покриваємо відрізок  $[0, T]$  сіткою з кроком  $h_t$ .

Крок 4: Для кожного вузла сітки та для кожного  $\tau_i, i = 1, \dots, N$  розв'язуємо задачу Коші для системи ЗДР (2.3) на часовому відрізку  $[0, T]$  та знаходимо функції  $c_i^*(x, \tau_i, t), i = 1, \dots, N$

Крок 5: Визначаємо характеристичну функцію  $\lambda^*(x)$  у вузлах сітки з пункту 1 за формулами (2.16), (2.17) при знайдених  $c_i^*(x, \tau_i, t), i = 1, \dots, N$ .

Крок 6: Знаходимо мінімальне значення цільового функціонала (2.14) при знайдених  $\lambda^*(x)$  та  $c_i^*(x, \tau_i, t), i = 1, \dots, N$ .

Алгоритм розв'язання неперервної однопродуктової динамічної задачі оптимального розбиття множин з  $E_n$  без обмежень з заданим положенням центрів підмножин описаний.

## 2.4 Розв'язання модельних задач та оцінка чисельних результатів

Описана вище задача з моделлю (2.5) у якості базової була розв'язана для різних модельних задач, що характеризуються різними початковими умовами системи диференціальних рівнянь (2.3) та іншими параметрами моделі. Модельні задачі є нескінченновимірними задачами розбиття заданої області з фіксованими центрами підприємств, що виробляють однорідну продукцію для розподілення в цій області із заданою щільністю споживачів.

### Модельна задача 2.1.

Споживачі деякої однорідної продукції, що виробляється двома підприємствами, неперервно розподілені в області  $\Omega = \{x = (x^{(1)}, x^{(2)}) : 0 \leq x^{(1)} \leq 1; 0 \leq x^{(2)} \leq 1\}$ , задані координати розміщення підприємств  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$ , для підприємств задана система (2.18), що описує зміну ціни на транспортування одиниці продукції, а саме:

$$\begin{cases} \frac{\partial c_1(x, \tau_1, t)}{\partial t} = A_{11} \cdot d_1 \cdot c_1(x, \tau_1, t) + (1 - A_{22}) \cdot d_2 \cdot c_2(x, \tau_2, t) \\ \frac{\partial c_2(x, \tau_2, t)}{\partial t} = (1 - A_{11}) \cdot d_1 \cdot c_1(x, \tau_1, t) + A_{22} \cdot d_2 \cdot c_2(x, \tau_2, t) \end{cases}, \quad 0 \leq t \leq T, \quad (2.18)$$

з початковими умовами

$$c_1^0 = 2, \quad c_2^0 = 1 \quad (2.19)$$

В цій та наступних модельних задачах у якості  $m(x, \tau_i)$  використана евклідова метрика.

Розв'язок задачі (2.18)-(2.19) описує вартість транспортування одиниці продукції з  $i$ -го підприємства до споживача з координатами  $(x, y)$ . Відомий попит  $\rho(x, y)$  на продукцію для кожного пункту споживання з координатами  $(x, y)$ . Для простоти вважається  $\rho(x, y) \equiv 1, \forall x \in \Omega$ .

Множину споживачів потрібно розбити на зони обслуговування  $\Omega_i$  споживачів  $i$ -им пунктом виробництва так, щоб

$$\bigcup_{i=1}^2 \Omega_i = \Omega, \quad \text{mes}(\Omega_i \cap \Omega_k)_{i \neq k} = 0, \quad i, k = 1, 2 \quad (2.20)$$

Для випадку  $N = 2$  формулу (2.20) можна було б записати в більш простому вигляді, проте у наступних модельних задачах  $N$  прийматиме значення 3, 4, 5, а відповідні умови залишатимуться у наведеному аналітичному вигляді (2.20). Зазначимо, що потужність  $i$ -го підприємства  $i = 1, 2$  визначається сумарним попитом споживачів, що належать  $\Omega_i, i = 1, 2$ .

Необхідно розбити множину споживачів  $\Omega$  так, щоб мінімізувати функціонал (2.1), (2.2) сумарних витрат на виробництво продукції та її доставку до споживача при (2.5) та за умов (2.3), (2.4) при  $N = 2$ .

Для розв'язання сформульованої задачі область  $\Omega$  була покрита прямокутною сіткою з вузлами  $(i, j), i = 1, \dots, 21, j = 1, \dots, 21$ . Було отримано чисельний розв'язок системи (2.18), для якого був використаний метод Рунге-Кути 4-го порядку. Пошук розв'язку системи (2.18) було здійснено при  $h_i = 0,001$ , а для отриманих точок фазової траєкторії при  $T = [0,5; 1; 1,5; 2; 2,5; 3; 5; 10]$  було розраховано вартість транспортування від кожного вузла  $(i, j)$  області  $\Omega$  до центрів  $\tau_1, \tau_2$ . Виходячи з критерію мінімальної вартості вузол  $(i, j)$  був віднесений до областей  $\Omega_1$  та  $\Omega_2$  відповідно. Результати чисельного експерименту наведені на рис.2.1 (а-з).

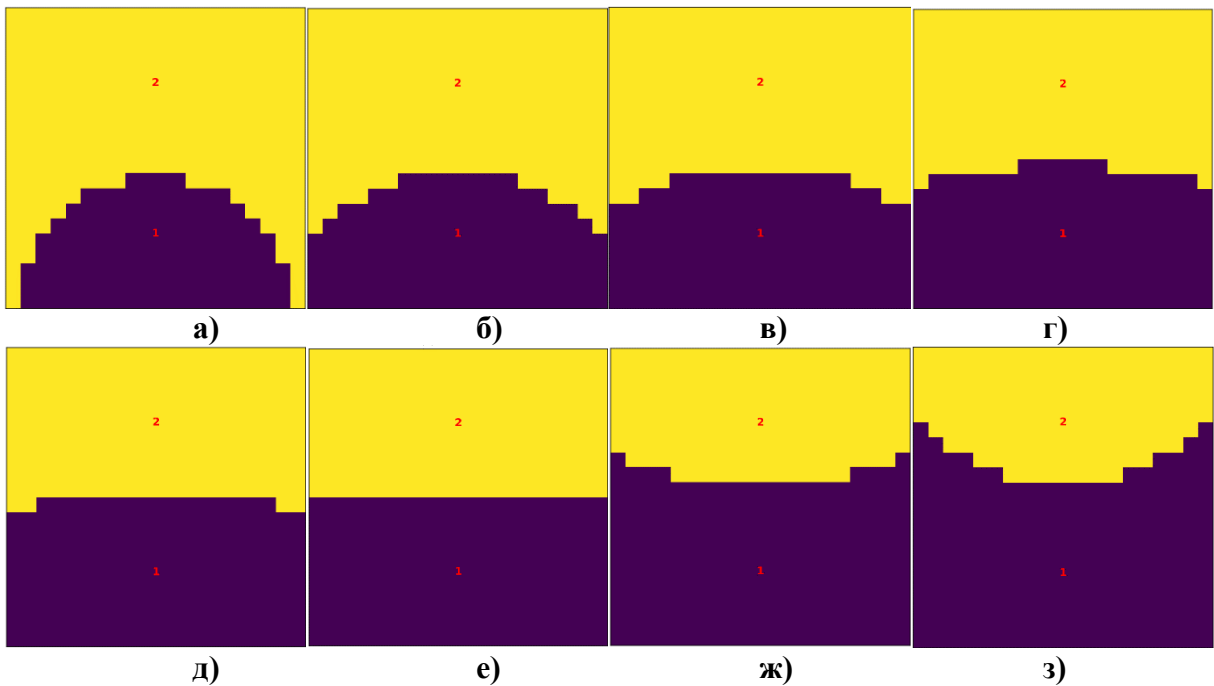


Рис.2.1 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$ ,  $\Delta x = \Delta y = 0,05$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ ,  $F = 0,245$ , б)  $T = 1$ ,  $F = 0,597$ , в)  $T = 1,5$ ,  $F = 1,079$ , г)  $T = 2$ ,  $F = 1,727$ , д)  $T = 2,5$ ,  $F = 2,59$ , е)  $T = 3$ ,  $F = 3,728$ , ж)  $T = 5$ ,  $F = 13,105$ , з)  $T = 10$ ,  $F = 182,925$

Отримані результати чисельного експерименту є достатньо прогнозованими та базуються, зокрема, на співвідношенні параметрів математичної моделі  $d_1, d_2$ . Так на Рис.2.1-а) площа підмножини  $\Omega_2$  майже у два рази більша за  $\Omega_1$ , що зумовлено відповідним початковим співвідношенням  $c_1^0 / c_2^0$ . Тобто виходячи з того, що вартість транспортування для першого виробника двічі більше за вартість транспортування для другого виробника, кількість споживачів відповідними, нерівномірним чином розподілено між центрами  $\tau_1$ ,  $\tau_2$ . Лінія розмежування представляє собою параболу. Зауважимо, що за відсутності в математичній моделі параметра  $a_i$  межі між підмножинами мали б форму прямої, проте саме наявність в математичній моделі параметра  $a_i$ , що відображає статичні витрати, а межа між підмножинами має більш складену форму. При  $t > 0$  значення  $c_1, c_2$  змінюються, проте не пропорційно. Зважаючи на те, що співвідношення  $d_2 / d_1 = 0,6$  вартість транспортування для другого виробника збільшується швидше та при  $T = 3$  досягається рівність  $c_1 \approx c_2$ . Відповідна ситуація

зображена на рис.2.1-е) де межа між підмножинами при оптимальному розбитті має форму прямої. Зауважимо, що рівні значення вартості транспортування  $c_1 \approx c_2$  не завжди провокують межу між підмножинами оптимального розбиття у формі прямої, проте у наведеному прикладі  $a_1 = a_2$ , а при одночасній рівності статичних та динамічних витрат для двох центрів межею є саме пряма лінія. Стан системи зображений на рис.2.1е) можна вважати біфуркаційним, адже при граничних (рівних) значеннях динамічних витрат на транспортування у двох центрів, топологія оптимального розбиття набуває якісних змін. Розбиття рис.2.1-ж)-з) демонструють аналогічне розбиття при якій кількість споживачів другого виробника зменшується. Загалом отримані результати наочно демонструють оптимальне розбиття, відповідають співвідношенням ключових параметрів, що в цілому визначають особливості динаміки в математичній моделі.

### **Модельна задача 2.2.**

Вхідні данні для модельної задачі 2.2 є такими ж, як і в модельній задачі 2.1, за виключенням координат двох центрів, які приймають значення  $\tau_1(0,1;0,2)$ ,  $\tau_2(0,8;0,6)$ . Такий приклад демонструє рівень впливу симетричності розташування центрів відносно межі області  $\Omega$  на результати оптимального розбиття. Результати розбиття рис.2.2 а-з) мають схожу структуру з результатами представленими в модельній задачі 2.1, за тією відмінністю, що віддаленість центру  $\tau_2$  від границі області впливає на нелінійність межі між областями  $\Omega_1, \Omega_2$  та на значення цільового функціоналу. Хочемо акцентувати увагу, що при  $T = 10$ , вартість транспортування одиниці продукції для центру  $\tau_2$  суттєво збільшується. В результаті розмір області  $\Omega_1$  збільшується, та суттєвого збільшення набуває і значення цільового функціоналу рис.2.2-з).

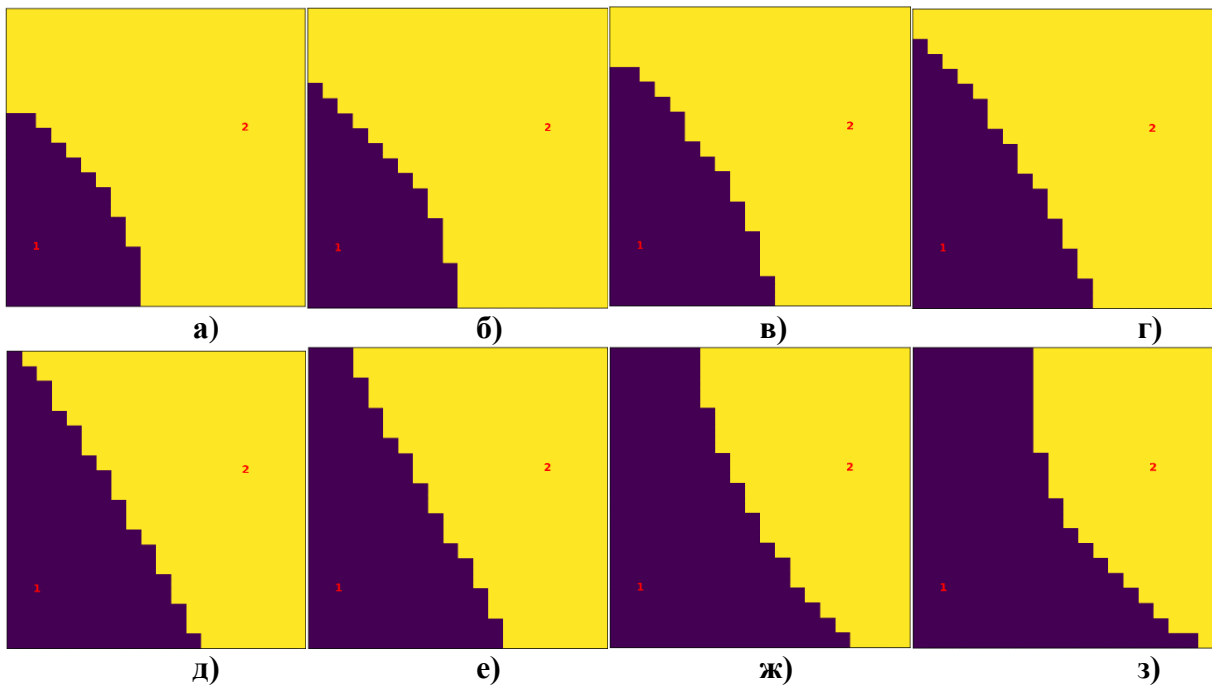


Рис.2.2 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,1;0,2)$ ,  $\tau_2(0,8;0,6)$ ,  $\Delta x = \Delta y = 0,05$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ ,  $F = 0,255$ , б)  $T = 1$ ,  $F = 0,638$ , в)  $T = 1,5$ ,  $F = 1,18$ , г)  $T = 2$ ,  $F = 1,924$ , д)  $T = 2,5$ ,  $F = 2,922$ , е)  $T = 3$ ,  $F = 4,246$ , ж)  $T = 5$ ,  $F = 15,158$ , з)  $T = 10$ ,  $F = 213,187$

Попри те, що вихідні данні для модельної задачі 2.2 у більшості своїх значень відповідають вхідним даним модельної задачі 2.1, координати розміщення центрів, які принципово відрізняються від рівномірного розміщення в модельній задачі 2.1, суттєвим чином впливають на результати оптимального розбиття. Так ми бачимо, що центри зміщено до того ж вони розміщені достатньо близько до межі області  $\Omega$ . Менша посадова ціна на транспортування для центру виробництва  $\tau_2$  сприяє нерівномірному розподілу споживачів між двома центрами. Але на відміну від модельної задачі 2.1 в якій співвідношення початкових витрат  $c_1^0 / c_2^0 = 0,5$ , в модельній задачі 2.2 близькість центрів до меж області суттєвим чином впливає як на площі підмножин, що є результатом оптимального розбиття, які в свою чергу є відображенням відповідної кількості споживачів двох центрів виробництва, так і на форму межі між підмножинами  $\Omega_1$  та  $\Omega_2$ . Виходячи з дискретності обраного поділу часового проміжку, поміж восьми чисельних експериментів наведених на рис.2.2 ми бачимо, що чисельного експерименту, що відображав би біфуркаційне співвідношення динамічних витрат відсутнє.

Очевидно, що біфуркаційного стану система набуває на проміжку  $t \in (3;5)$ . Саме на цьому проміжку часової змінної досягається рівність  $c_1 = c_2$  при якому межею між підмножинами є пряма лінія. У восьми наведених чисельних експериментах межею між підмножинами  $\Omega_1, \Omega_2$  отриманими в результаті оптимального розбиття множини  $\Omega$  є гіпербола. На проміжку  $t \in [0,5;3]$  більшу площу має підмножина  $\Omega_2$ , а для  $t \in [5;10]$  більше споживачів віднесені до підмножини  $\Omega_1$  відповідно. Наведені результати цілком відповідають співвідношенню ключових параметрів, ілюструють зміну оптимального розбиття з плином часу та при відповідній зміні співвідношення між вартістю транспортування одиниці продукції для різних виробників. Такі результати дозволяють наочно оцінити застосування саме динамічної постановки задачі та вплив зміни співвідношення ціни на транспортування одиниці продукції, як на кількісні так і на якісні результати оптимального розбиття.

### **Модельна задача 2.3.**

Для модельної задачі 2.3 та наступних більшість вхідних даних будуть обрані аналогічно вхідним даним модельних задач 2.1 та 2.2. Зазначимо, що перелік вхідних параметрів для реалізації чисельного експерименту може бути обрано достатньо різноманітним, проте, відмінні вхідні параметри не дозволяють достатньо глибоко провести порівняльний аналіз впливу розміщення центрів постачання, початкової вартості транспортування одиниці продукції та швидкості її зростання на оптимальне розбиття. Відмінністю модельної задачі 2.3 є координати центрів  $\tau_1, \tau_2$ , що обрані розташованими достатньо близько один до одного  $\tau_1(0,7;0,4)$ ,  $\tau_2(0,75;0,35)$ , а точки  $x$ , що наближені до межі області  $\Omega$ , прогнозовано, в результаті оптимального розбиття, віднесено до найближчого центру. Така топологія розміщення центрів постачання суттєвим чином впливає на результат оптимального розбиття, ускладнюючи його. Крім цього значення цільового

функціоналу оптимального розбиття при  $T = 10$ , аналогічно випадку представленого в модельній задачі 2.2, стрімко зростає. Такий ефект не свідчить про некоректність розбиття та не є підставою вважати розбиття неоптимальним, а зумовлений стрімким збільшенням вартості транспортування для обох центрів постачання, що в свою чергу впливає стрімке зростання значення цільового функціоналу. Модель інфляції/дефляції покладена в основу математичної моделі оптимального розбиття обрана з початковими умовами наближеними до нуля. Легко бачити, що особлива точка початку координат є нестійкою, і фазова траєкторія, що прямує до нескінченності зумовлює ефект суттєвого зростання загальної вартості перевезення і відповідного значення цільового функціоналу. З предметної точки зору такий приклад ілюструє ситуацію інфляційного збільшення вартості транспортування протягом багатьох років, або ситуацію гіперінфляції, коли вартість транспортування, що зокрема залежить від вартості паливно-мастильних матеріалів, стрімко зростає.

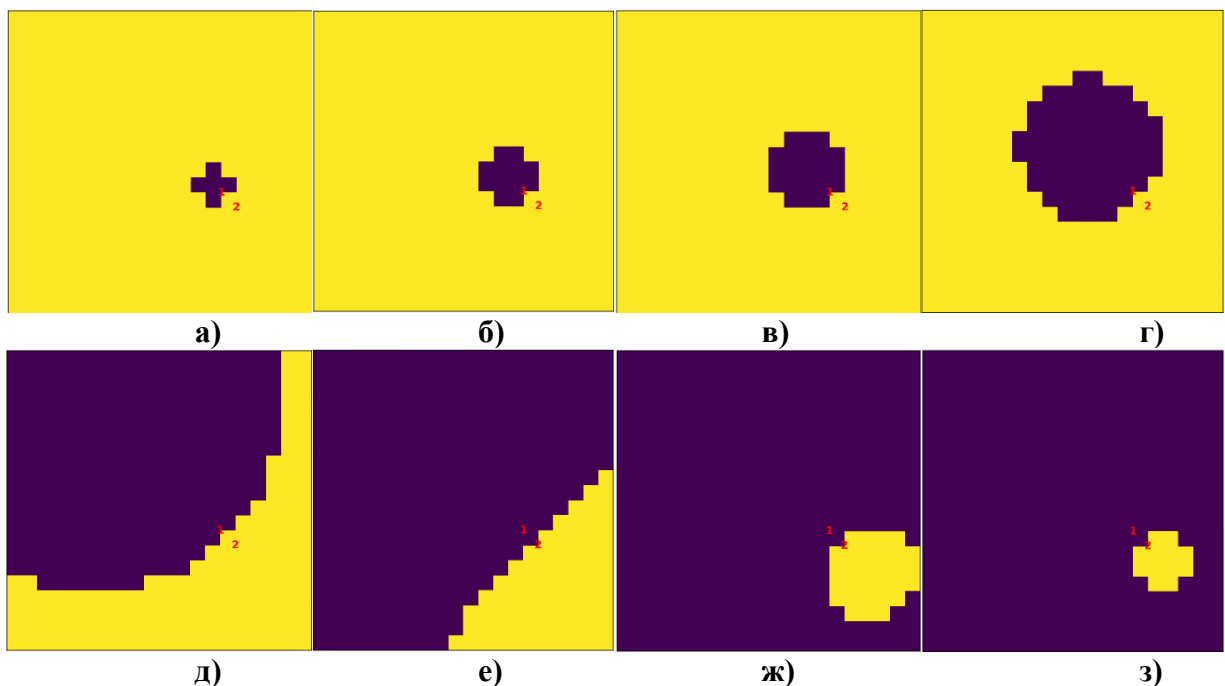


Рис.2.3 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,7;0,4)$ ,  $\tau_2(0,75;0,35)$ ,  $\Delta x = \Delta y = 0,05$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ ,  $F = 0,299$ , б)  $T = 1$ ,  $F = 0,769$ , в)  $T = 1,5$ ,  $F = 1,464$ , г)  $T = 2$ ,  $F = 2,449$ , д)  $T = 2,5$ ,  $F = 3,708$ , е)  $T = 3$ ,  $F = 5,207$ , ж)  $T = 5$ ,  $F = 16,849$ , з)  $T = 10$ ,  $F = 219,928$

З експериментальної точки зору випадок наведений на рис.2.3 є набагато цікавішим ніж наведені у модельних задачах 2.1 та 2.2. Зауважимо, що насамперед координати центрів обрані такими, що в порівнянні з розмірами області  $\Omega$  відстань між центрами є достатньо малою. Зважаючи на те, що інші параметри математичної моделі обрані аналогічно модельним задачам 2.1-2.2 кількість споживачів центру  $\tau_1$  при  $t \in [0,5;2]$  є достатньо малою в порівнянні з кількістю споживачів центру  $\tau_2$ . Розміщення пари центрів на достатній відстані від меж області  $\Omega$  призводить до формування ізольованої підмножини  $\Omega_1$  форми кола. З плином часу та відповідним зменшенням значення співвідношення  $c_1/c_2$  розмір кола збільшується відповідно до збільшення кількості споживачів центру  $\tau_1$ . При  $t \in (2;2,5)$  співвідношення площ підмножин  $\Omega_1/\Omega_2$  сягає біуркаційного значення, а крайові умови суттєвим чином впливають на форму межі між названими підмножинами перетворюючи її з кола на гіперболу. При  $t \rightarrow 10$  відбуваються аналогічні зміни тільки для іншого центру, а саме  $\tau_2$ . Такі результати наочно демонструють, що суттєвим чином оптимальне розбиття, при зміні співвідношення вартості транспортування, залежить не тільки від поточного співвідношення вартості транспортування для різних центрів, але і від крайових умов, зокрема, форми області  $\Omega$  та щільності розподілу споживачів на ній. Для даної модельної задачі, як було сказано вище,  $\rho = 1$ . Проте важливо підкреслити, що у разі нерівномірного розподілу споживачів на області  $\Omega$  форма межі між підмножинами буде суттєво відрізнятися від тривіальних. Як бачимо результати чисельного експерименту наведеного в модельній задачі 2.3 достатньо адекватно ілюструють зміни в оптимальному розбитті при зміні співвідношення вартості транспортування для двох центрів виробництва.

#### **Модельна задача 2.4.**

У наведеній модельній задачі 2.4 кількість центрів дорівнює трьом, а їх взаємне розташування характерне розміщенням на відрізку рис.2.4-а) з

рівним відстанями між першою та другою парою центрів  $\tau_1(0,2;0,5)$ ,  $\tau_2(0,5;0,7)$ ,  $\tau_3(0,85;0,85)$ . Крім цього крок прямокутної сітки прийнятий 0,01, що, зокрема, позитивно впливає на точність отриманого розв'язку. Параметри  $d_1=1$ ,  $d_2=2$ ,  $d_3=3$ , що на відміну від попередніх модельних задач зменшує відносну різницю між швидкістю зміни цін на транспортування одиниці продукції для трьох центрів, але при цьому між першим та третім центром швидкість зміни вартості вже є достатньо суттєвою та дорівнює 3. Така згортка параметрів спрямовує фазову траєкторію з більшою швидкістю від стаціонарної точки початку координат та вже при  $T=2$  можна бачити суттєву перевагу площі  $\Omega_1$  в порівнянні з  $\Omega_2$  та  $\Omega_3$ .

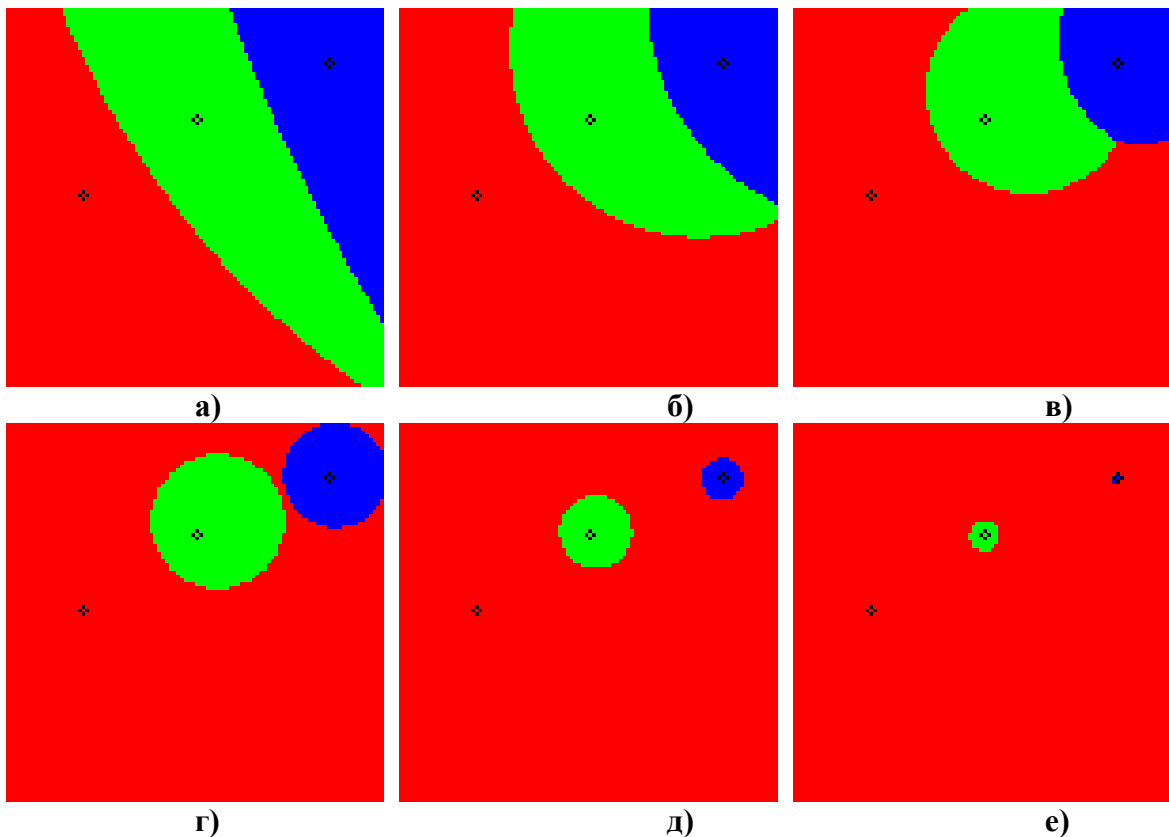


Рис.2.4 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною

1, на три підмножини з центрами  $\tau_1(0,2;0,5)$ ,  $\tau_2(0,5;0,7)$ ,  $\tau_3(0,85;0,85)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$  та а)  $T = 0,5$ ,  $F = 0,308$ , б)  $T = 1$ ,  $F = 0,919$ , в)  $T = 1,5$ ,  $F = 1,985$ , г)  $T = 2$ ,  $F = 3,723$ , д)  $T = 3$ ,  $F = 10,038$ , е)  $T = 5$ ,  $F = 48,889$

Наведений в даній модельній задачі випадок є більш цікавим з обчислювальної точки зору. Підкреслимо, що координати розташування центрів обрані таким чином, що всі вони розміщені на одному відрізку і дві

пари є рівновіддаленими один від іншого. Такий формат розміщення центрів був обумовлений бажанням зрозуміти попарну взаємодію між центрами при зміні співвідношення цін на транспортування одиниці продукції. Початкове співвідношення цін на транспортування можна описати подвійною нерівністю  $c_1^0 < c_2^0 < c_3^0$ , аналогічна нерівність має місце і для параметрів  $d_1 < d_2 < d_3$ . З плином часу другий та третій центр втрачають споживачів. При цьому крок  $h_t$  обрано таким чином, що ми можемо бачити поступову локалізацію підмножин навколо центрів  $\tau_2, \tau_3$  у формі еліпсів. Взаємне розташування центрів  $\tau_1$  та  $\tau_2$  спричиняє так назване зміщення центру  $\tau_2$  в підмножині  $\Omega_2$  в напрямку від центру при  $t \in [1,5;2]$ . При цьому для випадків  $t \in (3;5)$  центри постачання  $\tau_2$  та  $\tau_3$  геометрично наближені до центрів підмножин  $\Omega_2$  та  $\Omega_3$  відповідно. При цьому слід відмітити, що результати модельної задачі 2.4 достатньо наочно відображають загальнодинамічні тенденції системи та відповідають співвідношенню її ключових параметрів. Така відповідність дозволяє, зокрема, розуміти загальні правила функціонування системи, результати штучного (зовнішнього) втручання в систему та можливі наслідки спроб штучного управління нею. При достатньо великій різниці між вартістю транспортування одиниці продукції для різних центрів (див. наприклад рис.2.4-е)) оптимальне розбиття є тривіальним, проте для наукового дослідження та порівняння цікавими є саме граничні значення параметрів при яких в системі відбуваються якісні зміни. Крім біфуркаційних значень, при яких площа пари підмножин чи всіх (біфуркація ковимірності  $>1$ ) підмножин є рівною, з прикладної точки зору цікавими є випадки коли суттєвим чином змінюється форма підмножин отриманих в результаті оптимального розбиття, та місце розташування їх центрів. У випадку модельної задачі 2.4 вдало ілюструється не тільки взаємний вплив пар центрів виробництва  $\tau_1, \tau_2$  та  $\tau_1, \tau_3$ , але й пари центрів  $\tau_2, \tau_3$ . Знаходячись в не вигідному становищі в порівнянні з центром  $\tau_1$ , інші два центри, що мають більш високу ціну на транспортування починають конкурувати між собою.

Такий ефект ми бачимо на рис.2.4-в)-е). Співвідношення цін на транспортування між центрами  $\tau_2$  та  $\tau_3$  визначає кількість споживачів, що при рівномірному розподілі  $\rho = 1$  відповідає площі підмножин  $\Omega_2, \Omega_3$  відповідно.

### Модельна задача 2.5.

Відмінність модельної задачі 2.5, полягає у взаємному розташуванні центрів постачання продукції  $\tau_1(0,1;0,25)$ ,  $\tau_2(0,9;0,7)$ ,  $\tau_3(0,5;0,5)$ . Значення швидкості зміни вартості транспортування при цьому не змінено, що демонструє зміщення межі областей  $\Omega_1, \Omega_2, \Omega_3$  в сторону центрів  $\tau_2$  та  $\tau_3$ , як і у випадку модельної задачі 2.4. Значення цільового функціоналу при цьому з плином часу не набуває суттєвих відмінностей, в порівнянні з наведеними у попередній модельній задачі та при  $T = 5$ ,  $F = 59,269$  на відміну від контрольного значення модельної задачі 2.4 в якій при  $T = 5$ ,  $F = 48,889$ .

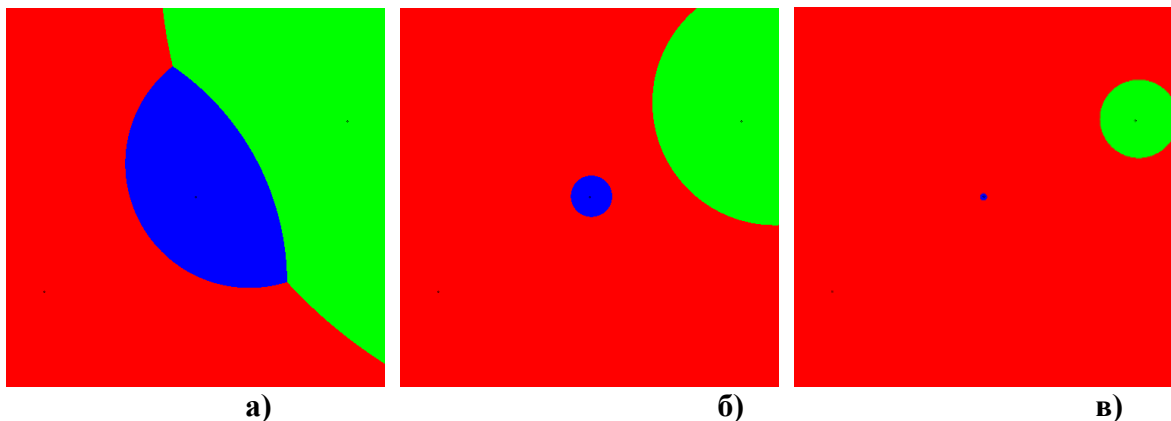


Рис.2.5 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на три підмножини з центрами  $\tau_1(0,1;0,25)$ ,  $\tau_2(0,9;0,7)$ ,  $\tau_3(0,5;0,5)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$  та а)  $T = 1$ ,  $F = 0,980$ , б)  $T = 2,5$ ,  $F = 7,113$ , в)  $T = 5$ ,  $F = 59,269$

Наведена модельна задача цілком відтворює випадок наведений у модельній задачі 2.4, за тією відмінністю, що центри розташовані не на відрізку, тобто між парами  $\tau_1, \tau_3$  та  $\tau_3, \tau_2$  відсутня пряма конкуренція. Крім того центри  $\tau_1$  та  $\tau_2$  розміщені ближче до меж області  $\Omega$ , що в свою чергу більше обумовлює картину оптимального розбиття крайовими умовами та формою області  $\Omega$ . Відповідно до початкових значень вартості транспортування одиниці продукції, центр постачання  $\tau_3$  знаходиться у

програшному стані та кількість споживачів, що отримує товари з нього суттєво менше ніж від центрів  $\tau_1$  та  $\tau_2$ . Проте швидкість зміни вартості для центрів не відповідає розподілу початкової цін на транспортування та найбільшим є для центру  $\tau_2$ . Таке співвідношення параметрів з плином часу призводить до монополізації обслуговування споживачів області  $\Omega$  центром  $\tau_1$ , проте зменшення підмножини  $\Omega_2$  отриманої в результаті оптимального розбиття відбувається швидше ніж  $\Omega_3$ . Такий приклад демонструє суттєву залежність оптимального розбиття від двох ключових параметрів, а саме початкової вартості перевезення одиниці продукції та швидкості її зростання. Як було сказано вище визначальним є не стільки ціна на транспортування, як їх співвідношення для різних центрів, що безпосередньо впливає на площу підмножин  $\Omega_i$  при оптимальному розбитті множини  $\Omega$ .

Аналогічно представленому в модельній задачі 2.4 випадку в модельній задачі 2.5 суттєво на форму оптимального розбиття впливає розташування центрів відносно межі множини  $\Omega$ . Як видно центр  $\tau_2$  розміщено у безпосередній близькості до межі множини  $\Omega$ , що впливає безпосередньо на форму оптимальної підмножини  $\Omega_2$ . В наступному розділі дисертаційної роботи буде наведено, зокрема, чисельні результати оптимального розбиття множин з відшукуванням координат центрів підмножин. Буде показано, що при можливості, відповідно до математичної постановки задачі, вільного обрання координат центрів, з додатковою умовою мінімізації цільового функціоналу, центри розміщуються на достатній відстані від границі множини  $\Omega$ . Такий ефект має достатньо вагомі прикладні обґрунтування та обумовлюють бажання власників центрів постачання продукції розміститись якомога ближче до центра. На прикладі задачі з фіксованими центрами координати центрів є заданими та не підлягають зміні, проте їх розміщення біля або на значній відстані від меж області  $\Omega$  суттєво впливає на форму відповідних підмножин  $\Omega_i$  при оптимальному розбитті, проте, слід зазначити, не впливає на їх площу (кількість споживачів.)

### Модельна задача 2.6.

Розглядається задача оптимального розбиття області  $\Omega$  на чотири підмножини. Чотири центри нерівномірно розміщені по області  $\Omega$ , а саме:  $\tau_1(0,3;0,8)$ ,  $\tau_2(0,1;0,9)$ ,  $\tau_3(0,4;0,2)$ ,  $\tau_4(0,5;0,6)$ . Параметри швидкості зростання вартості транспортування обрані наступним чином:  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$ . В результаті чисельного експерименту отримано розбиття (рис.2.6 а-з). Можемо бачити, що при  $T \in [0,1]$  суттєва перевага між центрами відсутня, але при  $T \rightarrow 5$  область майже повністю складається з підмножини  $\Omega_1$ , що обумовлено значно нижчою ціною у транспортуванні одиниці товару.

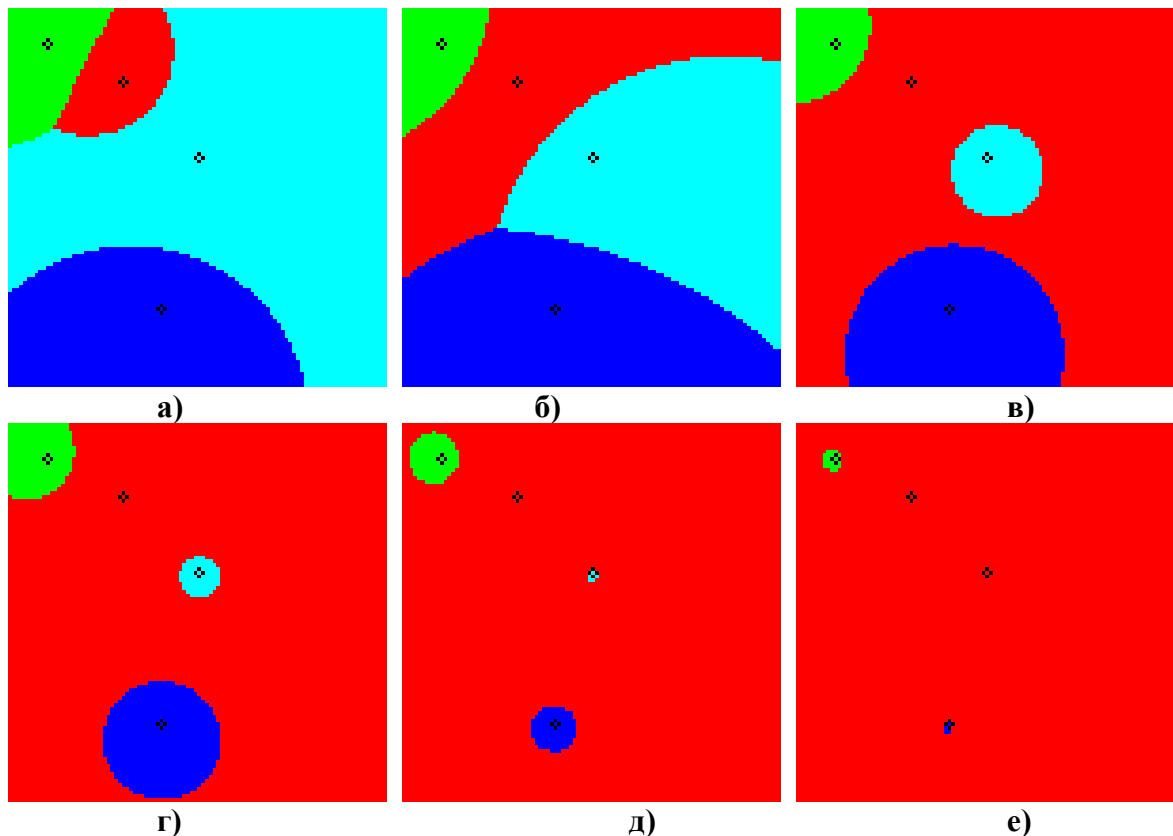


Рис.2.6 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на чотири підмножини з центрами  $\tau_1(0,3;0,8)$ ,  $\tau_2(0,1;0,9)$ ,  $\tau_3(0,4;0,2)$ ,  $\tau_4(0,5;0,6)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$  та а)  $T = 0,5$ ,  $F = 0,459$ , б)  $T = 1$ ,  $F = 2,254$ , в)  $T = 1,5$ ,  $F = 5,789$ , г)  $T = 2$ ,  $F = 10,832$ , д)  $T = 3$ ,  $F = 28,821$ , е)  $T = 5$ ,  $F = 139,731$

В модельних задачах 2.6-2.7 наведено випадки для чотирьох центрів. Такі випадки є більш цікавими з обчислювальної точки зору, адже при збереженні пропорцій між шириною та довжиною області розмір області не є настільки інформативним, як форма та співвідношення площ отриманих

підмножин та відповідного співвідношення кількостей споживачів центрів постачання продукції. В модельній задачі 2.6 центри розміщено в одній половині області  $\Omega$ , вони не розташовані на одній прямій та центри  $\tau_2, \tau_3$  розміщені у достатній близькості до границі області  $\Omega$ . Зауважемо, що початкова вартість транспортування продукції центрів  $\tau_1, \tau_2$  є в рази більшими за вартість в центрах  $\tau_3, \tau_4$ , що обумовлює нерівномірне розбиття для  $t \in [0,5;1]$ . Проте з плином часу швидкість зростання ціни на транспортування для центрів постачання, суттєвим чином визначає оптимальне розбиття при  $t > 1$ . Таким чином вже при  $t > 2$  переважна більшість споживачів загальної області  $\Omega$  споживатимуть товари з центру  $\tau_1$ . З наведених результатів можна зробити експериментальні висновки, щодо того які параметри моделі є більш визначальними для її динаміки та результату з плином часу. З високою ймовірністю можна стверджувати, що початкова вартість перевезення товарів з центру постачання до споживача, хоч і є вагомим для оптимального розбиття, проте з плином часу найбільш визначальним є саме швидкість зростання/спадання вартості транспортування продукції.

Важливим залишається близькість центрів до межі області, яка при сталому співвідношенні довжини та ширини області  $\Omega$  та знаходячись в межах 10% від її відповідного розміру не дозволяє локалізувати відповідну підмножину, а визначається, в тому числі, кількістю споживачів, розміщених біля межі області  $\Omega$ .

Фактично для задач з кількістю центрів  $> 3$  картина оптимального розбиття є вже достатньо типовою. При рівномірному розподілі центрів на області  $\Omega$  кожен з центрів покриває пропорційну вартості перевезення підмножину, а у разі їх попарної (або ін.) близькості один до одного може бути зведено до більш простих задач, на окремих частинах множини  $\Omega$ .

### **Модельна задача 2.7.**

Аналогічно модельній задачі 2.6 у даній модельній задачі завдання полягає в знаходженні оптимального розбиття на чотири підмножини з

центрами, розташованими симетрично один від одного  $\tau_1(0,25;0,25)$ ,  $\tau_2(0,25;0,75)$ ,  $\tau_3(0,75;0,25)$ ,  $\tau_4(0,75;0,75)$ . Зважаючи на пропорційну відмінність параметрів збільшення вартості перевезення між центрами, з плином часу радіус підмножин  $\Omega_2, \Omega_3, \Omega_4$  зменшуються, а площа підмножини  $\Omega_1$  наближається до площі множини  $\Omega$ . Результати розбиття представлені на Рис.2.7 а)-з).

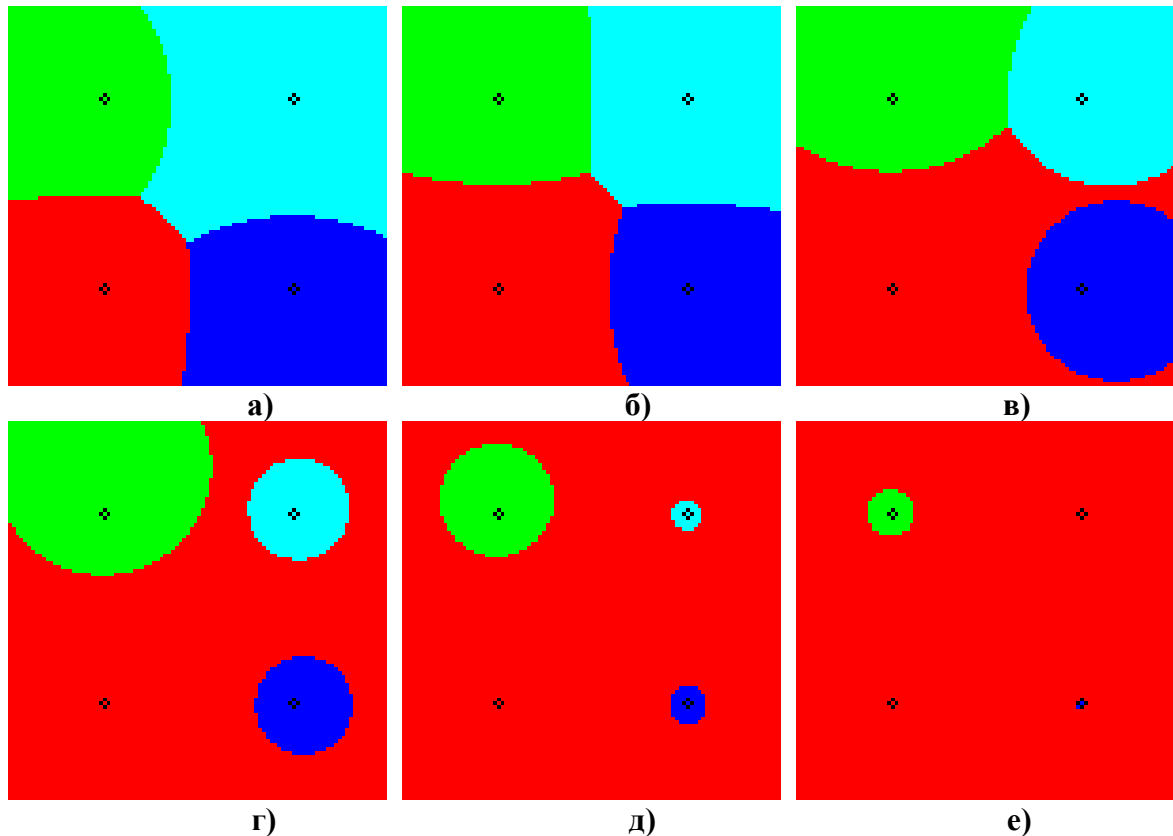


Рис.2.7 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на чотири підмножини з центрами  $\tau_1(0,25;0,25)$ ,  $\tau_2(0,25;0,75)$ ,  $\tau_3(0,75;0,25)$ ,  $\tau_4(0,75;0,75)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$  та а)  $T = 0,5$ ,  $F = 0,403$ , б)  $T = 1$ ,  $F = 1,577$ , в)  $T = 1,5$ ,  $F = 4,529$ , г)  $T = 2$ ,  $F = 9,832$ , д)  $T = 3$ ,  $F = 27,94$ , е)  $T = 5$ ,  $F = 137,981$

Як було показано вище рівномірний розподіл центрів підмножин на області  $\Omega$  відповідає типовій взаємодії між виробниками в рамках великих регіонів, або країн та зумовлює ефекти, коли відмінність вартості транспортування товарів до споживача, може не так суттєво вплинути на оптимальне розбиття. Як бачимо початкове співвідношення вартості транспортування продукції є достатньо різноманітним. Початкова вартість транспортування обрана такою ж, як і у випадку модельної задачі 2.6, проте

віддаленість центрів від межі області  $\Omega$  дозволило сформуванню колоподібну форму підмножин  $\Omega_2, \Omega_3, \Omega_4$  для  $t > 2$ . Слід підкреслити, що відносно ізольоване розташування центрів постачання один від одного формує підмножини в оптимальному розбитті таким чином, що співвідношення площ підмножин, отриманих в результаті оптимального розбиття, не завжди відповідає співвідношенню швидкості зростання вартості перевезення (див.рис.2.7-г)). При збереженні співвідношення в межах порядку, кожен центр постачання має свою обмежену множину споживачів, а при перевищенні співвідношення цін на транспортування порядку, відносить переважну більшість споживачів центру  $\tau_1$ . Випадок представлений на рис.2.7. є класичним, адже маючи можливість обрати кординати центрів вони були обрані максимально рівновіддаленими між собою та від межі області  $\Omega$ . Виходячи з таких початкових умов оптимальне розбиття при різних цінах є достатньо прогнозованим та пропорційно змінюється з плином часу та відповідно до зміни вартості транспортування одиниці продукції для різних центрів постачання.

### **Модельна задача 2.8.**

У представлений модельній задачі вхідні дані дані відповідають модельній задачі 2.7, але при цьому топологічно випадок є більш складеним. В задачі розглянуто питання оптимального розбиття області  $\Omega$  на п'ять підмножин, які відносно один одного розміщені нерівномірно, а саме:  $\tau_1(0,25;0,4)$ ,  $\tau_2(0,75;0,35)$ ,  $\tau_3(0,3;0,1)$ ,  $\tau_4(0,55;0,85)$ ,  $\tau_5(0,9;0,5)$ . При цьому збережено значну відмінність між швидкістю зростання вартості транспортування між центрами. Зважаючи на те що співвідношення швидкості зростання вартості транспортування продукції для різних пар центрів є відмінні загальною, тенденції динаміки зберігаються, але кожен з центрів втрачає вплив на множину споживачів у різний час.

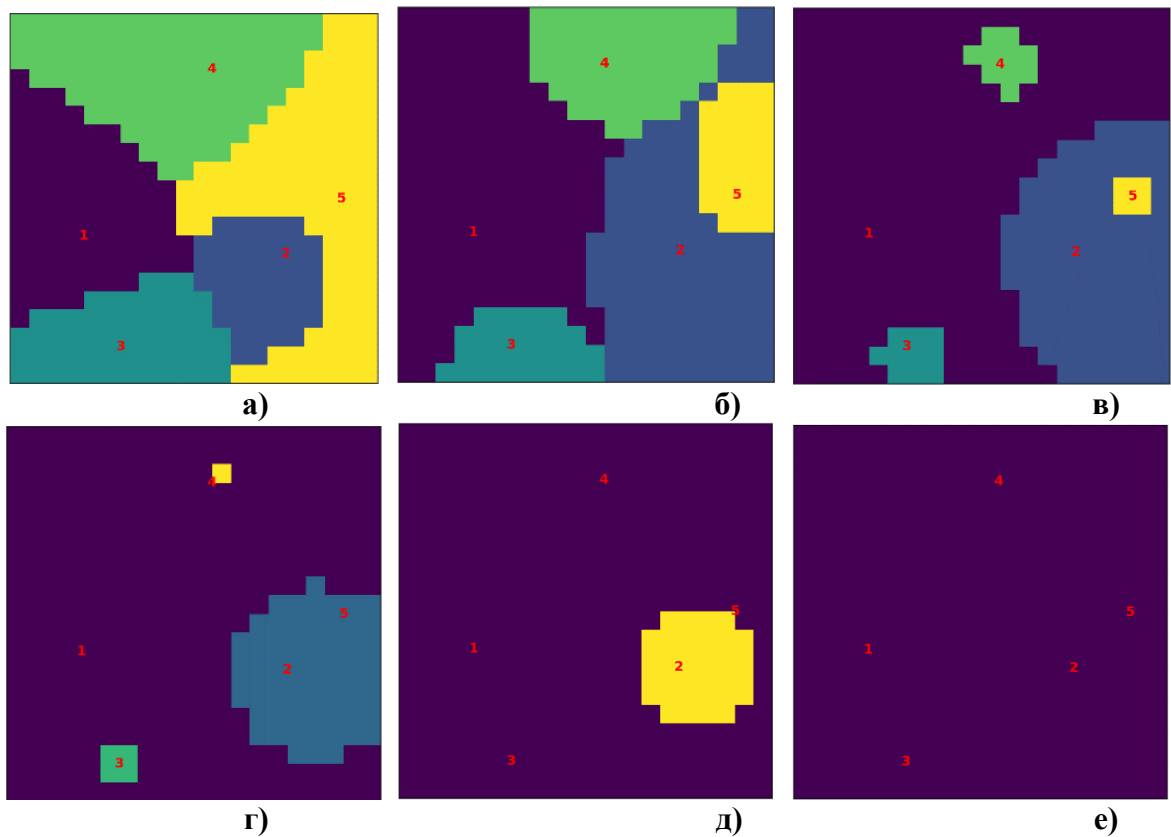


Рис.2.8 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на п'ять підмножин з центрами  $\tau_1(0,25;0,4)$ ,  $\tau_2(0,75;0,35)$ ,  $\tau_3(0,3;0,1)$ ,  $\tau_4(0,55;0,85)$ ,  $\tau_5(0,9;0,5)$ ,  $\Delta x = \Delta y = 0,05$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$ ,  $d_5 = 5$  та а)  $T = 0,5$ ,  $F = 0,606$ , б)  $T = 1$ ,  $F = 2,696$ , в)  $T = 1,5$ ,  $F = 6,771$ , г)  $T = 2$ ,  $F = 13,194$ , д)  $T = 2,5$ ,  $F = 23,092$ , е)  $T = 5$ ,  $F = 226,212$

В модельній задачі 2.8 наведено найбільш складений з модельних прикладів випадок. Кількість центрів з фіксованими координатами дорівнює 5, дотого ж їх розташування не є рівномірним, взаєморозташування різне та рівновіддаленість від межі області  $\Omega$  для деяких центрів є малим, а для інших більш великим. Зауважимо, що початкова вартість транспортування між центрами постачання була розподілена таким чином:  $c_1^0 = 1; c_2^0 = 2; c_3^0 = 3; c_4^0 = 4; c_5^0 = 5$ , а швидкості зміни вартості транспортування надано відповідні значення  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$ ,  $d_5 = 5$ . Наведена модельна задача наочно ілюструє суттєвий вплив на оптимальне розбиття взаєморозташування центрів. Як бачимо центр постачання  $\tau_2$  має достатньо низьку початкову вартість транспортування одиниці продукції в порівнянні з трьома центрами  $\tau_3, \tau_4$ , а тим паче  $\tau_5$ . Початкова вартість транспортування центру  $\tau_5$ , в порівнянні з центром  $\tau_2$  більше у 2,5 рази. Таке ж співвідношення

має місце і для швидкості зростання вартості транспортування центрів  $\tau_5$  та  $\tau_2$ . Проте ми бачимо, що розташування центру  $\tau_5$  в околі межі області  $\Omega$  суттєвим чином впливає на форму та площу підмножини  $\Omega_5$  рис.2.8-а зберігаючи її площу відмінною від 0. З плином часу співвідношення площ підмножин  $\Omega_2, \Omega_5$  наближається до співвідношення цін на транспортування одиниці продукції відповідних центрів, проте змін набуває і форма підмножин, що в черговий раз підтверджує вплив на оптимальне розбиття крайових умов задачі. Аналогічний ефект має місце і для пар центрів, один з яких розміщено в околі межі області  $\Omega$ , а саме це такі пари, як:  $\tau_1, \tau_4$ ;  $\tau_2, \tau_3$ ;  $\tau_3, \tau_5$ . Динамічні аспекти оптимального розбиття в залежності від зміни вартості транспортування продукції чисельно відрізняється проте якісно відповідає поведінці пари  $\tau_1, \tau_5$  описаної вище.

## 2.5 Висновки до розділу 2

У другому розділі дисертаційної роботи було формалізовано новий клас динамічних задач оптимального розбиття множин без обмежень із фіксованими координатами центрів. Формалізація проведена у контексті однопродуктового транспортування з урахуванням зміни вартості перевезень у часі. Такий підхід дозволив перейти від суто статичного формулювання до повноцінної динамічної моделі, що відображає часову мінливість витрат на транспортування між елементами множини та центрами.

Запропоновано методику побудови математичних моделей, в яких зміна транспортних витрат у часі задається за допомогою звичайних диференціальних рівнянь. Детально розглянуто чотири типи регуляторних сценаріїв зміни ціни: інфляційно-дефляційна модель, модель односторонньої цінової регуляції, двостороння регуляція та дотаційна модель. Для подальшого аналізу як базову обрано найпростішу модель інфляції/дефляції, яка дозволяє зберегти аналітичну прозорість результатів та водночас відтворити ключові властивості часової варіативності вартості.

На основі побудованої моделі було розроблено загальний алгоритм розв'язання динамічної задачі оптимального розбиття множини у випадку відомих координат центрів. Алгоритм передбачає: (1) формування системи диференціальних рівнянь, що описує зміну вартості перевезення у часі; (2) знаходження її розв'язку з початкових умов; (3) використання отриманих значень у функціоналі, що описує повну вартість транспортування; (4) побудову функцій приналежності точок до центрів на основі принципу мінімізації транспортних витрат.

Було проведено чисельне моделювання для восьми різних конфігурацій розміщення центрів та межових умов. У результаті одержано конкретні розв'язки систем диференціальних рівнянь, а також відповідні варіанти оптимального розбиття множини на підмножини. Отримані результати показали стійкість запропонованого підходу до змін параметрів задачі, а також адекватність моделі з прикладної точки зору.

Окрему увагу було приділено залежності характеру розбиття від кількості центрів та їх взаємного розміщення. Показано, що навіть незначна зміна просторової конфігурації центрів може призвести до суттєвого перерозподілу елементів між підмножинами, що підтверджує чутливість системи до геометричних факторів.

Запропонована методика є універсальною з точки зору її можливого розширення. Вона може бути адаптована для задач з обмеженнями на пропускну здатність, а також для випадків, коли координати центрів також мають визначатися в процесі оптимізації. У цьому контексті формалізована задача може слугувати базою для побудови більш складних моделей, що враховують множинні часові, просторові та економічні фактори.

Загалом, результати розділу демонструють доцільність переходу від статичних моделей до динамічних формулювань у задачах оптимального розбиття множин та відкривають перспективи для подальших досліджень у напрямі адаптивного та регульованого просторового планування.

### **Розділ 3. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН З ВІДШУКАННЯМ КООРДИНАТ ЦЕНТРІВ ПІДМНОЖИН ТА МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ**

Оптимальне розбиття множин, в своїй переважній більшості, безпосередньо пов'язано з мінімізацією цільового функціоналу, невід'ємною складовою якого є не тільки контури та інші кількісні параметри множини, що підлягає розбиттю, але й інші визначальні параметри, що є ключовими для шуканих підмножин. В класичних постановках такими параметрами є центри підмножин. В попередньому розділі показано алгоритми та методи розв'язання динамічних задач оптимального розбиття множин з фіксованими центрами. Така постановка має прикладне застосування, про яке, зокрема, і йшла мова у попередньому розділі. Проте часто центри підмножин не мають статичної структури тобто можуть змінювати своє розташування з часом, або ж однією з під задач є їх оптимальне розміщення. Прикладне застосування задач в такій постановці знаходять своє застосування в економіці, логістиці, медицині, архітектурі та інших галузях людського буття.

В даному розділі розглянуто динамічну задачу оптимального розбиття з розміщенням центрів підмножин. В детермінованій постановці така задача має аналітичне рішення, проте динамічна постановка має своїм розв'язком множину розбиттів, що слідує одне за одним з плином часу. В попередньому розділі було показано, що зміна співвідношення вартості транспортування для різних центрів суттєвим чином впливає на оптимальне розбиття та при  $t \rightarrow \infty$  і може виникнути ситуація при якій всі споживачі обслуговуються в єдиному центрі з найменшою вартістю виробництва та/або транспортування одиниці продукції. В динамічній постановці задачі оптимального розбиття з розміщенням центрів, зміна вартості перевезення одиниці товару для різних центрів може не тільки змінити оптимальне розбиття, але й змістити один або всі центри тим самим суттєвим чином

вплинувши, як на фінальне розбиття так і на значення цільового функціоналу. В цьому розділі будуть показані відмінності таких рішень в порівнянні з модельними задачами наведеними у попередньому розділі.

Вдалим прикладом для динамічних задач оптимального розбиття з розміщенням центрів можуть стати конкуренція в місті пересувних магазинів. Крім планових змін вартості товарів, для більшої зацікавленості покупців, такі магазини можуть змінювати місце свого розташування, що суттєвим чином впливає на зону покриття клієнтів такими пересувними магазинами. Таким чином зниження вартості продукції не завжди може призвести до збільшення кількості клієнтів, а збільшення ціни не завжди до його зменшення. Динамічні задачі оптимального розбиття множин з відшукуванням центрів підмножин — охоплюють широкий спектр методів для вирішення задач кластеризації, покриття, маршрутизації та розподілу ресурсів у змінюваних умовах. В Україні ця тематика активно досліджується як у прикладному плані, так і в межах фундаментальних наукових розвідок.

Серед основних підходів до вирішення згаданих задач, як в статичних так і в динамічних постановках є: математичне програмування, динамічне програмування, графові та мережеві моделі, евристичні та мета евристичні методи, машинне навчання тощо.

Математичне програмування, зокрема лінійне, цілочислові та нелінійні моделі – реалізують динамічну постановку з використанням багатостадійних формулювань або змінні параметри моделі. І хоча масштабованість задачі досі залишається проблемою, нові підходи, як, наприклад, *saturated cost partitioning* у класичному плануванні [172], [177], підвищують обчислювальну ефективність за рахунок динамічного розподілу вартостей операторів між евристичними, що охоплюють як фіксовані, так і рухомі центри.

Методи динамічного програмування забезпечують послідовну оптимізацію, але недоліком є проблема розмірності. Зокрема, в роботі [177] реалізовано підходи, що суттєво покращують розв'язання проблеми

масштабованості, зокрема, за допомогою оптимізації порядків евристик у динамічному плануванні.

При використанні графових та мережевих моделей, застосовують розрізи або потоки, динаміка задається змінами ваг, зв'язків чи топології. Ці моделі ефективні у маршрутизації та логістиці, приклади — FastMap для пошуку шляху, детально описаної в роботі [186] та розбиття мереж із навчанням з підкріпленням [78].

Евристичні та метаевристичні методи є достатньо ефективними при обмеженнях на обчислювальні ресурси. Так в роботі [172] застосовано алгоритми сходження на пагорбах для оптимізації порядку евристик у динамічному плануванні.

Підходи та методи машинного навчання використовують для прогнозування та адаптації розбиття за наявності історичних даних. Зокрема, застосовуються стратегії класифікації у динамічній багатокритеріальній оптимізації, що описано в роботах [191], [198] та навчання з підкріпленням для мережевого розбиття [78].

В роботі [8] застосовано варіаційні методи до задач мінімізації енергії з геометричними обмеженнями. В роботах [9], [23] приділено особливу увагу мінімізації власних значень Лапласових операторів, а серед сучасних робіт в яких, зокрема, застосоване підхід інтегрування рівневих множин та варіаційного числення для адаптивного розбиття можна назвати [68], [175].

В роботах [191], [198] наведено розроблені авторами стратегії прогнозування для відстеження множин Парето у змінних умовах. Оптимальне розбиття також застосовано до розподілу біомаси за градієнтами середовищами, описані, зокрема, в роботі [158].

Основи графових підходів, зокрема, таких як FastMap [186] і навчання з підкріпленням [78], демонструють гнучкість динамічного розбиття у логістиці та мережах. А в дослідженнях [10], [11] фокусуються на чисельних методах: рівневі множини, топологічна оптимізація, релаксація, регуляризація. Серед сучасних робіт опублікованих після 2020 року в яких,

зокрема, застосовуються алгоритми розв'язання задач оптимального транспортування у стохастичному контексті слід назвати роботу [175], а в роботі [68] розглядаються, зокрема, варіаційні методи для оптимізації форми області, що підлягає розбиттю.

