

Дніпровський національний університет імені Олеся Гончара
Міністерство освіти і науки України

Дніпровський національний університет імені Олеся Гончара
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Малієнко Ольга Олександрівна

УДК 519.854.2

ДИСЕРТАЦІЯ

ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ АНОМАЛЬНИХ ВИПАДКІВ У ЗАДАЧАХ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ ВЕРШИН ОРГРАФІВ

11 – Математика та статистика
113 – Прикладна математика

Подається на здобуття ступеня доктора філософії.

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

_____ О.О. Малієнко

Науковий керівник
Турчина Валентина Андріївна
кандидат фізико-математичних наук,
доцент

Дніпро – 2025

АНОТАЦІЯ

Малієнко О.О. Теоретичне обґрунтування аномальних випадків у задачах паралельного упорядкування вершин орграфів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 113 «Прикладна математика» (11 – Математика та статистика) – Дніпровський національний університет імені Олеся Гончара, Дніпро, 2025.

Дисертаційна робота присвячена дослідженню аномальних випадків у одному із підкласів задач дискретної оптимізації, а саме в задачах оптимального упорядкування вершин орграфів.

Постановки таких задач пов'язані з оптимальною організацією процесу розподілу скінченої множини завдань, при наявності технологічних обмежень на порядок їх слідування. Одна із задач стосується випадку, коли відома скінчена множина виконавців і необхідно мінімізувати час завершення всіх завдань без порушення технологічних обмежень. А інша полягає в мінімізації кількості виконавців при заданому часі завершення. У теорії упорядкувань вони формуються як задачі паралельного упорядкування вершин орієнтованих графів.

У класичних постановках ці задачі відносяться до класу NP -важких задач. Тому, для знаходження їх точних розв'язків застосовуються алгоритми, що базуються на схемах направленої перебору. Важливим напрямком досліджень класичних задач є розробка алгоритмів поліноміальної складності, що стосуються або певних структур графів, що задають технологічні обмеження на порядок виконання завдань, або спеціальних вимог до початкових даних.

Відомі результати поліноміально розв'язних задач стосуються випадків, коли графи технологічних обмежень відносяться до класу вхідних дерев та деяких паралельно-послідовних графів спеціального вигляду.

Також повністю досліджений випадок стосується обмеження, коли

кількість виконавців дорівнює двом.

При аналізі прикладних задач з'ясувалося, що в деяких їх узагальненнях можуть виникати непередбачувані ситуації, які можуть призводити до погіршення значення цільової функції. Такі випадки були названі аномальними.

Аномальні випадки в задачах паралельного упорядкування вперше були досліджені Рональдом Грехемом, який увів поняття аномалій та проаналізував їх вплив на ефективність результатів. У своїх дослідженнях [15] він показав, що зміна певних параметрів задачі може призводити до парадоксальних результатів, коли очікуване покращення призводить до погіршення розв'язків.

Подальші дослідження аномалій охопили аналіз у контексті систем реального часу. Так, у 1999 році Лундквіст Т. та Стенстрем у роботі [16] розглянули часові аномалії в організації роботи мікропроцесорів, показавши, що певні затримки можуть призводити до непередбачуваних ефектів у виконанні задач. Шнайдер Дж. у 2003 році у своїй дисертації [17] запропонував об'єднаний підхід до аналізу планування та оцінки найгіршого часу виконання для операційних систем реального часу, що дало змогу враховувати аномальні випадки при оцінці ефективності розкладів. У роботі [18] представлені результати досліджень, що стосуються принципів виникнення часових аномалій у суперскалярних процесорах.

Подальший розвиток теоретичних основ аномальних випадків був продовжений у працях Турчиної В. А. та Федоренко Н. К. Зокрема, у [19] розглянуто наближені алгоритми побудови оптимальних паралельних упорядкувань заданої довжини, що дозволяє аналізувати вплив різних факторів на отримані розв'язки. У [20] запропоновано алгоритми побудови всіх можливих паралельних упорядкувань заданої довжини, що сприяє глибшому розумінню структури таких упорядкувань та природи виникнення аномальних випадків. Відомі результати заклали основу для подальшого дослідження аномалій у задачах паралельного упорядкування вершин орграфів.

У даній роботі досліджено вплив зміни початкових умов, параметрів цільової функції та структури графів на виникнення аномальних випадків.

Запропоновано нові підходи до аналізу аномалій у класичних постановках задач та в деяких їх узагальненнях. Особливу увагу приділено випадкам, коли одночасне покращення двох або більше початкових характеристик (наприклад, збільшення кількості виконавців та послаблення технологічних обмежень) призводить до парадоксального погіршення результату. Проведено систематизацію таких ситуацій, наведено приклади, в яких вони можуть виникати, та визначено умови уникнення даних аномалій.

У межах дослідження введено поняття стабілізації довжини оптимального упорядкування, що дозволяє формально описати умови, при яких розв'язок задачі залишається незмінним. Це дає змогу визначати стійкі області параметричного простору та здійснювати стратегічний аналіз графових моделей ще до побудови конкретного розв'язку.

Запропоновано формалізовані умови, що дозволяють виявляти потенційно нестійкі розв'язки в класичних та узагальнених формулюваннях задач паралельного упорядкування. Розроблено критерії чутливості розкладу до змін у постановці задачі, зокрема при зміні продуктивності виконавців, фіксації деяких робіт за певними виконавцями, видаленні окремих завдань, скороченні списку пріоритетів або інверсії дуг графа (для класу інвертованих графів). У роботі введено відповідні аналітичні характеристики, які дозволяють визначати наперед можливість погіршення результату при заданих змінах у початкових даних.

Окремо досліджено ситуації, при яких зміна списку пріоритетів може призводити до виникнення аномалії. Для уникнення таких випадків запропоновано нові алгоритми побудови списку пріоритетів та динамічного перепризначення завдань.

У роботі приділено увагу питанню дозволу переривань при виконанні завдань як одному з факторів, що впливає на виникнення або запобігання аномальних випадків у задачах паралельного упорядкування. Досліджено, як для певних графових структур дозвіл на переривання виконання робіт може змінювати значення цільової функції.

Розроблено алгоритм пріоритетного перепризначення завдань з урахуванням можливості їх переривання. Встановлено, що такий підхід дозволяє зменшити вплив локальних аномалій та підвищити загальну стійкість розв'язку.

Наукова новизна роботи полягає у наступному:

- *вперше* розглянуто вплив одночасного покращення декількох початкових умов на появу аномальних випадків;

- *вперше* показано виникнення аномалій при скороченні списку пріоритетів, зменшенні кількості робіт, зміні продуктивності виконавців та фіксації робіт за виконавцями;

- запропоновано *нові умови* запобігання можливого виникнення аномалій.

- розроблено *нові* алгоритми зміни списку пріоритетів та динамічного перерозподілу завдань для забезпечення стабільності розкладу та мінімізації негативного впливу аномальних випадків на оптимальність розв'язку;

- *вперше* введено поняття стабілізації довжини оптимального упорядкування, сформульовано та обґрунтовано умови при яких вона відбувається;

- розглянуто вплив дозволу переривань при виконанні завдань на уникнення появи аномалій;

- досліджені аномальні випадки для класу інвертованих графів.

Отримані результати мають практичну цінність, оскільки їх можна використовувати для прогнозування можливих неочікуваних змін у якості розв'язків при оптимізації виробничих процесів, транспортних розкладів, логістичних систем, а також обчислювальних ресурсів, які формалізуються як оптимізаційні задачі на графах.

Практичне значення результатів полягає у можливості:

- використовувати запропоновані алгоритми для підвищення стійкості рішень у задачах планування;

- застосовувати критерії стабілізації для попереднього аналізу упорядкувань;

- прогнозувати виникнення нестійких сценаріїв у великих прикладних

системах;

- розробляти адаптивні системи управління, що реагують на локальні зміни в структурі задачі без необхідності повного перепланування;
- використовувати результати при побудові проєктів з високою вартістю і складною структурою залежностей, таких як енергетичні системи, обчислювальні кластери, багаторівневі логістичні мережі.

Основні результати дисертаційної роботи опубліковано в 14 наукових працях: 6 статей у наукових фахових виданнях України категорії Б з фізико-математичних наук; 1 стаття у періодичному науковому виданні країни ЄС; 7 тез доповідей у збірниках матеріалів міжнародних наукових конференцій та семінарів.

Ключові слова: дискретна оптимізація, задачі комбінаторної оптимізації, теорія розкладів, задачі паралельного упорядкування, оптимальне розбиття множин, розподіл ресурсів, графи, оптимальне упорядкування, аномалії, список пріоритетів, переривання.

ANNOTATION

Maliienko O.O. Theoretical justification of anomalous cases in problems of parallel ordering of vertices of a digraph. – Qualification scientific work in the form of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 113 "Applied Mathematics" (11 – Mathematics and Statistics) – Oles Honchar Dnipro National University, Dnipro, 2025.

The dissertation is devoted to the study of anomalous cases in one of the subclasses of discrete optimization problems, namely, in the problems of optimal ordering of vertices in directed graphs.

Formulations of such problems are related to the optimal organization of processes for distributing a finite set of tasks, subject to technological constraints on their execution order. One of the problems addresses the case where a finite set of executors is known, and the objective is to minimize the total completion time of all tasks without violating the precedence constraints. Another formulation focuses on minimizing the number of executors given a fixed completion time. In ordering theory, these are formulated as problems of parallel ordering of vertices in directed graphs.

In classical settings, such problems are classified as *NP*-hard. Therefore, to find their exact solutions, algorithms based on directed enumeration schemes are employed. An important direction of research in classical problems is the development of polynomial-time algorithms, which apply either to specific graph structures that define technological constraints or to special requirements of the initial data.

Known results for polynomially solvable cases refer to situations where the precedence constraint graphs belong to the class of input trees or certain specially structured series-parallel graphs. Also, the case where the number of executors equals two has been fully studied.

During the analysis of applied problems, it was found that some generalizations may lead to unpredictable situations causing deterioration in the value of the objective function. Such situations have been termed anomalies.

Anomalous cases in parallel ordering problems were first studied by Ronald Graham, who introduced the concept of anomalies and analyzed their impact on the efficiency of solutions. In his research [15], he showed that changes in certain parameters of the problem may lead to paradoxical outcomes, where the expected improvement causes deterioration of solutions.

Further studies of anomalies included analysis in the context of real-time systems. Thus, in 1999, Lundqvist T. and Stenstrom P., in their work [16], examined timing anomalies in microprocessor operation, showing that certain delays can lead to unpredictable effects in task execution. In 2003, Schneider J., in his dissertation [17], proposed an integrated approach to scheduling analysis and worst-case execution time estimation for real-time operating systems, which enabled taking anomalies into account when evaluating schedule efficiency. Research [18] presented principles of timing anomalies in superscalar processors.

Further development of the theoretical foundations of anomalous cases was continued in the works of Turchyna V.A. and Fedorenko N.K. In particular, [19] considers approximate algorithms for constructing optimal parallel orderings of a given length, allowing the analysis of various factors' influence on the obtained solutions. In [20], algorithms for constructing all possible parallel orderings of a given length are proposed, which contributes to a deeper understanding of the structure of such orderings and the nature of anomalies. These known results laid the groundwork for further studies of anomalies in parallel ordering problems on directed graphs.

This dissertation investigates the influence of changes in initial conditions, objective function parameters, and graph structures on the occurrence of anomalous cases. New approaches are proposed for analyzing anomalies in classical formulations and in certain generalizations. Particular attention is paid to cases where simultaneous improvement of two or more initial characteristics (e.g., increasing the number of executors and relaxing precedence constraints) leads to paradoxical worsening of the result. A systematization

of such situations is provided, along with examples of their occurrence and conditions for avoiding these anomalies.

The concept of stabilization of the optimal ordering length is introduced, allowing for a formal description of conditions under which the problem solution remains unchanged. This enables the identification of stable regions in the parameter space and strategic analysis of graph models before constructing a concrete solution.

Formalized conditions are proposed that enable the identification of potentially unstable solutions in both classical and generalized formulations of parallel ordering problems. Sensitivity criteria for the schedule are developed with respect to problem changes, particularly in scenarios involving changes in executor performance, task fixation to specific executors, task removal, shortening of the priority list, or inversion of graph arcs (for the class of inverted graphs). Corresponding analytical characteristics are introduced to determine in advance the possibility of result deterioration due to given changes in the initial data.

Situations where modifying the priority list can lead to anomalies are studied separately. To prevent such cases, new algorithms for priority list construction and dynamic task reassignment are proposed.

The dissertation also addresses the issue of allowing task interruptions as a factor influencing the emergence or prevention of anomalous cases in parallel ordering problems. It is shown that for certain graph structures, permitting task interruptions can alter the value of the objective function.

A task reassignment algorithm with interruption allowance is developed. It is established that this approach reduces the impact of local anomalies and increases overall solution robustness.

Scientific novelty of the work lies in the following:

- *for the first time*, the impact of simultaneous improvement of several initial conditions on the occurrence of anomalies is considered;
- *for the first time*, the emergence of anomalies is shown in the case of priority list reduction, task number reduction, changes in executor productivity, and task fixation by executors;

- *new conditions* are proposed to prevent the possible occurrence of anomalies. New algorithms are developed for modifying the priority list and dynamically redistributing tasks to ensure schedule stability and minimize the negative impact of anomalous cases on the optimality of the solution;

- the concept of optimal ordering length stabilization is introduced *for the first time*, and the conditions under which it occurs are formulated and substantiated;

- the impact of allowing task interruptions on avoiding anomalies is considered;

- anomalous cases are investigated for the class of inverted graphs.

The obtained results have practical value, as they can be used to predict possible unexpected changes in solution quality during the optimization of production processes, transport schedules, logistics systems, and computing resources formalized as optimization problems on graphs.

The practical significance of the results includes the ability to:

- apply the proposed algorithms to improve solution robustness in scheduling problems;

- use stabilization criteria for preliminary analysis of orderings; – Predict unstable scenarios in large-scale applied systems;

- develop adaptive control systems that respond to local changes in problem structure without full re-planning;

- employ the findings in the design of high-cost projects with complex dependency structures, such as energy systems, computing clusters, and multi-level logistics networks.

The main results of the dissertation are published in 14 scientific works: 6 articles in Ukrainian specialized scientific journals of category B in physical and mathematical sciences; 1 article in a scientific journal of an EU country; 7 conference abstracts in collections of materials from international scientific conferences and seminars.

Keywords: discrete optimization, combinatorial optimization problems, scheduling theory, parallel ordering problems, optimal partitioning of sets, resource allocation, graphs, optimal ordering, anomalies, priority list, interruption.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. **Малієнко О.О., Турчина В.А.** Задача оптимального розподілу ресурсів для випадку обмежень на їх використання. *Збірник наукових праць «Системні технології»*. 2025. Вип. 2. С. 75-81. doi: <https://doi.org/10.34185/1562-9945-2-157-2025-07>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1966/1236> (фахове видання, категорія Б).
2. **Maliienko O.O., Turchyna V.A.** Research on the relationship between anomalous cases in parallel scheduling problems and executor performance. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2024. Вип. 24. С. 127-133. doi: <https://doi.org/10.15421/322413>. Режим доступу до ресурсу: <https://pm-mm.dp.ua/index.php/pmmm/article/view/408/394> (фахове видання, категорія Б).
3. **Maliienko O.O., Turchyna V.A.** Analysis of the impact of task prioritization lists on the potential for avoiding anomalies in task scheduling. *Збірник наукових праць «Системні технології»*. 2024. Вип. 6. С. 167-174. doi: <https://doi.org/10.34185/1562-9945-6-155-2024-16>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1924> (фахове видання, категорія Б).
4. **Малієнко О.О., Турчина В.А.** Порівняльний аналіз аномалій для прямих та зворотних графів. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2023. Вип. 23. С. 161-170. doi: <https://doi.org/10.15421/322317>. Режим доступу до ресурсу: <https://pm-mm.dp.ua/index.php/pmmm/article/view/386> (фахове видання, категорія Б).
5. **Maliienko O.O., Turchyna V.A.** The study of the influence of combined changes in the initial data on the occurrence of anomalies for resource allocation. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2023. Вип. 23. С. 161-170. doi: <https://doi.org/10.15421/322317>. Режим доступу до ресурсу: <https://pm-mm.dp.ua/index.php/pmmm/article/view/386> (фахове видання, категорія Б).

модельовання». Дніпро, 2022. Вип. 22. С. 106-112. doi: <https://doi.org/10.15421/322211>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/345> (фахове видання, категорія Б).

6. **Челпанова (Малієнко) О.О.**, Турчина В.А. Узагальнення аномальних випадків у задачах упорядкування. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2021. Вип. 21. С. 220-226. doi: <https://doi.org/10.15421/322122>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/327> (фахове видання, категорія Б).

7. **Maliienko O.O.**, Turchyna V.A. Conditions for length stabilization in parallel ordering problems. *Theoretical and empirical scientific research: concept and trends*. Oxford, 2024. Вип. 6. С. 216-221. doi: <https://doi.org/10.36074/logos-02.02.2024.043>. Режим доступу до ресурсу: <https://archive.logos-science.com/index.php/conference-proceedings/article/view/1547>.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

8. Турчина В.А., **Малієнко О.О.** Про аномальні випадки в задачах розподілу електроенергії. *XXVII Міжнародна конференція «Автоматика 2024»*, 20-22 листопада 2024. м. Дніпро, 2024. С. 202-204. Режим доступу до ресурсу: [http://automatika2024.dp.ua/files/Автоматика-2024_\(тези_доповідей\).pdf](http://automatika2024.dp.ua/files/Автоматика-2024_(тези_доповідей).pdf) - page=202.

9. **Малієнко О.О.**, Коваленко Є.О. Дослідження впливу переривань на виникнення аномалій у задачах паралельного упорядкування. *Комбінаторні конфігурації та їхні застосування: Матеріали XXVI Міжнародного науково-практичного семінару присвяченого пам'яті професора Донця Г.П.* м. Кропивницький - Запоріжжя - Київ, 2024. С. 103-107. Режим доступу до ресурсу: https://zp.edu.ua/uploads/dept_s&r/2024/conf/6.3/CCTA-2024-proc.pdf#page=103.

10. **Малієнко О.О.** Аналіз впливу збільшення розмірності задачі на довжину паралельного упорядкування. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2023): Матеріали XXI міжнародної науково-практичної конференції до 105-річчя Дніпровського*

національного університету імені Олеся Гончара, 22-24 листопада 2023. м. Дніпро, 2023. С. 200-201. Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2023/11/mpzis-2023.pdf#page=200>.

11. Турчина В.А., **Малієнко О.О.** Стабілізація довжини упорядкування та аналіз впливу змін початкових даних на оптимальність. *Комбінаторні конфігурації та їхні застосування: Матеріали XXV Міжнародного науково-практичного семінару імені А.Я. Петренюка, 14-16 червня 2023. м. Запоріжжя - Кропивницький, 2023. С. 79-82.* Режим доступу до ресурсу: https://zp.edu.ua/uploads/dept_s&r/2023/conf/1.4/Petrenyuk_ISPS-25-proc.pdf#page=79.

12. Турчина В.А., **Малієнко О.О.** Наявність аномалій при побудові паралельних упорядкувань для одного класу графів. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2022): Матеріали XX ювілейної міжнародної науково-практичної конференції, 23-25 листопада 2022. м. Дніпро, 2022. С. 207.* Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf#page=207>.

13. Турчина В.А., **Челпанова (Малієнко) О.О.** Аномальні випадки при упорядкуванні деревовидних структур. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2021): Матеріали XIX міжнародної науково-практичної конференції, 17-19 листопада 2021. м. Дніпро, 2021. С. 194-195.* Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2021/12/mpzis-2021.pdf#page=194>.

14. Турчина В.А., **Челпанова (Малієнко) О.О.** Узагальнення аномалій в задачах упорядкування. *Комбінаторні конфігурації та їхні застосування: Матеріали XXIII Міжнародного науково-практичного семінару імені А.Я. Петренюка, присвяченого 70-річчю Льотної академії Національного авіаційного університету, 13-15 травня 2021. м. Запоріжжя - Кропивницький, 2021. С. 188-191.* Режим доступу до ресурсу: https://www.glau.kr.ua/images/docs/sbornik/materiali_23_mnp_seminaru.pdf#page=188

ЗМІСТ

ВСТУП.....	16
Розділ 1. ПОНЯТТЯ АНОМАЛІЙ ДЛЯ ПЕВНИХ КЛАСІВ ЗАДАЧ.....	24
1.1 Задачі оптимального упорядкування вершин оргграфів як підклас задач теорії розкладів.....	24
1.2 Аномалії та приклади їх виникнення.....	26
1.3 Місце аномалій у деяких прикладних задачах.....	29
1.4 Висновки до розділу.....	36
Розділ 2. ДОСЛІДЖЕННЯ АНОМАЛЬНИХ ВИПАДКІВ У ЗАДАЧАХ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ.....	38
2.1 Основні означення та постановки задач.....	38
2.2 Огляд методів побудови оптимальних упорядкувань.....	39
2.3 Вплив покращення початкових умов на появу аномальних випадків.....	41
2.4 Залежність значення цільової функції від змін параметрів упорядкування.....	46
2.5 Випадки для інвертованих графів.....	50
2.6 Аномальні випадки в задачах паралельного виконання програм.....	57
2.7 Висновки до розділу.....	60
Розділ 3. ВИДИ АНОМАЛІЙ ТА УМОВИ ЇХ ВИНИКНЕННЯ ДЛЯ СПЕЦІАЛЬНИХ КЛАСІВ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ.....	63
3.1 Умови виникнення аномалій в деяких класах задач.....	63
3.2 Випадки взаємно однозначної відповідності між певними завданнями та виконавцями.....	73
3.3 Умови стабілізації довжини упорядкування.....	77
3.4 Вплив переривань на виникнення аномалій.....	84
3.5 Висновки до розділу.....	93
Розділ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	95
4.1 Опис структури та функціональних модулів програми.....	95

	15
4.2 Інтерфейс програми.....	99
4.3 Інструкція користувача.....	102
ВИСНОВКИ.....	107
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	109
ДОДАТОК А. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА.....	121

ВСТУП

Задачі неперервної та дискретної оптимізації охоплюють широкий спектр прикладних задач, які виникають в різних галузях. Зокрема, це стосується логістики, планування ресурсів, оптимальної організації обчислювальних процесів та інше. Знаходження оптимальних розв'язків цих задач сприяє прийняттю сучасних технологічних рішень, від автоматизованих систем складання розкладів до моделей оптимального управління виробничими процесами.

Суттєвий внесок у теоретичні надбання, пов'язані з розробкою моделей відповідних задач методів та алгоритмів їх розв'язку, внесли Сергієнко І.В., Кісельова О.М., Яковлєв С.В., Стецюк П.І., Пічугіна О.С., Семенова Н.В., Новожилова М.В. [21-30].

Серед задач комбінаторної оптимізації особливої уваги потребують задачі, що належать до класу NP -важких, точні алгоритми розв'язання яких мають експоненційну складність. Серед них виділяють окремий підклас задач комбінаторної оптимізації, пов'язаних із спеціальними оптимізаційними задачами на графах. Традиційно дослідження таких задач, пов'язані з використанням апарату теорії графів при моделюванні, розробці нових точних та наближених алгоритмів.

Графи є зручним інструментом і при візуалізації деяких складових у досліджуваних процесах. Окремі роботи науковців Козіна І.В., Гук Н.А., Гуляницького Л.Ф., Донця Г.П., Петренюка А.Я., Семенюти М.Ф., Бурдюка В.Я., Турчиної В.А., Наконечної Т.В. пов'язані саме з такими дослідженнями [31-49].

Дана робота присвячена задачам, що відносяться до класу задач теорії розкладів і пов'язані з оптимальним упорядкуванням вершин орієнтованих графів. У загальній постановці ці задачі є NP -важкими [50]. Для їх розв'язання в загальному випадку відомі точні алгоритми, що реалізують схеми направленого перебору, та виявлені підкласи задач, для яких точні розв'язки можна знайти за поліноміальний час [51, 52].

Отримано також деякі оцінки значень цільових функцій, що дозволяють оцінювати якість наближених розв'язків [19, 53].

Поява роботи Грехема Р. [15] сприяла виникненню ще одного напрямку досліджень, що стосується вивчення можливих так званих аномальних випадків у цих задачах. Автор статті під «аномаліями» розуміє непрогнозоване погіршення значення цільової функції при покращенні початкових даних задачі, метою якого було поліпшення результату.

Цей напрям продовжив розвиток у дослідженнях Лундквіст Т., Колота Дж., Тові С. А., Панвалкар С. С. [16, 18, 54–56].

Незважаючи на певний прогрес у розробці методів задач, що розглядаються, дослідження аномальних випадків залишаються недостатньо вивченими. Аномалії, які можуть проявлятися у вигляді непередбачуваних або парадоксальних змін значень цільової функції при зміні початкових параметрів задачі, часто ускладнюють процес виконання завдань. Такі явища впливають як на точність, так і на стабільність результатів, що може мати критичне значення у прикладних галузях.

У своїх працях, Грехем Р. і Лундквіст Т. [15, 16, 57] досліджували окремі аспекти таких задач, однак питання класифікації та систематизації аномалій залишаються відкритими. Актуальність проблеми зростає в умовах сучасних високонавантажених обчислювальних систем, де навіть незначні зміни параметрів можуть спричинити суттєві відхилення у поведінці системи. Так, у реальних умовах виконання завдань, що пов'язані з логістикою, енергетикою, телекомунікаціями, кібербезпекою, індустрією 4.0, зростає потреба в надійних алгоритмах, здатних забезпечити стабільність результатів навіть за змінних початкових параметрів. Наявність аномальних випадків у розв'язках таких задач ускладнює застосування стандартних методів оптимізації, оскільки вони можуть призводити до непередбачуваного погіршення результатів, незважаючи на формальне поліпшення умов [58-61]. Це особливо критично для задач, де від стійкості результату залежать реальні фінансові або соціальні наслідки.

Таким чином, вивчення природи аномальних явищ, їх класифікація та розробка методів запобігання є актуальною науковою проблемою, вирішення якої дозволить підвищити якість прийняття рішень у системах управління.

Окрім цього, сучасні задачі оптимального упорядкування дедалі частіше мають справу зі складними структурами графів, змінною продуктивністю ресурсів, дозволом на переривання завдань, різними критеріями оптимальності. У таких умовах класичні алгоритми не завжди виявляються ефективними. Це обґрунтовує необхідність розробки нових підходів, які б враховували вищезазначені фактори та дозволяли моделювати ситуації з аномальним погіршенням значень цільових функцій.

У дисертаційній роботі акцент зроблено на розширення теоретичних уявлень щодо виникнення аномалій у задачах оптимального паралельного упорядкування. Зокрема, досліджується вплив скорочення списку пріоритетів, фіксації завдань за виконавцями, зміни продуктивності, а також ефекти переривань і інверсії у графових структурах. Окрема увага приділяється побудові умов стабілізації довжини оптимального упорядкування, що є важливим кроком до формалізації поведінки систем в умовах динамічних змін.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження проводилося в рамках теми науково-дослідної роботи Міністерства освіти і науки України «Розробка та реалізація методів оптимального функціонування складних систем» (№ держреєстрації 0122U001466, 2022–2024 рр.) при кафедрі обчислювальної математики та математичної кібернетики Дніпровського національного університету імені Олеся Гончара.

Мета і завдання дослідження. Метою роботи є теоретичне обґрунтування природи аномальних випадків у задачах паралельного упорядкування вершин орграфів, вивчення умов їх виникнення та розробка методів мінімізації їх впливу на розв'язок задачі.

Завдання дослідження:

- класифікація аномалій у одному класі оптимізаційних задач на графах;
- вивчення особливостей виникнення аномальних випадків у задачах паралельного упорядкування;
- розробка алгоритмів для розв'язання задач з урахуванням появи аномальних випадків;

- проведення аналізу впливу зміни параметрів задач на появу аномалій;
- розробка програмного забезпечення для обчислювального моделювання досліджуваних явищ.