Серед відкритих задач з цієї тематики слід виділити нечітку та стохастичну геометрію, багатокритеріальні формулювання у біомедицині та кібербезпеці, а також оптимізація у просторах імовірнісних мір, зокрема, описану в роботі [13]. Сучасні дослідження [175], [191], [198] зосереджені, зокрема, на напрямку інтеграції прогнозування та машинного навчання у динамічне розбиття. А в роботі [158] акцент зроблено на застосуваннях в екології та біомедицині.

Серед вітчизняних дослідників, що заклали основні, фундаментальні результати названої теорії оптимального розбиття множин, слід назвати професорку О.М. Кісельову та учнів її наукової школи. Основою наукових досліджень у цьому напрямку та фундаментом названої математичної теорії стали формалізовані постановки задач та їх аналітичні розв'язки, що викладені, зокрема в роботах [1], [6]. В названих роботах наведено формалізовані постановки задач та можливі їх аналітичні розв'язки. Розглянуті модельні задачі в чітких постановках, до яких наведено алгоритми їх розв'язання та результати багаточисельних комп'ютерних експериментів, які наочно ілюструють та підтверджують аналітичні результати.

Особливості розв'язання неперервно-дискретних задач, які обумовлені додатковими умовами розглянуті О.М.Кісельовою та учнями її наукової школи, зокрема, в роботах [4], [7], [22]. Дискретність задач обумовлено переліком обмежень як на постановку так і на алгоритми розв'язання таких задач. В роботі показано підходи, як узагальнені так і такі, що мають відношення до конкретних задач, з використанням яких авторці вдається отримати чисельний, а у більшості випадків і аналітичний розв'язок задачі.

### 3.1 Математична постановка однопродуктової динамічної задачі оптимального розбиття множин з відшукуванням координат центрів підмножин

Розглянемо неперервну однопродуктову динамічну задачу оптимального розбиття множин з  $E_n$  без обмежень з відшукуванням координат центрів підмножин [21] у такому вигляді.

Нехай  $\Omega$  – обмежена замкнена вимірна за Лебегом множина в  $n$ -вимірному евклідовому просторі  $E_n$ .

Сукупність вимірних за Лебегом підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega \subset E_n$  (серед яких можуть бути і порожні) назвемо можливим розбиттям цієї множини, якщо  $\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j)$ , де  $N > 0$  – задане натуральне число;  $\text{mes}(\cdot)$  – міра Лебега.

Позначимо через  $\sum_{\Omega}^N$  клас всіх можливих розбиттів множини  $\Omega \subset E_n$  на задану кількість  $N$  її вимірних за Лебегом підмножин:

$$\sum_{\Omega}^N = \{ \bar{\omega} \equiv \{ \Omega_1, \dots, \Omega_N \} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j \}$$

Далі, через  $\tau_1, \dots, \tau_N$  позначимо сукупність деяких еталонних точок для підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, які назвемо центрами цих підмножин:  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , і будемо вважати, що координати усіх центрів невідомими та такими, що підлягають знаходженню.

**Постановка задачі.** Знайти розбиття  $\bar{\omega} = \{ \Omega_1, \dots, \Omega_N \} \in \sum_{\Omega}^N$  множини  $\Omega \subset E_n$ , векторну функцію  $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$ , визначену майже всюди (м.в.) для  $x \in \Omega$ , координати центрів  $\tau = \{ \tau_1, \dots, \tau_N \} \subset \Omega^N$  та всіх  $t \in [0, T]$ , які забезпечують

$$\inf_{\bar{\omega} \in \sum_{\Omega}^N : \{ \tau_1, \dots, \tau_N \} \subset \Omega^N : c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot)) \quad (3.1)$$

де

$$F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad (3.2)$$

за наявності умов

$$\frac{dc_i(x, \tau_i, t)}{dt} = \sum_{j=1}^N A_{ij} \cdot f_j(c_j(x, \tau_j, t)), 0 \leq t \leq T \quad (3.3)$$

$$c_i(x, \tau_i, t_0) = c_{0i}(x, \tau_i), i = 1, \dots, N$$

м.в. для  $x \in \Omega$  та умови замкненості системи

$$\sum_{i=1}^N A_{ij} = 1, j = 1, \dots, N. \quad (3.4)$$

Тут  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  - шукані центри підмножин з координатами  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega, i = 1, \dots, N$ ;  $c_i(x, \tau_i, t), i = 1, \dots, N$ , - шукані дійсні функції, визначені на  $\Omega \times \Omega \times [0, T]$ , що є неперервно диференційованими за аргументом  $t$  на відрізку  $[0, T]$  м.в. для  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ , обмежені та вимірні за аргументом  $x$  на  $\Omega$  для всіх  $t \in [0, T]$ ;  $m(x, \tau_i), c_{0i}(x, \tau_i)$  - задані дійсні функції, визначені на  $\Omega \times \Omega$ , обмежені та вимірні за аргументом  $x \in \Omega$  (зокрема,  $m(x, \tau_i)$  може відігравати роль метрики на  $\Omega \times \Omega$ );  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , - задані дійсні функції, неперервні та ліпшицеві в області свого визначення;  $\rho(x)$  - задана невід'ємна функція, обмежена та вимірна на  $\Omega$ ;  $a_i, i = 1, \dots, N$ , - задані, як правило, невід'ємні числа;  $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$ , - задані числові параметри;  $T > 0$  та  $t_0 \in [0, T]$  задані.

Тут та в подальшому інтеграли розуміються в сенсі Лебега. Будемо вважати, що міра множини межових точок підмножин  $\Omega_1, \dots, \Omega_N$  дорівнює нулю.

Трійку  $(\bar{w}^*, \tau^*, c^*(x, \tau, t))$ , що доставляє функціоналу (3.2) мінімальне значення на множині  $\sum_{\Omega}^N \times L_2^N(\Omega \times \Omega \times [0, T])$  за умов (3.3), (3.4), назвемо оптимальним розв'язком задачі (3.1)–(3.4). При цьому розбиття  $\bar{w}^* = \{\Omega_1^*, \dots, \Omega_N^*\} \in \sum_{\Omega}^N$  назвемо оптимальним розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин,  $\tau^* = \{\tau_1^*, \dots, \tau_N^*\} \subset \Omega^N$ ,  $\tau_i^* = (\tau_i^{*(1)}, \dots, \tau_i^{*(n)}), i = 1, \dots, N$  оптимальними центрами та їх координатами відповідно, а векторну функцію  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)) \in L_2^N(\Omega \times \Omega \times [0, T])$  - оптимальною фазовою траєкторією динамічної системи (3.3) в задачі (3.1)–(3.4).

З предметної точки зору, незалежна змінна  $t \in [0, T]$  у наведеній математичній постановці динамічної задачі оптимального розбиття може відігравати роль часової змінної, а  $T > 0$  та  $t_0 \in [0, T]$  – задані кінцевий та початковий моменти часу у досліджуваному динамічному процесі відповідно. Так, функції  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , у диференціальних зв'язках (3.3), що відображають динаміку цін на транспортування, можуть мати різний вигляд (див. формули (2.5)-(2.8)), в залежності від предметного змісту, який в неї вкладається, наприклад: моделі інфляції/дефляції; моделі односторонньої цінової регуляції; моделі двосторонньої цінової регуляції; моделі дотаційної односторонньої цінової регуляції.

В якості базової в наступних пунктах буде використана модель інфляції/дефляції (2.5), як і в розділі 2 для наочної демонстрації позитивних змін у значеннях цільового функціоналу в однопродуктовій динамічній задачі оптимального розбиття множин з відшукуванням координат центрів підмножин.

### 3.2 Обґрунтування методу розв'язання

Нехай  $\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$  – деяке можливе розбиття вихідної замкненої обмеженої вимірної множини  $\Omega \subset E_n$  на  $N$  її вимірних за Лебегом підмножин, що не перетинаються.

Введемо у розгляд на множині  $\Omega \subset E_n$  характеристичні функції  $\lambda_1(x), \dots, \lambda_N(x)$  підмножин  $\Omega_1, \dots, \Omega_N$ , відповідно:

$$\lambda_i(x) = \begin{cases} 1, & x \in \Omega_i, \\ 0, & x \in \Omega \setminus \Omega_i, \end{cases} \quad i = 1, \dots, N \quad (3.5)$$

і переформулюємо задачу (3.1)–(3.4) у термінах характеристичних функцій підмножин у більш зручному вигляді. Потрібно знайти векторні функції  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x)), x \in \Omega$ ,  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)), x \in \Omega$  при  $t \in [0, T]$  та координати центрів  $\tau^* = \{\tau_1^*, \dots, \tau_N^*\} \subset \Omega^N$ , які забезпечують

$$\inf_{\lambda(\cdot) \in \Gamma_1; \{\tau_1, \dots, \tau_N\} \subset \Omega^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot)) \quad (3.6)$$

де функціонал  $J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot))$  має вигляд

$$J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) \lambda_i(x) dx dt, \quad (3.7)$$

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega; \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \} \quad (3.8)$$

за умов (3.3)–(3.4). Очевидно, що для функціонала  $F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot))$  виду (3.2) і функціонала  $J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot))$  виду (3.7) виконується рівність  $J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot)) = F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot))$ .

Далі від задачі (3.6)–(3.8) за умов (3.3)–(3.4) з булевими значеннями функцій  $\lambda_i(\cdot), i = 1, \dots, N$ , перейдемо до відповідної задачі зі значеннями  $\lambda_i(\cdot), i = 1, \dots, N$ , що змінюються на відрізьку  $[0, 1]$ .

**Задача.** Знайти векторні функції  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x)), x \in \Omega$  і  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)), x \in \Omega$ , центри  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  при  $t \in [0, T]$ , які забезпечують

$$\inf_{\lambda(\cdot) \in \Gamma; \{\tau_1, \dots, \tau_N\} \subset \Omega^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot)) \quad (3.9)$$

де функціонал  $J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot))$  має вигляд

$$J(\lambda(\cdot), \tau, c(\cdot, \cdot, \cdot)) = \int_0^T \int_{\Omega} \sum_{i=1}^N (c_i(x, \tau_i, t) m(x, \tau_i) + a_i) \rho(x) \lambda_i(x) dx dt, \quad (3.10)$$

$$\Gamma = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega; 0 \leq \lambda_i(x) \leq 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \} \quad (3.11)$$

за умов (3.3)–(3.4).

Перша компонента  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x))$  оптимального розв'язку такої задачі м.в. для  $x \in \Omega$  і для кожного  $i = 1, \dots, N$  має вигляд

$$\lambda_i^*(x) = \begin{cases} 1, \text{ якщо виконується умова (3.13)} \\ 0, \text{ в іншому випадку} \end{cases} \quad (3.12)$$

$$\int_0^T (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dt = \min_{k=1, N} \left( \int_0^T (c_k(x, \tau_k, t) \cdot m(x, \tau_k) + a_k) \rho(x) dt \right). \quad (3.13)$$

### 3.3 Алгоритм розв'язання однопродуктової динамічної задачі оптимального розбиття множин з розміщенням центрів підмножин

Запропонований алгоритм однопродуктової динамічної задачі оптимального розбиття множин з розміщенням центрів підмножин базується на одному з варіантів методу загального градієнтного спуску з розтягненням простору в напрямку різниці двох послідовних загальних градієнтів (r-алгоритм), спрямований в загальному випадку на пошук локальних мінімумів недиференційованої багатоекстремальної цільової функції

$$G(\tau) = \int_0^T \int_{\Omega} \inf_{k=1, \dots, N} (c_k(x, t) \cdot m(x, \tau_k) + a_k) \rho(x) dx dt \rightarrow \inf$$

$$\tau \in \Omega^N \subset \underbrace{E_n \times \dots \times E_n}_N$$

Необхідно визначити  $i$ -ту компоненту вектора субградієнту  $g_G^{\tau}(\tau) = (g_G^{\tau_1}(\tau), \dots, g_G^{\tau_i}(\tau), \dots, g_G^{\tau_N}(\tau))$  функції

$$G(\tau_1^{(1)}, \dots, \tau_1^{(n)}; \dots; \tau_i^{(1)}, \dots, \tau_i^{(n)}; \dots; \tau_N^{(1)}, \dots, \tau_N^{(n)}) =$$

$$= \int_0^T \int_{\Omega} \inf_{i=1, \dots, N} c_i(x^{(1)}, \dots, x^{(n)}; t) \cdot m(x^{(1)}, \dots, x^{(n)}; \tau_i^{(1)}, \dots, \tau_i^{(n)}) \times \rho(x^{(1)}, \dots, x^{(n)}) dx^{(1)} \dots dx^{(n)} dt$$

в точці  $\tau = (\tau_1^{(1)}, \dots, \tau_N^{(n)})$  наступним чином:

$$g_G^{\tau_i}(\tau) = g_G^{\tau_i}(\tau_1^{(1)}, \dots, \tau_N^{(n)}) = \int_0^T \int_{\Omega} c_i(x^{(1)}, \dots, x^{(n)}; t) \times g_m^{\tau_i}(x^{(1)}, \dots, x^{(n)}; \tau_i^{(1)}, \dots, \tau_i^{(n)}) \times \lambda_i(x^{(1)}, \dots, x^{(n)}) \times$$

$$\times p(x^{(1)}, \dots, x^{(n)}) dx^{(1)} \dots dx^{(n)} dt, i = 1, 2, \dots, N \quad (3.14)$$

де  $g_m^{\tau_i}(x, \tau)$  -  $i$ -та компонента  $N$ -вимірного вектора субградієнту  $g_m^{\tau}(x, \tau)$  функції  $m(x, \tau_i)$  в точці  $\tau = (\tau_1, \dots, \tau_i, \dots, \tau_N)$ .

Для розв'язання задачі r-алгоритмом у В-формі включаємо область  $\Omega$  у  $n$ -вимірний паралелепіпед  $\Pi$ , сторони якого паралельні осям декартової системи координат. Для простоти вважатимемо  $p(x) = 1$  для  $x \in \Pi \setminus \Omega$ . Паралелепіпед  $\Pi$  покриваємо прямокутною сіткою та переходимо до основного етапу.

#### Основний етап:

Крок 1. Обираємо початкове наближення  $\tau = \tau^{(0)}$  та коефіцієнт розтягу простору  $\alpha > 1$ . Покласти  $B_0^{\tau} = I_n$ .

Крок 2. Задаємо значення функцій  $c_{0i}(x)$ ,  $i=1,\dots,N$ , з початкових умов (3.3) в вузлах сітки, та покриваємо відрізок  $[0,T]$  сіткою з кроком  $h_t$ .

Крок 3. Для кожного вузла сітки розв'язуємо задачу Коші для системи ЗДР (3.3) на часовому відрізку  $[0,T]$  та знаходимо функції  $c_i^{(k)}(x,t)$ ,  $i=1,\dots,N$ .

Крок 4. Обчислюємо значення  $\lambda^{(k)}(x)$  вузлах сітки за формулами (3.5) коли  $\tau = \tau^{(k)}$ .

Крок 5. Обчислюємо  $g_G(\tau)$  – за формулою (3.14) коли  $\lambda(x) = \lambda^{(k)}(x)$ ,  $\tau = \tau^{(k)}$ .

Крок 6. Обчислюємо  $\xi_k^\tau$  за формулою  $\xi_k^\tau = \frac{(B_k^\tau)^T g_G(\tau)}{\|(B_k^\tau)^T g_G(\tau)\|}$  при  $\tau = \tau^{(k)}$ .

Крок 7. Розраховуємо  $\tau = \tau^{(k+1)}$  за формулою  $\tau^{k+1} = P_\Pi(\tau^k - h_k B_k^\tau \xi_k^\tau)$

де  $P_\Pi$  - оператор проектування на  $\Pi$ .

Крок 8. Для кожного вузла сітки розв'язуємо задачу Коші для системи ЗДР (3.3) на часовому відрізку  $[0,T]$  та знаходимо функції  $c_i^{(k+1)}(x,t)$ ,  $i=1,\dots,N$ .

Крок 9. Обчислюємо значення  $\lambda^{(k+1)}(x)$  вузлах сітки за формулами (3.5) коли  $\tau = \tau^{(k+1)}$ .

Крок 10. Обчислюємо  $g_G(\tau)$  – за формулою (3.14) коли  $\lambda(x) = \lambda^{(k+1)}(x)$ ,  $\tau = \tau^{(k+1)}$ .

Крок 11. Обчислити

$$\eta_k^\tau = \frac{(B_k^\tau)^T r_k^\tau}{\|(B_k^\tau)^T r_k^\tau\|}, \quad r_k^\tau = g_G(\tau^{(k+1)}) - g_G(\tau^{(k)})$$

$$R_\beta(\eta_k^\tau) = I_n + (\beta - 1)\eta_k^\tau (\eta_k^\tau)^T, \quad \beta = \frac{1}{a} < 1$$

$$B_{k+1} = B_k^\tau R_\beta(\eta_k^\tau)$$

Крок 12. Якщо умова

$$\|\tau^k - \tau^{k+1}\| \leq \varepsilon, \quad \varepsilon > 0 \quad (3.15)$$

не виконується, вважаємо  $k = k + 1$  і переходимо до кроку 6, в іншому випадку переходимо до кроку 13.

Крок 13. Вважаємо  $\lambda_*(x) = \lambda^{(n)}(x)$ ,  $c_*(x, t) = c^{(n)}(x, t)$ ,  $\tau_* = \tau^{(n)}$ , де  $n$  – номер ітерації на якій виконалась умова (3.15).

Крок 14. Розраховуємо оптимальне значення цільового функціоналу по формулі

$$G(\tau) = \int_0^T \int_{\Omega} \inf_{i=1, \dots, N} (c_i(x, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt$$

при  $\tau = \tau_*$

Алгоритм розв'язання неперервної однопродуктової динамічної задачі оптимального розбиття множин з  $E_n$  без обмежень з відшуканням координат центрів підмножин описаний.

*Зауваження.* Також відзначимо, що оператор проектування  $P_{\Pi}$  є обов'язковою компонентною цього алгоритму, тому що ми маємо обмеження паралелепіпедом  $\Pi$ , і при ітерації методу ми можемо вийти за обмеження. Існують різні методи його реалізації, більш детально описано в [22].

### 3.4 Розв'язання модельних задач та оцінка чисельних результатів

Описана вище неперервна однопродуктова динамічна задача оптимального розбиття множин з  $E_n$  без обмежень з відшуканням координат центрів підмножин з моделлю (2.5) у якості базової була розв'язана для різних модельних задач, для яких виконується обмеження (3.4), а початкові умови системи ЗДР (3.3) та інші параметри моделі обрані аналогічно модельним задачам 2.1-2.8. Модельні задачі є нескінченновимірними задачами розбиття заданої області з фіксованими центрами підприємств, що виробляють однорідну продукцію для розподілення в цій області із заданою щільністю споживачів. Параметри модельних задач 3.1-3.6 є достатньо різноманітними та дозволяють наочно дослідити переваги задачі з відшуканням центрів підмножин в порівнянні з задачами з фіксованими центрами. Обрання параметрів відмінними від модельних задач 2.1-2.6 не

дали б змоги наочно побачити ефект від постановки задачі з відшукуванням центрів тому були обрані аналогічними.

### Модельна задача 3.1.

Споживачі деякої однорідної продукції, що виробляється двома підприємствами, неперервно розподілені в області  $\Omega = \{x = (x^{(1)}, x^{(2)}) : 0 \leq x^{(1)} \leq 1; 0 \leq x^{(2)} \leq 1\}$ , кількість підприємств постачання продукції дорівнює двом, координати оптимального розміщення підприємств  $\tau_1, \tau_2$  невідомі та підлягають визначенню. Для двох підприємств задана система відповідно до якої змінюється ціна на транспортування одиниці продукції на одиницю відстані

$$\begin{cases} \frac{\partial c_1(x, \tau_1, t)}{\partial t} = A_{11} \cdot d_1 \cdot c_1(x, \tau_1, t) + (1 - A_{22}) \cdot d_2 \cdot c_2(x, \tau_2, t) \\ \frac{\partial c_2(x, \tau_2, t)}{\partial t} = (1 - A_{11}) \cdot d_1 \cdot c_1(x, \tau_1, t) + A_{22} \cdot d_2 \cdot c_2(x, \tau_2, t) \end{cases}, \quad 0 \leq t \leq T, \quad (3.16)$$

з початковими умовами

$$c_1^0 = 2.0, \quad c_2^0 = 1.0 \quad (3.17)$$

В цій та наступних модельних задачах у якості  $m(x, \tau_i)$  використана евклідова метрика.

Розв'язок задачі (3.16)-(3.17) описує вартість транспортування одиниці продукції з  $i$ -го підприємства до споживача з координатами  $(x, y)$ . Відомий попит  $\rho(x, y)$  на продукцію для кожного пункту споживання з координатами  $(x, y)$ . Для простоти вважається  $\rho(x, y) \equiv 1, \forall x \in \Omega$ .

Множину споживачів можна розбити на зони обслуговування  $\Omega_i, i = 1, 2$  споживачів  $i$ -им пунктом виробництва так, щоб

$$\bigcup_{i=1}^2 \Omega_i = \Omega, \quad \text{mes}(\Omega_i \cap \Omega_k)_{i \neq k} = 0, \quad i, k = 1, 2 \quad (3.18)$$

Для випадку  $N = 2$  формулу (3.18) можна було б записати в більш простому вигляді, проте у наступних модельних задачах значення  $N$  прийматиме значення 2, 3 та 4, а відповідні умови залишатимуться у

наведеному аналітичному вигляді. Зазначимо, що потужність  $i$ -го підприємства  $i=1,2$  дорівнює сумарному попиту споживачів  $\Omega_i, i=1,2$ .

Необхідно розбити множину споживачів  $\Omega$  так, що б мінімізувати функціонал (3.1), (3.2), (3.6) сумарних витрат на виробництво продукції та її доставку до споживача при умовах (3.3), (3.4) та  $N=2$ .

Для розв'язання сформульованої задачі область  $\Omega$  була покрита прямокутною сіткою з вузлами  $(i, j)$ ,  $i, j=1, \dots, 101$ . Було отримано чисельний розв'язок системи (3.16), для якого був використаний метод Рунге-Кути 4-го порядку. Пошук розв'язку системи (3.16) було здійснено при  $h_t = 0,001$ , а для отриманих точок фазової траєкторії при  $T = [0,5; 1; 1,5; 2; 3; 5; 10]$  було розраховано вартість транспортування від кожного вузла  $(i, j)$  області  $\Omega$  до центрів  $\tau_1, \tau_2$ . Виходячи з критерію мінімальної вартості вузол  $(i, j)$  був віднесений до областей  $\Omega_1$  та  $\Omega_2$  відповідно. Результати чисельного експерименту наведені на рис.3.1 (а-ж).

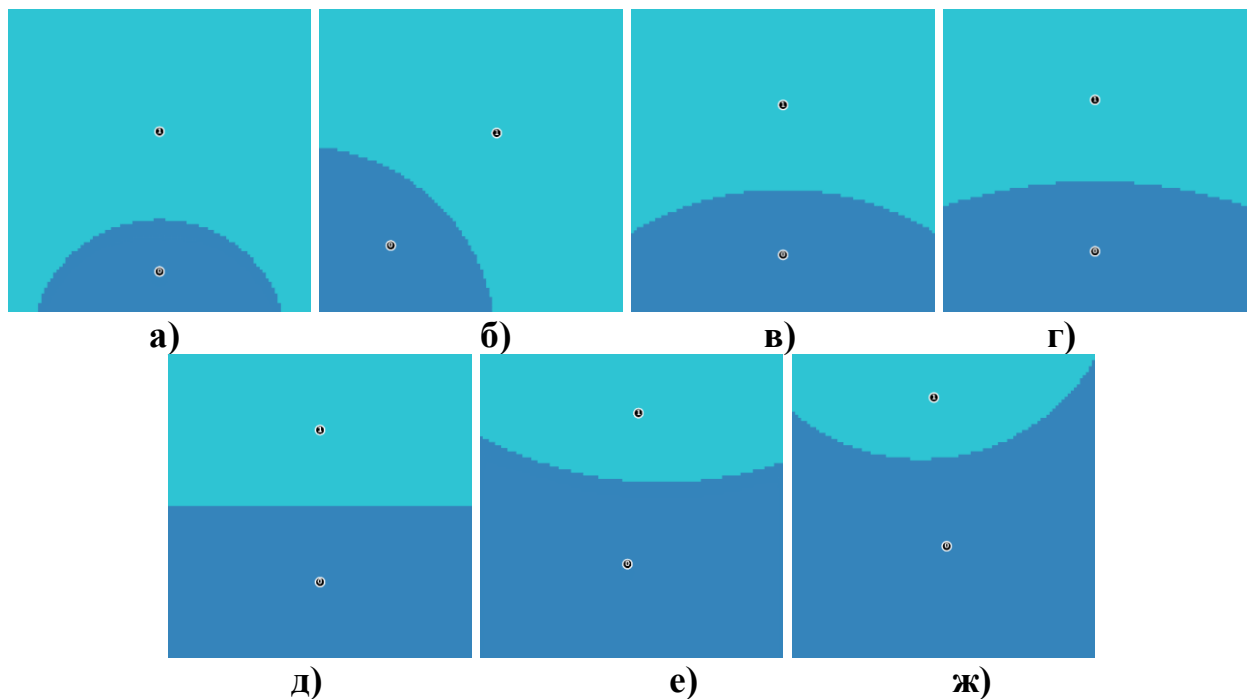


Рис.3.1 – Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з початковими центрами  $\tau_1^0(0,5;0,25)$ ,  $\tau_2^0(0,5;0,75)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ , б)  $T = 1$ , в)  $T = 1,5$ , г)  $T = 2$ , д)  $T = 3$ , е)  $T = 5$ , ж)  $T = 10$ .

Інші числові параметри чисельного експерименту (рис.3.1) наведені в таблиці 3.1.

Таблиця 3.1

Розгорнуті параметри числового експерименту наведеного на рис.3.1

Рис.3.1	Часовий проміжок	Кількість ітерацій	Значення цільового функціоналу	Оптимальні координати центрів $\tau_i$ підмножин $\Omega_i$
а)	[0; 0,5]	24	0,2321	$\tau_1 = (0,5; 0,134)$ ; $\tau_2 = (0,5; 0,596)$
б)	[0; 1]	38	0,5613	$\tau_1 = (0,236; 0,22)$ ; $\tau_2 = (0,585; 0,591)$
в)	[0; 1,5]	28	1,0484	$\tau_1 = (0,5; 0,189)$ ; $\tau_2 = (0,5; 0,684)$
г)	[0; 2]	31	1,6763	$\tau_1 = (0,5; 0,2)$ ; $\tau_2 = (0,5; 0,7)$
д)	[0; 3]	0	3,5395	$\tau_1 = (0,5; 0,25)$ ; $\tau_2 = (0,5; 0,75)$
е)	[0; 5]	29	11,4382	$\tau_1 = (0,487; 0,309)$ ; $\tau_2 = (0,523; 0,806)$
ж)	[0; 10]	29	112,7064	$\tau_1 = (0,51; 0,368)$ ; $\tau_2 = (0,468; 0,857)$

Як було сказано вище, чисельний експеримент проведено з параметрами аналогічними модельній задачі 2.1 за тією відмінністю, що розташування центрів підмножин відшукуються оптимальними для мінімізації цільового функціоналу. Кожне з наведених на рис.1 а)-ж) ілюструє оптимальне розміщення центрів при заданих параметрах моделі та значеннях вартості перевезення одиниці продукції для кожного з центрів  $\tau_i$ . Як бачимо, навіть при симетричній формі області  $\Omega$  розміщення центрів не завжди є симетричним відповідно одне одного. У випадку рис.1-а) вартість транспортування для центру  $\tau_1$ , майже вдвічі більше вартості для центру  $\tau_2$ . Це зумовлює нерівномірний розподіл споживачів між двома центрами. Проте, варто зазначити, що наведені в модельній задачі 2.1 координати центрів для першого ж кроку розв'язку системи (3.3) набули скорегованих значень  $\tau_1 = (0,5; 0,1337)$ ;  $\tau_2 = (0,5; 0,5956)$ . Результати такого корегування суттєвим чином вплинули на значення цільового функціоналу, що для координат центрів заданих в модельній задачі 2.1 мав значення 0,245, а для

скорегованих координат центрів набув значення 0,2321. В абсолютних значеннях така відмінність не сутєва, проте, у відносних показниках значення цільового функціоналу зменшилось більш ніж на 5%. В таблиці 3.2 наведені результати порівняльного аналізу результатів оптимального розбиття отриманих в модельній задачі 2.1 та результатів модельної задачі з відшукуванням оптимальних координат центрів підмножин 3.1.

В наступних таблицях розділу 3 будуть використані такі скорочення: МД – Модельна задача, ЦФ – цільовий функціонал,  $\Delta$  - абсолютна різниця значень цільових функціоналів (2.14) та (3.10),  $\varepsilon$  - відносна різниця значень цільових функціоналів (2.14) та (3.10).

Таблиця 3.2

Результати порівняльного аналізу результатів модельної задачі 2.1. та 3.1 з відшукуванням координат центрів підмножин

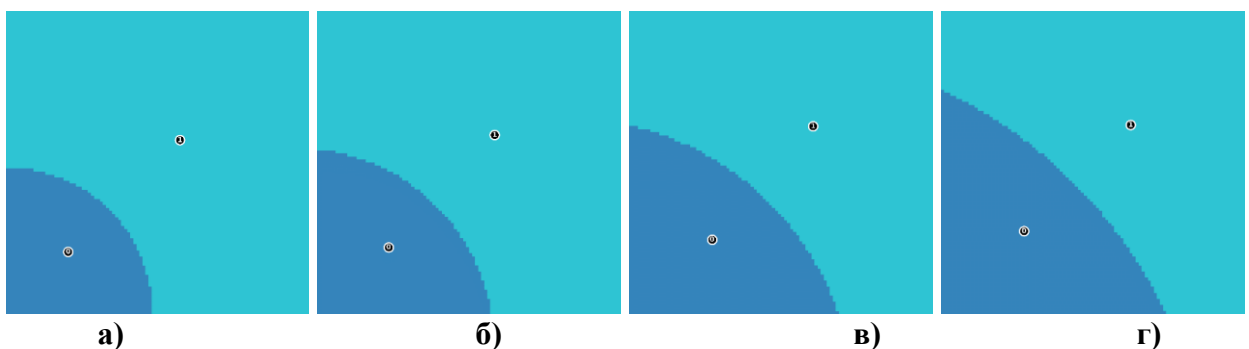
МД 2.1 $\tau_1(0.5;0.25)$ , $\tau_2(0.5;0.75)$	Значення ЦФ (2.14)	МД 3.1	Оптимальні координати центрів	Значення ЦФ (3.10)	$\Delta$	$\varepsilon$
Рис.2.1-а)	0,245	Рис.3.1-а)	$\tau_1 = (0,5;0,134)$ ; $\tau_2 = (0,5;0,596)$	0,232	0,013	-5,306%
Рис.2.1-б)	0,597	Рис.3.1-б)	$\tau_1 = (0,236;0,22)$ ; $\tau_2 = (0,585;0,591)$	0,561	0,036	-6,03%
Рис.2.1-в)	1,079	Рис.3.1-в)	$\tau_1 = (0,5;0,189)$ ; $\tau_2 = (0,5;0,684)$	1,048	0,031	-2,873%
Рис.2.1-г)	1,727	Рис.3.1-г)	$\tau_1 = (0,5;0,2)$ ; $\tau_2 = (0,5;0,7)$	1,676	0,051	-2,953%
Рис.2.1-е)	3,728	Рис.3.1-д)	$\tau_1 = (0,5;0,25)$ ; $\tau_2 = (0,5;0,75)$	3,54	0,188	-5,043%
Рис.2.1-ж)	13,105	Рис.3.1-е)	$\tau_1 = (0,487;0,309)$ ; $\tau_2 = (0,523;0,806)$	11,438	1,667	-12,72%
Рис.2.1-з)	182,925	Рис.3.1-ж)	$\tau_1 = (0,51;0,368)$ ; $\tau_2 = (0,468;0,857)$	112,706	70,219	-38,387%

Як бачимо, зі збільшенням вартості транспортування не тільки збільшується абсолютна різниця між значеннями цільового функціоналу задачі з фіксованими центрами та їх розміщенням, але й суттєвим чином збільшується відносна різниця та вже при  $T=10$  сягає більш ніж 38%. Таке відхилення дає змогу зрозуміти, що попри складності з відшукуванням оптимальних координат центрів підмножин, можливість їх вільного розміщення суттєво впливає на ефективність розв'язку задачі та значення

цільового функціоналу. В наступних модельних задачах будуть розглянуто випадки більш асиметричним розташування початкових центрів та вплив на ефективність розбиття їх кількості.

### Модельна задача 3.2.

Вхідні дані модельної задачі 3.2 обрано аналогічно до модельної задачі 3.1, за винятком початкових координат двох центрів  $\tau_1^0(0,1;0,2)$ ,  $\tau_2^0(0,8;0,6)$ , які, згідно із загальним виглядом цільового функціоналу (3.1), (3.2), умовами (3.3), (3.4) та описаним вище алгоритмом, обчислюється таким, що оптимізує значення цільового функціоналу, а у зв'язку зі зміною параметрів  $c_i(x, \tau_i, t), i = 1, \dots, N$  уточнюються на кожній ітерації. Цей приклад ілюструє, як симетричність розташування центрів впливає на оптимальне розбиття. Отримані результати представлені на рис. 3.2 а)–з), мають подібну структуру до результатів, наведених у задачі 3.1, але з тією відмінністю, що віддаленість одного з центрів від межі області впливає як на форму межі між областями  $\Omega_1, \Omega_2$ , так і на значення цільового функціоналу. Варто підкреслити, що при такому положенні центру  $\tau_1$  вартість транспортування одиниці продукції для нього суттєво зростає. Як наслідок, розмір відповідної області збільшується, що призводить до значного зростання цільового функціоналу — більш ніж на два порядки (див. рис. 3.2-з).



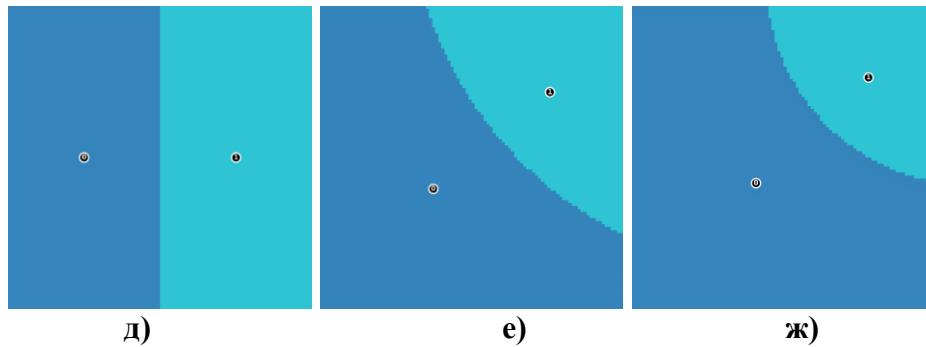


Рис.3.2 – Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з початковими центрами  $\tau_1^0(0,1;0,2)$ ,  $\tau_2^0(0,8;0,6)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1,0$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ , б)  $T = 1$ , в)  $T = 1,5$ , г)  $T = 2$ , д)  $T = 3$ , е)  $T = 5$ , ж)  $T = 10$ .

Інші числові параметри чисельного експерименту (рис.3.2) наведені в таблиці 3.3.

Таблиця 3.3

Розгорнуті параметри чисельного експерименту наведеного на рис.3.2

Рис.3.2	Часовий проміжок	Кількість ітерацій	Значення цільового функціоналу	Оптимальні координати центрів $\tau_i$ підмножин $\Omega_i$
а)	[0;0,5]	23	0,224	$\tau_1=(0,205;0,205)$ ; $\tau_2=(0,574;0,574)$
б)	[0;1]	26	0,561	$\tau_1=(0,236;0,22)$ ; $\tau_2=(0,585;0,591)$
в)	[0;1,5]	30	1,037	$\tau_1=(0,274;0,245)$ ; $\tau_2=(0,605;0,62)$
г)	[0;2]	29	1,679	$\tau_1=(0,273;0,273)$ ; $\tau_2=(0,624;0,624)$
д)	[0;3]	34	3,54	$\tau_1=(0,25;0,5)$ ; $\tau_2=(0,75;0,5)$
е)	[0;5]	29	11,409	$\tau_1=(0,374;0,396)$ ; $\tau_2=(0,758;0,716)$
ж)	[0;10]	28	109,886	$\tau_1=(0,409;0,416)$ ; $\tau_2=(0,78;0,764)$

В модельній задачі 3.2 наведено розбиття в якому початкові координати центрів були обрані симетрично відносно вісі абсцис. Результати розбиття мають схожість з результатами модельної задачі 3.1, проте їх відмінність полягає саме в оптимальному розташуванні центрів шуканих підмножин. Хочемо зазначити, що багато чисельними результатами числових експериментів, зокрема викладених в роботах [7], [12], [14], [32], [113], [114] продемонстровано, що обрання початкового розташування центрів в задачах про їх оптимальне розміщення має важливий вплив на розв'язність задачі, проте, не впливає на кінцевий оптимальний розв'язок. У більшості задач початкове розташування центрів обрано в початку координат. В модельних задачах 3.1-3.6 початкові координати центрів постачання продукції були обрані відповідно до центрів з модельних задач 2.1-2.6 з метою ініціації саме

уточнення їх розташування в порівнянні з модельними задачами із фіксованими центрами. Такий підхід дає можливість оцінити ефективність та результативність застосування динамічної постановки задачі оптимального розбиття множин з відшукуванням оптимальних координат центрів підмножин, абсолютний та відносний ефект зменшення значення цільового функціоналу.

Вище було сказано про те, що попри більш симетричну структуру розташування початкових центрів підмножин, координати яких підлягають уточненню в модельній задачі 3.2 в порівнянні з модельною задачею 3.1 результати розбиття та значення цільового функціоналу суттєво відрізняються. Порівнюючи результати чисельного експерименту рис.3.2-а) та рис.2.2-а) бачимо, що значення цільового функціоналу для задачі з фіксованими центрами приймає значення 0,255, а для задачі з відшукуванням оптимальних координат центрів 0,224. Маємо абсолютне відхилення рівним 0,031 проте відносне відхилення складає 12,157%, що більш ніж вдвічі більше в порівнянні за парою модельних задач 3.1-2.1. В таблиці 3.4 наведені результати порівняльного аналізу результатів модельної задачі 2.2 та 3.2.

Таблиця 3.4

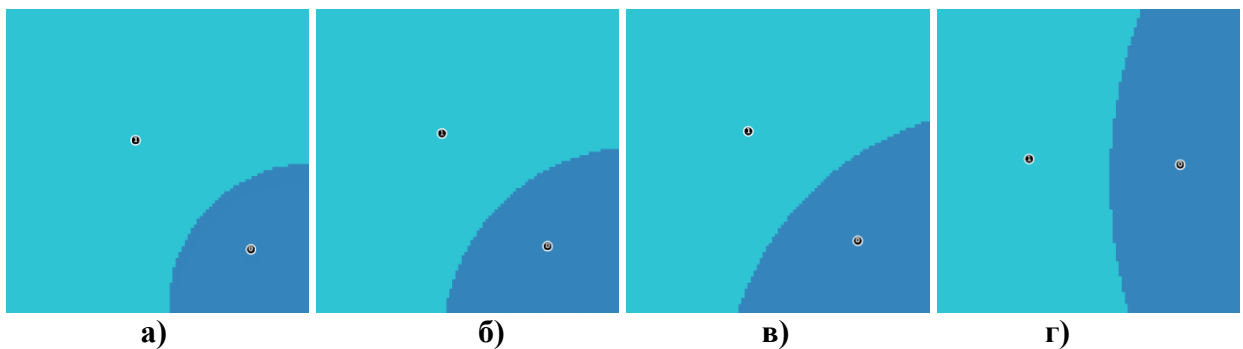
Результати порівняльного аналізу результатів модельних задач 2.2. та 3.2

МД 2.2 $\tau_1(0,1;0,2)$ , $\tau_2(0,8;0,6)$	Значення ЦФ (2.14)	МД 3.2	Координати центрів	Знач. ЦФ (3.10)	Абсол. різниця значень ЦФ	Відносна різниця значення ЦФ
Рис.2.2-а)	0,255	Рис.3.2-а)	$\tau_1 = (0,205;0,205)$ ; $\tau_2 = (0,574;0,574)$	0,224	0,031	-12,157%
Рис.2.2-б)	0,638	Рис.3.2-б)	$\tau_1 = (0,236;0,22)$ ; $\tau_2 = (0,585;0,591)$	0,561	0,077	-12,069%
Рис.2.2-в)	1,18	Рис.3.2-в)	$\tau_1 = (0,274;0,245)$ ; $\tau_2 = (0,605;0,62)$	1,037	0,143	-12,119%
Рис.2.2-г)	1,924	Рис.3.2-г)	$\tau_1 = (0,273;0,273)$ ; $\tau_2 = (0,624;0,624)$	1,679	0,245	-12,734%
Рис.2.2-е)	4,246	Рис.3.2-д)	$\tau_1 = (0,25;0,5)$ ; $\tau_2 = (0,75;0,5)$	3,54	0,706	-16,627%
Рис.2.2-ж)	15,158	Рис.3.2-е)	$\tau_1 = (0,374;0,396)$ ; $\tau_2 = (0,758;0,716)$	11,409	3,749	-24,733%
Рис.2.2-з)	213,187	Рис.3.2-ж)	$\tau_1 = (0,409;0,416)$ ; $\tau_2 = (0,78;0,764)$	109,886	103,301	-48,456%

Як бачимо з результатів наведених в таблиці 3.4 оптимальне розміщення центрів підмножин вже на перших етапах розбиття дозволяє покращити результат більш ніж на 12%. З плином часу ефективність такого розміщення збільшується в рази та при  $T = 10$  сягає майже 50%. При цьому розглянувши рис.3.2 а)-ж) ми можемо бачити, що оптимальні координати центрів розміщені відносно симетрично на області  $\Omega$ . В модельних задачах 3.3-3.6 будуть наведені результати розбиття з більш нерівномірним положенням центрів, що матиме відповідно суттєвий вплив на покращення розв'язку та відповідні значення цільового функціоналу.

### Модельна задача 3.3.

Вхідні дані модельної задачі відповідають параметрам задач 3.1–3.2, за винятком початкових координат центрів  $\tau_1, \tau_2$ . Центри розміщено досить близько один до одного  $\tau_1^0(0,7;0,4)$ ,  $\tau_2^0(0,75;0,35)$ , як наслідок, споживачі, що розташовані на значній відстані від обох центрів, при розбитті, належать до найближчого з них. Крім того, при отриманні оптимального розбиття у випадку  $T = 10$ , значення цільового функціоналу стрімко зростає, що узгоджується з результатами, порівняння пари модельних задач 3.2, 2.2. Такий ефект зумовлений істотним збільшенням вартості перевезення одиниці продукції з часом. Очевидно, що точка початку координат системи (3.3) є нестійкою, і фазова траєкторія, яка прямує до нескінченності, викликає суттєве зростання загальної вартості транспортування, а отже, і значення цільового функціоналу.



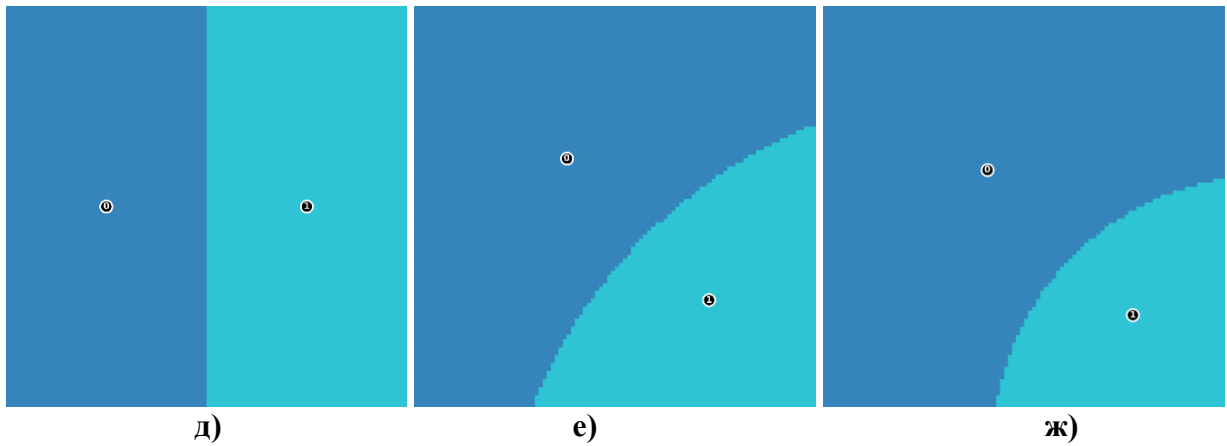


Рис.3.3 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,7;0,4)$ ,  $\tau_2(0,75;0,35)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = 0,5$ ,  $d_1 = 1$ ,  $d_2 = 1,5$  та а)  $T = 0,5$ , б)  $T = 1$ , в)  $T = 1,5$ , г)  $T = 2$ , д)  $T = 3$ , е)  $T = 5$ , ж)  $T = 10$ .

Інші числові параметри чисельного експерименту (рис.3.3) наведені в таблиці 3.5.

Таблиця 3.5

Розгорнуті параметри чисельного експерименту (рис.3.3)

Рис.3.3	Часовий проміжок	Кількість ітерацій	Значення цільового функціоналу	Оптимальні координати центрів $\tau_i$ підмножин $\Omega_i$
а)	[0;0,5]	28	0,224	$\tau_1 = (0,808; 0,21)$ ; $\tau_2 = (0,428; 0,568)$
б)	[0;1]	27	0,561	$\tau_1 = (0,764; 0,22)$ ; $\tau_2 = (0,416; 0,591)$
в)	[0;1,5]	29	1,037	$\tau_1 = (0,762; 0,238)$ ; $\tau_2 = (0,402; 0,598)$
г)	[0;2]	31	1,676	$\tau_1 = (0,803; 0,488)$ ; $\tau_2 = (0,304; 0,507)$
д)	[0;3]	32	3,54	$\tau_1 = (0,25; 0,5)$ ; $\tau_2 = (0,75; 0,5)$
е)	[0;5]	31	11,402	$\tau_1 = (0,380; 0,62)$ ; $\tau_2 = (0,734; 0,267)$
ж)	[0;10]	33	109,874	$\tau_1 = (0,409; 0,591)$ ; $\tau_2 = (0,771; 0,229)$

Як бачимо, визначальні параметри моделі обрані аналогічно модельній задачі 3.2, що дозволяє порівнювати результати розв'язків задач з фіксованими центрами та їх відшукуванням. Графічне представлення результатів суттєвим чином не відрізняється від модельної задачі 3.2 проте, важливим є саме абсолютне та відносне відхилення цільового функціоналу

для відповідної задачі з фіксованими центрами. В таблиці 3.6 наведено результати такого порівняльного аналізу, що в черговий раз підкреслюють доцільність використання алгоритму розв'язання задачі.

Таблиця 3.6

Результати порівняльного аналізу результатів модельної задачі 2.3. та 3.3 з відшукуванням координат центрів підмножин

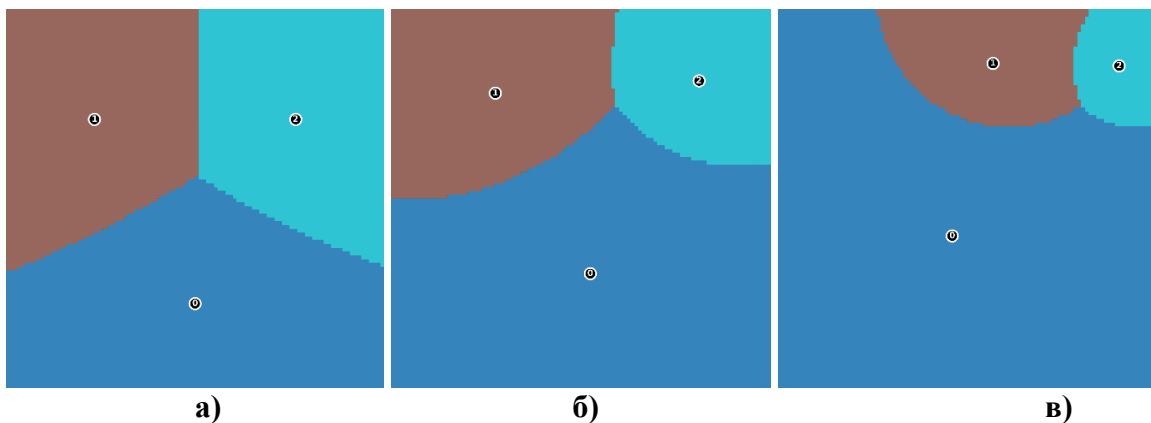
МД 2.3 $\tau_1(0.7;0.4)$ , $\tau_2(0.75;0.35)$	Знач. ЦФ (2.14)	МД 3.3 Рис.3.3	Координати центрів	Значен ня ЦФ (3.10)	Абсолют на різниця значень ЦФ	Відносна різниця значення ЦФ
Рис.2.3-а)	0,299	а)	$\tau_1 = (0,808; 0,21)$ ; $\tau_2 = (0,428; 0,568)$	0,224	0,075	-25,084%
Рис.2.3-б)	0,769	б)	$\tau_1 = (0,764; 0,22)$ ; $\tau_2 = (0,416; 0,591)$	0,561	0,208	-27,048%
Рис.2.3-в)	1,464	в)	$\tau_1 = (0,762; 0,238)$ ; $\tau_2 = (0,402; 0,598)$	1,037	0,427	-29,167%
Рис.2.3-г)	2,449	г)	$\tau_1 = (0,803; 0,488)$ ; $\tau_2 = (0,304; 0,507)$	1,676	0,773	-31,564%
Рис.2.3-е)	5,207	д)	$\tau_1 = (0,25; 0,5)$ ; $\tau_2 = (0,75; 0,5)$	3,54	1,667	-32,015%
Рис.2.3-ж)	16,849	е)	$\tau_1 = (0,380; 0,62)$ ; $\tau_2 = (0,734; 0,267)$	11,402	5,447	-32,328%
Рис.2.3-з)	219,928	ж)	$\tau_1 = (0,409; 0,591)$ ; $\tau_2 = (0,771; 0,229)$	109,874	110,054	-50,041%

Результати наведені в таблиці 3.6 наочно дозволяють бачити ефект про який було зазначено у попередній модельній задачі, адже, попри відносно стає оптимальне розбиття в модельній задачі 3.3 (рис.3.3-а)-ж)), відносне відхилення значення цільового функціоналу задачі з розміщенням центрів на перших ітераціях розв'язання задачі (3.3) сягає більше 25%. Такий ефект, в першу чергу, обумовлений невдалим розміщенням центрів в модельній задачі 2.3 (дивись, наприклад, пари рис.2.3-г), 3.3-г); рис.2.3-е), 3.3-д);). І попри те, що розв'язок модельної задачі 2.3 є оптимальним для заданих центрів вільне їх розміщення за критерієм мінімізації цільового функціоналу (3.10) дозволяє отримати відчутно якісніші результати в порівнянні з задачею з фіксованими центрами.

При  $T = 10$  відносна різниця цільового функціоналу сягає більш ніж 50%, що суттєво не відрізняється від попереднього порівняння пари модельних задач 2.2 та 3.2. Такий ефект зумовлений, насамперед більш повільним зростання відносного відхилення значення функціоналу та існуванням певної межі його значення. На прикладі наступних модельних задач ми проілюструємо емпіричний підхід для визначення межі такого відхилення.

#### Модельна задача 3.4.

В данній модельній задачі кількість центрів дорівнює трьом трьом, а їхнє взаємне розташування характеризується розміщенням на відрізку (див. рис. 3.4-а) з рівними відстанями між першою та другою парами центрів  $\tau_1^0(0,2;0,5)$ ,  $\tau_2^0(0,5;0,7)$ ,  $\tau_3^0(0,85;0,85)$ . Крім того, крок прямокутної сітки встановлено на рівні 0,01, що, зокрема, позитивно впливає на точність обчисленого розв'язку. Параметри  $d_1 = 1; d_2 = 2; d_3 = 3$  підібрані таким чином, що, на відміну від попередніх модельних задач, зменшується відносна різниця у швидкості зміни вартості транспортування. Проте при цьому вартість транспортування для третього центру вже втричі перевищує відповідну вартість для першого. Така конфігурація параметрів зумовлює швидше відхилення фазової траєкторії від стаціонарної точки початку координат, і вже при  $T = 2$  чітко простежується перевага першого центру порівняно з центрами  $\tau_2$  та  $\tau_3$ .



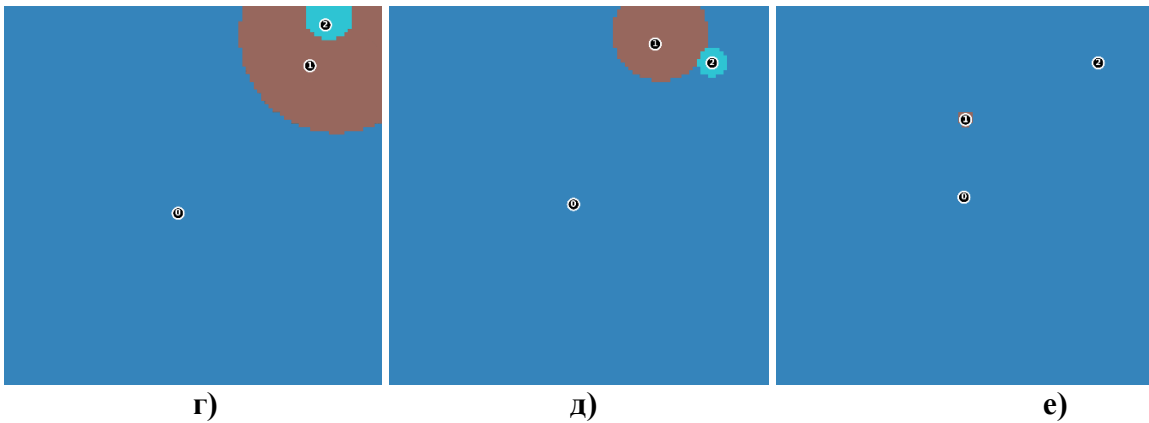


Рис.3.4 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на три підмножини з початковими центрами  $\tau_1^0(0,2;0,5)$ ,  $\tau_2^0(0,5;0,7)$ ,  $\tau_3^0(0,85;0,85)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$  та а)  $T = 0,5$ , б)  $T = 1$ , в)  $T = 1,5$ , г)  $T = 2$ , д)  $T = 3$ , е)  $T = 5$ .

Інші числові параметри чисельного експерименту (рис.3.3) наведені в таблиці 3.7.

Таблиця 3.7

## Розгорнуті параметри чисельного експерименту (рис.3.4)

Рис.3.4	Часовий проміжок	Кількість ітерацій	Значення цільового функціоналу	Оптимальні координати центрів $\tau_i$ підмножин $\Omega_i$
а)	[0;0,5]	38	0,239	$\tau_1=(0,5; 0,223)$ ; $\tau_2=(0,234; 0,709)$ ; $\tau_3 = (0,766; 0,709)$
б)	[0;1]	38	0,742	$\tau_1=(0,525; 0,302)$ ; $\tau_2=(0,274; 0,778)$ ; $\tau_3=(0,81; 0,81)$
в)	[0;1,5]	39	1,677	$\tau_1 = (0,46; 0,401)$ ; $\tau_2 = (0,567; 0,856)$ ; $\tau_3 = (0,9; 0,85)$
г)	[0;2]	37	3,042	$\tau_1 = (0,46; 0,454)$ ; $\tau_2 = (0,809; 0,843)$ ; $\tau_3 = (0,85; 0,95)$
д)	[0;3]	30	8,136	$\tau_1 = (0,485; 0,477)$ ; $\tau_2 = (0,7; 0,9)$ ; $\tau_3 = (0,85; 0,85)$
е)	[0;5]	20	39,972	$\tau_1 = (0,496; 0,496)$ ; $\tau_2 = (0,5; 0,7)$ ; $\tau_3 = (0,85; 0,85)$

На відміну від попередніх модельних задач в модельній задачі 3.4 кількість центрів дорівнює трьом. Початкове розміщення центрів підмножин не є симетричним відносно меж області  $\Omega$ , проте є достатньо рівномірним відносно одне одного. Як бачимо на рис.3.4 а)-е) таке взаємне розміщення центрів не зберігається та обумовлено і обумовлює відповідне мінімальне значення цільового функціоналу. В таблиці 3.8 наведено результати порівняльного аналізу результатів оптимального розбиття модельної задачі 2.4 та модельної задачі 3.4 за результатами якого можна оцінити

ефективність застосування алгоритму для задачі з відшукуванням оптимальних координат центрів підмножин.

Таблиця 3.8  
Результати порівняльного аналізу результатів модельної задачі 2.4. та 3.4 з відшукуванням координат центрів підмножин

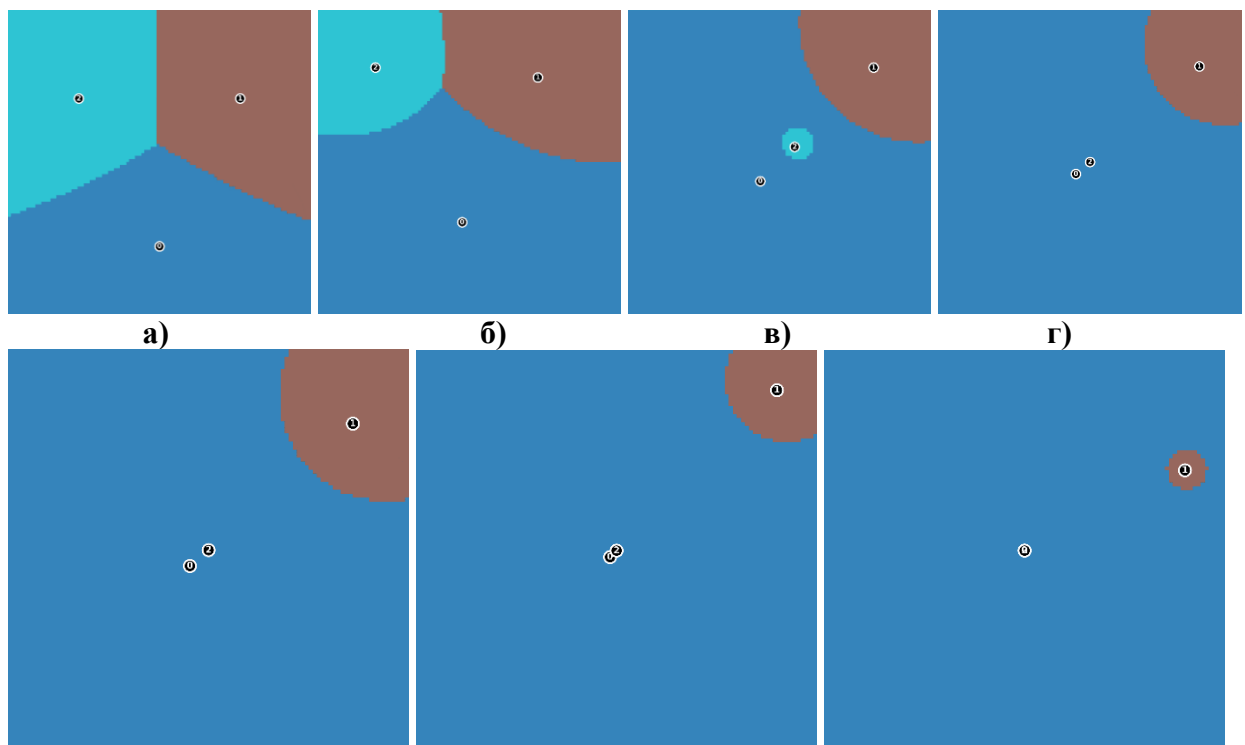
МЗ 2.4 $\tau_1(0,2;0,5)$ , $\tau_2(0,5;0,7)$ , $\tau_3(0,85;0,85)$	Значення ЦФ (2.14)	МЗ 3.4 Рис.3.4	Координати центрів	Значення ЦФ (3.10)	Абсолют на різниця значень ЦФ	Відносна різниця значення ЦФ
Рис.2.4-а)	0,308	а)	$\tau_1 = (0,5; 0,223)$ ; $\tau_2 = (0,234; 0,709)$ ; $\tau_3 = (0,766; 0,709)$	0,239	0,069	-22,403%
Рис.2.4-б)	0,919	б)	$\tau_1 = (0,525; 0,302)$ ; $\tau_2 = (0,274; 0,778)$ ; $\tau_3 = (0,81; 0,81)$	0,742	0,177	-19,260%
Рис.2.4-в)	1,985	в)	$\tau_1 = (0,46; 0,401)$ ; $\tau_2 = (0,567; 0,856)$ ; $\tau_3 = (0,9; 0,85)$	1,677	0,308	-15,516%
Рис.2.4-г)	3,723	г)	$\tau_1 = (0,46; 0,454)$ ; $\tau_2 = (0,809; 0,843)$ ; $\tau_3 = (0,85; 0,95)$	3,042	0,681	-18,292%
Рис.2.4-д)	10,038	д)	$\tau_1 = (0,485; 0,477)$ ; $\tau_2 = (0,7; 0,9)$ ; $\tau_3 = (0,85; 0,85)$	8,136	1,902	-18,948%
Рис.2.4-е)	48,889	е)	$\tau_1 = (0,496; 0,496)$ ; $\tau_2 = (0,5; 0,7)$ ; $\tau_3 = (0,85; 0,85)$	39,972	8,917	-18,239%

Як видно з результатів представлених в таблиці 3.8 відносна різниця значень цільових функціоналів відповідних числових експериментів для модельних задач 2.4 та 3.4 не перевищує 25%, а у більшості випадках і 20%. Слід зазначити, що числовий експеримент для  $T=10$  у даному прикладі не обраховувався, проте відчутно відслідковується тенденція саме зменшення відхилення значень відповідних цільових функціоналів зі зростанням  $T$ . Такий ефект, в першу чергу обумовлений збільшенням кількості центрів (підмножин), що впливає на кількість споживачів, що відносяться до кожного центру. Такий приклад наочно демонструє залежність та граничне значення відносної різниці двох цільових функціоналів при зростанні кількості центрів (підмножин). Ще одним цікавим ефектом є те, що кінцеве оптимальне розбиття модельної задачі 3.4 не повністю, проте схоже на відповідне

оптимальне розбиття модельної задачі 2.4. Саме цей факт суттєвим чином впливає на невеликі абсолютні та відносні показники різниці між цільовими функціоналами та дозволяє припустити, що в модельній задачі 2.4 координати центрів обрані хоч і не симетрично відносно границі області  $\Omega$  та відносно один одного, проте так, що максимально відповідає їх оптимальному розміщенню.

### Модельна задача 3.5.

Вхідні дані відповідають параметрам модельної задачі 3.4, за єдиним винятком, що полягає у початкових координатах центрів виробництва та їх взаємному розташуванню  $\tau_1^0(0,1;0,25)$ ,  $\tau_2^0(0,9;0,7)$ ,  $\tau_3^0(0,5;0,5)$ . Значення швидкості зміни вартості транспортування залишено без змін, що, як і в модельній задачі 3.4, призводить до зміщення межі між областями  $\Omega_1, \Omega_2, \Omega_3$  у напрямку до центрів  $\tau_2$  та  $\tau_3$ . При цьому значення цільового функціоналу в динаміці не демонструє суттєвих відхилень від результатів модельної задачі з фіксованими центрами, де для  $T = 5.0$ ,  $F = 59.269$ , на відміну від відповідного значення модельної задачі 3.4, де при  $T = 5.0$ ,  $F = 40.501$ .





значення цільового функціоналу, що можна бачити в результатах порівняльного аналізу у табл.3.10.

Таблиця 3.10

Результати порівняльного аналізу результатів модельної задачі 2.5. та 3.5 з відшукуванням координат центрів підмножин

МЗ 2.5 $\tau_1(0,1;0,25)$ , $\tau_2(0,9;0,7)$ , $\tau_3(0,5;0,5)$	Значення ЦФ (2.14)	МЗ 3.5 Рис.3.5	Координати центрів	Значення ЦФ (3.10)	Абсолют на різниця значень ЦФ	Відносна різниця значення ЦФ
Рис.2.5-а)	0,98	б)	$\tau_1 = (0,476; 0,302)$ ; $\tau_2 = (0,726; 0,778)$ ; $\tau_3 = (0,19; 0,811)$	0,742	0,238	-24,286%
Рис.2.5-б)	7,113	г)	$\tau_1 = (0,454; 0,46)$ ; $\tau_2 = (0,86; 0,815)$ ; $\tau_3 = (0,5; 0,5)$	3,064	4,049	-56,924%
Рис.2.5-в)	59,269	е)	$\tau_1 = (0,5; 0,5)$ ; $\tau_2 = (0,9; 0,7)$ ; $\tau_3 = (0,5; 0,5)$	40,501	18,768	-31,666%

Як було сказано вище, “неточне” обрання координат розміщення центрів підмножин в модельній задачі 2.5 призвело до суттєвої відмінності у значеннях цільового функціоналу відповідних числових експериментів модельних задач 2.5 та 3.5. Важливо відмітити, що при значних відхиленнях в координатах розташування центрів підмножин відсутня чітка тенденція до зростання або спадання різниці значень цільових функціоналів з плином часу. Зміна вартості транспортування одиниці продукції впливає на потужності підприємств (кількість споживачів) та незалежно від часу визначає оптимальне розміщення центрів підмножин. За результатами наведеними в табл. 3.10 можна бачити, що результати оптимального розбиття зменшують значення цільового функціоналу на 25%-50% в залежності від оптимального розміщення центрів підмножин.

### Модельна задача 3.6.

Вхідні дані відповідають параметрами модельної задачі 3.5. Аналогічно розглядається задача розбиття області, проте цього разу на більшу кількість підмножин. Чотири центри нерівномірно розміщено в межах області:

$\tau_1(0.3;0.8)$ ,  $\tau_2(0.1;0.9)$ ,  $\tau_3(0.4;0.2)$ ,  $\tau_4(0.5;0.6)$ , при цьому швидкість зростання вартості транспортування задана параметрами  $d_1=1.0$ ,  $d_2=2.0$ ,  $d_3=3.0$ ,  $d_4=4.0$ , які суттєво відрізняються між окремими центрами виробництва. У результаті чисельного експерименту отримано відповідне розбиття (рис. 3.6 а)–з)). Як видно, при малих значеннях параметра  $T \in [0,1]$  переважає перший центр виробництва, проте інші центри мають своїх споживачів, при цьому попри відмінність у вартості транспортування між центрами  $\tau_2, \tau_3, \tau_4$  площа підмножин  $\Omega_1, \Omega_2, \Omega_3$  майже однакова. Однак при збільшенні  $T \rightarrow 5$  область майже повністю охоплюється підмножиною  $\Omega_1$ , що відповідає центру  $\tau_1$ , що зумовлено значно нижчою вартістю транспортування одиниці продукції саме з цього центру.

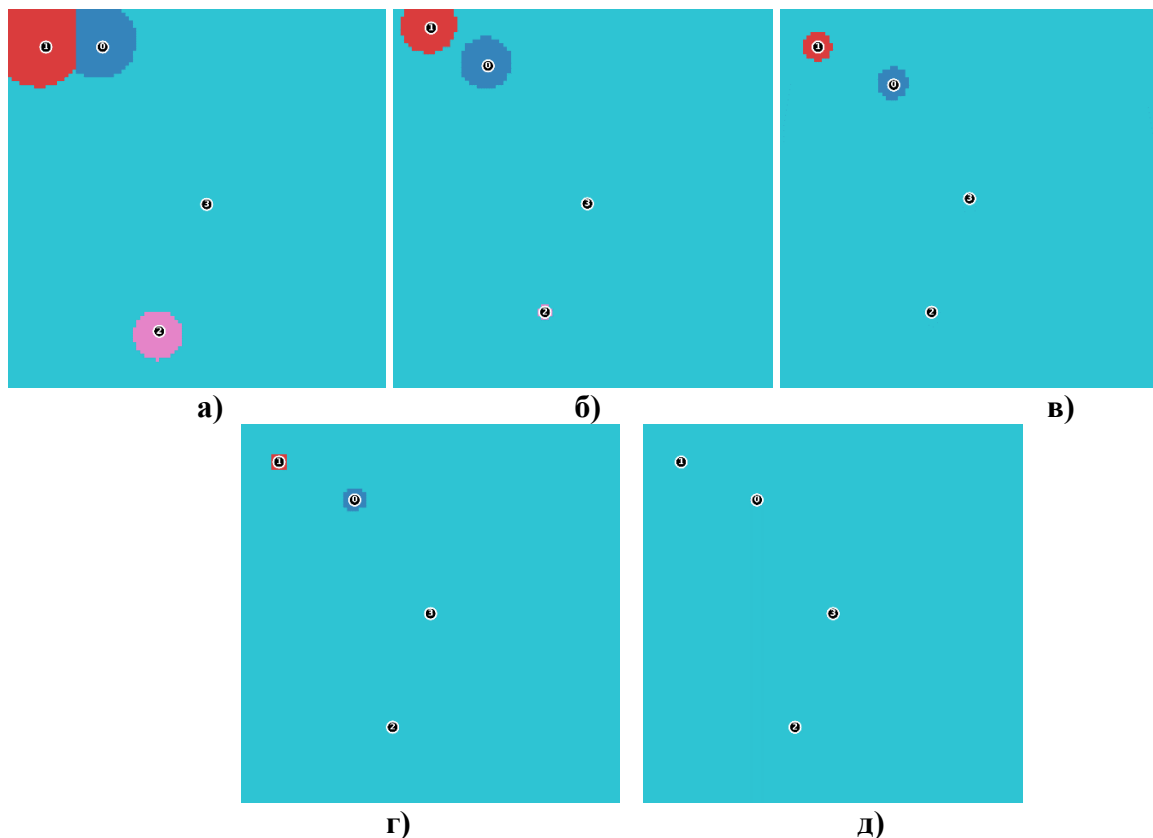


Рис.3.6 Результати розбиття множини  $\Omega$ , що має форму квадрата зі стороною 1, на чотири підмножини з початковими центрами  $\tau_1^0(0,3;0,8)$ ,  $\tau_2^0(0,1;0,9)$ ,  $\tau_3^0(0,4;0,2)$ ,  $\tau_4^0(0,5;0,6)$ ,  $\Delta x = \Delta y = 0,01$ ,  $A_{11} = A_{22} = A_{33} = 1$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $d_4 = 4$  та а)  $T = 0,5$ , б)  $T = 1$ , в)  $T = 1,5$ , г)  $T = 2$ , д)  $T = 3$ .

Інші числові параметри чисельного експерименту (рис.3.6) наведені в таблиці 3.11.

Таблиця 3.11

Розгорнуті параметри числового експерименту наведеного на рис.3.6

Рис.3. 5	Часовий проміжок	Кількість ітерацій	Значення цільового функціоналу	Оптимальні координати центрів $\tau_i$ підмножин $\Omega_i$
а)	[0;0,5]	34	0,183	$\tau_1 = (0,25; 0,9);$ $\tau_2 = (0,1; 0,9);$ $\tau_3 = (0,4; 0,15);$ $\tau_4 = (0,525; 0,485)$
б)	[0;1]	25	0,373	$\tau_1 = (0,25; 0,85);$ $\tau_2 = (0,1; 0,95);$ $\tau_3 = (0,4; 0,2);$ $\tau_4 = (0,511; 0,487)$
в)	[0;1,5]	2	0,572	$\tau_1 = (0,3; 0,8);$ $\tau_2 = (0,1; 0,9);$ $\tau_3 = (0,4; 0,2);$ $\tau_4 = (0,5; 0,5)$
г)	[0;2]	2	0,762	$\tau_1 = (0,3; 0,8);$ $\tau_2 = (0,1; 0,9);$ $\tau_3 = (0,4; 0,2);$ $\tau_4 = (0,5; 0,5)$
д)	[0;3]	2	1,144	$\tau_1 = (0,3; 0,8);$ $\tau_2 = (0,1; 0,9);$ $\tau_3 = (0,4; 0,2);$ $\tau_4 = (0,5; 0,5)$

Наведена модельна задача має більш високий прикладний інтерес, адже містить чотири центри. Така кількість центрів сприяє більш рівномірному розподілу споживачів між підприємствами, за умови відносної рівності цін на транспортування та навіть без такої рівності має більш рівномірну та складену структуру. Одним з найбільш цікавих особливостей такої задачі є максимальна відповідність оптимального розміщення центрів центрам заданим в модельній задачі 2.6, що суттєвим чином матиме своє відображення на значенні цільового функціоналу. Результати порівняльного аналізу модельної задачі 2.6 та 3.6 наведено в таблиці 3.12.

Таблиця 3.12

Результати порівняльного аналізу результатів модельної задачі 2.6. та 3.6 з відшукуванням координат центрів підмножин

МЗ 2.6 $\tau_1(0.3;0.8)$ , $\tau_2(0.1;0.9)$ , $\tau_3(0.4;0.2)$ , $\tau_4(0.5;0.6)$	Знач-ня ЦФ (2.14)	МЗ 3.6 Рис.3.6	Координати центрів	Знач-ня ЦФ (3.10)	Абсолютна різниця значень ЦФ	Відносна різниця значення ЦФ
Рис.2.6-а)	0,459	а)	$\tau_1 = (0,25; 0,9)$ ; $\tau_2 = (0,1; 0,9)$ ; $\tau_3 = (0,4; 0,15)$ ; $\tau_4 = (0,525; 0,485)$	0,183	0,276	-60,131%
Рис.2.6-б)	2,254	б)	$\tau_1 = (0,25; 0,85)$ ; $\tau_2 = (0,1; 0,95)$ ; $\tau_3 = (0,4; 0,2)$ ; $\tau_4 = (0,511; 0,487)$	0,373	1,881	-83,452%
Рис.2.6-в)	5,789	в)	$\tau_1 = (0,3; 0,8)$ ; $\tau_2 = (0,1; 0,9)$ ; $\tau_3 = (0,4; 0,2)$ ; $\tau_4 = (0,5; 0,5)$	0,572	5,217	-90,119%
Рис.2.6-г)	10,832	г)	$\tau_1 = (0,3; 0,8)$ ; $\tau_2 = (0,1; 0,9)$ ; $\tau_3 = (0,4; 0,2)$ ; $\tau_4 = (0,5; 0,5)$	0,762	10,070	-92,956%
Рис.2.6-д)	28,821	д)	$\tau_1 = (0,3; 0,8)$ ; $\tau_2 = (0,1; 0,9)$ ; $\tau_3 = (0,4; 0,2)$ ; $\tau_4 = (0,5; 0,5)$	1,144	27,677	-96,031%

Як було сказано вище, оптимальне розміщення чотирьох центрів суттєвим чином, в порівнянні з задачею з фіксованими центрами, знаходить їх оптимальне місце розташування та при зростанні ціни на транспортування витрати збільшуються не пропорційно, що дозволило отримати розбиття зі значенням цільового функціоналу на порядок меншим в порівнянні з оптимальним розбиттям з фіксованими центрами.

### 3.5 Висновки до розділу 3

У третьому розділі дисертації проведено формалізацію неперервної однопродуктової динамічної задачі оптимального розбиття множин без обмежень, яка передбачає відшукування оптимальних координат центрів підмножин. Задача належить до класу варіаційних динамічних задач з

розташуванням центрів підмножині потребує побудови такої структури підмножин, яка мінімізує заданий цільовий функціонал у часі. В межах цієї постановки розроблено відповідну методологію розв'язання задачі та сформульовано алгоритм, який дозволяє знаходити оптимальні координати центрів залежно від зміни параметрів системи у часі.

У розділі здійснено критичний огляд сучасних підходів до розв'язання подібного класу задач: проаналізовано наявні методи та алгоритми, наведено порівняння їх функціональних можливостей, а також виокремлено переваги й обмеження, як самих існуючих підходів, так і запропонованої в розділі методології. Показано, що в межах обраної формалізації досягається баланс між аналітичною прозорістю та можливістю ефективної чисельної реалізації.

Окрему увагу приділено моделюванню зміни вартості перевезень у часі. Математична модель отримала прикладне обґрунтування в контексті логістичних задач або задач управління потоками ресурсів. На базі побудованої математичної формалізації було сформовано шість модельних задач, у яких системи диференціальних рівнянь описують часову еволюцію вартості транспортування. Їхні розв'язки використовувалися як вихідні дані для обчислення цільового функціоналу, що залежить як від відшукуваних координат центрів, так і від часу.

Для кожної модельної задачі (3.1–3.6) проведено чисельний аналіз із застосуванням модифікованого  $\tau$ -алгоритму, який забезпечив відшукування оптимальних координат центрів підмножин. Таким чином було визначено оптимальне розбиття множини, що мінімізує заданий функціонал. Проведено порівняльний аналіз між двома підходами: із фіксованими центрами наведеними в розділі 2 та з їхнім динамічним визначенням. У результаті аналізу виявлено суттєві відмінності в значеннях функціоналу, зокрема в умовах часової зміни вартості перевезень. Ці результати дозволили оцінити ефективність методу з точки зору як абсолютних, так і відносних показників.

Особливу увагу приділено вибору параметрів модельних задач. Хоча допустимий діапазон значень є досить широким, довільні їх значення не

дозволили б здійснити коректний порівняльний аналіз між динамічними задачами оптимального розбиття фіксованими та відшукуваними центрами. Саме узгоджене моделювання набору параметрів у задачах 3.1–3.6 забезпечило можливість побудови змістовного аналізу.

Чисельні експерименти проводились із різною кількістю центрів, різними стартовими координатами та параметрами моделей. Усі отримані результати дістали детальну прикладну інтерпретацію, що дозволяє оцінити ступінь реалізованості моделі в умовах реальних прикладних сценаріїв. Встановлено високу адекватність розробленої моделі як з точки зору точності, так і з точки зору поведінки функціоналу на тестових прикладах.

Таким чином, результати, представлені в цьому розділі, не лише підтверджують ефективність запропонованого підходу, а й створюють підґрунтя для подальших досліджень. Зокрема, перспективними напрямками є вивчення аналогічних задач із використанням інших базових моделей зміни вартості транспортування продукції, введення додаткових обмежень на розміщення центрів або на доступні ресурси, а також узагальнення на випадок мультипродуктових задач або задач з ієрархічною структурою кластерів.

**Розділ 4. МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІЧНИХ ЗАДАЧ  
ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН ІЗ ЗАДАНИМ  
ПОЛОЖЕННЯМ ЦЕНТРІВ ПІДМНОЖИН В УМОВАХ  
НЕВИЗНАЧЕНОСТІ, МЕТОДИ ЇХ РОЗВ'ЯЗАННЯ ТА БІФУРКАЦІЙНІ  
ВЛАСТИВОСТІ**

**4.1 Математична постановка динамічної задачі оптимального розбиття множин в умовах невизначеності**

Розглянемо неперервну однопродуктову динамічну задачу оптимального розбиття множин з  $E_n$  без обмежень із заданим положенням центрів підмножин [21] у такому вигляді.

Нехай  $\Omega$  – обмежена замкнена вимірна за Лебегом множина в  $n$  - вимірному евклідовому просторі  $E_n$ .

Сукупність вимірних за Лебегом підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega \subset E_n$  (серед яких можуть бути і порожні) назвемо можливим розбиттям цієї множини, якщо  $\bigcup_{i=1}^N \Omega_i = \Omega, mes(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j)$ , де  $N > 0$  – задане натуральне число;  $mes(\cdot)$  – міра Лебега.

Позначимо через  $\sum_{\Omega}^N$  клас всіх можливих розбиттів множини  $\Omega \subset E_n$  на задану кількість  $N$  її вимірних за Лебегом підмножин:

$$\sum_{\Omega}^N = \{ \bar{\omega} \equiv \{ \Omega_1, \dots, \Omega_N \} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, mes(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j \}$$

Далі, через  $\tau_1, \dots, \tau_N$  позначимо сукупність деяких еталонних точок для підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, які назвемо центрами цих підмножин:  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , і будемо вважати, що координати усіх центрів задані.

**Постановка задачі.** Знайти розбиття  $\bar{\omega} = \{ \Omega_1, \dots, \Omega_N \} \in \sum_{\Omega}^N$  множини  $\Omega \subset E_n$  і векторну функцію  $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$ , визначену м.в. для

$x \in \Omega$  при заданому фіксованому наборі центрів  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та всіх  $t \in [0, T]$ , які забезпечують

$$\inf_{\bar{\omega} \in \sum_{\Omega}^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, c(\cdot, \cdot, \cdot)) \quad (4.1)$$

де

$$F(\bar{\omega}, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad (4.2)$$

за наявності умов

$$\begin{aligned} \frac{dc_i(x, \tau_i, t)}{dt} &= \sum_{j=1}^N A_{ij} \cdot f_j(c_j(x, \tau_j, t)), 0 \leq t \leq T \\ c_i(x, \tau_i, t_0) &= c_{0i}(x, \tau_i), i = 1, \dots, N \end{aligned} \quad (4.3)$$

м.в. для  $x \in \Omega$ , при фіксованих  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , та умови замкненості системи

$$\sum_{i=1}^N A_{ij} = 1, j = 1, \dots, N \quad . \quad (4.4)$$

Тут  $c_i(x, \tau_i, t), i = 1, \dots, N$ , – шукані дійсні функції, визначені на  $\Omega \times \Omega \times [0, T]$ , які при будь-якому фіксованому  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i$  неперервно диференційовані за аргументом  $t$  на відрізку  $[0, T]$  м.в. для  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ , обмежені та вимірні за аргументом  $x$  на  $\Omega$  для всіх  $t \in [0, T]$ ;  $m(x, \tau_i), c_{0i}(x, \tau_i)$  – задані дійсні функції, визначені на  $\Omega \times \Omega$ , обмежені та вимірні за аргументом  $x \in \Omega$  при будь-якому фіксованому  $\tau_i \in \Omega_i$  для всіх  $i = 1, \dots, N$  (зокрема,  $m(x, \tau_i)$  може відігравати роль метрики на  $\Omega \times \Omega$ );  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , – задані дійсні функції, неперервні та ліпшицеві в області свого визначення;  $\rho(x)$  – задана невід'ємна функція, обмежена та вимірна на  $\Omega$ ;  $a_i, i = 1, \dots, N$ , – задані, як правило, невід'ємні числа;  $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$ , – задані числові параметри;  $T > 0$  та  $t_0 \in [0, T]$  задані.

Тут та в подальшому інтеграли розуміються в сенсі Лебега. Будемо вважати, що міра множини межових точок підмножин  $\Omega_1, \dots, \Omega_N$  дорівнює нулю.

Пару  $(\bar{\omega}^*, c^*(x, \tau, t))$ , що доставляє функціоналу (4.2) мінімальне значення на множині  $\sum_{\Omega}^N \times L_2^N(\Omega \times \Omega \times [0, T])$  за умов (4.3), (4.4), назвемо оптимальним розв'язком задачі (4.1)–(4.4). При цьому розбиття  $\bar{\omega}^* = \{\Omega_1^*, \dots, \Omega_N^*\} \in \sum_{\Omega}^N$  назвемо оптимальним розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин, а векторну функцію  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)) \in L_2^N(\Omega \times \Omega \times [0, T])$  – оптимальною фазовою траєкторією динамічної системи в задачі (4.1)–(4.4).

З предметної точки зору, незалежна змінна  $t \in [0, T]$  у наведеній математичній постановці динамічної задачі оптимального розбиття може відігравати роль часової змінної, а  $T > 0$  та  $t_0 \in [0, T]$  – задані кінцевий та початковий моменти часу у досліджуваному динамічному процесі відповідно. Так, функції  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , у диференціальних зв'язках (4.3), що відображають динаміку цін на транспортування, можуть мати різний вигляд в залежності від предметного змісту, який в неї вкладається, наприклад: модель інфляції/дефляції відображає схильність цін до постійного (експоненціального) зростання/спадання

$$f_i(c_i(x, \tau_i, t)) = d_i \cdot c_i(x, \tau_i, t), x \in \Omega, \tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = \overline{1, N}, 0 \leq t \leq T \quad (4.5)$$

**Метод розв'язання задачі.** Нехай  $\{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$  – деяке можливе розбиття вихідної замкненої обмеженої вимірної множини  $\Omega \subset E_n$  на  $N$  її вимірних за Лебегом неперетинних підмножин. Відповідно до теорії оптимального розбиття множин [21], введемо у розгляд на множині  $\Omega \subset E_n$  характеристичні функції  $\lambda_1(x), \dots, \lambda_N(x)$  підмножин  $\Omega_1, \dots, \Omega_N$ :

$$\lambda_i(x) = \begin{cases} 1, & x \in \Omega_i, \\ 0, & x \in \Omega \setminus \Omega_i, \end{cases} \quad i = 1, \dots, N \quad (4.6)$$

Тоді у термінах характеристичних функцій  $\lambda_1(x), \dots, \lambda_N(x)$  задачу (4.1)–(4.4) можна записати так: знайти векторні функції  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x)), x \in \Omega$  і  $c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)), x \in \Omega$ , при фіксованих  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та  $t \in [0, T]$ , які забезпечують

$$\inf_{\lambda(\cdot) \in \Gamma; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) \quad (4.7)$$

де функціонал  $J(\lambda(\cdot), c(\cdot, \cdot, \cdot))$  має вигляд

$$J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega} (c_i(x, \tau_i, t) m(x, \tau_i) + a_i) \rho(x) \lambda_i dx dt, \quad (4.8)$$

$$\Gamma_1 = \{ \lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) : \sum_{i=1}^N \lambda_i(x) = 1 \text{ м.в. для } x \in \Omega; \\ \lambda_i(x) = 0 \vee 1 \text{ м.в. для } x \in \Omega, i = 1, \dots, N \}$$

за умов (4.3)-(4.4).

У [21] доведено, що перша компонента  $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_N^*(x))$  оптимального розв'язку такої задачі м.в. для  $x \in \Omega$  і для кожного  $i = 1, \dots, N$  має вигляд

$$\lambda_i^*(x) = \begin{cases} 1, \text{ якщо виконується умова (4.10)} \\ 0, \text{ в іншому випадку} \end{cases} \quad (4.9)$$

$$\int_0^T (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dt = \min_{k=1, N} \left( \int_0^T (c_k(x, \tau_k, t) \cdot m(x, \tau_k) + a_k) \rho(x) dt \right) \quad (4.10)$$

при заданих  $\tau_i \in \Omega_i, i = 1, \dots, N$ .

У задачі (4.7)-(4.8) за умов (4.3)-(4.4) передбачалося, що усі параметри задані константами. Однак на практиці точні значення цих констант, як правило, невідомі або задані лінгвістично (нечітко). Розглянемо більш детально параметри  $a_1, \dots, a_N$  у цільовому функціоналі (4.8). Для виробничо-транспортних задач ці параметри задають постійні витрати підприємств. Залежно від обставин постійні витрати можуть включати витрати з експлуатації заводу, страховку і, можливо, утримання мінімального числа працівників. Ці витрати постійні і залишаються незмінними незалежно від обсягу випуску фірми. Зрозуміло, що у багатьох практичних виробничо-транспортних задачах величини цих параметрів (постійних витрат) залежать від багатьох факторів реального процесу, не врахованих у наведеній в роботі моделі. Модель з фіксованими значеннями параметрів  $a_1, \dots, a_N$  може виявитися занадто «грубою», тому що на практиці відомими бувають не конкретні точні значення параметрів, а множини їх можливих значень.

Для зняття невизначеності при завданні параметрів  $a_1, \dots, a_N$  у цільовому функціоналі (4.8) розглядатимемо їх як нечіткі змінні, що, в свою чергу, залежать від нечітких факторів  $\gamma = (\gamma_1, \dots, \gamma_n)$ . Такими факторами для постійних витрат виробництва можуть бути вартості засобів виробництва, вартості трудових, природних ресурсів та ін. Урахування подібної додаткової інформації ускладнює вихідну математичну модель, але, тим не менше, вона може виявитися прийнятно більш точною та адекватною, з прикладної точки зору, тому що враховує вплив додаткових факторів на параметри  $a_1, \dots, a_N$ .

Для формалізації нечіткості в подальшому будемо використовувати нейро-нечіткі технології, описані в роботі [30]. Для цього параметри  $a_i, i = 1, \dots, N$ , представимо як параметри, що залежать від нечітких факторів  $\gamma_j, j = 1, \dots, n$ , у вигляді

$$a_i \equiv a_i(\gamma_1, \dots, \gamma_n) \quad (4.11)$$

Тоді цільовий функціонал (4.8) можна записати з нечіткими параметрами (4.11) у такому вигляді:

$$J(\lambda(\cdot), c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega} (c_i(x, \tau_i, t) m(x, \tau_i) + a_i(\gamma_1, \dots, \gamma_n)) \rho(x) \lambda_i dx dt \quad (4.12)$$

## 4.2 Метод розв'язання динамічної задачі ОРМ з нечітким параметром у цільовому функціоналі

У теорії оптимального розбиття множин (ОРМ) розроблено єдиний підхід до розв'язання нечітких задач ОРМ, у яких нечіткість може бути наявна у елементах задачі, які не підлягають оптимізації, тобто з нечіткими початковими даними задачі.

Єдиний підхід до розв'язання задач ОРМ з нечіткими вхідними даними заснований на застосуванні математичного апарату теорії ОРМ і нейро-нечітких технологіях [21]. Спочатку для зняття нечіткості в обмеженнях застосовується метод нейро-нечіткої ідентифікації складних нелінійних залежностей, а потім застосовується математичний та алгоритмічний апарати

розв'язання задач ОРМ в умовах визначеності. Цим підтверджується універсальність теорії ОРМ, а саме, моделі і методи розв'язання неперервних задач оптимального розбиття можуть бути узагальнені на випадок нечіткого завдання початкових даних задачі.

Такий підхід до розв'язання нечітких задач ОРМ, у яких початкові дані задачі можуть бути неточні, неповні, недовизначені, застосуємо для розв'язання найпростішої динамічної задачі ОРМ з нечіткими параметрами, що входить до цільового функціоналу (4.12).

### Нейро-нечітка ідентифікація нечітких параметрів

Позначимо, для спрощення опису методу нейронечіткої ідентифікації, кожний нечіткий параметр  $a_i, i=1, \dots, N$ , як  $a$  та розглянемо функціональну залежність виходу  $a$  від входів  $\gamma_1, \dots, \gamma_n$  у вигляді:

$$a = a(\gamma), \text{ де } \gamma = (\gamma_1, \dots, \gamma_n). \quad (4.13)$$

Для задачі ідентифікації передбачаються відомими:

1) області визначення входів і область зміни виходу для (4.13):

$$Y_i = [\gamma_{\min_i}, \gamma_{\max_i}] - \text{області визначення вхідних змінних } \gamma_i, i = 1, \dots, n, \quad (4.14)$$

$$D = [a_{\min}, a_{\max}] - \text{область зміни вихідної змінної } a; \quad (4.15)$$

2) експертно-експериментальна інформація про залежність (4.13) у вигляді вибірки з  $M$  пар даних про входи та вихід об'єкта  $\{\gamma^{(m)}_{\text{exp}}, a^{(m)}_{\text{exp}}\}$ , де  $\gamma^{(m)}_{\text{exp}} = \{\gamma^{(m)}_{1_{\text{exp}}}, \gamma^{(m)}_{2_{\text{exp}}}, \dots, \gamma^{(m)}_{n_{\text{exp}}}\}$  - вхідний вектор в  $m$ -ій парі,  $m = 1, \dots, M$ .

Нейро-нечітка ідентифікація складної, як правило нелінійної, функціональної залежності складається з двох етапів [21]:

**перший етап** (структурна ідентифікація) - побудова нечіткої моделі об'єкта (4.13) у вигляді нечіткої продукційної моделі за експертно-експериментальними даними про взаємозв'язок виходу  $a$  та входів  $\gamma_1, \dots, \gamma_n$ , як узгодженої множини окремих нечітких продукційних правил «ЯКЩО – ТО»;

**другий етап** (параметрична ідентифікація) - настройка параметрів нечіткої моделі з метою мінімізації відхилення результатів моделювання від експериментальних даних.

Перший етапнейро-нечіткої ідентифікації об'єкта (4.13) з декількома виходами і одним виходом складається з таких блоків: фазифікація, нечітке логічне виведення, дефазифікація.

У блоці фазифікація для знаходження залежності (4.13) в явному вигляді будемо розглядати вхідні змінні  $\gamma_1, \dots, \gamma_n$  і вихідну змінну  $a$  як лінгвістичні змінні, задані на універсальних множинах (4.14)-(4.15) відповідно. Для оцінки лінгвістичних змінних  $a$  та  $\gamma_1, \dots, \gamma_n$  будемо використовувати терми з таких терм-множин:

$D = \{D_k\}$  – терм-множина змінної  $a$ , де  $D_k$  –  $k$ -й лінгвістичний терм змінної  $a$ ,  $k = 1, 2, \dots, L$ ,  $L$  – число різних класів виходу  $a$ . Для кожного класу  $D_k$  виберемо «найбільш характерного представника»  $d_k \in D_k$  і назвемо його центром класу;

$Y_i = \{Y_{ir}\}$  – терм-множина змінної  $\gamma_i$  де  $Y_{ir}$  –  $r$ -й лінгвістичний терм змінної  $\gamma_i$ ,  $i = 1, \dots, n$ ,  $r = 1, \dots, p_i$ ,  $p_i$  – кількість термів в терм-множині  $Y_i$  змінної  $\gamma_i$ . Причому, в загальному випадку,  $p_1 \neq p_2 \neq \dots \neq p_n$ .

Значення лінгвістичних термів  $D_k$  та  $Y_{ir}$  отримаємо на основі експертно-лінгвістичної інформації про об'єкт, що моделюється (4.13).

Лінгвістичні терми  $Y_{ir} \in Y_i$  та  $D_k \in D$  розглядаються як нечіткі множини, задані на універсальних множинах, визначених в (4.14)-(4.15) відповідно. Кожний з нечітких термів  $Y_{ir} \in Y_i$  представляє відповідну нечітку підмножину на множині значень  $\gamma_i$ . А кожному нечітку множині, в свою чергу, задамо її узагальненою дзвоною функцією належності  $\mu_{ir}(\gamma_i)$ , яку представимо у вигляді:

$$\mu_{ir}(\gamma_i) = \frac{1}{1 + \left| \frac{\gamma_i - b_{ir}}{c_{ir}} \right|^{2t_{ir}}}, i = 1, \dots, n, r = 1, \dots, p_i \quad (4.16)$$

де  $\mu_{ir}(\gamma_i)$  – число у діапазоні  $[0,1]$ , яке характеризує суб'єктивну міру відповідності значення  $\gamma_i$  нечіткому терму  $Y_{ir}$ ,  $b_{ir}, c_{ir}, t_{ir}$  – параметри, які спочатку обираються експертом, а потім налаштовуються на

експериментальні дані:  $b_{ir}$  – координата максимуму функції  $\mu_{ir}(\gamma_i)$ , причому  $\mu_{ir}(b_{ir})=1$ ,  $c_{ir}$  – коефіцієнт концентрації-розтягнення функції  $\mu_{ir}(\gamma_i)$ ,  $t_{ir}$  – коефіцієнт крутизни функції  $\mu_{ir}(\gamma_i)$ ,  $t_{ir} > 0$ .

Провівши операцію побудови нечіткої моделі по вхідним експертним даним, отримаємо нечітку модель у вигляді:

$$a = F(\gamma, B, C, T, W) \quad (4.17)$$

де  $\gamma = (\gamma_1, \dots, \gamma_n)$  – вхідний вектор,

$W = \{w_j^k\}, j = 1, \dots, s_k, k = 1, \dots, L$ , набір вагових коефіцієнтів правил,  
 $B = \{b_{ij}^k\}, C = \{c_{ij}^k\}, T = \{t_{ij}^k\}$  – набори параметрів настройки функцій належності,  
 $i = 1, \dots, n, j = 1, \dots, s_k, k = 1, \dots, L$ ;

$F$  – функція зв'язку вхід-вихід, яка включає перетворення.

Якщо побудована модель (4.17) недостатньо точно описує модельований об'єкт (4.13), то необхідно налаштувати її, тобто знайти такі параметри  $B, C, T, W$ , які мінімізують відхилення між модельними (теоретичними, отриманими за моделлю (4.17)) і експериментальними даними.

Перейдемо до етапу настройки нечіткої моделі (4.17). Тоді в термінах математичного програмування завдання настройки нечіткої моделі може бути сформульована таким чином: знайти вектор  $(B, C, T, W)$ , який забезпечує

$$\|F(\gamma, B, C, T, W) - a_{\text{exp}}\| \rightarrow \min_{B, C, T, W} \quad (4.18)$$

Позначимо вектор  $Q = (B, C, T, W)$  і запишемо задачу оптимізації у вигляді:

$$\varphi(Q) \rightarrow \min_Q, \quad (4.19)$$

де  $\varphi(Q) = \|a - a_{\text{exp}}\|$ ,  $a$  – результати, розраховані за моделлю (4.17),  $a_{\text{exp}}$  – експериментальні дані.

Тут норма  $\|\cdot\|$  визначається як евклідова метрика

$$\|a - a_{\text{exp}}\| = \frac{1}{M} \sqrt{\sum_{m=1}^M (a^{(m)} - a_{\text{exp}}^{(m)})^2}, \quad (4.20)$$

де  $a^{(m)}_{\text{exp}}$  – значення вихідної змінної  $a$  з  $m$ -ої пари експериментальної вибірки  $\{\gamma^{(m)}_{\text{exp}}, a^{(m)}_{\text{exp}}\}, m = 1, \dots, M$ ,  $a^{(m)}$  – результати, розраховані за моделлю для значень вхідних змінних з  $m$ -ої пари експериментальної вибірки.

Для розв’язання задачі оптимізації (4.19) застосуємо метод мінімізації з розтягненням простору в напрямку різниці двох послідовних узагальнених антиградієнтів ( $r$ -алгоритм Шора [21]). Задача оптимізації (4.19) може виявитися нелінійною, тому застосування класичних градієнтних методів, що вимагають гладкості функцій, може викликати проблеми в збіжності цих методів.

В результаті розв’язання задачі оптимізації (4.19) отримуємо вектор  $Q^* = (B^*, C^*, T^*, W^*)$ , який доставляє мінімальне значення цільовій функції  $\varphi(Q)$ . Тобто отримуємо такі значення  $W^* = \{w_j^{k*}\}$  – для набору вагових коефіцієнтів правил і наборів параметрів  $B^* = \{b_{ij}^{k*}\}, C^* = \{c_{ij}^{k*}\}, T^* = \{t_{ij}^{k*}\}$  функцій належності, для яких відхилення вигляду (4.20) між експериментальними і модельними даними, отриманими після настройки нечіткої моделі, досягає мінімального значення.

Запишемо результуючу нейронечітку ідентифікацію нечіткого параметра  $a$ , яку далі будемо використовувати для розв’язання поставленої задачі, у вигляді таких відношень:

$$a(\gamma) = \frac{\sum_{k=1}^L d_k \cdot \mu_{D_k}^*(a)}{\sum_{k=1}^L \mu_{D_k}^*(a)}, \quad \gamma \in Y = Y_1 \times \dots \times Y_i \times \dots \times Y_n, \quad (4.21)$$

де

$$\mu_{D_k}^*(a) = \begin{cases} \sum_{j=1}^{S_k} p_j^{k*}(\gamma_1, \gamma_2, \dots, \gamma_n), & \text{якщо } \sum_{j=1}^{S_k} p_j^{k*}(\gamma_1, \gamma_2, \dots, \gamma_n) \leq 1 \\ 1 & \text{інакше} \end{cases} \quad (4.22)$$

$$p_j^{k*}(\gamma_1, \gamma_2, \dots, \gamma_n) = w_j^{k*} \prod_{i=1}^n \mu_{ij}^{k*}(\gamma_i) \quad (4.23)$$

$$\mu_{ij}^{k*}(\gamma_i) = \frac{1}{1 + \left( \frac{\gamma_i - b_{ij}^{k*}}{c_{ij}^{k*}} \right)^2}, i=1, \dots, n, j=1, \dots, s_k, k=1, \dots, L. \quad (4.24)$$

В співвідношеннях (4.21)-(4.24) значення  $\mu_{ij}^{k*}(\gamma_i)$ ,  $p_j^{k*}(\gamma_1, \gamma_2, \dots, \gamma_n)$ ,  $\mu_{D_k}^*(a)$  розраховуються при оптимальних значеннях параметрів  $W^* = \{w_j^{k*}\}$ ,  $B^* = \{b_{ij}^{k*}\}$ ,  $C^* = \{c_{ij}^{k*}\}$ , отриманих після настройки нечіткої моделі (4.17).

### 4.3 Алгоритм роз'язання динамічної задачі ОРМ з нечіткими параметрами у цільовому функціоналі

Крок 1: Множину  $\Omega$  включаємо в  $n$ -вимірний паралелепіпед  $\Pi$ , сторони якого паралельні осям декартової системи координат. Паралелепіпед  $\Pi$  покриваємо прямокутною сіткою.

Крок 2: Задаємо значення функцій  $c_{0i}(x, \tau_i), i=1, \dots, N$ , з початкових умов (4.3) в вузлах сітки.

Крок 3: Покриваємо відрізок  $[0, T]$  сіткою з кроком  $h_i$ .

Крок 4: Для кожного вузла сітки та для кожного  $\tau_i, i=1, \dots, N$  розв'язуємо задачу Коші для системи ЗДР (4.3) на часовому відрізку  $[0, T]$  та знаходимо функції  $c_i^*(x, \tau_i, t), i=1, \dots, N$ .

Крок 5: Розраховуємо значення параметрів  $a_i, i=1, \dots, N$ , за формулами (4.21)-(4.24).

Крок 6: Визначаємо характеристичну функцію  $\lambda^*(x)$  у вузлах сітки з пункту 1 за формулами (4.9), (4.10) при знайдених  $c_i^*(x, \tau_i, t), i=1, \dots, N$  та відновлених у пункті 5 значеннях нечітких параметрів  $a_i, i=1, \dots, N$ .

Крок 7: Знаходимо мінімальне значення цільового функціонала (4.8) при знайдених  $\lambda^*(x)$  та  $c_i^*(x, \tau_i, t), i=1, \dots, N$ .

Алгоритм розв'язання неперервної однопродуктової динамічної задачі оптимального розбиття множин з  $E_n$  без обмежень з заданим положенням центрів та нечіткими параметрами у цільовому функціоналі підмножин описаний.

#### 4.4 Розв'язання модельних тестових задач

Описана вище задача була розв'язана для тестового прикладу нескінченновимірної задачі розбиття заданої області з фіксованими центрами підприємств, що виробляють однорідну продукцію, для розподілення в цій області із заданою щільністю споживачів, у постановці наведеній нижче.

Споживачі деякої однорідної продукції, що виробляється двома підприємствами, неперервно розподілені в області  $\Omega = \{x = (x^{(1)}, x^{(2)}) : 0 \leq x^{(1)} \leq 1; 0 \leq x^{(2)} \leq 1\}$ , задані координати розміщення підприємств  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$ , для підприємств задана система, відповідно до якої змінюється ціна на транспортування

$$\begin{cases} \frac{\partial c_1(x, \tau_1, t)}{\partial t} = A_{11} \cdot d_1 \cdot c_1(x, \tau_1, t) + (1 - A_{22}) \cdot d_2 \cdot c_2(x, \tau_2, t) \\ \frac{\partial c_2(x, \tau_2, t)}{\partial t} = (1 - A_{11}) \cdot d_1 \cdot c_1(x, \tau_1, t) + A_{22} \cdot d_2 \cdot c_2(x, \tau_2, t) \end{cases}, \quad 0 \leq t \leq T, \quad (4.25)$$

з початковими умовами

$$c_{01} = 2, \quad c_{02} = 1 \quad (4.26)$$

У якості  $m(x, \tau_i)$  використана евклідова метрика.

Розв'язок задачі (4.25)-(4.26) описує вартість транспортування одиниці продукції з  $i$ -го підприємства до споживача з координатами  $(x^{(1)}, x^{(2)})$ . Відомий попит  $\rho(x^{(1)}, x^{(2)})$  на продукцію для кожного пункту споживання з координатами  $(x^{(1)}, x^{(2)})$ . Для простоти вважається  $\rho(x^{(1)}, x^{(2)}) \equiv 1, \forall x = (x^{(1)}, x^{(2)}) \in \Omega$ .

Множину споживачів можна розбити на зони обслуговування  $\Omega_i$  споживачів  $i$ -им пунктом виробництва так, щоб

$$\Omega_1 \cup \Omega_2 = \Omega, \quad mes(\Omega_1 \cap \Omega_2) = 0 \quad (4.27)$$

Зазначимо, що потужність  $i$ -го виробника визначається сумарним попитом споживачів, що належать  $\Omega_i$ :

$$\iint_{\Omega_1} \rho(x) dx = b_1, \quad \iint_{\Omega_2} \rho(x) dx \leq b_2, \quad b_1 + b_2 = 1. \quad (4.28)$$

Необхідно розбити множину споживачів  $\Omega$  так, щоб мінімізувати функціонал (4.12) сумарних витрат на виробництво продукції та її доставку до споживача за умов (4.3)-(4.4) та  $N = 2$ .

Розглянемо більш детально параметри  $a_i, i=1,2$ , у цільовому функціоналі (4.12). Як було зазначено вище, це постійні витрати  $i$ -го підприємства. У модельній задачі пропонується провести штучний експеримент щодо параметрів  $a_i$ . З одного боку, побудувати оптимальне динамічне розбиття регіону на підмножини для точних значень параметрів  $a_i$ . З іншого боку, побудувати оптимальні розбиття для відновлених значень параметрів  $a_i$  на основі статистичної вибірки за допомогою методу нейронечіткої ідентифікації (до та після настройки нечіткої моделі) та порівняти отримані оптимальні динамічні розбиття.

На практиці часто точне значення постійних витрат підприємства (параметр  $a_i$ ) розраховують як лінійну комбінацію факторів, які на них впливають. Розглянемо, наприклад, такі фактори:

$\gamma_1$  - вартості засобів виробництва (виробничі потужності та засоби праці);

$\gamma_2$  - вартості трудових ресурсів;

$\gamma_3$  - вартості природних ресурсів (сировинних і енергетичних).

Обчислення параметрів  $a_i, i=1,2$ , як чітких значень, проводять за формулою:

$$a_i = k_{i1}\gamma_1 + k_{i2}\gamma_2 + k_{i3}\gamma_3. \quad (4.29)$$

Нехай для деяких точних значень  $k_{i1}, k_{i2}, k_{i3}, \gamma_1, \gamma_2, \gamma_3$  (де  $k_{ij}$  - вагові коефіцієнти, значення факторів  $\gamma_i$  вимірюються в умовних одиницях,  $i=1,2; j=1,2,3$ ) з (4.29) отримані наступні точні значення параметрів:  $a_1 = 3; a_2 = 1$ .

Зауважимо, що залежність (4.29) на практиці, як правило, невідома та може бути складною, нелінійною. Тому виконаємо відновлення значень нечітких параметрів  $a_i$  на основі експертно-експериментальної інформації за допомогою нейро-нечіткої ідентифікації (до  $i$  після настройки). Опишемо відновлення значення параметра  $a_i$  на основі експертно-експериментальної інформації за допомогою методу нейро-нечіткої ідентифікації, та зауважимо, що значення параметра  $a_2$  можуть бути відновлені аналогічно.

Для побудови математичної моделі залежності параметра постійних витрат  $a_1$  від факторів  $\gamma_1, \gamma_2, \gamma_3$  по наявним даним, які наведені у табл. 4.1, розглянемо цю залежність у вигляді

$$a_1 = a_1(\gamma_1, \gamma_2, \gamma_3), \quad \gamma_{\min_i} \leq \gamma_i \leq \gamma_{\max_i}, \quad i = 1, 2 \quad (4.30)$$

Таблиця 4.1

Вхідні дані для відновлення значення параметра  $a_1$ 

Номер у вибірці	$a_1$	Вхідні змінні			Номер у вибірці	$a_1$	Вхідні змінні		
		$\gamma_1$	$\gamma_2$	$\gamma_3$			$\gamma_1$	$\gamma_2$	$\gamma_3$
1	3,00	15	12	11	5	3,07	5	4	10
2	1,00	7	9	7	6	1,15	1	5	12
3	2,96	3	1	8	7	0,87	4	3	4
4	2,69	11	12	15	8	1,34	9	7	5

Виконаємо *перший* етап методу нейро-нечіткої ідентифікації.

*Фазифікація.* Для побудови термів для змінних  $\gamma_1, \gamma_2, \gamma_3$  моделі, візьмемо відрізки, правою границею кожного з яких буде найбільше значення кожної змінної, а лівою – найменше. Розділимо це відрізки на три рівні частини, яким відповідатимуть нечіткі терми – *низький* (Н), *середній* (С), *високий* (В). Отримані відрізки наведені в табл. 4.2.

Таблиця 4.2

Терми вхідних змінних

	$\gamma_1$	$\gamma_2$	$\gamma_3$
Н	[0;5)	[0;4)	[0;5)
С	[5;10)	[4;8)	[5;10)
В	[10, 15]	[8, 12]	[10, 15]

Множину значень  $a_1$  було розділено на три класи  $D_1$ ,  $D_2$  та  $D_3$  з центрами  $d_1 = 0,8$ ;  $d_2 = 2$  та  $d_3 = 3,1$ , відповідно, таким чином, щоб до кожного класу потрапили близькі значення вихідної змінної  $a_1$ .

*Нечітке логічне виведення.* Сформуємо нечітку продукційну базу знань, яка є сукупністю нечітких експертно-лінгвістичних правил типу ЯКЦО-ТО, які зв'язують між собою лінгвістичні оцінки вхідних змінних ( $\gamma_1, \gamma_2, \gamma_3$ ) та вихідної змінної  $a_1$ . Лінгвістичні оцінки будуть отримані на основі наведеної у табл. 4.1 статистичної інформації.

Виходячи з наявних даних, побудованих нами нечітких термів (табл. 4.2) та лінгвістичних висловлювань типу ЯКЦО-ТО можна побудувати нечітку базу знань, компактний вид якої представлено в табл. 4.3

Таблиця 4.3

Компактний вигляд нечіткої бази знань

Номер класу вихідної змінної	Номер у вибірці	Номер правила, $p$	Вага правила до настройки, $w_p$	Вага правила після настройки, $w_p^*$	Вхідні змінні		
					$\gamma_1$	$\gamma_2$	$\gamma_3$
$k = 1$	3	1	1	0,99885	В	В	С
	7	2	1	0,28560	С	С	С
$k = 2$	2	3	1	0,74009	Н	Н	С
	4	4	1	0,22743	С	В	В
	5	5	1	0,28546	С	С	С
	6	6	1	0,62461	Н	С	В
	8	7	1	0,00552	Н	Н	Н
$k = 3$	1	8	1	0,28548	С	С	С

Таким чином, для побудови моделі (4.30) в явному вигляді побудована нечітка база знань як система нечітких лінгвістичних висловлювань типу «ЯКЦО-ТО», які зв'язують лінгвістичні оцінки («низький», «середній» та «високий») вхідних змінних з вихідною змінною  $a_1$ . Лінгвістичні оцінки «низький», «середній» та «високий» формалізуємо за допомогою функцій належності. Задамо ці функції у наступному вигляді

$$\mu_T(\gamma) = \frac{1}{1 + \left(\frac{\gamma - b}{c}\right)^2} \quad (4.31)$$

де  $\mu_T(\gamma)$  - число у діапазоні  $[0,1]$ , яке характеризує суб'єктивну міру відповідності значення  $\gamma$  нечіткому терму  $T$  (“низький”, “нижчий за середнє” та ін.),  $b$  та  $c$  параметри, які спочатку вибираються експертом, а потім настроюються на експериментальні дані:  $b$  - координата максимуму функції  $\mu^T(\gamma)$ , причому  $\mu^T(b) = 1$ ,  $c$  - коефіцієнт концентрації – розтягнення функції  $\mu^T(\gamma)$ . Значення параметрів  $b$  і  $c$  для вхідних змінних  $\gamma_1, \gamma_2, \gamma_3$  наведено у табл. 4.4.

*Дефазифікація.* Нечіткій базі знань (табл. 4.3) відповідає наступне представлення об'єкту (4.30) у явному вигляді:

$$a_1 = \frac{\sum_{k=1}^3 d_k \cdot \mu_{D_k}(a_1)}{\sum_{k=1}^3 \mu_{D_k}(a_1)} \quad (4.32)$$

$$\mu_{D_1}(a_1) = \min \left\{ 1, w_3 \mu_{1B}(\gamma_1) \mu_{2B}(\gamma_2) \mu_{3C}(\gamma_3) + w_7 \mu_{1C}(\gamma_1) \mu_{2C}(\gamma_2) \mu_{3C}(\gamma_3) \right\} \quad (4.33)$$

$$\mu_{D_2}(a_1) = \min \left\{ 1, w_2 \mu_{1H}(\gamma_1) \mu_{2H}(\gamma_2) \mu_{3C}(\gamma_3) + w_4 \mu_{1C}(\gamma_1) \mu_{2B}(\gamma_2) \mu_{3B}(\gamma_3) + w_5 \mu_{1C}(\gamma_1) \mu_{2C}(\gamma_2) \mu_{3C}(\gamma_3) + w_6 \mu_{1H}(\gamma_1) \mu_{2C}(\gamma_2) \mu_{3B}(\gamma_3) + w_8 \mu_{1H}(\gamma_1) \mu_{2H}(\gamma_2) \mu_{3H}(\gamma_3) \right\} \quad (4.34)$$

$$\mu_{D_1}(a_1) = \min \left\{ 1, w_1 \mu_{1B}(\gamma_1) \mu_{2B}(\gamma_2) \mu_{3C}(\gamma_3) \right\} \quad (4.35)$$

Тут  $w_p$  – число в діапазоні  $[0,1]$ , яке характеризує суб'єктивну міру впевненості експерта відносно висловлювання з номером  $p$  з бази знань, тобто є вагою  $p$ -го правила.

Однак, як було зазначено вище, побудована нечітка модель (4.32)-(4.34) грубо описує шукану залежність (4.30), тому виконаємо другий етап методу нейролінгвістичній ідентифікації - настройку параметрів нечіткої моделі.

На виході отримуємо оптимальні значення векторів параметрів  $W^*$ ,  $B^*$  і  $C^*$  для задачі мінімізації середньоквадратичного відхилення (критерій точності нечіткої моделі) експериментальних даних від модельних, які представлені в табл. 4.5 для параметрів  $B^*$  і  $C^*$  та у табл. 4.4 для параметра  $W^*$ .

Таблиця 4.4

Параметри функцій належності для термів вхідних змінних до та після  
настройки

	до настройки						після настройки					
	$\gamma_1$		$\gamma_2$		$\gamma_3$		$\gamma_1$		$\gamma_2$		$\gamma_3$	
	$b$	$c$	$b$	$c$	$b$	$c$	$b^*$	$c^*$	$b^*$	$c^*$	$b^*$	$c^*$
Н	1	3,5	1	2,75	1	3,5	2,12	2,66	1,02	0,14	1,02	1,19
С	8	3,5	6,5	2,75	8	3,5	7,72	0,07	3,80	0,38	8,59	0,83
В	15	3,5	12	2,75	15	3,5	14,99	0,11	11,89	0,45	14,9	0,26

Середньоквадратичне відхилення (MSE) складає до настройки 1,9556, після настройки  $9 \cdot 10^{-6}$ .

На рис. 4.1 представлені графіки модельних значень вихідної змінної  $a_1$ , отримані до та після настройки, а також для порівняння - експериментальні дані.

Значення параметра  $a_2$  знаходяться аналогічно методу, описаному для відновлення параметра  $a_1$ .



Рисунок 4.1 – Графіки значень вихідної змінної  $a_1$

Отже, після застосування метода нейро-нечіткої ідентифікації для відновлення значень нечітких параметрів  $a_1$  та  $a_2$  отримуємо такі їхні значення: до настройки  $a_1 = 2,6246$ ;  $a_2 = 1,9613$  та після настройки  $a_1 = 3,0644$ ;  $a_2 = 0,9828$ .

На рис. 4.2 представлені отримані оптимальні динамічні розбиття множини споживачів  $\Omega$  на зони обслуговування кожним підприємством для точок фазової траєкторії при  $T = [1;2;3;4;5;6;7;8]$ . На рис. 4.2 символами « $\circ$ » з відповідним порядковим номером позначені центри розміщення підприємств.

Коментар до роботи алгоритму: для побудови оптимального розбиття області  $\Omega$  була покрита прямокутною сіткою з вузлами  $(i, j)$ ,  $i = 1, \dots, 101$ ,  $j = 1, \dots, 101$ . Для отриманих точок фазової траєкторії було розраховано вартість транспортування від кожного вузла  $(i, j)$  області  $\Omega$  до центрів  $\tau_1$ ,  $\tau_2$ . Виходячи з критерію мінімальної вартості вузол  $(i, j)$  був віднесений до областей  $\Omega_1$  та  $\Omega_2$  відповідно. Точність г-алгоритму як за змінними, так і за субградієнтом була однаковою і складала  $10^{-4}$ . Числові результати роботи алгоритму наведені у табл. 4.5

На наступних восьми рисунках наведено трійки графіків, що відображають результати чисельного експерименту для точних значень  $a_1, a_2$ , їх відновлених значень за наведеним вище алгоритмом та відновлених значень після настройки нечіткої моделі. Порівняння результатів при змінних  $c_1, c_2$  дають змогу не тільки підтвердити результати застосування алгоритму нечіткого підходу, але й побачити вплив на його якісні показники зміни вартості транспортування одиниці продукції з часом. Результати наведені на рис.4.2 - 4.9 побудовані для задачі (4.12), (4.25) при  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$ ,  $d_1 = 1$ ,  $d_2 = 2$ ,  $d_3 = 3$ ,  $A_{11} = A_{22} = 1$ ,  $A_{12} = A_{21} = 0$ ,  $c_1^0 = 1$ ,  $c_2^0 = 2$  та різними значеннями  $c_1^k, c_2^k$ .

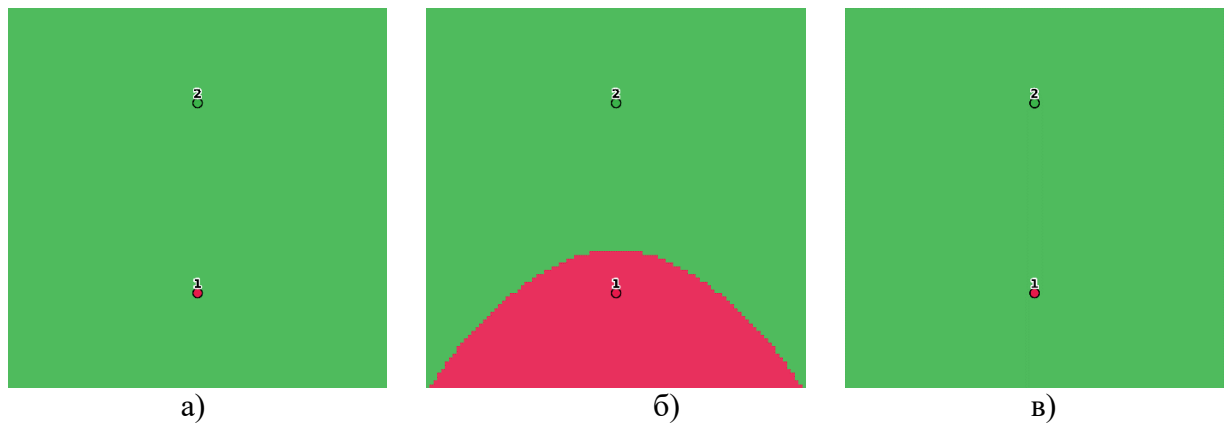


Рисунок 4.2 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;1]$ ,  $c_1 = 4,6967$ ,  $c_2 = 5,0451$  та а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$ .

На рис.4.2-а) наведено результати чисельного експерименту з відшукування чіткого оптимального розбиття множини  $\Omega$  на дві підмножини з заданими координатами центрів  $\tau_1, \tau_2$ . Зважаємо на те що співвідношення  $(c_1 + a_1)/(c_2 + a_2) = 1,2732$ , а центр  $\tau_2$  знаходиться у більш “вигідному” становищі, маючи сумарні витрати (сума статичних та динамічних витрат на транспортування) менші за конкуруючий з ним центр  $\tau_1$ . Таким чином в результаті застосування алгоритму оптимального розбиття всі точки області  $\Omega$  вважаються такими, що підпадають під забезпечення центром  $\tau_2$ . За

результатами підрахунку площі підмножин  $\Omega_1, \Omega_2$  набувають значень  $S(\Omega_1) = 0; S(\Omega_2) = 1$ , при цьому значення цільового функціоналу  $F = 2,1534$ . Застосувавши процедуру визначення значень нечітких параметрів без настройки зміни набули значення  $a_1 = 2,6246; a_2 = 1,9613$  (див.рис4.2-б)). Результати чисельного експерименту, при цьому суттєвим чином відрізняється від чіткого. Співвідношення відповідних параметрів набуває значення  $(c_1 + a_1)/(c_2 + a_2) = 1,0449$ , що залишає перевагу у центру постачання  $\tau_2$ , але близькість значення наведеного співвідношення до 1 зумовлює відмінність розбиття від тривіального. При нечітих значеннях параметрів площі підмножин  $\Omega_1, \Omega_2$  набувають значень  $S(\Omega_1) = 0,2362; S(\Omega_2) = 0,7638$ , а значення цільового функціону при цьому погішується та сягає значення  $F = 3,0425$ . Після настройки нечітких параметрів, виникає ситуація, аналогічна чіткому розбиттю.  $S(\Omega_1) = 0; S(\Omega_2) = 1$ , а значення цільового функціоналу відрізняється від чіткого несуттєво. Таким чином в результаті настройки параметрів отримано розбиття, що із заданою точністю відповідає оптимальному (точному) розбиттю.

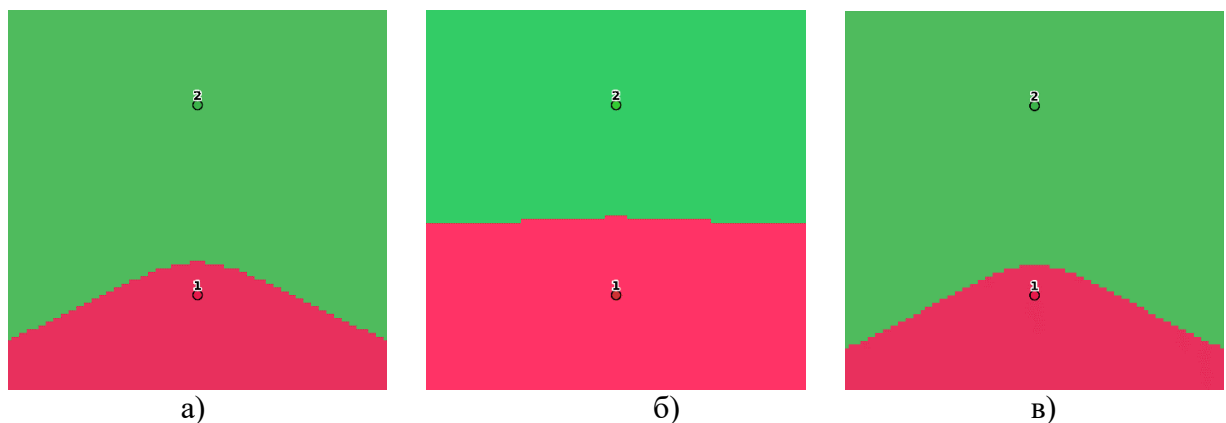


Рисунок 4.3 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;2]$ ,  $c_1 = 11,9758$ ,  $c_2 = 15,9638$  та а)  $a_1 = 3; a_2 = 1$ , б)  $a_1 = 2,6246; a_2 = 1,9613$ , в)  $a_1 = 3,0604; a_2 = 0,9828$

На рис.4.3-а) наведено випадок оптимального розбиття області  $\Omega$  на дві підмножини при змінених динамічних витратах, які для наведеного прикладу приймаються відповідні значення  $c_1 = 11,9758$ ;  $c_2 = 15,9638$ . Зауважимо, що співвідношення  $(c_1 + a_1)/(c_2 + a_2) = 0,8828$ . Значення цільового функціоналу при цьому набуває оптимального значення рівним 6,9089. Оптимальне розбиття з нечіткими параметрами, а саме при  $a_1 = 2,6246$ ;  $a_2 = 1,9613$  представлено на рис.4.3-б). Співвідношення суми статичних та динамічних витрат на транспортування збільшується і хоча центр  $\tau_2$  все ще залишається у більш “вигідному” становищі, на графіку (рис.4.3-б)) бачимо, що межа між підмножинами змістилась ближче до нього тим самим зменшивши кількість споживачів названого центру. Чисельно такий результат можна підтвердити виходячи зі значень площі підмножин  $\Omega_1, \Omega_2$ , що набувають значень  $S(\Omega_1) = 0,4456$ ;  $S(\Omega_2) = 0,5544$  відповідно. При цьому значення цільового функціоналу  $F = 7,892$ . Після настройки оптимальне розбиття з заданою точністю співпадає з точним. Значення цільового функціоналу суттєво покращується  $F = 6,9115$ , а площі підмножин  $\Omega_1, \Omega_2$ , набувають значень  $S(\Omega_1) = 0,2306$ ;  $S(\Omega_2) = 0,7694$  відповідно.

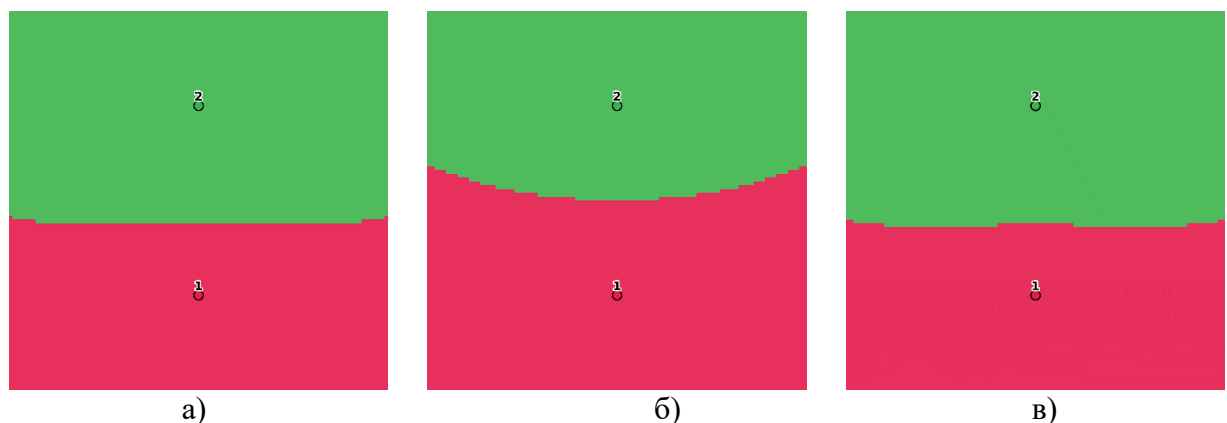


Рисунок 4.4 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;3]$ ,  $c_1 = 29,7873$ ,  $c_2 = 42,6809$  та а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

На рис.4.4.-а) представлено точне оптимальне розбиття області  $\Omega$  на дві підмножини при  $c_1=29,7873$ ,  $c_2=42,6809$  та при значенні ключового

співвідношення  $(c_1 + a_1)/(c_2 + a_2) = 0,7506$ . Виходячи з того, що значення  $(c_1 + a_1)/(c_2 + a_2) < 1$  у більш “вигідному” стані знаходиться центр  $\tau_1$ , а значення цільового функціоналу  $F=15,6989$ . Оптимальне розбиття при нечітких значеннях параметрів  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , вочевидь, погіршує значення цільового функціоналу  $F=16,6386$ , а площі підмножин нечіткого розбиття відповідно дорівнюють  $S_1=0,5298$ ,  $S_2=0,4702$ . Після настройки оптимальне розбиття з відповідною точністю співпадає з точним при цьому значення цільового функціоналу  $F=15,749$ , а площі підмножин  $S(\Omega_1) = 0,4344$ ;  $S(\Omega_2) = 5656$  відповідно.