Об'єкт дослідження – підклас задач дискретної оптимізації, у яких можливе виникнення аномальних випадків.

Предмет дослідження – методи і алгоритми аналізу виникнення аномалій у задачах паралельного упорядкування вершин орграфів.

Методи дослідження: у процесі розв'язання завдань були застосовані підходи дискретної й комбінаторної оптимізації; методи, що ґрунтуються на теорії графів і теорії розкладів; інструменти теорії оптимального впорядкування.

Наукова новизна одержаних результатів:

– *вперше* розглянуто вплив одночасного покращення декількох початкових умов на появу аномальних випадків;

– *вперше* показано виникнення аномалій при скороченні списку пріоритетів, зменшенні кількості робіт, зміні продуктивності виконавців та фіксації робіт за виконавцями;

– *отримано нові* умови, що запобігають можливому виникненню аномалій;

– *розроблено нові* алгоритми зміни списку пріоритетів та динамічного перерозподілу завдань для забезпечення стабільності розкладу та мінімізації негативного впливу аномальних випадків на оптимальність розв'язку;

– *вперше* введено поняття стабілізації довжини оптимального упорядкування, сформульовано та обґрунтовано умови при яких вона відбувається;

– *розглянуто* вплив дозволу переривань на виникнення аномальних випадків;

– *вперше* проаналізовано можливість аномальних випадків для інвертованих графів;

– *подальшого розвитку набули* методи дослідження впливу аномалій на ефективність управління процесами.

Практичне значення одержаних результатів. Отримані результати можуть

бути використані для розв'язання широкого кола прикладних задач, пов'язаних з ефективним плануванням, управлінням ресурсами та побудовою стійких розкладів. Розроблені теоретичні підходи та алгоритми враховують можливу наявність аномальних випадків, аналіз яких дозволяє уникати парадоксальних ситуацій, коли покращення вхідних параметрів спричинює погіршення цільових показників. Це особливо важливо у задачах, що мають критичне значення для забезпечення стабільності функціонування систем – у виробництві, логістиці, телекомунікаціях, обчислювальних середовищах, енергетиці та ін.

Запропоновані методи динамічного перерозподілу завдань, побудови пріоритетних списків і врахування особливостей графової структури задач дозволяють підвищити ефективність розкладів, зменшити ризик виникнення конфліктів у послідовності виконання робіт та забезпечити адаптивність до змін початкових умов.

Результати дослідження можуть бути впроваджені у системах реального часу, зокрема в мультипроцесорних і розподілених середовищах, де необхідно забезпечити гарантоване виконання задач при обмежених ресурсах за прийнятний час.

Крім того, програмна реалізація запропонованих алгоритмів може використовуватись як інструмент підтримки прийняття рішень. Подальші дослідження можуть передбачати виділення підкласів графів та відповідних початкових умов, для яких аномалії не виникають, або тих підкласів, для яких ці аномалії прогнозовані.

Особистий внесок здобувача. Результати дисертаційної роботи викладені у 14 наукових працях [1–14]. Усі результати, що виносяться на захист, отримані автором особисто. У співавторстві результати представлені у 13 наукових працях. У статті [1] здобувачеві належить формулювання задачі та доведення тверджень про умови існування допустимого упорядкування; співавторці Турчиній В.А. визначення загального підходу до узагальнення задач розподілу з урахуванням фіксованих виконавців та методологія аналітичного доведення. У статті [2] здобувачеві належить формулювання задачі дослідження аномальних випадків у

задачах розподілу з частковим порядком, аналіз умов, що впливають на виникнення "вузьких місць" при нерівномірному розподілі завдань, та розробка рекомендацій щодо уникнення таких аномалій; співавторці Турчиній В.А. належить визначення загального підходу до формалізації моделей з неоднорідною продуктивністю виконавців; постановка задачі в загальному вигляді. У статті [3] здобувачеві належить аналіз впливу списків пріоритетів на можливість уникнення аномалій у задачах розподілу, розробка алгоритму динамічного перерозподілу задач для зменшення затримок; співавторці Турчиній В.А. належить обґрунтування доцільності розгляду динамічних алгоритмів у задачах паралельного упорядкування та аналіз застосовності розробленого підходу до ширшого класу задач дискретної оптимізації. У статті [4] здобувачеві належить дослідження аномальних випадків для інвертованих графів, розробка умов, що дозволяють запобігти виникненню аномалій при інверсії; співавторці Турчиній В.А. належить постановка задачі в контексті узагальненого аналізу прямих та зворотних графів, формулювання залежностей між структурними характеристиками графа. У статті [5] здобувачеві належить аналіз трьох випадків одночасного покращення початкових параметрів і виникнення аномального погіршення цільової функції у задачах оптимізації на графах, розробка основ для подальшого вивчення підкласів графів із подібними аномаліями; співавторці Турчиній В.А. належить постановка задачі в узагальненому вигляді та обґрунтування впливу комбінованих змін на оптимальність розв'язку. У статті [6] здобувачеві належить узагальнення випадків аномалій у задачах побудови паралельних упорядкувань мінімальної довжини, аналіз впливу початкових умов; співавторці Турчиній В.А. належить визначення напрямку досліджень для виокремлення підкласів графів, у яких аномалії не впливають на оптимальність розв'язку та побудова контрприкладів, що ілюструють парадоксальні випадки збільшення довжини упорядкування. У статті [7] здобувачеві належить формулювання та доведення умов стабілізації довжини упорядкування, побудова та обґрунтування алгоритму визначення моменту стабілізації довжини розкладу залежно від кількості виконавців; співавторці Турчиній В.А. належить постановка питання про залежність стабілізації від

ширини розкладу та аналітична підтримка доведення узагальнених оцінок для випадків з різними типами вихідних обмежень. У публікації [8] здобувачеві належить постановка задачі динамічного планування енергетичних систем із урахуванням денних/нічних тарифів і використанням сонячної енергії, аналіз впливу таких факторів на виникнення аномалій; співавторці Турчиній В.А. належить концептуалізація підходу до моделювання аномальних випадків у системах енергетичного розподілу та обґрунтування доцільності застосування формального апарату теорії розкладів для задач динамічного енергопланування. У публікації [9] здобувачеві належить дослідження впливу переривань на виникнення аномалій у задачах паралельного упорядкування, розробка алгоритму пріоритетного перерозподілу з урахуванням можливих переривань; співавтору Коваленко Є.О. належить формалізація початкових умов задачі, побудова математичної моделі з урахуванням ширини упорядкування та пріоритетів, а також участь у формулюванні умов виникнення аномалій. У публікації [11] здобувачеві належить введення концепції стабілізації довжини упорядкувань і аналіз впливу змін початкових даних на оптимальність розв'язку; співавторці Турчиній В.А. належить формулювання математичних умов стабілізації, побудова прикладів, що демонструють виникнення аномалій після стабілізації. У публікації [12] здобувачеві належить аналіз аномальних випадків при спрощенні технологічних обмежень; співавторці Турчиній В.А. формулювання умов, за яких аномалії виникають у деревовидних структурах. У публікації [13] здобувачеві належить розробка умов запобігання аномаліям при упорядкуванні деревовидних структур; співавторці Турчиній В.А. аналіз впливу зваженості дерев на оптимальність упорядкування. У публікації [14] здобувачеві належить аналітичний огляд відомих результатів досліджень, що лягли в основу даного напрямку; співавторці Турчиній В.А. належить формалізація задачі узагальненого паралельного упорядкування та класифікація випадків комбінованих змін параметрів.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідались і обговорювались на міжнародних наукових конференціях «Математичне та програмне забезпечення інтелектуальних систем»

(м. Дніпро, 2021, 2022, 2023 рр.); XXVII міжнародній науковій конференції «Автоматика 2024» (м. Дніпро, 2024 р.); міжнародних науково-практичних семінарах «Комбінаторні конфігурації та їхні застосування» (м. Запоріжжя – Кропивницький, 2021, 2023, 2024 рр.), на щорічних конференціях за підсумками науково-дослідної роботи ДНУ ім. О. Гончара.

Публікації. Основні результати дисертаційної роботи опубліковано в 14 наукових працях: 6 статей ([1-6]) у наукових фахових виданнях України категорії Б з фізико-математичних наук, 1 стаття [7] в іншому науковому виданні, 7 тез доповідей у збірниках матеріалів міжнародних та регіональних наукових конференцій та семінарів ([8-14]).

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел, що містить 110 найменувань на 12 сторінках та додатків на 3 сторінках. Загальний обсяг дисертації – 123 сторінки, обсяг основного тексту – 104 сторінки. Робота містить 33 рисунки та 48 таблиць.

Розділ 1 ПОНЯТТЯ АНОМАЛІЙ ДЛЯ ПЕВНИХ КЛАСІВ ЗАДАЧ

1.1 Задачі оптимального упорядкування вершин оргграфів як підклас задач теорії розкладів

Теорія розкладів – це напрям дискретної математики, що досліджує задачі організації виконання множини робіт або операцій у часі, часто з урахуванням обмежень на ресурси, виконавців або послідовність дій. Основна мета полягає в побудові оптимального або допустимого порядку виконання завдань відповідно до заданих критеріїв ефективності.

Певний внесок у розробку деяких моделей задач дискретної оптимізації відображено у працях [62-70], в яких наводяться деякі постановки та прикладні сфери використання. Наведено також певні підкласи поліноміально розв'язних оптимізаційних задач на графах.

Задачі теорії розкладів можна розділити на дві групи:

1) задачі із перериваннями: у певні моменти часу виконання роботи може бути зупинене (з можливістю продовження або завершення з використанням того ж, або іншого ресурсу) задля виконання іншої операції;

2) задачі без переривань: кожна робота виконується неперервно.

Одним із основних питань при дослідженні відповідних моделей є питання визначення складності відповідних задач. У залежності від того чи відома складність, а якщо відома, то якому класу задач NP -повних чи поліноміально-розв'язних вона належить, і розробляються відповідні методи пошуку оптимальних розв'язків.

За типом розв'язання виділяють наступні класи задач теорії розкладів:

1) задачі узгодження;

2) задачі розподілу;

3) задачі упорядкування.

У задачах узгодження досліджується зв'язок між заданою тривалістю виконання робіт та часом їх надходження в обслуговуючу систему. Задачі розподілу передбачають пошук оптимального розподілу робіт за виконавцями. У задачах упорядкування необхідно скласти розклад (чи порядок) виконання робіт кожним виконавцем при заданих умовах розподілу робіт за виконавцями та усіх параметрах робіт (тривалість виконання, час надходження тощо).

Один із підкласів задач теорії розкладів стосується випадків, коли на порядок виконання завдань накладаються певні технологічні обмеження і потрібно або мінімізувати обсяг ресурсів при заданих часових характеристиках, або при відомих обсягах ресурсів час завершення відповідного процесу.

Якщо представляти технологічні обмеження у вигляді відповідного графу, то отримуємо оптимізаційні задачі на графах. Розглянемо формулювання практичних задач, що можуть бути зведені до таких задач.

Нехай для виконання роботи i потрібний один виконавець, $i=1,2,\dots,n$, де n – кількість робіт. Вважається, що кількість виконавців фіксована і менша за n . Вводяться наступні припущення:

- 1) кожен виконавець може виконувати будь-яку роботу;
- 2) у залежності від постановки тривалість виконання роботи i не залежить від виконавця і дорівнює однаковому проміжку часу для всіх робіт або для кожної роботи i задається тривалість її виконання t_i ;
- 3) не забороняється паралельне виконання декількох робіт;
- 4) зв'язок між деякими роботами підпорядкований заданим технологічним обмеженням: робота j не може бути почата, поки не завершена робота i .

Розклад, який не порушує технологічних обмежень називається *допустимим*.

Допустимий розклад, згідно з яким роботи завершуються за мінімальний термін при заданій кількості виконавців або при заданому терміні завершення потребує мінімальну кількість виконавців, називається *оптимальним*.

Нехай m – кількість виконавців, t – час завершення виконання всіх робіт.

Задача 1. За заданим значенням t побудувати допустимий розклад, що

мінімізує t .

Задача 2. За заданим значенням t побудувати допустимий розклад, що мінімізує t .

Враховуючи пункт 4, наведених вище припущень, технологічні обмеження на порядок виконання робіт можуть бути задані орієнтованим графом $G=\{V,U\}$. З урахуванням умови 3 задачі 1 і 2 можуть бути сформульовані як задачі спеціальних упорядкувань вершин орграфів. Відповідні означення та їх постановки наводяться у розділі 2.

1.2 Поняття аномалій та приклади їх виникнення

Однією із проблем при розв'язанні класичних і узагальнених задач є наявність у них ситуацій, які можна вважати аномаліями. Тобто, таких випадків, коли зміна входних параметрів задачі, що з логічних міркувань повинна зменшувати оптимальне значення цільової функції, або принаймні не погіршувати його, призводить до його збільшення. Наприклад, непрогнозоване збільшення загального часу виконання за умови зменшення часів виконання окремих завдань, або наявність в орграфі транзитивних, неінформативних дуг, які впливають на оптимальність розв'язку, отриманого за деяким алгоритмом.

Поняття аномалій в задачах теорії упорядкувань було вперше введено Грехемом Р. [15]. Він розглянув задачу, коли задана скінчена множина робіт та скінчена множина виконавців. Виконання робіт з відомим часом виконання кожної підпорядковано певним технологічним обмеженням. Грехемом були проаналізовані сама множина виконавців, структура технологічних обмежень, час виконання робіт. Крім того, він досліджував випадки, коли задані бажані пріоритети виконання робіт. При аналізі цієї задачі такий аномальний ефект може бути виявлений при наступних змінах параметрів:

- 1) зменшенні часу виконання робіт;
- 2) послабленні технологічних обмежень на порядок робіт;

Довжина $l=16$ свідчить про те, що у цьому випадку також виникає аномалія.

4) Змінюємо порядок виконання робіт. Нехай

$L'=(1,3,4,5,6,7,8,9,2,10,11,12,13,14)$, граф G_l залишається без змін.

Таблиця 1.5 – Оптимальне упорядкування для графа G_l при L'

1	1	1	1	1	5	5	5	10	12	12								
3	3	3	9	2	2	2	7	7	13	13	13	13	13	13	13	13	13	14
4	4	8	8	11	11	6	6	6										

Довжина $l=19$, тобто інтуїтивно вдала зміна порядку робіт знову призвела до погіршення значення цільової функції.

1.3 Місце аномалій у деяких прикладних задачах

У прикладних задачах важливо розуміти причини та умови виникнення аномалій, щоб передбачати їх можливий вплив на ефективність організації самих процесів. Вони можуть призводити до неочікуваних затримок, погіршення ефективності системи, збоїв у виконанні завдань, втрати даних або навіть до помилок у результаті.

Так, наприклад, у роботі [18] розглядається важливість паралельного виконання завдань в прискоренні розрахунків електромагнітних феноменів, які включають моделювання 3D-систем з розподіленими параметрами.

Чисельна реалізація передбачає вирішення крайових задач з урахуванням змінних електромагнітного поля, таких як магнітний векторний потенціал A та електричний скалярний потенціал V , за допомогою методів скінченних елементів і спряженого градієнта. Система рівнянь включає проникність μ , провідність σ , швидкість рухомого якоря v і густину струму тонкої котушки $j(t)$.

Паралельне виконання завдань є загальним інструментом, який використовується для скорочення часу обчислень. Даний підхід передбачає поділ

обчислювального простору на кілька підпросторів, кожен з яких обробляється окремим вузлом у кластері. Зв'язок між вузлами встановлюється за допомогою сокетів та використання технології MPI (Message Passing Interface). Завдання, що виконуються, включають передачу даних, обчислення та валідацію.

Спосіб реалізації паралельної системи представлено на основі моделі прямого орієнтованого графа. Задається кількість завдань та час їх виконання, кількість кластерів (виконавців), множина обмежень та список пріоритетності виконання завдань. Дослідження спрямоване на три випадки, які найчастіше використовуються, для прискорення обчислень: зменшення часу виконання завдань, збільшення обчислювальної потужності та зменшення обраних обмежень прецедентів. Для кожного з випадків автори на прикладах проілюстрували погіршення значення цільової функції, що і відповідає виникненню аномалій.

Результати авторів підкреслюють необхідність удосконалення методів роботи з паралельними системами для досягнення ефективнішого використання обчислювальних ресурсів.

У бездротових сенсорних мережах (БСМ) оптимізація енергоспоживання є ключовим завданням, оскільки сенсори зазвичай мають обмежені ресурси живлення. Аномалії можуть виникати через нерівномірний розподіл навантаження між вузлами, що призводить до передчасного виснаження батарей окремих сенсорів і, як наслідок, до зниження загальної продуктивності мережі. У дослідженні [71] розглядаються різні алгоритми кластеризації, спрямовані на зменшення енергоспоживання в БСМ. Одним із найбільш важливих факторів є енергетична ефективність, оскільки сенсори часто мають обмежені ресурси батарей. Погіршення розподілу енергії серед вузлів може призвести до швидкої розрядки сенсорів, що в свою чергу спричиняє збої в роботі мережі.

У сфері кібербезпеки виявлення аномалій є критичним для запобігання несанкціонованому доступу та захисту інформаційних систем. Аномальні дії можуть свідчити про потенційні загрози, такі як атаки типу "відмова в обслуговуванні" або спроби проникнення в систему. У статті [72] представлено огляд методів виявлення аномалій у мережевому трафіку. Автори розглядають різні

підходи до ідентифікації відхилень від нормальної поведінки та аналізують ефективність цих методів у реальних умовах.

Слід відмітити і наступні прикладні сфери, де використовується паралелізм при виконанні операцій і цілком можливо можуть з'являтися аномалії.

1) У біоінформатиці такі алгоритми використовуються для аналізу геномних даних, метаболомних реакцій, молекулярної моделювання та прогнозування властивостей біомолекул. Паралельне виконання завдань дозволяє здійснювати швидкий пошук у великих наборах даних для виявлення мутацій, прогнозування функцій білків та розв'язання складних генетичних проблем [73].

2) У фінансовому секторі обмеження часткового порядку на операції, що проводяться для аналізу ринків, прогнозування цін на акції, оптимізації інвестиційних портфелів та ризик-менеджменту [74]. Вони дозволяють швидко обробляти великі обсяги фінансових даних та розраховувати складні фінансові моделі.

3) У метеорології та кліматології паралельні обчислення використовуються для створення складних моделей прогнозування погоди, аналізу кліматичних змін, моделювання природних катастроф та оцінки їх наслідків. Величезні обсяги даних та складність моделей роблять оптимізаційні методи невід'ємною складовою цих досліджень [75].

4) У світі штучного інтелекту і машинного навчання розподілена обробка інформації використовується для навчання складних моделей глибокого навчання, обробки великих обсягів даних та покращення продуктивності алгоритмів [76].

5) У галузі створення відеоігор паралельне виконання завдань допомагає в реалізації реалістичних графічних ефектів, швидкому обчисленні фізичних процесів та створенні відповідних сценаріїв поведінки персонажів.

Ці приклади вказують на широкий спектр областей, де оптимізаційні методи моделювання задач відіграють важливу роль у вирішенні складних завдань, прискоренні обчислень та підвищенні продуктивності у великих дослідницьких проєктах.

Аномалії у таких задачах можуть виникати через:

1) нерівномірний розподіл завдань між виконавцями, що призводить до дисбалансу завантаження та затримок;

2) невідповідність обмежень при моделюванні, наприклад, через конфлікти у пріоритетах або ресурсах;

3) збої у комунікації між вузлами паралельної системи, що ускладнює передачу даних і синхронізацію;

4) чисельні похибки при моделюванні, які зростають у складних системах.

Наведемо приклади впливу аномалій на значення цільової функції:

1) у біоінформатиці неоднозначний розподіл завдань може призводити до збільшення часу пошуку в базах даних або неточності у прогнозах, що впливає на результати генетичних досліджень;

2) у фінансовому секторі, коли алгоритми оптимізації не враховують часові обмеження, можливі затримки в обробці транзакцій, що впливає на швидкість виконання ринкових операцій;

3) у метеорології неправильна обробка даних через аномалії у розподіленій системі може спричиняти похибки у прогнозуванні погоди, що особливо критично при прогнозуванні природних катастроф.

Розглянемо методи виявлення та усунення аномалій при аналізі практичних задач:

1) аналітичні методи: використання графів залежностей для виявлення вузьких місць у системі; аналіз часових серій для виявлення невідповідностей у завантаженні виконавців;

2) моделювання та оптимізація: застосування методів вирівнювання навантаження для зменшення затримок; використання стохастичних моделей для прогнозування виникнення аномалій;

3) апаратні рішення: впровадження додаткових ресурсів для підтримки розподілених систем; оптимізація мережі передачі даних між вузлами.

Дослідження аномалій у прикладних задачах дозволяє:

1) підвищити ефективність паралельних обчислень;

2) зменшити загальний час виконання завдань;

- 3) забезпечити стабільність роботи систем навіть у складних умовах;
- 4) створити адаптивні алгоритми для складних обчислювальних задач.

Такі підходи особливо важливі у високоточних галузях, де навіть незначні похибки можуть призводити до значних втрат.

У сучасних енергетичних системах існує необхідність в оптимальному управлінні споживанням електроенергії для забезпечення максимально ефективного використання доступних ресурсів. Однак, навіть при покращенні початкових умов, таких як збільшення потужності системи або використання відновлюваних джерел енергії, можуть виникати аномалії.

Динамічне планування споживання енергії є важливою складовою управління сучасними енергетичними системами. Основна мета полягає в тому, щоб оптимізувати розподіл доступної електроенергії між споживачами з різними потребами та пріоритетами, зважаючи на змінні умови тарифікації та джерела енергії. Наприклад, лікарні або критична інфраструктура повинна надаватись перевага у порівнянні з іншими менш пріоритетними споживачами. При динамічному плануванні необхідно враховувати змінні умови розподілу енергії, зокрема:

- 1) вартість енергії змінюється залежно від часу доби: денний та нічний тарифи;
- 2) існує можливість використання відновлюваних джерел енергії, зокрема сонячної енергії, яка може бути безкоштовною у певні періоди доби;
- 3) пріоритетність завдань, яка може змінюватися в залежності від потреб споживачів.

Проте, навіть при покращенні початкових умов, таких як збільшення пропускної здатності системи або встановлення нових джерел енергії, можуть виникати аномалії, а саме: збільшення загальних витрат споживачів.

Розглянемо задачу динамічного планування енергетичних систем з урахуванням денних і нічних тарифів, а також можливості використання безоплатної сонячної енергії. Для вирішення задачі планування необхідно враховувати пріоритетність завдань та умови тарифікації енергії. Введемо наступні змінні:

E_i – кількість енергії, спожитої на i -му етапі планування (кВт·год);

$P_{\text{ден}}, P_{\text{ніч}}$ – тариф на електроенергію за кВт·год у денний і нічний час (грн/кВт·год).

Функція вартості може бути виражена як:

$$C(E) = \sum_{i=1}^n (P_{\text{ден}} * E_i + P_{\text{ніч}} * E_i). \quad (1.1)$$

У випадку ускладнених умов функція вартості може виражатись таким чином:

$$C(E) = \sum_{i=1}^n \sum_{j=1}^k (P_i * E_{ij}), \quad (1.2)$$

де E_{ij} – кількість енергії, спожитої j -м споживачем у період i (кВт·год);

P_i – тариф на електроенергію за кВт·год у період i (грн/кВт·год).

Ця формула дозволяє врахувати як різницю в тарифах на електроенергію залежно від часу доби, так і можливість безкоштовного використання сонячної енергії у денний час.

Для реальних початкових умов, коли вартість електроенергії за денним тарифом дорівнює 4.32 грн/кВт·год, за нічним 2.16 грн/кВт·год та безкоштовна за умови використання сонячної енергії, була змодельована система з трьома розподільчими станціями та десятьма споживачами.

Очікувалося, що додавання сонячної енергії та покращення початкових умов, таких як спрощення технологічних обмежень та зменшення часу використання електроенергії деякими споживачами, призведуть до зменшення загальної вартості. Проте зміщення пріоритетів споживачів змусило одного з них використовувати більшу частину енергії за денним тарифом. Це призвело до збільшення загальної вартості з 194,4 грн до 259,2 грн, що є аномалією, оскільки при покращенні умов, що стосуються вартості та кількості спожитої електроенергії, очікування не справдилися.

Розглянемо тепер задачу оптимізації заряджання електромобілів у розподіленій системі зарядних станцій. Задача полягає у плануванні черговості заряджання електромобілів на розподіленій мережі зарядних станцій з обмеженою кількістю доступних станцій і батарей.

Метою є мінімізація часу заряджання, враховуючи пріоритети електромобілів (наприклад, електромобілі з меншим запасом ходу отримують вищий пріоритет).

Аномалія: покращення початкових умов, таких як додавання нових зарядних станцій або збільшення швидкості зарядки, може парадоксально призвести до збільшення загального часу очікування. Це може статися через перерозподіл черг або через те, що користувачі з нижчим пріоритетом займають станції швидше, ніж ті, що мають високий пріоритет. Необхідно оптимізувати процес заряджання електромобілів на різних станціях з урахуванням тарифів, часу заряджання та доступності станцій.

Вартісна функція: позначимо E_i – кількість енергії для заряджання i -го автомобіля (кВт·год); $P_{\text{зар}}$ – тариф на електроенергію за кВт·год (грн/кВт·год); τ_i – час заряджання автомобіля (години).

Вартісна функція заряджання виглядатиме так:

$$C(E) = \sum_{i=1}^n (P_{\text{зар}} * E_i * \tau_i). \quad (1.3)$$

Розглянемо приклад з наступними початковими даними:

- енергія для заряджання трьох автомобілів: 50 кВт·год, 80 кВт·год, 100 кВт·год;
- час заряджання: 2 години, 4 години, 3 години;
- вартість за заряджання: 3 грн/кВт·год.

Отримуємо загальну вартість, що дорівнює 2160 грн.

При наступних дослідженнях виявляється, що такі покращення, як наприклад, збільшення кількості зарядних станцій, може призводити до зміни черговості, що в свою чергу призводить до аномального зростання загальної функції вартості.

Розглянемо також задачу розподілу потужностей у мережах передачі даних для мобільних пристроїв.

У мобільних мережах, зокрема в умовах 5G, є проблема оптимального розподілу пропускної здатності між мобільними пристроями з урахуванням їх пріоритетності (наприклад, пристрої з критичними для життя додатками отримують вищий пріоритет) [77]. Однак, навіть при покращенні пропускної здатності мережі можуть виникнути аномалії.

Аномалія: при збільшенні пропускної здатності мережі чи підвищенні швидкості обробки даних в базових станціях може виникати затримка через

збільшення конкуренції за ресурси між мобільними пристроями.

Необхідно оптимізувати розподіл потужностей для передачі даних через базові станції, щоб мінімізувати витрати на використання ресурсу каналу. Вартість може залежати від зайнятості мережі і тарифів за передачу даних (грн/Мб).

Функція вартості: P_i – потужність, передана для i -го пристрою (Мб); τ_i – довжина часового проміжку передачі, $C_{\text{перед}}$ – тариф на передачу даних (грн/Мб); V_j – пропускна здатність мережі j -ої станції.

Вартісна функція має наступний вигляд:

$$C_{\text{мережа}}(P) = \sum_{i=1}^n (C_{\text{перед}} * \frac{P_i}{V_j} * \tau_i). \quad (1.4)$$

Розглядаємо наступні умови задачі:

- передача даних для трьох пристроїв: 500 МБ, 300 МБ, 700 МБ;
- швидкість передачі: 50 МБ/с, 30 МБ/с, 70 МБ/с;
- вартість передачі: 0,05 грн за МБ.

Загальна вартість дорівнює 1,5 грн.

У ході дослідження також виявляється, що покращення пропускної здатності мережі може призводити до аномальних результатів через зміни в черговості передачі даних.