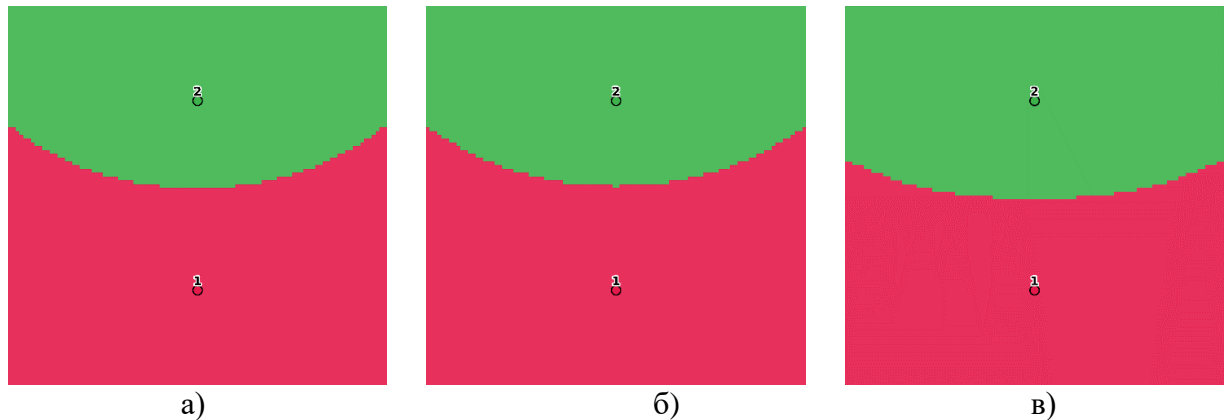


Рисунок 4.5 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;4]$ ,  $c_1 = 69,998$ ,  $c_2 = 102,997$  та а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

На рисунках 4.5-4.9 представлені випадки які топологічно не відрізняються, але при цьому демонструють дуже важливу закономірність нечіткого розбиття та оптимального розбиття після настройки нечіткого параметру. Наведені п'ять рисунків відображають оптимальне розбиття при змінних значення вартості транспортування між двома центрами  $\tau_1, \tau_2$ . Відповідно до (4.25)-(4.26) вартість транспортування одиниці продукції  $c_1, c_2$  змінюються з часом. Проте співвідношення названих параметрів моделі зберігаючи своє співвідношення, зберігає форму та площу відповідних підмножин отриманих в результаті розв'язання задачі оптимального розбиття. Співвідношення динамічних витрат (рис.4.5-рис.4.9) знаходиться в

межах  $(0,66;0,68)$ , співвідношення площ відповідних підмножин  $\Omega_1, \Omega_2$  знаходиться у межах  $(0,1;0,15)$ , а значення цільового функціоналу, при цьому для нечітких значень постійних витрат та для їх значень після налаштування суттєво не відрізняється та приймає значення пропорційні суммі відповідних постійних та змінних витрат на транспортування одиниці продукції.

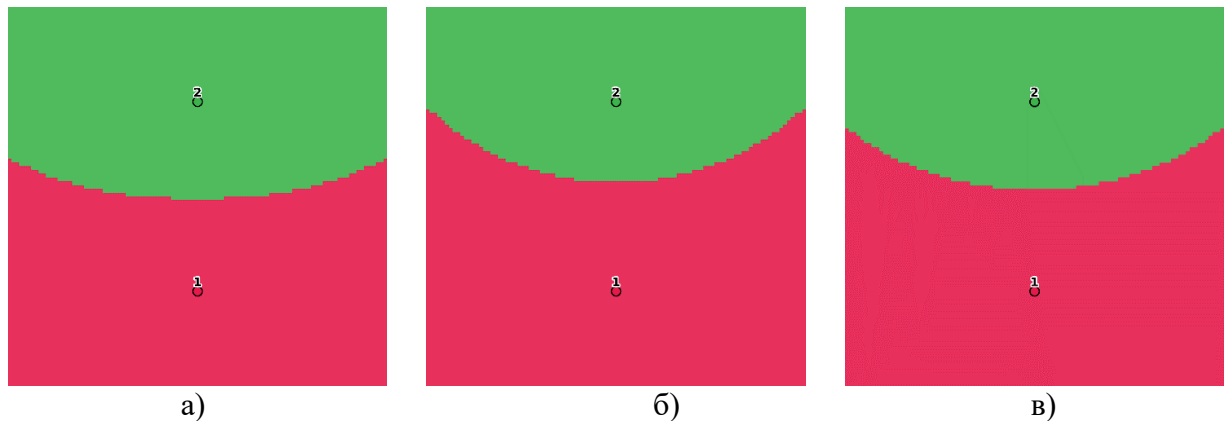


Рисунок 4.6 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;5]$ ,  $c_1 = 154,8688$ ,  $c_2 = 230,3033$  та  
 а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

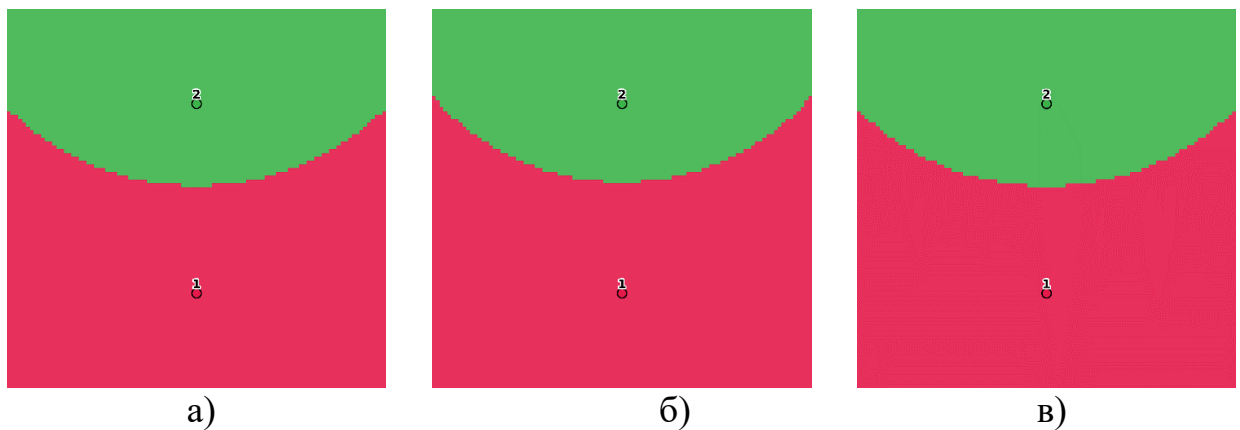


Рисунок 4.7 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;6]$ ,  $c_1 = 324,0672$ ,  $c_2 = 484,1009$  та  
 а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

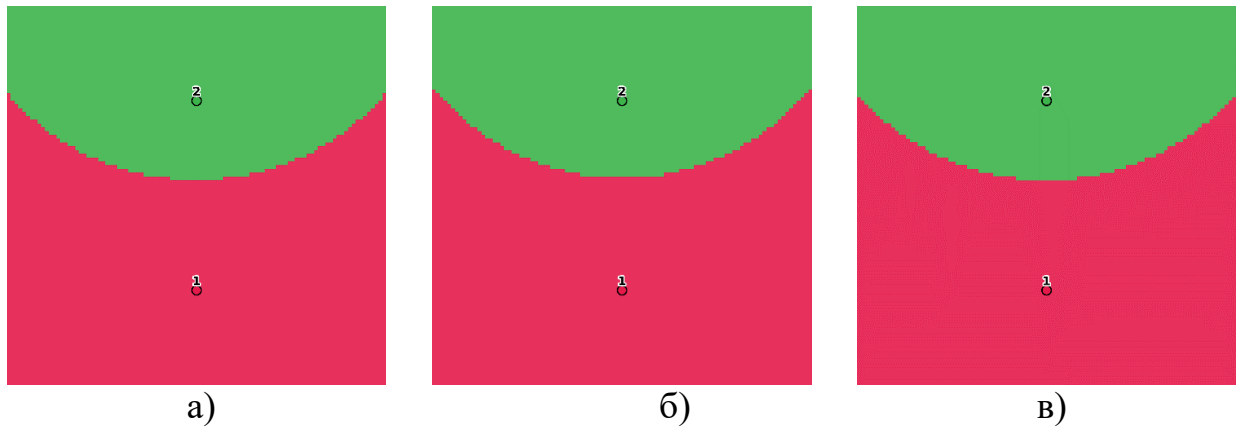


Рисунок 4.8 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;7]$ ,  $c_1 = 645,257$ ,  $c_2 = 965,8855$  та а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

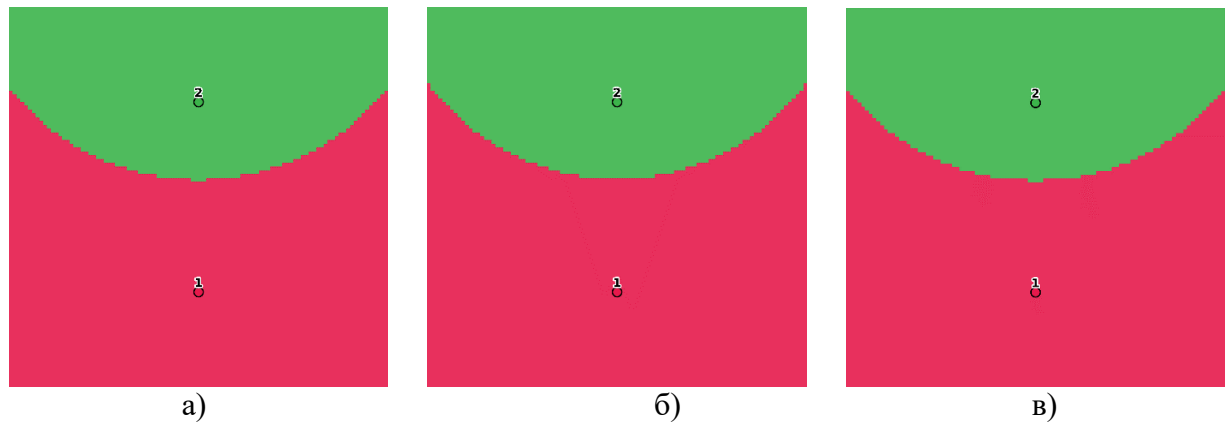


Рисунок 4.9 - Оптимальне розбиття множини споживачів  $\Omega$ , що має форму квадрата зі стороною 1, на дві підмножини з центрами  $\tau_1(0,5;0,25)$ ,  $\tau_2(0,5;0,75)$  (■ -  $\Omega_1$ ; ■ -  $\Omega_2$ ) при  $t \in [0;8]$ ,  $c_1 = 1229,6$ ,  $c_2 = 1842,4$  та а)  $a_1 = 3$ ;  $a_2 = 1$ , б)  $a_1 = 2,6246$ ;  $a_2 = 1,9613$ , в)  $a_1 = 3,0604$ ;  $a_2 = 0,9828$

Числові результати роботи алгоритму для задач рис.4.2 - рис.4.9 наведені у табл. 4.5.

Таблиця 4.5

Числові результати оптимального розбиття множини споживачів  $\Omega$  на дві підмножини ( $F$  – значення цільового функціоналу,  $S_1, S_2$  - обсяги виробництва підприємств (площі підмножин  $\Omega_1, \Omega_2$ ), відповідно)

Часовий проміжок, змінні витрати	Для точних значень $a_1 = 3; a_2 = 1$	Для відновлених значень до настройки нечіткої моделі $a_1 = 2,6246; a_2 = 1.9613$	Для відновлених значень після настройки нечіткої моделі $a_1=3,0604; a_2 = 0,9828$
Початкові значення змінних витрат $c_1 = 2,000; c_2 = 1,000$			
[0;1] $c_1=4,6967$ $c_2=5,0451$	F=2,1534 $S_1=0$ $S_2=1$	F=3,042508 $S_1=0,2362$ $S_2=0,7638$	F=2,1362 $S_1=0$ $S_2=1$
[0;2] $c_1=11,9758$ $c_2=15,9638$	F=6,9089 $S_1=0,2458$ $S_2=0,7542$	F=7,892 $S_1=0,4456$ $S_2=0,5544$	F=6,9115 $S_1=0,2306$ $S_2=0,7694$
[0;3] $c_1=29,7873$ $c_2=42,6809$	F=15,6989 $S_1=0,4416$ $S_2=0,5584$	F=16,6386 $S_1=0,5298$ $S_2=0,4702$	F=15,749 $S_1=0,4344$ $S_2=0,5656$
[0;4] $c_1=69,998$ $c_2=102,997$	F=33,1421 $S_1=0,5254$ $S_2=0,4746$	F=34,0467 $S_1=0,5748$ $S_2=0,4252$	F=33,236 $S_1=0,5222$ $S_2=0,4778$
[0;5] $c_1=154,8688$ $c_2=230,3033$	F=68,7654 $S_1=0,571$ $S_2=0,429$	F=69,6608 $S_1=0,5994$ $S_2=0,4006$	F=68,9008 $S_1=0,5696$ $S_2=0,4304$
[0;6] $c_1=324,0672$ $c_2=484,1009$	F=140,5654 $S_1=0,597$ $S_2=0,403$	F=141,4844 $S_1=0,6124$ $S_2=0,3876$	F=140,7398 $S_1=0,5952$ $S_2=0,4048$
[0;7] $c_1=645,257$ $c_2=965,8855$	F=280,9088 $S_1=0,6102$ $S_2=0,3898$	F=281,8808 $S_1=0,62$ $S_2=0,38$	F=281,1197 $S_1=0,6096$ $S_2=0,3904$
[0;8] $c_1=1229,6$ $c_2=1842,4$	F=545,6868 $S_1=0,6188$ $S_2=0,3812$	F=546,7315 $S_1=0,6242$ $S_2=0,3758$	F=545,9333 $S_1=0,6184$ $S_2=0,3816$

#### 4.5 Біфуркаційні ефекти в динамічних задачах оптимального розбиття множин та підходи до їх дослідження

Як було показано в розділах 2, 3 та 4 математичні постановки чітких та нечітких динамічних задач оптимального розбиття множин з фіксованими

центрами та відшуканням координат містять систему, що визначає правила зміни вартості перевезення одиниці продукції від центрів постачання споживачеві. Неодноразово підкреслювалось, що для шуканого оптимального розбиття не настільки суттєвою є абсолютна вартість транспортування продукції, як їх співвідношення між різними центрами. В розділі 2 наведено кілька прикладів математичних моделей, що можуть бути використані у якості базових для досліджуваної моделі, а саме (2.5) – (2.8). Проте поведінка таких систем є достатньо простою та лінійною при певних співвідношеннях параметрів цих моделей. Задачею біфуркаційного аналізу динамічних підзадач, як і загальної задачі оптимального розбиття є розрахунок таких біфуркаційних значень параметрів моделі при яких якісно змінюється її поведінка. В параграфі 1.4 здійснено огляд підходів до аналітичного дослідження біфуркаційних ефектів в моделях, що можуть бути обрані у базовій якості для динамічної задачі оптимального розбиття множин. Нижче розглянемо математичну модель (2.3) з додатковою умовою (2.4) та моделями (2.6), (2.7) у якості базових.

Очевидно, що однією з точок рівноваги системи

$$\frac{\partial c_i(;;,)}{\partial t} = \sum_{j=1}^N A_{ij} \cdot f_j(c_j(;;,)), i = 1, \dots, N \quad (4.36)$$

де  $f_j(c_j(;;,))$  може мати вигляд (4.37) або (4.38)

$$f_j(c_j(;;,)) = d_j \cdot c_j(;;,) \cdot \left(1 - \frac{c_j(;;,)}{K_1}\right) \quad (4.37)$$

$$f_j(c_j(;;,)) = d_j \cdot (c_j(;;,) - K_2) \cdot \left(1 - \frac{c_j(;;,)}{K_1}\right) \quad (4.38)$$

є початок координат. Також існує нескінченна кількість точок рівноваги системи (4.36)-(4.37), що належать гіперплощині

$$\sum_i c_i(;;,) = K_1. \quad (4.39)$$

Прикладний зміст точок рівноваги є достатньо зрозумілим. Нижче покажемо, що система (4.36), (4.37) є виродженою в околі особливих точок стаціонарної гіперплощини (4.39).

Загальний вигляд  $ij$ -ї компоненти матриці Якобі  $J$  системи (4.36), (4.39) має такий вигляд:

$$J_{ij} = A_{ij}d_j \left( 1 - \frac{\sum_{k=1}^n c_k(;\cdot;\cdot)}{K_1} \right) - \frac{\sum_{p=1}^n A_{ip}d_p c_p(;\cdot;\cdot)}{K_1}. \quad (4.40)$$

Враховуючи (4.39) загальний вигляд  $ij$ -го компонента матриці Якобі  $J$  системи (4.36), (4.37) в точках стаціонарної гіперплощини (4.39) можна переписати у вигляді

$$J_{ij} = -\frac{1}{K_1} \sum_{k=1}^n A_{ik} \cdot d_k \cdot c_k(;\cdot;\cdot). \quad (4.41)$$

Оскільки  $J_{ij}$  в (4.41) не залежить безпосередньо від  $j$ , справедливим є  $J_{pi} = J_{hi}$ ,  $p, h = \overline{1, n}$ , із чого випливає, що вектор-стовпці матриці Якобі є лінійно залежними,  $Det(J) = 0$ , а отже, система є виродженою в особливих точках стаціонарної гіперплощини (4.39).

Система (4.36), (4.37) для  $N = 2$  може зазнавати трьох біфуркацій. Необхідною умовою виникнення біфуркації вважатимемо рівність нулю дійсної частини хоча б одного з власних значень матриці Якобі.

**Теорема 4.1:** система (3.36), (3.37) для  $N = 2$  має вигляд (3.40)

$$\begin{cases} \frac{\partial c_1(;\cdot;\cdot)}{\partial t} = (A_{11}d_1c_1(;\cdot;\cdot) + (1 - A_{22})d_2c_2(;\cdot;\cdot)) \left( 1 - \frac{c_1(;\cdot;\cdot) + c_2(;\cdot;\cdot)}{K_1} \right) \\ \frac{\partial c_2(;\cdot;\cdot)}{\partial t} = ((1 - A_{11})d_1c_1(;\cdot;\cdot) + A_{22}d_2c_2(;\cdot;\cdot)) \left( 1 - \frac{c_1(;\cdot;\cdot) + c_2(;\cdot;\cdot)}{K_1} \right) \end{cases} \quad (3.42)$$

вироджена в околі особливої точки початку координат за чотирьох умов (3.43) – (3.46).

$$d_1 = 0 \quad (3.43)$$

$$d_2 = 0 \quad (3.44)$$

$$A_{11} + A_{22} = 1. \quad (3.45)$$

$$\begin{cases} d_1 d_2 (A_{11} + A_{22} - 1) > 0 \\ A_{11} d_1 + A_{22} d_2 = 0 \end{cases} \quad (3.46)$$

**Доведення:** матриця Якобі системи (3.42) в початку координат має вигляд  $J = \begin{pmatrix} A_{11}d_1 & (1 - A_{22})d_2 \\ (1 - A_{11})d_1 & A_{22}d_2 \end{pmatrix}$ , а її характеристичне рівняння

представлене так:  $t^2 - Tr(J)t + Det(J) = 0$  де  $Tr(J) = A_{11}d_1 + A_{22}d_2$ , а  $Det(J) = d_1 d_2 (A_{11} + A_{22} - 1)$ . Очевидно, що при виконанні будь-якої з умов

$\begin{cases} d_1 = 0 \\ d_2 = 0 \\ A_{11} + A_{22} = 1 \end{cases}$  якобіан  $Det(J) = 0$ , а отже, в околі початку координат система

є виродженою. За умови  $\begin{cases} Det(J) = d_1 d_2 (A_{11} + A_{22} - 1) > 0 \\ Tr(J) = A_{11}d_1 + A_{22}d_2 = 0 \end{cases}$  початок координат

є еліптичною точкою, а дійсна частина пари комплексних коренів характеристичного рівняння, що має вигляд  $Tr(J) = A_{11}d_1 + A_{22}d_2$ , дорівнює нулю. Отже, система є виродженою в околі початку координат за умови

$$\begin{cases} Det(J) = d_1 d_2 (A_{11} + A_{22} - 1) > 0 \\ Tr(J) = A_{11}d_1 + A_{22}d_2 = 0 \end{cases}.$$

Теорему доведено.

**Теорема 4.2:** Система (3.42) в околі початку координат має тип сідло за умови, що  $d_1 d_2 (A_{11} + A_{22} - 1) < 0$ . Якщо

$0 < d_1 d_2 (A_{11} + A_{22} - 1) < \frac{(A_{11}d_1 + A_{22}d_2)^2}{4}$ , початок координат є вузлом, а за

умови  $d_1 d_2 (A_{11} + A_{22} - 1) > \frac{(A_{11}d_1 + A_{22}d_2)^2}{4}$  - фокусом. В останніх двох

випадках точка є стійкою за умови, що  $A_{11}d_1 + A_{22}d_2 < 0$ , та нестійкою при

$A_{11}d_1 + A_{22}d_2 > 0$ .

**Доведення:** матриця Якобі системи (3.42) в початку координат має

вигляд  $J = \begin{pmatrix} A_{11}d_1 & (1 - A_{22})d_2 \\ (1 - A_{11})d_1 & A_{22}d_2 \end{pmatrix}$ , а її характеристичне рівняння

$t^2 - (A_{11}d_1 + A_{22}d_2)t + d_1d_2(A_{11} + A_{22} - 1) = 0$ . Очевидно, що за умови

$Det(J) = d_1d_2(A_{11} + A_{22} - 1) < 0$  дискримінант характеристичного рівняння,

що має вигляд  $D(J) = Tr(J)^2 - 4Det(J)$ , більше нуля, а добуток коренів

характеристичного рівняння (за теоремою Вієта) – менше нуля, тож початок

координат за визначенням має тип сідло. Далі будемо вважати, що

$Det(J) > 0$ . Дискримінант  $D(J)$  характеристичного рівняння має вигляд

$D(J) = (A_{11}d_1 + A_{22}d_2)^2 - 4d_1d_2(A_{11} + A_{22} - 1)$ . Очевидно, що за умови

$d_1d_2(A_{11} + A_{22} - 1) > \frac{(A_{11}d_1 + A_{22}d_2)^2}{4}$  особлива точка (початок координат) є

еліптичною, а дійсна частина пари комплексних коренів має вигляд

$A_{11}d_1 + A_{22}d_2$ , тож при  $A_{11}d_1 + A_{22}d_2 < 0$  особлива точка (початок

координат) має тип стійкий фокус, а за умови  $A_{11}d_1 + A_{22}d_2 > 0$  - нестійкий

фокус. Якщо  $d_1d_2(A_{11} + A_{22} - 1) < \frac{(A_{11}d_1 + A_{22}d_2)^2}{4}$ , особлива точка є

гіперболічною. За припущенням  $Det(J) > 0$  отже, корені характеристичного

рівняння (як наслідок теореми Вієта) мають однаковий знак, а отже, при

$A_{11}d_1 + A_{22}d_2 < 0$  початок координат має тип стійкий вузол, а за умови

$A_{11}d_1 + A_{22}d_2 > 0$  - нестійкий вузол.

Теорему доведено.

Крім розглянутих випадків, може мати місце біфуркація ще одного типу,

коли  $(A_{11} + A_{22} - 1)d_1d_2 > 0$ . У цьому випадку критичне значення параметрів

визначається рівністю

$$A_{11}d_1 + A_{22}d_2 = 0. \quad (3.47)$$

Очевидно, що в околі точки (4.47) дискримінант характеристичного рівняння є від'ємним, тобто особлива точка є фокусом. Біфуркація в цьому випадку є зміною стійкого фокуса на нестійкий чи навпаки.

#### **4.6 Висновки до розділу 4**

У розділі 4 дисертаційної роботи здійснено формалізацію нечіткої динамічної задачі оптимального розбиття множин з фіксованими центрами, що є узагальненням класичних задач кластеризації у напрямі врахування апріорної нечіткості вхідних параметрів. На основі адаптації існуючої методологічної основи побудовано розширену методику ідентифікації нечітких параметрів математичної моделі, яка передбачає можливість використання експериментальних даних та суб'єктивних експертних оцінок. Запропонований підхід дозволяє враховувати як початкову невизначеність геометричних характеристик множини, так і структурну невизначеність у процесі її еволюції в динамічному середовищі.

Розроблено покроковий алгоритм налаштування нечітких параметрів, який включає процедури початкового фузифікування вхідних даних, побудови функцій належності, застосування механізмів адаптивної ідентифікації за допомогою критерію оптимальності, та кінцеву дефузифікацію отриманих результатів. Цей алгоритм дозволяє на практиці реалізовувати нечітке моделювання динамічних процесів з урахуванням наявних апріорних обмежень та цільових критеріїв оптимальності.

На базі створеного алгоритмічного апарату здійснено чисельну реалізацію процесу розв'язання задачі оптимального розбиття множини, заданої у формі квадрату зі стороною одиничної довжини. Особливу увагу приділено аналізу впливу нечіткості на результати розбиття: здійснено порівняльний аналіз трьох сценаріїв — розбиття без урахування нечіткості (чіткого), розбиття з початковими нечіткими параметрами, та розбиття після адаптивного налаштування цих параметрів. Отримані результати засвідчують, що після процедури налаштування нечітких параметрів

досягається висока ступінь відповідності між результатами нечіткого та чіткого підходів як за значеннями цільового функціоналу, так і за геометричними характеристиками підмножин (зокрема, площами отриманих кластерів).

Ключовою особливістю побудованої методології є гнучкість її адаптації до різних прикладних ситуацій. Запропонована модельна вибірка, що використовується для ідентифікації нечітких параметрів, може бути змінена залежно від конкретного типу задачі, структури даних або гіпотез дослідника щодо природи апріорної нечіткості. Це розширює сферу застосування розробленого підходу — зокрема, в задачах, що виникають у прикладній екології, біоінформатиці, логістиці, розподілених обчисленнях тощо.

Окремим напрямом дослідження в межах розділу стало вивчення біфуркаційних ефектів, що виникають у нечіткій динамічній системі. З огляду на наявність широкого спектра якісних режимів поведінки динамічної моделі, яка використовується як допоміжна до базової задачі, проаналізовано умови виникнення біфуркацій — критичних змін у поведінці системи за варіації вхідних параметрів. Проведено глибокий якісний аналіз таких режимів та формалізовано аналітичні умови їх появи. У вигляді строго математичних тверджень сформульовано та доведено відповідні теореми про умови виникнення біфуркацій, що дозволяє надалі використовувати ці результати для попереднього оцінювання стабільності розбиття та побудови сценарного аналізу.

Таким чином, у межах розділу 4 запропоновано комплексну методологію математичного моделювання, аналізу та розв'язання нечіткої динамічної задачі оптимального розбиття множин з фіксованими центрами, що охоплює як ідентифікацію параметрів, так і їхню адаптацію та аналітичне дослідження динаміки модельної системи.

## Розділ 5. РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ВОГНЕВИХ ГРУП З ВИКОРИСТАННЯМ МЕТОДІВ ТЕОРІЇ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН

### 5.1 Огляд прикладної проблеми

З початком повномасштабного вторгнення держави-агресорки на територію України перед Збройними силами України постала нагальна потреба в ефективному виявленні та своєчасному знищенні ворожих повітряних об'єктів, таких як ракети, безпілотники, літаки та гелікоптери. Існуючі на той момент технічні засоби виявлення — зокрема, наземні радіолокаційні станції П-18 «Терек», П-19 «Дуга», 36Д6М1-1, 80К6КС1, мобільні РЛС TRML-4D та IRIS-T SLM, а також системи теплового й візуального спостереження, акустичні комплекси, пасивні сенсори і засоби радіоелектронної розвідки (РЕР) — виявилися доволі ефективними в умовах бойових дій. Проте, незважаючи на їхню функціональність і бойову стійкість, кількість таких засобів виявилася недостатньою для забезпечення безперервного контролю за повітряним простором у межах усієї території України.

Ключовим чинником успішного знищення ворожих цілей є їх своєчасне виявлення. Але ця задача суттєво ускладнюється низкою чинників: великою площею держави, недостатньою кількістю технічних засобів спостереження, а також специфікою застосування ворогом озброєння. Зокрема, держава-агресор активно використовує ракети та безпілотники, що рухаються на наднизьких висотах і з високою швидкістю, що ускладнює їх виявлення класичними засобами ППО.

Водночас очевидним є значний ресурс, який Україна має у вигляді активного громадянського суспільства. Зокрема, цивільні особи нерідко здатні зафіксувати повітряні цілі візуально або за звуком, що стає особливо ефективним з огляду на низьковисотну траєкторію польоту таких об'єктів. Це створює потенціал для залучення цивільного населення до процесу

оперативного інформування сил протиповітряної оборони про виявлені загрози, а також для подальшого аналітичного опрацювання отриманих даних задля вдосконалення рішень у сфері ППО.

У перші тижні повномасштабного вторгнення вже фіксувалися окремі випадки, коли військовослужбовці, виявивши повітряну ціль, швидко передавали інформацію підрозділам ППО, що дозволяло успішно знищити загрозу. Проте такі епізоди залишалися радше поодинокими прикладами, ніж системним підходом. Адже для того, щоб знищення ворожих цілей відбувалося оперативно та ефективно, необхідне одночасне виконання ряду умов:

1. Своєчасне візуальне або звукове виявлення повітряного об'єкта;
2. Наявність надійного та оперативного каналу комунікації з підрозділами ППО;
3. Передача максимально точної інформації щодо місця спостереження об'єкта, а також його траєкторії або, принаймні, напрямку руху.

Ці складові є критичними, оскільки в умовах надвисокої швидкості переміщення ворожих засобів ураження навіть незначна затримка у передачі даних може звести нанівець можливість їх своєчасного перехоплення. Саме тому постала нагальна потреба у створенні програмного комплексу, здатного автоматизувати процес збору, обробки та передачі інформації від спостерігачів до операторів ППО, а також накопичувати статистичні дані для аналітичного використання та стратегічного планування.

У відповідь на це завдання було розроблено спеціалізований програмний продукт, який, окрім базової функції сповіщення операторів протиповітряної оборони про зафіксовані повітряні загрози, також містив потужний аналітичний модуль. У межах цього комплексу були реалізовані математичні моделі прогнозування, а також алгоритми динамічного поділу території Дніпропетровської області для оптимального розміщення стаціонарних засобів ППО. Ці моделі враховували обмежену кількість систем

та складність їх передислокації, дозволяючи раціонально планувати їхнє використання.

Розроблений програмний комплекс відповідав усім вимогам інформаційної безпеки, включно із захистом особистих даних цивільних повідомлювачів, і отримав назву «Система альтернативного оповіщення «Світло» (САО «Світло»).

Особливо варто відзначити, що система «Світло» активно застосовувалася в період із березня по грудень 2022 року та не мала жодних повноцінних аналогів у той час. Хоча станом на сьогодні як в Україні, так і за її межами з'явилися різні системи, які частково виконують аналогічні функції — такі як «WireXAlertSystem», «AlertHubPro», «SafeAlertPlus», «EverSafeNotifications», «OnSolve MIR3», система ППО компанії «BI Group», а також «Єдина система протиповітряної оборони (ЄППО)», проте САО «Світло» і досі залишається в експлуатації окремими регіональними підрозділами ППО завдяки своїм унікальним перевагам.

Частина перелічених систем, розроблених фахівцями іноземних держав, була створена ще до початку повномасштабної агресії, що дозволило команді розробників САО «Світло» врахувати як їхні сильні сторони, так і обмеження.

Наприклад, «WireXAlertSystem» — це високофункціональна система оповіщення, що відзначається винятковою гнучкістю. Вона дозволяє користувачам самостійно обирати канали комунікації — від SMS, електронної пошти, повідомлень у месенджерах до інтеграції з іншими цифровими платформами. Така модульність дає змогу адаптувати систему під різні сценарії: як для масового інформування населення, так і для цільових повідомлень окремим групам. Станом на нині ця система успішно застосовується в багатьох країнах світу, зокрема у США, де її використовують для швидкого інформування населення про стихійні лиха — зокрема урагани, землетруси, лісові пожежі.

Однією з ключових переваг «WireXAlertSystem» є можливість призначення пріоритетності повідомлень: критичні сповіщення можуть одночасно надсилатися через усі доступні канали, включаючи голосові дзвінки. Технічно система базується на мікросервісній архітектурі, що дозволяє ефективно обробляти значні обсяги паралельних запитів. У її основі лежать високопродуктивні сервери, які забезпечують надійну доставку повідомлень із мінімальною затримкою. Завдяки своїй архітектурі, система легко масштабується — як для невеликих організацій, так і для національних служб оповіщення. Особлива увага приділяється безпеці: усі повідомлення шифруються з використанням протоколу SSL, що мінімізує ризики витоку або перехоплення даних.

Ця система є найбільш ефективною у випадках, коли виникає потреба швидко повідомити великі маси людей про загрозу, наприклад, техногенну аварію чи природну катастрофу — особливо в густонаселених урбанізованих регіонах.

Іншим прикладом є «AlertHubPro», що виконує не лише функцію розсилки повідомлень, а й забезпечує об'єднання сигналів із різних джерел у єдину інтегровану систему. Це дозволяє користувачам централізовано керувати всіма оповіщеннями, моніторити їхній статус у реальному часі та переглядати детальну хронологію подій. Такий підхід є особливо цінним для великих організацій та державних структур, де критично важливо швидко та чітко обробляти дані з різних джерел.

Система також надає можливість створювати власні канали сповіщення, а також використовувати готові шаблони для типових надзвичайних ситуацій. Вона дозволяє налагоджувати взаємодію між користувачами завдяки функціям коментування та обговорення повідомлень, що покращує координацію дій у кризових умовах. Крім того, «AlertHubPro» підтримує інтеграцію з іншими інформаційними системами — наприклад, з екстреними службами або органами місцевої влади. Це дає змогу оперативно

запускати сповіщення в разі виникнення аварійних ситуацій, таких як пожежі, витoki хімічних речовин тощо.

Однак важливим недоліком «AlertHubPro» є її залежність від безперервного підключення до мережі Інтернет. У разі його втрати можливості системи істотно звужуються, що може мати критичні наслідки в екстрених обставинах.

Система «SafeAlertPlus» спеціалізується на забезпеченні оповіщення в корпоративному середовищі та орієнтована на потреби великих компаній і установ. Вона дозволяє гнучко формувати групи користувачів із різними рівнями доступу до інформації, що забезпечує можливість налаштовувати повідомлення відповідно до ієрархії персоналу або функцій конкретного відділу.

Однією з ключових особливостей SafeAlertPlus є її тісна інтеграція з внутрішніми системами корпоративної безпеки. Це дає змогу не лише інформувати співробітників про інциденти, а й здійснювати моніторинг ситуацій на об'єктах у режимі реального часу. Наприклад, менеджери можуть отримувати розгорнуті звіти про події, тоді як інші працівники — лише короткі попередження щодо потенційної загрози.

Крім того, SafeAlertPlus підтримує взаємодію з пристроями Інтернету речей (IoT), що дозволяє автоматизувати процес інформування у випадку збоїв у роботі обладнання. Завдяки цим можливостям система є надзвичайно ефективною для використання на великих промислових об'єктах.

Проте слід враховувати, що впровадження SafeAlertPlus потребує значних фінансових і технічних ресурсів. Це може бути непосильним для малих підприємств або організацій із обмеженим бюджетом.

«EverSafe Notifications» — це рішення, розроблене насамперед для громадських установ, зокрема медичних закладів та освітніх установ. Головною перевагою цієї системи є її здатність використовувати широкий набір каналів для передачі сповіщень — від голосових дзвінків і Push-повідомлень до електронної пошти.

Особливістю EverSafe є її орієнтація на потреби людей із порушеннями зору або слуху. Так, система здатна надсилати аудіоповідомлення для людей із вадами зору або ж текстові сповіщення з візуальними елементами — для користувачів із вадами слуху. Завдяки цьому вона є особливо цінною для закладів охорони здоров'я, де необхідна оперативна реакція на надзвичайні ситуації — спалахи захворювань, технічні несправності, евакуації тощо.

Втім, ця система має і свої недоліки. Зокрема, вона обмежена у можливості глибокої персоналізації повідомлень для окремих груп користувачів. Така особливість може бути незручною у складних організаційних структурах, де кожна категорія персоналу повинна отримувати індивідуалізовану інформацію.

«OnSolve MIR3» є передовим рішенням для кризового менеджменту, що активно застосовує штучний інтелект і технології машинного навчання для автоматизації реагування на надзвичайні події. Система інтегрується з камерами відеоспостереження, системами контролю доступу та іншими засобами безпеки, дозволяючи створювати складні сценарії реагування на різні типи загроз.

Завдяки використанню інтелектуальних алгоритмів, MIR3 здатна в реальному часі аналізувати ситуації та генерувати відповідні повідомлення, залежно від ступеня ризику. Це робить її надзвичайно ефективною для підприємств, які мають високі вимоги до комплексної системи безпеки. MIR3 здатна оперативно реагувати на терористичні загрози, природні катастрофи, аварії та інші кризові події.

Система має високу адаптивність і може бути налаштована під специфіку конкретної організації. Проте її впровадження вимагає серйозної підготовки і залучення технічних фахівців, що ускладнює її використання для малих компаній або установ із обмеженими ресурсами.

Система ППО компанії «VI Group» була розроблена виключно для внутрішнього використання військовими підрозділами, з акцентом на

захищену передачу інформації. Вона не призначалася для використання цивільними користувачами, оскільки її головна функція — забезпечення надійної комунікації між підрозділами протиповітряної оборони. Такий підхід дає змогу ефективно передавати критичну інформацію, мінімізуючи ризик перехоплення або втрати даних.

Система єдиної протиповітряної оборони (ЄППО) сьогодні є наймасовішим інструментом оповіщення в Україні. За останніми даними, кількість її користувачів перевищила 400 тисяч. Система була вперше запущена у тестовому режимі в листопаді 2022 року, а повноцінне впровадження відбулося вже в грудні.

Варто підкреслити, що САО «Світло» було створено ще раніше — у березні 2022 року. Вона пройшла повний цикл розробки, тестування, вдосконалення та інтеграції вже на початку повномасштабного вторгнення. За інформацією Повітряного командування «Схід», хоч ця система і вважалася альтернативною, її ефективність була доведена на практиці. Зокрема, завдяки САО «Світло» вдалося виявити понад два десятки ворожих повітряних цілей, серед яких — навіть балістична ракета типу «Іскандер-М».

## **5.2 Предметна постановка задачі**

Зоною відповідальності ПвК «Схід» є східні та частина центральних регіонів України, де спостерігається найбільша активність держави-агресорки у повітрі. Зокрема, це Харківська, Дніпропетровська, Донецька (включно з ТОТ), Луганська (включно з ТОТ), а також частина Запорізької, Кіровоградської, Полтавської та Сумської областей України. Програмний комплекс мав включати три основні компоненти, а саме: клієнтську (мобільний додаток для цивільного населення, представників ТРО та громадських об'єднань), панель адміністратора (для представників ППО) та серверну частину орієнтовану на організацію, зберігання, передачу та відображення інформації.

Клієнтський модуль потрібно реалізувати у вигляді програмного додатку, що маючи максимально простий інтерфейс має дозволити з мінімальними витратами часу передати інформацію про координати виявлення та напрям руху ворожої повітряної цілі.

Панель адміністратора використовувалась у браузерному форматі з додержанням усіх вимог безпеки представниками ППО. На панелі адміністратора відображалась карта України з нанесенням на неї позначок про виявлені повітряні цілі, їх координати та напрям руху. Налаштування панелі адміністратора мають дозволити коригувати принцип відображення позначок про виявлені повітряні цілі, час, що пройшов з моменту її виявлення, відновлювати та видаляти їх. Крім цього однією з функціональних особливостей панелі адміністратора було відслідковування місця розташування кореспондентів у яких встановлений клієнтський модуль навіть за умови якщо користувач не передає серверній частині повідомлення про наявні ворожі повітряні цілі. Наявність такого функціонального елемента зумовлений необхідністю розуміння на яких рубежах зони відповідальності ПвК “Схід” розміщені користувачі, що відслідковують ворожі повітряні цілі.

Серверна частина це комплекс програмних та апаратних засобів, що використовувались для організації, зберігання, передачі та відображення інформації.

### **5.3 Математична постановка задачі**

Нехай  $\Omega$  – обмежена множина, що включає територію Харківської, Дніпропетровської, Донецької (включно з ТОТ), Луганської (включно з ТОТ), а також частину Запорізької, Кіровоградської, Полтавської та Сумської областей України (Рис.5.1).



Рисунок 5.1. Територія України, що є зоною відповідальності ПвК “Схід”

Сукупність підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega \subset E_n$  назвемо можливим розбиттям цієї множини, за умови  $\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j)$ , де  $N > 0$  – задане натуральне число. Далі будемо вважати  $N = 100$ .

Позначимо через  $\sum_{\Omega}^N$  клас всіх можливих розбиттів множини  $\Omega \subset E_n$  на задану кількість  $N$  її вимірних за Лебегом підмножин:

$$\sum_{\Omega}^N = \{\bar{\omega} \equiv \{\Omega_1, \dots, \Omega_N\} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j\}$$

Далі, через  $\tau_1, \dots, \tau_N$  позначимо сукупність модельних центрів розміщення засобів ППО для підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, які назвемо центрами цих підмножин:  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , і будемо вважати, що координати усіх центрів задані. Для визначеності впишемо виділену на Рис.5.1 зону відповідальності ПвК “Схід” в квадрат зі стороною 1 та розмістимо центри  $\tau_1, \dots, \tau_N$  рівномірно на цьому квадраті. Таким чином

$$\begin{aligned} &\tau_1(0,05;0,05), \tau_2(0,05;0,15), \dots, \tau_{10}(0,05;0,95), \\ &\tau_{11}(0,15;0,05), \tau_{12}(0,15;0,15), \dots, \tau_{20}(0,15;0,95), \\ &\dots \\ &\tau_{91}(0,95;0,05), \tau_{92}(0,95;0,15), \dots, \tau_{100}(0,95;0,95) \end{aligned} \quad (5.1)$$

На Рис.5.2 наведено область  $\Omega$  вписану в квадрат зі стороною 1 та сіткою з кроком 0,1 в комірках якої розташовані центри (5.1).

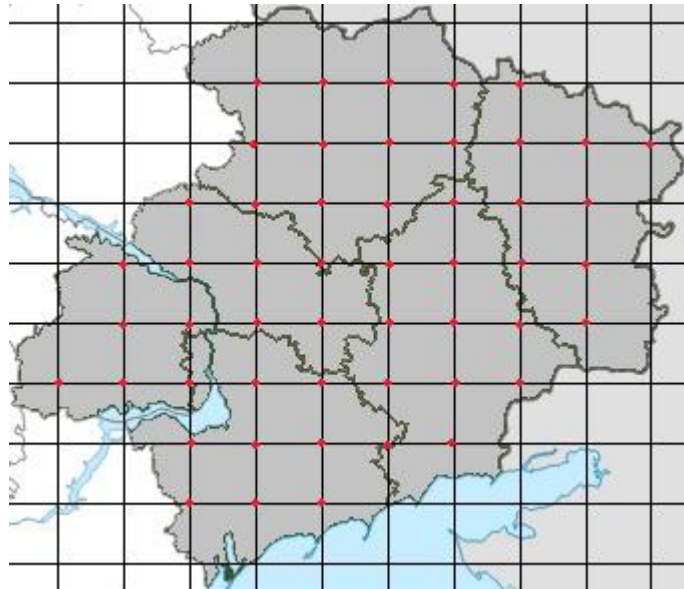


Рисунок 5.2. Область  $\Omega$  вписана в квадрат зі стороною 1 та сіткою з кроком 0,1 в комірках якої розташовані центри (5.1).

Як видно з Рис.5.2 центри з порядковими номерами 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 19, 20, 21, 22, 23, 31, 32, 40, 41, 50, 51, 60, 69, 70, 71, 72, 78, 79, 80, 81, 82, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100 знаходяться поза межами області, що досліджується. Таким чином 49-ть центрів будуть виключені з розгляду задачі у зв'язку з неможливістю розміщення в них центрів  $\tau_i$ .

**Постановка задачі.** Знайти розбиття  $\bar{\omega} = \{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$  множини  $\Omega \subset E_n$  і векторну функцію  $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$ , визначену м.в. для  $x \in \Omega$  при заданому фіксованому наборі центрів  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$  та всіх  $t \in [0, T]$ , які забезпечують 
$$\bar{\omega} \in \sum_{\Omega}^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T]) \quad \inf_{\bar{\omega} \in \sum_{\Omega}^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, c(\cdot, \cdot, \cdot)) \quad \text{де}$$

$$F(\bar{\omega}, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad \text{за наявності умов}$$

$$\frac{dc_i(x, \tau_i, t)}{dt} = \sum_{j=1}^N A_{ij} \cdot d_j \cdot c_j(x, \tau_j, t), 0 \leq t \leq T, \quad c_i(x, \tau_i, t_0) = c_{0i}(x, \tau_i), i = 1, \dots, N \quad \text{м.в.}$$

для  $x \in \Omega$ , при фіксованих  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega, i = 1, \dots, N$ , та умови замкненості

системи  $\sum_{i=1}^N A_{ij} = 1, j = 1, \dots, N$ . Тут  $c_i(x, \tau_i, t), i = 1, \dots, N$ , – шукані дійсні функції,

визначені на  $\Omega \times \Omega \times [0, T]$ , які при будь-якому фіксованому  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i$  неперервно диференційовані за аргументом  $t$  на відрізку  $[0, T]$  м.в. для  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ , обмежені та вимірні за аргументом  $x$  на  $\Omega$  для всіх  $t \in [0, T]$ ;  $m(x, \tau_i)$ ,  $c_{0i}(x, \tau_i)$  – задані дійсні функції, визначені на  $\Omega \times \Omega$ , обмежені та вимірні за аргументом  $x \in \Omega$  при будь-якому фіксованому  $\tau_i \in \Omega_i$  для всіх  $i = 1, \dots, N$ ;  $\rho(x)$  – задана невід'ємна функція, обмежена та вимірна на  $\Omega$ ;  $a_i, i = 1, \dots, N$ , – задані, як правило, невід'ємні числа;  $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$ , – задані числові параметри;  $T > 0$  та  $t_0 \in [0, T]$  задані. Далі для простоти будемо вважати  $\rho(x) = 1$ ,  $d_i = 1$ ,  $a_i = 0$ .

#### 5.4 Результати чисельного експерименту

Система автоматизованого оповіщення “Світло” була створена спеціально для використання у воєнних умовах. Її головне призначення — швидке інформування про ракетні атаки та інші загрози, пов’язані з військовою агресією з боку держави-агресора. Основна перевага цієї системи полягає в її високій швидкодії, що дає змогу своєчасно отримувати дані про потенційну небезпеку та оперативно реагувати на неї.

Порівняно з міжнародними рішеннями, такими як OnSolve MIR3 чи AlertHub Pro, САО “Світло” має більш чітке спрямування — вона спеціалізується саме на воєнних ризиках. Така особливість робить її особливо ефективною саме в українських реаліях, де своєчасна інформація про загрози має вирішальне значення для безпеки країни.

Важливою перевагою “Світла” є її взаємодія з системами протиповітряної оборони. Це забезпечує високу точність та оперативність оповіщень, що значно підвищує рівень захищеності населення, даючи людям більше часу на підготовку до можливої небезпеки.

Інтерфейс системи передбачає перегляд інтерактивної карти, що охоплює як усю країну, так і конкретні регіони. У випадку відсутності

активних сигналів на карті нічого не відображається — вона виглядає порожньою, як показано на Рис.5.3.

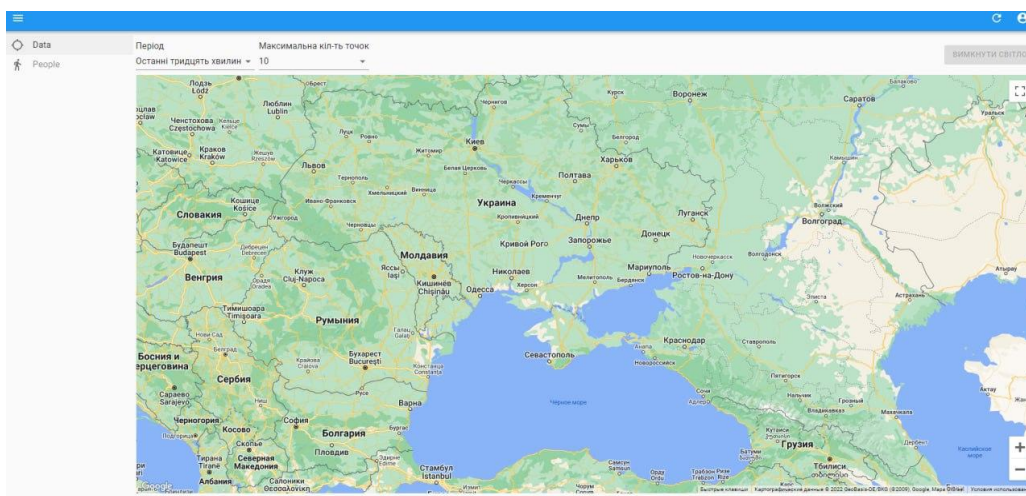
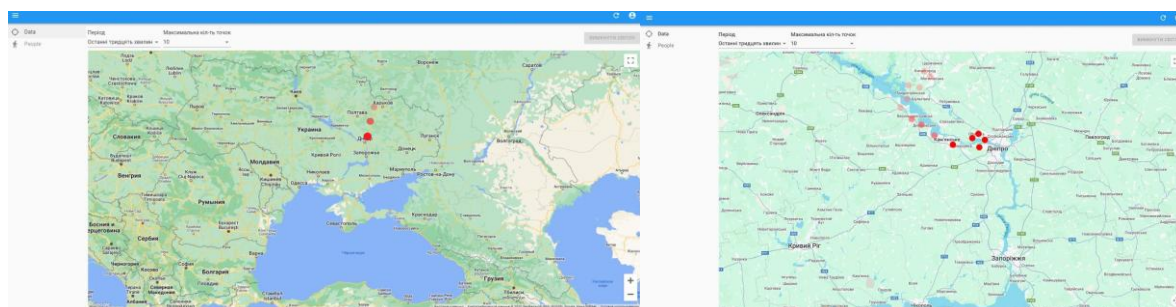


Рисунок 5.3 – Загальний вигляд інтерфейсу панелі адміністратора

Зауважимо, що при надходженні повідомлення про виявлення ворожої повітряної цілі на інтерфейсі панелі адміністратора з'являтимуться маркери, інтенсивність кольору яких сигналізує про час, що минув з моменту надходження повідомлення. З часом маркери стають більш блідою кольору, а час їх відображення може бути налаштований в головному меню панелі адміністратора. Маркери, що відобразились останніми відобразатимуться найбільш яскравим кольором, а більш старі відповідно блідим. На Рис.5.4 зображено, як «ворожа повітряна ціль» рухається по нелінійній траєкторії а) зі сторони Харкова на Дніпро та б) в межах міста Дніпро .



а)

б)

Рисунок 5.4 – Маштабований вигляд інтерфейсу панелі адміністратора з отриманими вхідними даними диференційованими за часом отримання

Оскільки карта САО “Світло” може бути масштабована, існує можливість деталізованого опрацювання території усєї країни. На Рис.5.5 наведено модельне зображення нелінійної траєкторії руху ворожої повітряної цілі у межах міста Кам’янське. Перші шість точок, наведені на рисунку градієнтом червоного кольору є фактично зафіксованими локаціями в яких було зафіксовано ворожу повітряну ціль. Наступні дві точки є екстраполяцією її подальшого руху, з урахування її нелінійної траєкторії.

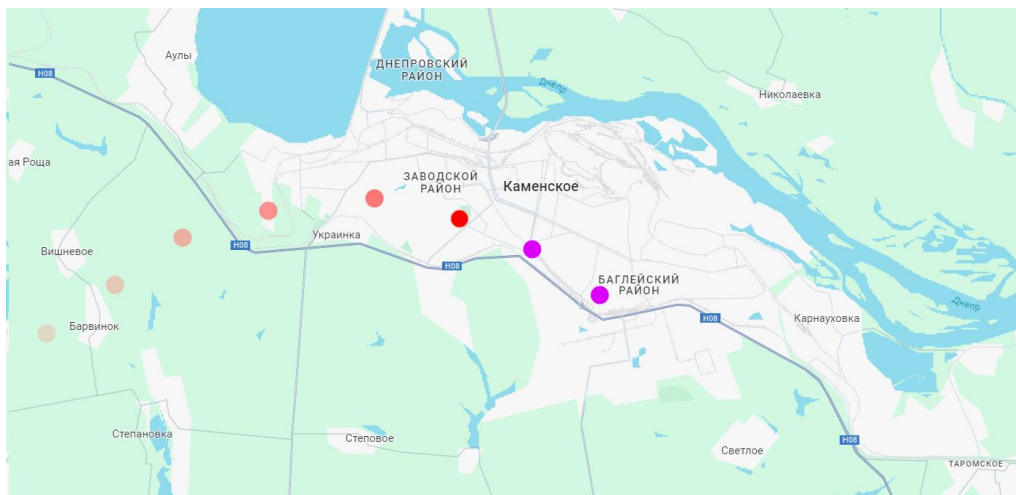


Рисунок 5.5 – Масштабований інтерфейс панелі адміністратора з екстрапольованою траєкторією руху ворожої повітряної цілі.

Задля розуміння подальшого руху ракети було додано алгоритм екстраполяції, що був реалізований з використанням алгоритму екстраполяції через метод найменшого квадратичного відхилення. Ця функція панелі адміністратора може бути активована за допомогою кнопки “Припущення” що доступна у вкладці Data панелі адміністратора САО “Світло”. Враховуючи постійну зміну траєкторії руху ворожих повітряних цілей екстраполяція виявила свою ефективність лише у 15% випадків.

Візуальне представлення на панелі адміністратора АСО “Світло” масованих дронних атак, коли над обмеженою територією фіксується проліт значної кількості ворожих повітряних цілей, представлено на рис.5.6.

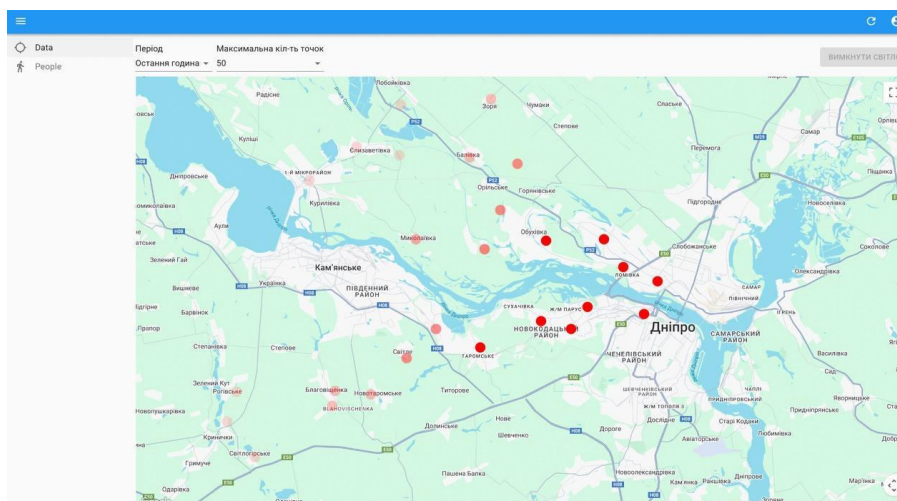


Рисунок 5.6 – Вигляд панелі адміністратора АСО “Світло” при масованих дронних атаках в межах окремого населеного пункту

У цьому випадку деталізація часо польоту дає розуміння не тільки напрямку руху ворожої повітряної цілі, але й їх кількості та дозволяє розрізнити їх між собою.

В таблиці 5.1 наведено результати чисельного експерименту для різної кількості вогневих груп, що були застосовані для Дніпропетровського регіону. В дослідженні було враховано, зокрема, загальну площу, яку означена кількість вогневих груп покриє, частку цієї площі від площі Дніпропетровської області, середній час відгуку та відповідне значення цільового функціоналу.

Таблиця 5.1

Результати чисельного експерименту при різній кількості вогневих груп у Дніпропетровській області

Кількість вогневих груп	Загальна площа покриття (км <sup>2</sup> )	Частина площі від загальної площі Дніпропетровського регіону (%)	Середній час відгуку (хв..)	Значення цільового функціоналу
10	12,850	40,2	12,4	0,71
20	21,500	67,2	9,3	0,82
30	27,850	87,0	7,1	0,89

Як можна бачити з таблиці 5.1 – збільшення кількості вогневих груп на території фіксованої площі пне призводить до пропорційного покращення результату, проте суттєво зменшує середній час відгуку тим самим якісно збільшуючи ефективність застосування вогневих груп.

Виходячи з наведеної в даному розділі математичної постановки динамічної задачі оптимального розбиття множин та даних про ворожі повітряні цілі, отримані з використання АСО “Світло”, під час девяти масованих атак, здійснених росією протягом 2022-2024 років, було обчислено оптимальне розташування вогневих груп на території Дніпропетровської області. Результати чисельного розв’язання задачі (2.1)-(2.4) на території Дніпропетровської області за даними про ворожі повітряні цілі наведені нижче у девяти прикладах. У наведених девяти прикладах використовуватимуться такі позначення:

Дата та час постанови чисельного експерименту наведені у форматі рік/місяць/день-година/хвилина. Характеристика ворожих повітряних цілей містить інформацію про напрям руху цілі (N – з півночі, S – з півдня, W – із заходу, E – зі сходу) та координати перетину ворожою повітряною ціллю межі Дніпропетровської області. Для обчислення координат перетину межі Дніпропетровської області ворожими повітряними цілями картографічне зображення Дніпропетровської області було вписано в прямокутник зі сторонами 500\*300 од.

1. Дата-час: 22/10/04-12:04. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): S:53:112 N:63:209 E:32:-13 E:113:-85 S:24:141 N:89:304 N:110:224 E:10:-18 S:115:163 W:84:219 E:190:44 E:163:-76 N:13:207 N:36:230 E:111:24. Координати оптимального розміщення вогневих груп: (170;115), (196;187), (120;67), (113;25), (293;27), (90;46).

2. Дата-час: 22/11/18-08:01. Кількість вогневих груп: 5. Характеристика ворожих повітряних цілей (13 шт.): N:146:353 N:135:193 E:144:36 S:141:75 W:184:136 S:93:28 W:85:268 E:65:-63 W:79:150 N:110:213 W:0:185 N:45:352

S:108:41. Координати оптимального розміщення вогневих груп: (122;158), (97;21), (195;162), (180;191), (272;52).

3. Дата-час: 22/12/16-14:12. Кількість вогневих груп: 8. Характеристика ворожих повітряних цілей (17 шт.): W:9:122 W:11:217 N:124:305 N:95:335 E:58:-79 E:10:-75 N:145:223 N:138:259 S:100:118 E:133:47 W:198:138 N:138:327 W:159:245 S:28:34 S:9:14 S:10:121 W:175:260. Координати оптимального розміщення вогневих груп: (196;27), (40;175), (42;44), (108;77), (174;121), (174;106), (217;173), (89;111).

4. Дата-час: 23/05/12-03:12. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): W:102:233 S:53:131 W:167:96 E:174:10 S:30:36 W:161:182 N:143:241 S:38:114 E:51:-8 E:196:85 N:3:254 W:18:26 W:153:111 S:119:134 N:133:352. Координати оптимального розміщення вогневих груп: (103;2), (99;189), (242;29), (37;197), (43;130), (189;34).

5. Дата-час: 23/06/21-23:05. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): W:92:129 W:6:168 S:115:10 N:128:184 N:71:223 N:50:214 S:54:151 S:26:74 S:50:166 N:132:181 W:82:142 S:20:72 W:17:142 E:169:52 S:11:99. Координати оптимального розміщення вогневих груп: (185;47), (187;105), (90;35), (277;90), (62;165), (268;54).

6. Дата-час: 23/08/03-16:13. Кількість вогневих груп: 4. Характеристика ворожих повітряних цілей (15 шт.): S:58:34 E:3:32 W:179:257 S:47:128 S:80:110 N:117:315 E:36:17 S:92:70 W:129:198 E:65:69 E:43:13 E:142:-82 N:17:331 S:72:100 N:107:256. Координати оптимального розміщення вогневих груп: (149;152), (204;147), (145;199), (163;113).

7. Дата-час: 24/01/08-18:00. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): S:110:81 E:94:-20 E:59:-67 E:139:77 S:43:153 S:42:164 N:39:332 E:199:72 E:167:-14 E:147:29 W:160:229 N:50:356 S:48:142 E:47:51 N:13:221. Координати оптимального розміщення вогневих груп: (9;153), (231;14), (82;168), (160;198), (125;155), (171;54).

8. Дата-час: 24/03/14-16:42. Кількість вогневих груп: 4. Характеристика ворожих повітряних цілей (15 шт.): S:127:142 S:46:51 W:180:184 W:88:197

W:1:100 N:18:183 W:171:176 W:46:132 S:77:40 E:138:82 N:78:238 S:124:33  
S:29:172 W:14:195 N:86:244. Координати оптимального розміщення вогневих груп: (165;122), (222;160), (146;114), (118;11).

9. Дата-час: 24/08/20-10:19. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (14 шт.): E:111:24 S:11:99 S:109:36 E:101:-4  
W:46:254 E:58:71 N:138:352 E:121:-60 W:133:105 W:189:129 E:103:49  
N:146:329 S:55:83 S:72:153. Координати оптимального розміщення вогневих груп: (277;182), (217;64), (80;143), (62;133), (289;104), (154;40).

### 5.5 Оцінка ефективності роботи програмного комплексу

Для оцінки ефективності роботи програмного комплексу були виділені такі ключові показники ефективності:

час реакції системи

надійність системи (частота збоїв, час простою)

безпека

Зокрема для оцінки здатності системи до масштабування проведено тестування на навантаження та аналіз продуктивності системи під час пікових навантажень. На Рис.5.7 наведено графік результатів пікового навантаження на САО “Світло”.

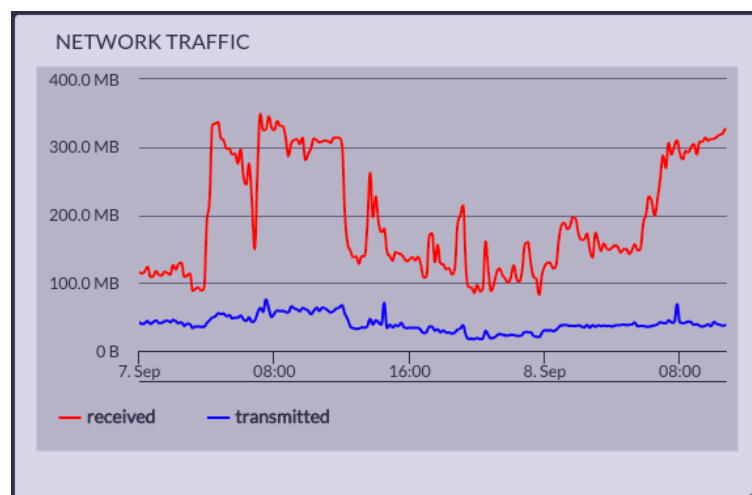


Рисунок 5.7 - Графік завантаження системи

Основна частина програмного комплексу САО “Світло” була реалізована з використанням мови програмування JavaScript. На Рис.5.8

наведено сумісність та адаптивність інтерфейсу до різних екранів та платформ, що є ключовим фактором забезпечення зручності та доступності системи для широкого спектру користувачів, адже мобільний додаток САО “Світло” реалізовувався на платформах різних мобільних пристроїв.



Рисунок 5.8 - Підтримка JavaScript різними системми

В таблиці 5.2 наведено підсумки аналізу комплексної ефективності програмного комплексу, який враховує здебільшого технічні аспекти його подальшого використання. Це дозволяє не тільки зрозуміти поточний рівень ефективності системи, але й визначити можливі напрямки для її удосконалення та адаптації до майбутніх викликів.

Таблиця 5.2

Результати комплексної оцінки програмного комплексу САО “Світло”

Метрика	Оцінка та коментар
Час Реакції Системи (Response Time)	Середній. Зумовлено акцентом на швидкості передачі даних до адміністративної панелі та мінімізацією навантаження на вхідний трафік.
Надійність (Reliability)	Високий. Забезпечується оптимальними алгоритмами удосконаленими за результатами отриманих даних в результаті активного використання системи.
Масштабованість (Scalability)	Висока. Основний акцент додатку це його використання великою кількістю користувачів.
Безпека (Security)	Висока. Початкова умова при розробці обумовлена, зокрема, зберіганням місць скупчення цивільного населення.

## 5.6 Висновки до розділу 5

У п'ятому розділі дисертаційного дослідження розглянуто розробку та впровадження системи альтернативного оповіщення «Світло» у контексті підвищення ефективності реагування на загрози, пов'язані з ракетними та іншими повітряними ударами в умовах воєнного стану. Основну увагу було зосереджено на аналізі її функціональних можливостей, зокрема інтеграції з системами ППО, і потенційному використанні геоінформаційних даних для оперативного відображення ситуації у режимі реального часу.

Окрему увагу приділено динамічній задачі оптимального розбиття території області, зокрема зони відповідальності Повітряного командування «Схід», з метою розміщення 51 вогневої групи, координати центрів яких задаються вручну з урахуванням тактичної доцільності. Розроблена математична модель дозволяє гнучко адаптуватися до змін у розстановці сил, напрямках загроз і структурі місцевості. Такий підхід має суттєве значення в умовах мобільного театру бойових дій, де швидкість прийняття рішень визначає ефективність оборони.

Завдяки поєднанню геоінформаційної платформи з динамічними алгоритмами розбиття, досягається можливість майже в реальному часі перераховувати оптимальні зони відповідальності, оперативно переорієнтовуючи ППО у разі зміни конфігурації повітряної загрози. Цей підхід закладає основу для побудови адаптивної оборонної системи, яка здатна не лише реагувати на факти ворожих дій, але й діяти на випередження.

Було здійснено порівняльний аналіз між системою «Світло» та низкою міжнародних рішень, таких як OnSolve MIR3 та AlertHub Pro, який засвідчив переваги української розробки в контексті її адаптованості до специфіки бойових дій на території України. Завдяки своїй вузькій спеціалізації та гнучкій архітектурі, «Світло» дозволяє враховувати не лише координати загроз, але й динаміку їх розвитку, що забезпечує більш точне і вчасне сповіщення підрозділів та командувань.

Таким чином, система альтернативного оповіщення «Світло» виступає не лише як інструмент інформування, але і як частина інтегрованої оборонної інфраструктури, що забезпечує зворотний зв'язок, аналітику загроз і підґрунтя для прийняття рішень щодо розгортання сил ППО. Використання ГІС-інструментів у поєднанні з алгоритмами динамічного розбиття дозволяє наблизитися до рівня автоматизованого управління розосередженими оборонними силами в межах визначених оперативних зон.

## **Розділ 6. РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ПОБУДОВИ ОПТИМАЛЬНИХ ТРАЄКТОРІЙ ДЛЯ ТРИВИМІРНОГО ДРУКУ З ВИКОРИСТАННЯМ МЕТОДІВ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН**

### **6.1 Огляд прикладної проблеми**

У сучасних умовах стрімкий розвиток технологій 3D друку привертає значну увагу науковців і промисловців. Ця технологія застосовується в широкому спектрі сфер — від виробництва до медицини та архітектури. Водночас, попри всі її можливості, оптимізація процесу й надалі залишається важливою задачею, що потребує постійного вдосконалення алгоритмів і методів формування траєкторій друку.

Ключовим елементом такої оптимізації є побудова раціональних траєкторій для 3D друку. Кожне переміщення друкувальної головки та зміна положення обладнання потребують часу, матеріалів і подальшого виправлення можливих похибок. Основна мета дослідження полягає у створенні алгоритмів і методів, що забезпечують оптимізацію траєкторій, зменшуючи витрати часу й матеріалів, що підвищує продуктивність і здешевлює виготовлення виробів. Головним завданням розділу є розроблення ефективних алгоритмів побудови траєкторій у режимі реального часу, що важливо для автоматизації виробничих процесів і скорочення використання ресурсів. Якісно сформовані траєкторії не тільки прискорюють друк, а й покращують властивості готових виробів, що має значення для сталого розвитку та екологічної безпеки.

Дослідження передбачає врахування фізичних характеристик прикладної задачі й їх відображення в математичних моделях, що дозволить створити точніші та адаптивні алгоритми, здатні компенсувати обмеження процесу й оптимізувати рухи задля економії часу та енергії. Удосконалення алгоритмів оптимізації траєкторій також включає аналіз різних типів матеріалів і особливостей конструкцій різних моделей 3D принтерів.

Використання розроблених алгоритмів може суттєво вплинути на галузі авіабудування, автомобілебудування, електроніки, медицини, освіти та наукової діяльності. Значущість такого дослідження полягає у сприянні подальшому розвитку технологій 3D друку, підвищенні продуктивності виробництва та зниженні витрат на матеріали й енергоресурси, що, у підсумку, здатне зменшити собівартість продукції та підтримати формування екологічно стійких виробничих підходів.

Багато робіт вітчизняних та закордонних авторів присвячено тематиці 3D друку та розв'язанню оптимізаційних задач, що виникають в цій проблематиці. Так робота [148] присвячена практичній архітектурі та порівнянню алгоритмів для оптимізації траєкторій сопла у технології FDM. Автори формалізують набір метрик (довжина робочих і холостих переміщень, кількість підйомів/опускань сопла, сумарний час друку) і описують підхід «двоступеневого» типу: побудова швидкого жадібного початкового рішення, а потім його вдосконалення локальними операціями (Two-Opt) та стохастичними процедурами, подібними до *simulated annealing*. Значну увагу приділено реалізації експериментальної платформи — генератору тестових моделей, модулю симуляції руху сопла з урахуванням прискорень/гальмувань та інструментам збору метрик для порівняння алгоритмів на реальних STL. У результатах автори демонструють, що гібридні стратегії зменшують сумарну довжину траєкторії і час друку в порівнянні зі «стандартним» слайсерним плануванням, але підкреслюють чутливість до вибору параметрів стохастичної частини та потребу у балансі між швидкістю збіжності й якістю кінцевого рішення. Робота є цікавою тим, що поєднує алгоритмічну частину з інструментальною реалізацією експериментів — це дає можливість відтворити досвід і порівняти підходи на однакових тестових наборах.

В роботі [154] автори зосереджуються на специфіці адитивного будівництва матеріалами типу бетону, у якому критичними є не лише

траєкторії руху сопла, а й час набору міцності суміші, послідовність шарів і структурна стабільність. Автори запропонували графо-орієнтований підхід: шар розбивається на підділянки, які формалізовано як вершини/ребра графа; ваги ребер кодують витрати на переміщення, ризик дефектів через переривання екструзії та вплив на якість форми. Ключовою ідеєю є двоетапна схема: (1) партиціонування області друку на контрольовані підзадачі з метою покращення формоутворення (graph-based partitioning), (2) застосування алгоритму мурашиної колонії (АСО) в межах кожної підзадачі для мінімізації сумарного часу і холостих переміщень. В роботі наведено як алгоритмічний опис, так і експериментальні верифікації — симуляції й фізичні друки бетонною сумішшю — які демонструють скорочення часу холостих ходів та кращу послідовність нанесення шару у порівнянні з найвними підходами. Автори також обговорюють, як легко вводити додаткові інженерні обмеження (максимальна довжина елемента, параметри схоплювання) через зміну ваг ребер, що робить підхід гнучким для різних бетонних матеріалів і умов будівництва.

У статті [94] докладно описано підхід, який перетворює проблему планування траєкторії сопла в задачу на графі (узагальнення URPP/TSP) і вирішує її модифікованим алгоритмом мурашиної колонії. Ключові елементи: представлення сегментів шару як вузлів графа, введення ваг ребер, що відображають реальні витрати (час руху, ризик ниткування, штрафи за розриви), і застосування методів кластеризації (наприклад, K-means) для розбиття великої задачі на керовані підзадачі. Автори пропонують механізми для уникнення передчасної конвергенції АСО і інтеграцію з вихідними рішеннями слайсерів. У сукупності метод демонструє зменшення часу друку, скорочення холостих переміщень та покращення візуальної якості моделей у порівнянні з традиційними підходами; в статті також детально аналізовано вплив параметрів АСО та надано рекомендації по налаштуванню для різних типів моделей.

Робота [88] переносить класичну комбінаторну задачу — Chinese Postman Problem (CPP) — у контекст друку решітчастих (lattice) структур. Автори формалізують lattice-структуру як граф, де ребра відповідають ділянкам, які потрібно екструдувати; задачею стає мінімізація загального часу за рахунок побудови ейлерового або мінімально дубльованого ейлероподібного маршруту. Особливості застосування у АМ: необхідно враховувати напрямки шари, правила безперервної екструзії, термічні наслідки повторних проходів та механічні властивості решітки. Пропонована методика комбінує рішення CPP з евристичними для мінімального дублювання ребер і дає практичні алгоритми для перетворення «звичайних» lattice-графів у маршрутні послідовності, що зменшують холості переміщення і гарантують покриття всіх необхідних ділянок. В роботі наведено симуляції та приклади, що ілюструють зниження часу та поліпшення однорідності екструзії у порівнянні з простими серіями проходів.

У статті [108] формалізовано проблему “3D printing routing” і запропоновано точну цілочисельну лінійну модель (ILP) для оптимізації шляху сопла з метою мінімізації часу друку, з урахуванням операційних обмежень (послідовність обробки контурів, унікальність проходів, обмеження на прискорення/гальмування). Оскільки точне LP/ILP-розв’язання масштабно зростає по складності, автори пропонують набір евристичних і гібридних методів: ініціалізація через LP/relaxation, локальний пошук (Two-Opt, k-opt), генетичні та жадібні стратегії, а також комбінування цих підходів для великих практичних інстанцій. На реальних тестах (реальні STL/шари) порівняння з одним із популярних відкритих слайсерів показало суттєві скорочення часу друку при збереженні або покращенні якості. Стаття також містить аналіз чутливості і обговорення можливих напрямків (паралельні алгоритми, інтеграція динамічних процесних обмежень).

Робота [95] є подальшим розвитком АСО-підходів: автори пропонують «розширену» версію мурашиного алгоритму, спеціально адаптовану до особливостей 3D-моделей і задач URPP/Nozzle-routing. Розширення включає

adaptive-механізми (динамічне регулювання кількості ітерацій залежно від складності моделі), модифіковані правила оновлення феромону із врахуванням локальних характеристик шляху (щоб уникнути передчасної конвергенції), а також спеціальні евристичні правила для зменшення ниткування при переходах. Експерименти на наборі реальних моделей показують, що розширений АСО прискорює збіжність і скорочує сумарний час друку без значного погіршення якості траєкторій; також розглянуті питання обчислювальної вартості і можливості апаратного прискорення (паралельні «мурашки», GPU).

Робота [79] присвячена плануванню і координації в системах з кількома екструдерами (multi-head printing). Автор формалізує задачу як комбіноване планування: розподіл робіт між головами (partitioning), побудова індивідуальних маршрутів і їх синхронізація з урахуванням часових вікон, уникнення колізій та оптимізації сумарного часу системи. Розглядаються стратегії балансу навантажень (щоб уникнути простоїв однієї голови та надмірного навантаження іншої) і алгоритми координації (жадібні, евристичні, правила пріоритетів). Показано симуляції, де кооперативне планування скорочує загальний час друку для великомасштабних деталей у порівнянні з поетапним друком однією головою; також обговорюються практичні аспекти — комунікаційні затримки, спільні ресурси (подача матеріалу, платформа), питання безпеки та розширюваність підходів у промислових умовах.

Вище наведено сучасні роботи із тематики оптимізаційних підходів до побудови траєкторій тривимірного друку. Як бачимо більшість досліджень зосереджені на розв'язанні задачі у дискретній постановці та використанні евристичних алгоритмів. В порівнянні з наведеними підходами в дисертаційній роботі пропонується використання неперервних моделей та методів для отримання оптимальних траєкторій без дискретних обмежень.

Для більш змістовного розуміння прикладної проблеми виділемо чотири основних технології 3D друку, що на даний час є поширеними у світі. Ці технології мають принципові відмінності та побудовані на базі різних

концепцій та підходів. Нижче описано основні відмінності технологій їх можливості та обмеження.

### **Fused Deposition Modeling (FDM)**

FDM — одна з найпоширеніших технологій 3D друку, у якій пластик плавиться та подається через екструдер, формуючи об'єкт шарами. Технологія є доступною й надійною, проте має певні обмеження щодо точності та деталізації.

### **Stereolithography (SLA)**

SLA використовує ультрафіолетове світло для полімеризації рідкої смоли. Вона забезпечує високу точність і дозволяє отримувати дуже деталізовані моделі, хоча вибір матеріалів і вартість обладнання можуть бути суттєвими обмеженнями.

### **Selective Laser Sintering (SLS)**

SLS застосовує лазер для спікання порошкового матеріалу, формуючи тверді об'єкти. Технологія підтримує різні матеріали й дає змогу створювати складні геометричні форми, але потребує дорогого обладнання та значних ресурсів.

### **Multi-Jet Fusion (MJF)**

MJF працює за принципом струменевого нанесення, використовуючи порошкові матеріали. Вона забезпечує високу швидкість і добру точність, проте вимагає спеціальних матеріалів.

Кожна з технологій має власні сильні сторони й обмеження, що визначає їх застосування в різних задачах.

Функціональні можливості систем 3D друку є ключовими для оцінки їх ефективності.

### **Максимальні розміри друку:**

FDM: Обмежені площею робочої платформи, що може бути критично для великих моделей.

SLA: Дозволяє друкувати досить великі деталі, але залежить від об'єму смоляного резервуара.

SLS: Підтримує виготовлення середніх і великих об'єктів у межах робочої камери.

MJF: Має широкі можливості щодо друку середніх і великих виробів.

### **Точність:**

FDM: Поступається іншим технологіям, зазвичай 0.1–0.2 мм.

SLA: Забезпечує дуже високу точність, менше ніж 0.05 мм.

SLS: Дає стабільну точність на рівні близько 0.1 мм.

MJF: Відзначається високою точністю 0.02–0.03 мм.

### **Швидкість:**

FDM: Має середній темп друку й зростання часу при друці великих об'єктів.

SLA: Швидкість нижча, особливо при високих вимогах до деталізації.

SLS: Забезпечує швидке виготовлення, зручне для серійних виробів.

MJF: Характеризується високою продуктивністю у серійному виробництві.

### **Матеріали:**

FDM: Підтримує широкий спектр матеріалів — пластики, гнучкі полімери, композити.

SLA: Використовує різні фотополімерні смоли та забезпечує якісну поверхню.

SLS: Працює з порошковими полімерними й металевими матеріалами.

MJF: Використовує термопластичні порошки з високою міцністю.

### **Процесор і ПЗ:**

FDM: Простий в опануванні, підходить для домашніх і малих виробництв.

SLA: Потребує суворого дотримання безпеки та спеціалізованих програмних рішень.

SLS: Вимагає високоточного промислового обладнання та складного ПЗ.

MJF: Потребує продуктивного обладнання та спеціального програмного забезпечення.

Під час вибору системи друку необхідно враховувати ці параметри, оскільки вони прямо визначають здатність обладнання виконувати конкретні виробничі задачі.

Оцінка переваг і недоліків кожної технології є важливою, оскільки дозволяє сформулювати розуміння їх можливостей і обмежень. Узагальнений порівняльний аналіз наведено в таблиці 6.1.

Таблиця 6.1.

Переваги та недоліки найбільш розповсюджених технологій 3D друку

Технологія	Переваги	Недоліки
FDM	<ul style="list-style-type: none"> <li>- Доступність обладнання та матеріалів і прийнятна їх вартість.</li> <li>- Розмаїття доступних матеріалів, серед яких пластики, еластомери та композитні суміші.</li> <li>- Легкість у використанні та поширена підтримка серед різних друкарських систем.</li> </ul>	<ul style="list-style-type: none"> <li>- Нижча точність і роздільна здатність у порівнянні з іншими методами.</li> <li>- Обмежена здатність до виготовлення високодеталізованих виробів.</li> </ul>
SLA	<ul style="list-style-type: none"> <li>- Висока точність і роздільна здатність, підходить для виготовлення деталізованих виробів.</li> <li>- Можливість використання різних типів смол і матеріалів.</li> </ul>	<ul style="list-style-type: none"> <li>- Значні витрати на матеріали та UV-обладнання.</li> <li>- Менша масштабованість порівняно з іншими методами.</li> </ul>
SLS	<ul style="list-style-type: none"> <li>- Швидкий друк, особливо ефективний для серійного виробництва.</li> <li>- Можливість застосування різноманітних порошкових матеріалів, включаючи полімери та метали.</li> </ul>	<ul style="list-style-type: none"> <li>- Дорога технологія через застосування дорогих порошкових матеріалів.</li> <li>- Вимагає спеціалізованого обладнання та керування процесом.</li> </ul>
MJF	<ul style="list-style-type: none"> <li>- Швидкий друк з високою точністю.</li> </ul>	<ul style="list-style-type: none"> <li>- Дорожча порівняно з FDM та іншими методами.</li> </ul>

- Можливість виготовлення серійних виробів з високою якістю.	- Можливі обмеження у виборі матеріалів.
--	--

Важливими факторами при обранні технології є Вартість та доступність для широкого кола користувачів, якість та деталізація отриманих деталей, а також швидкість друку. Відповідна до названих критеріїв технології, що підлягають порівняльному аналізу можна розділити на групи наведені в таблиці 6.2.

Таблиця 6.2

## Характеристики технологій тривимірного друку.

Характеристика	Технології, що мають перевагу	Технології. Що мають недоліки
Вартість та доступність	FDM, MJF	SLA, SLS
Якість та деталізація	SLA, MJF	FDM, SLS
Швидкість	MJF, SLS	SLA

Серед основних методів побудови траєкторії, що використовують координати та оснований на визначенні точок у просторі, розглядаються такі:

- Метод прямолінійних траєкторій: з'єднання точок лініями, що найчастіше використовується для заповнення контуру об'єкта.
- Метод криволінійних траєкторій: використання дуг та кривих для апроксимації форми об'єкта, що може покращити зовнішній вигляд та точність.
- Метод, що використовує вектори, орієнтовані на рух друкарської головки у визначеному напрямку. Для цього метод розглядаються такі підходи:
  - Методи різблення (slicing): Розбиття об'єкта на тонкі шари для подальшого побудови траєкторії для кожного шару;
  - Методи використання векторів шляху.

Серед методів, які базуються на математичних моделях форм та геометрії об'єктів для побудови точних траєкторій. Розглядаються такі:

- Методи, що використовують неперервні криві та поверхні;
- Методи оптимізації траєкторії.

Наступним важливим аспектом є алгоритми, що регулюють параметри руху. Вони визначають оптимальні швидкості та точність переміщень, забезпечуючи необхідну якість виробу. Частина алгоритмів використовує статичні параметри, задані до початку друку, тоді як інші застосовують динамічні налаштування, що підлаштовуються під час процесу. Сучасні 3D принтери пропонують різні методи контролю траєкторій, які визначають рух екструдера та впливають на форму й якість об'єкта, зокрема:

**PID-регулятори (Пропорційно-Інтегрально-Диференційні)**

Цей метод поєднує пропорційне, інтегральне та диференційне керування. Використовується зворотний зв'язок для корекції руху в реальному часі: пропорційна складова реагує на поточні відхилення, інтегральна враховує накопичені помилки, а диференційна прогнозує майбутні відхилення, забезпечуючи стабільність і точність.

**Траєкторії з урахуванням матеріалу**

Деякі методи беруть до уваги властивості матеріалу, такі як температура плавлення та в'язкість. Це особливо важливо при роботі з біоматеріалами або композитами, де врахування їхніх характеристик підвищує якість і надійність виробів.

**Методи зменшення вібрацій**

Контроль вібрацій під час руху друкарської головки є критично важливим. Використання алгоритмів для мінімізації вібрацій допомагає уникнути дефектів і забезпечує точність друку.

**Методи адаптивного керування**

Адаптивні алгоритми дозволяють динамічно змінювати параметри керування залежно від умов друку та геометрії об'єкта, що підвищує ефективність друку виробів різної складності та форми.

Завдання побудови оптимальних траєкторій 3D-друку передбачає системний підхід до створення та оптимізації руху екструдера. Мета полягає

у розробці програмного забезпечення, яке автоматично генерує оптимальні шляхи для друку об'єктів у форматі STL.

#### **Основні вимоги та завдання:**

1. Розробка алгоритму оптимізації траєкторії.
2. Реалізація програмного продукту.
3. Візуалізація процесу для аналізу та оптимізації друку на різних етапах.
4. Генерація G-коду, що відповідає змодельованій траєкторії, з можливістю збереження у файл.

#### **Ключові алгоритми для вирішення таких задач:**

- Алгоритм на основі задачі комівояжера (TSP) моделює планування траєкторії як пошук найкоротшого маршруту через усі точки шарів, мінімізуючи переміщення головки.
- Генетичні алгоритми імітують природний відбір та еволюцію, покращуючи рішення через відбір, схрещування та мутацію.
- Алгоритми розкладання та навчання словника розбивають модель на простіші елементи та визначають оптимальну послідовність друку через навчання.
- Алгоритм на основі колонії мурах (ACO) імітує поведінку мурах, що досліджують простір, поступово поліпшуючи траєкторії друку.

#### **Аналіз алгоритмів:**

##### **TSP**

Переваги: мінімальні переміщення, скорочення часу друку і зносу обладнання.

Недоліки: висока обчислювальна складність для великих моделей.

Ринкова вартість: високі обчислювальні витрати, які компенсуються економією часу та матеріалів.

Результати: зменшення часу друку та зносу, підвищення продуктивності.

##### **Генетичні алгоритми**

Переваги: висока адаптивність, здатність знаходити рішення у складних просторах.

Недоліки: значні обчислювальні витрати для великих моделей.

Ринкова вартість: потреба у потужному обладнанні.

Результати: покращення якості друку та ефективності використання матеріалів.

### **Розкладання та навчання словника**

Переваги: ефективно для моделей з повторюваними елементами.

Недоліки: малоефективно для унікальних або складних моделей.

Ринкова вартість: необхідне спеціалізоване ПЗ, що підвищує вартість інтеграції.

Результати: скорочення часу друку та витрат на матеріали, особливо у серійному виробництві.

### **Колонія мурах (АСО)**

Переваги: ефективно у складних просторах, адаптивно до змін умов друку.

Недоліки: потребує значного часу та ресурсів для досягнення оптимального рішення.

Ринкова вартість: високі обчислювальні витрати.

Результати: висока точність і адаптивність, що підвищує ефективність процесу та якість виробу.

**Підсумок:** вибір алгоритму має враховувати баланс між витратами на впровадження та очікуваними результатами. Складні алгоритми потребують ресурсів і високої початкової вартості, але дозволяють зменшити час друку та витрати на матеріали. Менш складні алгоритми доцільні для простіших завдань або обмежених ресурсів. Ключовим є компроміс між вартістю, ефективністю та якістю відповідно до конкретних потреб виробництва.

## **6.2. Предметна постановка задачі.**

1. Завантаження моделі: спочатку необхідно імпортувати тривимірну модель об'єкта у форматі STL або іншому, яка представляє геометричне тіло для подальшого розбиття на шари.
2. Обчислення об'єму моделі: визначається об'єм об'єкта для оцінки масштабу та глибини шарування.
3. Визначення кількості шарів: на основі заданої товщини шару обчислюється кількість шарів для поділу об'єкта.

4. Розділення на шари: об'єкт розбивається на послідовні шари, при цьому визначаються межі кожного шару, які служитимуть основою для формування шару.
5. Генерація контурів шарів: для кожного шару обчислюються замкнені контури, що описують зовнішні та внутрішні межі об'єкта на відповідному шарі.
6. Оптимізація траєкторії руху: визначаються шляхи руху для нанесення матеріалу на кожен шар. Оптимізація враховує порядок проходження контурів, щоб мінімізувати відстань переміщень і скоротити час друку.
7. Формування G-коду: для кожного шару створюється G-code, що містить команди для 3D-принтера, включаючи координати XYZ, керування соплом та подачу матеріалу.

### **6.3. Математична постановка задачі**

В рамках розробки алгоритму побудови оптимальної траєкторії тривимірного друку, як було сказано, вище виникають локалізовані області, що вимагають зупинок друкарської головки з можливістю її подільшої релокації, в тому числі на великі відстані в межах певного рівня. Такі зупинки дозволяють розділити задачі на підзадачі, для їх покрокового вирішення. Кожна з таких областей суттєвим чином пов'язана з точкою, що зумовлює необхідність такого розбиття. З математичної точки зору така постановка відповідає динамічній задачі оптимального розбиття множин з відшукуванням оптимальних координат центрів. Зауважимо, що для даної задачі, безпосередні координати розміщення центрів не є визначальними для шуканого оптимального рішення, проте надважливим є саме кількість таких центрів, тому що саме кількість центрів обумовлює кінцеву кількість зупинок та відповідні “несприятливі” події при реалізації алгоритму пошуку оптимальної траєкторії.

В математичній постановці задача має такий вигляд.

Нехай  $\Omega$  – обмежена замкнена вимірنا за Лебегом множина в 2-вимірному евклідовому просторі  $E_2$ .

Сукупність вимірних за Лебегом підмножин  $\Omega_1, \dots, \Omega_N$  множини  $\Omega \subset E_2$  назвемо можливим розбиттям цієї множини, якщо  $\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j)$ , де  $N > 0$  – задане натуральне число, що дорівнює кількості умовно-ізолюваних точок  $z$ -го рівня деталі, кількість точок в околі Мура/Неймана якої  $\leq 1$ ;  $\text{mes}(\cdot)$  – міра Лебега.

Позначимо через  $\sum_{\Omega}^N$  клас всіх можливих розбиттів множини  $\Omega \subset E_2$  на задану кількість  $N$  її вимірних за Лебегом підмножин:

$$\sum_{\Omega}^N = \{ \bar{\omega} \equiv \{ \Omega_1, \dots, \Omega_N \} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j \}$$

Далі, через  $\tau_1, \dots, \tau_N$  позначимо сукупність деяких умовно-локалізованих точок  $z$ -го рівня деталі, кількість точок в околі Мура/Неймана якої  $\leq 1$  для підмножин  $\Omega_1, \dots, \Omega_N$  відповідно, які назвемо центрами цих підмножин:  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$ , і будемо вважати, що координати усіх центрів співпадають з координатами умовно-локалізованих точок  $z$ -го рівня деталі, кількість точок в околі Мура/Неймана якої  $\leq 1$ .

**Постановка задачі.** Знайти розбиття  $\bar{\omega} = \{ \Omega_1, \dots, \Omega_N \} \in \sum_{\Omega}^N$  множини  $\Omega \subset E_n$ , векторну функцію  $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$ , визначену майже всюди (м.в.) для  $x \in \Omega$ , кординати центрів  $\tau = \{ \tau_1, \dots, \tau_N \} \subset \Omega^N$  та всіх  $t \in [0, T]$ , які забезпечують

$$\inf_{\bar{\omega} \in \sum_{\Omega}^N : \{ \tau_1, \dots, \tau_N \} \subset \Omega^N ; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot)) \quad (6.1)$$

де

$$F(\bar{\omega}, \tau, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad (6.2)$$

за наявності умов

$$\frac{dc_i(x, \tau_i, t)}{dt} = \sum_{j=1}^N A_{ij} \cdot f_j(c_j(x, \tau_j, t)), 0 \leq t \leq T \quad (6.3)$$

$$c_i(x, \tau_i, t_0) = c_{0i}(x, \tau_i), i = 1, \dots, N$$

м.в. для  $x \in \Omega$  та умови замкненості системи

$$\sum_{i=1}^N A_{ij} = 1, j = 1, \dots, N. \quad (6.4)$$

Тут  $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^2$  - координати умовно-локалізованих точок  $z$ -го рівня деталі, кількість точок в околі Мура/Неймана якої  $\leq 1$ , що мають координати  $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega, i = 1, \dots, N$ ;  $c_i(x, \tau_i, t), i = 1, \dots, N$ , - шукані дійсні функції, визначені на  $\Omega \times \Omega \times [0, T]$ , що є неперервно диференційованими за аргументом  $t$  на відрізку  $[0, T]$  м.в. для  $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$ , обмежені та вимірні за аргументом  $x$  на  $\Omega$  для всіх  $t \in [0, T]$ ;  $m(x, \tau_i), c_{0i}(x, \tau_i)$  - задані дійсні функції, визначені на  $\Omega \times \Omega$ , обмежені та вимірні за аргументом  $x \in \Omega$  (зокрема,  $m(x, \tau_i)$  може відігравати роль метрики на  $\Omega \times \Omega$ );  $f_i(c_i(x, \tau_i, t)), i = 1, \dots, N$ , - задані дійсні функції, неперервні та ліпшицеві в області свого визначення;  $\rho(x)$  - задана невід'ємна функція, обмежена та вимірна на  $\Omega$ , для простоти розв'язання задачі приймемо  $\rho = 1$ ;  $a_i, i = 1, \dots, N$ , - задані, як правило, невід'ємні числа, для простоти розв'язання задачі приймемо  $a_i = 0, i = 1, \dots, N$ ;  $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$ , - задані числові параметри;  $T > 0$  та  $t_0 \in [0, T]$  задані.

Тут та в подальшому інтеграли розуміються в сенсі Лебега. Будемо вважати, що міра множини межових точок підмножин  $\Omega_1, \dots, \Omega_N$  дорівнює нулю.

Трійку  $(\bar{w}^*, \tau^*, c^*(x, \tau, t))$ , що доставляє функціоналу (6.2) мінімальне значення на множині  $\sum_{\Omega}^N \times L_2^N(\Omega \times \Omega \times [0, T])$  за умов (6.3), (6.4), назвемо оптимальним розв'язком задачі (6.1)–(6.4). При цьому розбиття  $\bar{w}^* = \{\Omega_1^*, \dots, \Omega_N^*\} \in \sum_{\Omega}^N$  назвемо оптимальним розбиттям множини  $\Omega \subset E_n$  на  $N$  підмножин,  $\tau^* = \{\tau_1^*, \dots, \tau_N^*\} \subset \Omega^N$ ,  $\tau_i^* = (\tau_i^{*(1)}, \dots, \tau_i^{*(n)}), i = 1, \dots, N$  оптимальними центрами та їх координатами відповідно, а векторну функцію

$c^*(x, \tau, t) = (c_1^*(x, \tau_1, t), \dots, c_N^*(x, \tau_N, t)) \in L_2^N(\Omega \times \Omega \times [0, T])$  – оптимальною фазовою траєкторією динамічної системи (6.3) в задачі (6.1)–(6.4).

#### 6.4. Розробка алгоритму побудови оптимальних траєкторії 3D друку.

Виходячи з того, що траєкторія друку є послідовністю точок, що мають бути пройдені дрікарською голівкою, важливим фактором для обрання локального напрямку переміщення друкарської головки є наявність або відсутність точок, що підлягають друку в околі поточної точки. В теорії клітинних автоматів розрізняють два околии, зокрема Мура та Неймана.

##### 1. Використання околу Мура при 3D-друці:

- Оскільки алгоритм Мура враховує всі 8 сусідніх клітин (в 2D контексті), він може бути адаптований для 3D-принтерів, враховуючи додаткові напрямки руху в просторі.

- Цей алгоритм дозволить друкувальній голівці ефективно переміщуватися по діагоналях, забезпечуючи більш гладкий і неперервний процес друку, що може бути корисним для складних моделей з багатьма діагональними елементами.

- Використання алгоритму Мура може зменшити час друку і покращити якість поверхні, мінімізуючи кількість зупинок і поворотів друкувальної головки.

##### 2. Алгоритм Неймана у 3D-друку:

-Алгоритм Неймана, який враховує лише чотири ближчі клітини (в 2D контексті), може бути використаний для простіших або більш лінійних моделей, де переважає горизонтальний або вертикальний рух.

- Цей метод може допомогти у випадках, де важливо мінімізувати складність друку, зокрема, при виготовленні більш простих частин або при використанні матеріалів, які вимагають стабільності процесу друку.

- Алгоритм Неймана може також сприяти ефективнішому використанню матеріалу і зменшенню зайвих рухів друкувальної головки, що особливо

Оскільки алгоритм Мура враховує всі 8 сусідніх клітин у 2D, його можна адаптувати для 3D-друку з додатковими напрямками руху в просторі.

- Цей алгоритм дозволяє друкарській головці ефективно переміщуватися по діагоналях, забезпечуючи більш плавний і безперервний друк, що особливо корисно для складних моделей із багатьма діагональними елементами.

- Застосування алгоритму Мура може скоротити час друку та покращити якість поверхні, мінімізуючи кількість зупинок і поворотів головки.

### 3. Алгоритм Неймана у 3D-друку:

- Алгоритм Неймана, який враховує лише чотири найближчі клітини у 2D, підходить для простіших або більш лінійних моделей з переважанням горизонтального чи вертикального руху.

- Метод корисний для зменшення складності друку, зокрема при виготовленні простих деталей або використанні матеріалів, що потребують стабільності процесу.

- Алгоритм Неймана також сприяє економнішому використанню матеріалу та зменшенню зайвих переміщень головки, що важливо при великомасштабному виробництві або застосуванні дорогих матеріалів.

### Розділення на шари.

-Нехай маємо об'єкт  $O$  з об'ємом, описаним функцією  $f(x, y, z)$ , де  $0 \leq z \leq h$ , де  $h$  - висота об'єкту. Товщина шару  $dz$ . Мета - визначити границі шарів, що є основами для подальших шарів. Позначимо  $z_k$  - висота  $k$ -го шару, де  $z_k = k \cdot dz$ .

Формально:

$$S_{\downarrow k} = \{(x, y, z) \mid z = z_{\downarrow k}(x, y) \in D\}, \quad (6.5)$$

де  $D$  - площа об'єкту на рівні  $z = z_k$ , яка може бути отримана перетином об'єкта з площиною  $z = z_k$ . Цей перетин можна представити як область

$D \subset R^2$  за допомогою параметризації  $D(u, v)$ , де  $u$  та  $v$  параметри площини,  $z = z_k$ .

$$S_{\downarrow k} = \{(u, v, z_{\downarrow k}) \mid (x, y) \in D\} \quad (6.6)$$

Це дозволяє отримати границі кожного шару  $S_k$  на основі параметризації об'єкта та висоти шару. Після отримання границь шарів вони можуть бути використані для подальшої обробки та генерації контурів.

Розглянемо загальний приклад, де фігура задається функцією  $f(x, y, z)$ , яка описує об'єкт, і висоту шару  $z_k$ . Позначимо параметризацію площини на рівні  $z = z_k$ , як  $D(u, v)$ , де  $u$  та  $v$  параметри на площині  $z = z_k$ .

Таким чином, границі шару  $S_k$  можна визначити, використовуючи параметризацію  $D(u, v)$  та висоту  $z_k$  у вигляді (6.6):

Для прикладу, розглянемо параметризацію площини  $D(u, v)$  у вигляді прямокутника від 0 до 1 по обидві вісі  $u$  та  $v$ :

$$D(u, v) = \{(u, v) \mid 0 \leq u \leq 1, 0 \leq v \leq 1\} \quad (6.7)$$

Нехай висота шару  $z_k = 0,1$ . Тоді границі шару  $S_k$  будуть:

$$S_{\downarrow k} = \{(u, v, 0,1) \mid 0 \leq u \leq 1, 0 \leq v \leq 1\} \quad (6.8)$$

Це означає, що границі кожного шару представляють собою прямокутник на площині  $z = 0,1$ , що може бути використаний для подальшої обробки і генерації контурів. Параметри  $u$  та  $v$  можуть варіюватися від 0 та 1, що визначає область на площині  $z = 0,1$ .

### **Алгоритм побудови оптимальної траєкторії тривимірного друку.**

Вхідними даними алгоритму є границі об'єкта на шарі  $S_k$  у вигляді послідовності точок.

Вихідними даними є внутрішні та зовнішні контури представлені у вигляді послідовності точок.

#### **Крок 1: Визначення напрямку обходу границі:**

Крок 1.1: обираємо довільну точку на границі об'єкта як стартову.

Крок 1.2: визначаємо найближчу до неї точку, що також належить границі.

Крок 1.3: встановлюємо напрямок обходу — за або проти годинникової стрілки.

## **Крок 2: Обхід границі**

Крок 2.1: починаємо обхід з обраної стартової точки.

Крок 2.2: знаходимо наступну точку на границі, дотримуючись заданого напрямку.

Крок 2.3: повторюємо процес, поки не повернемося до стартової точки.

## **Крок 3: Розділення на внутрішні та зовнішні контури**

Крок 3.1: під час обходу фіксуємо точки у відповідний контур (внутрішній або зовнішній).

Крок 3.2: контролюємо, щоб точка обходила границю в межах обраного напрямку.

## **Крок 4: Фільтрація контурів**

Крок 4.1: відсікаємо внутрішні контури, які повністю містяться всередині інших контурів.

Крок 4.2: залишаємо лише зовнішні контури та внутрішні, що розділяють об'єкт.

## **Крок 5: Побудова оптимальної траєкторії друку 2D-зрізу**

Крок 5.1: стартова точка обирається як центр вписаного кола, що охоплює двовимірний переріз. Якщо центр належить фігурі, його використовують як початкову точку. Якщо ні, будується коло мінімального радіуса в центрі вписаного кола, яке «видаляється» з фігури. Радіус поступово збільшується до перетину з фігурою, а з точок перетину обирається початкова для побудови траєкторії друку.

Крок 5.2: Для кожної з восьми сусідніх точок до  $P_i$  із її оточення Мура розраховується вага на основі витрат та небажаних вихідних подій. Кожна з небажаних подій має свій ваговий коефіцієнт, відображаючий вплив на кінцевий результат та значення цільового функціоналу. Серед точок околу Мура обирається та, що має найменшу вагу.

Крок 5.3: Оптимальне обрання початкової точки на новому шарі здійснюється виходячи з критерію мінімальної відстані до неї від останньої надрукованої точки попереднього зрізу.

Алгоритм побудови оптимальної траєкторії тривимірного друку описаний та може бути представлений блок-схемою наведеною на рис.6.1.

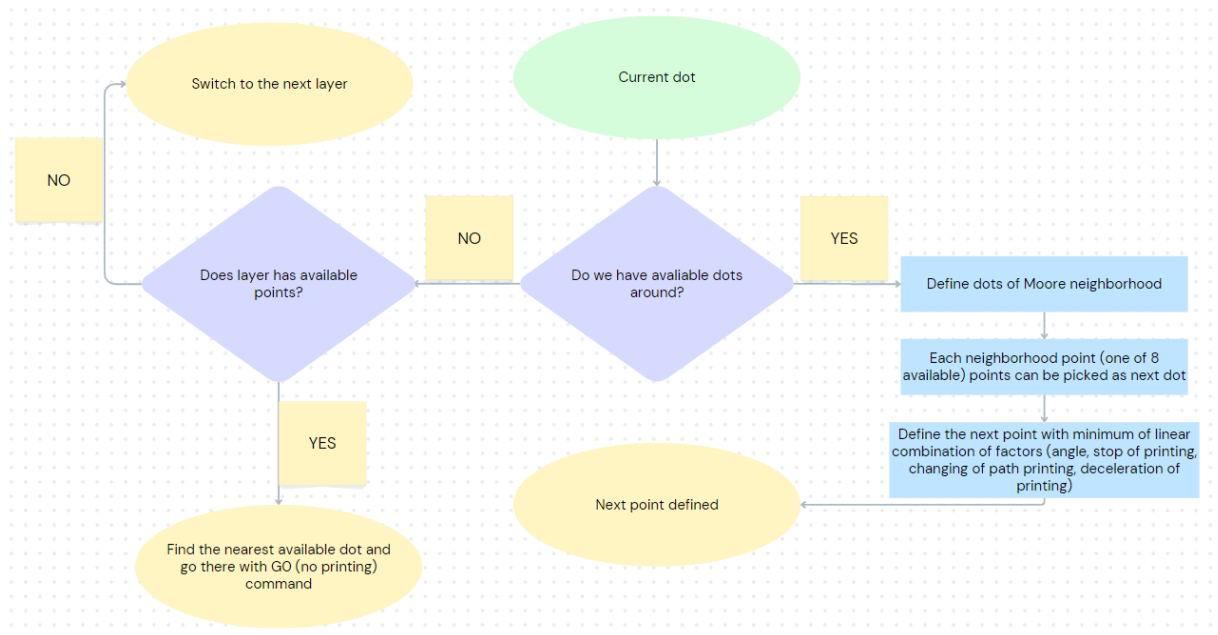


Рисунок 6.1 - Блок-схема алгоритму пошуку наступних точки

Нижче наведені критерії оптимальності, що використовуються для оцінки небажаних подій із визначенням вагового коефіцієнта залежно від прикладної задачі:

**Сповільнення (Deceleration):** ситуація, коли принтер змінює швидкість руху, наприклад, при переході на новий шар або друку складних ділянок. Перехід від високої швидкості до низької може збільшувати загальний час друку.

**Зупинка та очікування (Stopping and Waiting):** подія, що виникає, коли принтер тимчасово зупиняється перед початком нового шару або чекає підтвердження від користувача, наприклад, при зміні матеріалу чи корекції моделі.

**Переміщення між областями (Travel Moves):** відбувається, коли друкуюча головка переміщується між різними ділянками об'єкта для подальшого друку.

**Регулювання температури:** подія, пов'язана зі зміною температури екструдера або нагрівача, що може спричинити зупинки або сповільнення процесу.

**Зміна параметрів друку в реальному часі:** виникає, якщо користувач змінює параметри друку або моделі під час роботи принтера, що також може призвести до зупинок або уповільнення.

## 6.5 Результати чисельного експерименту та опис програмного додатку

На рис.6.2 наведена блок-схема, якщо описує принцип роботи програмного додатку в цілому.

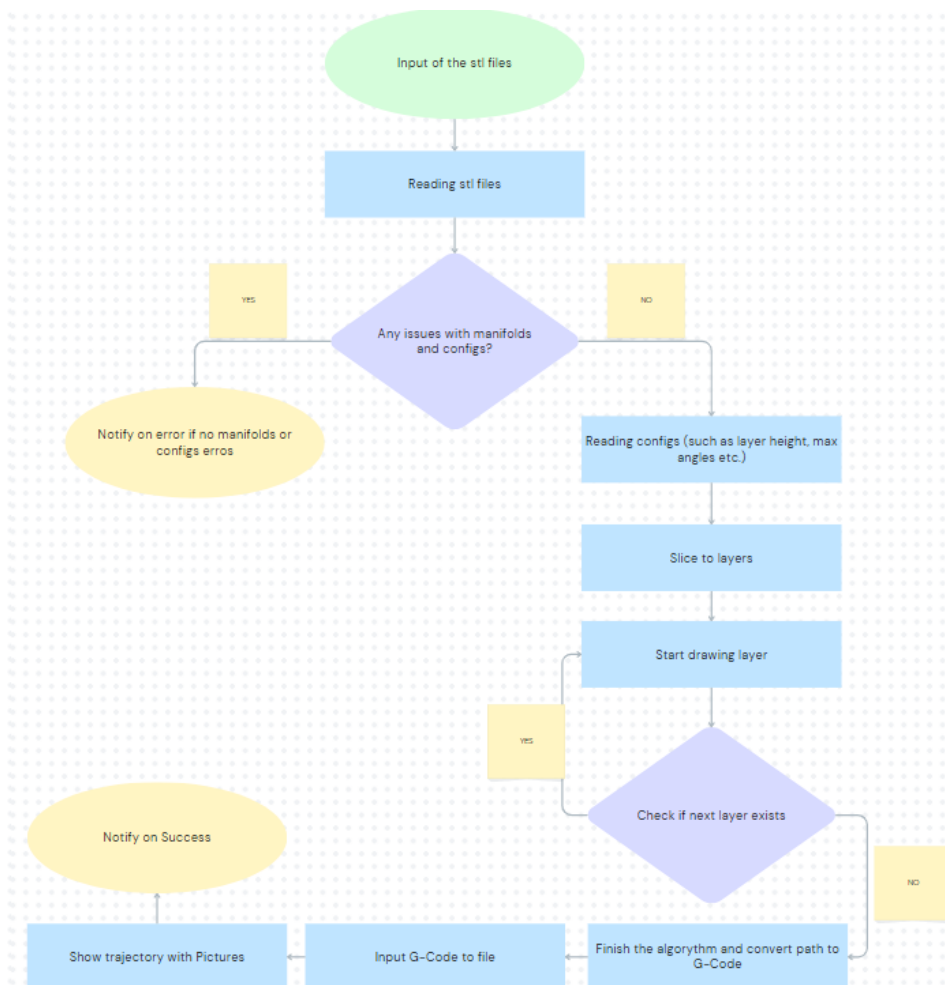


Рисунок 6.2 - Блок-схема програмного додатку

Програмний продукт розроблено на мові Python із використанням графічної бібліотеки PyQt5. Було створено графічний інтерфейс для настільних комп'ютерів і ноутбуків під MacOS, Windows та Linux. Для зручного запуску код було скомпільовано за допомогою бібліотеки Auto PY та створено виконуваний файл printer.exe.

В програмному додатку реалізовано інтерфейс, що ініціюється після його запуску. На рис.6.3 представлена стартова сторінка додатку.

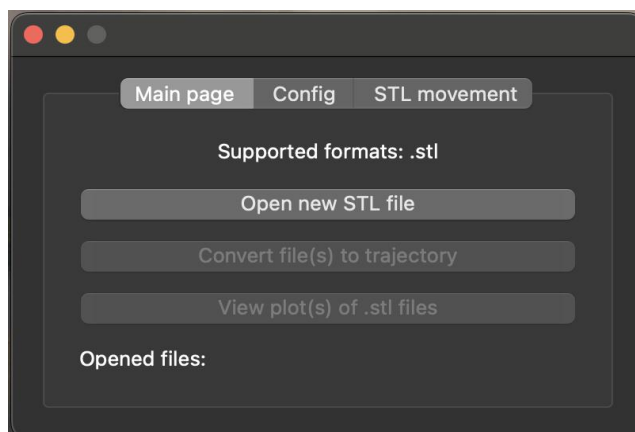


Рисунок 6.3. Стартова сторінка програмного додатку

Пункт меню “Open new STL file” надає користувачеві змогу відкрити Провідник операційної системи та обрати необхідний файл для подальшої роботи (див.рис.6.4).

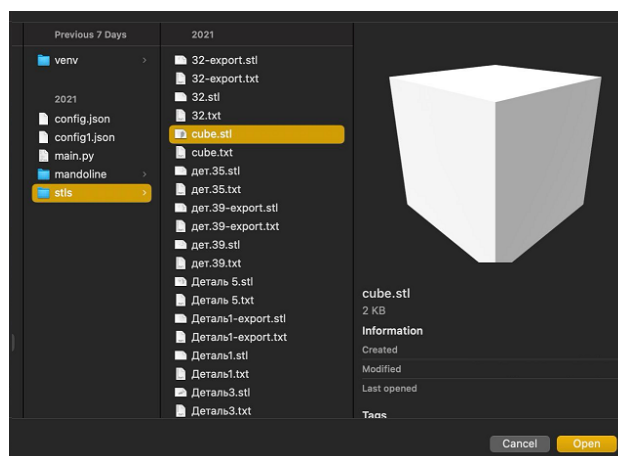


Рисунок 6.4. Інтерфейс програмного додатку для обрання вхідного файлу.

Пункт меню “Config” дозволяє користувачеві налаштувати визначальні параметри моделі, зокрема обравши крок розбиття за віссю  $z$ , ширину та висоту друкарського потоку, швидкість друку, найбільший допустимий кут між суміжними шарами тощо (див.рис.6.5).

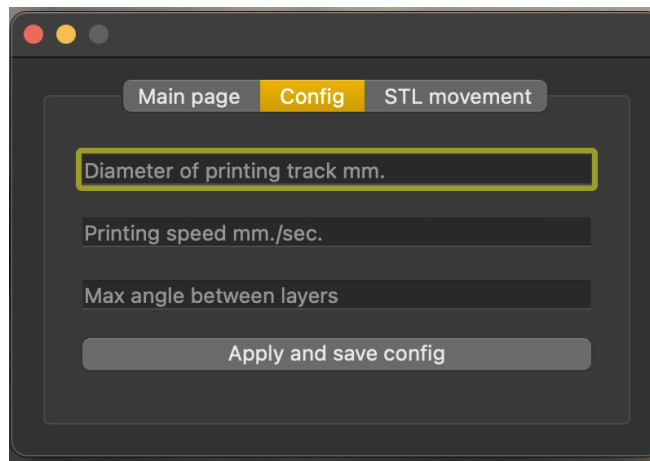


Рисунок 6.5 – Пункт меню програмного додатку для налаштувань алгоритму побудови оптимальної траєкторії тривимірного друку

Зазначимо, що переважна більшість підготовлених до опрацювання stl файлів містить деталь, умовний центр якої знаходиться в початку координат. Початкові значення центру фігури встановлені значенням 0, проте вони також можуть модеруватись можуть модеруватись користувачем до початку реалізації алгоритму (див.рис.6.6).

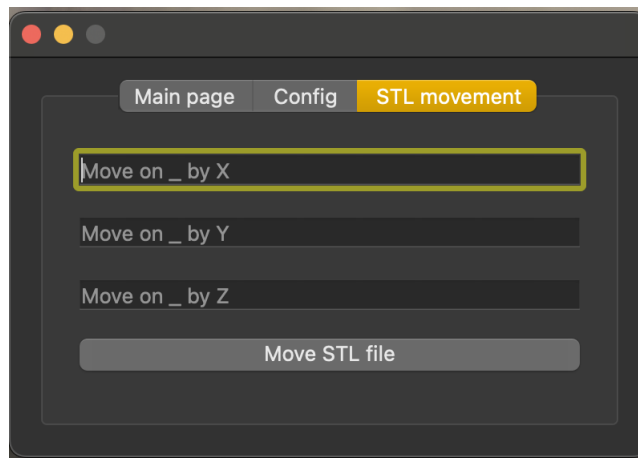


Рисунок 6.6 - Функціонал для центрування фігури, що буде друкуватися

Пункт головного меню програмного додатку “Convert file(s) to trajectory” ініціює виконання алгоритму побудови тривимірної траєкторії із обраного stl файлу до файлів-траєкторій що мають розширення .txt. Крім цього після реалізації алгоритму компонента графічної фізуюалізації програмного додатку візуалізує отриману траєкторію друку.

На рис.6.7 наведено приклад тривимірного друку для достатньо простої фігури “куб”.

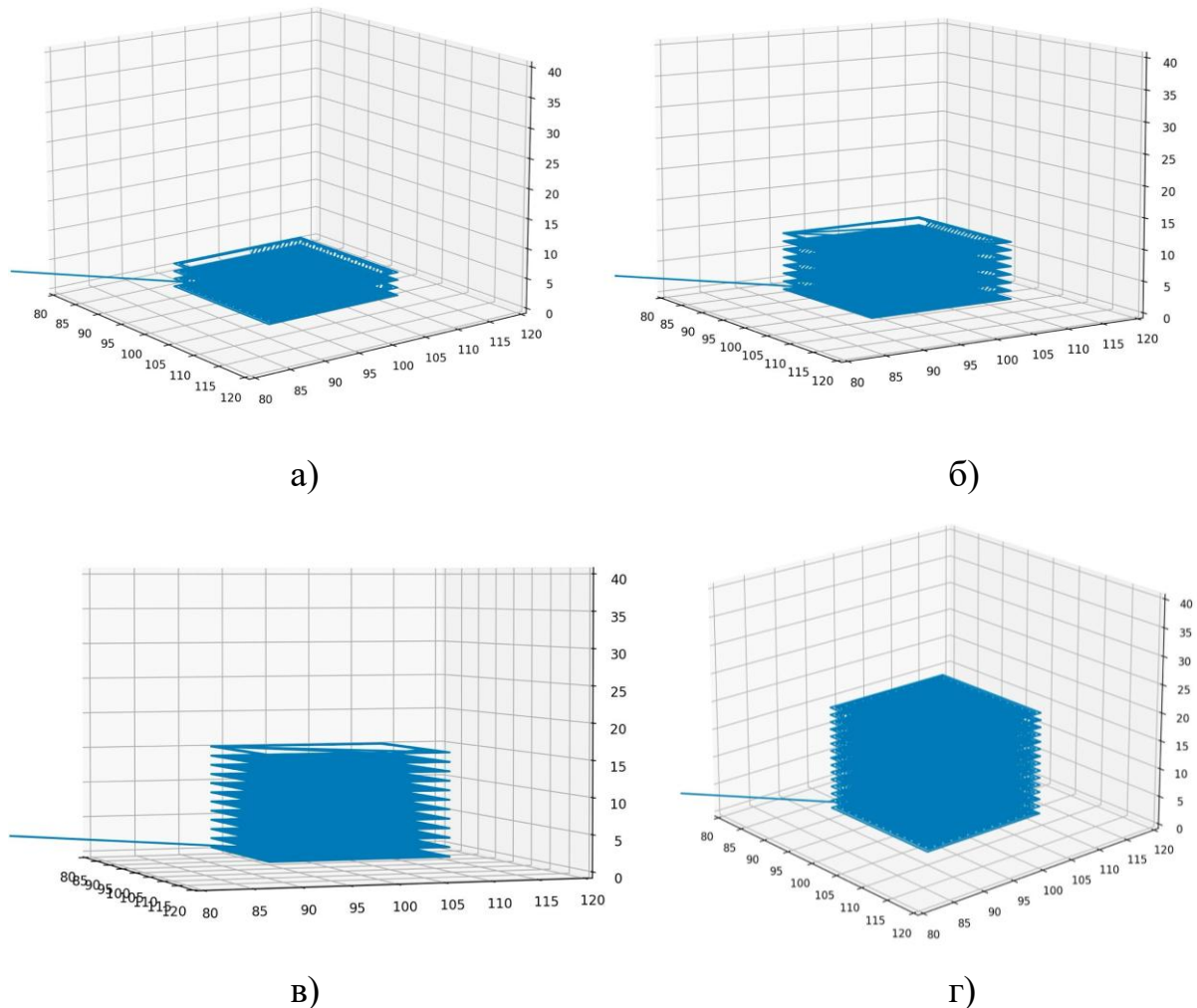


Рисунок 6.7 – Результати поетапної побудови оптимальної траєкторії деталі “куб” а) 25%, б) 50%, в) 75% г) 100%

Приклад наведений на рис.6.7 є достатньо простим, проте від достатньо наочно ілюструє результати оптимального друку та коректність роботи алгоритму на простому прикладі. Кожен зі зрізів куба визначається коректно, траєкторія друку повністю заповнює відповідний зріз та без зупинок переходе до наступного рівня.

Зважаючи на те, що кожен з рівнів деталі “куб” є подібними один одному на рис.6.8 наведено результати побудови двовимірної траєкторії на одному з рівнів названої деталі.

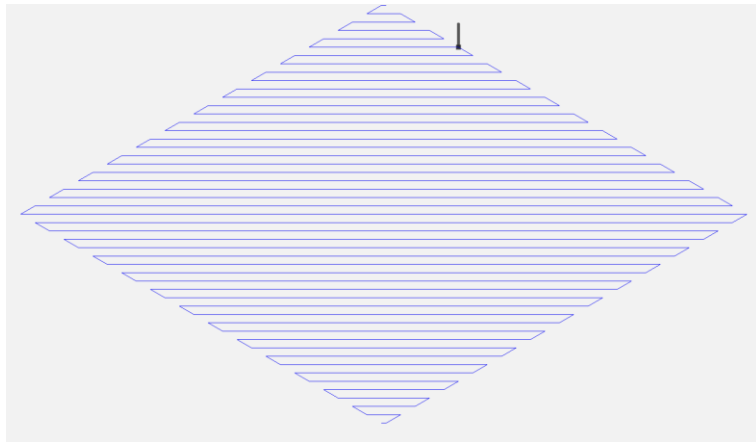


Рисунок 6.8 - Траєкторія друку першого шару фігури куб

Як бачимо, початкова точка  $P_0$  обрана на межі контуру, коректно обрано напрям обходу та кожна наступна точка обирається за принципом найближчої точки з мінімальним кутом відхилення від траєкторії на попередньому кроці. За результатами наведеними на рис.6.8 бачимо, що максимальний кут зміни траєкторії сягає  $135^0$ , що для прикладної задачі не є оптимальним рішенням, проте обумовлено формою фігури та необхідністю здійснювати розворот при проходженні конкретного зрізу деталі. Рух по контуру деталі також є можливим, але при такому рішенні середній кут зміни траєкторії є набагато більшим, на відміну від наведеного на рис.6.8 при якому переважна більшість кутів зміни траєкторії дорівнює  $0^0$ .

Більш складений приклад наведено на рис.6.9. Розглядається деталь “кільце”, що фактично має форму циліндра. Деталь розташована перпендикулярна площині  $x:0:y$ , у зв’язку з чим друк відбуватиметься вздовж вісі  $z$ .

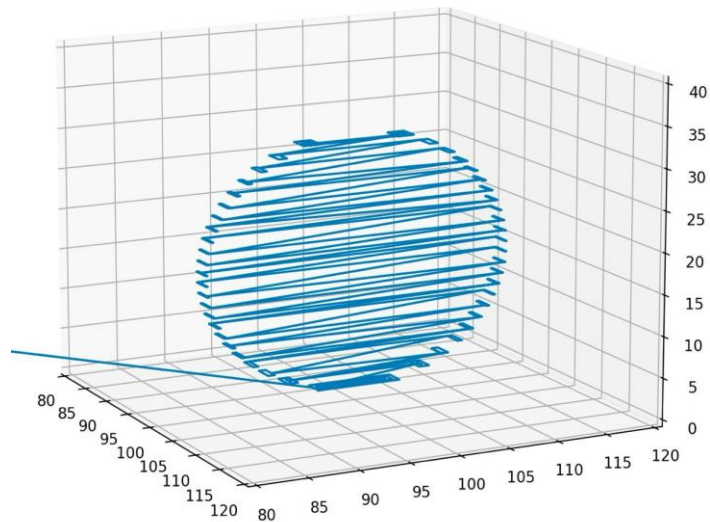


Рисунок 6.9 – Результати реалізації алгоритму побудови оптимальної траєкторії для деталі “кільце”

Як бачимо стінки деталі є достатньо тонкими, що зумовлює невелику кількість траєкторій на кожному рівні деталі. Розривність деталі на більшості зрізів призводить до зупинок принтера та ротації друкуючої голівки. На рис.6.10 представлено результати друку всієї деталі з урахуванням ширини та висоти друкуючого треку. На рис.6.10 видно, що траєкторія побудована з урахуванням мінімізації сповільнень, зупинок, розворотів і змін напрямку руху, а також зменшення прямокутних кутів, що також відображено у візуалізації траєкторії друку.

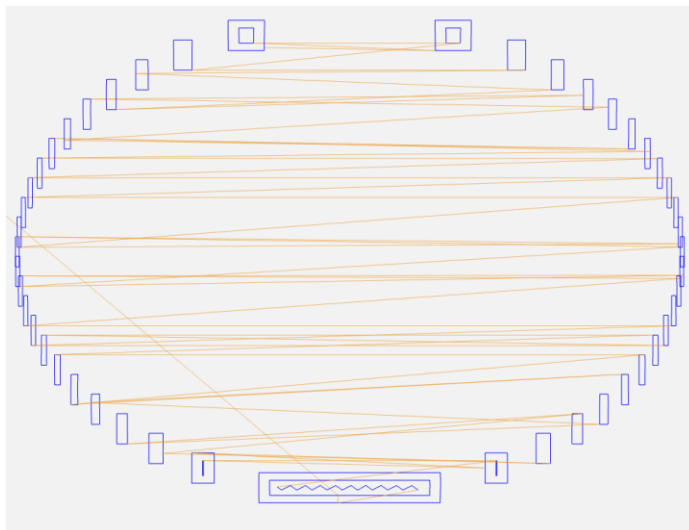


Рисунок 6.10 - Траєкторія друку деталі “кільце”

В рамках дисертаційного дослідження було побудовано велику кількість простих та складних фігур, складність яких достатньо повно

відповідає двом наведеним на рис.6.7-6.10 прикладах. Проте існують деталі параметри яких суттєвим чином конфліктують з фізичними обмеженнями принтера на якому проводились дослідження.

### **Друк надмалих деталей.**

У разі коли розміри деталей співставні, або лише у кілька разів перевищують ширину та висоту друкарського треку виникають певні складнощі з їх друком. В цілому запропонований в роботі алгоритм дає змогу побудувати оптимальну траєкторію друку деталей, проте цільовий функціонал набуває невиправдано великих значень. Такий ефект зумовлено, насамперед відносно великою кількістю зупинок, що виникає при друці, частій релокації друкарської головки, а у деяких випадках недотриманням максимального куту нахилу рівнів між собою. Проте отримані рішення є допустимими, Але вимагають подільшої і достатньо коштовного доопрацювання деталей.

На рис.6.11 наведено деталь розміри якої співставні з шириною та висотою друкарського треку.



Рисунок 6.11 - Вхідна модель фігури у форматі .stl

На першому етапі застосування запропонованого в роботі алгоритму ескіз g-code мав зайві точки та неправильну траєкторію, що було зумовлено малими розмірами деталі. Реалізація g-code цієї моделі має вигляд наведений на рис.6.12-а).

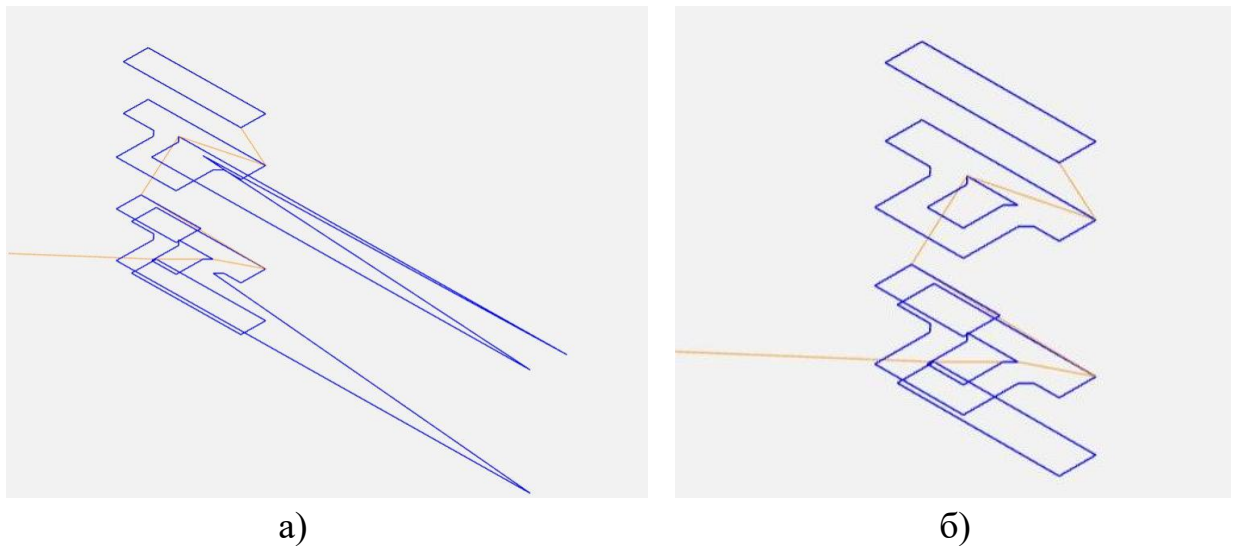


Рисунок 6.12 - результат побудови траєкторії фігури зображеної на рис.6.11 у форматі g-code

Описана проблема була вирішена шляхом масштабування розмірів друкарського треку відповідно до масштабів та розмірів деталі. Зауважимо, що у зв'язку з відсутністю можливості змінити розмір деталі, було прийнято рішення змінити розмір інструментів, які працюють з деталлю. Зміна розміру трека дозволило більш коректно відтворити її контури та інші нелінійні особливості форми. На рис.6.12-б) представлено скорегований g-code деталі, зображеної на рис.6.11 зі зміненими параметрами друкарського треку.