Ці приклади підкреслюють важливість прогнозування виникнення аномалій, їх виявлення та за можливості усунення у різних прикладних задачах для забезпечення надійності та ефективності систем.

1.4 Висновки до розділу

Задачі оптимального упорядкування вершин орграфів – це підклас задач теорії розкладів. Як в класичних, так і в узагальнених постановках вони виникають при моделюванні процесів, пов'язаних з раціональним використанням ресурсів, зменшенням часових витрат та підвищенням продуктивності. Це зокрема стосується управління виробничими процесами, логістики, складання розкладів

роботи транспорту чи обслуговуючого персоналу. Аналіз таких задач дозволяє виявити оптимальні стратегії, які відповідають як поточним вимогам, так і перспективам розвитку. Упорядкування операцій із врахуванням обмежень, таких як доступність ресурсів чи пріоритети завдань, дає змогу підвищити загальну ефективність системи.

Та як виявилось, ще одним із актуальних напрямків досліджень цих задач є аналіз можливого виникнення аномальних випадків. Дана робота присвячена отриманню теоретичних результатів, що продовжують дослідження в даному напрямку.

Як з'ясувалося аномальні випадки виникають і в інших задачах дискретної оптимізації та в деяких конкретних прикладних сферах. Це стосується у тому числі і задач пакування. Ефективне вирішення задач цього класу дозволяє мінімізувати витрати на транспортування, максимізувати використання складських площ або забезпечити належну безпеку зберігання товарів.

Аномалії, як специфічні відхилення від передбачуваної поведінки, є суттєвим викликом у багатьох прикладних задачах. Їх виявлення, аналіз та врахування мають велике значення для підвищення надійності моделей та прогнозів. У задачах оптимізації аномалії можуть призводити до некоректних рішень, тому важливим є їх своєчасне виявлення або виділення підкласів задач, для яких аномалії не виникають. Наприклад, у системах моніторингу, аналізу даних чи прогнозування, аномалії можуть сигналізувати про несправності, критичні ризики або значні зміни в умовах роботи. Використання динамічних алгоритмів дозволяє автоматизувати процес виявлення аномалій, що особливо важливо для великих обсягів даних.

Таким чином, дослідження, що стосуються виявлення аномалій у певних класах задач дискретної оптимізації підтверджує їх важливість як для наукового надбання, так і для потреб практики. Подальший розвиток методів у цих напрямках сприятиме підвищенню ефективності вирішення задач, що мають значення для сучасної економіки, науки та технологій. У даній роботі такі дослідження проведені для задач паралельного упорядкування вершин орграфів.

Розділ 2 ДОСЛІДЖЕННЯ АНОМАЛЬНИХ ВИПАДКІВ У ЗАДАЧАХ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ

2.1 Основні означення та постановки задач

Сформулюємо задачі, наведені в пункті 1.1, на мові теорії упорядкувань.

Нехай $G=\{V,U\}$ – оргграф, $V=\{1,2,...,n\}$ – множина вершин, U – множина дуг. Тоді, G – модель відповідного технологічного процесу, V – множина робіт, U – множина технологічних обмежень, а $(i,j)\in U$ означає, що робота i повинна бути виконана до початку виконання роботи j .

Наведемо деякі відомі поняття, що використовуються в роботі [78].

Під *упорядкуванням* S скінченної n -елементної множини V , розуміють розміщення елементів цієї множини, при якому кожний елемент розташовується на одному місці.

Зазвичай вважають, що деякі місця можуть бути і порожніми. Якщо не накладаються додаткові умови, то порожні місця не будемо брати до уваги.

За цих умов під *довжиною* $l(S)$ *упорядкування* S розуміється кількість непорожніх місць у ньому.

Нехай $S[p]$ – множина елементів з V , які розташовані в упорядкуванні S на місці p .

Шириною $h(S)$ *упорядкування* S називається кількість елементів найбільшої за потужністю множини $S[p]$.

Серед різних можливих конструкцій, що можна розглядати як упорядкування особливе місце займає спеціальний вид упорядкувань, що дістав назву паралельних упорядкувань вершин оргграфів.

Паралельним упорядкуванням вершин оргграфа G називається таке упорядкування S , яке не порушує порядок слідування вершин в графі, тобто при наявності дуги (i, j) , вершина i повинна розміщуватися лівіше вершини j .

Паралельне упорядкування називається *оптимальним*, якщо воно має

мінімальну ширину при заданій довжині або мінімальну довжину при заданій ширині.

Тобто, задачі 1 і 2 можна сформулювати наступним чином:

Задача 1. Якщо відомий граф технологічних обмежень G та задана довжина упорядкування, то необхідно знайти упорядкування мінімальної ширини.

Задача 2. При відомій ширині упорядкування та заданому графу G технологічних обмежень побудувати паралельне упорядкування з мінімальним значенням l .

2.2 Огляд методів побудови оптимальних упорядкувань

Поліноміальним називається алгоритм, складність якого обмежена поліномом від n . Ці алгоритми називають ефективними.

Експоненціальним називається алгоритм, часова складність якого експоненційно залежить від вхідних даних [50].

Що стосується алгоритмів поліноміальної складності, то перевага віддається тим, складність яких лінійна. Алгоритми, складність яких оцінюється порядком $o(n^4)$, не вважаються ефективними серед алгоритмів цього класу часової складності [79]. Більшість експоненціальних алгоритмів засновані на схемах направленої перебору [80].

Алгоритм називається *точним* для графів із певного класу, якщо він мінімізує цільову функцію для будь-якого графа цього класу. Якщо хоча б для одного графа він дає наближений розв'язок, то алгоритм називається *наближеним* [81].

При аналізі точності наближених алгоритмів використовують наступні оцінки:

$$|l_A(S) - l(S^*)| \leq \varepsilon_1, \quad (2.1)$$

$$\frac{|l_A(S) - l(S^*)|}{l(S^*)} \leq \varepsilon_2, \quad (2.2)$$

$$\frac{l_A(S)}{l(S^*)} \leq \varepsilon_3, \quad (2.3)$$

де $l_A(S)$ – довжина упорядкування, отримана за алгоритмом A , $l(S^*)$ – довжина оптимального упорядкування.

На практиці частіше застосовується остання оцінка, так як вона показує у скільки разів довжина отриманого упорядкування більша за довжину оптимального [82].

До точних методів розв’язання задач паралельного упорядкування відносяться методи, засновані на схемах перебору (метод повного перебору, метод гілок та меж, а також інші методи, що реалізують переборні схеми), та точні алгоритми поліноміальної складності, які розробляються для спеціальних випадків.

Розглянемо схеми перебору.

Процес вибору вершин, що передбачає перебір усіх можливих варіантів у тому ж порядку, в якому вони стають доступними для вибору називається *методом повного перебору*. Тобто, фіксується чергове місце в упорядкуванні (починаючи з i ($i=1, 2, \dots$)) послідовно перебираються усі можливі варіанти розташування вільних вершин на цих місцях). Цей метод має експоненційну складність. Такий підхід не застосовується на практиці, оскільки для більшості NP -важких задач розробляються схеми направленного перебору.

Найбільш поширеним методом, що реалізує такі схеми і дозволяє отримати точне рішення задачі, є *метод гілок та меж* [83]. Основна ідея базується на схемі направленного перебору, яка для наглядності ілюструється побудовою дерева варіантів рішень. У ході роботи виконуються дві процедури: розгалуження і знаходження оцінок для значення цільової функції. Для кожної конкретної задачі дискретної оптимізації цей метод має свою специфіку, яка полягає в способах розбиття всієї допустимої множини розв’язків на відповідні підмножини на кожному кроці.

Якщо задача дискретної оптимізації належить до класу NP -повних, то

виділяють підкласи таких задач, для яких вдається отримати точні алгоритми поліноміальної складності [84]. До такого випадку відносяться задачі, для яких на кожне місце упорядкування можливо поставити не більше двох вершин. Для вирішення цих задач відомі два точні методи поліноміальної складності. Один з них було запропоновано в 1971 році групою японських вчених [51]. Метод базується на пошуку максимальних паросполучень у неорієнтованому графі, який будується за початковим орієнтованим графом. Інший заснований на лексикографічному позначенні вершин графа.

Ідея алгоритму, що базується на максимальних паросполученнях полягає у побудові за заданим графом G неорієнтованого графа G' , в якому знаходяться максимальні паросполучення, що попарно не мають спільних вершин [52]. Далі, за визначеними правилами відбувається процес занесення вершин в упорядкування, видалення та перепозначення паросполучень.

Основна робота другого алгоритму складається з позначення вершин (за допомогою лексикографічно мінімальних послідовностей) та побудови упорядкування (за правилом: спочатку вершини, що не мають вхідних дуг, потім у порядку спадання позначок вершин).

2.3 Вплив покращення початкових умов на виникнення аномальних випадків

Зупинимось на питанні, чи будуть мати місце аномалії, якщо розглядати узагальнення декількох випадків, наприклад:

- 1) одночасного збільшення кількості ресурсів та зменшення часу виконання завдань;
- 2) зменшення часу виконання завдань та послаблення технологічних обмежень;
- 3) послаблення технологічних обмежень та зміни порядку виконання робіт.

Приклад 2.1. Задано зважений орієнтований ациклічний граф G_2 (рис. 2.1),

Довжина оптимального упорядкування $l=19$, тобто при логічному покращенні декількох умов отримали погіршення значення цільової функції.

Розглянемо випадок одночасного зменшення часу виконання завдань та послаблення технологічних обмежень. Видаляємо дуги (i_1, i_6) та (i_2, i_6) , $t_1=2$. Отримуємо граф G'_2 (рис. 2.2).

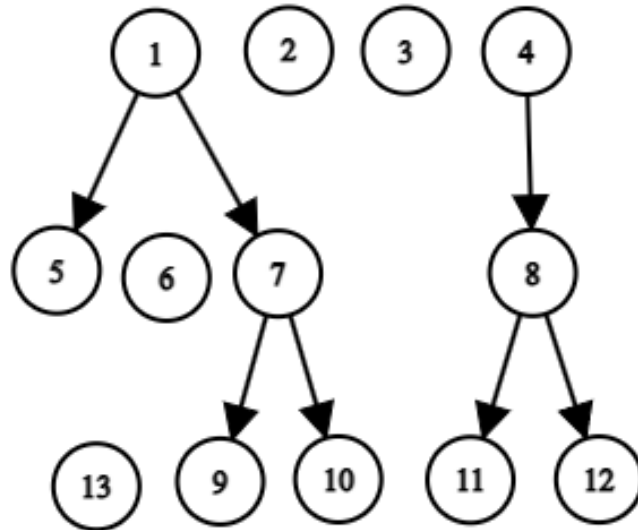


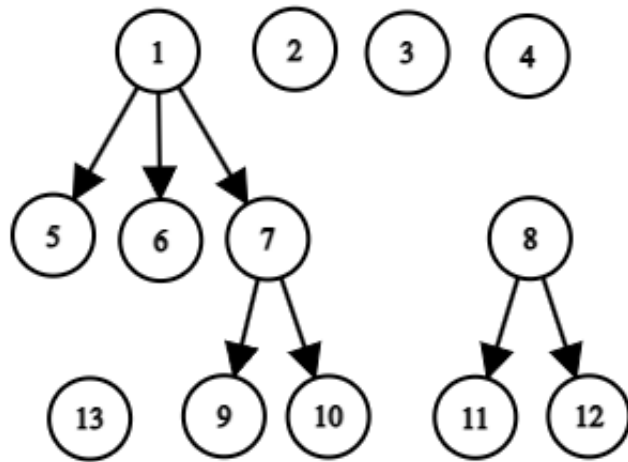
Рис. 2.2. Граф G'_2

Таблиця 2.3 – Оптимальне упорядкування для графа G'_2

μ_1	μ_{11}	μ_{22}
1	1	4	4	7	7	8	8	8	11					
2	2	2	5	5	5	5	9	9	9					
3	3	3	6	6	6	6	6	10	12	13	13

Довжина $l=22$, що на 46,6% більше за довжину оптимального упорядкування вершин графа G_2 .

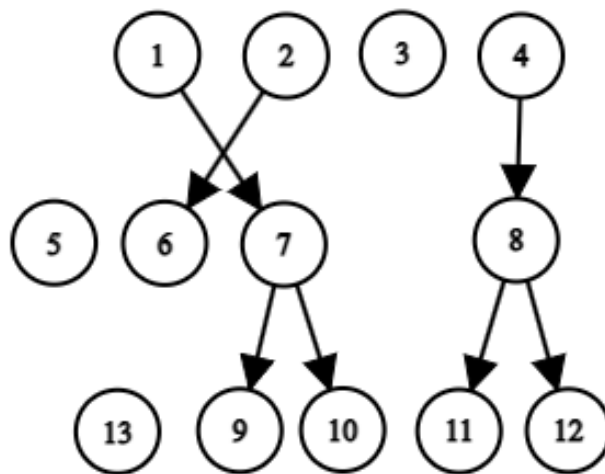
Послабимо технологічні обмеження та змінимо порядок виконання робіт. З графа G_2 видаляємо дуги (i_4, i_8) та (i_2, i_6) , позначаємо як граф G''_2 (рис. 2.3). Нехай $L'=(4,3,2,1,5,6,7,8,11,12,9,10,13)$.

Рис. 2.3. Граф G''_2 Таблиця 2.4 – Оптимальне упорядкування для графа G''_2

4	4	1	1	1	1	1	5	5	5	5				
3	3	3	8	8	8	10	6	6	6	6	6			
2	2	2	11	12	9	9	9	7	7	13	13	13	...	13

$l = 22$, знову спостерігаємо аномальне збільшення довжини упорядкування.

Виникає питання чи будуть мати місце аномалії при одночасному покращенні трьох основних початкових умов. Нехай $h' = 4$, $t_1 = 3$, видаляємо дуги (i_1, i_5) та (i_1, i_6) . Позначаємо граф G'''_2 (рис. 2.4).

Рис. 2.4. Граф G'''_2

Таблиця 2.5 – Оптимальне упорядкування для графа G'''_2

1	1	1	6	6	6	6	6				
2	2	2	7	7	9	9	9				
3	3	3	8	8	8	10	12				
4	4	5	5	5	5	11	13	13	13	...	13

Навіть при одночасному зменшенні часу виконання завдань, послабленні технологічних обмежень та збільшенні кількості виконавців, отримуємо погіршення значення цільової функції: $l=19$.

Грехемом були запропоновані наступні умови для уникнення аномалій при покращенні усіх початкових даних [15]:

1) Збалансованість ресурсів та завдань.

Якщо кількість виконавців збільшується, система повинна прагнути до оптимального розподілу завдань серед доступних виконавців. Завдання повинні бути рівномірно розподілені, щоб уникнути перевантаження деяких виконавців і недовантаження інших.

2) Перевірка технологічних обмежень.

Якщо обмеження на завдання були послаблені, необхідно перевірити, чи не з'явилася надмірна паралельність у виконанні завдань, яка може призвести до перезавантаження одних і недовантаження інших виконавців.

3) Оцінка критичного шляху.

Необхідно постійно слідкувати за значенням довжини критичного шляху при зміні початкових умов. Якщо нові умови викликають зростання часу виконання завдань на критичному шляху, то це може свідчити про порушення оптимальної стратегії.

4) Ітеративна перевірка та коригування.

При кожній зміні умов виконання задач, потрібно перевіряти результат через ітераційний підхід. Оскільки зміни можуть мати накопичувальний ефект, ітерації

дозволяють коригувати стратегію в режимі реального часу.

2.4 Залежність значення цільової функції від змін параметрів упорядкування

Аномальні випадки, які були описані Грехемом, є найбільш вивченими в подальших дослідженнях. Але у ході аналізу, викликаному практичними потребами з'являються нові неочікувані випадки, які також можна розглядати як аномальні, а саме: скорочення списку пріоритетів; зменшення кількості робіт. Розглянемо дані покращення вихідних умов на прикладах.

Приклад 2.2. Задано граф G_3 (рис. 2.5), список пріоритетів $L=(1,2,3,4,5,6,7,8,9)$, множина ваг вершин $T=(4,2,2,4,2,6,4,4,2)$, $h=3$.

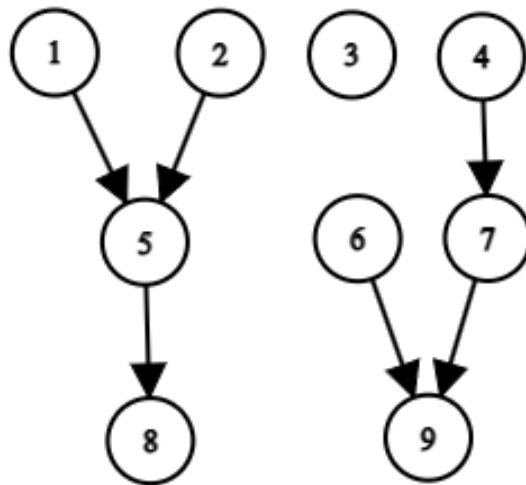


Рис. 2.5. Граф G_3

Знайдемо оптимальне упорядкування (табл. 2.6) та мінімальну довжину l .

Таблиця 2.6 – Оптимальне упорядкування для графа G_3

1	1	1	1	5	5	7	7	7	7
2	2	4	4	4	4	8	8	8	8
3	3	6	6	6	6	6	6	9	9

Довжина $l=10$. Зменшимо тепер список пріоритетів: $L'=(1,4,6)$. Будуємо оптимальне упорядкування (табл. 2.7).

Таблиця 2.7 – Оптимальне упорядкування для графа G_3 при L'

1	1	1	1	2	2	5	5	8	8	8	8
4	4	4	4	3	3	7	7	7	7	9	9
6	6	6	6	6	6						

Довжина $l=12$, тобто зменшення кількості списку пріоритетності вершин призвело до збільшення загального часу виконання робіт.

Зменшимо кількість робіт. $L''=(1,2,4,5,6,7,8,9)$, $T''=(4,2,4,2,6,4,4,2)$. Розглянемо граф G'_3 (рис. 2.6).

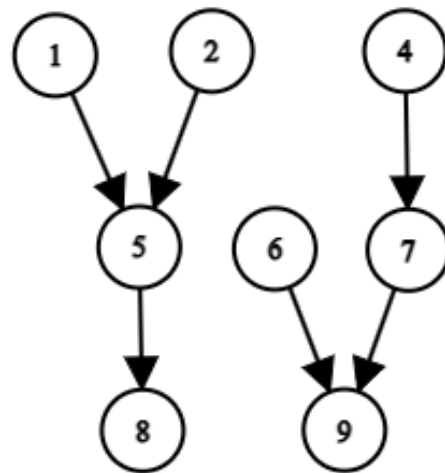


Рис. 2.6. Граф G'_3

Знаходимо оптимальне упорядкування (табл. 3) та мінімальну довжину l для цього випадку.

Таблиця 2.8 – Оптимальне упорядкування для графа G'_3

1	1	1	1	5	5	8	8	8	8
2	2	6	6	6	6	6	6		
4	4	4	4	7	7	7	7	9	9

Довжина $l=11$, тобто інтуїтивно зменшення кількості робіт також призвело до збільшення довжини оптимального упорядкування.

Розглянемо тепер випадок одночасного зменшення кількості робіт та скорочення списку пріоритетів, задаючи наступні умови: граф G'_5 , список пріоритетів $L'=(1,4,6)$ та множину ваг вершин $T''=(4,2,4,2,6,4,4,2)$. Оптимальне упорядкування матиме вигляд:

Таблиця 2.9 – Оптимальне упорядкування для графа G'_3

1	1	1	1	2	2	5	5	8	8	8	8
4	4	4	4	7	7	7	7	9	9		
6	6	6	6	6	6						

Довжина $l=12$, знову виникає аномальне погіршення значення цільової функції.

Розглянемо тепер умови, за яких не виникатимуть аномалії у разі покращення початкових даних. Зупинимось окремо на кожному з наступних випадків:

- 1) скорочення списку пріоритетів;
- 2) зменшення кількості робіт.

Для уникнення аномалій при скороченні списку пріоритетів необхідно дотримуватись виконання наступних умов:

- 1) Умова збереження критичного шляху: усі вершини, що належать критичному шляху, повинні залишатися у списку пріоритетів L' .

Критичний шлях у графі визначається як найдовший шлях за сумарною вагою робіт [85-89]. Якщо видалити з L' хоча б одну вершину критичного шляху, це може призвести до затримок, а відповідно і вплинути на оптимальність упорядкування.

- 2) Умова збереження вершин із суттєвою вагою:

Окрім вершин критичного шляху, у скороченому списку L' повинні залишатись вершини, вага яких не менша за середню вагу вершин графа G . Такі вершини мають вагу $w(v)$, що задовольняє нерівність:

$$w(v) \geq \bar{w}, \quad (2.4)$$

де: $w(v)$ – вага вершини v , $w(v) = t(v)$, \bar{w} – середня вага вершин у графі G .

Вершини, вага яких перевищує середню, також можуть мати значний вплив на загальний час виконання робіт. Тобто, якщо такі вершини не будуть включені до L' , це може призвести до затримки у розкладі, що вплине на загальну тривалість.

3) Умова розташування вершин з урахуванням ваг відповідних шляхів w (P_i): порядок розташування вершин узгоджується з вагою відповідних критичних шляхів, які не повинні збільшуватися.

Нехай P_i – шлях у графі G , що складається з послідовності вершин v_1, v_2, \dots, v_k , де v_1 не має вхідних дуг, а v_k – вихідних. Тоді вага шляху P_i , позначена як $w(P_i)$, визначається сумою ваг його вершин:

$$w(P_i) = \sum_{v_j \in P_i} w(v_j). \quad (2.5)$$

Вершини, які належать до шляхів із великою вагою, є критичними з точки зору пропускну здатності системи. Їх збереження у списку L' дозволяє уникнути перевантаження інших шляхів, зберігаючи топологічний баланс.

Іншим практичним підходом до покращення початкових умов є зменшення кількості робіт, яке передбачає вилучення деяких вершин з графа G . Для уникнення аномалій у цьому випадку необхідно дотримуватись наступних умов:

1) Умова рівномірного перерозподілу завдань: після вилучення деяких вершин необхідно забезпечити рівномірний розподіл завдань між виконавцями. Нерівномірний розподіл робіт може призвести до ситуації, коли деякі виконавці залишаться недозавантаженими, а інші будуть перевантажені, що може призвести до збільшення загального часу виконання робіт.

Для уникнення аномалій необхідно забезпечити таке розподілення, при якому відхилення ваги завдань для кожного виконавця від середньої ваги було мінімальним.

Позначимо навантаження на виконавця i як W_i . Для рівномірного розподілу необхідно мінімізувати дисперсію вагових навантажень:

$$D = \frac{1}{m} \sum_{i=1}^m (W_i - \bar{W})^2, \quad (2.6)$$

де \bar{W} – середнє навантаження на виконавця:

$$\bar{W} = \frac{1}{m} \sum_{i=1}^m W_i. \quad (2.7)$$

Щоб уникнути аномалій, необхідне виконання умови:

$$D \rightarrow \min. \quad (2.8)$$

2) Умова збереження топологічного порядку: видалення завдань повинно зберегти частковий порядок слідування тих вершин, що залишилися таким же як і до видалення вершин [90]. Введемо позначення:

$$v_i \rightarrow v_j, \quad (2.9)$$

яке означає, що в списку пріоритетів L вершина v_i передуює вершині v_j .

Видалення вершини v_k не повинно призводити до порушення порядку, тобто має виконуватися:

$$\forall (v_i, v_k), (v_k, v_j) \in U : V = V \setminus \{v_k\} \Rightarrow v_i \rightarrow v_j. \quad (2.10)$$

Якщо після видалення вершини виникає порушення цієї умови, необхідно провести відповідний перерозподіл залежностей, щоб зберігся вихідний порядок виконання робіт.

2.5 Випадки для інвертованих графів

Одним із цікавих підходів у теорії графів є використання інвертованих (зворотних) графів.

Прямі та інвертовані графи відрізняються напрямком передачі інформації між об'єктами. Якщо у прямому графі інформація передається в одному напрямку, тоді в інвертованому – у протилежному.

Розглянемо кілька прикладів застосування інвертованих графів.

Ранжування веб-сторінок: інвертовані графи допомагають пошуковим системам аналізувати взаємозв'язки веб-сторінок та зворотні посилання, що дозволяє точніше визначати важливість сторінок у результатах пошуку [91].

Маршрутизація трафіку в комп'ютерних мережах: в інформаційних мережах інвертовані графи дозволяють визначити оптимальні маршрути передачі даних з

урахуванням зворотного шляху.

Аналіз текстів та обробка природної мови: зворотні графи можуть використовуватися для вивчення граматики та виявлення зв'язків між словами у текстах [92].

Аналіз інформаційних потоків: інвертовані графи застосовуються для вивчення впливу користувачів у соціальних мережах, таких як блогери та інфлюенсери, що допомагає відстежувати поширення інформації [93].

Виникають наступні питання:

- 1) якщо в прямому графі виникають аномалії при зміні тих чи інших початкових умов, то чи будуть виникати такі ж аномалії для інвертованих графів;
- 2) якщо в прямому графі аномалії не виникають, то чи можуть вони виникнути в інвертованому.

Означення 2.1. Інвертований (зворотний) граф – це граф, який утворюється шляхом зміни напрямку всіх дуг вихідного графа. Граф $G' = \{V, U'\}$ називається інвертованим щодо графа G , якщо:

$$(i, j) \in U \Rightarrow (j, i) \in U'. \quad (2.11)$$

Означення 2.2. Якщо список пріоритетів $L = \{i_1, \dots, i_n\}$ є допустимим для графа G , то інвертований список пріоритетів $L' = \{i_n, \dots, i_1\}$ є допустимим для графа G' .

Розглянемо алгоритм побудови узагальненого паралельного упорядкування для інвертованого графа G' , який є аналогічним побудові паралельного упорядкування для початкового графа G . Логічно припустити, що довжини упорядкувань S та S' , отриманих при розв'язанні задач для графів G і G' , будуть однаковими. Зупинимось на цьому питанні більш детально.

Означення 2.3. Аномалією інвертування називається випадок, коли $l(S') > l(S)$, де $l(S')$ – довжина упорядкування для графа G' з інвертованим списком пріоритетів, а $l(S)$ – для початкового графа G .

Досліджуючи графи, що мають структуру дерева, можна дійти висновку, що аномалії інвертування виникають через зміни у пріоритетах. Для інвертованих графів – це зміни у зворотному списку пріоритетів. Розглянемо структуру графів

детальніше:

Означення 2.4. Критичний шлях у графі – це шлях від початкової вершини до кінцевої, що має максимальну суму ваг серед усіх можливих шляхів.

Аналізуючи структуру критичних шляхів, можна побачити, що після інвертування вершини критичного шляху можуть знижувати пріоритет у порівнянні з іншими вершинами, що призводить до збільшення довжини упорядкування. Це означає, що для запобігання аномаліям необхідно зберігати високий пріоритет для вершин критичного шляху навіть після інвертування.

Твердження 2.1. Аномалії інвертування не виникатимуть, якщо для інвертованого списку пріоритетів L' будуть виконуватися наступні умови:

- 1) вершини критичного шляху, розташовані на початку списку L' , повинні відповідати рівневому принципу;
- 2) інші вершини розташовуються у порядку не збільшення їх ваг, також дотримуючись рівневого принципу;
- 3) вершини, що не мають вхідних та вихідних дуг і вага яких перевищує вагу критичного шляху, повинні мати найвищий пріоритет.

При виконанні цих умов для інвертованих графів аномалії не виникають.

Доведення. Нехай граф $G = \{V, U\}$ містить критичний шлях P , тобто такий шлях, який має максимальну суму ваг серед усіх можливих шляхів у графі. Позначимо цю суму як $w(P)$. Спочатку побудуємо оптимальне упорядкування S для графа G з використанням списку пріоритетів L , де вершини критичного шляху мають високий пріоритет.

При побудові інвертованого графа $G' = \{V, U'\}$ шлях P у графі G стає інвертованим. Всі дуги змінюють свій напрямок, і тепер вершини, що були початковими у G , стають кінцевими, і навпаки.

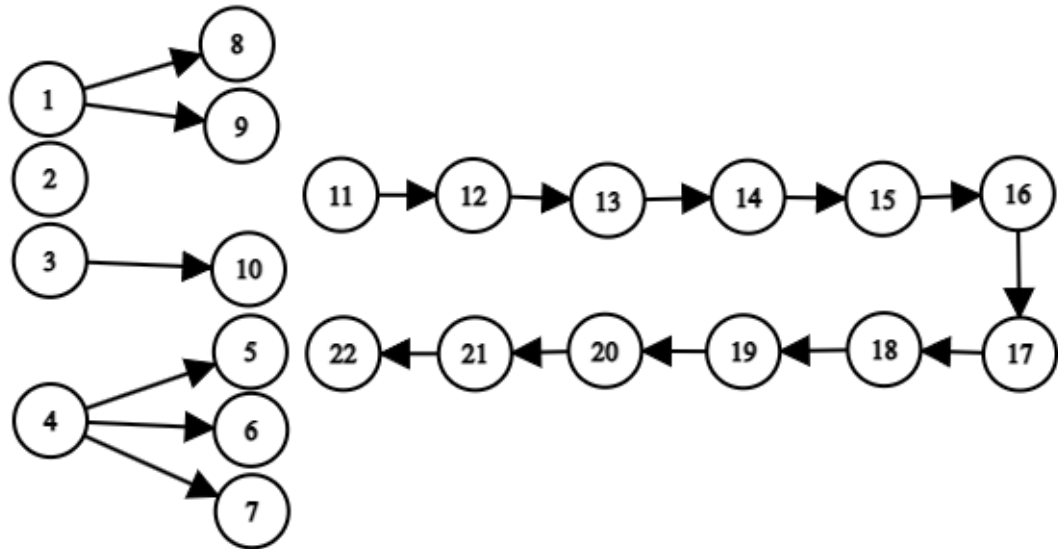
За умовою, вершини критичного шляху P повинні розташовуватися на початку інвертованого списку L' , дотримуючись рівневого принципу. Розміщення вершин критичного шляху на початку списку L' гарантує, що ці вершини будуть обрані першими при побудові упорядкування S' для графа G' . Це дозволяє зберегти ту саму або схожу структуру критичного шляху в G' , яка була у G . Тобто,

максимальна сумарна вага залишиться тією ж, оскільки вершини з високою вагою і високим пріоритетом залишаються на критичних позиціях. Вершини, що не належать до критичного шляху, розташовуються за зменшенням ваг, дотримуючись рівневого принципу. Це зменшує можливість зміщення критичних вершин вправо у списку L' , що могло б призвести до збільшення довжини розкладу.

Якщо існують вершини, що не мають вхідних та вихідних дуг і мають вагу більшу, ніж критичний шлях, вони мають бути розташовані на початку списку L' . Це пояснюється тим, що ці вершини можуть мати незалежний вплив на довжину розкладу і тому, їх виконання повинно починатися якомога раніше. Такий підхід дозволяє уникнути ситуації, коли вершини з високою вагою зміщуються вправо у списку, що призводить до затримок у виконанні наступних завдань.

Таким чином, побудова інвертованого списку пріоритетів L' за вказаними умовами гарантує, що довжина розкладу $l(S')$ для графа G' не перевищить довжину розкладу $l(S)$ для графа G . Критичні вершини зберігають свій пріоритет, а інші вершини розташовуються так, щоб мінімізувати можливий вплив на загальну довжину розкладу. Отже, аномалія інвертування не виникатиме, оскільки ми запобігаємо ситуації, коли критичний шлях стає менш пріоритетним, ніж інші шляхи. Таким чином, доведено, що при дотриманні зазначених умов інвертований граф G' не буде демонструвати аномалії інвертування, що дозволяє зберегти ефективність початкового розкладу.

Приклад 2.3. Нехай задано орієнтований граф G_3 (рис. 2.7), $h=3$,
 $L_3=(1,2,3,4,10,11,12,13,14,15,16,17,18,19,20,21,22,5,6,7,8,9)$,
 $T_3=(5,3,3,2,3,3,2,5,4,9,1,1,1,1,1,1,1,1,1,1,1)$.

Рис. 2.7. Граф G_4

Знаходимо оптимальне упорядкування (табл. 2.10).

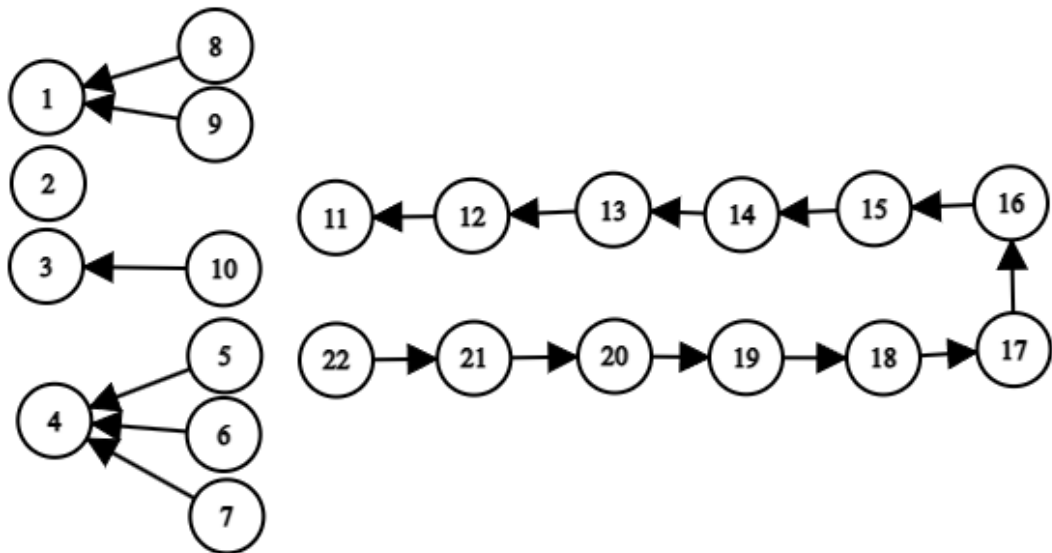
Таблиця 2.10 – Оптимальне упорядкування для графа G_4

1	1	1	1	1	11	12	13	14	15	16	17	18	19	20	21	22
2	2	2	4	4	5	5	5	6	6	6	7	7	9	9	9	9
3	3	3	10	10	10	10	10	10	10	10	10	8	8	8	8	8

Довжина $l=17$.

Знайдемо тепер оптимальне упорядкування та його довжину для інвертованого графа G'_3 (рис. 2.8). Задано наступні параметри: $h=3$, $L'_3=(9,8,7,6,5,22,21,20,19,18,17,16,15,14,13,12,11,10,4,3,2,1)$,

$T_3=(5,3,3,2,3,3,2,5,4,9,1,1,1,1,1,1,1,1,1,1,1,1)$.

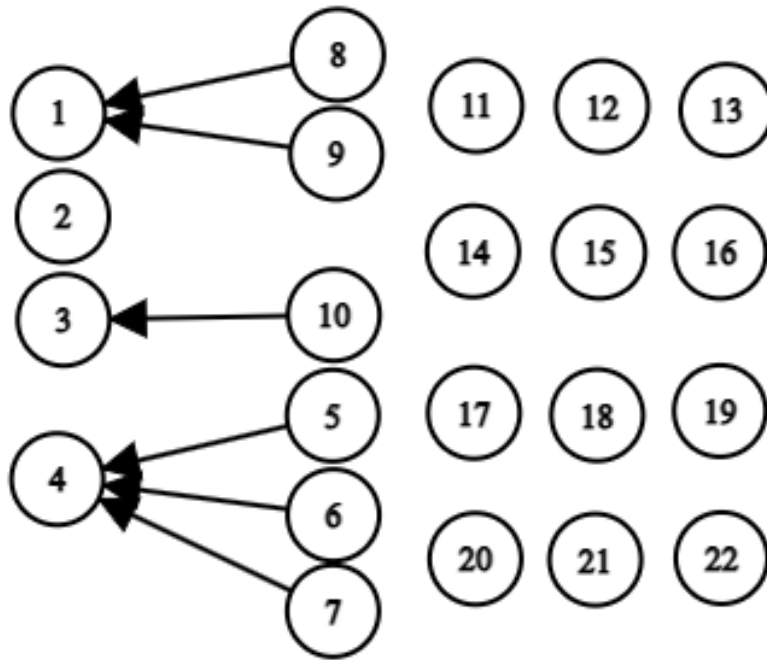
Рис. 2.8. Граф G'_4

Знаходимо оптимальне упорядкування (табл. 2.11).

Таблиця 2.11 – Оптимальне упорядкування для графа G'_3 та L'_3

9	9	9	9	5	5	5	20	19	18	17	16	15	14	13	12	11
8	8	8	8	8	22	21	4	4	2	2	2	1	1	1	1	1
7	7	6	6	6	10	10	10	10	10	10	10	10	10	3	3	3

Довжина $l=17$, що дорівнює довжині оптимального упорядкування вершин прямого графа. Перевіримо чи буде аномальне збільшення довжини при покращенні початкових умов для зворотного графа G'_3 . Видаляємо усі дуги з підграфа, що включає вершини з 22-ої до 11-ої. Отримуємо граф G''_3 (рис. 2.9).

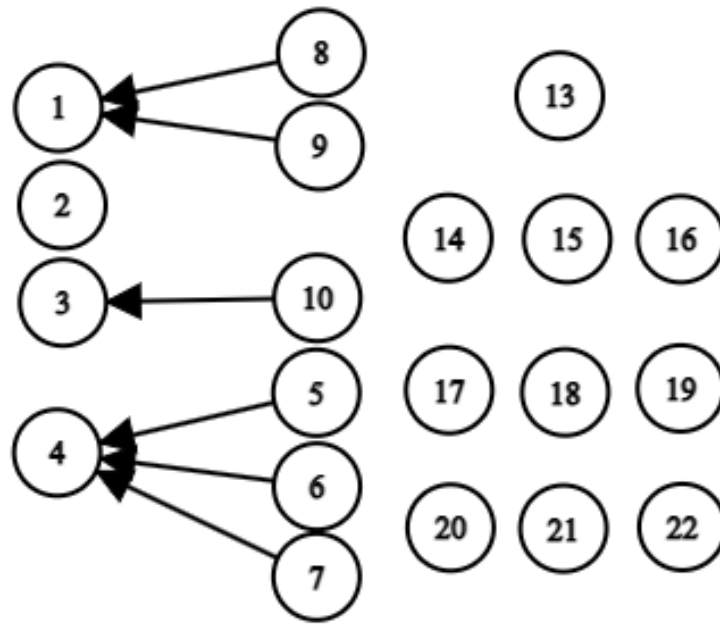
Рис. 2.9. Граф G''_3

Побудуємо оптимальне упорядкування (табл. 2.12).

Таблиця 2.12 – Оптимальне упорядкування для графа G''_3

9	9	9	9	5	5	5	18	15	12	4	4	1	1	1	1	1		3	3	3
8	8	8	8	8	22	20	17	14	11	2	2	2								
7	7	6	6	6	21	19	16	13	10	10	10	10	10	10	10	10	10			

Довжина збільшилась, $l=21$, можемо казати, що аномалія має місце і для інвертованого графа. Подивимось чи виникатиме аномалія при одночасному покращенні декількох умов. Видаляємо з графа G'_3 вершини i_{12} та i_{11} , $t_1=2$, видаляємо дуги з підграфа, що включає вершини з 22-ої до 13-ої (рис. 2.10).

Рис. 2.10. Граф G'''_3

Будуємо оптимальне упорядкування (табл. 2.13).

Таблиця 2.13 – Оптимальне упорядкування для графа G'''_3

9	9	9	9	5	5	5	18	15	10	10	10	10	10	10	10	10	10	3	3	3
8	8	8	8	8	22	20	17	14	4	4	1	1								
7	7	6	6	6	21	19	16	13	2	2	2									

Спостерігаємо збільшення довжини, $l=21$. Таким чином, для зворотного графа мають місце ті ж самі аномалії, що і для прямого графа.

2.6 Аномальні випадки в задачах паралельного виконання програм

У контексті сучасних паралельних обчислень, зокрема з використанням мікропроцесорів, також виникають аномалії, пов'язані зі зміною часу виконання програм. Ці аномалії вказують на ситуації, коли час виконання інструкцій виявляється нелінійно залежним від їх порядку або ресурсів, що використовуються. Розгляд таких випадків є важливим для забезпечення

коректності та оптимізації роботи програм у багатоядерних і паралельних обчислювальних системах.

У статті [16] детально проаналізовано випадки аномального збільшення часу виконання інструкцій у динамічно розподілених мікропроцесорах. Дослідження демонструє, що в таких системах найгірший час виконання інструкцій (WCET, Worst-Case Execution Time) не завжди відповідає найбільш несприятливій поведінці системи. Наприклад, у деяких випадках пропуск у кеш-пам'яті (коли дані не знайдені в кеші та мають бути завантажені з пам'яті нижчого рівня) може призводити до коротшого часу виконання, ніж ситуація, коли дані вже знаходяться в кеші.

Аномалії в задачах паралельного виконання програм переважно виникають через архітектурні особливості динамічно розподілених процесорів, які виконують інструкції поза порядком їхнього виклику в програмі.

Розглянемо основні причини, що зумовлюють такі аномалії:

1) Ресурси, виділені поза порядком програми. Динамічно розподілені мікропроцесори здійснюють розподіл ресурсів, таких як функціональні блоки, кеш-пам'ять і реєстри, в динамічному режимі. Це означає, що виконання інструкцій може не відповідати їхньому початковому порядку у програмі. Такий підхід дозволяє підвищити продуктивність за рахунок оптимізації використання апаратних ресурсів, проте водночас створює можливості для виникнення аномалій.

2) Залежності між інструкціями. Інструкції програми часто мають логічні або часові залежності, які визначають їхній порядок виконання. У динамічно розподілених процесорах ці залежності можуть бути розірвані, дозволяючи виконання інструкцій у непослідовному порядку. Однак це може призводити до непередбачуваних змін у часі виконання, наприклад, через конфлікти в доступі до ресурсів або затримки через врахування залежностей.

3) Вплив кеш-пам'яті та інших факторів. Особливості функціонування кеш-пам'яті також відіграють ключову роль у виникненні аномалій. Наприклад, процес завантаження даних у кеш може змінювати порядок доступу до пам'яті, що

впливає на загальний час виконання інструкцій. Додаткові фактори, такі як резервування функціональних блоків чи регістрів, також можуть спричиняти аномалії.

4) Нестабільність часових залежностей. Як показують результати дослідження [16], деякі аномалії можуть виникати через відмінності у доступі до ресурсів, зокрема коли одна й та сама інструкція в різних сценаріях виконання може споживати різний обсяг ресурсів.

Дослідження [16] також визначає конкретні архітектурні особливості, які сприяють виникненню аномалій у часі виконання інструкцій. До них належать:

- 1) використання багатоядерної архітектури з динамічним розподілом ресурсів;
- 2) значна складність логіки управління кеш-пам'яттю;
- 3) високий ступінь залежності між інструкціями програми.

Окрім цього, автори ідентифікують вплив програмних факторів, таких як спосіб організації коду та рівень оптимізації компілятора. Цікаво, що прості техніки модифікації коду можуть суттєво зменшити або повністю усунути вплив аномалій. Такі техніки дозволяють уникнути недооцінки WCET, хоча, як зазначається у роботі, навіть у разі їх недооцінки помилка рідко перевищує 27%.

Таким чином, аномалії в задачах паралельного виконання програм створюють суттєві виклики для аналізу найгіршого варіанту часу виконання (WCET). Традиційні методи аналізу, які базуються на припущеннях щодо фіксованих ресурсів і порядку виконання інструкцій, виявляються неефективними через часову складність. З огляду на те, що аномалії можуть призводити до некоректних оцінок WCET, у майбутніх дослідженнях доцільно звернути увагу на розробку ефективних алгоритмів, які враховують архітектурні особливості процесорів та мінімізують вплив аномалій.

Серед архітектурних особливостей процесорів слід звернути увагу на:

- 1) Механізми позачергового виконання інструкцій (out-of-order execution).

Цей підхід дозволяє процесорам виконувати інструкції не в тому порядку, в якому вони задані в програмі, з метою максимального використання

функціональних блоків та уникнення простоїв [94]. Однак це може створити потенційний ризик аномалій, коли інструкції, що залежать одна від одної, змінюють свій порядок виконання, що може вплинути на загальний час виконання.

2) Спекулятивне виконання (speculative execution).

Ця технологія дозволяє процесорам передбачати наступні інструкції, які потрібно виконати, і виконувати їх до підтвердження правильності прогнозу [95]. У разі помилкових прогнозів спекулятивно виконані інструкції мають бути відкинуті, що може призводити до додаткових затримок та непрогнозованому збільшенні часу виконання.

3) Механізми попередньої вибірки даних (prefetching).

Попередня вибірка дозволяє завантажувати дані в кеш до того, як вони будуть потрібні програмі. Хоча це значно прискорює доступ до даних у більшості випадків, невдалі передбачення або конфлікти в кеші можуть також спричиняти непередбачувані затримки [96].

4) Мультипрограмування та конкурентний доступ до спільних ресурсів.

У багатоядерних системах ядра можуть конкурувати за доступ до кешу, пам'яті або інших спільних ресурсів. Це створює багатоваріантність у часі доступу до цих ресурсів, що може змінювати продуктивність і викликати аномалії.

5) Інструкції з залежностями від даних (data-dependent instructions).

Вплив порядку виконання інструкцій, які мають залежність від результатів попередніх операцій, може змінюватися залежно від обраної динаміки виконання. Такі інструкції можуть чекати завершення попередніх обчислень або конфліктувати через обмеженість ресурсів [97-99].

2.7 Висновки до розділу

У даному розділі наводяться деякі відомі постановки задач паралельного упорядкування вершин орієнтованих графів та огляд методів знаходження оптимальних розв'язків. Зокрема, розглянуто класичні постановки, пов'язані з

мінімізацією довжини упорядкування при заданій ширині та побудова упорядкувань мінімальної ширини при заданій довжині. Оскільки, розглядаються задачі, що відносяться до класу задач дискретної оптимізації, які є NP-важкими, то для них пошук нових алгоритмів поліноміальної складності для певних підкласів графів, або для фіксованих значень параметрів упорядкувань на сьогоднішній день є актуальним.

Відмічено доцільність застосування схем направленого перебору для отримання наближених розв'язків за прийнятний обчислювальний час.

Основна увага у розділі приділяється питанням пов'язаним з так званими аномальними випадками в задачах, що розглядаються.

Досліджено умови уникнення аномальних випадків при зміні початкових параметрів, таких як зменшення кількості робіт, скорочення списку пріоритетів або послаблення технологічних обмежень. Запропоновано критерії збалансованості ресурсів, збереження критичного шляху, рівномірного перерозподілу ваг, які дозволяють уникати аномалій.

Виявлено та описано додаткові аномальні випадки, пов'язані з практичними сценаріями, які раніше не були враховані в дослідженнях, а саме: скорочення списку пріоритетів та видалення деяких завдань.

Також досліджено підклас інвертованих графів, які застосовуються для задач, що враховують зворотні зв'язки між об'єктами, наприклад у пошукових системах або маршрутизації трафіку. Вивчення таких графів дозволяє моделювати та оптимізувати складні системи зі взаємозалежними елементами.

Отримані результати можуть бути застосовані для аналізу задач планування технологічних процесів, маршрутизації, управлінні проєктами та інших галузях, що потребують оптимального розподілу ресурсів та ефективного виконання завдань у багатозадачному середовищі. Подальших досліджень потребує виділення нових підкласів графів, для яких можуть існувати точні алгоритми поліноміальної складності, а також розробка наближених методів для задач великої розмірності з апіорної оцінкою їх точності. Особливу увагу слід приділити практичному застосуванню інвертованих графів.

Враховуючи зроблені висновки, подальший аналіз спрямований на вивчення комбінованих випадків змін початкових даних, їх впливу на виникнення аномалій у задачах паралельного упорядкування та деяких спеціальних питань.

Розділ 3 ВИДИ АНОМАЛІЙ ТА УМОВИ ЇХ ВИНИКНЕННЯ ДЛЯ СПЕЦІАЛЬНИХ КЛАСІВ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ

3.1 Умови виникнення аномалій в деяких класах задач

У випадку коли всі виконавці мають однакову продуктивність задачу рівномірного розподілу завдань між виконавцями розв'язати простіше. Якщо на рівномірний розподіл і буде щось суттєво впливати, то це структура самого графа. Для таких задач основним є такий розподіл, який зменшує загальний час виконання [100-101]. Якщо продуктивності виконавців різні, то забезпечення рівномірного розподілу завдань стає більш складнішою задачею. Для цифрових обчислювальних пристроїв це може бути через різні апаратні можливості, кількість ядер, об'єми пам'яті або інші фактори. Тут ціль – ефективно розподіляти завдання між різними виконавцями, враховуючи різну продуктивність таким чином, щоб одночасно максимізувати загальну продуктивність та мінімізувати загальний час завершення операцій.

Означення 3.1. Швидкість (продуктивність) s_k k -го виконавця у задачах планування визначає з якою ефективністю виконавець може виконувати завдання. Це цілочисельна характеристика, яка визначає у скільки разів швидше виконавець може завершити завдання порівняно з еталонним виконавцем зі стандартною швидкістю $s_e=1$.

Загальний час виконання j -го завдання k -м виконавцем зі швидкістю s_k дорівнює:

$$t_{k,j} = \frac{\tau_j}{s_k}. \quad (3.1)$$

Твердження 3.1. Якщо завдання між виконавцями розподілені нерівномірно, то збільшення продуктивності s_k k -го виконавця може призводити до аномального збільшення значення цільової функції.

Доведення. Позначимо через I_k ($I_k \subseteq I$) множину завдань, які виконуються k -м виконавцем.

Тоді сумарний час виконання завдань k -м виконавцем дорівнює:

$$T_k = \frac{1}{s_k} \sum_{j=1}^n \tau_j. \quad (3.2)$$

Загальний час завершення всіх завдань T_{total} можна оцінити наступним чином:

$$T_{total} \geq \max_k T_k. \quad (3.3)$$

Якщо завдання нерівномірно розподілені між виконавцями, наприклад, виконавець з меншою продуктивністю отримує багато завдань або завдання з великими часами виконання, це може призвести до збільшення загального часу виконання, незважаючи на збільшення продуктивності інших виконавців.

Отже, загальний час може збільшуватися навіть за умови збільшення продуктивності окремих виконавців.

Нехай для кожного k -го виконавця множина завдань $I_k \subseteq I$ розподілена таким чином, що сума навантажень на всіх виконавців є збалансованою.

Твердження 3.2. Оптимальний розподіл завдань забезпечується тоді, коли навантаження для кожного виконавця приблизно рівні, тобто:

$$T_k \approx T_j, \forall k, j = \overline{1, m}. \quad (3.4)$$

Доведення. Якщо навантаження на k -го виконавця значно більше, ніж на j -го, то k -й виконавець стане "вузьким місцем", і загальний час виконання всього розкладу визначатиметься саме його часом T_k .

Щоб уникнути цього, необхідно досягти певної збалансованості, при якій виконується умова (3.4).

Припустимо, що умова (3.4) не виконується: існують такі виконавці k і j , що $T_k \gg T_j$. Це означає, що час завершення всіх завдань для виконавця k набагато більший, ніж для виконавця j , і існує значна нерівномірність у розподілі завдань між виконавцями.

Через те, що $T_k \gg T_j$, виконавець k стає "вузьким місцем" у системі. Інші виконавці завершують свої завдання раніше і простоюють, поки виконавець k ще

виконує свої завдання. Це призводить до неефективного використання ресурсів і збільшення часу виконання завдань T_{total} . Таким чином, розподіл завдань, при якому T_k значно перевищує T_j для деяких k і j , є неоптимальним, оскільки не мінімізує загальний час виконання завдань. Цей розподіл не забезпечує баланс між навантаженнями виконавців і призводить до простоїв, що є ознакою неефективності.

Розглянемо тепер випадок, коли виконується умова (3.4): навантаження між усіма виконавцями приблизно рівне, тобто $T_k \approx T_j$ для всіх $k, j = 1, \dots, m$. При такому розподілі навантаження серед виконавців максимальний час виконання серед усіх виконавців, який визначає загальний час виконання T_{total} , буде мінімізований. Це досягається через балансування навантаження, при якому немає "вузьких місць" і ресурси використовуються ефективніше. Отже, при виконанні умови (3.3) загальний час виконання T_{total} зменшується у порівнянні з випадком, коли навантаження розподілене нерівномірно. Це доводить, що рівномірний розподіл навантаження є оптимальним, оскільки мінімізує час виконання всіх завдань.

Таким чином, оптимізація розподілу завдань передбачатиме рівномірний розподіл навантаження з урахуванням швидкості кожного виконавця.

Для того, щоб мінімізувати випадки виникнення аномалій, необхідне виконання наступних трьох умов:

1) Призначення завдань: Відсортувати завдання на основі списку пріоритетів і технологічних обмежень. Призначати завдання виконавцям так, щоб виконавці за найбільшою продуктивністю отримували завдання з найдовшим часом виконання (у класичному сенсі), оскільки це максимізує перевагу швидкості.

2) Балансування навантаження: Забезпечити рівномірне навантаження між виконавцями, уникаючи вузьких місць, де менш продуктивні виконавці створюють затримки.

3) Динамічне планування: Періодично переглядати упорядкування після завершення завдань і перерозподіляти решту завдань, щоб забезпечити повне використання більш продуктивних виконавців і уникнути затримок з боку менш

продуктивних [102, 103].

Основний алгоритм розподілу завдань із використанням продуктивностей виконавців:

1) Вхідні дані.

Множина завдань $V = \{v_1, v_2, \dots, v_n\}$, де кожне завдання має певний час виконання τ_i .

Кількість виконавців m , де кожен виконавець має свою швидкість (продуктивність) s_k .

Список пріоритетів L .

Орієнтований ациклічний граф $G=(V,U)$, що описує обмеження на порядок виконання завдань.

2) Умови оптимального перерозподілу.

3) Основний алгоритм:

Крок 1: Ініціалізація

Створити пустий розподіл для кожного виконавця: $V_k = \emptyset$ для всіх $k=\overline{1, m}$.

Визначити початковий набір завдань, які не мають попередників у графі $G=(V,U)$.

Крок 2: Призначення завдань

Сортування завдань: відсортувати завдання за спаданням їх пріоритетів та тривалості часу виконання.

Призначення за пріоритетом: починаючи з найвищого пріоритетного завдання, призначити його тому виконавцю, який має найменший поточний час виконання T_k .

Адаптація для продуктивності: час виконання завдання j для k -го виконавця коригується за формулою:

$$t_{k,j} = \frac{\tau_j}{s_k}. \quad (3.5)$$

Додати завдання до списку завдань k -го виконавця, оновивши значення T_k для цього виконавця.

Крок 3: Балансування навантаження

Перевірка балансу: Після призначення всіх завдань виконати перевірку балансу для всіх виконавців. Якщо один із виконавців має значно більший час завершення T_k , ніж інші, виконати перерозподіл завдань.

Перерозподіл: Перемістити завдання з перевантаженого виконавця на іншого виконавця з меншою сумою T_j , дотримуючись обмежень пріоритету і технологічних обмежень.

Крок 4: Динамічне планування

Оновлення стану: Після завершення кожного завдання переглянути стан всіх виконавців.

Розподіл нових завдань: Додавати нові завдання для виконання відповідно до їх пріоритетів і технологічних обмежень у графі.