На рис.6.13 представлено результати побудови оптимального алгоритму тривимірного друку для ще одного прикладу надмалої фігури розміри якої співставні з шириною та висотою друкарського треку.

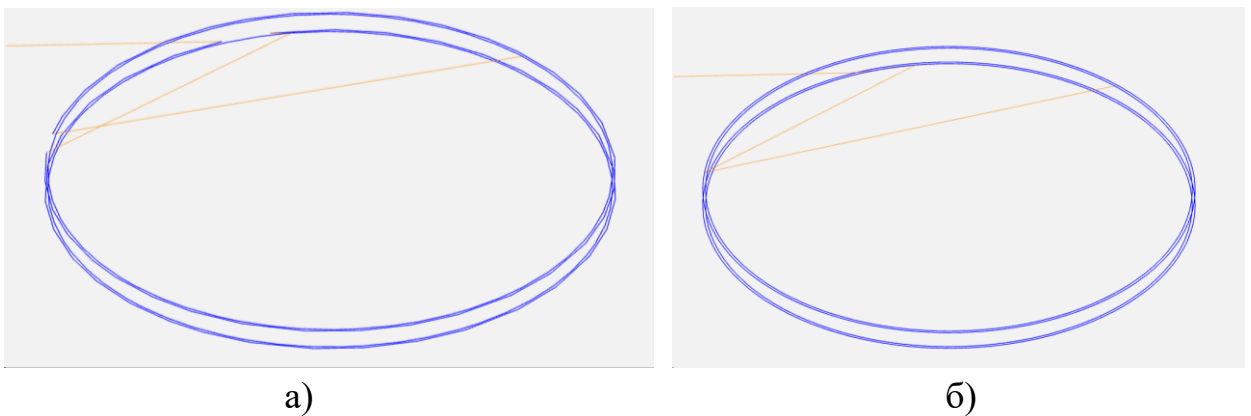


Рисунок 6.13 – Оптимальна траєкторія друку надмалюх фігури “Кільце2” із застосуванням при реалізації алгоритму околу Неймана

Як бачимо результати представлені на рис.6.13-а) ілюструють наявність певно кількості зупинок та релокації друкарської головки. Такі особливості негативно впливають на значення цільового функціоналу та якість надрукованої деталі. Масштабування ширини треку до розмірів фігури не дало необхідного результату, у зв'язку з чим було щастосовано метод відповідно до якого в алгоритмі був застосований окіл Неймана замість околу Мура. Як було сказано у цьому розділі вище, оківл Неймана застосовується сааме дл друку більш простих та менших за розміром фігур для уникнення ефекту робасного зміну кута між траєкторіями друку. На рис.6.13-б) басимо, що отримана траєкторія має менше зупинок та потребує менше релокацій друкарської головки, що у свою чергу позитивно сприяє якості отриманої деталі та зменшує значення цільового функціоналу.

#### **Друк деталей з нестійкою основою.**

Іноді, незалежно від форми, трапляються випадки, коли деталь не має точки стійкої рівноваги. У таких ситуаціях важливо передбачити додавання так званої «підпорки» для забезпечення зручного друку та коректного виготовлення деталі. Хоча це не впливає безпосередньо на слайсінг і побудову траєкторії кожного шару, під час розробки було обчислено співвідношення площі найнижчого шару до найбільшого шару моделі. Якщо це співвідношення менше заданого значення, площу нульового рівня нарощували до визначеного відсотка площі найбільшого шару. Приклад такої фігури наведено на рис.6.14-а)

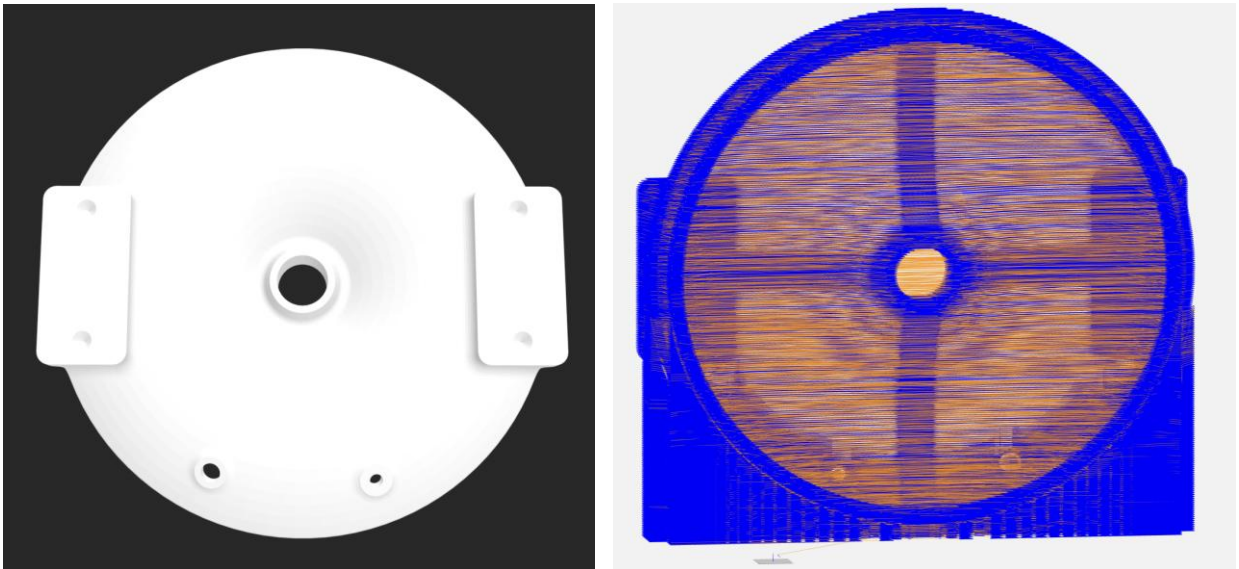


Рисунок 6.14 – Модель “Деталь17” у форматі а).stl та б) g-code

Наведені на рис.6.14-б) результати побудови оптимальної траєкторії для складної деталі “Деталь17” ілюструє адекватну поведінку запропонованого в роботі алгоритму який мінімізує кількість зупинок, ротацій друкарської головки та інших “негативних” подій, що у свою чергу негативно впливають на якість деталі та значення цільового функціоналу.

## 6.6 Висновки до розділу 6

З використанням методів та алгоритмів розв’язання динамічних задач оптимального розбиття множин, запропонованих у попередніх розділах дисертаційної роботи (як у випадку з фіксованими центрами, так і з визначенням їх координат), була розв’язана задача побудови оптимальної траєкторії тривимірного друку. У розділі виконано огляд ключових чинників, що визначають процес 3D-друку, а також факторів, які суттєво впливають на швидкість і якість виготовлення деталей зазначеним способом.

Під час розв’язання задачі побудови оптимальної траєкторії 3D-друку застосовано не лише методи та алгоритми розв’язання динамічних задач оптимального розбиття множин, але й локальні методи оптимізації траєкторії, що базуються, зокрема, на підходах математичного моделювання типу «клітинні автомати». Результати чисельних експериментів, які реалізують зазначені алгоритми, підтвердили їх ефективність для задач

такого класу. Отримані траєкторії характеризуються мінімальною кількістю переривань (зупинок), сповільнень та релокацій друкарської головки, що не лише скорочує час виготовлення та доопрацювання деталей, але й суттєво підвищує їх якість.

Програмний додаток реалізовано таким чином, щоб користувач мав змогу самостійно конфігурувати процес друку, визначаючи більшість ключових параметрів. Перелік цих параметрів сформовано так, щоб унеможливити їх конфліктні значення та забезпечити чітке налаштування програмного додатку.

Чисельні експерименти виконано як на прикладі простих деталей (куб, циліндр), так і для складніших об'єктів, зокрема тих, розміри яких співставні з шириною та висотою робочої зони друкарського треку, або тих, що мають малу чи нестійку основу. У програмі передбачено можливість усунення таких проблем ще на етапі підготовки до побудови оптимальної траєкторії. Запропонований підхід дає змогу розрахувати початкове наближення оптимальної траєкторії навіть для деталей, які іншими програмними засобами не конвертуються у траєкторію друку. Надалі це наближення може бути вдосконалене оператором, що значно скорочує його робочий час та автоматизує значну частину технічних операцій.

Вхідні та вихідні дані програмного додатку реалізовано у форматах, сумісних із сучасними системами. Вхідними даними є файл формату `\*.stl`, а вихідними — файл із `g-code`, що є загальноприйнятим стандартом числового програмного керування пристроями і містить точні інструкції для відтворення заданої моделі.

Запропонований алгоритм побудови траєкторії ґрунтується на системному підході, який дозволяє ефективно враховувати широкий спектр параметрів та обмежень. Він оптимізує довжину шляху, витраченого на друк кожного шару, та застосовує вдосконалені методи для розв'язання задачі оптимізації маршруту.

## **Розділ 7. РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ РОЗБИТТЯ ТЕРИТОРІЇ НА ЗОНИ ЗАБЕЗПЕЧЕННЯ МЕДИЧНИМИ ЗАКЛАДАМИ ПРИ СПАЛАХАХ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ**

### **7.1 Огляд прикладної проблеми**

З кінця 2020 року розповсюдження вірусу SARS-CoV-2 призвело до спалаху епідемії COVID-19 у Китаї. Масштаби розповсюдження інфекції серед населення багатьох країн було настільки масовим, що вже у січні 2020 року всесвітньою організацією охорони здоров'я спалах епідемії було кваліфіковано як надзвичайну ситуацію в області охорони здоров'я міжнародного значення, а у березні 2020 року ситуація була названа пандемією.

Термін розвитку епідемії від кількох інфікованих до десятків, а у деяких випадках до сотень вимірювався тижнями, а у деяких країнах днями. Кожна з країн, що визнала епідемію COVID-19, знаходилась у різних умовах (кількість інфікованих, стан системи охорони здоров'я, кількість обладнаних належним чином інфекційних лікарень, наявність необхідного медичного обладнання, наявність кваліфікованого медичного персоналу, законодавство, що регулює можливість введення обмежувальних заходів тощо), що безумовно вплинуло на перебіг епідемій в кожній окремій країні. Стрімкий розвиток пандемії показав, що не завжди стала економіка та врегульована законодавча база дає змогу оперативно реагувати на виклики такої надзвичайної ситуації, якою виявилась пандемія COVID-19.

Сьогодні небезпека розповсюдження названої інфекції залишається, проте за п'ять років науковцями всього світу отримано достатньо інформації, що дозволяє будувати адекватні математичні моделі та прогнозувати стан розповсюдження хвороби, як на окремих територіях так і в рамках міст, країн, континентів. Очевидним є той факт, що організаційні методи боротьби з розповсюдженням COVID-2019 за результативністю не поступаються медичним. Після стрімкого зростання кількості інфікованих у

густонаселених містах та їх районах, відсутності в окремих країнах необхідної кількості обладнання для догляду за інфікованими пацієнтами, відмовами деяким пацієнтам з тяжким та середньо тяжким станом захворюваності у госпіталізації важливою залишається задача прогнозування чисельності інфікованих на деякий достатній для прийняття необхідних організаційних заходів період та оцінка якісних етапів розвитку епідемій на окремих територіях. Зрозумілим є той факт, що метою математичного моделювання розповсюдження COVID-2019 є не тільки оцінка швидкості, термінів та етапів її розповсюдження, а й ефективність тих чи інших заходів по боротьбі з названою інфекцією. Очевидно, що кожен комплекс заходів (закриття магазинів, місць громадського харчування, державних установ, розважальних центрів, навчальних закладів, обмеження руху громадського транспорту тощо) має суттєві соціальні, виробничі та економічні наслідки і саме їх збалансоване застосування дає можливість ефективної боротьби з розповсюдженням інфекції, знизивши негативний вплив перерахованих наслідків.

На сьогоднішній день існує достатня кількість математичних моделей для оцінки швидкості розповсюдження COVID-19, при цьому кожна з таких моделей, як правило орієнтована на окремий регіон (країну, область, місто тощо). Кожен з таких регіонів суттєвим чином відрізняється один від одного віковим складом населення, хронічними захворюваннями, характерними для того чи іншого регіону, соціальною активністю різних вікових груп тощо.

Кожен з цих факторів окремо не чинить суттєвого впливу на динаміку зростання кількості інфікованих, але у зв'язку з тим, що така динаміка є, як правило, експоненційною, то чисельні показники на достатньому проміжку часу суттєвим чином відрізняються. Прикладом може бути Італія, де люди похилого віку, на відміну від України, характеризуються високим рівнем соціальної активності. Така ситуація у 2020-му році суттєвим чином вплинула на кількість інфікованих серед групи населення похилого віку і, як

наслідок, на завантаження лікарень “тяжкими” хворими та колапсу системи охорони здоров’я в Італії.

У зв’язку з тим, що загальна кількість інфікованих за останні п’ять років у світі вимірюється мільйонами, існує достатньо достовірна інформація про перебіг епідемії на окремих територіях, при цьому навіть досвідчені фахівці (епідеміологи) можуть лише описати перебіг захворюваності, але “механізми” інфекції та її вплив на організм описати у більшості випадках складно.

Більшість фахівців для моделювання процесу розповсюдження інфекції використовують SIR моделі та їх розширені варіанти [64], [76], [156]. Враховуючи той факт, що більшість провідних епідеміологів вважає, що “групового імунітету” для COVID-19 не існує, такий підхід є достатньо виправданим, але очевидно, що у більшості країн достовірні данні про кількість інфікованих та померлих відсутні, похибка для моделі SIR призводить до її стрімкого накопичення та, як наслідок, хибних результатів моделювання. Крім цього SIR модель враховує лише три принципові показники тобто не може передбачити вплив на систему обмежувальних заходів, а їх на сьогодні було винайдено чимало. Фактична констатація перебігу розповсюдження інфекції малоінформативна, а бажаною є комплекс заходів для щонайменш чисельного сповільнення розвитку епідемії або якісних змін у її перебігу.

Динаміка інфекції значно варіювалися між країнами і залежала від численних факторів, зокрема від густоти населення, стану системи охорони здоров'я та обмежень, накладених органами влади. Багато дослідників успішно застосовували відомі та доволі адекватні математичні моделі, такі як SIR, SEIR та SIS, для моделювання цих процесів. Однак ключове обмеження цих моделей полягає в їх типовому застосуванні без врахування просторового розподілу. Такий підхід не враховує кілька важливих факторів, таких як густота населення, транспортні потоки, місця з високою концентрацією людей та ефективність протиепідемічних заходів.

Інфекційні хвороби представляють серйозну загрозу для громадського здоров'я вже більше століття, призводячи до численних регіональних епідемій та глобальних пандемій. Історичні записи свідчать, що кількість інфікованих коливалася від тисяч до сотень мільйонів, при цьому смертність варіювалася від сотень до десятків мільйонів. Незважаючи на значний прогрес в діагностиці, профілактиці та лікуванні, точне прогнозування динаміки інфекцій — зокрема пікових значень захворюваності та тимчасової еволюції — залишається важливою складовою ефективного управління епідеміями. Надійне прогнозування дозволяє своєчасно реагувати на загрози громадського здоров'я та приймати обґрунтовані політичні рішення, що можуть значно пом'якшити соціальні та економічні наслідки спалахів захворювань.

Пандемія COVID-19, викликана послідовними хвилями різних штамів вірусу SARS-CoV-2, підкреслила важливість надійного епідемічного моделювання. З 2020 по 2025 рік вірус більше 700 мільйонів людей по всьому світу, а смертність досягла десятків мільйонів.

Математичні моделі, такі як SIS, SIR та SEIR, широко використовуються для вивчення поширення інфекційних хвороб та підтримки прийняття рішень під час спалахів. Однак класичні компартментні моделі зазвичай не враховують просторову гетерогенність, що обмежує їх здатність охоплювати специфічну для регіону динаміку. Зокрема, вони часто не враховують важливі просторові фактори, такі як географічний розподіл населення, мережі мобільності, регіональні відмінності у потужностях системи охорони здоров'я та локалізовані ефекти заходів з обмеження поширення. Усунення цієї проблеми є важливим кроком у розробці більш точних і дієвих моделей для прогнозування епідемій та контролю над ними.

Для побудови запропонованої в роботі математичної моделі були прийняті за основу моделі, які не тільки адекватно, з нашої точки зору, описують динаміку інфекційних захворювань, а й ті, що на практиці довели свою адекватність та достовірність.

Одним з фундаторів математичного моделювання епідемій можна вважати американського дослідника Росса який вперше розробив моделі міграції носіїв інфекції у 1905 році [170], а у 1908-1910 роках [169], [171] запропонував модель розповсюдження самої інфекції. Результати, отримані Россом, були частково покладені в основу досліджень Макдональда [155], завдяки чому, вже в середині ХХ сторіччя вдалось побудувати модель розповсюдження інфекції, враховуючи ентомологічні та демографічні особливості популяції.

Подальший розвиток означені моделі отримали кілька десятиліть потому в роботах Хоппенстеда 1974р., в яких була опублікована нова модель епідемії, що враховує вікову структуру популяції [103], а також був запропонований клас диференціальних систем [104], за допомогою яких можна описати динаміку розповсюдження інфекції.

Також свій розвиток моделі, запропоновані Россом, набули в роботах Бейлі у 90-х роках ХХ сторіччя. Бейлі вдалось розширити модель Росса та представити модель розповсюдження інфекції у вигляді SEIR та SEI моделей. Після публікації Бейлі розвиток SEIR та SEI моделей було покладено в роботу Ньютона та Рейтера, дослідження яких були присвячені питанню розповсюдження лихоманки денге. Інші якісні результати дослідження SEIR та SEI моделей були отримані в роботах [97], [150], [168], [178], [196]. В [97] за допомогою поділу людської популяції на скінчену кількість вікових груп. Автор визначив базовий репродуктивний номер хвороби для задачі SEIR системи, що залежить від віку.

Серед найбільш адекватних моделей по суті, на думку авторів, можна вважати модель, запропоновану в роботі [150]. Метапопуляція представлена як сукупність трьох (SEI), а у деяких випадках чотирьох (SEIR) підгруп. У запропонованій авторами моделі різні підгрупи S, E, I та R означають, відповідно: сприйнятливих до хвороби; інфікованих, але не заразних; інфікованих заразних та імуностійких до хвороби особин. Особливістю запропонованої моделі є можливість врахування віку людей, що особливо

актуально при нерівномірному розподілі ризику зараження та наслідків коронавірусного зараження людей різного віку. Крім цього в роботі поставлена та успішно вирішена лінійна за фазовою змінною задача оптимального керування системою.

Серед успішних прикладів математичного моделювання розповсюдження вірусних інфекцій можна назвати модель, запропоновану в роботах [168], [178]. Серед особливостей математичних моделей у зазначеній роботі, можна виділити те, що в них враховані ті самі фактори, якими у наш час керівництво різних країн намагається боротись із розповсюдженням коронавірусної інфекції. У роботах [168], [178] автор використовує у якості інструменту моделювання системи автономних диференціальних рівнянь (САДР). Проблему неможливості врахування особливостей просторового розподілу системи при використанні САДР автор вирішує шляхом збільшення кількості рівнянь, що відображають особливості розподілу вірусної інфекції на різних територіях.

Традиційно для таких досліджень використовуються класичні компартментальні моделі. Такі моделі дозволяють, розділяючи населення на класи відповідно до стану здоров'я, забезпечують аналітично керовану структуру для оцінки епідемічних порогів, чисел відтворення та потенційних масштабів спалахів [196]. Однак, попри їхню високу адекватність, ці моделі часто передбачають гомогенне змішування населення та не враховують просторову або поведінкову деталізацію, що обмежує їхню точність у відтворенні складності реальних епідемій.

Враховуючи ці обмеження, нещодавні дослідження розширили класичні підходи, включивши додаткові чинники, такі як часові затримки, поведінкові реакції та просторову гетерогенність. Наприклад, динамічні моделі із затримками враховують інкубаційні періоди та латентні ефекти, критично важливі для таких захворювань, як COVID-19 [52], [59], [60]-[63], [67], [81], [84]. Клітинні автомати та мультиагентні симуляції використовувалися для моделювання локальних взаємодій та гетерогенних

контактних патернів, що дозволяє більш реалістично досліджувати стратегії втручання [52], [59]. Такі підходи поєднують епідеміологію зі складною динамікою систем, що спостерігається в екологічному моделюванні, підкреслюючи спільні проблеми прогнозування нелінійних явищ.

Просторова структура стала центральним аспектом у моделюванні епідемій, особливо для відображення розповсюдження захворювань у географічно гетерогенних територіях. Просторово розподілена модель SIR, інтегрована з поведінкою населення, демонструє, як локальна щільність, мобільність і рівні імунітету впливають на динаміку передачі [60]-[63]. Такі просторові моделі підвищують точність прогнозів порівняно з гомогенними моделями, враховуючи регіонально специфічні фактори та дозволяючи здійснювати цільові заходи громадського здоров'я, такі як локалізовані локдауни або кампанії з вакцинації [67], [81], [84]. Подібні підходи, застосовані до спалахів вірусу Ебола, демонструють, як доступ до медичних послуг та розподіл населення модулюють результати епідемій [109], [128], [130], [135], [146].

У рамках цього ширшого тренду регіональне моделювання епідемій стало практично орієнтованим підкласом просторових моделей, де географічна область відповідає адміністративним одиницям (наприклад, регіони, області, муніципалітети) з наявними епідеміологічними та організаційними даними. Ці моделі надають практично корисну інформацію для регіональних органів влади, поєднуючи динаміку інфекцій із плануванням ресурсів охорони здоров'я, стратегіями стримування та логістичною координацією. Регіональні моделі також дозволяють проводити ретроспективну перевірку на основі детальних даних та підтримують моделювання сценаріїв за обмеженої доступності даних, що є важливою рисою для країн із нерівномірною інфраструктурою спостереження.

Поза просторовими аспектами, останні досягнення підкреслюють інтеграцію часової динаміки, стохастичності та теорії мереж. Огляди показують, що класичні моделі мають розвиватися шляхом включення

розгалужених процесів, параметрів, що змінюються з часом, та методів машинного навчання, керованих даними, для уточнення прогнозів та оцінки параметрів [147], [152]. Байєсівські підходи, зокрема, забезпечують надійну роботу з невизначеністю та гетерогенними джерелами даних, покращуючи як числові, так і якісні результати моделювання [156]. Гібридні моделі, що поєднують компартменти SEIR із регресією Кокса або іншими моделями виживання, додатково підвищують аналітичну глибину, особливо коли доступні дані за віковими групами або контактними патернами [157].

Практичне застосування епідемічних моделей значною мірою залежить від їхнього калібрування та валідації на основі реальних даних. Ретроспективні аналізи, що порівнюють прогнози моделей із спостережуваною епідеміологічною статистикою, підкреслюють необхідність ітеративного вдосконалення моделей і обережності при їх інтерпретації [168], [174]. Включення даних про мобільність, матриць соціальних контактів та динамічної оцінки параметрів підвищує адаптивність моделей до змінних епідемічних умов, таких як поява нових варіантів вірусу або зміни у поведінці населення [178], [184].

Крім того, метапопуляційні та регіонально структуровані моделі, що поділяють населення на епідеміологічно різні підгрупи, відтворюють гетерогенність, критичну для точних прогнозів, особливо в різноманітних місько–сільських контекстах [187].

Попри ці досягнення, прості моделі залишаються цінними інструментами. Парсимоніозні моделі можуть ефективно описувати фази епідемії та вплив заходів із меншим числом параметрів, що полегшує аналітичне розуміння та підвищує обчислювальну ефективність [190]. Порівняльні дослідження також демонструють, що математичні моделі часто перевершують експертні оцінки у прогнозуванні динаміки спалахів, підкреслюючи їхню роль у прийнятті рішень на основі даних [196].

Важливим сучасним викликом є динамічний характер епідемічних даних, обумовлений факторами, такими як поява нових вірусних варіантів,

кампанії вакцинації та зміни в поведінці людей. Це потребує, щоб моделі були не лише просторово та поведінково інформованими, але й динамічно адаптивними. Техніки машинного навчання та асиміляції даних все частіше інтегруються в рамки епідемічного моделювання для забезпечення оновлення параметрів у режимі реального часу та підвищення точності прогнозів [81], [109], [147]. Такі гібридні підходи поєднують механістичне розуміння з емпіричними даними, пропонуючи потужну парадигму для прогнозування та контролю епідемій.

У цілому, сучасний стан наукових досліджень з цієї тематики окреслює чітку траєкторію: моделювання епідемій поступово стає більш просторово орієнтованим, поведінково інформованим, обчислювально складним і адаптивним до даних. Інтеграція механізмів просторового розповсюдження в класичні компартментальні рамки, підтримана асиміляцією даних у реальному часі та перевірена комплексними ретроспективними дослідженнями, ілюструє цю тенденцію [59]–[63], [67], [81], [84]. Регіональне моделювання епідемій є практичною та масштабованою реалізацією цієї траєкторії, адаптованою до потреб прийняття рішень місцевими органами охорони здоров'я.

У цьому розділі дисертаційної роботи буде створено просторово та регіонально адаптоване розширення сімейства моделей SIS/SIR/SEIR, каліброване для прогнозування епідемії у двох українських регіонах — Дніпропетровській та Харківській областях. Запропонований в роботі клас моделей інтегрує гетерогенні взаємодії населення, моделі мобільності та поведінкові реакції, зокрема для Дніпропетровської та Харківської областей під час пандемії SARS-CoV-2. Поєднуючи класичну теорію з просторовою та поведінковою реалістичністю і включаючи компоненти, що адаптуються до даних, це дослідження має на меті підвищити точність прогнозів та надати практично корисну інформацію для локалізованого контролю епідемій.

## 7.2 Предметна постановка задачі

Більшість математичних моделей для вивчення та прогнозування динаміки інфекційних захворювань зосереджені на передбаченні кількості сприйнятливих, інфікованих та вилучених осіб, де до групи «вилучених» належать ті, хто або одужав і набув імунітету, або помер. Основною математичною базою для таких моделей є системи звичайних диференціальних рівнянь, які підтримуються відповідними аналітичними або чисельними методами. Однак цей підхід зазвичай обмежується так званим «точковим» моделюванням, яке передбачає, що все населення зосереджене в одній просторовій точці, ігноруючи просторовий розподіл. Зважаючи на те, що інфекції передаються через міжособистісні контакти, просторовий розподіл суттєво впливає на результати моделювання і має бути врахований. Не менш важливими є транспортні потоки між регіонами, які не лише сприяють регіональному поширенню інфекції, але й створюють середовища підвищеного ризику масової передачі, такі як громадський транспорт. Ці чинники не враховуються у точкових моделях.

Класична модель SIR поділяє населення на три групи:

- **S (Susceptible)** — сприйнятливі особи, які можуть інфікуватися при контакті з інфікованими з певною ймовірністю;
- **I (Infected)** — інфіковані особи, здатні передавати інфекцію сприйнятливим через інфекційний контакт;
- **R (Removed)** — вилучені особи, які більше не є інфікованими, включаючи тих, хто одужав з набутим імунітетом, або помер.

Ключовим припущенням моделі SIR є те, що повторне інфікування неможливе. Однак реальні дані підтверджують випадки повторного зараження, навіть тією ж мутацією вірусу, що ставить під сумнів це припущення. Сучасні моделі, які краще відповідають реальним умовам, враховують можливість реінфекції та використовують підходи із запізненням, щоб врахувати час, необхідний для формування імунітету.

Попри свою корисність, модель SIR має кілька обмежень. До них належать: ігнорування просторового розподілу, нездатність враховувати чинники, що зумовлюють швидке поширення хвороби (наприклад, транспортні потоки або щільність населення), часові обмеження моделювання та жорстка структура, яка передбачає лише переходи між трьома дискретними станами. Хоча модель SIR може адекватно описувати короткострокову динаміку захворювання в популяції, її точність у часовому вимірі часто є обмеженою, особливо при довгострокових прогнозах.

У загальному вигляді модель SIR [76] описується наступною системою звичайних диференціальних рівнянь:

$$\begin{cases} \dot{S} = -\beta \cdot \frac{S \cdot I}{N} \\ \dot{I} = \beta \cdot \frac{S \cdot I}{N} - \gamma \cdot I \\ \dot{R} = \gamma \cdot I \end{cases} \quad (7.1)$$

де:

$S$  — кількість сприйнятливих осіб;

$I$  — кількість інфікованих осіб;

$R$  — кількість вилучених осіб (тих, хто одужав з імунітетом або помер);

$N = S + I + R$  — загальна чисельність популяції (метапопуляції);

$\beta$  — коефіцієнт інфікування, який відображає ймовірність того, що сприйнятлива особа заразиться при контакті з інфікованою;

$\gamma$  — коефіцієнт вибування, який визначає середню ймовірність того, що інфікована особа одужає або помре.

Ця модель є базовою основою, однак потребує розширення для врахування просторових і часових складностей, які спостерігаються в реальних епідеміях.

Ключовим розширенням моделі SIR є модель SEIR, яка вводить додаткову категорію —  $E$  (Exposed), тобто осіб, які перебувають у періоді інкубації: вони вже інфіковані, але ще не є заразними [64]. У цій моделі

сприйнятливі особи ( $S$ ) стають підданими ( $E$ ) після контакту з інфікованими ( $I$ ) і переходять до інфекційного стану ( $I$ ) після завершення інкубаційного періоду.

Модель SEIR описується наступною системою звичайних диференціальних рівнянь:

$$\begin{cases} \dot{S} = -\beta \cdot \frac{S \cdot I}{N} \\ \dot{E} = \beta \cdot \frac{S \cdot I}{N} - \alpha \cdot E \\ \dot{I} = \alpha \cdot E - \gamma \cdot I \\ \dot{R} = \gamma \cdot I \end{cases} \quad (7.2)$$

де:

$E$  — кількість осіб у періоді інкубації (піддані, але ще не заразні);

$\alpha$  — швидкість переходу від стану підданості до інфікованості, яка є оберненою до середньої тривалості інкубаційного періоду;

інші позначення ( $S, U, R, \beta, \gamma, N$ ) відповідають моделі SIR.

Основною перевагою моделі SEIR є здатність враховувати інкубаційний період, що дозволяє моделювати затримку у зростанні кількості інфікованих осіб ( $I$ ). Ретроспективні аналізи продемонстрували вищу точність цієї моделі порівняно з SIR, особливо для захворювань зі значним латентним періодом [64].

Існує багато модифікацій моделі SIR, які в сукупності отримали назву компартментних моделей (compartmental models) і адаптуються до специфіки різних захворювань. До них належать:

1. Карантинні заходи: модель SIQR [160] включає додаткову категорію — Q (Quarantine) — осіб, які перебувають в ізоляції, що дозволяє знизити рівень передачі інфекції.
2. Втрата імунітету: модель SIRS [53] дозволяє особам, які одужали (R), втрачати імунітет і з певною швидкістю повертатися до категорії сприйнятливих (S), що дозволяє враховувати повторне зараження.

3. Групи ризику: деякі моделі стратифікують сприйнятливую популяцію на  $K$  груп ризику, кожна з яких має власний коефіцієнт зараження  $\beta_k$ , де  $k = 1, 2, \dots, K$ . Сприйнятлива популяція в  $k$ -й групі позначається як  $S_k$ , що дозволяє враховувати відмінності в ризиках передачі залежно від віку, соціальної активності, умов праці тощо.

Об'єктом дослідження є людська популяція Дніпропетровської та Харківської областей України та процес розповсюдження в ній інфекції COVID-19. Необхідно підкреслити, що на момент дослідження відсутні достовірні данні про джерело, “нульового пацієнта”, супутніх захворюваннях та ін.. Статистичні дані зібрані про інфікованих, як правило, не є репрезентативними. Актуальність, повнота, достовірність та інші характеристики статистичної інформації не є задовільними.

В основу дослідження було покладено розуміння про популяцію як про множину особин, яку умовно можна поділити на такі групи:

- здорове (неінфіковане) населення;
- інфіковані пацієнти;
- пацієнти які вибули з дослідження (такі, що померли або такі що вилікувались);
- пацієнти, що мешкають у певних територіальних секторах;
- чоловіки та жінки;
- вікова група (до 15 років, від 15 до 55 років, старше 55 років);
- медичний та допоміжний персонал задіяний в обслуговуванні інфікованих.

На момент дослідження прийнято вважати, що повторне інфікування людини малоімовірно. Факт отримання інфікованим пацієнтом імунітету після одужання не підтверджено, але при цьому світова статистика захворюваності показує, що приклади повторного інфікування або відсутні, або зустрічаються досить рідко, що у свою чергу може бути обумовлено незавершеністю лікування та хибно негативними результатами тестування на момент виписки з лікарні.

Дослідження розповсюдження COVID-19 у Дніпропетровській області було проведено авторами відповідно до п.1 протокольного доручення від 08.07.2020 року Голови надзвичайної протиепідемічної комісії при Дніпропетровській обласній державній адміністрації та Дніпропетровській обласній раді. В основу дослідження були покладені дані отримані відповідно до п.п. 2.2 - 2.5 п.2., п.п. 3.1 п.3, п.п. 4.1 - 4.2 п.4, п.п. 5.1 п.5, п.п. 6.1 - 6.4 п.6, п.п. 7.1 п.7 названого вище протокольного доручення (табл.7.2.1). Фактори, що були враховані джерело та відповідальні особи за їх надання наведені в таблиці 7.1.

Таблиця 7.1

Фактори, що були враховані в математичній моделі, джерела, з яких була отримана інформація та відповідальні особи за їх надання

<b>Фактори</b>	<b>Джерело інформації</b>	<b>Відповідальна особа</b>
Територіальний поділ Дніпропетровської області	Головне управління статистики у Дніпропетровській області	Лукашева С.В.
Гендерна та вікова структура населення	Головне управління статистики у Дніпропетровській області	Лукашева С.В.
Гендерна та вікова структура інфікованих	Дніпропетровський обласний лабораторний центр	Чуб Р.В.
Фактор відсутності карантинних обмежень	Дніпропетровська обласна державна адміністрація	Падалко Г.В.
Кількість підозр, інфікованих, померлих, одужавших по дням	Дніпропетровський обласний лабораторний центр	Чуб Р.В.
Статистичні данні про протікання епідемії COVID-19 у інших країнах	Відкриті джерела	Кузенков О.О.
Особливості протікання хвороби	Експертна думка	Шостакович-Корецька Л.Р., Степанський Д.О.

### 7.3 Математична постановка задачі

Популяція інфікованих осіб розглядається, як скінченна множина підгруп, кожна з яких характеризується власним рівнем сприйнятливості до інфекції та тривалістю перебігу захворювання. Ймовірність зараження для кожної підгрупи може бути описана специфічними параметрами.

Виходячи з цих припущень, компартментну модель динаміки інфекційних захворювань можна аналітично подати у такій формі:

$$\left\{ \begin{array}{l} \dot{S} = -\sum_{i=1}^p \beta_i \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j \\ \dot{I}_1 = \beta_1 \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j - \gamma_1 \cdot I_1 \\ \dot{I}_2 = \beta_2 \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j - \gamma_2 \cdot I_2 \\ \dots \\ \dot{I}_p = \beta_p \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j - \gamma_p \cdot I_p \\ \dot{R} = \sum_{i=1}^p \gamma_i \cdot I_i \end{array} \right. \quad (7.3)$$

У моделі (7.3) використовуються такі позначення:

$S$  — кількість осіб у субпопуляції, які не інфіковані (сприйнятливі);

$I_i$  — кількість інфікованих осіб в  $i$ -тій субпопуляції, яка характеризується окремою ймовірністю зараження та тривалістю перебігу хвороби;

$p$  — кількість субпопуляцій інфікованих осіб, кожна з яких має унікальні значення ймовірності інфікування та тривалості захворювання;

$N$  — загальна кількість осіб у мета популяції. При цьому вважається, що

$$N = S + \sum_{i=1}^p I_i + R;$$

$\beta_i$  — ймовірність зараження для осіб з  $i$ -тої субпопуляції;

$\gamma_i$  — ймовірність одужання для осіб з  $i$ -тої субпопуляції;

$R$  — кількість осіб, які вибули з популяції (тобто тих, що одужали або померли).

Слід підкреслити, що ключова особливість запропонованої моделі полягає у поділі інфікованих пацієнтів на окремі підгрупи, кожна з яких має суттєво різні характеристики перебігу захворювання. Зокрема, деякі особи можуть мати короткий інкубаційний період, що робить їх менш значущими для загальної динаміки системи. Водночас інші особи, через низку факторів, можуть залишатися заразними протягом тривалого часу, суттєво впливаючи не лише на кількісні показники динаміки інфекції, а й на швидкість її поширення у межах метапопуляції.

Описаний ефект має подібності з відомою моделлю поширення інфекційних захворювань у теорії клітинних автоматів. Хоча ця модель є дискретною, що накладає певні обмеження на вхідні дані, структуру моделі та результати симуляцій, вона демонструє аналогічний парадокс: зменшення ймовірності зараження може призвести до збільшення загальної тривалості хвороби в ураженій області або навіть до нездатності системи досягти повного одужання, що виявляється у хаотичній міграції інфекції по просторовому середовищу (див.рис.7.1).

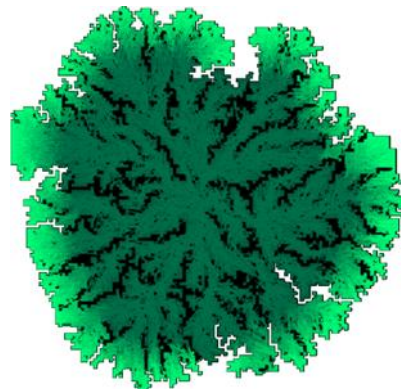


Рисунок 7.1. Візуалізація процесу поширення інфекційного захворювання на дискретній сітці розміром  $[100;100]$  з кроком 0,5 та ймовірністю зараження  $\beta = 0,5$  математичної моделі клітинних автоматів.

З прикладної точки зору, цей ефект можна пояснити тим, що низька ймовірність зараження, хоча й уповільнює прогресування хвороби в межах

області, перешкоджає формуванню імунного бар'єру між інфікованими та здоровими особами.

На рисунку 7.1 ми бачимо результат імітації розвитку інфекційного захворювання, отримане із застосуванням мультиагентного клітинного автомату. Такий підхід дозволяє реалізувати, у тому числі модель SIR. На рисунку білим кольором позначено здорові особини, чорним кольором – вибувші, а градієнтом зеленого інфіковані різних стадій протікання хвороби. Визначальними в такій моделі є ймовірність інфікування сусідніх клітин, та окіл можливого інфікування сусідів (окіл Мура або Неймана). Такий підхід враховує просторовий розподіл та має два цікавих ефекти, виникнення яких не завжди є очевидним на перший погляд. По-перше, при зменшенні ймовірності інфікування сусідів ймовірність загального оздоровлення системи зменшується, а не навпаки, що було б логічно. Причину такого ефекту очевидні якщо розглянути модель й ймовірністю інфікування сусідів рівними 100%. У такому випадку система з плином часу матиме форму квадрату (розмір якого збільшуватиметься) в центрі якого з плином часу утворюватиметься “пояс” вибувших (також у формі квадрату), які з причини наявності в них імунітету не дозволять інфекції повторно потрапити до центру системи. При зменшенні ймовірності інфікування сусідів в системі відбувається сценарій зображений на Рис.7.1. При ймовірності інфікування сусідів  $>0,6$  система зберігає високу ймовірність загального одужання. При ймовірності інфікування  $<0,4$  в системі відбуваються хаотичне інфікування, малоімовірно утворення “поясу” вибувших і така система знаходиться у постійній динаміці.

Схожий ефект пов'язаний з типом околу, що використовується при визначенні інфікованих сусідів. Використання околу Мура збільшує ймовірність інфікування вдвічі, а використання околу Неймана, відповідно, зменшує. Таким чином при використанні околу Неймана на границях відбувається затримка на один крок, та ймовірність хаотичного інфікування

збільшується вдвічі. Штатна робота системи, у випадку використання околу Неймана, реалізується лише при імовірності інфікування  $>80\%$ .

Узагальнюючи можемо підкреслити, що використання клітинних автоматів, для реалізації SIR моделей є достатньо зручним, а можливість їх візуалізації дозволяє наочно оцінити результати динаміки хвороби, проте рішення є нестійким до значень ключових параметрів системи, містить стохастичну складову та не є застосовним до реальних ареалів, що містять дороги, пасажиропотоки та інші індивідуальні особливості системи.

Причини, через які інфіковані підпопуляції можуть мати різні коефіцієнти ймовірності зараження, можна пояснити такими факторами:

- рівень імунітету окремих осіб;
- територіальне розташування осіб або їхніх груп на певній ділянці;
- обмежувальні заходи, запроваджені в регіонах або на окремих територіях;
- індивідуальне застосування протиепідемічних заходів (наприклад, носіння масок, дезінфекція, уникнення скупчень людей тощо).

Ключова особливість запропонованої моделі полягає в її застосовності до просторово розподілених задач. У класичних моделях SIS, SIR і SEIR населення вважається зосередженим у одній точці, а всі процеси моделюються без урахування просторового розподілу. Однак більшість протиепідемічних заходів, запроваджених у різних країнах, були безпосередньо пов'язані з відокремленням інфікованих особин від здорових (наприклад, закриття парків, магазинів та інших місць масового скупчення людей) та, що найголовніше, продемонстрували свою високу ефективність. Тому врахування просторового розподілу у математичному моделюванні, хоча й значно ускладнює чисельні розрахунки та аналітичні дослідження, дає більш точні результати і краще адаптовані до конкретних регіонів.

У цьому розділі дисертаційної роботи здійснено моделювання поширення інфекційних захворювань у популяціях, які розподілені гетерогенно по географічно зв'язаних регіонах з нерегулярними межами,

таких як міські та сільські території. Просторова метрика є неевклідовою і визначається такими факторами, як центри густини населення, транспортні мережі та їх пропускна спроможність, дорожня інфраструктура та інтенсивність руху. У якості тестових-досліджень проаналізовано два різні регіони України, а саме Дніпропетровська та Харківська області, з населенням понад 3 мільйони в кожному, де гетерогенний розподіл населення та концентрація в міських осередках прискорюють локальне поширення інфекції, а транспортні потоки сприяють швидкому розповсюдженню між регіонами.

Щоб подолати недоліки компартментальних моделей, ми використовуємо просторово-явну компартментальну модель, представлену в загальному вигляді рівнянням (7.3). У цій моделі регіон  $\Omega$  поділено на  $p$  підрегіонів  $\Omega_i, i=1,2,\dots,p$ , де  $I_i$  позначає кількість інфікованих осіб у підрегіоні. Загальна популяція  $N$  задовольняє умову замкненості  $N = S + \sum_{i=1}^p I_i + R$ , де  $S$  - кількість сприйнятливих до інфекції осіб, а  $R$  - кількість вилікуваних або померлих (видалених із популяції). Ця модель враховує просторову гетерогенність, зокрема відмінності між підрегіонами у рівнях інфекції та динаміці населення.

Одна з модифікацій моделі (7.3), що має більш спрощену структуру та дозволить суттєвим чином спростити постанову чисельного експерименту представлена у вигляді (7.4).

$$\begin{cases} \dot{V}_i = -\alpha_i \cdot V_i + (1 - \beta_i) \cdot (I_i + C_i) + \sum_{j=1, j \neq i}^n (1 - \alpha_j - G_{ij}) \cdot \varphi_{ij} \\ \dot{I}_i = \alpha_i \cdot V_i - \beta_i \cdot I_i + \gamma_i \cdot c_i + \sum_{j=1, j \neq i}^n (\alpha_j + G_{ij}) \cdot \varphi_{ij} \\ \dot{P}_i = \beta_i (I_i + C_i) \end{cases} \quad (7.4)$$

де

$i = \overline{1..n}$  ;

$V_i$  - кількість здорових мешканців  $i$ -го територіального сектору;

$I_i$  - кількість інфікованих пацієнтів  $i$ -го територіального сектору;

$P_i$  - кількість померлих від COVID-19 в  $i$ -му територіальному секторі;

$N_i$  - загальна кількість інфікованих за весь період дослідження в  $i$ -му територіальному секторі;

$n_i$  - загальна кількість мешканців в  $i$ -му територіальному секторі;

$D$  - загальна кількість тижнів дослідження;

$d_i$  - загальна кількість померлих від COVID-19 в  $i$ -му територіальному секторі;

$c_i$  - загальна кількість персоналу (медичного та допоміжного), задіяного в обслуговуванні інфікованих пацієнтів в  $i$ -му територіальному секторі;

$C_i$  - кількість інфікованого персоналу (медичного та допоміжного), який був задіяний в обслуговуванні інфікованих пацієнтів в  $i$ -му територіальному секторі;

$\varphi_{ij}$  - пасажиропотік, що прибуває в сектор  $i$  за прийнятий часовий інтервал (тиждень) з сектора  $j$ ;

$m_{ij}$  - середня кількість рейсів із сектора  $j$  у сектор  $i$  за прийнятий часовий інтервал (тиждень);

$s_{ij}$  - середня площа транспортних засобів, що застосовуються для здійснення перевезень із сектора  $j$  у сектор  $i$ .

$n$  - кількість секторів, на які була поділена досліджувана область.

Допоміжні параметри  $\alpha_i, \beta_i, \gamma_i, G_{ij}, S_{ij}$  пропонується розраховувати наступними наступним чином:

$$\alpha_i = \frac{N_i}{n_i \cdot D}; \beta_i = \frac{d_i}{n_i \cdot D}; \gamma_i = \frac{C_i}{c_i \cdot D}; G_{ij} = \frac{1}{S_{ij} + \frac{1}{1 - a_j}}; S_{ij} = \frac{m_{ij} \cdot s_{ij}}{\varphi_{ij}}. \quad (7.5)$$

Узагальнений часовий інтервал для моделювання пропонується прийняти рівним одному тижню. Ітераційні розрахунки здійснюватимуться відповідно до кожного дня для зниження похибки, яка характерна для моделювання по тижням. Особливості організації роботи лабораторних

центрів Дніпропетровської та Харківської областей, а саме вихідний день у неділю, призводить до нерівномірного розподілу кількості проведених ПЛР тестів та підтверджених випадків по дням. Виходячи з вищевикладеного, для оцінки похибки запропонованої моделі обрано відповідність прогнозованої та реальної кількості підтверджених випадків за один (календарний) тиждень.

Виходячи з отриманих вихідних даних параметри математичної моделі (7.4) набули таких значень:

$$V_1 - 3\,176\,404; I_1 - 244; P_1 - 7; N_1 - 445; n_1 - 3\,176\,648; D - 10; d_1 - 7; \quad (7.6)$$

У зв'язку з повною відсутністю на момент дослідження достовірної інформації щодо параметрів  $c_1, C_1, \varphi_{ij}, m_{ij}, s_{ij}$  їх значення були прийняті рівними 0. Використовуючи (7.5), розраховані початкові значення інших параметрів моделі:

$$\alpha_1 = 0,000015; \beta_1 = 0,00000022; \gamma_1 = 0; G_{ij} = 0; S_{ij} = 0. \quad (7.7)$$

Крім перелічених у таблиці 7.1 джерелах інформації інформація про поширення COVID-19 у Харківській області здійснювалось на основі даних з відкритих джерел, зокрема:

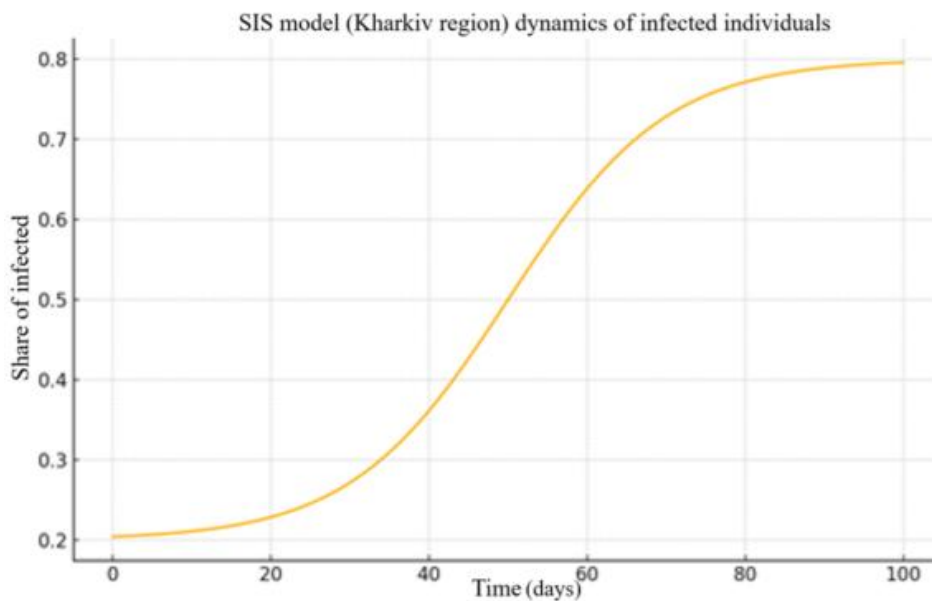
1. Центр громадського здоров'я України (<https://phc.org.ua/>).
2. Державна служба статистики України (<https://ukrstat.gov.ua/>).
3. Ресурсний центр COVID-2019 (<https://coronavirus.jhu.edu/map.html>).
4. ВООЗ (<https://www.who.int/emergencies/diseases/novel-coronavirus-2019>).
5. Our World in Data (<https://ourworldindata.org/coronavirus>).

#### 7.4 Результати чисельного експерименту

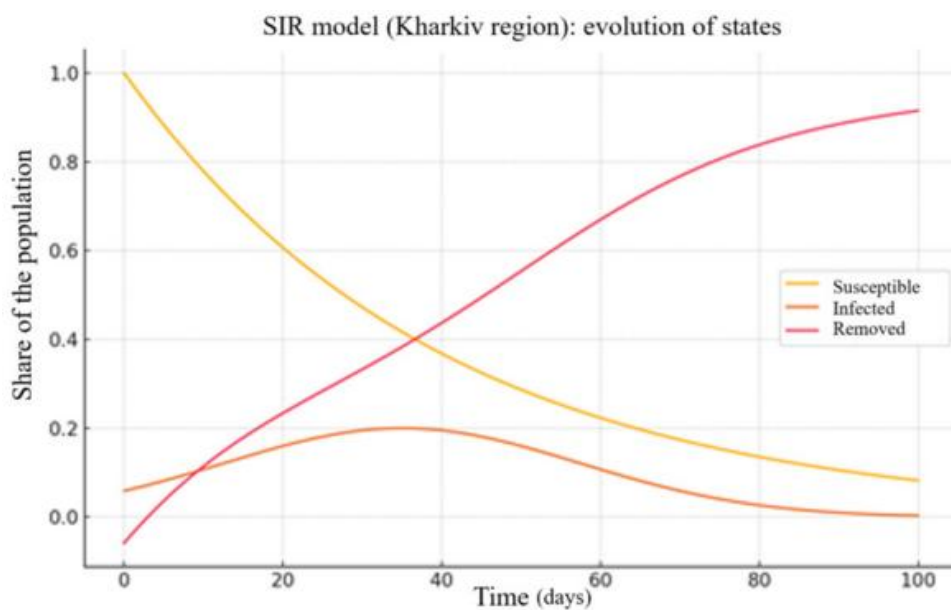
##### Математичне моделювання розповсюдження COVID-2019 на території Харківської області

На рисунку 7.2 наведено результати чисельного моделювання для Харківської області за період 2020–2024 років. Рисунок 7.2-а ілюструє результати моделі SIS, а рисунок 7.2-б — моделі SIR, подані у відносних

одиницях, де вісь  $Y$  показує частку інфікованого населення відносно загальної чисельності населення області (2,658 мільйона осіб).



а)



б)

Рисунок 7.2. Результати чисельної реалізації моделей: а) SIS, б) SIR для Харківської області за період 2020-2024 років.

Рисунок 7.3 ілюструє результати моделі SEIR за той самий період.

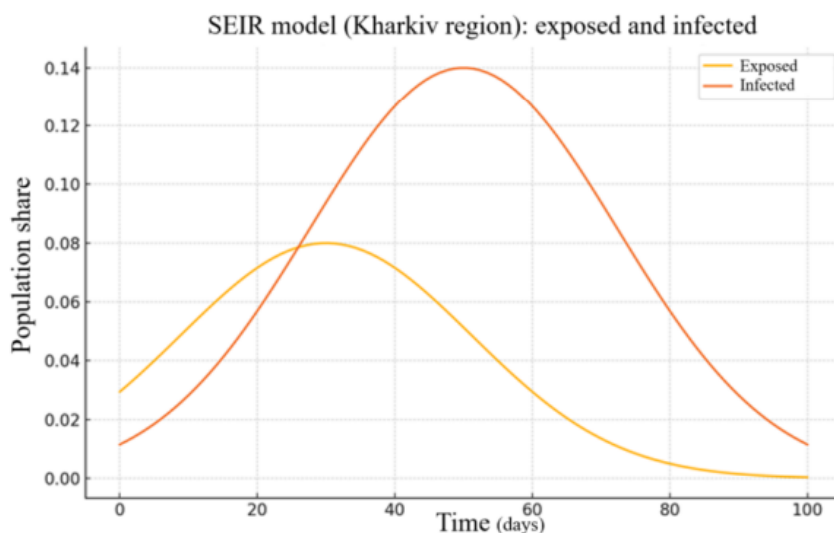


Рисунок 7.3. Результати чисельної реалізації моделі SEIR для Харківської області за період 2020-2024 років.

У таблиці 7.2. наведено результати аналізу похибки використаних моделей для Харківської області.

Таблиця 7.2.

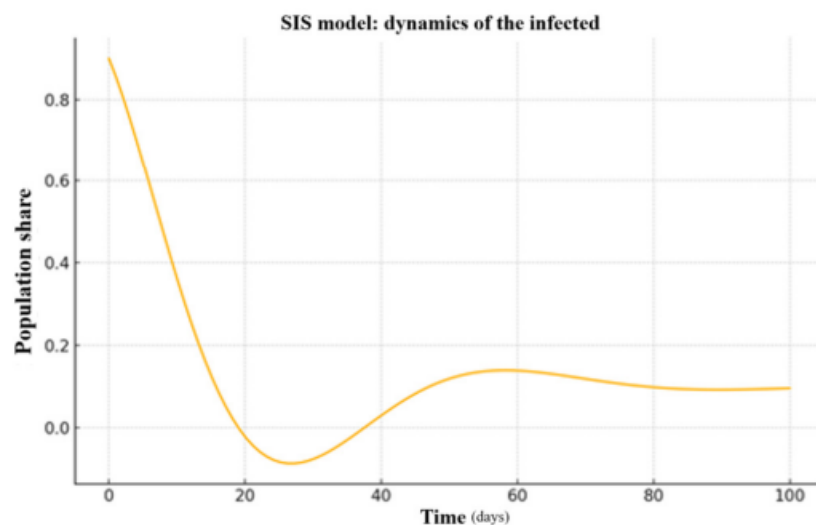
Відносна похибка результатів моделювання динаміки поширення COVID-19 у Харківській області за період 2020–2024 років для моделей SIS, SIR і SEIR.

Період	SIS	SIR	SEIR	Період	SIS	SIR	SEIR
04.2020-06.2020	6,75%	4,99%	3,65%	10.2022-12.2022	7,77%	6,49%	3,62%
07.2020-09.2020	3,71%	4,39%	4,74%	01.2023-03.2023	7,59%	2,02%	4,21%
10.2020-12.2020	2,92%	2,75%	2,64%	04.2023-06.2023	3,17%	7,29%	4,46%
01.2021-03.2021	6,35%	6,76%	3,62%	07.2023-09.2023	5,04%	6,64%	4,81%
04.2021-06.2021	3,62%	4,03%	4,50%	10.2023-12.2023	7,47%	2,42%	3,84%
07.2021-09.2021	5,21%	6,23%	3,11%	01.2024-03.2024	6,95%	7,12%	2,01%
10.2021-12.2021	6,74%	2,28%	3,32%	04.2024-06.2024	4,85%	4,44%	2,06%
02.2022-03.2022	5,14%	2,57%	4,08%	07.2024-09.2024	6,05%	7,50%	4,07%
04.2022-06.2022	2,74%	4,11%	3,55%	10.2024-12.2024	2,26%	2,32%	2,37%
07.2022-09.2022	4,85%	4,40%	2,31%				
Максимальна відносна похибка	6,75%	6,76%	4,74%	Максимальна відносна похибка	7,77%	7,50%	4,81%

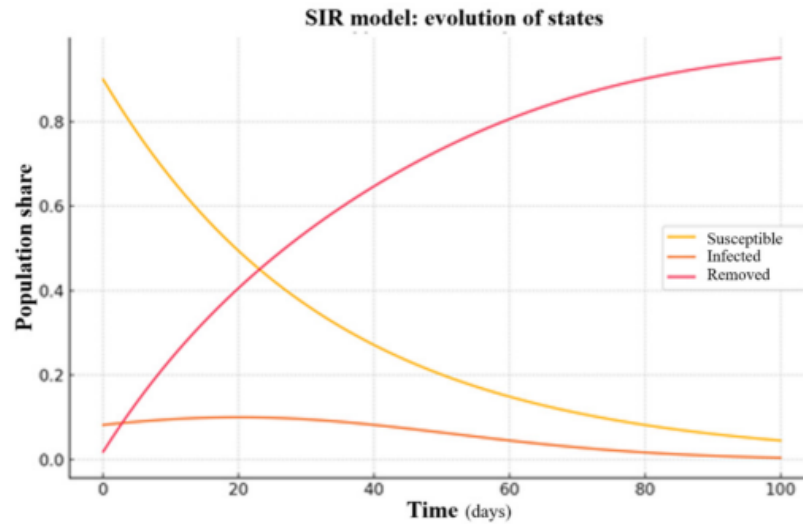
Результати представлені на рис.7.2-7.3 отримані в результаті постановки чисельного експерименту застосованого для розв’язання запропонованої в роботі системи (7.4) для двох різних регіонів. Для чисельного розв’язання був застосований метод Рунге-Кути 4-го порядку. У якості початкових точок були застосовані статистичні данні отримані в результаті аналізу відкритих джерел та джерел наведених в таблиці 7.1. Математична модель є неперервною, проте отримані на кожній ітерації точки не підлягали округленню, що було зроблено виходячи з великого обсягу вибірки для уникнення накопичування похибки. Такий підхід застосований з метою отримання більш точних результатів та застосовувався лише для репрезентативних значень кількості інфікованих більше 3000 (0,1% від чисельності популяції).

### **Математичне моделювання розповсюдження COVID-2019 на території Дніпропетровської області**

На рисунку 7.4 наведено результати чисельного експерименту результатів моделювання розповсюдження COVID-2019 для Дніпропетровської області України за період 2020–2024 років. Рисунок 7.4-а демонструє динаміку поширення COVID-19 відповідно до моделі SIS, а рисунок 7.4-б - відповідні результати для моделі SIR.



a)



б)

Рисунок 7.4. Результати чисельної реалізації моделей: а) SIS, б) SIR для Дніпропетровської області за період 2020–2024 років.

Результати, представлені на рисунку 7.4, свідчать про те, що обидві моделі адекватно відтворюють динаміку захворюваності в абсолютних величинах. Вісь ординат відображає частку інфікованого населення відносно загальної чисельності населення регіону (3,1 мільйона осіб). Максимальна відносна похибка для всіх моделей не перевищує 8%, що вважається прийнятним рівнем для епідеміологічного прогнозування. Відтак запропонована модель була використана для прогнозування кількості інфікованих осіб і оцінки потреб у ліжко-місцях у регіоні.

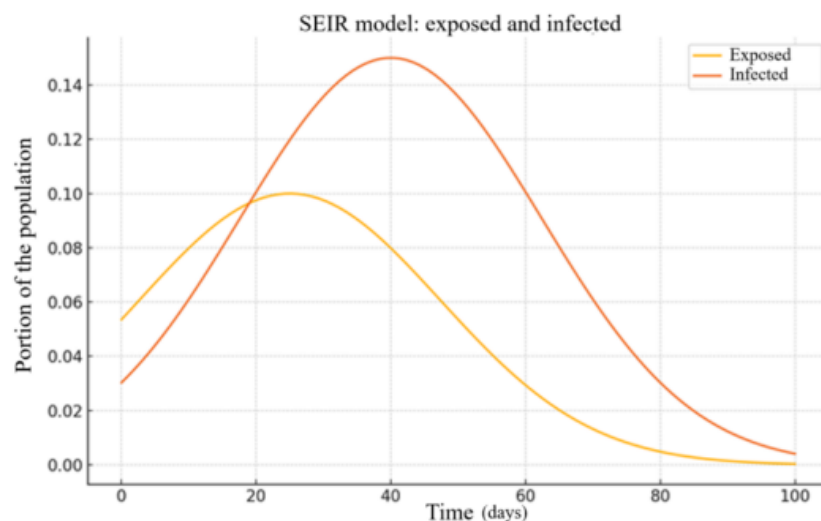


Рисунок 7.5. Результати чисельної реалізації моделі SEIR для Дніпропетровської області за період 2020–2024 років.

Рисунок 7.5 ілюструє результати застосування моделі SEIR на тому ж наборі даних, що і для чисельного експерименту наведеному на рис. 7.4.

Як видно з рисунку 7.5, модель SEIR дає більш детальне представлення процесу розповсюдження інфекції ніж SIS та SIR моделі. Втім, її довгострокова точність обмежена через труднощі ідентифікації параметрів, що не повністю враховують такі фактори, як нереєстровані випадки (наприклад, особи, які лікуються вдома без ПЛР тестування). Це сприяє збільшенню рівня непередбачуваності захворювання, що у свою чергу впливає на коректність та ефективність обмежувальних заходів та як наслідок інших негативних наслідків. Незважаючи на ці обмеження, модель SEIR точно прогнозує кількість позитивних ПЛР тестів і необхідну кількість ліжко-місць, з максимальною відносною похибкою 5,6%, що перевищує точність моделей SIS та SIR похибка якої складає, як було показано вище близько 8%.

У таблиці 7.3 наведено результати аналізу похибки використаних моделей для Дніпропетровської області.

Таблиця 7.3.

Відносна похибка результатів застосування SIS, SIR та SEIR моделей на прикладі Дніпропетровської області (2020–2024 роки)

Період	SIS	SIR	SEIR	Період	SIS	SIR	SEIR
04.2020-06.2020	6,54%	2,88%	4,69%	10.2022-12.2022	6,93%	6,42%	2,6%
07.2020-09.2020	6,92%	3,60%	3,79%	01.2023-03.2023	6,15%	7,89%	3,51%
10.2020-12.2020	4,16%	7,34%	2,02%	04.2023-06.2023	2,16%	4,13%	4,2%
01.2021-03.2021	3,57%	7,49%	3,61%	07.2023-09.2023	5,87%	5,09%	3,27%
04.2021-06.2021	7,64%	6,23%	3,51%	10.2023-12.2023	4,59%	5,66%	2,94%
07.2021-09.2021	2,71%	4,74%	3,74%	01.2024-03.2024	5,32%	6%	5,6%
10.2021-12.2021	4,94%	2,54%	3,94%	04.2024-06.2024	6,22%	2,92%	2,06%
02.2022-03.2022	2,67%	7,07%	2,89%	07.2024-09.2024	2,81%	3,81%	4,68%
04.2022-06.2022	4,38%	2,89%	3,43%	10.2024-12.2024	5,78%	5,86%	4,3%
07.2022-09.2022	4,82%	5,36%	3,22%				
Максимальна відносна похибка	7,64%	7,49%	4,69%	Максимальна відносна похибка	6,93%	7,89%	5,6%

## Математичне моделювання розповсюдження COVID-2019 на території Дніпропетровської області з використанням спароженої компартментальної моделі SIR

З використанням математичної моделі (7.4) параметри моделі були обчислені за формулою (7.5) та набули значень (7.6)-(7.7). Належним чином вхідні дані було отримано лише станом на 20.07.2020 року. 21.07.2020 року було проведено перший чисельний експеримент. Головою надзвичайної протиепідемічної комісії при Дніпропетровській обласній державній адміністрації та Дніпропетровській обласній раді було визначено необхідність побудови математичної моделі на період 90 днів. Такий термін був обраний для забезпечення необхідного часового інтервалу вчинення необхідних організаційних заходів з розширення кількості необхідних лікарняних місць та медичного обладнання для забезпечення ними хворих пацієнтів на COVID-19. Було розроблено реалістичний (РП), оптимістичний (ОП та песимістичний (ПП) прогнози. Результати числового експерименту, проведеного 21.07.2020 року, на період 28 днів (4 тижні) представлено у таблиці 7.4.