Адаптивне перепризначення: Якщо час завершення одного з виконавців значно менший, ніж в інших, перерозподілити невиконані завдання для оптимального використання всіх ресурсів.

Розглянемо рівномірну продуктивність виконавців на прикладі.

Приклад 3.1. Задано граф G_4 (рис. 3.1), $h=3$, список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10)$, множина ваг вершин $T=(4,2,2,2,4,1,1,1,1,6)$, швидкість виконавців однакова $s=(1,1,1)$.

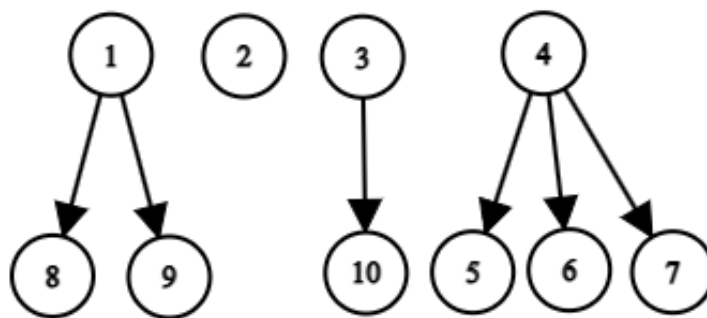


Рис.3.1. Граф G_4

Знайдемо оптимальне упорядкування та його довжину.

Таблиця 3.1 – Оптимальне упорядкування для графа G_4

1	1	1	1	5	5	5	5
2	2	4	4	6	7	8	9
3	3	10	10	10	10	10	10

Довжина $l_1=8$.

Змінимо продуктивність одного з виконавців, нехай $s=(2,1,1)$.

Таблиця 3.2 – Оптимальне упорядкування для графа G_4 при $s=(2,1,1)$

1	1	4	5	5					
2	2	8	6	10	10	10	10	10	10
3	3	9	7						

Довжина $l_2=10$, виникає аномалія.

Розглянемо тепер узагальнення кількох випадків, а саме виникнення аномалій при одночасному виконанні кількох умов.

Збільшуємо продуктивність першого виконавця $s=(2,1,1)$ та кількість виконавців $h=4$.

Таблиця 3.3 – Оптимальне упорядкування для графа G_4 при $s=(2,1,1)$, $h=4$

1	1	5	5					
2	2	6	9					
3	3	7	10	10	10	10	10	10
4	4	8						

Довжина $l_3=9$.

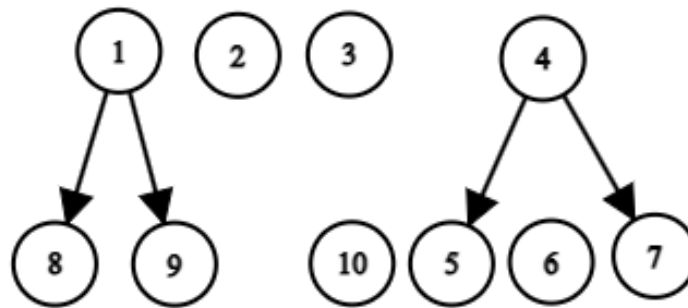
Збільшуємо продуктивність першого виконавця $s=(2,1,1)$ та скорочуємо час виконання завдань $T'=(2,2,2,2,2,1,1,1,1,6)$.

Таблиця 3.4 – Оптимальне упорядкування для графа G_4 при $s=(2,1,1)$, T'

1	4	5	8					
2	2	6	9					
3	3	7	10	10	10	10	10	10

Довжина $l_4=9$.

Збільшуємо продуктивність першого виконавця $s=(2,1,1)$ та послаблюємо обмеження пріоритету (рис. 3.2).

Рис.3.2. Граф G'_4

Будуємо оптимальне упорядкування та знаходимо його довжину.

Таблиця 3.5 – Оптимальне упорядкування для графа G'_4 при $s=(2,1,1)$

1	1	4	5	5					
2	2	6	7	10	10	10	10	10	10
3	3	8	9						

Довжина $l_5=10$, значення цільової функції знову збільшується.

Виникає питання, чи буде відбуватися аномальне погіршення довжини оптимального впорядкування, коли всі проілюстровані вище умови виконуються одночасно.

Нехай задано граф G'_4 . Збільшуємо продуктивність першого виконавця та кількість виконавців, зменшуємо час виконання деяких завдань та спрощуємо

технологічні обмеження.

Таблиця 3.6 – Оптимальне упорядкування для графа G_4 при $s=(2,1,1,1)$, $h=4$, $T'=(3,2,2,2,3,1,1,1,1,6)$

1	1; 8	5	5; _					
2	2	6						
3	3	7	10	10	10	10	10	10
4	4	9						

Довжина $l_6=9$.

Аналізуючи дані випадки, виникає питання чи будуть мати місце аномалії, якщо нерівномірна продуктивність буде задана як початкова умова. Проілюструємо на прикладі.

Приклад 3.2. Задано граф G_5 (рис. 3.3), $h=3$, список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10)$, множина ваг вершин $T=(5,3,6,2,4,4,6,2,2,10)$, продуктивність виконавців $s=(1,1,2)$.

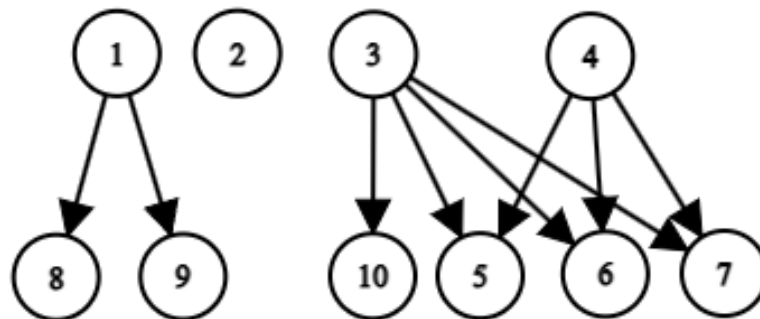


Рис.3.3. Граф G_5

Таблиця 3.7 – Оптимальне упорядкування для графа G_5

1	1	1	1	1	5	5	5	5	8	8
2	2	2	4	4	6	6	6	6	9	9
3	3	3	10	10	10	10	10	7	7	7

Довжина оптимального упорядкування $l_7=11$.

Розглянемо появу аномальних випадків. Збільшимо кількість виконавців,

нехай $h=5$, $s=(1,1,2,1,1)$.

Таблиця 3.8 – Оптимальне упорядкування для графа G_5 при $h=5$, $s=(1,1,2,1,1)$

1	1	1	1	1	8	8						
2	2	2	5	5	5	5						
3	3	3	6	6	9							
4	4		7	7	7	7	7	7				
			10	10	10	10	10	10	10	10	10	10

Довжина для даного випадку $l_8=13$.

Зменшимо час виконання завдань, нехай $T=(3,3,4,2,4,4,6,2,2,8)$.

Таблиця 3.9 – Оптимальне упорядкування для графа G_5 при $T=(3,3,4,2,4,4,6,2,2,8)$

1	1	1	5	5	5	5	9	9						
2	2	2	6	6	6	6	10	10	10	10	10	10	10	10
3	3	4	7	7	7	8								

Довжина збільшується, $l_9=15$.

Послаблюємо технологічні обмеження (рис. 3.4), позначаємо як граф G'_5 .

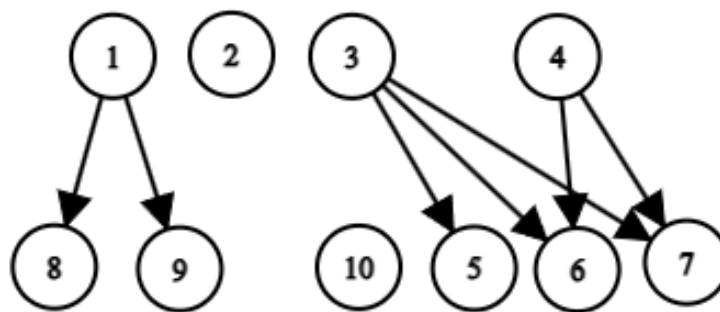


Рис.3.4. Граф G'_5

Таблиця 3.10 – Оптимальне упорядкування для графа G'_5

1	1	1	1	1	6	6	6	6			
2	2	2	4	4	7	7	7	7	7	7	
3	3	3	5	5	8	9	10	10	10	10	10

Довжина $l_{10}=12$. Таким чином, знову спостерігаємо погіршення значення цільової функції при покращенні вихідних параметрів.

Для запобігання виникнення аномалій важливо ретельно розподіляти завдання та використовувати адаптивні стратегії розподілу ресурсів.

Розглянемо більш детально невиконання умов для наведених вище прикладів.

Якщо завдання між виконавцями розподілені нерівномірно, збільшення продуктивності одного з виконавців може призвести до погіршення цільової функції. В упорядкуванні з табл. 3.2 (G_4 , $s=(2,1,1)$) продуктивність першого виконавця збільшено, але довжина зростає з 8 до 10. В упорядкуванні вершин графа G'_4 , за умови $s=(2,1,1)$ аналогічно, довжина знову зростає до 10. Тобто, аномалія спостерігається через надмірне навантаження виконавців з меншою продуктивністю та блокування вершин критичного шляху.

Порушення балансу: якщо навантаження на одного з виконавців значно перевищує навантаження іншого ($T_k \gg T_j$), це створює «вузьке місце» та збільшує загальний час. Оптимальне упорядкування для графа G_5 , $h=5$, $s=(1,1,2,1,1)$: критичний шлях повністю сконцентрований у третього виконавця (із $s=2$), а інші простоюють. Довжина зростає з 11 до 13. Аналогічно з прикладом у таблиці 3.9: аномалія посилюється — довжина ще збільшується до 15. Як результат незбалансоване навантаження призводить до негативного ефекту.

Також варто відмітити, що послаблення технологічних обмежень не гарантує покращення значення цільової функції. Приклади у табл. 3.5 (для G'_4) та табл. 3.10 (для G'_5): попри видалення дуг, яке призводить до спрощення графів, довжина зростає, $l_6=10$ і $l_7=12$ відповідно.

Як показано вище в прикладі 3.1 з початковими умовами: задано граф G_4 (рис. 3.1), $h=3$, список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10)$, множина ваг вершин $T=(4,2,2,2,4,1,1,1,1,6)$, при продуктивності виконавців $s=(1,1,1)$ отримуємо упорядкування довжини 8. А при покращенні продуктивності першого виконавця згідно з вектором $s=(2,1,1)$, довжина збільшується до 10.

Змінімо список пріоритетів згідно до алгоритму побудови оптимального

списку пріоритетів.

Знаходимо загальний час виконання завдань, що лежать на відповідних шляхах, дорівнює довжині цих шляхів: $l_{1-8} = 5, l_{1-9} = 5, l_2 = 2, l_{3-10} = 8, l_{4-5} = 6, l_{4-6} = 3, l_{4-7} = 4, l_{4-7} = 3$.

Оновлюємо список пріоритетів для виконавців, що мають більшу продуктивність: $L' = (3, 10, 4, 5, 1, 8, 9, 6, 7, 2)$. Тобто для 1-го виконавця ми будемо використовувати L' .

Знайдемо оптимальне упорядкування та його довжину для L' та $s=(2, 1, 1)$.

Таблиця 3.11 – Оптимальне упорядкування для графа G_6 при L' та $s=(2, 1, 1)$

3	10	10	10	5	5
1	1	1	1	6	8
2	2	4	4	7	9

Довжина $l=6$, отримали значення на 40% менше за початкову довжину упорядкування.

3.2 Випадки взаємно однозначної відповідності між певними завданнями та виконавцями

Розглянемо ще одне актуальну задачу, а саме: розширення виробництва. При покупці нового обладнання бувають випадки, коли на нових пристроях неможливо виконати деякі завдання, або, навпаки, може виникнути необхідність виконати новий вид робіт, який неможливо виконати на старому обладнанні. У цьому випадку вводиться додаткова умова фіксації деяких робіт за певними виконавцями.

Означення 3.2. Множина зафіксованих робіт $V_r \subseteq V$ визначає ті роботи, які можуть виконуватися лише r -им виконавцем. Це означає, що роботи $i \in V_r$ можуть виконуватися виключно r -им виконавцем, але сам виконавець може виконувати й

інші роботи, що не входять до V_r .

Приклад 3.3. Нехай задано граф G_4 (рис. 3.1), $h=3$, список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10)$, множина ваг вершин $T=(4,2,2,2,4,1,1,1,1,6)$. Зафіксуємо роботи j_3 та j_{10} за третім виконавцем: $V_3=(j_3, j_{10})$. Оптимальне упорядкування та значення довжини залишаються без змін.

Проілюструємо появу аномалій при покращенні початкових умов задачі.

Збільшимо кількість виконавців, $h=4$ та знайдемо оптимальне упорядкування та його довжину.

Таблиця 3.12 – Оптимальне упорядкування для графа G_4 при $h=4$

1	1	1	1	8					
2	2	5	5	5	5				
3	3	6	10	10	10	10	10	10	10
4	4	7		9					

Довжина $l=9$, значення цільової функції збільшується.

Зменшимо час виконання деяких завдань. Нехай $T'=(2,2,2,2,4,1,1,1,1,6)$.

Таблиця 3.13 – Оптимальне упорядкування для графа G_4 при T'

1	4	5	5	5	5				
2	2	6	8						
3	3	7	9	10	10	10	10	10	10

Довжина $l=10$, що є на 25% більше за довжину початкового упорядкування.

Послабимо технологічні обмеження, видалимо дуги (j_4, j_5) , (j_3, j_{10}) .

Таблиця 3.14 – Оптимальне упорядкування для графа G_4 при T'

1	1	1	1	6	8						
2	2	4	4	7	9						
3	3	5	5	5	5	10	10	10	10	10	10

Довжина $l=12$, тобто значення довжини збільшилось у цьому випадку на 50%.

Сформулюємо умови для уникнення аномалій у задачах із фіксованими роботами.

Умова узгодженості фіксації. Для будь-якого виконавця множина його фіксованих робіт повинна бути сумісною з технологічними обмеженнями графа $G=(V, U)$. Тобто, якщо $i_m \in V$, а $i_k \in V_r$, і між ними існує дуга $(i_m, i_k) \in U$, то робота i_m повинна виконуватися або тим самим виконавцем, або іншим виконавцем до початку роботи i_k .

Умова розв'язності задачі. Введення множини фіксованих робіт V_r не повинно робити задачу нерозв'язною. Тобто, навіть якщо частина робіт закріплена за конкретними виконавцями, існує допустиме упорядкування виконання всіх робіт з урахуванням технологічних обмежень і ресурсів.

Умова балансу завантаження. Фіксація робіт за виконавцем не повинна призводити до непропорційного завантаження виконавця. З цією метою для кожного виконавця повинна виконуватись умова:

$$\sum_{i \in V_r} T_i + \sum_{i \in \bar{V}} T_i \leq T_{max}, \quad (3.6)$$

де T_i – час виконання роботи i , \bar{V} – множина робіт, що не входять до V_r , але виконуються виконавцем r , а T_{max} – максимально допустимий час завантаження виконавця.

Твердження 3.3. Якщо введено множини фіксованих робіт V_r , то побудоване допустиме упорядкування буде оптимальним, якщо виконуються умови узгодженості фіксації, розв'язності задачі та балансу завантаження.

Доведення. Нехай граф $G = (V, U)$ описує залежності між роботами. Якщо для кожного виконавця множина V_r фіксованих робіт задовольняє умову узгодженості (тобто всі залежності між роботами зберігаються), умову розв'язності (тобто існує шлях у графі, який враховує фіксації) та умову балансу завантаження (виконавець не перевантажений), тоді за допомогою алгоритму побудови списку пріоритетів можливо знайти послідовність виконання робіт, яка задовольняє всі обмеження. Таким чином, оптимальне упорядкування існує.

Твердження 3.4. Зменшення часу виконання робіт або послаблення технологічних обмежень може збільшити довжину оптимального упорядкування.

Доведення. Розглянемо зміну графа G_6 при зменшенні часу виконання робіт (T') або видаленні дуг. Як показано у прикладі, у разі зменшення T' , довжина упорядкування l може збільшитись, оскільки змінюється баланс між завантаженням виконавців і кількістю виконуваних робіт. Аналогічно, послаблення технологічних обмежень (видалення дуг) збільшує кількість допустимих шляхів у графі, що може призводити до менш ефективного розподілу ресурсів.

Розглянемо граф G_4 (рис. 3.1) при заданому списку пріоритетів L , кількості виконавців $h = 4$ та з початковими вагами T . За цих умов отримали довжину упорядкування $l = 9$ (табл. 3.12).

Після зменшення часу деяких завдань T' довжина упорядкування збільшується до $l = 10$ (табл. 3.13), тобто на 25%. Далі при видаленні дуг (j_4, j_5) , (j_3, j_{10}) , при тих самих вагах T' , довжина зростає ще більше – до $l=12$ (табл. 3.14), що становить збільшення на 33% порівняно з початковим значенням. Це ілюструє, що навіть позитивні зміни у параметрах задачі, такі як зменшення часу виконання чи послаблення технологічних обмежень, не гарантують покращення, а іноді, навпаки, погіршують результат.

Розглянемо інші початкові умови задачі та невиконання необхідних умов для уникнення аномальних випадків. Нехай задано граф G_4 (рис. 3.1), список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10)$, ваги завдань $T=(4,2,2,2,4,1,1,1,1,6)$, кількість виконавців $h = 3$, фіксовані множини завдань: $V_1=\{1,8,9\}$, $V_2=\{2,5,6\}$, $V_3=\{3,4,7,10\}$.

На перший погляд фіксація здається повною: усі завдання фіксуються за конкретними виконавцями. Проте:

- 1) між роботами існують залежності $(4 \rightarrow 5)$ та $(3 \rightarrow 10)$;
- 2) тобто 3-ій виконавець повинен завершити роботу j_4 , перш ніж 2-ий виконавець розпочне j_5 , і завершити j_3 до початку j_{10} , яка також належить йому;
- 3) водночас тривалість роботи $\tau_{10}=6$ при загальному навантаженні на 3-го виконавця: $T(V_3)=11$, що перевищує середнє навантаження $T_{cp}=8$.

Таким чином, порушується умова балансу завантаження та узгодженості

фіксації, що робить задачу нерозв'язною в заданих обмеженнях. І не існує такого допустимого упорядкування, яке б задовольняло всі залежності та водночас не перевантажувало жодного виконавця.

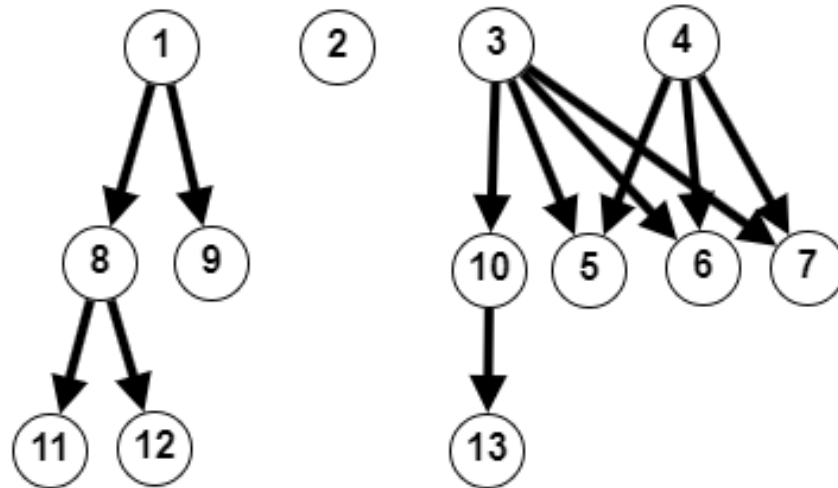
3.3 Умови стабілізації довжини упорядкування

Оцінюючи мінімальне значення довжини упорядкування, ми враховуємо, що воно не може бути меншим довжини критичного шляху $l_{кр}$ в графі. Якщо час виконання кожної роботи одиничний (інколи це значення приймають за вагу відповідних вершин), то під довжиною критичного шляху розуміють кількість вершин на найдовшому шляху, що веде з вершин, що не мають вхідних дуг до вершин, що не мають вихідних. Якщо існують вершини, для яких вага, відмінна від одиничної, то під довжиною критичного шляху будемо розуміти суму ваг тих вершин, які відповідають найдовшому шляху. Якщо існують ізольовані вершини, серед яких є вершина v_i , вага t_i якої більша за довжину критичного шляху $l_{кр}$, то $l \geq t_i$. Таким чином, $l \geq \max(t_i, l_{кр})$.

Означення 3.1. Будемо казати, що довжина оптимального упорядкування стабілізована, якщо $l = t_i$ або $l = l_{кр}$.

Розглянемо як впливає значення ширини на стабілізацію довжини.

Приклад 3.4. Нехай задано граф G_5 (рис.3.5), $h=3$,
 $L=(1,2,3,4,5,6,7,8,9,10,11,12,13)$, $T=(5,3,3,2,8,8,7,4,2,9,3,2,1)$.

Рис.3.5. Граф G_5

Представимо у вигляді таблиці 3.15 усі шляхи, що відповідають цьому графу.

Таблиця 3.15 – Шляхи графа G_5

1	1	1	1	1	8	8	8	8	11	11	11	
1	1	1	1	1	8	8	8	8	12	12		
1	1	1	1	1	9	9						
2	2	2										
3	3	3	10	10	10	10	10	10	10	10	10	13
3	3	3	5	5	5	5	5	5	5	5		
3	3	3	6	6	6	6	6	6	6	6		
3	3	3	7	7	7	7	7	7	7			
4	4	5	5	5	5	5	5	5	5			
4	4	6	6	6	6	6	6	6	6			
4	4	7	7	7	7	7	7	7				

З таблиці видно, що довжина критичного шляху дорівнює 13. Знайдемо оптимальне упорядкування та його довжину.

Таблиця 3.16 – Оптимальне упорядкування вершин графа G_5 при $h=3$

1	1	1	1	1	5	5	5	5	5	5	5	5	8	8	8	8	11	11	11
2	2	2	4	4	6	6	6	6	6	6	6	6	9	9			12	12	13
3	3	3	10	10	10	10	10	10	10	10	10	10	7	7	7	7	7	7	

Довжина $l=20$. Збільшимо кількість виконавців. Нехай $h=5$. Отримаємо упорядкування (табл. 3.17) з довжиною $l=13$.

Таблиця 3.17 – Оптимальне упорядкування вершин графа G при $h=5$

1	1	1	1	1	8	8	8	8	9	9	12	12
2	2	2	5	5	5	5	5	5	5	5		
3	3	3	6	6	6	6	6	6	6	6		
4	4		7	7	7	7	7	7	7	11	11	11
			10	10	10	10	10	10	10	10	10	13

Стабілізація означає, що усі роботи вже оптимально розподілені між доступними виконавцями, і немає потреби змінювати порядок виконання. Виникає питання за яких умов (при якому значенні h) довжина упорядкування стабілізується.

Позначимо через l_{ij} довжину тієї частини упорядкування, що включає місця, на яких стоять вершини i та j , між якими існує шлях (вершина i не має вхідних дуг, вершина j не має вихідних дуг).

Необхідно:

- 1) Знайти усі l_{ij} ;
- 2) Знайти t_k для кожної ізольованої вершини k .

Допустимі місця μ_i , на яких може стояти вершина i в упорядкуванні – це цілі значення проміжку $[\underline{\mu}_i, \bar{\mu}_i]$ (допустимі місця μ_i , значення яких натуральні числа, задовільняють нерівність $\underline{\mu}_i \leq \mu_i \leq \bar{\mu}_i$).

Твердження 3.5. Стабілізація відбувається за наступних умов:

$$\underline{\mu}_i \leq l_{kp} - l_{ij}, \quad (3.7)$$

$$\bar{\mu}_j \leq l_{кр}, \quad (3.8)$$

$$\underline{\mu}_k \leq l_{кр} - t_k, \quad (3.9)$$

де $\underline{\mu}_i, \underline{\mu}_k$ – номери перших місць вершин i та k в упорядкуванні S , $\bar{\mu}_j$ – номер останнього місця вершини j в упорядкуванні.

Доведення. Довжина оптимального упорядкування стабілізується, коли вона дорівнює довжині критичного шляху $l(S) = l_{кр}$ або найбільшому часу виконання ізольованої роботи $l(S) = t_k$. Таким чином, умова (3.2) запобігає невиконанню рівності $l(S) = l_{кр}$, а (3.3) рівності $l(S) = t_k$.

Звідси логічно випливає, що для обох випадків $l = l_{кр}$ або $l = t_k$ виконуються наступні відповідні рівності:

$$\underline{\mu}_i = 1, \underline{\mu}_k = 1. \quad (3.10)$$

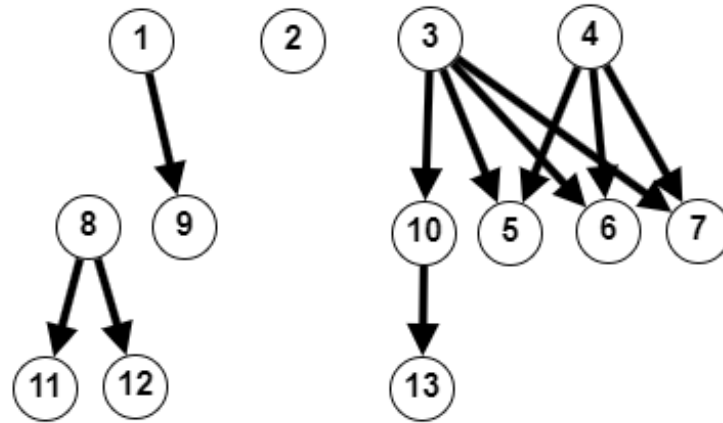
Нехай $Q = \sum_{k=1}^n t_k$ – сумарний час виконання усіх завдань. Якщо список пріоритетів L буде побудовано за алгоритмом *, то має місце наступне твердження.

Твердження 3.6. Довжина оптимального упорядкування може бути стабілізована, коли задана ширина упорядкування h дорівнює:

$$h_{ст} = \lceil \frac{Q}{l_{кр}} \rceil \quad (3.11)$$

Доведення. Побудова списку пріоритетів за описаним вище алгоритмом виключає основну причину виникнення аномального збільшення довжини упорядкування, – відтермінування виконання завдань із більшим значенням t_i . Таким чином, будується максимально щільне впорядкування, тобто кількість вільних місць не перевищує значення $h(S)$. Відповідно, справедлива рівність (3.11).

Постає питання чи будуть мати місце аномалії після стабілізації довжини упорядкування. Розглянемо на прикладі 3.1 покращення інших умов. Послабимо обмеження, що стосується технологічного порядку. Отримаємо граф G' (рис.3.6).

Рис.3.6. Граф G_5'

Замість очікуваної довжини $l=13$, як видно із табл. 3.3, довжина $l=15$. Тобто, послаблення технологічних обмежень погіршує значення цільової функції.

Таблиця 3.18 – Оптимальне упорядкування вершин графа G' при $h=5$

1	1	1	1	1	10	10	10	10	10	10	10	10	10	10	13
2	2	2	5	5	5	5	5	5	5	5					
3	3	3	6	6	6	6	6	6	6	6					
4	4		7	7	7	7	7	7	7	12	12				
8	8	8	8	9	9	11	11	11							

Погіршення можемо отримати і якщо зменшимо час виконання робіт. При $\tau_l'=3$ отримаємо оптимальне упорядкування (табл. 3.19), довжина якого $l=15$.

Таблиця 3.19 – Оптимальне упорядкування вершин графа G_5 при $h=5$, $\tau_l'=3$

1	1	1	5	5	5	5	5	5	5	5					
2	2	2	6	6	6	6	6	6	6	6					
3	3	3	7	7	7	7	7	7	7	12	12				
4	4		8	8	8	8	11	11	11						
			9	9	10	10	10	10	10	10	10	10	10	10	13

Інтуїтивне покращення початкових умов знову призвело до погіршення

результату на 16,7%.

Перевіримо виникнення аномалій після застосування алгоритму*.

Знаходимо довжини усіх шляхів графа та розташовуємо їх у порядку незростання:

$l_{3,13}=13, l_{1,11}=12, l_{1,12}=11, l_{3,5}=11, l_{3,6}=11, l_{3,7}=10, l_{4,5}=10, l_{4,6}=10, l_{4,7}=9, l_{1,9}=7, l_2=3$.

Будуємо список пріоритетів: $L'=(3,10,13,1,8,11,12,5,6,7,4,9,2)$.

Знайдемо оптимальне упорядкування, послабивши технологічні обмеження (рис. 3.6, табл. 3.20).

Таблиця 3.20 – Оптимальне упорядкування вершин графа G' при L'

3	3	3	10	10	10	10	10	10	10	10	10	13	
1	1	1	1	1	12	12	7	7	7	7	7	7	7
8	8	8	8	11	11	11	9	9					
4	4		5	5	5	5	5	5	5	5			
2	2	2	6	6	6	6	6	6	6	6			

Довжина $l=14$ є меншою за довжину упорядкування з таблиці 3.18, але більша за довжину оптимального упорядкування (табл. 3.17). Спробуємо застосувати алгоритм після послаблення технологічних обмежень.

Розглянемо граф G' (рис. 3.2).

$l_{3,13}=13, l_{3,5}=11, l_{3,6}=11, l_{3,7}=10, l_{4,5}=10, l_{4,6}=10, l_{4,7}=9, l_{1,9}=7, l_{8,11}=7, l_{8,12}=6, l_2=3$.

Будуємо список пріоритетів: $L''=(3,10,13,5,6,7,4,1,9,8,11,12,2)$.

Знайдемо оптимальне упорядкування (табл. 3.21).

Таблиця 3.21 – Оптимальне упорядкування вершин графа G'_5 при L''

3	3	3	10	10	10	10	10	10	10	10	10	10	13
4	4		5	5	5	5	5	5	5	5			
1	1	1	1	1	9	9	11	11	11	12	12		
8	8	8	8	7	7	7	7	7	7	7			
2	2	2	6	6	6	6	6	6	6	6			

Отримали довжину оптимального упорядкування $l=13$.

Перевіримо припущення на іншому аномальному випадку.

Нехай $\tau_l' = 3$.

Тоді $l_{3,13}=13$, $l_{3,5}=11$, $l_{3,6}=11$, $l_{1,11}=10$, $l_{3,7}=10$, $l_{4,5}=10$, $l_{4,6}=10$, $l_{1,12}=9$, $l_{4,7}=9$, $l_{1,9}=5$, $l_2=3$.

Будуємо список пріоритетів: $L''' = (3, 10, 13, 5, 6, 1, 8, 11, 7, 4, 12, 9, 2)$ та упорядкування.

Таблиця 3.22 – Оптимальне упорядкування вершин графа G_5 , $\tau_l' = 3$, L'''

3	3	3	10	10	10	10	10	10	10	10	10	13
1	1	1	5	5	5	5	5	5	5	5		
4	4		6	6	6	6	6	6	6	6		
2	2	2	8	8	8	8	11	11	11	12	12	
			7	7	7	7	7	7	7	9	9	

Довжина $l=13$, величина є мінімально можливою.

Твердження 3.7. У випадках стабілізації довжини оптимального упорядкування, аномалії не виникатимуть.

Згідно з твердженням 3.2, стабілізація довжини оптимального упорядкування відбувається, коли ширина h досягає максимального значення:

$$h_{\text{ст}} = \lceil Q / l_{\text{кр}} \rceil. \quad (3.12)$$

Значення $h_{\text{ст}}$ обирається так, щоб забезпечити рівномірний розподіл Q , при цьому довжина l є близькою до $l_{\text{кр}}$ або дорівнює їй.

Рівномірний розподіл при $h = h_{\text{ст}}$.

Ширина h досягає значення $h_{\text{ст}}$, це означає, що елементи рівномірно розподіляються таким чином, що довжина l є близькою до $l_{\text{кр}}$, або дорівнює $l_{\text{кр}}$. Завдяки округленню вверх ($\lceil \cdot \rceil$), навіть якщо $Q / l_{\text{кр}}$ є дробовим числом, всі елементи розподіляються без залишків, тобто на кожному місці в упорядкуванні є вершини.

Відсутність аномалій: усі вершини графа повністю розподілені; упорядкування не потребує додаткових змін у розподілі, оскільки довжина l стабілізувалася; нестабільність, пов'язана зі зміною довжини l , виключена,

оскільки упорядкування є оптимальним.

Оптимальність розподілу: при досягненні $h = h_{\text{ст}}$, довжина l стабілізується, оскільки ширина h забезпечує необхідний баланс. За цих умов більше не виникає потреби в корекції l , оскільки всі елементи враховані, а упорядкування є оптимальним.

У випадку стабілізації довжини l , розподіл елементів є оптимальним, і упорядкування досягає стійкого стану. Завдяки цьому: аномалії, пов'язані з нестабільністю чи нерівномірним розподілом елементів, не виникають; розподіл відповідає умовам задачі. Таким чином, у випадках стабілізації довжини оптимального упорядкування аномалії не виникатимуть.

Окремі методи побудови стабілізованих рішень розглядаються в рамках комбінаторної оптимізації, зокрема у фундаментальній праці [104].

3.4 Вплив переривань на виникнення аномалій

Частина досліджень цієї роботи зосереджується на аномальних випадках, які можуть виникати через зміщення виконання завдань з високим пріоритетом на користь менш важливих завдань. Тому, особливий інтерес становить вивчення впливу переривань на виникнення аномалій та умови, за яких переривання дозволяють уникнути аномальних випадків.

Переривання – це механізм, що дозволяє тимчасово призупинити виконання одного завдання для виконання іншого, а потім завершити призупинене [46]. Цей підхід відіграє вирішальну роль у задачах оптимізації розподілу ресурсів у багатозадачних системах. Загальна ідея використання переривань полягає в тому, щоб запобігти "блокуванню" ресурсів менш пріоритетними завданнями, коли більш важливі завдання стають доступними. У системах реального часу або у хмарних обчисленнях, де обмежені ресурси мають бути розподілені між кількома процесами, переривання дозволяють підвищити ефективність використання цих ресурсів [105-107].

Переривання завдань, з математичної точки зору, змінюють класичні алгоритми розподілу, адже додають до них елементи динамічної оптимізації. Це зумовлено тим, що переривання вимагають врахування зміни доступності ресурсів у реальному часі. Задачі з перериваннями стають складнішими з точки зору обчислювальної складності, але водночас вони забезпечують кращі результати у випадках динамічних систем.

Переривання можна класифікувати за декількома ознаками:

1) Пріоритетність переривань.

Завдання з високим пріоритетом можуть переривати виконання завдань з нижчим пріоритетом. Це запобігає тому, щоб критично важливі завдання були відкладені на пізніший час через тривалі процеси з вищим пріоритетом.

2) Тип переривань:

- Жорсткі переривання. Переривання завдань є обов'язковим і не може бути відкладено. Наприклад, у системах реального часу, таких як управління польотами літаків, завдання з найвищим пріоритетом мають виконуватися негайно, незалежно від поточного стану інших завдань.

- Гнучкі переривання. Дозволяється короткочасне відкладання переривання, якщо поточне завдання є критичним для завершення. Це може бути корисним, наприклад, у виробничих системах, де неповне завершення деяких завдань може призвести до втрати часу і ресурсів.

3) Частота переривань.

Деякі системи дозволяють часті переривання, тоді як інші обмежують кількість переривань, аби уникнути надмірного збільшення часу на завершення кожного окремого завдання. Наприклад, у хмарних системах обчислення занадто часті переривання можуть спричинити значне зростання витрат на контекстне перемикання [108].

Наведемо умови при яких дозвіл на переривання не призводить до аномальних випадків.

Розглянемо оргграф $G = (V, U)$, список пріоритетів $L = (i_1, \dots, i_n)$ та ваги вершин $T = (\tau_1, \dots, \tau_n)$. Додатково припустимо, що наявні обмеження щодо ширини

упорядкування та дозволяється переривання під час виконання робіт. Потрібно з'ясувати переривання виконання яких саме робіт призведе до зменшення загального часу завершення процесу.

Твердження 3.8. Переривання завдань, у випадках, коли спостерігаються аномалії, може дозволити уникнути їх за таких умов:

1) Пріоритетність. Список пріоритетів виконання завдань необхідно змінити за відповідним алгоритмом пріоритетного динамічного перерозподілу, що наводиться нижче. Завдання з вищим пріоритетом можуть переривати виконання завдань із нижчим пріоритетом, що забезпечує виконання критичних робіт у першу чергу.

2) Переривання завдань. Будь-яке завдання, яке було перервано, може бути продовжене пізніше відповідно до пріоритетного списку. Це дозволяє гарантувати, що більш пріоритетні завдання будуть виконані на максимально ранніх можливих етапах, мінімізуючи загальну тривалість виконання.

3) Динамічна актуалізація. На кожному етапі виконання дозволяється переривати роботи та перерозподіляти ресурси. Це дає змогу враховувати зміни в доступності ресурсів і стані виконання завдань у режимі реального часу [9].

Доведення.

1. Пріоритетність

Список пріоритетів $L = (i_1, i_2, \dots, i_n)$ визначає порядок виконання завдань. Для уникнення аномалій важливо, щоб завдання з вищим пріоритетом виконувалися раніше за рахунок переривання при виконанні менш важливих завдань.

Нехай завдання мають тривалості $T = (\tau_1, \tau_2, \dots, \tau_n)$, пріоритети $P = (p_1, p_2, \dots, p_n)$, де $p_i > p_j$ означає, що i завдання має вищий пріоритет за j . Якщо виконання завдання i із високим пріоритетом p_i відкладається через необхідність виконання завдання j з нижчим пріоритетом, то це може призвести до непередбаченого збільшення довжини упорядкування через затримку виконання критичного завдання. Згідно з умовою, завдання з вищим пріоритетом мають переривати j завдання, тобто: якщо $p_i > p_j$ і $\tau_j > \tau_i$, а завдання i на черговому кроці ще недоступне, то як тільки воно стане доступним, завдання j

переривається для виконання завдання i .

Алгоритм динамічного перерозподілу змінює список L таким чином, що завдання i виконується перед j , і це дозволяє уникнути затримок у виконанні завдань із високим пріоритетом. Таким чином, першочергове виконання критичних завдань запобігає виникненню аномалій.

2. Переривання завдань

Нехай на деякому кроці завдання j було перерване для виконання завдання i . Після завершення i , робота j повертається в чергу виконання відповідно до оновленого списку пріоритетів L .

Умови оптимального виконання:

Якщо час виконання роботи j ще не завершений і робота переривається, то на її виконання залишається час:

$$\tau'_j = \tau_j - \bar{\tau}_j, \quad (3.13)$$

де $\bar{\tau}_j$ – це сумарний час, який вона вже виконувалася.

Якщо всі критичні завдання завершені ($p_k < p_j$ для всіх k), j завершується без затримок. Це забезпечує мінімізацію загального часу виконання, де T_{total} залишається оптимальним, оскільки завдання із високими значеннями p_i не блокуються.

3. Динамічна актуалізація

На кожному етапі виконання завдань система повинна дозволяти динамічний перерозподіл ресурсів. Це забезпечує адаптацію до змін у доступності ресурсів та стані завдань у реальному часі.

Нехай $G = (V, U)$ – орграф залежностей між завданнями, де V – множина завдань, U – залежності між ними. Динамічний алгоритм на кожному кроці оновлює доступність виконавців M_t . Переривання дозволяють змінювати порядок виконання завдань:

$$\forall t : L \rightarrow L', \quad (3.14)$$

де L' враховує зміни у M_t .

Умови для уникнення аномалій:

1) якщо виконавець $k \in M$ звільняється, він одразу перепризначається для

завдань із найвищим пріоритетом;

2) якщо завдання i завершене, воно видаляється з G , і оновлюється множина доступних вершин $V' \subset V$;

3) динамічний розподіл ресурсів мінімізує затримки та гарантує виконання всіх перерваних завдань.

Таким чином, динамічна актуалізація дозволяє враховувати зміни у реальному часі та уникати аномалій, забезпечуючи оптимальний розподіл завдань.

Алгоритм пріоритетного динамічного перерозподілу.

1) Ініціалізація: всі виконавці позначаються як вільні.

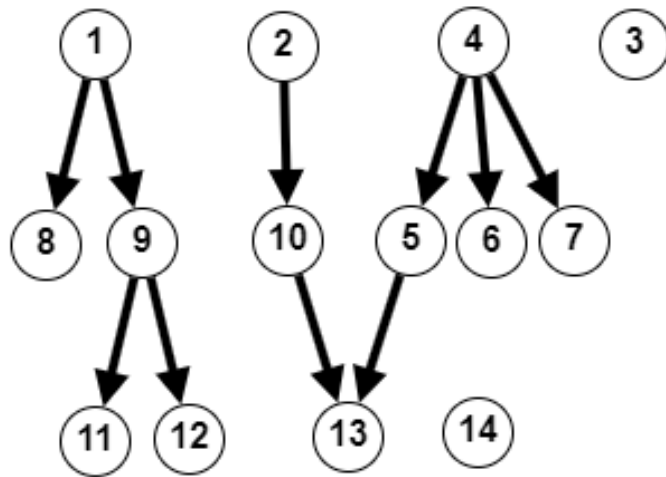
2) Призначення роботи: коли нова робота i стає доступною для виконання, вона призначається вільному виконавцю.

3) Перерозподіл: на кожному кроці виконання роботи i перевіряється доступність виконання більш пріоритетних робіт. Якщо знайдена більш пріоритетна робота j , яка стає доступною до виконання на даному кроці, виконання роботи i переривається, а робота j призначається даному виконавцю.

У цьому алгоритмі особлива увага приділяється розподілу пріоритетів та ефективному управлінню перериваннями, що дозволяє отримати упорядкування оптимальної довжини.

Проілюструємо роботу алгоритму на прикладі.

Приклад 3.5. Нехай задано граф G_6 (рис.3.7), ширина упорядкування $h = 3$ список пріоритетів $L = (1,2,3,4,5,6,7,8,9,10,11,12,13,14)$ та час виконання завдань $T = (4,2,2,2,4,2,2,1,1,6,3,1,4,2)$.

Рис. 3.7 – Граф G_6

Знайдемо оптимальне упорядкування та його довжину.

Таблиця 3.23 – Оптимальне упорядкування вершин графа G_6 при $h = 3$

1	1	1	1	5	5	5	5	8	11	11	11
2	2	4	4	6	6	7	7	9	12	14	14
3	3	10	10	10	10	10	10	13	13	13	13

Довжина $l = 12$.

Побудуємо тепер оптимальне упорядкування з дозволеними перериваннями (табл. 3.24).

Таблиця 3.24 – Оптимальне упорядкування вершин графа G_6 з дозволеним перериванням завдань

1	1	1	1	5	5	5	5	12	14	14	13	13	13	13
2	2	4	4	6	6	8	10	10	10	10				
3	3	10	10	7	7	9	11	11	11					

Довжина упорядкування $l = 15$, значення збільшилось на 25%.

Змінюємо список пріоритетів згідно до алгоритму побудови оптимального списку пріоритетів.

Знаходимо загальний час виконання завдань, який дорівнює довжині

відповідних шляхів у графі: $l_{1-8} = 5, l_{1-11} = 8, l_{1-12} = 6, l_{2-13} = 12, l_{4-13} = 10, l_{4-6} = 4, l_{4-7} = 4, l_3 = 2, l_{14} = 2$.

Оновлюємо список пріоритетів: $L' = (2, 10, 13, 4, 5, 1, 9, 11, 12, 8, 6, 7, 3, 14)$.

Будуємо оптимальне упорядкування з новим значенням L' (табл. 3.25).

Таблиця 3.25 – Оптимальне упорядкування вершин графа G_6 з дозволим перериванням завдань та L'

2	2	1	1	9	11	11	11	13	13	13	13
4	4	10	10	10	10	10	10	6	6	3	3
1	1	5	5	5	5	12	8	7	7	14	14

$l = 12$, довжина упорядкування дорівнює довжині початкового упорядкування, тобто значення цільової функції покращилось.

Приклад 3.6. Нехай задано граф G_7 (рис.3.8), ширина упорядкування $h = 3$ список пріоритетів $L = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21)$ та час виконання завдань $T = (5, 2, 5, 3, 2, 2, 1, 1, 1, 7, 6, 3, 1, 4, 4, 2, 4, 1, 1, 1, 1)$.

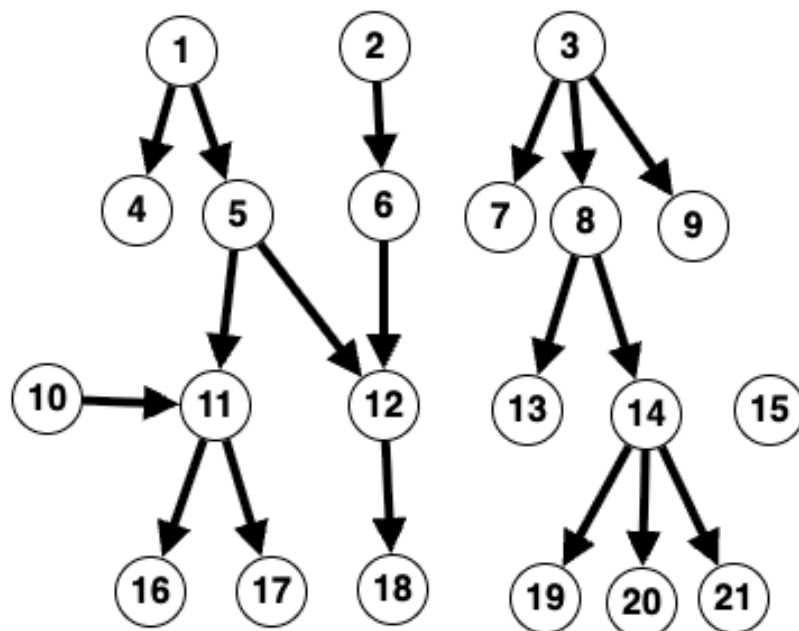


Рис. 3.8 – Граф G_7

Знайдемо оптимальне упорядкування та його довжину.

Таблиця 3.26 – Оптимальне упорядкування вершин графа G_7 при $h = 3$

1	1	1	1	1	4	4	4	7	8	12	12	12	15	15	15	15	18	20
2	2	10	10	10	10	10	10	10	9	13	14	14	14	14	16	16	19	21
3	3	3	3	3	5	5	6	6	11	11	11	11	11	11	17	17	17	17

Довжина $l = 19$.

Побудуємо тепер оптимальне упорядкування з дозволеними перериваннями (табл. 3.).

Таблиця 3.27 – Оптимальне упорядкування вершин графа G_7 з дозволеним перериванням завдань

1	1	1	1	1	4	4	4	9	13	14	14	14	14	14	18	19	21					
2	2	10	10	10	5	5	7	10	10	10	10	11	11	11	11	11	11	17	17	17	17	
3	3	3	3	3	6	6	8	12	12	12	15	15	15	15	20							

Довжина упорядкування $l = 22$, значення збільшилось на 15%.

Змінюємо список пріоритетів згідно до алгоритму побудови оптимального списку пріоритетів.

Знаходимо загальний час виконання завдань, який дорівнює довжині відповідних шляхів у графі: $l_{1-4} = 8, l_{1-16} = 15, l_{1-17} = 17, l_{1-18} = 8, l_{10-16} = 15, l_{10-17} = 17, l_{2-18} = 8, l_{3-6} = 7, l_{3-7} = 6, l_{3-13} = 7, l_{3-19} = 11, l_{3-20} = 11, l_{3-21} = 11, l_{3-9} = 6, l_{15} = 4$.

Оновлюємо список пріоритетів:

$L' = (1, 5, 11, 17, 10, 16, 3, 8, 14, 19, 20, 21, 4, 12, 18, 2, 6, 13, 7, 9, 15)$.

Будуємо оптимальне упорядкування з новим значенням L' (табл. 3.28).

Таблиця 3.28 – Оптимальне упорядкування вершин графа G_7 з дозволеним перериванням завдань та L'

1	1	1	1	1	5	5	11	11	11	11	11	11	11	17	17	17	17	18	18
10	10	10	10	10	10	10	10	14	14	14	19	21	13	6	6	12	12	12	21
3	3	3	3	3	8	14	4	4	4	20	2	2	7	9	15	15	15	15	

Довжина $l = 12$, тобто дорівнює довжині оптимального упорядкування та довжині критичного шляху.

Таким чином, дозвіл переривань може сприяти уникненню аномалій у паралельних системах і підвищенню ефективності використання ресурсів. Переривання дозволяють забезпечити пріоритетне виконання важливих завдань і адаптуватися до динамічних змін у системі.

3.5 Висновки до розділу

У цьому розділі наведено аналіз видів аномалій, які виникають у спеціальних класах задач дискретної оптимізації.

Для задач оптимального розподілу вершин орграфів було виділено основні типи аномалій, зокрема аномалії, що можуть виникати навіть за умови покращення декількох початкових параметрів (наприклад, послаблення технологічних обмежень та зменшення часу виконання деяких завдань, ...).

Виявлено, що більшість аномалій обумовлені комплексною взаємодією між структурою задачі, технологічними обмеженнями, пріоритетами розподілу завдань та впливом ефективності виконавців. Це в свою чергу ускладнює передбачення результатів оптимізації навіть у простих ситуаціях.

Встановлено, що наявність різних продуктивностей виконання завдань відповідними виконавцями суттєво ускладнює побудову збалансованого розподілу завдань, а для задач з фіксованими технологічними обмеженнями найважливішим фактором залишається узгодження між обмеженнями та розподілом завдань між виконавцями.

Доведено, що нерівномірний розподіл завдань між виконавцями є основною причиною збільшення загального часу виконання, незалежно від продуктивності виконавців.

Запропоновано три ключові умови для уникнення виникнення аномалій:

- 1) сортування завдань за пріоритетами з урахуванням технологічних

обмежень;

- 2) забезпечення рівномірного розподілу навантаження між виконавцями;
- 3) використання динамічного планування для адаптації розподілу завдань у реальному часі.

Аналіз довів, що стабілізація довжини оптимального упорядкування є критерієм завершення процесу оптимізації. Умови стабілізації включають збалансованість навантаження та послідовне виконання завдань критичних шляхів у графі, що задає технологічні обмеження на порядок.

Для стабілізації також важливо:

- 1) аналізувати критичні шляхи в графі, оскільки вони визначають мінімально можливу довжину розв'язку;
- 2) рівномірно розподіляти завдання з урахуванням продуктивності виконавців, технологічних обмежень і пріоритетності робіт.

Узагальнюючи проведені дослідження можна стверджувати, що ключем до ефективного вирішення задач дискретної оптимізації, в яких можуть виникати аномалії, є своєчасне їх виявлення, коректне балансування обмежень і динамічний аналіз для збереження продуктивності та стабільності розв'язків.

Отримані результати можуть бути використані для розробки ефективних алгоритмів розподілу завдань у задачах планування, що дозволить зменшити ризик виникнення аномалій і забезпечити раціональне використання ресурсів. Це особливо актуально для задач розкрою та пакування, де ключову роль відіграє оптимізація порядку виконання завдань і використання матеріалів.

Розділ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Опис структури та функціональних модулів програми

Для виявлення аномалій, оцінки ефективності реалізованих алгоритмів та їх порівняння із існуючими методами було розроблено програмне забезпечення на основі об'єктно-орієнтованого підходу з використанням мови програмування C# [109-110].

Клас *Graph* є основним компонентом програми, який використовується для представлення орграфів. Цей клас містить необхідні атрибути та методи для роботи із графами, включаючи створення, редагування та аналіз структури графа. Для збереження зв'язків між вершинами використовується представлення у вигляді двох списків: список вершин та список дуг.

```
public class Graph {
    public List<Node> Nodes { get; set; }
    public List<Edge> Edges { get; set; }

    public Graph() {
        Nodes = new List<Node>();
        Edges = new List<Edge>();
    }
}
```

Клас *Graph* підтримує функції для створення, збереження та аналізу графів. Введення даних можливе також і з файлу.

```
public void LoadGraphFromFile(string filePath) {
    string[] lines = File.ReadAllLines(filePath);
    foreach (var line in lines.Skip(1)) {
        var parts = line.Split(' ');
        int from = int.Parse(parts[0]);
        int to = int.Parse(parts[1]);
        Edges.Add(new Edge(from, to));
    }
}
```

У даному програмному продукті реалізовано:

- 1) алгоритми побудови відповідних оптимальних упорядкувань;
- 2) покращення початкових умов шляхом скорочення часу виконання завдань, послаблення обмежень на порядок робіт, зміни кількості виконавців та модифікації списку пріоритетів;
- 3) виявлення та усунення аномалій, які можуть виникати при покращенні початкових умов;
- 4) реалізація виконання комбінацій покращень початкових умов;
- 5) аналіз впливу параметрів графа на ефективність планування.

Розглянемо структуру програмної реалізації, яка включає основні етапи вирішення задачі.

1. Ініціалізація. Вхідні дані:

- орієнтований граф задач G , V – множина вершин (завдань), U – множина дуг (залежностей між завданнями);
- список пріоритетів L , який задає порядок виконання завдань;
- кількість виконавців h ;
- множина ваг вершин T ;
- інші параметри задачі: продуктивність виконавців, фіксація робіт за виконавцями.

2. Виконання завдань:

- визначення довжини l ;
- аналіз критичного шляху;
- алгоритм розподілу завдань між виконавцями;
- алгоритм побудови оптимального списку пріоритетів.

3. Зміна початкових даних:

- реалізація покращень, таких як зміна списку пріоритетів, збільшення кількості виконавців, вилучення завдань;
- аналіз впливу покращень на якість розв'язку та виявлення аномалій.

4. Виведення результатів:

- загальний час виконання завдань;

- оптимальне упорядкування;
- розподіл завдань між виконавцями;
- графічна візуалізація.

Розглянемо більш детально основні компоненти програми.

1. Для реалізації функціональної частини використані такі бібліотеки C#:

- *GraphX* – для роботи з графами та візуалізації структури задач;
- *Accord.NET* – для алгоритмів оптимізації та обчислення критичних шляхів;
- *System.Collections.Generic* – для збереження графів у вигляді списків суміжності.

Структури даних:

- *Graph* – клас для представлення орієнтованого графа задач;
- *PriorityQueue* – черга завдань із списку пріоритетів;
- *WorkerPool* – список виконавців, що містить інформацію про їхній поточний стан завантаження.

Опишемо модулі програми.

Модуль 1: Вхідні дані та ініціалізація.

Функції:

- *load_graph (file)* — зчитує граф G із вхідного файлу (файл має містити вершини, дуги та ваги завдань);
- *generate_graph ()* — генерує граф довільним чином (згідно до параметрів задачі);
- *visualize_graph (G)* — візуалізує граф;
- *initialize_resources(h, s, T)* — ініціалізує ресурси, задані продуктивності виконавців та встановлює ваги завдань.

Модуль 2: Моделювання виконання завдань.

Функції:

- *simulate_execution (G, L, h)* — реалізує алгоритм побудови паралельного упорядкування за графом G , списком пріоритетів L та кількістю виконавців h . Виводить загальний час виконання робіт та дозволяє оцінити оптимальність розподілу;

- *calculate_critical_path* (G) — визначає критичний шлях у графі.

- *update_execution_times* (G, h, s_new) — оновлює час виконання завдань при зміні швидкості роботи виконавців.

Модуль 3: Покращення початкових умов.