Таблиця 7.4

Результати чисельного експерименту для моделі (7.4)-(7.7) на період з 22.07.2020р. по 18.08.2020р. (Дніпропетровська область)

Дата	РП	ОП	ПП	Дата	РП	ОП	ПП
22.07.2020	13	10	22	05.08.2020	18	15	31
23.07.2020	14	11	23	06.08.2020	19	15	32
24.07.2020	15	12	25	07.08.2020	19	15	33
25.07.2020	15	12	25	08.08.2020	20	16	34
26.07.2020	15	12	25	09.08.2020	20	16	35
27.07.2020	15	12	26	10.08.2020	20	16	35
28.07.2020	17	14	29	11.08.2020	20	16	34
29.07.2020	16	13	28	12.08.2020	20	16	35
30.07.2020	19	15	33	13.08.2020	21	17	35
31.07.2020	21	16	35	14.08.2020	21	17	35
01.08.2020	17	14	30	15.08.2020	21	17	36
02.08.2020	18	14	30	16.08.2020	22	17	37
03.08.2020	18	14	30	17.08.2020	22	18	37
04.08.2020	18	14	31	18.08.2020	22	18	38

За результатами числового розрахунку було побудовано шість прогнозів (від 22.07, 29.07, 05.08, 12.08, 19.08 та 06.10). Кожен з прогнозів був побудований на термін 90 діб. У таблиці 7.5 наведено інформацію про похибку моделювання кожного з шести прогнозів.

Таблиця 7.5

Аналіз похибки шести розроблених прогнозів динаміки COVID-19 у  
Дніпропетровській області

№ з/п	Дата побудови прогнозу	Термін моделювання	Похибка
1	22 липня 2020р.	1 тиждень	0,0%
2	29 липня 2020р.	1 тиждень	5,(45)%
3	05 серпня 2020р.	1 тиждень	8,6%
4	12 серпня 2020р.	1 тиждень	7,0%
5	19 серпня 2020р.	1 місяць (вересень)	3,8%
6	06 листопада 2020р.	1 місяць (жовтень)	3,1%

### 7.5 Висновки до розділу 7

У сьомому розділі дисертаційної роботи запропоновано та детально описано дві нові компартментальні математичні моделі для прогнозування поширення COVID-19 та інших подібних інфекційних захворювань. Проведено змістовний аналіз літературних джерел з тематики математичного моделювання інфекційних захворювань, висвітлені переваги та недоліки існуючих підходів.

Використання у якості інструменту моделювання систем автономних диференціальних рівнянь дозволило побудувати динамічну модель, яка відтворює часові зміни чисельності різних груп населення, що характеризуються специфічним станом здоров'я: сприйнятливі, інфіковані, одужуючі та інші класи, що включаються в моделі залежно від її конкретної конфігурації. Ідентифікацію параметрів запропонованих в розділі моделей здійснено таким чином, що їх можна оперативно адаптувати на основі даних із офіційних джерел, що забезпечує гнучкість та практичну застосовність моделей у реальному часі для різних регіональних та просторових сценаріїв.

Адекватність і достовірність моделей підтверджено шляхом їхнього тестування на прикладі поширення COVID-19 у двох найбільших регіонах

України — Дніпропетровській та Харківській областях, що відзначаються значною чисельністю населення та великою площею території. Проведений чисельний експеримент дозволив створити шість прогнозів поширення інфекції на 90 днів для кожного регіону. Кожен з прогнозів продемонстрував високий рівень точності, при цьому середня похибка становила не більше 8,6%, що підтверджує високу надійність моделі та її практичну цінність для прогнозування епідемічних процесів на регіональному рівні.

Результати чисельного моделювання дозволили сформулювати обґрунтовані управлінські рекомендації для регіональних органів охорони здоров'я щодо оптимального впровадження карантинних заходів та регулювання мобільності населення. Важливим аспектом є те, що запропоновані моделі дозволяють не лише прогнозувати поточну епідемічну ситуацію, а й оцінювати потенційний вплив різних сценаріїв втручання на подальший розвиток епідемії, що сприяє прийняттю ефективних рішень на основі кількісного аналізу.

Отримані результати сприяють поглибленню розуміння механізмів динаміки поширення COVID-19 та подібних інфекційних захворювань. Зокрема, вони дозволяють більш точно оцінити вплив окремих параметрів системи на загальний перебіг епідемії, включаючи такі фактори, як швидкість поширення вірусу, час інкубаційного періоду та ефективність заходів контролю. Це, у свою чергу, забезпечує наукову основу для підтримки процесів прийняття рішень у сфері громадського здоров'я та розробки стратегій мінімізації ризиків, пов'язаних з інфекційними захворюваннями.

Практична новизна наведених в цьому розділі результатів полягає у можливості використання запропонованих підходів та математичних моделей для прогнозування динаміки поширення інфекційних хвороб у реальних умовах. Ретроспективний аналіз показав високу точність моделей та їхню адекватність для формулювання управлінських рекомендацій. Це дозволяє не лише отримати актуальний прогноз стану епідемічної системи, а

й здійснювати науково обґрунтоване управління процесами її функціонування з метою зміни якісних режимів епідемічного розвитку.

Отже, розроблені моделі представляють собою ефективний інструмент для прогнозування та управління поширенням інфекційних захворювань на регіональному рівні, поєднуючи гнучкість налаштування параметрів, високий рівень точності прогнозів та можливість практичного використання для прийняття рішень у сфері охорони здоров'я. Вони можуть бути інтегровані в системи моніторингу та прогнозування епідемій, слугуючи надійною основою для науково обґрунтованого планування карантинних заходів та оцінки ефективності різних сценаріїв втручання.

## ВИСНОВКИ

У дисертаційній роботі проведено системне дослідження задач оптимального розбиття множин в динамічних постановках, що включає формалізацію математичних моделей, розробку методів алгоритмів розв'язання, аналіз впливу нечітких параметрів та прикладну реалізацію у різних сферах. Таким чином всі поставлені в роботі завдання виконано у повному обсязі. Основні результати роботи можна узагальнити за розділами.

В першому розділі дисертаційної роботи здійснено критичний огляд сучасного стану досліджень у сфері оптимального розбиття множин, кластеризації та динамічного просторового планування. Проведено аналіз теоретичних підходів до побудови математичних моделей кластеризації, варіаційних і динамічних задач, а також розглянуто методи чисельного розв'язання, включно з  $r$ -алгоритмами та їх модифікаціями. Особлива увага приділена проблемам нечітких даних та обмежень на розміщення центрів підмножин. Окремо проаналізовано прикладні аспекти використання таких моделей у логістиці, обороні, промисловому виробництві та охороні здоров'я, що дозволило сформулювати основні вимоги до подальшого дослідження. Розділ окреслює ключові теоретичні проблеми, що потребують вирішення, визначає пріоритетні напрями розвитку методології та обґрунтовує вибір базових математичних інструментів для побудови динамічних моделей у наступних розділах дисертації.

У другому розділі присвячено формалізації класичної неперервної однопродуктової динамічної задачі оптимального розбиття множин із фіксованими координатами центрів підмножин. Розроблено математичну модель, в якій зміна транспортних витрат у часі описується системою диференціальних рівнянь. Запропоновано чотири типи регуляторних сценаріїв вартості перевезень — інфляційно-дефляційна модель, одностороння та двостороння цінова регуляція, дотаційна модель. Обрана як базова інфляційно-дефляційна модель дозволила зберегти аналітичну прозорість та водночас відтворити ключові властивості часової варіативності

витрат. Розроблений алгоритм розв'язання задачі включає побудову системи диференціальних рівнянь, інтегрування з початкових умов, обчислення функціоналу повної вартості перевезень та формування функцій приналежності точок до центрів на основі принципу мінімізації транспортних витрат. Чисельне моделювання для різних конфігурацій центрів та межових умов показало стійкість моделі до змін параметрів задачі та підтвердило її адекватність з прикладної точки зору.

У третьому розділі розглянуто динамічну задачу оптимального розбиття множин з відшукуванням оптимальних координат центрів підмножин. Розроблено варіаційну модель, яка дозволяє знаходити оптимальні координати центрів у процесі еволюції системи. Для розв'язання застосовано модифікований  $r$ -алгоритм, що забезпечує ефективне чисельне рішення для різної кількості центрів, стартових координат та параметрів моделі. Порівняльний аналіз показав суттєві відмінності функціоналу між моделями з фіксованими центрами та оптимально визначеними, що підтверджує практичну цінність адаптивного підходу. Окрему увагу приділено узгодженому вибору параметрів для чисельних експериментів, що забезпечило коректне порівняння результатів та їх змістовну інтерпретацію.

У четвертому розділі сформульовано та розв'язано динамічну задачу оптимального розбиття множин з фіксованими центрами в нечіткій постановці. Запропоновано методику адаптивної ідентифікації нечітких параметрів, яка включає початкове фузифікування, побудову функцій належності, оптимізацію за критерієм мінімізації функціоналу та дефузифікацію результатів. Чисельні експерименти для моделі квадрату зі стороною одиничної довжини показали, що після адаптивного налаштування нечітких параметрів досягається висока відповідність результатів нечіткої та чіткої моделей за значеннями функціоналу та геометричними характеристиками кластерів. Досліджено біфуркаційні ефекти, визначено аналітичні умови виникнення критичних змін у поведінці системи, що дозволяє прогнозувати стабільність розбиття та оцінювати сценарії розвитку.

У п'ятому розділі здійснена прикладна реалізація алгоритму динамічного розбиття для системи альтернативного оповіщення «Світло». Розроблено модель оптимального розбиття території зони відповідальності одного з командувань протиповітряної оборони для 51 вогневої групи з урахуванням динаміки загроз та топографії. Інтеграція геоінформаційних даних із динамічними алгоритмами дозволяє майже в реальному часі перераховувати оптимальні зони відповідальності та оперативно переорієнтовувати вогневі групи ППО. Проведено порівняльний аналіз із міжнародними системами OnSolve MIR3 та AlertHub Pro, який показав переваги «Світло» в умовах специфіки бойових дій на території України.

У шостому розділі успішно реалізовано застосування алгоритмів динамічного розбиття для побудови оптимальної траєкторії 3D-друку. Поєднання глобальних методів динамічного розбиття з локальною оптимізацією траєкторії на основі математичного моделювання типу «клітинні автомати» дозволило мінімізувати переривання, уповільнення та релокації друкарської головки, підвищуючи швидкість і якість виготовлення деталей. Реалізовано програмний додаток із гнучким налаштуванням параметрів, підтримкою форматів STL та g-code, що дозволяє автоматизувати процес побудови траєкторії навіть для складних об'єктів.

У сьомому розділі дисертаційної роботи розроблено дві компартментальні моделі для прогнозування поширення COVID-19 та подібних інфекційних захворювань. Використання автономних систем диференціальних рівнянь дозволило моделювати часову змінність чисельності різних класів населення. Ідентифікація параметрів здійснюється на основі офіційних даних, що забезпечує оперативну адаптацію моделей до регіональних сценаріїв. Чисельне тестування для Дніпропетровської та Харківської областей показало високу точність прогнозів із середньою похибкою не більше 8,6%. Результати надали підстави для формулювання управлінських рекомендацій щодо карантинних заходів та регулювання мобільності населення.

Загалом дисертаційна робота демонструє ефективність комплексного підходу до розв'язання задач оптимального розбиття множин в динамічних постановках.. Запропоновані математичні моделі, алгоритми та методи забезпечують: аналітичну прозорість, чисельну ефективність, адаптацію до нечітких даних, можливість оперативного використання в прикладних задачах. Практична значимість підтверджується застосуванням у військовій логістиці, системах оповіщення, 3D-друку та при прогнозуванні епідемій. Запропоновані методи поєднують універсальність, точність прогнозів та ефективність управління динамічними системами.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бейко І. В., Киселева Є. М. Умова оптимальності межі розподілу зон територіального обслуговування // Застосування математичних методів до розв'язання виробничо-економічних задач. – Дніпропетровськ : ДГУ, 1973. – С. 53–55.
2. Бублик Б. Н., Кириченко Н. Ф. Основи теорії управління. – Київ : Вища школа, 1975. – 327 с.
3. Грицик В. В., Шевченко А. І., Стецюк П. І. та ін. Математичні методи оптимізації та інтелектуальні комп'ютерні технології моделювання складних процесів і систем з урахуванням просторових форм об'єктів. – Донецьк : ІП «Наука і освіта», 2011. – 650 с.
4. Жалдак М. І., Триус Ю. В. Основи теорії і методів оптимізації : навч. посіб. – Черкаси : Брама-Україна, 2005. – 608 с.
5. Капустян В. Є., Киселева Є. М., Кроха Л. С. Розв'язання деяких задач стартового управління методом оптимального розбиття множин // Міжнародний наук.-техн. журн. «Проблеми управління та інформатики». – 1995. – № 5. – С. 80–88.
6. Киселева Є. М. Алгоритм розв'язання задачі оптимального розбиття з обмеженнями // Кібернетика. – 1983. – № 1. – С. 115–120.
7. Киселева Є. М. Математичні методи оптимального розбиття множин та їх застосування. – Дніпропетровськ : ДГУ, 1982. – 108 с.
8. Киселева Є. М. Становлення і розвиток теорії оптимального розбиття множин  $n$ -мірного евклідового простору. Теоретичні та практичні застосування // Проблеми управління та інформатики. – 2018. – № 5. – С. 114–135.
9. Киселева Є. М., Гарт Л. Л., Притоманова О. М. Алгоритм побудови діаграм Вороного з оптимальним розміщенням точок-генераторів на основі теорії оптимального розбиття множин // Проблеми управління та інформатики. – 2020. – № 2. – С. 5–15.
10. Киселева Є. М., Жильцова А. А. Неперервна задача оптимального

нечіткого розбиття множин без обмежень із заданим положенням центрів підмножин // Питання прикладної математики та математичного моделювання. – Дніпропетровськ : ДНУ, 2005. – С. 121–136.

11. Киселева Є. М., Коряшкіна Л. С. Моделі та методи розв'язання неперервних задач оптимального розбиття множин : лінійні, нелінійні, динамічні задачі : монографія. – Київ : Наукова думка, 2013. – 606 с.

12. Киселева Є. М., Коряшкіна Л. С. Неперервні задачі оптимального розбиття множин і  $r$ -алгоритми : монографія. – Київ : Наукова думка, 2015. – 400 с.

13. Киселева Є. М., Коряшкіна Л. С., Ус С. А. Теорія оптимального розбиття множин у задачах розпізнавання образів, аналізу та ідентифікації систем. – Дніпро : НГУ, 2015. – 270 с.

14. Киселева Є. М., Притоманова О. М. До розв'язання неперервних задач оптимального розбиття множин із нечіткими параметрами // Матеріали 5-ї Міжнар. конф. «Математичне моделювання, оптимізація та інформаційні технології». – Т. II. – Кишинів, 2016. – С. 188–193.

15. Киселева Є. М., Притоманова О. М. Розв'язання двоетапної неперервно-дискретної задачі оптимального розбиття-розподілу з заданим положенням центрів підмножин // Кібернетика і системний аналіз. – 2020. – № 1. – С. 3–15.

16. Киселева Є. М., Притоманова О. М., Шаравара В. В., Журавель С. В. Об'єктно-орієнтований підхід до програмної реалізації алгоритму розв'язання нелінійних задач оптимального розбиття множин // Питання прикладної математики і математичного моделювання. – Дніпро : Ліра, ДНУ, 2017. – Вип. 17. – С. 87–95.

17. Киселева Є. М., Степанчук Т. Ф. Пошук глобального мінімуму недиференційовної функції методом оптимального розбиття множин // Проблеми управління та інформатики. – 2002. – № 2. – С. 45–60.

18. Киселева Є. М., Строева В. А. Алгоритм розв'язання нелінійної неперервної багатопродуктової задачі оптимального розбиття множин із

розміщенням центрів підмножин // Проблеми управління та інформатики. – 2012. – № 1. – С. 40–53.

19. Киселева Є. М., Ус С. А., Станіна О. Д. Про задачі оптимального розбиття множин із додатковими зв'язками // Питання прикладної математики і математичного моделювання. – Дніпро : ДНУ, 2016. – С. 67–78.

20. Киселева Є. М., Шор Н. З. Дослідження алгоритмів розв'язання одного класу неперервних задач розбиття // Кібернетика і системний аналіз. – 1994. – № 1. – С. 84–96.

21. Киселева Є. М., Шор Н. З. Неперервні задачі оптимального розбиття множин : теорія, алгоритми, застосування : монографія. – Київ : Наукова думка, 2005. – 564 с.

22. Кісельова О. М. Становлення та розвиток теорії оптимального розбиття множин. Теоретичні і практичні застосування : монографія. – Дніпро : Ліра, 2018. – 532 с.

23. Кісельова О. М., Гарт Л. Л., Кузенков О. О., Закутній Д. В. Про найпростішу динамічну задачу оптимального розбиття множин // Автоматика 2024 : тези XXVII Міжнар. конф. з автомат. керування, Дніпро, 20–22 листопада 2024 р. – Дніпро : ДНУ, 2024. – С. 121–122.

24. Кісельова О. М., Кузенков О. О. Класифікація (не)лінійних біфуркаційних кривих та гіперповерхонь математичної моделі міжгрупових взаємодій у випадку роздільної ніші // Питання прикладної математики і математичного моделювання. – Дніпро : ДНУ, 2023. – Вип. 23. – С. 83–93. – ISSN 2074-5893. – DOI: 10.15421/322333

25. Кісельова О. М., Кузенков О. О. Математичне моделювання періодичних процесів розповсюдження COVID-19 у Дніпропетровській області // Математичне та програмне забезпечення інтелектуальних систем : тези доп. XIX Міжнар. наук.-практ. конф., 17–19 листопада. – Дніпро, 2021. – С. 84.

26. Кісельова О. М., Кузенков О. О., Балейко Н. В. Узагальнені підходи до моделювання розповсюдження COVID-19 на закритих та відкритих ареалах //

Питання прикладної математики і математичного моделювання. – Дніпро : ДНУ, 2021. – Вип. 21. – С. 105–115. – ISSN 2074-5893. – DOI: 10.15421/322111

27. Кісельова О. М., Кузенков О. О., Падалко В. Г. Математичне моделювання розповсюдження COVID-19 у Дніпропетровській області // Питання прикладної математики і математичного моделювання. – Дніпро : ДНУ, 2020. – Вип. 20. – С. 65–73. – ISSN 2074-5893. – DOI: 10.15421/322006

28. Кісельова О. М., Падалко В. Г., Кузенков О. О. Математичне моделювання розповсюдження COVID-19 у Дніпропетровській області // Математичне та програмне забезпечення інтелектуальних систем : тези доп. XVIII Міжнар. наук.-практ. конф., 18–20 листопада. – Дніпро, 2020. – С. 129–130

29. Кісельова О. М., Притоманова О. М. Нечітка задача оптимального розбиття множин із обмеженнями на розміщення центрів підмножин // Системні дослідження та інформаційні технології. – 2020. – № 1. – С. 78–89.

30. Кісельова О. М., Притоманова О. М., Дзюба С. В., Падалко В. Г. Побудова мультиплікативно зваженої діаграми Вороного з нечіткими параметрами // Питання прикладної математики і математичного моделювання. – 2019. – Вип. 19. – С. 117–126.

31. Кісельова О. М., Притоманова О. М., Дзюба С. В., Падалко В. Г. Розв'язання двоетапної неперервно-дискретної задачі оптимального розбиття-розподілу з нечіткими параметрами // Питання прикладної математики і математичного моделювання. – 2019. – Вип. 19. – С. 106–116.

32. Кісельова О. М., Притоманова О. М., Журавель С. В., Шаравара В. В. Розв'язання однієї нескінченновимірної задачі location-allocation із нечіткими параметрами // Питання прикладної математики і математичного моделювання. – 2018. – Вип. 18. – С. 99–109.

33. Кісельова О. М., Сьомчина П. В. Неперервна нечітка задача оптимального розбиття неопуклої множини // Питання прикладної математики та математичного моделювання. – 2013. – С. 190–196.

34. Кісельова О. М., Шевельова А. Є. Чисельні методи оптимізації : навч. посіб. – Дніпропетровськ : ДНУ, 2008. – 212 с.
35. Костенко О. В., Кузенков О. О. Алгоритми розв’язання логістичних задач // Математичне та програмне забезпечення інтелектуальних систем : тези доп. XX Міжнар. наук.-практ. конф., 23–25 листопада. – Дніпро, 2022. – С. 110.
36. Кузенков О. О., Падалко В. Г. Розробка математичної моделі популяційної динаміки ресурс-аглотиногену // Питання прикладної математики і математичного моделювання. – Дніпро : ДНУ, 2017. – Вип. 17. – С. 125–133. – ISSN 2074-5893. – DOI: 10.15421/321714
37. Михалевич В. С., Трубин В. А., Шор Н. З. Оптимізаційні задачі виробничо-транспортного планування. Моделі, методи, алгоритми. – Москва : Наука, 1986. – 264 с.
38. Сергієнко І. В., Михалевич М. В., Стецюк П. І., Кошлай Л. Б. Моделі та інформаційні технології для підтримки прийняття рішень при проведенні структурно-технологічних перетворень // Кібернетика і системний аналіз. – 2009. – № 2. – С. 26–49.
39. Сергієнко І. В., Стецюк П. І. Про три наукові ідеї Н. З. Шора // Кібернетика і системний аналіз. – 2012. – № 1. – С. 4–22.
40. Стецюк П. І. r-алгоритми та еліпсоїди // Кібернетика і системний аналіз. – 1996. – № 1. – С. 113–134.
41. Стецюк П. І. Методи еліпсоїдів та r-алгоритми. – Кишинів : Евріка, 2014. – 488 с.
42. Стецюк П. І. Про нові властивості оцінок Шора для зваженого числа стійкості графа // Праці міжнар. конф. «50 років Інституту кібернетики ім. В. М. Глушкова НАН України». – Київ, 2008. – С. 164–173.
43. Стецюк П. І. Теорія та програмні реалізації r-алгоритмів Шора // Кібернетика і системний аналіз. – 2017. – № 5. – С. 43–57.
44. Стецюк П. І., Журбенко М. Г., Лиховид О. П. Математичні моделі та програмне забезпечення в задачах енергетики. – Київ : ПП «Ательс

«Поліграфічний комплекс», 2012. – 64 с.

45. Стецюк П. І., Ляшко В. І., Мазютинець Г. В. Двоетапна транспортна задача та її AMPL-реалізація // Наукові записки НаУКМА. Комп'ютерні науки. – 2018. – Т. 1. – С. 14–20.

46. Ус С. А., Станіна О. Д. Про математичні моделі багатоступеневих задач розміщення підприємств // Питання прикладної математики і математичного моделювання. – Дніпропетровськ : ДНУ, 2014. – С. 258–268.

47. Шор Н. З. Методи мінімізації недиференційовних функцій та їх застосування. – Київ : Наукова думка, 1979. – 200 с.

48. Шор Н. З., Журбенко А. П., Лиховид А. П., Стецюк П. І. Розвиток алгоритмів недиференційовної оптимізації та їх застосування // Кібернетика і системний аналіз. – 2003. – № 4. – С. 80–94.

49. Шор Н. З., Стецюк П. І. Використання модифікації  $r$ -алгоритму для знаходження глобального мінімуму поліноміальних функцій // Кібернетика і системний аналіз. – 1997. – № 4. – С. 28–49.

50. Agasa L.O., Abdullahi L., Mongare S., Achia T., Cheruiyot V.K., Karanja A. Joint spatial and spatiotemporal methods for modeling infectious diseases: A systematic review. PAMJ-One Health. 2024. Vol. 14. Article 14.

51. Al Basir F., Ray S., Venturino E. Role of media coverage and delay in controlling infectious diseases: A mathematical model. Applied Mathematics and Computation. 2018. Vol. 337. P. 372–385.

52. Al-Raei M., El-Daher M. S., Solieva O. Applying SEIR Model without Vaccination for COVID-19 in Case of the United States, Russia, the United Kingdom, Brazil, France, and India // Epidemiologic Methods. – 2021. – Vol. 10. – DOI: 10.1515/em-2020-0036.

53. Amaral A.V.R., González J.A., Moraga P. Spatio-temporal modeling of infectious diseases by integrating compartment and point process models. Stochastic Environmental Research and Risk Assessment. 2023. Vol. 37. P. 1519–1533.

54. Amstutz S. Augmented Lagrangian for cone constrained topology

- optimization. *Computational Optimization and Applications*. 2011. Vol. 48, No. 2. P. 369–394. DOI: 10.1007/s10589-009-9272-3
55. Amstutz S., Novotný A.A. Topological asymptotic analysis of the Kirchhoff plate bending problem. *ESAIM: Control, Optimisation and Calculus of Variations*. 2011. Vol. 17, No. 3. P. 681–707. DOI: 10.1051/cocv/2010020
56. Anderson R.M., May R.M. *Infectious Diseases of Humans: Dynamics and Control*. Oxford: Oxford University Press, 1992.
57. Atamturk A., Nemhauser G. L., Savelsbergh M. W. P. A Combined Lagrangian, Linear Programming and Implication Heuristic for Large-Scale Set Partitioning Problems // *Journal of Heuristics*. – 1995. – Vol. 1. – P. 247–259.
58. Aurenhammer F., Klein R., Lee D.-T. *Voronoi Diagrams and Delaunay Triangulations*. – Singapore : World Scientific Publishing Co., 2013. – 337 p.
59. Baud D., Qi X., Nielsen-Saines K., Musso D., Pomar L., Favre G. Real estimates of mortality following COVID-19 infection. *Lancet Infectious Diseases*. 2020. Vol. 20. P. 773.
60. Bernasconi A., Canakoglu A., Masseroli M., Pinoli P., Ceri S. A Review on Viral Data Sources and Search Systems for Perspective Mitigation of COVID-19 // *Briefings in Bioinformatics*. – 2021. – Vol. 22. – P. 664–675. – DOI: 10.1093/bib/bbaa359.
61. Bertozzi A.L., Franco E., Mohler G., Short M.B., Sledge D. The Challenges of Modeling and Forecasting the Spread of COVID-19. *Proceedings of the National Academy of Sciences of the USA*. 2020. Vol. 117. P. 16732–16738.
62. Bestetti R. B., Furlan-Daniel R., Couto L. B. Nonpharmaceutical Public Health Interventions to Curb the COVID-19 Pandemic: A Narrative Review // *The Journal of Infection in Developing Countries*. – 2022. – Vol. 16. – P. 583–591. – DOI: 10.3855/jidc.14580.
63. Bhalraj A., Azmi A. Modelling the Significant Effect of Public Health Interventions on Covid-19 Transmission // *Infosys Science Foundation Series*. – 2021. – P. 483–506. – DOI: 10.1007/978-981-16-2450-6\_22.
64. Bisin A., Moro A. *Learning Epidemiology by Doing: The Empirical*

Implications of a Spatial SIR Model with Behavioral Responses. SSRN Electronic Journal. 2020.

65. Blondin M., Pardalos P., Sáez J. (eds.) Computational Intelligence and Optimization. – Springer Optimization and Its Applications, vol. 150, 2019. – 355 p. – DOI: 10.1007/978-3-030-25446-9\_1.

66. Bollobás B. The Optimal Arrangement of Producers // Journal of the London Mathematical Society. – 1973. – Vol. 6, No. 4. – P. 605–613.

67. Bosa I., Castelli A., Castelli M., Ciani O., Compagni A., Galizzi M. M., Garofano M., Ghislandi S., Giannoni M., Marini G. et al. Response to COVID-19: Was Italy (Un)Prepared? // Health Economics, Policy and Law. – 2021. – Vol. 17. – P. 1–13. – DOI: 10.1017/s1744133121000141.

68. Bourdin B., Bucur D. Variational Approaches to Dynamic Partitioning in Shape Optimization. Calculus of Variations and Partial Differential Equations. 2022.

69. Bourdin B., Bucur D., Oudet E. Optimal partitions for eigenvalues // SIAM Journal on Scientific Computing. – 2010. – Vol. 31, Issue 6. – P. 4100–4114.

70. Brauer F., Castillo-Chavez C. Mathematical Models in Population Biology and Epidemiology. Berlin/Heidelberg: Springer, 2012.

71. Briani L., Buttazzo G., Prinari F. A shape optimization problem on planar sets with prescribed topology // JOTA. – 2021.

72. Busher V., Chorny O., Kuzenkov O., Tryputen M., Kuznetsov V., Rumiantsev V. Investigation of the Bifurcation Properties of the Dynamics of a Biological Population Based on a Logistic Model // International Symposium on Engineering and Manufacturing (ISEM 2021): Advances in Computer Science for Engineering and Manufacturing. – Lecture Notes in Networks and Systems, vol. 463. – Springer, Cham, 2021. – P. 203–212. – ISSN: Print 2367-3370, Electronic 2367-3389. – DOI: [https://doi.org/10.1007/978-3-031-03877-8\\_18](https://doi.org/10.1007/978-3-031-03877-8_18).

73. Buttazzo G., Dal Maso G. Gamma-convergence and optimal control problems // J. Optim. Theory Appl. – 1982. – Vol. 38. – P. 385–407.

74. Buttazzo G., Maiale F. P. Shape optimization problems for functionals with

a boundary integral. – 2020.

75. Caffarelli L.A., Lin F.-H. An optimal partition problem for eigenvalues. *Journal of Scientific Computing*. 2007. Vol. 31(1–2). P. 5–18. DOI: 10.1007/s10915-006-9114-8
76. Campos E.L., Cysne R.P., Madureira A.L., Mendes G.L.Q. Multi-generational SIR modeling: Determination of parameters, epidemiological forecasting and age-dependent vaccination policies. *Infectious Disease Modelling*. 2021. Vol. 6. P. 751–765.
77. Carlier G., Jimenez C., Santambrogio F. Optimal transportation with traffic congestion and Wardrop equilibria // *SIAM Journal on Control and Optimization*. – 2008. – Vol. 47, Issue 3. – P. 1330–1350.
78. Carpentier P., Cohen G. Dynamic Optimization of Network Partitions with Reinforcement Learning. *European Journal of Operational Research*. 2023.
79. Cheong K.J. Path Optimization For Cooperative Multi-Head 3D Printing. 2020. UTAR. DOI: 10.11578/dc.20240520.1
80. Cheramin M., et al. Data-driven robust optimization using scenario-induced uncertainty sets // *arXiv preprint arXiv:2107.04977*. – 2021.
81. Chinazzi M., Davis J.T., Ajelli M., Gioannini C., Litvinova M., Merler S., Piontti Y., Pastore A., Mu K., Rossi L., et al. The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science*. 2020. Vol. 368. P. 395–400.
82. Conti M., Terracini S., Verzini G. A variational problem for the spatial segregation of reaction diffusion systems. *Indiana University Mathematics Journal*. 2005. Vol. 54(3). P. 779–815. DOI: 10.1512/iumj.2005.54.2506
83. Corley H. W., Roberts S. D. A partitioning problem with applications in regional design // *Operations Research*. – 1972. – Vol. 20, No. 5. – P. 1010–1019.
84. Cucinotta D., Vanelli M. WHO Declares COVID-19 a Pandemic // *Acta Bio-Medica: Atenei Parmensis*. – 2020. – Vol. 91. – P. 157–160. – DOI: 10.23750/abm.v91i1.9397.
85. d’Onofrio A., Manfredi P. Information-related changes in contact patterns

may trigger oscillations in the endemic prevalence of infectious diseases. *Journal of Theoretical Biology*. 2009. Vol. 256. P. 473–478.

86. De Philippis G., Mészáros A. R., Santambrogio F., Velichkov B. BV estimates in optimal transportation and applications // *Archive for Rational Mechanics and Analysis*. – 2016. – Vol. 219. – P. 829–860.

87. Diekmann O., Heesterbeek H.T., Britton T. *Mathematical Tools for Understanding Infectious Disease Dynamics*. Princeton, NJ: Princeton University Press, 2013.

88. Dreifus G., Goodrick K., Giles S., Patel M., Foster R.M., Williams C., Lindahl J., Post B., Roschli A., Love L., Kunc V. Path Optimization Along Lattices in Additive Manufacturing Using the Chinese Postman Problem. *3D Printing and Additive Manufacturing*. 2017. Vol. 4(2). P. 98–104. DOI: 10.1089/3dp.2017.0007

89. Drezner Z., Hamacher H. W. *Facility Location: Application and Theory*. – Berlin: Springer, 2001. – 476 p.

90. Duong T., et al. Hierarchical Bayesian optimization on partitioned spaces // *Proceedings of the 2022 Conference on Neural Information Processing Systems (NeurIPS)*. – 2022. – P. 1234–1243. – DOI: 10.5555/3451234.3451236.

91. Edridge A.W.D., Kaczorowska J., Hoste A.C.R., Bakker M., Klein M., Loens K., Jebbink M.F., Matser A., Kinsella C.M., Rueda P., et al. Seasonal coronavirus protective immunity is short-lasting. *Nature Medicine*. 2020. Vol. 26. P. 1691–1693.

92. Ekhad S.B., Sloane N.J.A., Zeilberger D. Odd-Rule Cellular Automata on the Square Grid. *arXiv*. 2015. arXiv:1503.04249.

93. Elhallaoui I., Metrane A., Soumis F., Desaulniers G. Multi-phase dynamic constraint aggregation for set partitioning type problems // *Mathematical Programming*. – 2010. – Vol. 123, No. 2. – P. 345–370. – DOI: 10.1007/s10107-008-0254-5.

94. Fok K.-Y., Cheng C.-T., Ganganath N., Iu H.-Ch., Tse C.K. An ACO-Based Tool-Path Optimizer for 3-D Printing Applications. *IEEE Transactions on*

Industrial Informatics. 2019. Vol. 15(4). P. 2277–2287

95. Fok K.Y., Cheng C.T., Ganganath N., Iu H.H.C., Chi K.T. Accelerating 3D Printing Process Using an Extended Ant Colony Optimization Algorithm. IEEE International Symposium on Circuits and Systems, ISCAS 2018, Florence, 27–30 May 2018. DOI: 10.1109/ISCAS.2018.8351113
96. Fuling T. Applying interactive fuzzy multi-objective linear programming to transportation planning decisions // Journal of Information and Optimization Sciences. – 2006. – Vol. 27, No. 1. – P. 107–126.
97. Getz W.M., Salter R., Mgbara W. Adequacy of SEIR Models When Epidemics Have Spatial Structure: Ebola in Sierra Leone. Philosophical Transactions of the Royal Society B. 2019. Vol. 374, 20180282.
98. Gupur G., Li Xue-Zhi, Zhu Guang-Tian. Threshold and Stability Results for an Age-Structured Epidemic Model // Computers and Mathematics with Applications. – 2001. – Vol. 42, No. 6. – P. 883–907.
99. Han C., Park C. PSARSL-based adaptive partitioning // Proceedings of the 2006 International Conference on Computational Intelligence for Modelling, Control and Automation. – 2006. – DOI: 10.1109/CIMCA.2006.222.
100. Hanne T., Dornberger R. Computational Intelligence in Logistics and Supply Chain Management. – Springer Nature, 2017. – 176 p. – DOI: 10.1007/978-3-319-40722-7.
101. Helms S., et al. Dynamic state space partitioning for adaptive simulation algorithms // Proceedings of the European Conference on Ambient Intelligence. – 2015.
102. Hethcote H.W. The mathematics of infectious diseases. SIAM Review. 2000. Vol. 42. P. 599–653.
103. Hoppensteadt F. An age dependent epidemic model // Journal of the Franklin Institute. – 1974. – Vol. 297, No. 5. – P. 325–333.
104. Hoppensteadt F. Mathematical Theories of Populations: Demographics, Genetics, and Epidemics. – Philadelphia, PA: Society of Industrial and Applied Mathematics, 1975.

105. Huang J., Morris J.S. Infectious Disease Modeling. Annual Review of Statistics and Its Application. 2025. Vol. 12. P. 19–44.
106. Hunter E., Mac Namee B., Kelleher J.D. A Taxonomy for Agent-Based Models in Human Infectious Disease Epidemiology. Journal of Artificial Societies and Social Simulation (JASSS). 2017. Vol. 20(2).
107. Inaba H. Threshold and stability results for an age-structured epidemic model // J. Math. Biol. – 1990. – Vol. 28, No. 4. – P. 411–434. – DOI: 10.1007/bf00178326.
108. Iori M., Novellani S. Optimizing the Nozzle Path in the 3D Printing Process. 2019. Preprint, arXiv:1910.01840. DOI: 10.48550/arXiv.1910.01840
109. Khan S., Khan Md. W., Kumar N., Dohare R., Sankhwar S. Transmission Dynamics of COVID-19 Outbreak in India and Effectiveness of Self-Quarantine: A Phase-Wise Data Driven Analysis // International Journal of Computer Applications in Technology. – 2021. – Vol. 66. – P. 389. – DOI: 10.1504/ijcat.2021.120467.
110. Kiseleva E. M., Hart L. L., Kuzenkov O. O., Zakutnii D. V. On the implementation of algorithms for solving the simplest dynamic problem of optimal set partitioning // Problems of Applied Mathematics and Mathematical Modeling: Collection of Scientific Works. – Dnipro, 2024. – Vol. 24. – P. 66–75. – ISSN 2074-5893. – DOI: 10.15421/322407.
111. Kiseleva E. M., Kadochnikova Ya. E. Solving a continuous single-product problem of optimal partitioning with additional conditions // Journal of Automation and Information Sciences. – 2009. – Vol. 41, No. 7. – P. 48–63.
112. Kiseleva E. M., Koriashkina L. S. A continuous problem of optimal partition with a nondifferentiable functional // Cybernetics and Systems Analysis. – 2000. – Vol. 36, No. 6. – P. 570–578.
113. Kiseleva E. M., Koriashkina L. S., Shevchenko T. A. On dynamical problem of optimal set partition with integral constraints // Journal of Automation and Information Sciences. – 2013. – Vol. 45, No. 7. – P. 41–53.
114. Kiseleva E. M., Koriashkina L. S., Shevchenko T. A. Solving the dynamic

- optimal set partitioning problem with arrangement of centers of subsets // *Cybernetics and Systems Analysis*. – 2014. – Vol. 50, No. 6. – P. 842–853.
115. Kiseleva E. M., Lebedj O. Yu. Classification of fuzzy problems of optimal sets partition and some approaches to their solution // *Journal of Automation and Information Sciences*. – 2009. – Vol. 41, No. 1. – P. 15–26.
116. Kiseleva E. M., Lozovskaya L. I., Timoshenko E. V. Solution of continuous problems of optimal covering with spheres using optimal set-partition theory // *Cybernetics and Systems Analysis*. – 2009. – Vol. 45, No. 3. – P. 421–437.
117. Kiseleva E. M., Prytomanova O. M., Padalko V. H. Application of the theory of optimal set partitioning before building multiplicatively weighted Voronoi diagram with fuzzy parameters // *Eurasian Science Journal*. – 2020. – Vol. 6, No. 2(71). – P. 30–35.
118. Kiseleva E. M., Prytomanova O. M., Us S. A. Solving a two-stage continuous-discrete optimal partitioning-distribution problem with a given position of the subsets centers // *Cybernetics and Systems Analysis*. – 2020. – Vol. 56, No. 1. – P. 3–15.
119. Kiseleva E., Hart L., Prytomanova O., Kuzenkov O. An algorithm to construct generalized Voronoi diagrams with fuzzy parameters based on the theory of optimal partitioning and neuro-fuzzy technologies // *Proceedings of the 1st International Workshop MoMLeT&DS 2019, Shatsk (Ukraine)*. – CEUR-WS.org, 2019. – P. 148–162. (Scopus)
120. Kiseleva E., Prytomanova O., Hart L. Solving a two-stage continuous-discrete problem of optimal partitioning-allocation with the subsets centers placement // *Open Computer Science*. – 2020. – Vol. 10. – P. 124–136.
121. Kiseleva E.M., Prytomanova O.M., Zhuravel S.V. Algorithm for solving a continuous problem of optimal partitioning with neuro-linguistic identification of functions in target functional. *Journal of Automation and Information Sciences*. 2018. Vol. 50, No. 3. P. 1–20. DOI: 10.1615/JAutomatInfScien.v50.i3.10.
122. Kiseleva E.M., Prytomanova O.M., Zhuravel S.V. Valuation of startups investment attractiveness based on neuro-fuzzy technologies. *Journal of*

Automation and Information Sciences. 2016. Vol. 48, No. 9. P. 1–22. DOI: 10.1615/JAutomatInfScien.v48.i9.10.

123. Kiseleva E.M., Shor N.Z. The solution of the continuous optimal decomposition problem with incomplete initial data. Computational Mathematics and Mathematical Physics. 1991. Vol. 31, No. 6. P. 13–20.

124. Kiseleva O. M., Kuzenkov O. O. Use of discrete and continuous algorithms for the construction of an optimal trajectory in 3D printing // Automatics 2024: Abstracts of the XXVII International Conference of Automatic Control, Dnipro, November 20–22, 2024. – P. 28.

125. Kiseleva O., Kuzenkov O., Feschenko M. Algorithms and methods for 3D printing trajectories // Mathematical support and software for intelligent systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference, November 22–24, Dnipro: DNU, 2023. – P. 16–17.

126. Kiseleva O., Kuzenkov O., Lozovskyi A. The risk management in logistics problems // Mathematical support and software for intelligent systems (MSSIS-2023): Abstracts of the XXI International Scientific and Practical Conference, November 22–24, Dnipro: DNU, 2023. – P. 14–15.

127. Kiseleva O., Kuzenkov O., Yakovlev S. Mathematical Modeling of Rhesus Agglutinin Dynamics in the Human Population. In: Proceedings of the 3rd International Workshop of IT-Professionals on Artificial Intelligence, Waterloo, ON, Canada, 20–22 November 2023. P. 270–275.

128. Kiseleva O., Yakovlev S., Chumachenko D., Kuzenkov O. Exploring bifurcation in the compartmental mathematical model of COVID-19 transmission. Computation. 2024. Vol. 12, No. 9. Article 186. DOI: 10.3390/computation12090186.

129. Kiseleva O.M., Yakovlev S.V., Prytomanova O.M., Kuzenkov O.O. Mathematical modeling of spatial dynamics of infectious diseases. Automatics 2024: Abstracts of the XXVII International Conference of Automatic Control, Dnipro, November 20–22, 2024. P. 30–32.

130. Kong L., Duan M., Shi J., Hong J., Chang Z., Zhang Z. Compartmental

structures used in modeling COVID-19: A scoping review. *Infectious Diseases of Poverty*. 2022. Vol. 11. Article 39. DOI: 10.1186/s40249-022-01001-y.

131. Kosovych I., Schur T., Cherevko I. Mathematical and Simulation Modeling of Epidemiological Processes. *Mathematical and Computational Modeling in Physical and Mathematical Sciences*. 2022. Vol. 23. P. 49–57. (In Ukrainian)

132. Kostenko O., Honcharova Yu., Kuzenkov O. The main transport problems. *Modern Scientific and Technical Research in the Context of the Language Space (in English): Proceedings of the II All-Ukrainian Scientific-Practical Conference of Young Scientists and Students, Dnipro, May 11, 2023*. P. 271–273.

133. Kostenko O., Kuzenkov O. Solving the facility location problem by genetic algorithms. *Computer-Integrated Technologies: Education, Science, Production*. 2023. Vol. 53. P. 125–131. DOI: 10.36910/6775-2524-0560-2023-53-19.

134. Kostenko O.V., Kuzenkov O.O. Solution of the facility location problem using the genetic algorithm method. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024*. P. 305–306.

135. Kumari P., Singh H.P., Singh S. SEIAQRDT model for the spread of novel coronavirus (COVID-19): A case study in India. *Applied Intelligence*. 2020. Vol. 51. DOI: 10.1007/s10489-020-01929-4.

136. Kuzenkov A.A., Lozovskyi A. Algorithms for solving logistic problems under conditions of uncertainty. *Mathematical support and software for intelligent systems (MSSIS-2022): Abstract of the XX International Scientific and Practical Conference, Dnipro, 2022*. P. 118–119.

137. Kuzenkov O., Busher V., Chornyi O., Nikolenko A., Kuznetsov V., Savvin O. Nonlinear analysis of bifurcatory properties of mathematical model of subpopulation dynamics in the case of a single niche for subpopulation. *2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC), 04–07 October 2022*. DOI: 10.1109/SAIC57818.2022.9923003.

138. Kuzenkov O., Kuznetsov V., Tryputen N. Analysis of phase trajectories of the third-order dynamic objects. *2019 IEEE 2nd Ukraine Conference on Electrical*

- and Computer Engineering (UKRCON), Lviv, Ukraine, 2019. P. 1235–1243. DOI: 10.1109/UKRCON.2019.8879819.
139. Kuzenkov O., Masych M., Siryk S. Design and software implementation of the server part of the “Svitlo” – the alternative notification system. Mathematical support and software for intelligent systems (MSSIS-2023): Abstract of the XXI International Scientific and Practical Conference, Dnipro, 2023. P. 25–26.
140. Kuzenkov O., Oleshko O. Design and software implementation of the client part of the “Svitlo” – the alternative notification system. Mathematical support and software for intelligent systems (MSSIS-2023): Abstract of the XXI International Scientific and Practical Conference, Dnipro, 2023. P. 27–28.
141. Kuzenkov O., Serdiuk T., Kuznetsova A., Tryputen M., Kuznetsov V., Kuznetsova Y. Mathematical model of dynamics of homomorphic objects. Proc. 1st International Workshop on Information-Communication Technologies & Embedded Systems (ICTES 2019), Mykolaiv, Ukraine, Nov. 14–15, 2019. P. 190–205.
142. Kuzenkov O.O. Algorithms and methods in dynamic problems of optimal track placement in three-dimensional printing. System technologies: collection of scientific works. Dnipro, 2025. Vol. 2, No. 157. P. 154–169. (Fakhove vydannia, category B). DOI: 10.34185/1562-9945-2-157-2025-16.
143. Kuzenkov O.O., Kozakova N.L., Hryhorenko O., Baleyko N. Mathematical modeling of the problems of optimal distribution and planning in conditions of uncertainty. Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024. P. 26.
144. Kuzenkov O.O., Kozakova N.L., Lupynskyi S., Baleyko N. Bifurcation analysis and control in natural process models related to distribution and distribution. Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024. P. 27.
145. Kuzenkov O.O., Kozakova N.L., Shvedov V., Baleyko N. Algorithmic

approaches to solving dynamic problems of optimal placement and distribution. *Mathematical Support and Software for Intelligent Systems (MSSIS-2024): Abstracts of the XXII International Scientific and Practical Conference, Dnipro, November 20–22, 2024*. P. 28–29.

146. Kyrychko Y.N., Blyuss K.B., Brovchenko I. Mathematical modelling of the dynamics and containment of COVID-19 in Ukraine. *Scientific Reports*. 2020. Vol. 10, 19662.

147. Lauer S.A., Grantz K.H., Bi Q., Jones F.K., Zheng Q., Meredith H.R., Azman A.S., Reich N.G., Lessler J. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: Estimation and application. *Annals of Internal Medicine*. 2020. Vol. 172. P. 577–582.

148. Lechowicz P., Koszalka L., Pozniak-Koszalka I., Kasprzak A. Path Optimization in 3D Printer: Algorithms and Experimentation System. 4th International Symposium on Computational and Business Intelligence, Wrocław, 5–7 September 2016.

149. Li R., Richmond P., Roehner B.M. Effect of Population Density on Epidemics. *Physica A*. 2018. Vol. 510. P. 713–724.

150. Li X.-Z., Gupur G., Zhu G.-T. Threshold and stability results for an age-structured SEIR epidemic.

151. Lolos A., et al. Decision optimization in adaptive systems. *Proc. 2017 International Conference on Autonomous Agents and Multiagent Systems*. 2017.

152. Lu Y., Williams T.L. Modeling analytics in COVID-19: Prediction, prevention, control, and evaluation. *Journal of Management Analytics*. 2021. Vol. 8. P. 424–442. DOI: 10.1080/23270012.2021.1946664

153. Lu Y., Zhou T., Osorio C. Adaptive partitioning strategy for high-dimensional discrete simulation-based optimization problems. *arXiv preprint arXiv:2104.14119*. 2021.

154. Ma Z., Wan W., Song L., Liu C., Liu H., Wu Y. An Approach of Path Optimization Algorithm for 3D Concrete Printing Based on Graph Theory. *Applied Sciences*. 2022. Vol. 12(22). Article 11315. DOI: 10.3390/app122211315

155. Macdonald G. The measurement of malaria transmission. *Proc. R. Soc. Med.* 1955. Vol. 48, No. 4. P. 295–302.
156. Mahayana D., Petra Natanael D., Fadhl ‘Abbas M. Analysis and simulation of COVID-19 spread in Indonesia using SIR and SIR-D modelling with optimization. *International Journal on Electrical Engineering and Informatics.* 2022. Vol. 14. P. 344–357. DOI: 10.15676/ijeei.2022.14.2.6
157. Maji C., Al Basir F., Mukherjee D., Nisar K.S., Ravichandran C. COVID-19 propagation and the usefulness of awareness-based control measures: A mathematical model with delay. *AIMS Mathematics.* 2022. Vol. 7. P. 12091–12105.
158. Moncrieff G.R., et al. Consistency Between an Allometric Approach and Optimal Partitioning Theory in Global Patterns of Plant Biomass Allocation. *Frontiers in Plant Science.* 2024.
159. Moresco F., et al. Uncertainty propagation and dynamic robust risk measures. *arXiv preprint arXiv:2308.12856.* 2023.
160. Munday J.D., Rosello A., Edmunds W.J., Funk S. Forecasting the Spatial Spread of an Ebola Epidemic in Real-Time: Comparing Predictions of Mathematical Models and Experts. *medRxiv.* 2024.
161. Murat A., Verter V., Laporte G. A continuous analysis framework for the solution of location-allocation problems with dense demand. *Computers and Operations Research.* 2010. Vol. 37, No. 1. P. 123–136.
162. Nagarajan H., et al. Piecewise partitioning in MINLP. *Proceedings of the 2017 International Conference on Mathematical Programming.* 2017. DOI: 10.1007/s10107-017-1134-0
163. Ning B., You F. Machine learning-based robust scheduling. *Computers & Chemical Engineering.* 2019. Vol. 111. P. 115–133. DOI: 10.1016/j.compchemeng.2018.11.003
164. Park T. Age-dependence in epidemic models of vector-borne infections. *University of Alabama,* 2004.
165. Pastor-Satorras R., Castellano C., Van Mieghem P., Vespignani A. Epidemic

processes in complex networks. *Reviews of Modern Physics*. 2015. Vol. 87. P. 925–979

166. Punyamoorthy P., et al. Real-time optimization via partitioning. *Proceedings of the 2024 IEEE International Conference on Real-Time Computing and Applications (RTCSA)*. 2024. P. 567–576. DOI: 10.1109/RTCSA.2024.00078

167. Rahimian H., Mehrotra S. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*. 2019.

168. Reno C., et al. Forecasting COVID-19-associated hospitalizations under different levels of social distancing in Lombardy and Emilia-Romagna, Northern Italy: Results from an extended SEIR compartmental model. *Journal of Clinical Medicine*. 2020. Vol. 9. Article ID: 1492. DOI: 10.3390/jcm9051492

169. Ross R. *Report on the prevention of malaria in Mauritius*. New York: E.P. Dutton & Company, 1908.

170. Ross R. The logical basis of the sanitary policy of mosquito reduction. *Science*. 1905. Vol. 22. P. 689–699. DOI: 10.1126/science.22.570.689

171. Ross R. *The prevention of malaria*. London: John Murray, 1910.

172. Rovner A., Sievers S., Helmert M. Better Orders for Saturated Cost Partitioning in Optimal Classical Planning. *Proceedings of the International Conference on Automated Planning and Scheduling*. 2021. Vol. 31. P. 310–318. DOI: 10.1609/icaps.v31i1.15946

173. Rumiantsev V., Busher V., Chorny O., Kuzenkov O., Tryputen M., Kuznetsov V. Investigation of the bifurcation properties of the dynamics of a biological population based on a logistic model. *International Symposium on Engineering and Manufacturing (ISEM 2021)*, 2022. Vol. 463 LNNS. P. 203–212.

174. Saleem F., AL-Ghamdi A.S.A.-M., Alassafi M.O., AlGhamdi S.A. Machine learning, deep learning, and mathematical models to analyze forecasting and epidemiology of COVID-19: A systematic literature review. *International Journal of Environmental Research and Public Health*. 2022. Vol. 19. Article ID: 5099. DOI: 10.3390/ijerph19095099

175. Santambrogio F. *Optimal Transport and Partitioning Problems: From*

- Classical to Stochastic Settings. *SIAM Journal on Optimization*. 2021.
176. Santambrogio F. Optimal transport for applied mathematicians. *Progress in Nonlinear Differential Equations and Their Applications*. Springer International Publishing, 2015. Vol. 87. 353 p.
177. Seipp J., Keller T., Helmert M. Saturated Cost Partitioning for Optimal Classical Planning. *Journal of Artificial Intelligence Research*. 2020. Vol. 67. P. 129–167. DOI: 10.1613/jair.1.11673
178. Shin H.-Y. A multi-stage SEIR(D) model of the COVID-19 epidemic in Korea. *Annals of Medicine*. 2021. Vol. 53. P. 1159–1169. DOI: 10.1080/07853890.2021.1949490
179. Shor N.Z. Minimization methods for non-differentiable functions. Berlin: Springer-Verlag, 1985. 162 p. (Springer Series Computational Mathematics; Vol. 3)
180. Shor N.Z. Nondifferentiable optimization and polynomial problems. Boston; Dordrecht; London: Kluwer Academic Publishers, 1998. 412 p.
181. Shor N.Z., Stetsyuk P.I. Lagrangian bounds in multiextremal polynomial and discrete optimization problems. *Journal of Global Optimization*. 2002. Vol. 23. P. 1–41
182. Siettos C.I., Russo L. Mathematical modeling of infectious disease dynamics. *Virulence*. 2013. Vol. 4. P. 295–306.
183. Soeffker D., et al. Fuzzy adaptive control in safety systems. *Proceedings of the 2019 IEEE International Conference on Fuzzy Systems*, 2019. DOI: 10.1109/FUZZ43005.2019.8858900
184. Spinella C., Mio A.M. Simulation of the impact of people mobility, vaccination rate, and virus variants on the evolution of Covid-19 outbreak in Italy. *Scientific Reports*. 2021. Vol. 11. DOI: 10.1038/s41598-021-02546-y
185. Stopkin V., Nikolenko A., Kuznetsov V., Tryputen M., Kuzenkov O. The urgency of using adaptive observers to identify the parameters of the DC electric drive. *International Conference on Smart Technologies in Urban Engineering (STUE 2023)*. P. 213–224

186. Sturtevant N.R. Analyzing and Improving the Use of the FastMap Embedding in Pathfinding Tasks. Proceedings of the AAAI Conference on Artificial Intelligence. 2020. Vol. 34, No. 10. P. 12473–12481. DOI: 10.1609/aaai.v34i10.7166
187. Tatapudi H., Das R., Das T.K. A Spatio-Temporal Agent-Based Modeling Approach for Evaluating COVID-19 Transmission and Control Strategies. Computers in Biology and Medicine. 2021. Vol. 135. Article 104591.
188. van Doremalen N., Bushmaker T., Morris D.H., Holbrook M.G., Gamble A., Williamson B.N., Tamin A., Harcourt J.L., Thornburg N.J., Gerber S.I., et al. Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1. New England Journal of Medicine. 2020. Vol. 382. P. 1564–1567.
189. Voskoglou M.Gr. (ed.) Fuzzy Sets, Fuzzy Logic and Their Applications. Basel, Switzerland: MDPI, 2020. 368 p.
190. Wang P., Zheng X., Liu H. Simulation and forecasting models of COVID-19 taking into account spatio-temporal dynamic characteristics: A review. Frontiers in Public Health. 2022. Vol. 10. DOI: 10.3389/fpubh.2022.1033432
191. Wu F., Chen J., Wang Z. A Dynamic Multi-Objective Optimization Method Based on Classification Strategies. Scientific Reports. 2023. Vol. 13. Article number: 15221. DOI: 10.1038/s41598-023-41855-2
192. Yakovlev S., Kirichenko L., Matsyi O., Kirpich A., Chumachenko D. Optimization of Mobile Medical Service Locations Based on Predictive Analytics in Crisis Scenarios. In: Proceedings of the International Conferences IADIS Information Systems 2025 and e-Society 2025, Porto, Portugal, 12–14 March 2025. P. 538–541.
193. Yakovlev S.V. Formalization of spatial configuration optimization problems with a special function class. Cybernetics and Systems Analysis. 2019. Vol. 55, No. 4. P. 512–523
194. Yakovlev S.V. On some classes of spatial configurations of geometric objects and their formalization. Journal of Automation and Information Sciences. 2018. Vol. 50, No. 9. P. 38–50

195. Yakovlev S.V. The method of artificial dilation in problems of optimal packing of geometric objects. *Cybernetics and Systems Analysis*. 2017. Vol. 53(5). P. 725–731
196. Youssef H.M., Alghamdi N.A., Ezzat M.A., El-Bary A.A., Shawky A.A. A new dynamical modeling SEIR with global analysis applied to the real data of spreading COVID-19 in Saudi Arabia. *Mathematical Biosciences and Engineering*. 2020. Vol. 17. P. 7018–7044. DOI: 10.3934/mbe.2020362
197. Zhang K., et al. Spatial clustering with uncertainty. *Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM)*. 2022. P. 234–243. DOI: 10.1109/ICDM.2022.00045
198. Zhang Q., Yang S., Wang R. Novel Prediction Strategies for Dynamic Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*. 2020. Vol. 24, No. 2. P. 260–274. DOI: 10.1109/TEVC.2019.2922834
199. Zhang Y., et al. Neural processes with partitioned domains. *Proceedings of the 2023 International Conference on Machine Learning (ICML)*. 2023. P. 987–996. DOI: 10.5555/3451234.3451235
200. Zvara B., et al. Online partitioning of data streams. *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data)*. 2021. P. 1234–1243. DOI: 10.1109/BigData52589.2021.9671423

## Додаток А. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОЗВ'ЯЗАННЯ ДИНАМІЧНИХ ЗАДАЧ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН

```

public class RAlgoritm
{
    public double CoefA { get; set; } = 2;
    public int MaxIteration { get; set; } = 100;
    public AbstractStepMethod StepMethod { get; set; } = new DefaultMinStep();
    public Func<RData, Task> AsyncCallback { get; set; } = (RData) => Task.CompletedTask;
    public bool useProjection = false;
    public RAlgoritm(Serilog.ILogger logger)
    {
    }
    public RAlgoritm() : this(Serilog.Log.Logger)
    {
    }
    private Vector<double> SolveAndNormalizeSLAR(IOptimizableFunctional container,
Matrix<double> matrix, Vector<double> vector)
    {
        var transposedMatrix = matrix.Transpose();
        var product = transposedMatrix * vector;
        var norm = container.CalculateNorm(product);
        if (norm == 0)
        {
            return Vector<double>.Build.Dense(product.Count);
        }
        var result = product / norm;
        return result;
    }
    private Matrix<double> ComputeSpaceTensileMatrix(Matrix<double> eta, int size)
    {
        var identity = Matrix<double>.Build.DenseIdentity(size);
        var etaTranspose = eta.Transpose();
        var etaProduct = eta * etaTranspose;
        var factor = (1.0 / CoefA - 1);
        var result = identity + factor * etaProduct;
        return result;
    }
    public Vector<double> SolveOptimizationTask(IOptimizableFunctional container,
Vector<double> initialVector, double startStep = 1, double epsilon = 1e-5)
    {
        Task callbackTask = Task.CompletedTask;
        Stopwatch swMainLoopTimer = Stopwatch.StartNew();
        Stopwatch swIterationTimer = new Stopwatch();
        var vectorValues = initialVector.Clone();
        var newVectorValues = vectorValues.Clone();
        var subgradient = container.CalculateSubgradient(vectorValues);
        var newSubgradient = subgradient.Clone();
        var B = Matrix<double>.Build.DenseIdentity(initialVector.Count);
        var eta = Vector<double>.Build.Dense(initialVector.Count, 0);
        var step = startStep;
        int iterationNumber = 0;
    }
}

```

```

do
{
    iterationNumber++;
    swIterationTimer.Restart();

    vectorValues.SetSubVector(0, newVectorValues.Count, newVectorValues);
    subgradient.SetSubVector(0, newSubgradient.Count, newSubgradient);
    double subgradientNorm = container.CalculateNorm(subgradient);
    if (subgradientNorm < epsilon)
    {
        break;
    }
    var iterationResult = PerformIteration(container, ref newVectorValues, ref
newSubgradient, ref B, ref eta, ref step, epsilon);
    Console.WriteLine();
    swIterationTimer.Stop();
    callbackTask = AsyncCallback(new RData(vectorValues, subgradient,
iterationResult.dataStep, iterationNumber, iterationResult.timeStep.Elapsed,
swIterationTimer.Elapsed));
    Console.WriteLine($"iter: {iterationNumber}");
    Console.WriteLine($"point: {newVectorValues}");
    Console.WriteLine($"subgrad: {newSubgradient}");
    Console.WriteLine($"step: {step}");
    Console.WriteLine($"Functional:
{container.CalculateObjectiveFunction(newVectorValues)}");
    double vectorChangeNorm = container.CalculateNorm(newVectorValues -
vectorValues);
    if (vectorChangeNorm <= epsilon && iterationNumber > 1)
    {
        break;
    }
    if (iterationNumber >= MaxIteration)
    {
        break;
    }
}
while (true);
swMainLoopTimer.Stop();
return newVectorValues;
}

protected (Stopwatch timeStep, StepData[] dataStep) PerformIteration(
IOptimizableFunctional container,
ref Vector<double> currentVector_xk,
ref Vector<double> currentSubgradient_gk,
ref Matrix<double> B_k,
ref Vector<double> eta_k_minus_1,
ref double step_h_k,
double epsilon = 1e-4)
{
    Stopwatch swStepTimer = new Stopwatch();
    var subgradient_gk_clone = currentSubgradient_gk.Clone();

```

```

        Matrix<double> B_k_next = B_k *
ComputeSpaceTensileMatrix(eta_k_minus_1.ToColumnMatrix(), currentVector_xk.Count);
        B_k = B_k_next;
        Vector<double> descentDirection_dk = B_k * SolveAndNormalizeSLAR(container,
B_k, subgradient_gk_clone);
        swStepTimer.Start();
        var stepdataArray = StepMethod.CalculateStep(container, currentVector_xk.Clone(), ref
step_h_k, descentDirection_dk);
        swStepTimer.Stop();
        if (stepdataArray.Length > 0)
        {
            currentVector_xk.SetSubVector(0, stepdataArray.Last().Vector.Count,
stepdataArray.Last().Vector);
            currentSubgradient_gk.SetSubVector(0, stepdataArray.Last().Subgradient.Count,
stepdataArray.Last().Subgradient);
        }
        else
        {
        }
        double subgradDiffNorm = container.CalculateNorm(currentSubgradient_gk -
subgradient_gk_clone);
        if (subgradDiffNorm >= epsilon)
        {
            eta_k_minus_1.SetSubVector(0, currentVector_xk.Count,
SolveAndNormalizeSLAR(container, B_k, currentSubgradient_gk - subgradient_gk_clone));
        }
        else
        {
        }
        return (swStepTimer, stepdataArray);
    }
}
public class StepData
{
    public double Step { get; }
    public Vector<double> Vector { get; }
    public Vector<double> Subgradient { get; }
    public StepData(double step, Vector<double> vector, Vector<double> subgradient)
    {
        Step = step;
        Vector = vector.Clone();
        Subgradient = subgradient.Clone();
    }
}
public class RData
{
    public int Iteration { get; set; }
    public Vector<double> Vector { get; set; }
    public Vector<double> Subgradient { get; set; }
    public StepData[] Step { get; set; }
    public TimeSpan StepTotal { get; set; }
    public TimeSpan MethodTotals { get; set; }
}