Функції:

- *reduce_execution_time* (G, L, h) — реалізує алгоритми для скорочення часу виконання робіт (зменшення ваг завдань);

- *relax_order_constraints* (G) — послаблює обмеження на порядок виконання завдань шляхом видалення дуг між вершинами графа;

- *increase_worker_count* (G, h, h_new) — додає нових виконавців і перерозподіляє завдання для скорочення загального часу;

- *modify_priority_order* (G, L) — змінює порядок виконання завдань відповідно до ваг завдань, критичного шляху тощо.

Модуль 4: Перевірка наявності аномалій.

Функція:

- *check_anomalies* (G, h) — перевіряє, чи виникають аномалії при зміні умов (наприклад, збільшення загального часу при зменшенні кількості завдань).

Модуль 5: Випадки одночасного покращення умов.

Функції:

- *combine_improvements* ($G, h, L, options$) — реалізує одночасне покращення декількох умов (наприклад, зміна списку пріоритетів і збільшення кількості виконавців);

- *analyze_combined_effects* (G, h) — оцінює вплив одночасних змін на загальний час виконання задач та можливі аномалії.

Модуль 6: Додаткові операції.

Функції:

- *shorten_priority_list* (G, L) — скорочує список пріоритетів;

- *remove_tasks* (G, L, h) — вилучає певні вершини із графа;

- *remove_edges* (G, L, h) — вилучає певні дуги із графа;

- *adjust_worker_speed* (h, s_new) — змінює для певних виконавців їх

продуктивність.

Алгоритми:

- алгоритм перевірки критичного шляху: використовується пошук у глибину (DFS) або алгоритм топологічного сортування для виявлення критичного шляху в графі G ;
- алгоритм зміни списку пріоритетів, при якому аномалії не виникатимуть;
- алгоритм одночасного покращення декількох початкових умов;
- алгоритм перерозподілу завдань.

Розроблене програмне забезпечення дозволяє моделювати процес виконання завдань у графі, аналізувати оптимальність розподілу завдань та коригувати умови для підвищення ефективності результату.

4.2 Інтерфейс програми

Інтерфейс розробленого програмного забезпечення є інтуїтивно зрозумілим і побудований відповідно до принципів функціональної доступності. Основне вікно програми містить панель керування параметрами задачі, поле візуалізації результатів та інформаційний блок для виводу допоміжних повідомлень та підсумкових даних (рис.4.1).

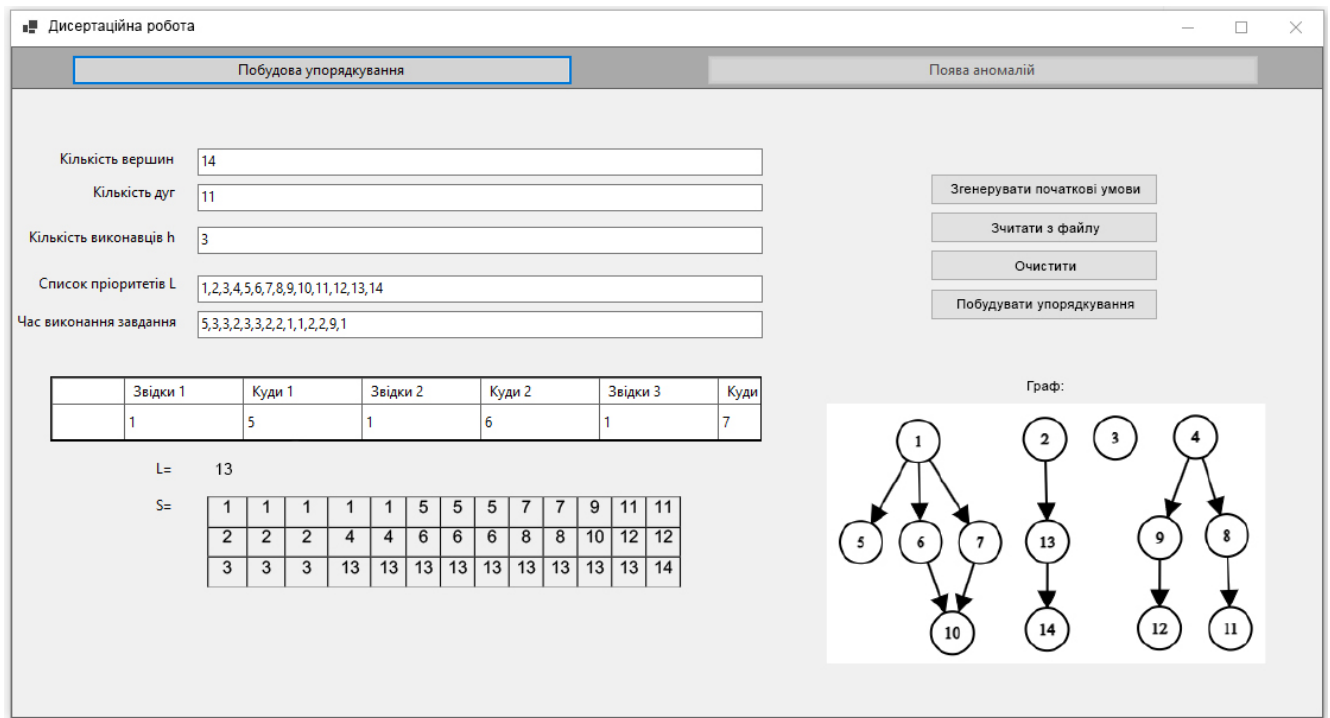


Рис. 4.1. – Інтерфейс програмного забезпечення

У верхній частині інтерфейсу реалізовано меню вибору режимів роботи, що дозволяє користувачеві перемикатися між основними функціональними модулями програми «Побудова упорядкування» та «Поява аномалій» (рис. 4.2).

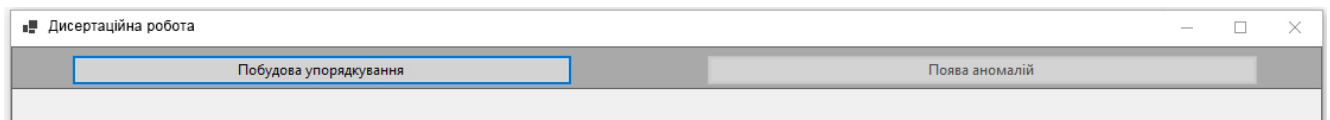


Рис. 4.2. – Режими роботи

Інтерактивна частина інтерфейсу дає змогу вводити вхідні параметри вручну, завантажувати їх із файлу або генерувати довільним чином. Також за допомогою кнопки «Побудувати упорядкування» будувати оптимальне упорядкування (рис. 4.3).

Кількість вершин	14
Кількість дуг	11
Кількість виконавців h	3
Список пріоритетів L	1,2,3,4,5,6,7,8,9,10,11,12,13,14
Час виконання завдання	5,3,3,2,3,3,2,2,1,1,2,2,9,1

	Звідки 1	Куди 1	Звідки 2	Куди 2	Звідки 3	Куди
	1	5	1	6	1	7

Рис. 4.3. – Поля для вводу початкових даних

Візуальний блок забезпечує наочне відображення структури вхідних даних, а також графічне представлення результатів розв'язання (рис. 4.4).

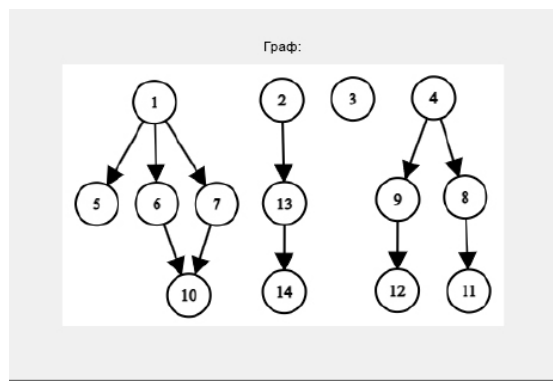


Рис. 4.4. – Візуальне представлення графа G

Візуалізація розділу «Поява аномалій» зображена на рис. 4.5.

Дисертаційна робота
Побудова упорядкування
Поява аномалій

Введіть кіл-ть змінних початкових даних:

Введіть нові значення:
 h
 T

I' = 19 ΔI = +26.7%

S' =

1	1	1	5	5	5	5	12												
2	2	2	6	6	6	6	6												
3	3	3	7	7	9	9	9												
4	4	8	8	8	10	11	13	13	13	13	13	13	13	13	13	13	13	13	13

L' = 13, 1, 6, 5, 7, 9, 10, 2, 4, 8, 11, 12, 3

I'' = 15 ΔI = 0%

S'' =

13	13	13	13	13	13	13	13	13	13	13	13	3	3	3					
1	1	1	1	1	6	6	6	6	6	8	8	8	8	10	11				
2	2	2	4	4	5	5	5	5	7	7	9	9	9	9	12				

Рис. 4.5. – Пошук аномальних випадків

4.3 Інструкція користувача

На прикладі 1 з початковими умовами: задано граф G_4 (рис.1.4), список пріоритетів $L=(1,2,3,4,5,6,7,8,9,10,11,12,13)$, $h=3$ та множина ваг вершин $T=(5,3,3,2,4,5,2,3,3,1,1,1,12)$, – опишемо інструкцію користувача.

Після запуску програми відкривається вікно, зображене на рис. 4.6.

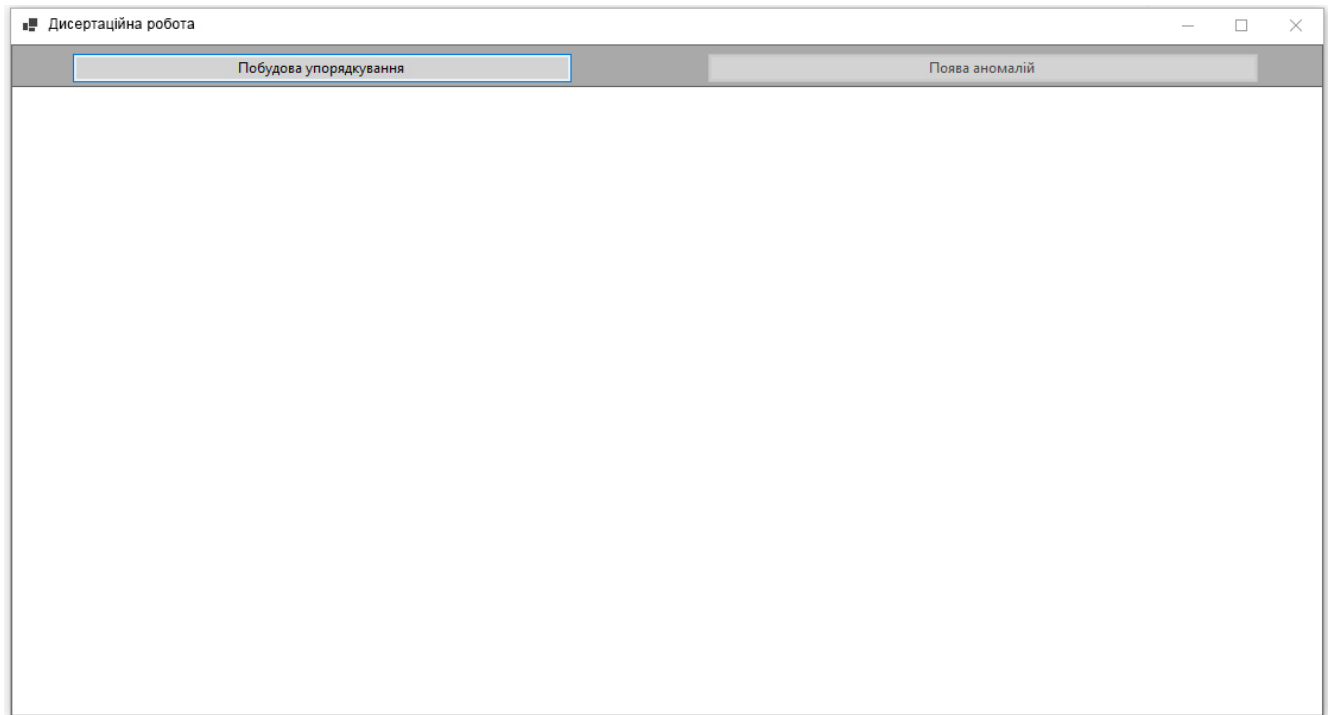


Рис. 4.6. – Стартовий екран програмного забезпечення

Після натискання кнопки «Побудова упорядкування», з'являються основні поля для вводу початкових умов задачі: «Кількість вершин», «Кількість дуг», «Кількість виконавців h », «Список пріоритетів L », «Час виконання завдань» та таблиця для вводу заданих дуг. За допомогою кнопок «Згенерувати початкові умови», «Зчитати з файлу» та «Очистити» можна виконати відповідні дії. За допомогою кнопки «Побудувати упорядкування» - побудувати упорядкування та візуально зобразити граф (рис. 4.7, 4.8).

Дисертаційна робота

Побудова упорядкування

Поява аномалій

Кількість вершин

Кількість дуг

Кількість виконавців h

Список пріоритетів L

Час виконання завдання

Згенерувати початкові умови

Зчитати з файлу

Очистити

Побудувати упорядкування

	Звідки	Куди

L=

S=

Рис. 4.7. – Розділ «Побудова упорядкування»

Дисертаційна робота

Побудова упорядкування

Поява аномалій

Кількість вершин

Кількість дуг

Кількість виконавців h

Список пріоритетів L

Час виконання завдання

Згенерувати початкові умови

Зчитати з файлу

Очистити

Побудувати упорядкування

	Звідки 1	Куди 1	Звідки 2	Куди 2	Звідки 3	Куди
	1	5	1	6	1	7

L= 15

S=

Граф:

Рис. 4.8. – Заповнення початкових полів задачі

На рис. 4.9 зображено розв’язання прикладу 1.

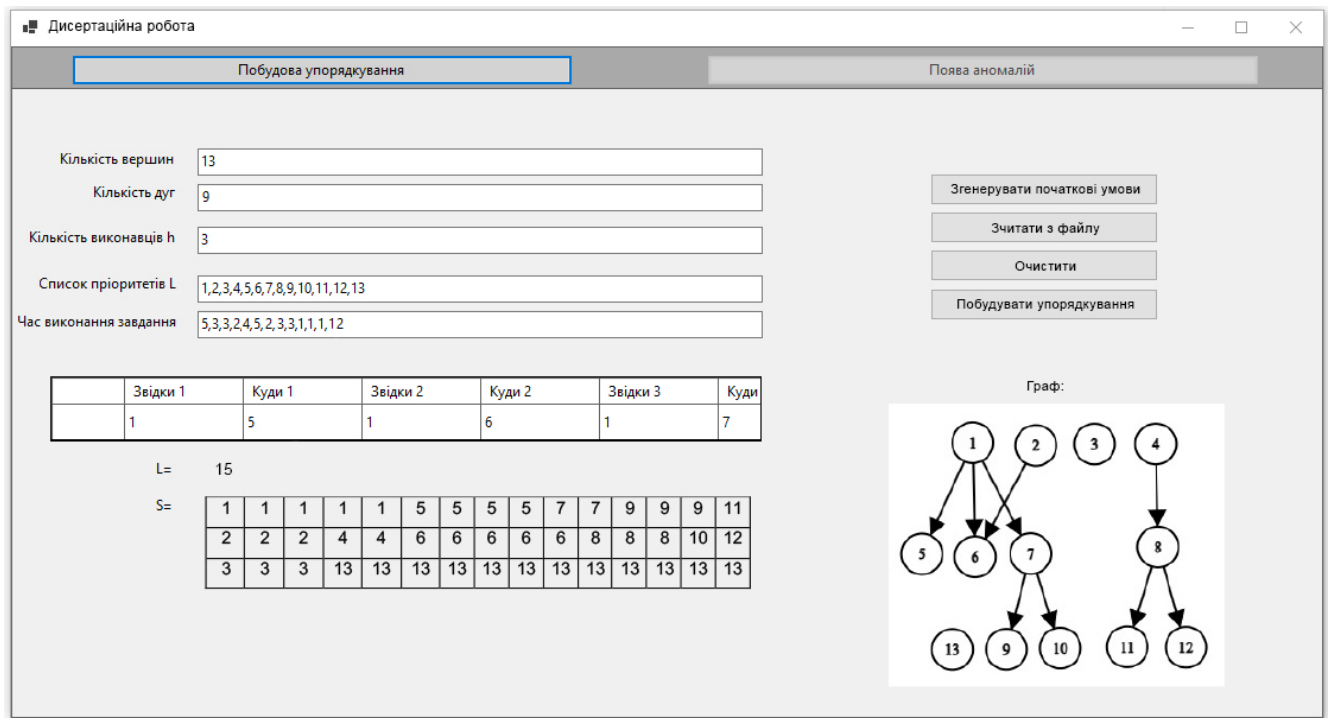


Рис. 4.9 – Побудова оптимального упорядкування та візуалізація графу G_1

Для аналізу виникнення аномалій відкриваємо вікно «Поява аномалій» (рис 4.10). Воно стає активним за умови успішної побудови упорядкування у першій вкладці «Побудова упорядкування».

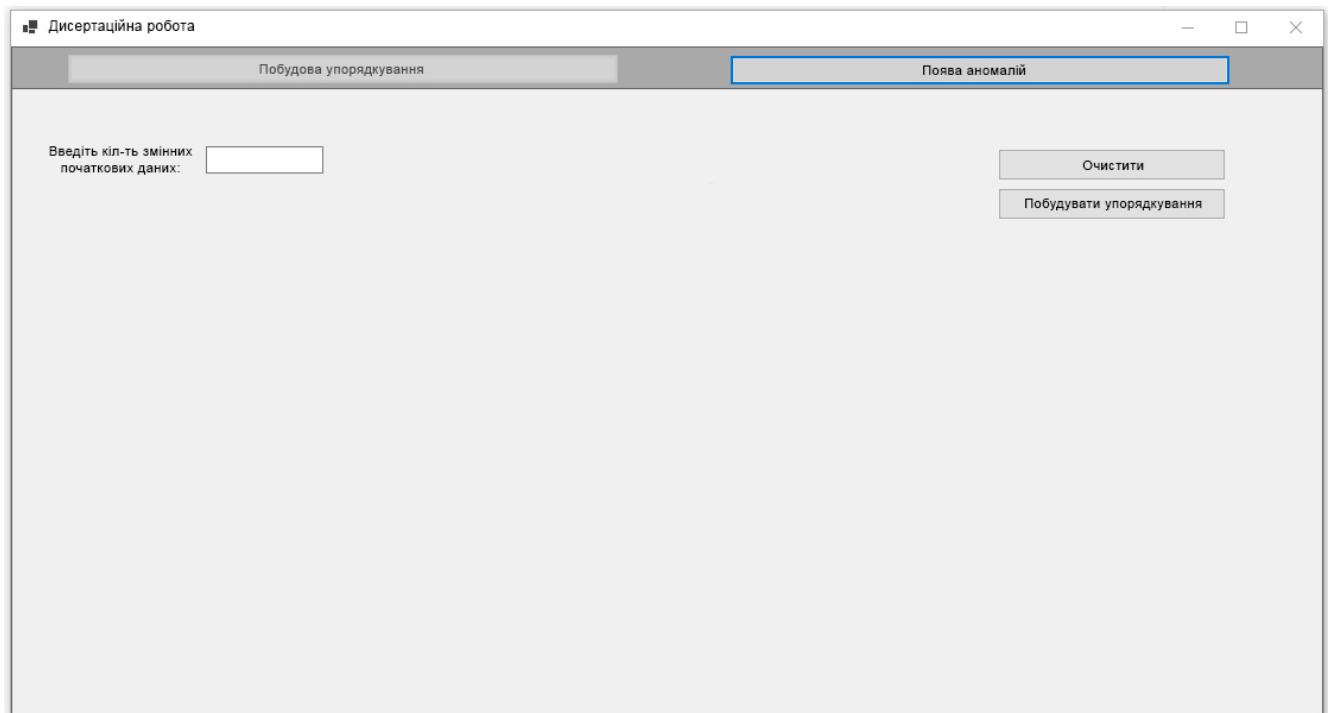


Рис. 4.10. – Розділ «Поява аномалій»

Після натискання кнопки «Поява аномалій», з'являється поле «Введіть кількість змінних початкових даних». Після введення значення з'являється відповідна кількість випадаючих списків, у яких необхідно обрати відповідний змінний параметр та задати нове значення (рис 4.11).

The screenshot shows a window titled "Дисертаційна робота". Inside, there are two tabs: "Побудова упорядкування" and "Поява аномалій". The "Поява аномалій" tab is selected and highlighted with a blue border. Below the tabs, there are input fields and buttons. On the left, there is a label "Введіть кількість змінних початкових даних:" followed by a text input field containing the number "2". Below this, there is a label "Введіть нові значення:" followed by two sets of controls. The first set has a dropdown menu showing "h" and a text input field containing "4". The second set has a dropdown menu showing "T" and a text input field containing the list "5,3,3,2,4,5,2,3". On the right side of the form, there are two buttons: "Очистити" (top) and "Побудувати упорядкування" (bottom).

Рис. 4.11. – Заповнення змінних параметрів задачі

Для побудови оптимального упорядкування з оновленими початковими параметрами, необхідно натиснути кнопку «Побудувати упорядкування» (рис 4.12). На екрані з'являється упорядкування в табличному вигляді, визначається його довжина та співвідношення значень довжини упорядкування до покращень початкових параметрів задачі та після. Якщо значення зменшилось, отримуємо від'ємне значення, якщо збільшилось – додатне. Також є можливість очистити поля (кнопка «Очистити») та «Зберегти результати» у файл (рис 4.12). Також реалізована можливість застосування алгоритму для зміни списку пріоритетів (відповідна кнопка під побудованим упорядкуванням).

Дисертаційна робота

Побудова упорядкування

Поява аномалій

Введіть к-ть змінних початкових даних:

Введіть нові значення:

$I = 19$ $\Delta I = +26.7\%$

$S =$

1	1	1	5	5	5	5	12												
2	2	2	6	6	6	6	6												
3	3	3	7	7	9	9	9												
4	4	8	8	8	10	11	13	13	13	13	13	13	13	13	13	13	13	13	13

Застосувати алгоритм зміни списку пріоритетів

Очистити

Побудувати упорядкування

Зберегти результати

Рис. 4.12. – Пошук аномальних випадків

Після застосування алгоритму зміни списку пріоритетів, будується нове упорядкування та визначаються нові L' , значення довжини l'' та ΔI (рис 4.13).

Дисертаційна робота

Побудова упорядкування

Поява аномалій

Введіть к-ть змінних початкових даних:

Введіть нові значення:

$I = 19$ $\Delta I = +26.7\%$

$S =$

1	1	1	5	5	5	5	12												
2	2	2	6	6	6	6	6												
3	3	3	7	7	9	9	9												
4	4	8	8	8	10	11	13	13	13	13	13	13	13	13	13	13	13	13	13

Застосувати алгоритм зміни списку пріоритетів

$L' = 13, 1, 6, 5, 7, 9, 10, 2, 4, 8, 11, 12, 3$

$I'' = 15$ $\Delta I = 0\%$

$S'' =$

13	13	13	13	13	13	13	13	13	13	13	13	3	3	3					
1	1	1	1	1	6	6	6	6	6	8	8	8	10	11					
2	2	2	4	4	5	5	5	5	7	7	9	9	9	12					

Рис. 4.13. – Застосування алгоритму зміни пріоритетів

ВИСНОВКИ

У дисертаційній роботі вивчалася актуальна наукова проблема, пов'язана з теоретичним обґрунтуванням природи виникнення аномальних випадків у задачах комбінаторної оптимізації, зокрема в задачах паралельного упорядкування вершин орграфів.

Вперше комплексно досліджено механізми виникнення аномалій у задачах даного типу. Сформульовано нові умови можливої появи аномальних випадків та умови, при яких цих випадків можна уникнути.

Основні результати, отримані в роботі, узагальнено у таких висновках:

1. Визначено поняття нових аномалій у задачах дискретної оптимізації, які проявляються у вигляді парадоксального погіршення значення цільової функції при покращенні параметрів задачі.

2. Проведено дослідження впливу змін нових початкових умов на виникнення аномалій, зокрема:

- зміни продуктивності виконавців;
- модифікації списків пріоритетів;
- видаленні або фіксації робіт за виконавцями.

3. Вперше показано, що одночасне покращення декількох параметрів (наприклад, зменшення часу виконання робіт і збільшення кількості виконавців) може спричиняти нові типи аномальних ефектів, що можуть не виникати при зміні одного з параметрів.

4. Запропоновано нові алгоритмічні підходи для подолання впливу аномалій, зокрема:

- динамічна побудова списків пріоритетів з урахуванням структури графа;
- моделі стабілізації довжини упорядкувань;
- алгоритми з підтримкою переривань задач і перепризначенням ресурсів.

Дані підходи дозволяють адаптивно реагувати на зміни умов задачі та забезпечувати стабільність отриманих упорядкувань.

5. Вперше введено поняття стабілізації довжини оптимального упорядкування, що дозволяє оцінити чутливість задачі до змін вхідних параметрів.

6. Продemonстровано прикладні можливості використання отриманих результатів у задачах:

- планування виробничих і транспортних процесів;
- оптимізації навантажень на обчислювальні кластери;
- моделювання роботи енергетичних систем;
- розподілу задач у хмарних та мультипроцесорних обчисленнях;
- аналізу ефективності сенсорних мереж та систем реального часу.

Результати дослідження можуть бути адаптовані до задач з певних галузей, які зводяться до задач паралельного упорядкування вершин орграфів.

Загалом, результати дослідження становлять вагомий внесок у розвиток теорії дискретної оптимізації, зокрема в контексті спеціальних задач розкладів, і мають як теоретичне, так і практичне значення. Вони створюють підґрунтя для подальших досліджень у напрямку вивчення поведінки складних оптимізаційних систем, побудови адаптивних алгоритмів і розвитку методів виявлення нестабільних режимів функціонування відповідних систем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Малієнко О.О. Задача оптимального розподілу ресурсів для випадку обмежень на їх використання. / О.О. Малієнко, В.А. Турчина // *Збірник наукових праць «Системні технології»*. 2025. Вип. 2. С. 75-81.
2. Maliienko O.O. Research on the relationship between anomalous cases in parallel scheduling problems and executor performance. / О.О. Maliienko, V.A. Turchyna // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2024. Вип. 24. С. 127-133.
3. Maliienko O.O. Analysis of the impact of task prioritization lists on the potential for avoiding anomalies in task scheduling. / О.О. Maliienko, V.A. Turchyna // *System technologies*. 2024. Вип. 6. С. 167-174.
4. Малієнко О.О. Порівняльний аналіз аномалій для прямих та зворотних графів. / О.О. Малієнко, В.А. Турчина // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2023. Вип. 23. С. 161-170.
5. Maliienko O.O. The study of the influence of combined changes in the initial data on the occurrence of anomalies for resource allocation. / О.О. Maliienko, V.A. Turchyna // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2022. Вип. 22. С. 106-112.
6. Челпанова О.О. Узагальнення аномальних випадків у задачах упорядкування. / О.О. Челпанова, В.А. Турчина // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2021. Вип. 21. С. 220-226.
7. Maliienko O.O. Conditions for length stabilization in parallel ordering problems. / О.О. Maliienko, V.A. Turchyna // *Theoretical and empirical scientific research: concept and trends*. Oxford, 2024. Вип. 6. С. 216-221.
8. Турчина В.А. Про аномальні випадки в задачах розподілу електроенергії. / В.А. Турчина, О.О. Малієнко // *XXVII Міжнародна конференція «Автоматика 2024»*, 20-22 листопада 2024. м. Дніпро, 2024. С. 202-204.