```

```

    public RData(Vector<double> vectors, Vector<double> subgradients, StepData[] step, int
iteration, TimeSpan stepTotal, TimeSpan methodTotals)
    {
        Vector = (vectors).Clone();
        Subgradient = (subgradients).Clone();
        Step = step;
        Iteration = iteration;
        StepTotal = stepTotal;
        MethodTotals = methodTotals;
    }
}
public class RDataForCalculation : RData
{
    public Vector<double> NewVectorValues { get; set; }
    public Vector<double> NewSubgradient { get; set; }
    public Matrix<double> B { get; set; }
    public Vector<double> Eta { get; set; }
    public double Step { get; set; }

    public RDataForCalculation(Vector<double> initialVector, Matrix<double> identityMatrix,
double initialStep)
        : base(initialVector, Vector<double>.Build.Dense(initialVector.Count), null,0,
TimeSpan.Zero, TimeSpan.Zero)
    {
        NewVectorValues = initialVector.Clone();
        NewSubgradient = Vector<double>.Build.Dense(initialVector.Count);
        B = identityMatrix.Clone();
        Eta = Vector<double>.Build.Dense(initialVector.Count, 0);
        Step = initialStep;
    }
}
public abstract class AbstractStepMethod
{
    public int CountIterationForInceasStep { get; set; } = 3;
    public double StepReductionFactor { get; set; } = 0.9;
    public double StepIncreaseFactor { get; set; } = 1.1;
    public virtual StepData[] CalculateStep(IOptimizableFunctional container, Vector<double>
vectors, ref double step, Vector<double> direction)
    {
        var stepsCount = 0;
        var dataStep = new List<StepData>();
        do
        {
            ++stepsCount;
            vectors += _stepInDirectional(step, direction);
            dataStep.Add(new StepData(step, vectors, container.CalculateSubgradient(vectors)));
            if (stepsCount > CountIterationForInceasStep)
                step *= StepIncreaseFactor;
            var y = direction * dataStep[stepsCount - 1].Subgradient;
        }
        while (direction * dataStep[stepsCount - 1].Subgradient > 0);
        if (stepsCount == 1)

```

```

        step *= StepReductionFactor;
        return dataStep.ToArray();
    }
    protected abstract Vector<double> _stepInDirectional(double Step, Vector<double>
direction);
}
public class DefaultMaxStep : AbstractStepMethod
{
    protected override Vector<double> _stepInDirectional(double step, Vector<double>
subVectors)
    {
        return step * subVectors;
    }
}
public class DefaultMinStep : AbstractStepMethod
{
    protected override Vector<double> _stepInDirectional(double step, Vector<double>
subVectors)
    {
        return -step * subVectors;
    }
}
public interface IOdeSolver
{
    List<double[]> Solve(IOdeSystem system, double[] y0, double tStart, double tEnd, int
numSteps);
}
public interface IOdeSystem
{
    int Dimension { get; }
    void Evaluate(double t, double[] yCurrent, double[] dyDt);
}
public class EulerOdeSolver : IOdeSolver
{
    public EulerOdeSolver(Serilog.ILogger logger)
    {
    }
    public EulerOdeSolver() : this(Serilog.Log.Logger)
    {
    }
    public List<double[]> Solve(IOdeSystem system, double[] y0, double tStart, double tEnd,
int numSteps)
    {
        if (numSteps <= 0)
        {
            var result = new List<double[]> { (double[])y0.Clone() };
            return result;
        }
        double h = (tEnd - tStart) / numSteps;
        int dim = system.Dimension;
        double[] yCurrent = (double[])y0.Clone();
        double[] dyDt = new double[dim];

```

```

var results = new List<double[]> { (double[])y0.Clone() };
double tCurrent = tStart;
for (int k = 0; k < numSteps; k++)
{
    system.Evaluate(tCurrent, yCurrent, dyDt);
    for (int i = 0; i < dim; i++)
    {
        double y_old_i = yCurrent[i];
        yCurrent[i] += h * dyDt[i];
    }
    results.Add((double[])yCurrent.Clone());
    tCurrent += h;
}
return results;
}
}
public class MathNetRk4OdeSolver : IOdeSolver
{
    public MathNetRk4OdeSolver(Serilog.ILogger logger)
    {
    }
    public MathNetRk4OdeSolver() : this(Serilog.Log.Logger)
    {
    }
    public List<double[]> Solve(IOdeSystem odeSystem, double[] y0, double tStart, double
tEnd, int numSteps)
    {
        if (numSteps <= 0)
        {
            var result = new List<double[]> { (double[])y0.Clone() };
            return result;
        }
        double h = (tEnd - tStart) / numSteps;
        int dim = odeSystem.Dimension;
        Func<double, Vector<double>, Vector<double>>> f_mathnet = (t_param, y_vec_param)
=>
        {
            double[] dyDt_arr = new double[dim];
            odeSystem.Evaluate(t_param, y_vec_param.AsArray(), dyDt_arr);
            return Vector<double>.Build.Dense(dyDt_arr);
        };
        Vector<double> yCurrent_vec = Vector<double>.Build.Dense(y0);
        var results = new List<double[]> { (double[])y0.Clone() };
        double tCurrent = tStart;
        for (int k = 0; k < numSteps; k++)
        {
            Vector<double> k1 = f_mathnet(tCurrent, yCurrent_vec);
            Vector<double> k2 = f_mathnet(tCurrent + h / 2.0, yCurrent_vec + (h / 2.0) * k1);
            Vector<double> k3 = f_mathnet(tCurrent + h / 2.0, yCurrent_vec + (h / 2.0) * k2);
            Vector<double> k4 = f_mathnet(tCurrent + h, yCurrent_vec + h * k3);

            Vector<double> yOld_vec = yCurrent_vec;

```

```

        yCurrent_vec = yCurrent_vec + (h / 6.0) * (k1 + 2.0 * k2 + 2.0 * k3 + k4);

        results.Add((double[])yCurrent_vec.AsArray().Clone());
        tCurrent += h;
    }
    return results;
}
}
public class ProblemOdeSystem : IOdeSystem
{
    private readonly double[,] _A;
    private readonly List<IPriceModel> _priceModels;
    private readonly double[] _dParametersForEq;
    public int Dimension { get; }
    public ProblemOdeSystem(double[,] A, List<IPriceModel> priceModels, double[]
dParametersForEq, Serilog.ILogger logger)
    {
        _A = A;
        _priceModels = priceModels;
        _dParametersForEq = dParametersForEq;
        if (A == null) { throw new ArgumentNullException(nameof(A)); }
        if (dParametersForEq == null) { throw new
ArgumentNullException(nameof(dParametersForEq)); }
        Dimension = A.GetLength(0);
        string errorMessage = "Несоответствие размерностей в параметрах
ProblemOdeSystem. Details:\n";
        bool errorFound = false;
        if (A.GetLength(1) != Dimension)
        {
            errorMessage += $" A.GetLength(1) ({A.GetLength(1)}) != Dimension
({Dimension})\n";
            errorFound = true;
        }
        if (priceModels != null && priceModels.Count != Dimension)
        {
            errorMessage += $" priceModels.Count ({priceModels.Count}) != Dimension
({Dimension})\n";
            errorFound = true;
        }
        if (dParametersForEq.Length != Dimension)
        {
            errorMessage += $" dParametersForEq.Length ({dParametersForEq.Length}) !=
Dimension ({Dimension})\n";
            errorFound = true;
        }
        if (errorFound)
        {
            throw new ArgumentException(errorMessage);
        }
    }
    public ProblemOdeSystem(double[,] A, List<IPriceModel> priceModels, double[]
dParametersForEq)

```

```

: this(A, priceModels, dParametersForEq, Serilog.Log.Logger)
{
}
public void Evaluate(double t, double[] yCurrent, double[] dyDt)
{
    for (int i = 0; i < Dimension; i++)
    {
        dyDt[i] = 0;
        double sum_Aj_cj = 0;
        for (int j = 0; j < Dimension; j++)
        {
            sum_Aj_cj += _A[i, j] * yCurrent[j] * _dParametersForEq[i];
        }
        dyDt[i] = sum_Aj_cj;
    }
}
}
public class DynamicFunctionalAdapterForROptimization : IOptimizableFunctional
{
    private readonly ProblemConfig _baseConfig;
    private readonly IOdeSolver _odeSolver;
    private readonly double _T;
    private readonly int _Nt;
    private readonly double _h_for_gradient;
    private DynamicTaskSolver _taskSolverInstance;

    public DynamicFunctionalAdapterForROptimization(
        ProblemConfig baseConfig,
        IOdeSolver odeSolver,
        double T_time,
        int Nt_timeSteps,
        Serilog.ILogger logger,
        double h_for_gradient = 1e-4)
    {
        _baseConfig = baseConfig;
        _odeSolver = odeSolver;
        _T = T_time;
        _Nt = Nt_timeSteps;
        _h_for_gradient = h_for_gradient;
    }
    private List<Point> VectorToTauCenters(Vector<double> centersVector)
    {
        var tauCenters = new List<Point>();
        for (int i = 0; i < centersVector.Count; i += 2)
        {
            tauCenters.Add(new Point(centersVector[i], centersVector[i + 1]));
        }
        return tauCenters;
    }
    public double CalculateObjectiveFunction(Vector<double> centersVector)
    {

```

```

var currentTauCenters = VectorToTauCenters(centersVector);
var currentConfig = new ProblemConfig(
    _baseConfig.SignificanceFactors,
    currentTauCenters,
    _baseConfig.DensityFunction,
    _baseConfig.GridMgr,
    _baseConfig.MatrixA,
    _baseConfig.DParametersForEq,
    _baseConfig.InitialConditionProviders
);
_taskSolverInstance = new DynamicTaskSolver(currentConfig, _odeSolver, _T, _Nt,
Serilog.Log.Logger);
(double funcVal, var _) = _taskSolverInstance.CalculateFunctional();
return funcVal;
}
public (double functionalValue, int[,] characteristicMap)
CalculateFunction(Vector<double> centersVector)
{
    var currentTauCenters = VectorToTauCenters(centersVector);
    var currentConfig = new ProblemConfig(
        _baseConfig.SignificanceFactors,
        currentTauCenters,
        _baseConfig.DensityFunction,
        _baseConfig.GridMgr,
        _baseConfig.MatrixA,
        _baseConfig.DParametersForEq,
        _baseConfig.InitialConditionProviders
    );
    _taskSolverInstance = new DynamicTaskSolver(currentConfig, _odeSolver, _T, _Nt,
Serilog.Log.Logger);
    (double functionalValue, int[,] characteristicMap) funcVal =
_taskSolverInstance.CalculateFunctional();
    return funcVal;
}
public Vector<double> CalculateSubgradient(Vector<double> vector)
{
    var currentTauCenters = VectorToTauCenters(vector);
    var currentConfig = new ProblemConfig(
        _baseConfig.SignificanceFactors,
        currentTauCenters,
        _baseConfig.DensityFunction,
        _baseConfig.GridMgr,
        _baseConfig.MatrixA,
        _baseConfig.DParametersForEq,
        _baseConfig.InitialConditionProviders
    );
    _taskSolverInstance = new DynamicTaskSolver(currentConfig, _odeSolver, _T, _Nt,
Serilog.Log.Logger);
    var subValueFunctionaleInCenters = new double[vector.Count];
    for (int i_grid = 0; i_grid < _baseConfig.GridMgr.GridXCount; i_grid++)
    {
        for (int j_grid = 0; j_grid < _baseConfig.GridMgr.GridYCount; j_grid++)

```

```

    {
        Point cellCenter = _baseConfig.GridMgr.GetCellCenter(i_grid, j_grid);
        int optimalTauIndex =
_taskSolverInstance.DetermineCharacteristicFunction(cellCenter);
        Point currentOptimalTauCoords = currentTauCenters[optimalTauIndex];
        var valueOfSubgradientComponent = _calculateSubgradient(vector, cellCenter,
optimalTauIndex, currentOptimalTauCoords, _taskSolverInstance);
        subValueFunctionaleInCenters[optimalTauIndex * 2] +=
valueOfSubgradientComponent.x;
        subValueFunctionaleInCenters[optimalTauIndex * 2 + 1] +=
valueOfSubgradientComponent.y;
    }
}
for (int i = 0; i < currentTauCenters.Count; i++)
{
    var valueOfFine = CalculateFine(currentTauCenters[i]);
    subValueFunctionaleInCenters[i * 2] += valueOfFine.x;
    subValueFunctionaleInCenters[i * 2 + 1] += valueOfFine.y;
}
var resultVector =
Vector<double>.Build.DenseOfArray(subValueFunctionaleInCenters);
return resultVector;
}
private (double x, double y) _calculateSubgradient(Vector<double> vector, Point point, int
index, Point currentTau_k_coords, DynamicTaskSolver dynamicTaskSolver)
{
    (double cellHalfWidth, double cellHalfHeight) indent =
(_baseConfig.GridMgr.CellWidth / 2, _baseConfig.GridMgr.CellHeight / 2);

    Func<double, double, double> fx = (x_param, y_param) =>
SolveOdeAndCalculateTimeIntegralForCenter(
    new Point(x_param, y_param),
    (currentTau_k_coords.X, currentTau_k_coords.Y),
    index,
    PartialDerivativeX);
    Func<double, double, double> fy = (x_param, y_param) =>
SolveOdeAndCalculateTimeIntegralForCenter(
    new Point(x_param, y_param),
    (currentTau_k_coords.X, currentTau_k_coords.Y),
    index,
    PartialDerivativeY);
    double integralX = MathNet.Numerics.Integrate.OnRectangle(
    fx,
    point.X - indent.cellHalfWidth, point.X + indent.cellHalfWidth,
    point.Y - indent.cellHalfHeight, point.Y + indent.cellHalfHeight,
    2);
    double integralY = MathNet.Numerics.Integrate.OnRectangle(
    fy,
    point.X - indent.cellHalfWidth, point.X + indent.cellHalfWidth,
    point.Y - indent.cellHalfHeight, point.Y + indent.cellHalfHeight,
    2);
    return (integralX, integralY);
}

```

```

    }
    private (double x, double y) CalculateFine(Point tau)
    {
        double fine_x = 0, fine_y = 0;
        double penaltyFactor = 10000;
        fine_x -= penaltyFactor * Math.Max(0, Math.Sign(0 - tau.X));
        fine_x += penaltyFactor * Math.Max(0, Math.Sign(tau.X -
_baseConfig.GridMgr.SizeX));
        fine_y -= penaltyFactor * Math.Max(0, Math.Sign(0 - tau.Y));
        fine_y += penaltyFactor * Math.Max(0, Math.Sign(tau.Y -
_baseConfig.GridMgr.SizeY));
        return (fine_x, fine_y);
    }
    public double SolveOdeAndCalculateTimeIntegralForCenter(Point gridCellPoint, (double
x, double y) tauPointForDerivative, int tauIndex, Func<Func<double, double, double>, double,
double, double> partialDerivativeFunc)
    {
        int numTotalTauCenters = _baseConfig.TauCenters.Count;
        double[] y0 = new double[numTotalTauCenters];
        for (int i = 0; i < numTotalTauCenters; i++)
        {
            y0[i] = _taskSolverInstance._odeSystem == null
                ? _baseConfig.InitialConditionProviders[i].GetInitialValue(gridCellPoint,
_baseConfig.TauCenters[i])
                :
            _taskSolverInstance._config.InitialConditionProviders[i].GetInitialValue(gridCellPoint,
            _taskSolverInstance._config.TauCenters[i]);
        }
        List<double[]> yValuesOverTime = _odeSolver.Solve(_taskSolverInstance._odeSystem,
y0, 0, _T, _Nt);
        double p_x = _baseConfig.DensityFunction.GetValue(gridCellPoint);
        double partialDerivativeOfDistance = partialDerivativeFunc(
            (tauX, tauY) => Point.Distance(gridCellPoint, new Point(tauX, tauY)),
            tauPointForDerivative.x,
            tauPointForDerivative.y
        );
        double[] integrandValues = new double[_Nt + 1];
        for (int k = 0; k <= _Nt; k++)
        {
            double c_k_t = yValuesOverTime[k][tauIndex];
            integrandValues[k] = (c_k_t * partialDerivativeOfDistance) * p_x;
        }
        double timeIntegral = 0;
        double h_time = _Nt > 0 ? _T / _Nt : _T;
        if (_Nt > 0)
        {
            timeIntegral = (integrandValues[0] + integrandValues[_Nt]) / 2.0;
            for (int k = 1; k < _Nt; k++)
            {
                timeIntegral += integrandValues[k];
            }
            timeIntegral *= h_time;
        }
    }

```

```

    }
    else if (_Nt == 0 && _T > 0)
    {
        timeIntegral = integrandValues.Length > 0 ? integrandValues[0] * _T : 0;
    }
    else
    {
        timeIntegral = 0;
    }
    return timeIntegral;
}
public double CalculateNorm(Vector<double> vector)
{
    double norm = vector.L2Norm();
    return norm;
}
public static double PartialDerivativeX(Func<double, double, double> f, double x, double
y)
{
    double h = 1e-5;
    double result = (f(x + h, y) - f(x - h, y)) / (2 * h);
    return result;
}
public static double PartialDerivativeY(Func<double, double, double> f, double x, double
y)
{
    double h = 1e-5;
    double result = (f(x, y + h) - f(x, y - h)) / (2 * h);
    return result;
}
}
public class DynamicTaskSolver
{
    public readonly ProblemConfig _config;
    public readonly IOdeSystem _odeSystem;
    private readonly IOdeSolver _odeSolver;
    private readonly double _T;
    private readonly int _Nt;
    public DynamicTaskSolver(ProblemConfig config, IOdeSolver odeSolver, double T, int Nt,
Serilog.ILogger logger)
    {
        _config = config;
        _odeSolver = odeSolver;
        _T = T;
        _Nt = Nt;
        _odeSystem = new ProblemOdeSystem(config.MatrixA, null, config.DParametersForEq,
Serilog.Log.Logger);
    }
    public double SolveOdeAndCalculateTimeIntegralForCenter(Point gridCellPoint, int
tauCenterIndex)
    {
        int numTauCenters = _config.TauCenters.Count;

```

```

double[] y0 = new double[numTauCenters];
for (int i = 0; i < numTauCenters; i++)
{
    y0[i] = _config.InitialConditionProviders[i].GetInitialValue(gridCellPoint,
_config.TauCenters[i]);
}
List<double[]> yValuesOverTime = _odeSolver.Solve(_odeSystem, y0, 0, _T, _Nt);
double p_x = _config.DensityFunction.GetValue(gridCellPoint);
double distanceMetric = Point.Distance(gridCellPoint,
_config.TauCenters[tauCenterIndex]);
double significance = _config.SignificanceFactors[tauCenterIndex];
double[] integrandValues = new double[_Nt + 1];
for (int k = 0; k <= _Nt; k++)
{
    double c_j_t = yValuesOverTime[k][tauCenterIndex];
    integrandValues[k] = (c_j_t * distanceMetric + significance) * p_x;
}
double timeIntegral = 0;
double h_time = _T / _Nt;
if (_Nt > 0)
{
    timeIntegral = (integrandValues[0] + integrandValues[_Nt]) / 2.0;
    for (int k = 1; k < _Nt; k++)
    {
        timeIntegral += integrandValues[k];
    }
    timeIntegral *= h_time;
}
else if (_Nt == 0 && _T > 0)
{
    timeIntegral = integrandValues[0] * _T;
}
else
{
    timeIntegral = 0;
}
return timeIntegral;
}

```

```

public int DetermineCharacteristicFunction(Point gridCellPoint)
{
    int bestTauIndex = 0;
    double minVal = SolveOdeAndCalculateTimeIntegralForCenter(gridCellPoint, 0);
    for (int i = 1; i < _config.TauCenters.Count; i++)
    {
        double currentVal = SolveOdeAndCalculateTimeIntegralForCenter(gridCellPoint, i);
        if (currentVal < minVal)
        {
            minVal = currentVal;
            bestTauIndex = i;
        }
    }
}

```

```

        return bestTauIndex;
    }
    private double IntegrateOverDomainCell(Point cellCenterPoint, int
optimalTauCenterIndex)
    {
        double valueAtCellCenter =
SolveOdeAndCalculateTimeIntegralForCenter(cellCenterPoint, optimalTauCenterIndex);
        double cellArea = _config.GridMgr.CellArea;
        double integralContribution = valueAtCellCenter * cellArea;
        return integralContribution;
    }
    public (double functionalValue, int[,] characteristicMap) CalculateFunctional()
    {
        double totalFunctionalValue = 0;
        int[,] lambdaMap = new int[_config.GridMgr.GridYCount,
_config.GridMgr.GridXCount];
        long totalCells = (long)_config.GridMgr.GridXCount * _config.GridMgr.GridYCount;
        long processedCells = 0;
        for (int j = 0; j < _config.GridMgr.GridYCount; j++)
        {
            for (int i = 0; i < _config.GridMgr.GridXCount; i++)
            {
                int python_i = i;
                int python_j = _config.GridMgr.GridYCount - 1 - j;
                Point cellCenter = _config.GridMgr.GetCellCenter(python_i, python_j);
                int optimalTauIndex = DetermineCharacteristicFunction(cellCenter);
                lambdaMap[j, i] = optimalTauIndex;
                double cellContribution = IntegrateOverDomainCell(cellCenter, optimalTauIndex);
                totalFunctionalValue += cellContribution;
                processedCells++;
            }
        }
        return (totalFunctionalValue, lambdaMap);
    }
}
public class ProblemConfig
{
    public List<double> SignificanceFactors { get; }
    public List<Point> TauCenters { get; }
    public IDensityFunction DensityFunction { get; }
    public GridManager GridMgr { get; set; }
    public double[,] MatrixA { get; }
    public double[] DParametersForEq { get; }
    public List<IInitialConditionProvider> InitialConditionProviders { get; }

    public ProblemConfig(
        List<double> significanceFactors, List<Point> tauCenters,
        IDensityFunction densityFunction, GridManager gridMgr,
        double[,] matrixA, double[] dParams,
        List<IInitialConditionProvider> initialConditionProviders,
        Serilog.ILogger logger)
    {

```

```

        if (significanceFactors.Count != tauCenters.Count ||
            dParams.Length != tauCenters.Count ||
            initialConditionProviders.Count != tauCenters.Count ||
            matrixA.GetLength(0) != tauCenters.Count ||
            matrixA.GetLength(1) != tauCenters.Count)
        {
            string errorMsg = $"Несоответствие размерностей в параметрах ProblemConfig.
TauCenters.Count={tauCenters.Count}, " +
                $"SignificanceFactors.Count={significanceFactors.Count},
DParameters.Length={dParams.Length}, " +
                $"InitialConditionProviders.Count={initialConditionProviders.Count},
MatrixA.Dims=({matrixA.GetLength(0)},{matrixA.GetLength(1)})";
            throw new ArgumentException(errorMsg);
        }
        SignificanceFactors = significanceFactors;
        TauCenters = tauCenters;
        DensityFunction = densityFunction;
        GridMgr = gridMgr;
        MatrixA = matrixA;
        DParametersForEq = dParams;
        InitialConditionProviders = initialConditionProviders;
    }
    public ProblemConfig(
        List<double> significanceFactors, List<Point> tauCenters,
        IDensityFunction densityFunction, GridManager gridMgr,
        double[,] matrixA, double[] dParams,
        List<IInitialConditionProvider> initialConditionProviders)
        : this(significanceFactors, tauCenters, densityFunction, gridMgr, matrixA, dParams,
initialConditionProviders, Serilog.Log.Logger)
    {
    }
}
public class ExponentialPriceModel : IPriceModel
{
    private readonly double _dParameter;
    public ExponentialPriceModel(double dParameter, Serilog.ILogger logger)
    {
        _dParameter = dParameter;
    }
    public ExponentialPriceModel(double dParameter) : this(dParameter, Serilog.Log.Logger)
    {
    }
    public double CalculateValue(double c_j)
    {
        double value = _dParameter * c_j;
        return value;
    }
}
public interface IPriceModel
{
    double CalculateValue(double c_j);
}

```

```

public class PriceRegulationOneSidedModel : IPriceModel
{
    private readonly double _d;
    private readonly double _K;
    public PriceRegulationOneSidedModel(double d, double K, Serilog.ILogger logger)
    {
        _d = d; _K = K;
    }
    public PriceRegulationOneSidedModel(double d, double K) : this(d, K,
Serilog.Log.Logger)
    {
    }
    public double CalculateValue(double c_j)
    {
        double value = _d * c_j * (1 - c_j / _K);
        return value;
    }
}
public class ConstantInitialCondition : IInitialConditionProvider
{
    private readonly double _value;
    public ConstantInitialCondition(double value, Serilog.ILogger logger)
    {
        _value = value;
    }
    public ConstantInitialCondition(double value) : this(value, Serilog.Log.Logger)
    {
    }
    public double GetInitialValue(Point gridPoint, Point tauCenter)
    {
        return _value;
    }
}
public class DistanceInitialCondition : IInitialConditionProvider
{
    public DistanceInitialCondition(Serilog.ILogger logger)
    {
    }
    public DistanceInitialCondition() : this(Serilog.Log.Logger)
    {
    }
    public double GetInitialValue(Point gridPoint, Point tauCenter)
    {
        double value = Point.Distance(gridPoint, tauCenter);
        return value;
    }
}
public interface IInitialConditionProvider
{
    double GetInitialValue(Point gridPoint, Point tauCenter);
}
public interface IDensityFunction

```

```
{
    double GetValue(Point p);
}
public class UniformDensityFunction : IDensityFunction
{
    public UniformDensityFunction(Serilog.ILogger logger)
    {
    }
    public UniformDensityFunction() : this(Serilog.Log.Logger)
    {
    }
    public double GetValue(Point p)
    {
        double value = 1.0;
        return value;
    }
}
}
```

## Додаток Б. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛЬТЕРНАТИВНОЇ СИСТЕМИ ОПОВІЩЕННЯ “СВІТЛО”

### Б.1. Компонент відображення мапи з кореспондентами.

```

import {Box, Button, CircularProgress, FormControl, MenuItem, Select, Typography} from
 '@material-ui/core';
import DirectionsWalkIcon from '@material-ui/icons/DirectionsWalk';
import GoogleMapReact from 'google-map-react';
import React, {useCallback, useEffect, useMemo, useRef, useState} from 'react';
import {crudUpdate, useGetList} from 'react-admin';
import {useDispatch, useSelector} from 'react-redux';
import useSupercluster from 'use-supercluster';
import './PeopleMap.css';
const Marker = ({children}) => (children);
const options = {
  radius: 75,
  maxZoom: 20,
  map: (props) => ({
    ...props,
    numberOfActive: Number(props.isActive),
  }),
  reduce: (accumulated, props) => {
    accumulated.numberOfActive += props.numberOfActive;
  },
};
const PeopleMap = () => {
  const mapRef = useRef();
  const mapsRef = useRef();
  const circlesRef = useRef([]);
  const dispatch = useDispatch();
  const [bounds, setBounds] = useState(null);
  const [zoom, setZoom] = useState(10);
  const [limit, setLimit] = useState(10);
  const threshold = (limit >= 200 || !limit);
  const lastSocketUpdate = useSelector((state) => state.socket.lastSocketUpdate);
  const {data, loading, refetch} = useGetList(
    'people',
    {page: 0, perPage: limit || undefined},
    {field: 'id', order: 'ASC'},
  );
  const dataArray = useMemo(() => data ? Object.values(data) : [], [data]);
  useEffect(() => {
    if (lastSocketUpdate === -1) {
      return;
    }
    refetch();
  }, [lastSocketUpdate]);
  useEffect(() => {
    dispatch({type: 'START_CHANNEL'});
  }, []);

```

```

const points = useMemo(() => {
  if (loading || !dataArray.length) {
    return [];
  }
  return dataArray.map((el) => ({
    type: 'Feature',
    properties: {
      cluster: false,
      id: el.id,
      date: new Date(el.lastLocationUpdate).getTime(),
      isActive: el.isActive,
    },
    geometry: {
      type: 'Point',
      coordinates: [
        parseFloat(el.point.longitude),
        parseFloat(el.point.latitude),
      ],
    },
  }));
}, [dataArray, loading]);
const {clusters, supercluster} = useSupercluster({
  points,
  bounds,
  zoom,
  options,
});
useEffect(() => {
  if (!mapsRef.current || !mapsRef.current) {
    return;
  }
  clusters.forEach((cluster) => {
    const [longitude, latitude] = cluster.geometry.coordinates;
    const circle = new mapsRef.current.Circle({
      strokeColor: 'black',
      strokeOpacity: 0.5,
      strokeWeight: 2,
      fillColor: 'green',
      fillOpacity: 0.1,
      map: mapRef.current,
      center: new mapsRef.current.LatLng({
        lat: latitude,
        lng: longitude,
      }),
      radius: 5000,
    });
    circlesRef.current.push(circle);
  });
  return () => {
    circlesRef.current.forEach((circle) => {
      circle.setMap(null);
    });
  };
});

```

```

    };
    }, [clusters, mapRef.current]);

    if (loading && !dataArray.length) {
      return (
        <div style={{display: 'flex', alignItems: 'center', justifyContent: 'center'}}>
          <CircularProgress size={40} />
        </div>
      );
    }
    return (
      <Box display="flex" flexDirection="column">
        <Box>
          <FormControl>
            <Typography>Максимальна кіл-ть людей</Typography>
            <Select
              value={limit}
              onChange={(event) => {
                setLimit(event.target.value);
              }}
              displayEmpty
            >
              <MenuItem value="">Усі</MenuItem>
              <MenuItem value={10}>10</MenuItem>
              <MenuItem value={25}>25</MenuItem>
              <MenuItem value={50}>50</MenuItem>
              <MenuItem value={100}>100</MenuItem>
              <MenuItem value={200}>200</MenuItem>
              <MenuItem value={500}>500</MenuItem>
              <MenuItem value={1000}>1000</MenuItem>
              <MenuItem value={5000}>5000</MenuItem>
              <MenuItem value={10000}>10000</MenuItem>
            </Select>
          </FormControl>
        </Box>
        {threshold && (
          <Box display="flex" flexDirection="column" paddingBottom={1}>
            <Typography>Обрані фільтри натякають на завантаження великого обсягу
данях,</Typography>
            <Typography>будь ласка, не залишайте обрані фільтри на довгий час для запобігання
навантаження на
сервер</Typography>
          </Box>
        )}
      <Box style={{height: '82.5vh', width: '100%', padding: 1}}>
        <GoogleMapReact
          bootstrapURLKeys={{key: 'AIzaSyBfhsYSTvx_5V3ZU2DRiNueMqly6TcxBOY'}}
          defaultCenter={{
            lat: points.length ? points[points.length - 1].geometry.coordinates[1] : 50.477311,
            lng: points.length ? points[points.length - 1].geometry.coordinates[0] : 30.461800,
          }}
          defaultZoom={11}
        />
      </Box>
    );
  }
}

```

```

yesIWantToUseGoogleMapApiInternals
onGoogleApiLoaded={({map, maps}) => {
  mapRef.current = map;
  mapsRef.current = maps;
}}
onChange={({zoom, bounds}) => {
  setZoom(zoom);
  setBounds([
    bounds.nw.lng,
    bounds.se.lat,
    bounds.se.lng,
    bounds.nw.lat,
  ]);
}}
>
{clusters.map(cluster => {
  const [longitude, latitude] = cluster.geometry.coordinates;
  const {
    cluster: isCluster,
    point_count: pointCount,
    numberOfActive,
  } = cluster.properties;

  if (isCluster) {
    const resultCount = numberOfActive > 0 ?
      `${numberOfActive}/${pointCount}` :
      `${numberOfActive}/${pointCount}`;
    return (
      <Marker
        key={`cluster-${cluster.id}`}
        lat={latitude}
        lng={longitude}
      >
      <div
        className="cluster-marker-p"
        style={{
          width: 10 + (pointCount / points.length) * 20,
          height: 10 + (pointCount / points.length) * 20,
          marginLeft: -(15 + (pointCount / points.length) * 20) / 2,
          marginTop: -(20 + (pointCount / points.length) * 20) / 2,
        }}
        onClick={() => {
          const expansionZoom = Math.min(
            supercluster.getClusterExpansionZoom(cluster.id),
            20,
          );
          mapRef.current.setZoom(expansionZoom);
          mapRef.current.panTo({lat: latitude, lng: longitude});
        }}
      >
      {pointCount}
    </div>

```

```

    </Marker>
  );
}
return (
  <Marker
    key={`cluster-${cluster.properties.id}`}
    lat={latitude}
    lng={longitude}
  >
    <div
      className="cluster-marker-p-one"
      style={{
        marginLeft: -5,
        marginTop: -5,
      }}
    />
  </Marker>
);
}}
</GoogleMapReact>
</Box>
</Box>
);
};
export default PeopleMap;

```

## Б.2. Компонент відображення мапи з маркерами та траєкторіями.

```

import {CircularProgress, MenuItem, Select, Box, FormControl, Typography, Button} from
 '@material-ui/core';
import GoogleMapReact from 'google-map-react';
import React, {useCallback, useEffect, useMemo, useRef, useState} from 'react';
import {useGetList, crudUpdate} from 'react-admin';
import {useDispatch, useSelector} from 'react-redux';
import useSupercluster from 'use-supercluster';
import './Map.css';
import audioFile from './119302760.mp3';
const Marker = ({children}) => children;
const options = {
  radius: 75,
  maxZoom: 20,
};
const MapWithMarkers = () => {
  const mapRef = useRef();
  const dispatch = useDispatch();
  const [bounds, setBounds] = useState(null);
  const [zoom, setZoom] = useState(10);
  const [minutes, setMinutes] = useState(Number(localStorage.getItem('MINUTES_LOCAL')) ||
  30);
  const [limit, setLimit] = useState(10);
  const threshold = (limit >= 200 || limit === '') || (minutes > 1440 || minutes === '');
  const {lastSocketUpdate, sound} = useSelector((state) => state.socket);
  const {data, loading, refetch} = useGetList(

```

```

    'data',
    {page: 0, perPage: limit},
    {field: 'id', order: 'ASC'},
    {minutes},
  );
  const dataArray = useMemo(() => data ? Object.values(data) : [], [data]);
  useEffect(() => {
    if (lastSocketUpdate === -1) {
      return;
    }
    refetch();
    if (!sound) {
      return;
    }
    const audio = new Audio(audioFile);
    audio.play();
  }, [lastSocketUpdate]);
  useEffect(() => {
    return () => {
      dispatch({type: 'RESET_UPDATE'});
    };
  }, []);
  useEffect(() => {
    dispatch({type: 'START_CHANNEL'});
  }, []);
  console.log(dataArray);
  const points = useMemo(() => {
    if (loading || !dataArray.length) {
      return [];
    }
    return dataArray.map((el) => ({
      type: 'Feature',
      properties: {
        cluster: false,
        id: el.id,
        date: new Date(el.time).getTime(),
      },
      geometry: {
        type: 'Point',
        coordinates: [
          parseFloat(el.point.longitude),
          parseFloat(el.point.latitude),
        ],
      },
    }));
  }, [dataArray, loading]);
  const {clusters, supercluster} = useSupercluster({
    points,
    bounds,
    zoom,
    options,
  });

```

```

const handleHide = useCallback(() => {
  if (!dataArray.length) {
    return;
  }
  const ids = dataArray.map((el) => el.id);
  dispatch(crudUpdate('data', 'hide', {points: ids}, {}, '/data', false));
}, [dataArray]);
if (loading) {
  return (
    <div style={{display: 'flex', alignItems: 'center', justifyContent: 'center'}}>
      <CircularProgress size={40} />
    </div>
  );
}
return (
  <Box display="flex" flexDirection="column">
    <Box display="flex" paddingY={1} gridGap={8} alignItems="center"
justifyContent="space-between">
      <Box display="flex" gridGap={8}>
        <Box>
          <FormControl>
            <Typography>Період</Typography>
            <Select
              value={minutes}
              onChange={(event) => {
                setMinutes(event.target.value);
                localStorage.setItem('MINUTES_LOCAL', event.target.value);
              }}
              displayEmpty
            >
              <MenuItem value="">Весь час</MenuItem>
              <MenuItem value={10}>Останні десять хвилин</MenuItem>
              <MenuItem value={15}>Останні п'ятнадцять хвилин</MenuItem>
              <MenuItem value={30}>Останні тридцять хвилин</MenuItem>
              <MenuItem value={60}>Остання година</MenuItem>
              <MenuItem value={120}>Останні дві години</MenuItem>
              <MenuItem value={1440}>Остання доба</MenuItem>
              <MenuItem value={2880}>Останні 2 доби</MenuItem>
              <MenuItem value={7200}>Останні 5 діб</MenuItem>
              <MenuItem value={10080}>Останній тиждень</MenuItem>
            </Select>
          </FormControl>
        </Box>
        <Box>
          <FormControl>
            <Typography>Максимальна кіл-ть точок</Typography>
            <Select
              value={limit}
              onChange={(event) => {
                setLimit(event.target.value);
              }}
              displayEmpty
            >

```

```

    >
    <MenuItem value="">Усі</MenuItem>
    <MenuItem value={10}>10</MenuItem>
    <MenuItem value={25}>25</MenuItem>
    <MenuItem value={50}>50</MenuItem>
    <MenuItem value={100}>100</MenuItem>
    <MenuItem value={200}>200</MenuItem>
    <MenuItem value={500}>500</MenuItem>
    <MenuItem value={1000}>1000</MenuItem>
    <MenuItem value={5000}>5000</MenuItem>
    <MenuItem value={10000}>10000</MenuItem>
  </Select>
</FormControl>
</Box>
</Box>
<Box>
  <Button
    variant="contained"
    onClick={handleHide}
    disabled={!dataArray.length}
  >
    Вимкнути світло
  </Button>
</Box>
</Box>
{threshold && (
  <Box display="flex" flexDirection="column" paddingBottom={1}>
    <Typography>Обрані фільтри натякають на завантаження великого обсягу
данях,</Typography>
    <Typography>будь ласка, не залишайте обрані фільтри на довгий час для запобігання
навантаження на
сервер</Typography>
  </Box>
)}
<Box style={{height: '82.5vh', width: '100%', padding: 1}}>
  <GoogleMapReact
    bootstrapURLKeys={{key: 'AIzaSyBfhsYSTvx_5V3ZU2DRiNueMqly6TcxB0Y'}}
    defaultCenter={{
      lat: points.length ? points[points.length - 1].geometry.coordinates[1] : 48.450001,
      lng: points.length ? points[points.length - 1].geometry.coordinates[0] : 34.983334,
    }}
    defaultZoom={11}
    yesIWantToUseGoogleMapApiInternals
    onGoogleApiLoaded={({map}) => {
      mapRef.current = map;
    }}
    onChange={({zoom, bounds}) => {
      setZoom(zoom);
      setBounds([
        bounds.nw.lng,
        bounds.se.lat,
        bounds.se.lng,

```

```

    bounds.nw.lat,
  ]);
}}
>
{clusters.map(cluster => {
  const [longitude, latitude] = cluster.geometry.coordinates;
  const {
    cluster: isCluster,
    point_count: pointCount,
  } = cluster.properties;
  if (isCluster) {
    return (
      <Marker
        key={`cluster-${cluster.id}`}
        lat={latitude}
        lng={longitude}
      >
      <div
        className="cluster-marker"
        style={{
          width: `${10 + (pointCount / points.length) * 20}px`,
          height: `${10 + (pointCount / points.length) * 20}px`,
        }}
        onClick={() => {
          const expansionZoom = Math.min(
            supercluster.getClusterExpansionZoom(cluster.id),
            20,
          );
          mapRef.current.setZoom(expansionZoom);
          mapRef.current.panTo({lat: latitude, lng: longitude});
        }}
      >
        {pointCount}
      </div>
    </Marker>
  );
}
return (
  <Marker
    key={`cluster-${cluster.properties.id}`}
    lat={latitude}
    lng={longitude}
  >
  <div
    className="cluster-marker-one"
    style={{
      width: `${20}px`,
      height: `${20}px`,
    }}
  >
    Світло!
  </div>

```

```

    </Marker>
  );
  })}
  </GoogleMapReact>
</Box>
</Box>
);
};
export default MapWithMarkers;

```

### Б.3. Обробка запитів.

```

import {stringify} from 'query-string';
import {fetchUtils} from 'ra-core';
const httpClientWithToken = (url, options = {}) => {
  const token = localStorage.getItem('token');
  options.user = {
    authenticated: true,
    token,
  };
  return fetchUtils.fetchJson(url, options);
};
const _reKeyResponse = (json, key) => {
  if (key === null) {
    return json;
  } else if (Array.isArray(json)) {
    return json.map((x) => {
      x.id = x[key];
      delete x[key];
      return x;
    });
  } else if (json[key] !== null) {
    json.id = json[key];
    delete json[key];
    return json;
  } else {
    console.error('undhandled scenario of _reKeyResponse', key, json);
  }
};
const _xFormResponse = (json, xFormFn) => {
  if (!xFormFn) {
    return json;
  }
  if (json === null) {
    return json;
  } else if (Array.isArray(json)) {
    return json.map(xFormFn);
  } else {
    return xFormFn(json);
  }
};
const _reKeyPayload = (data, reParam) => {
  if (reParam === null) {

```

```

    return data;
  }

  var {id, ...others} = data;
  others[reParam] = data.id;
  return others;
};
const _reKeyFilter = (filter, reParam) => {
  if (filter === null) {
    return null;
  }
  var {id, ...reFilter} = filter;
  if (reParam !== null && id) {
    reFilter[reParam] = id;
  }
  return reFilter;
};
export default (
  apiUrl,
  keysByResource = {} /* ex: {'posts':'key',...} */,
  responseTransformsByResource = {},
  httpClient = httpClientWithToken,
  countHeader = 'X-Total-Count',
) => ({
  keysByResource: keysByResource,
  xFormBy: responseTransformsByResource,
  getList: (resource, params) => {
    const reParam =
      resource in keysByResource ? keysByResource[resource] : null;
    const {page, perPage} = params.pagination;
    let {field, order} = params.sort;
    const reFilter = _reKeyFilter(params.filter, reParam);
    if (reParam !== null && field === 'id') {
      field = reParam;
    }
    const rangeStart = (page - 1) * perPage;
    const rangeEnd = page * perPage - 1;
    const query = {
      sort: JSON.stringify([field, order]),
      limit: perPage,
      skip: page * perPage,
      filter: JSON.stringify(reFilter),
    };
    const url = `${apiUrl}/${resource}?${stringify(query)}`;
    const options =
      countHeader === 'Content-Range'
      ? {
        // Chrome doesn't return `Content-Range` header if no `Range` is provided in the request.
        headers: new Headers({
          Range: `${resource}=${rangeStart}-${rangeEnd}`,
        }),
      }
      : {}
  }
});

```

```

: {});

return httpClient(url, options).then(({headers, json}) => {
  if (!headers.has(countHeader)) {
    throw new Error(
      `The ${countHeader} header is missing in the HTTP Response. The simple REST data
      provider expects responses for lists of resources to contain this header with the total number of
      results to build the pagination. If you are using CORS, did you declare ${countHeader} in the
      Access-Control-Expose-Headers header?`,
    );
  }
  return {
    data: _xFormResponse(
      _reKeyResponse(json, reParam),
      responseTransformsByResource[resource],
    ),
    total:
      countHeader === 'Content-Range'
        ? parseInt(headers.get('content-range').split('/').pop(), 10)
        : parseInt(headers.get(countHeader.toLowerCase()),
  );
});
},
getOne: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;

  return httpClient(`${apiUrl}/${resource}/${params.id}`).then(
    ({json}) => ({
      data: _xFormResponse(
        _reKeyResponse(json, reParam),
        responseTransformsByResource[resource],
      ),
    })),
  );
},
getMany: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  var q = {};
  q[reParam ?? 'id'] = params.ids;
  const query = {
    filter: JSON.stringify(q),
  };
  const url = `${apiUrl}/${resource}?${stringify(query)}`;
  return httpClient(url).then(({json}) => ({
    data: _xFormResponse(
      _reKeyResponse(json, reParam),
      responseTransformsByResource[resource],
    ),
  }));
},

```

```

getManyReference: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  const {page, perPage} = params.pagination;
  var {field, order} = params.sort;
  const reFilter = _reKeyFilter(params.filter, reParam);
  if (reParam !== null && field === 'id') {
    field = reParam;
  }
  const rangeStart = (page - 1) * perPage;
  const rangeEnd = page * perPage - 1;
  const query = {
    sort: JSON.stringify([field, order]),
    limit: perPage,
    skip: page * perPage,
    filter: JSON.stringify({
      ...reFilter,
      [params.target]: params.id,
    }),
  };
  const url = `${apiUrl}/${resource}?${stringify(query)}`;
  const options =
    countHeader === 'Content-Range'
    ? {
      // Chrome doesn't return `Content-Range` header if no `Range` is provided in the request.
      headers: new Headers({
        Range: `${resource}=${rangeStart}-${rangeEnd}`,
      }),
    }
    : {};
  return httpClient(url, options).then(({headers, json}) => {
    if (!headers.has(countHeader)) {
      throw new Error(
        `The ${countHeader} header is missing in the HTTP Response. The simple REST data
        provider expects responses for lists of resources to contain this header with the total number of
        results to build the pagination. If you are using CORS, did you declare ${countHeader} in the
        Access-Control-Expose-Headers header?`,
      );
    }
    return {
      data: _xFormResponse(
        _reKeyResponse(json, reParam),
        responseTransformsByResource[resource],
      ),
      total:
        countHeader === 'Content-Range'
        ? parseInt(headers.get('content-range').split('/').pop(), 10)
        : parseInt(headers.get(countHeader.toLowerCase()),
    );
  });
},
update: (resource, params) => {

```

```

const reParam =
  resource in keysByResource ? keysByResource[resource] : null;
const reData = _reKeyPayload(params.data, reParam);
return httpClient(`${apiUrl}/${resource}/${params.id}`, {
  method: 'PUT',
  body: JSON.stringify(reData),
}).then(({json}) => ({
  data: _xFormResponse(
    _reKeyResponse(json, reParam),
    responseTransformsByResource[resource],
  ),
}));
},
// simple-rest doesn't handle provide an updateMany route, so we fallback to calling update n
times instead
updateMany: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  const idKey = reParam ?? 'id';
  return Promise.all(
    params.ids.map((id) => {
      const reData = _reKeyPayload(params.data, reParam);
      return httpClient(`${apiUrl}/${resource}/${id}`, {
        method: 'PUT',
        body: JSON.stringify(reData),
      });
    })
  ).then((responses) => ({data: responses.map(({json}) => json[idKey])}));
},
create: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  const reData = _reKeyPayload(params.data, reParam);
  const idKey = reParam ?? 'id';
  return httpClient(`${apiUrl}/${resource}`, {
    method: 'POST',
    body: JSON.stringify(reData),
  }).then(({json}) => ({
    data: {...params.data, id: json[idKey]},
  }));
},
delete: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  return httpClient(`${apiUrl}/${resource}/${params.id}`, {
    method: 'DELETE',
  }).then(({json}) => ({
    data: _xFormResponse(
      _reKeyResponse(json, reParam),
      responseTransformsByResource[resource],
    ),
  }));
}

```

```

},

// simple-rest doesn't handle filters on DELETE route, so we fallback to calling DELETE n
times instead
deleteMany: (resource, params) => {
  const reParam =
    resource in keysByResource ? keysByResource[resource] : null;
  const idKey = reParam ?? 'id';
  return Promise.all(
    params.ids.map((id) =>
      httpClient(`${apiUrl}/${resource}/${id}`, {
        method: 'DELETE',
      })
    ),
  )
  .then((responses) => ({data: responses.map(({json}) => json[idKey])}));
},
});

```

#### Б.4. Перевірка авторизації користувача.

```

import { Test, TestingModule } from '@nestjs/testing';
import { AuthController } from './auth.controller';
import { AuthService } from './auth.service';
describe('AuthController', () => {
  let controller: AuthController;
  beforeEach(async () => {
    const module: TestingModule = await Test.createTestingModule({
      controllers: [AuthController],
      providers: [
        {
          provide: AuthService,
          useValue: {},
        },
      ],
    }).compile();
    controller = module.get<AuthController>(AuthController);
  });
  it('should be defined', () => {
    expect(controller).toBeDefined();
  });
});

```

#### Б.5. Структура даних маркерів.

```

import { ApiProperty } from '@nestjs/swagger';
import { BelongsTo, Column, DataType, ForeignKey, Model, Sequelize, Table } from
"sequelize-typescript";
import { ILocation } from '../types';
import { TrackingUser } from ".";
export class IDataModelAttributes {
  @ApiProperty({ readOnly: true })
  id?: string;
  @ApiProperty({ type: ILocation })
  location: ILocation;
}

```

```

@ApiProperty()
time: Date;

@ApiProperty()
isHidden?: boolean;
@ApiProperty()
trackingUserId: string;
}
export class IDataCreateAttributes {
  @ApiProperty({ readOnly: true })
  id?: string;
  @ApiProperty({ type: ILocation })
  location?: ILocation;
  @ApiProperty()
  time?: Date;
  @ApiProperty()
  isHidden?: boolean;
  @ApiProperty()
  trackingUserId?: string;
}
@Table({
  timestamps: false,
  tableName: 'data',
  freezeTableName: true,
})
export class Data extends Model<IDataModelAttribute, IDataCreateAttributes> {
  @ApiProperty({ readOnly: true })
  @Column({
    type: DataType.INTEGER,
    unique: true,
    autoIncrement: true,
    primaryKey: true,
    allowNull: false,
  })
  readonly id: number;
  @ApiProperty({
    type: ILocation,
  })
  @Column({
    type: DataType.GEOGRAPHY,
    allowNull: true,
  })
  location: ILocation;
  @ApiProperty()
  @Column({
    type: DataType.DATE,
    allowNull: true,
    defaultValue: Sequelize.fn('now'),
  })
  time: Date;
  @ApiProperty()
  @Column({

```

```

    type: DataType.BOOLEAN,
    allowNull: false,
    defaultValue: false,
  })
  isHidden: boolean;
  @ForeignKey(() => TrackingUser)
  @ApiProperty()
  @Column({
    type: DataType.TEXT,
    allowNull: false,
  })
  trackingUserId: string;

  @BelongsTo(() => TrackingUser)
  trackingUser: TrackingUser;
}

```

## Б.6. Структура профілю користувача.

```

import { ApiHideProperty, ApiProperty } from '@nestjs/swagger';
import { BeforeBulkUpdate, BeforeCreate, BeforeUpdate, Column, DataType, Model, Table }
from 'sequelize-typescript';
import { ILocation } from '../types';
import { PasswordService } from '../utils/password/password.service';
export class IUserModelAttributes {
  @ApiProperty({ readOnly: true })
  id?: string;
  @ApiProperty()
  username?: string;
  @ApiProperty()
  firstName?: string;
}
export class IUserCreateAttributes {
  @ApiProperty({ readOnly: true })
  id?: string;
  @ApiProperty()
  username?: string;
  @ApiProperty()
  firstName?: string;
  @ApiProperty()
  password: string;
}
@Table({
  timestamps: false,
  tableName: 'user',
  freezeTableName: true,
})
export class User extends Model<IUserModelAttributes, IUserCreateAttributes> {
  @ApiProperty({ readOnly: true })
  @Column({
    type: DataType.UUID,
    unique: true,
    primaryKey: true,

```

```

    allowNull: false,
    defaultValue: DataType.UUIDV4,
  })
  readonly id: string;
  @ApiProperty()
  @Column({
    type: DataType.TEXT,
    allowNull: true,
    unique: {
      msg: 'User with current username already exists',
      name: 'username',
    },
  })
  username: string;
  @ApiProperty()
  @Column({
    type: DataType.TEXT,
    allowNull: true,
    field: 'first_name',
  })
  firstName: string;
  @ApiHideProperty()
  @Column({
    type: DataType.TEXT,
    allowNull: false,
    get: function () {
      return () => this.getDataValue('password');
    },
  })
  password: string;
  @BeforeUpdate
  @BeforeCreate
  static async beforeUpdateAndCreateHook(user: User) {
    if (user.changed('password')) {
      const passwordGetter = user.password as unknown as (() => string);
      user.password = await PasswordService.hashPassword(passwordGetter());
    }
  }
  @BeforeBulkUpdate
  static async beforeBulkUpdateHook(options) {
    if (options.attributes.password) {
      options.attributes.password = await
PasswordService.hashPassword(options.attributes.password);
    }
  }
}

```

### Б.7. Загальний модуль програми.

```

import { Module } from '@nestjs/common';
import { ConfigModule, ConfigService } from '@nestjs/config';
import { SequelizeModule } from '@nestjs/sequelize';
import { AccessTokenModule } from './access-token/access-token.module';

```

```

import { AppController } from './app.controller';
import { AuthModule } from './auth/auth.module';
import config from './config';
import { configValidationSchema } from './config/config-validation-schema';
import { CONFIG_KEY_DATABASE, ENV_TEST } from './constants';
import { DataReceiverModule } from './data-receiver/data-receiver.module';
import { NotificationsModule } from './notifications/notifications.module';
import { AccessTokenHeaderStrategy, UserStrategy, ValidateDataStrategy,
ValidateLocationStrategy } from './strategies';
import { UserModule } from './user/user.module';
import { UtilsModule } from './utils/utils.module';
@Module({
  imports: [
    ConfigModule.forRoot({
      isGlobal: true,
      load: [config],
      validationSchema: configValidationSchema,
      envFilePath: process.env.NODE_ENV === ENV_TEST ? '.env.test' : '.env',
    }),
    SequelizeModule.forRootAsync({
      imports: [ConfigModule],
      inject: [ConfigService],
      useFactory: (config: ConfigService) => config.get(CONFIG_KEY_DATABASE)(),
    }),
    UtilsModule,
    UserModule,
    AuthModule,
    AccessTokenModule,
    DataReceiverModule,
    NotificationsModule,
  ],
  controllers: [AppController],
  providers: [
    UserStrategy,
    AccessTokenHeaderStrategy,
    ValidateDataStrategy,
    ValidateLocationStrategy,
  ],
})
export class AppModule {
}

```

## Додаток В. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОБУДОВИ ОПТИМАЛЬНОЇ ТРАЄКТОРІЇ ДЛЯ ТРИВИМІРНОГО ДРУКУ

```

from stl import mesh

from mpl_toolkits import mplot3d
from matplotlib import pyplot
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QLabel, QVBoxLayout,
QFileDialog, QTabWidget, QLineEdit
from PyQt5 import QtCore
from mandoline import inside
import json
default_feed_rate = 490 / 60
default_config = {
    'layer_height': 1.2,
    'feed_rate': default_feed_rate,
    'travel_rate_xy': default_feed_rate,
    'travel_rate_z': default_feed_rate,
    'overhang_angle': 5
}
}
class WindowWidget(QWidget):
    def __init__(self):
        super().__init__()
        self.files_names = []
        self.stl_movements = {
            'x': 0,
            'y': 0,
            'z': 0
        }
        self.config = self.get_config()
        self.setFixedSize(400, 250)
        # region Elements
        self.button = QPushButton("Open new STL file")
        self.text = QLabel("Supported formats: .stl")
        self.opened_files_list = QLabel("Opened files:")
        self.button_gcode = QPushButton("Convert file(s) to trajectory")
        self.button_plot = QPushButton("View plot(s) of .stl files")
        self.layer_height_textbox = QLineEdit(self)
        self.layer_height_textbox.setPlaceholderText('Diameter of printing track mm.')
        self.layer_height_textbox.setText(str(self.config['layer_height']))
        self.feed_rate_textbox = QLineEdit(self)
        self.feed_rate_textbox.setPlaceholderText('Printing speed mm./sec.')
        self.feed_rate_textbox.setText(str(self.config['feed_rate']))
        self.overhang_angle_textbox = QLineEdit(self)
        self.overhang_angle_textbox.setPlaceholderText('Max angle between layers')
        self.overhang_angle_textbox.setText(str(self.config['overhang_angle']))
        self.apply_config_button = QPushButton("Apply and save config")
        self.stl_x_textbox = QLineEdit(self)
        self.stl_x_textbox.setPlaceholderText('Move on _ by X')
        self.stl_y_textbox = QLineEdit(self)

```

```

self.stl_y_textbox.setPlaceholderText('Move on _ by Y')
self.stl_z_textbox = QLineEdit(self)
self.stl_z_textbox.setPlaceholderText('Move on _ by Z')
self.apply_stl_movement_button = QPushButton("Move STL file")
# endregion Elements
self.initGui()
def initGui(self):
    tabs = self.createTabs()
    layout = QVBoxLayout()
    layout.addWidget(tabs)
    self.setLayout(layout)
def createTabs(self):
    # region Main page
    main_page = QWidget()
    main_page_layout = QVBoxLayout()
    main_page_layout.addWidget(self.text)
    main_page_layout.addWidget(self.button)
    main_page_layout.addWidget(self.button_gcode)
    main_page_layout.addWidget(self.button_plot)
    main_page_layout.addWidget(self.opened_files_list)
    main_page.setLayout(main_page_layout)
    self.button.clicked.connect(self.open_file)
    self.button_gcode.clicked.connect(self.create_gcode_file)
    self.button_plot.clicked.connect(self.create_plot)
    self.button_gcode.setEnabled(False)
    self.button_plot.setEnabled(False)
    self.text.setAlignment(QtCore.Qt.AlignCenter)
    # endregion Main page
    # region Config page
    config_page = QWidget()
    config_page_layout = QVBoxLayout()
    config_page_layout.addWidget(self.layer_height_textbox)
    config_page_layout.addWidget(self.feed_rate_textbox)
    config_page_layout.addWidget(self.overhang_angle_textbox)
    config_page_layout.addWidget(self.apply_config_button)
    self.apply_config_button.clicked.connect(self.apply_config)
    config_page.setLayout(config_page_layout)
    # endregion Config page
    # region STL config
    stl_config_page = QWidget()
    stl_config_layout = QVBoxLayout()
    stl_config_layout.addWidget(self.stl_x_textbox)
    stl_config_layout.addWidget(self.stl_y_textbox)
    stl_config_layout.addWidget(self.stl_z_textbox)
    stl_config_layout.addWidget(self.apply_stl_movement_button)
    self.apply_stl_movement_button.clicked.connect(self.move_stl)
    stl_config_page.setLayout(stl_config_layout)
    # endregion STL config
    tabs = QTabWidget()
    tabs.addTab(main_page, 'Main page')
    tabs.addTab(config_page, 'Config')
    tabs.addTab(stl_config_page, 'STL movement')

```

```

    return tabs
def get_config(self):
    try:
        with open("config.json") as json_data_file:
            return json.load(json_data_file)
    except FileNotFoundError:
        return default_config
def open_file(self):
    files = QFileDialog.getOpenFileName(self, 'Open file', 'stls')
    self.files_names = []
    for file in files:
        if file != "All Files (*)":
            self.files_names.append(file)
    files_str = ""
    for file_name in self.files_names:
        files_str += file_name + "\n"
    self.opened_files_list.setText("Opened files:\n" + files_str)
    self.button_gcode.setEnabled(True)
    self.button_plot.setEnabled(True)
def create_gcode_file(self):
    for file_path in self.files_names:
        inside(
            file_path,
            layer_height=self.config['layer_height'],
            feed_rate=self.config['feed_rate'],
            travel_rate_xy=self.config['travel_rate_xy'],
            travel_rate_z=self.config['travel_rate_z'],
            overhang_angle=self.config['overhang_angle']
        )
    file_path_array = file_path.split(".")
    file_path_array[-1] = "txt"
    a = ".".join(file_path_array)
    file = open(a)
    dots = file.read()
    dots_array = dots.split("\n")
    fig = pyplot.figure()
    ax = fig.add_subplot(projection='3d')
    ax.set_xlim3d(80, 120)
    ax.set_ylim3d(80, 120)
    ax.set_zlim3d(0, 40)
    dots_x = []
    dots_y = []
    dots_z = []
    for dot in dots_array:
        dot_array = dot.split(" ")
        if len(dot_array) == 4:
            dots_x.append(float(dot_array[1][1:]))
            dots_y.append(float(dot_array[2][1:]))
            dots_z.append(float(dot_array[3][1:]))
    ax.plot(xs=dots_x, ys=dots_y, zs=dots_z)
    fig25 = pyplot.figure()

```

```

ax25 = fig25.add_subplot(projection='3d')
ax25.set_xlim3d(80, 120)
ax25.set_ylim3d(80, 120)
ax25.set_zlim3d(0, 40)
dots25_x = dots_x[:int(len(dots_x)/4)]
dots25_y = dots_y[:int(len(dots_y)/4)]
dots25_z = dots_z[:int(len(dots_z)/4)]
ax25.plot(xs=dots25_x, ys=dots25_y, zs=dots25_z)
fig50 = pyplot.figure()
ax50 = fig50.add_subplot(projection='3d')
ax50.set_xlim3d(80, 120)
ax50.set_ylim3d(80, 120)
ax50.set_zlim3d(0, 40)
dots50_x = dots_x[:int(len(dots_x) / 2)]
dots50_y = dots_y[:int(len(dots_y) / 2)]
dots50_z = dots_z[:int(len(dots_z) / 2)]
ax50.plot(xs=dots50_x, ys=dots50_y, zs=dots50_z)
fig75 = pyplot.figure()
ax75 = fig75.add_subplot(projection='3d')
ax75.set_xlim3d(80, 120)
ax75.set_ylim3d(80, 120)
ax75.set_zlim3d(0, 40)
dots75_x = dots_x[:int(len(dots_x) * 0.75)]
dots75_y = dots_y[:int(len(dots_y) * 0.75)]
dots75_z = dots_z[:int(len(dots_z) * 0.75)]
ax75.plot(xs=dots75_x, ys=dots75_y, zs=dots75_z)
pyplot.show()
def apply_config(self):
    result_config = {}
    result_config['layer_height'] = float(self.layer_height_textbox.text())
    result_config['feed_rate'] = float(self.feed_rate_textbox.text())
    result_config['travel_rate_xy'] = result_config['feed_rate']
    result_config['travel_rate_z'] = result_config['feed_rate']
    result_config['overhang_angle'] = int(self.overhang_angle_textbox.text())
    with open('config.json', 'w') as f:
        json.dump(result_config, f)
    self.config = result_config
    self.layer_height_textbox.setText(str(self.config['layer_height']))
    self.feed_rate_textbox.setText(str(self.config['feed_rate']))
    self.overhang_angle_textbox.setText(str(self.config['overhang_angle']))
def create_plot(self):
    for file_path in self.files_names:
        figure = pyplot.figure()
        axes = mplot3d.Axes3D(figure)
        # Load the STL files and add the vectors to the plot
        your_mesh = mesh.Mesh.from_file(file_path)
        your_mesh.x += self.stl_movements['x']
        your_mesh.y += self.stl_movements['y']
        your_mesh.z += self.stl_movements['z']
        axes.add_collection3d(mplot3d.art3d.Poly3DCollection(your_mesh.vectors))
        # Auto scale to the mesh size
        scale = your_mesh.points.flatten()

```

```
axes.auto_scale_xyz(scale, scale, scale)
# Show the plot to the screen
figure.show()
def move_stl(self):
    self.stl_movements['x'] = int(self.stl_x_textbox.text() or '0')
    self.stl_movements['y'] = int(self.stl_y_textbox.text() or '0')
    self.stl_movements['z'] = int(self.stl_z_textbox.text() or '0')
    self.create_plot()
def main():
    app = QApplication(sys.argv)
    widget = WindowWidget()
    widget.show()
    sys.exit(app.exec_())
```