9. Малієнко О.О. Дослідження впливу переривань на виникнення аномалій у задачах паралельного упорядкування. / О.О. Малієнко, Є.О. Коваленко // *Комбінаторні конфігурації та їхні застосування: Матеріали XXVI Міжнародного науково-практичного семінару присвяченого пам'яті професора Донця Г.П.* м. Кропивницький - Запоріжжя - Київ, 2024. С. 103-107.
10. Малієнко О.О. Аналіз впливу збільшення розмірності задачі на довжину паралельного упорядкування. / О.О. Малієнко // *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2023): Матеріали XXI міжнародної науково-практичної конференції до 105-річчя Дніпровського національного університету імені Олеся Гончара, 22-24 листопада 2023.* м. Дніпро, 2023. С. 200-201.
11. Турчина В.А. Стабілізація довжини упорядкування та аналіз впливу змін початкових даних на оптимальність. / В.А. Турчина, О.О. Малієнко // *Комбінаторні конфігурації та їхні застосування: Матеріали XXV Міжнародного науково-практичного семінару імені А.Я. Петренюка, 14-16 червня 2023.* м. Запоріжжя - Кропивницький, 2023. С. 79-82.
12. Турчина В.А. Наявність аномалій при побудові паралельних упорядкувань для одного класу графів. / В.А. Турчина, О.О. Малієнко // *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2022): Матеріали XX ювілейної міжнародної науково-практичної конференції, 23-25 листопада 2022.* м. Дніпро, 2022. С. 207.
13. Турчина В.А. Аномальні випадки при упорядкуванні деревовидних структур. / В.А. Турчина, О.О. Челпанова // *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2021): Матеріали XIX міжнародної науково-практичної конференції, 17-19 листопада 2021.* м. Дніпро, 2021. С. 194-195.
14. Турчина В.А. Узагальнення аномалій в задачах упорядкування. / В.А. Турчина, О.О. Челпанова // *Комбінаторні конфігурації та їхні застосування: Матеріали XXIII Міжнародного науково-практичного семінару*

імені А.Я. Петренюка, присвяченого 70-річчю Льотної академії Національного авіаційного університету, 13-15 травня 2021. м. Запоріжжя - Кропивницький, 2021. С. 188-191.

15. Graham R. L. Bounds on multiprocessing timing anomalies. / R. L. Graham // *SIAM J. Appl. Math.* 1969. Vol. 17, no. 2. P. 416–429.
16. Lundqvist T. Timing anomalies in dynamically scheduled microprocessors. / T. Lundqvist, P. Stenstrom // *Proceedings of the 20th IEEE Real-Time Systems Symposium.* 1999. P. 12–21.
17. Schneider J. *Combined schedulability and WCET analysis for real-time operating systems.* / J. Schneider // PhD thesis, Saarland University, Germany. 2003. 202 p.
18. Kolota J. Graham's anomalies in case of parallel computation electromagnetic phenomena. / J. Kolota, J. Smykowski, S. Stepień // *International Journal Of Communications.* 2007. Vol. 1, no. 2. P. 17–21.
19. Турчина В. А. Наближені алгоритми побудови оптимальних паралельних упорядкувань заданої довжини. / В. А. Турчина, Н. К. Федоренко // *Питання прикладної математики і математичного моделювання: зб. наук. праць.* Дніпропетровськ: ДНУ, 2010. С. 312–319.
20. Турчина В. А. Алгоритми побудов всіх паралельних упорядкувань заданої довжини. / В. А. Турчина, Н. К. Федоренко // *Питання прикладної математики і математичного моделювання: зб. наук. праць.* Дніпропетровськ: ДНУ, 2011. С. 263–268.
21. Sergienko I. V. Algorithm Unions for Solving Discrete Optimization Problems. / I. V. Sergienko, V. P. Shylo, V. O. Roshchyn // *Cybernetics and Systems Analysis.* 2023. Vol. 59, no. 5. P. 753–762.
22. Gulyanitskii L. F. Discrete optimization methods for multiprocessor computer systems. / L. F. Gulyanitskii, I. V. Sergienko, A. N. Khodzinskii // *Cybernetics.* 1988. Vol. 24, no. 4. P. 418–427.
23. Kiseleva E.M. Application of optimal set partitioning theory to solving problems of artificial intelligence and pattern recognition. / E. M. Kiseleva,

O. M. Prytomanova, L. L. Hart // *System Research & Information Technologies*. 2021. Vol. 4. P. 91–101.

24. Kiseleva O. M. Software for solving the two-stage Location-allocation problems. / O.M. Kiseleva, O.M. Prytomanova, D.M. Lebediev, O.A. Filat // *Питання прикладної математики і математичного моделювання*. Дніпро, 2023. Вип. 23. С. 94–100.

25. Bulat A.F. Generalized models of logistics problems and approaches to their solution based on the synthesis of the theory of optimal partitioning and neuro-fuzzy technologies. / A. F. Bulat, E. M. Kiseleva, L. L. Hart, O. M. Prytomanova // In: Zgurovsky M., Pankratova N. (eds) *Studies in Computational Intelligence*. Springer: Cham, 2023. Vol. 1107, Chapter 21. P. 355–376.

26. Yakovlev S. Optimization on Combinatorial Configurations Using Genetic Algorithms. / S. Yakovlev, O. Kartashov, O. Pichugina // *Proceedings of The Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, 2019. P. 28–40.

27. Romanova T. E. Parallel computing technologies for solving optimization problems of geometric design. / T. E. Romanova, P. I. Stetsyuk, A. M. Chugay, S. B. Shekhovtsov // *Cybernetics and Systems Analysis*. 2019. Vol. 55. P. 894–904.

28. Koliechkina L. A graph-theoretic approach to multiobjective permutation-based optimization. / L. Koliechkina, O. Pichugina, S. Yakovlev // *International Conference on Optimization and Applications*. Cham: Springer International Publishing. 2019. P. 383–400.

29. Pichugina O. Continuous extensions on Euclidean combinatorial configurations. / O. Pichugina, S. Yakovlev // *Buletinul Academiei de Ştiinţe a Republicii Moldova Matematica*. 2022. Vol. 98. P. 3–21.

30. Чуб І. А. Математична модель та розв’язок оптимізаційної задачі розподілу ресурсів проекту. / І. А. Чуб, М. В. Новожилова, І. В. Беленченко // *Системи обробки інформації*. 2011. Вип. 2. С. 291–294.

31. Kozin I.V. Simulation of Annealing Algorithm for the Flat Rectangular Cutting Problem. / I. V. Kozin, U. Kh. Narzullaev, O. V. Sardak, Z. R. Sabirov //

International Journal of Theoretical and Applied Issues of Digital Technologies. 2023. Vol. 1, no. 3. P. 16–24.

32. Kozin I.V. Mathematical model of different type bin packing. / I. V. Kozin, S. Yu. Borue, O. V. Kryvtsun // *Bulletin of Zaporizhzhia national university. Economic sciences*. 2016. Vol. 2. P. 85–92.

33. Kozin I.V. Fragmentary Structures in a Two-Dimensional Strip Packing Problem. / I. V. Kozin, S. E. Batovskyi // *Cybernetics and Systems Analysis*. 2019. Vol. 55. P. 943–948.

34. Гук Н. Аналіз структури сайту з використанням поняття модулярності. / Н. Гук, С. Диханов, І. Долотов // *Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки*. 2020. С. 99–114.

35. Долотов І. О. Кластеризація зваженого вебграфу із використанням модулярності. / І. О. Долотов, Н. А. Гук // *Питання прикладної математики і математичного моделювання*. Дніпро, 2023. Вип. 23. С. 45–52.

36. Guk N. Design of a Recommendation System Based on the Transition Graph. / N. Guk, O. Verba, V. Yevlakov // *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 3, no. 4. P. 24–31.

37. Hulianytskyi L. Formalization and classification of combinatorial optimization problems. / L. Hulianytskyi, I. Riasna // *Optimization Methods and Applications: In Honor of Ivan V. Sergienko's 80th Birthday*. 2017. P. 239–250.

38. Gulyanitskii L. F. Discrete optimization methods for multiprocessor computer systems. / L. F. Gulyanitskii, I. V. Sergienko, A. N. Khodzinskii // *Cybernetics*. 1988. Vol. 24, no. 4. P. 418–427.

39. Донець Г. П. Графовий підхід до розв'язання задач комбінаторного розпізнавання. / Г. П. Донець // *Кібернетика і системний аналіз*. 2017. Вип. 53, №. 6. С. 44–53.

40. Semeniuta M. F. Group labeling of some graphs. / M. F. Semeniuta, G. A. Donets // *Cybernetics and Systems Analysis*. 2020. Vol. 56, no. 5. P. 701–709.

41. Semeniuta M. F. Combinatorial Configurations in the Definition of Antimagic Labelings of Graphs. / M. F. Semeniuta // *Cybernetics and Systems Analysis*. 2021. Vol. 57, no. 2. P. 30–40.
42. Турчина В. А. Узагальнення задачі паралельного упорядкування на випадок двох типів вершин. / В. А. Турчина, Н. К. Федоренко // *Питання прикладної математики і математичного моделювання: зб. наук. праць*. Дніпропетровськ: ДНУ, 2007. С. 303–316.
43. Турчина В. А. Алгоритми розв'язання узагальненої задачі упорядкування. / В. А. Турчина, Н. К. Федоренко // *Питання прикладної математики і математичного моделювання: зб. наук. праць*. Дніпропетровськ: ДНУ, 2006. С. 191–199.
44. Турчина В. А. Дослідження узагальнених моделей в задачах упорядкування. / В. А. Турчина, Н. К. Федоренко // *Збірник праць 6-ї міжнародної конференції «Сучасні проблеми гуманізації та гармонізації управління»*. Харків, 2005. С. 190–192.
45. Турчина В. А. Дослідження оцінок довжини паралельного упорядкування вершин графу. / В. А. Турчина, К. Д. Караваєв // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. м. Дніпро, 2018. Вип. 18. С. 186–195.
46. Турчина В. А. Дослідження задачі упорядкування з перериваннями для одного підкласу дерев. / В. А. Турчина, Є. О. Коваленко // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. м. Дніпро, 2023. Вип. 23, 118–125.
47. Турчина В. А. Вплив початкових даних задачі паралельного упорядкування з перериваннями на оптимальність розв'язку. / В. А. Турчина, Є. О. Коваленко // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. м. Дніпро, 2022. Вип. 22, 158–167.
48. Василькович П. Алгоритми побудови паралельних упорядкувань, засновані на аналізі структури графів. / П. Василькович, В. Турчина // *Grail of Science*. 2022. Вип. 12-13. С. 300–306.

49. Наконечна Т. В. Про вплив переривань у виконанні робіт при управлінні проєктами. / Т. В. Наконечна // *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. м. Дніпро, 2024. Вип. 24. С. 151–158.
50. Garey M.R. Computers and Intractability: A Guide to the Theory of NP-Completeness. / M.R. Garey, D.S. Johnson // San Francisco: W.H. Freeman. 1979. 340p.
51. Coffman, E. G. Computer and job-shop scheduling theory. / E. G. Coffman, J. Bruno // *New York: Wiley*. 1976. 299 p.
52. Hu T. C. Parallel Sequencing and Assembly Line Problems. / T. C. Hu // *Operations Research*. 1961. Vol. 9, no. 6. P. 841–848.
53. Baptiste P. Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems. / P. Baptiste, C. Le Pape, W. Nuijten // *Norwell: Kluwer Academic*. 2001. 421 p.
54. Tovey C. A. A simplified anomaly and reduction for precedence constrained multiprocessor scheduling. / C. A. Tovey // *SIAM Journal on Discrete Mathematics*. 1990. Vol. 3, no. 4. P. 567–574.
55. Panwalkar S. S. Analysis of flow shop scheduling anomalies. / S. S. Panwalkar, C. Koulamas // *European Journal of Operational Research*. 2020. Vol. 280, no. 1. P. 25–33.
56. Panwalkar S. S. Anomalies in special permutation flow shop scheduling problems. / S. S. Panwalkar, C. Koulamas // *Chinese Journal of Mechanical Engineering*. 2020. Vol. 33, no. 1. Article 46.
57. Graham R. L. Bounds for certain multiprocessing anomalies. / R. L. Graham // *Bell System Technical Journal*. 1966. Vol. 45, no. 9. P. 1563–1581.
58. Zhang R. Learning combinatorial optimization algorithms over graphs. / R. Zhang // *Proceedings of NeurIPS 2020*. 2020. [arXiv:2007.11258].
59. Della Croce F. Longest Processing Time rule for identical parallel machines revisited. / F. Della Croce, R. Scatamacchia // arXiv preprint. 2018. [arXiv:1801.05489].

60. Gatto M. On the robustness of Graham's algorithm for online scheduling. / M. Gatto, P. Widmayer // *Algorithms and Data Structures. WADS 2007*. 2007. Vol. 4619. P. 349–361.
61. Reddi S. S. Graham's schedules and the number partition problem. / S. S. Reddi // arXiv preprint. 2008. [arXiv:0808.1119].
62. Taha H. A. Operations Research: An Introduction. / H. A. Taha // 9th ed. Pearson, 2011. 832 p.
63. Garey M. R. Scheduling tasks with non-uniform deadlines on two processors. / M. R. Garey, D. S. Johnson // *Journal of the ACM*. 1976. Vol. 23, no. 3. P. 461–467.
64. Garey M. R. Two-Processor Scheduling with Start-Times and Deadlines. / M. R. Garey, D. S. Johnson // *SIAM J. Comput.* 1977. Vol. 6, no. 3. P. 416–426.
65. Sethi R. Scheduling graphs on two processors. / R. Sethi // *SIAM Journal on Computing*. 1976. Vol. 5, no. 1. P. 73–82.
66. Коломійцев, О. Задачі дискретної оптимізації та їх постановка. / О. Коломійцев, С. Осієвський, В. Третяк, З. Закіров, А. Романюк, В. Нікітченко, Є. Логвиненко, А. Лисиця // *InterConf*. 2021. Вип. 75. С. 285–302.
67. Третяк, В. Архітектури паралельних обчислювальних структур для рішення задач дискретної оптимізації. / В. Третяк, С. Осієвський, О. Усачова, А. Ірха, А. Булай, О. Бабіч, Н. Шамрай // *InterConf*. 2021. С. 462–479.
68. Коломійцев, О. Задачі дискретної оптимізації та їх постановка для розміщення засобів захисту в розподіленій системі. / О. Коломійцев, Д. Голубничий, Г. Коц, В. Третяк, Д. Євстрат, А. Лисиця // *Збірник наукових праць ЛОГОΣ*. 2020. С. 36-41.
69. Fujii M. Optimal sequencing of two equivalent processors. / M. Fujii, T. Kasami, K. Ninomiya // *SIAM J. Appl. Math.* 1969. Vol. 17, no. 4. P. 784–789.
70. Lenstra J. K. Complexity of scheduling under precedence constraints. / J. K. Lenstra, A. H. G. Rinnooy-Kan // *Operations Res*. 1978. Vol. 26. P. 22–35.

71. Chithra A. Energy Efficient Clustering Algorithms for Wireless Sensor Networks – A Survey. / A. Chithra, R. Shantha Selva Kumari // *International Journal of Review in Electronics & Communication Engineering*. 2014. Vol. 2, no. 6. P. 176–180.
72. Garcia-Teodoro P. Anomaly-based network intrusion detection: Techniques, systems and challenges. / P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez // *Computers & Security*. 2009. Vol. 28, no. 1-2. P. 18–28.
73. Liu, Y. CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. / Y. Liu, B. Schmidt, D. L. Maskell // *BMC Research Notes*. 2009. Vol. 2, no. 1. Article 73.
74. Degiannakis, S. Oil and stock returns: Evidence from European industrial sector indices in a time-varying environment. / S. Degiannakis, G. Filis, C. Floros // *Journal of International Financial Markets, Institutions and Money*. Vol. 26. P. 175–191.
75. Michalakes, J. The Weather Research and Forecast Model: Software Architecture and Performance. / J. Michalakes, J. Dudhia, D. Gill, T. Henderson, J. B. Klemp, W. Skamarock, W. Wang // *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing in Meteorology*. 2004. P. 156–168.
76. Dean, J. Large scale distributed deep networks. / J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, A. Y. Ng // *Advances in Neural Information Processing Systems*. 2012. Vol. 25. P. 1223–1231.
77. Dhillon H. S. Load-Aware Modeling and Analysis of Heterogeneous Cellular Networks. / H. S. Dhillon, R. K. Ganti, J. G. Andrews // 2013. arXiv.1204.1091.
78. Турчина, В. А. Декомпозиційний підхід до розв’язання узагальненої задачі паралельного упорядкування. / В. А. Турчина, Є. О. Коваленко // *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2016): Матеріали XIV міжнародної науково-практичної конференції, 16-18 листопада 2016. м. Дніпро, 2016. Вип. 22. С. 94–95.*
79. Cormen T. H. Introduction to Algorithms. / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein // *MIT Press*. 2009. 1313 p.
80. Nemhauser G. L. Integer and Combinatorial Optimization. /

G. L. Nemhauser, L. A. Wolsey // *Wiley-Interscience*. 1999. 772 p.

81. Graham R. L. Optimization and approximation in deterministic sequencing and scheduling: a survey. / R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan // *Annals of Discrete Mathematics*. 1979. Vol. 5. P. 287–326.

82. Coffman E.G. Approximation algorithms for bin packing: A survey. / E.G. Coffman, M.R. Garey, D.S. Johnson // *Approximation Algorithms for NP-Hard Problems*. Ed. D.S. Hochbaum. Boston: PWS Publishing. 1996. P. 46–93.

83. Pinedo M. L. Scheduling: Theory, Algorithms, and Systems. 5th ed. / M. L. Pinedo // Springer, 2016. 698 p.

84. Lenstra J. K. Elements of scheduling. / J. K. Lenstra, D. B. Shmoys // *arXiv:2001.06005*. 2020. P. 1–30.

85. LeBlanc L. J. Critical path method in project scheduling. / L. J. LeBlanc, E. K. Morlok // *Journal of Operations Management*. 2005. Vol. 25, no. 2. P. 211–222.

86. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12th ed. / H. Kerzner // Wiley, 2017. 848 p.

87. Nemhauser, G. L., Wolsey, L. A. Integer and Combinatorial Optimization. / G. L. Nemhauser, L. A. Wolsey // *Wiley-Interscience*. 1999. 772 p.

88. Gibbons A. Algorithmic Graph Theory. / A. Gibbons // Cambridge University Press. 1985. 272 p.

89. Ranjbar M. Resource leveling in project scheduling with activity splitting allowed. / M. Ranjbar, F. Kianfar, S. Shadrokh // *International Journal of Project Management*. 2011. Vol. 29, no. 5. P. 564–574.

90. Diestel R. Graph Theory. 5th ed. / R. Diestel // Springer, 2017. 429 p.

91. Page L. The PageRank citation ranking: Bringing order to the web. / L. Page, S. Brin, R. Motwani, T. Winograd // *Technical Report*. Stanford InfoLab. 1999. 17 p.

92. Jurafsky D. Speech and Language Processing. 3rd ed. / D. Jurafsky, J.H. Martin // 2023. 599 p.

93. Kempe D. Maximizing the spread of influence through a social network. / D. Kempe, J. Kleinberg, E. Tardos // *Theory of Computing*. 2003. P. 137–146.

94. Wenzel I. Principles of timing anomalies in superscalar processors. /

- I. Wenzel, R. Kirner, P. Puschner, B. Rieder // *Fifth International conference on quality software*. 2005. P. 295–306.
95. Kocher P. Spectre attacks: Exploiting speculative execution. / P. Kocher, D. Genkin, D. Gruss // *2019 IEEE Symposium on Security and Privacy (SP)*. 2019. 19 p.
 96. Mowry T. C. Design and evaluation of a compiler algorithm for prefetching. / T. C. Mowry, M. S. Lam, A. Gupta // *Proceedings of the 5th international conference on Architectural support for programming languages and operating systems*. 1992. P. 62–73.
 97. Hennessy J. L. Computer Architecture: A Quantitative Approach. 6th ed. / J. L. Hennessy, D. A. Patterson // Morgan Kaufmann. 2017. 936 p.
 98. McCool, M. Structured parallel programming: patterns for efficient computation. 1st ed. / M. McCool, A. D. Robison, J. Reinders // Morgan Kaufmann. 2012. 432 p.
 99. Lam M. S. Software pipelining: An effective scheduling technique for VLIW machines. / M. S. Lam // *ACM SIGPLAN Notices*. 2004. Vol. 39, no. 4. P. 244–256.
 100. Graham R. L. Bounds for certain multiprocessing anomalies. / R. L. Graham // *Bell System Technical Journal*. 1966. Vol. 45, no. 9. P. 1563–1581.
 101. Coffman E. G. Optimal scheduling for two-processor systems. / E. G. Coffman, R. L. Graham // *Acta Informatica*. 1972. Vol. 1. P. 200–213.
 102. Brucker P. List scheduling. / P. Brucker // *Springer Reference Work Entry*. 2013. P. 100–110.
 103. Brucker P. Scheduling Algorithms. 5th ed. / P. Brucker // *Springer*. 2006. 367 p.
 104. Nemhauser G. L., Wolsey L. A. Integer and Combinatorial Optimization. / G. L. Nemhauser, L. A. Wolsey // Wiley-Interscience, 1999. 766 p.
 105. Liu C. L., Layland J. W. Scheduling algorithms for multiprogramming in a hard-real-time environment. / C. L. Liu, J. W. Layland // *Journal of the ACM*. 1973. Vol. 20, no. 1. P. 46–61.
 106. Liu J. W. S. Real-Time Systems. / J. W. S. Liu // *Prentice Hall*. 2000. 536p.

107. Silberschatz A. Operating System Concepts. 10th ed. / A. Silberschatz, P. B. Galvin, G. Gagne // Wiley. 2018. 1278 p.
108. Khalid O. Task scheduling in cloud computing: Preemptive vs non-preemptive algorithms. / O. Khalid, A. Malik // *Journal of Cloud Computing*. 2019. Vol. 8, no. 1. P. 1–15.
109. Gamma E. Design Patterns: Elements of Reusable Object-Oriented Software. / E. Gamma, R. Helm, R. Johnson, J. Vlissides // Addison-Wesley. 1994. 395p.
110. Tarjan R. E. Depth-first search and linear graph algorithms. / R. E. Tarjan // *SIAM Journal on Computing*. 1972. Vol. 1, no. 2. P. 146–160.

ДОДАТОК А. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. **Малієнко О.О., Турчина В.А.** Задача оптимального розподілу ресурсів для випадку обмежень на їх використання. *Збірник наукових праць «Системні технології»*. 2025. Вип. 2. С. 75-81. doi: <https://doi.org/10.34185/1562-9945-2-157-2025-07>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1966/1236> (фахове видання, категорія Б).
2. **Maliienko O.O., Turchyna V.A.** Research on the relationship between anomalous cases in parallel scheduling problems and executor performance. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2024. Вип. 24. С. 127-133. doi: <https://doi.org/10.15421/322413>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/408/394> (фахове видання, категорія Б).
3. **Maliienko O.O., Turchyna V.A.** Analysis of the impact of task prioritization lists on the potential for avoiding anomalies in task scheduling. *Збірник наукових праць «Системні технології»*. 2024. Вип. 6. С. 167-174. doi: <https://doi.org/10.34185/1562-9945-6-155-2024-16>. Режим доступу до ресурсу: <https://journals.nmetau.edu.ua/index.php/st/article/view/1924> (фахове видання, категорія Б).
4. **Малієнко О.О., Турчина В.А.** Порівняльний аналіз аномалій для прямих та зворотних графів. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2023. Вип. 23. С. 161-170. doi: <https://doi.org/10.15421/322317>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/386> (фахове видання, категорія Б).
5. **Maliienko O.O., Turchyna V.A.** The study of the influence of combined changes in the initial data on the occurrence of anomalies for resource allocation. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2022. Вип. 22. С. 106-112. doi:

<https://doi.org/10.15421/322211>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/345> (фахове видання, категорія Б).

6. **Челпанова (Малієнко) О.О., Турчина В.А.** Узагальнення аномальних випадків у задачах упорядкування. *Збірник наукових праць «Питання прикладної математики і математичного моделювання»*. Дніпро, 2021. Вип. 21. С. 220-226. doi: <https://doi.org/10.15421/322122>. Режим доступу до ресурсу: <https://pmm.dp.ua/index.php/pmmm/article/view/327> (фахове видання, категорія Б).

7. **Maliienko O.O., Turchyna V.A.** Conditions for length stabilization in parallel ordering problems. *Theoretical and empirical scientific research: concept and trends*. Oxford, 2024. Вип. 6. С. 216-221. doi: <https://doi.org/10.36074/logos-02.02.2024.043>. Режим доступу до ресурсу: <https://archive.logos-science.com/index.php/conference-proceedings/article/view/1547>.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

8. Турчина В.А., **Малієнко О.О.** Про аномальні випадки в задачах розподілу електроенергії. *XXVII Міжнародна конференція «Автоматика 2024»*, 20-22 листопада 2024. м. Дніпро, 2024. С. 202-204. Режим доступу до ресурсу: [http://automatika2024.dp.ua/files/Автоматика-2024_\(тези_доповідей\).pdf](http://automatika2024.dp.ua/files/Автоматика-2024_(тези_доповідей).pdf) - [page=202](#).

9. **Малієнко О.О., Коваленко Є.О.** Дослідження впливу переривань на виникнення аномалій у задачах паралельного упорядкування. *Комбінаторні конфігурації та їхні застосування: Матеріали XXVI Міжнародного науково-практичного семінару присвяченого пам'яті професора Донця Г.П.* м. Кропивницький - Запоріжжя - Київ, 2024. С. 103-107. Режим доступу до ресурсу: https://zp.edu.ua/uploads/dept_s&r/2024/conf/6.3/CCTA-2024-proc.pdf#page=103.

10. **Малієнко О.О.** Аналіз впливу збільшення розмірності задачі на довжину паралельного упорядкування. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2023): Матеріали XXI міжнародної науково-практичної конференції до 105-річчя Дніпровського національного університету імені Олеся Гончара*, 22-24 листопада 2023. м.

Дніпро, 2023. С. 200-201. Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2023/11/mpzis-2023.pdf#page=200>.

11. Турчина В.А., **Малієнко О.О.** Стабілізація довжини упорядкування та аналіз впливу змін початкових даних на оптимальність. *Комбінаторні конфігурації та їхні застосування: Матеріали XXV Міжнародного науково-практичного семінару імені А.Я. Петренюка, 14-16 червня 2023. м. Запоріжжя - Кропивницький, 2023. С. 79-82.* Режим доступу до ресурсу: https://zp.edu.ua/uploads/dept_s&r/2023/conf/1.4/Petrenyuk_ISPS-25-proc.pdf#page=79.

12. Турчина В.А., **Малієнко О.О.** Наявність аномалій при побудові паралельних упорядкувань для одного класу графів. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2022): Матеріали XX ювілейної міжнародної науково-практичної конференції, 23-25 листопада 2022. м. Дніпро, 2022. С. 207.* Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2022/12/MPZIS-2022-1.pdf#page=207>.

13. Турчина В.А., **Челпанова (Малієнко) О.О.** Аномальні випадки при упорядкуванні деревовидних структур. *Математичне та програмне забезпечення інтелектуальних систем (MSSIS-2021): Матеріали XIX міжнародної науково-практичної конференції, 17-19 листопада 2021. м. Дніпро, 2021. С. 194-195.* Режим доступу до ресурсу: <http://mpzis.dnu.dp.ua/wp-content/uploads/2021/12/mpzis-2021.pdf#page=194>.

14. Турчина В.А., **Челпанова (Малієнко) О.О.** Узагальнення аномалій в задачах упорядкування. *Комбінаторні конфігурації та їхні застосування: Матеріали XXIII Міжнародного науково-практичного семінару імені А.Я. Петренюка, присвяченого 70-річчю Льотної академії Національного авіаційного університету, 13-15 травня 2021. м. Запоріжжя - Кропивницький, 2021. С. 188-191.* Режим доступу до ресурсу: https://www.glau.kr.ua/images/docs/sbornik/materiali_23_mnp_seminaru.pdf#page=188.