

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кваліфікаційна наукова
праця на правах рукопису

Скибицький Нікіта Максимович

УДК 519.6

ДИСЕРТАЦІЯ

**РОБАСТНІ АЛГОРИТМИ ДЛЯ КОМБІНАТОРНИХ ЗАДАЧ
МАРШРУТИЗАЦІЇ ТА ТРЕЙДІНГУ**

113 — Прикладна математика

11 — Математика та статистика

Подається на здобуття наукового ступеня
доктора філософії

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

_____ Н.М. Скибицький

Науковий керівник: **Семенов Володимир Вікторович**,
доктор фізико-математичних наук, професор

Київ — 2026

АНОТАЦІЯ

Скибицький Н.М. Робастні алгоритми для комбінаторних задач маршрутизації та трейдингу — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 113 — “Прикладна математика” (11 — “Математика та статистика”). — Київський національний університет імені Тараса Шевченка, Київ, 2026.

У роботі отримано нові теоретичні результати у п’яти галузях дискретної математики, що стали підґрунтям для побудови робастних методів оптимізації. Зокрема, досліджено алгебраїчні властивості операції побітового додавання в контексті адитивної комбінаторики. Для інтервальних множин натуральних чисел A та B доведено, що розмір їхньої побітової суми $|A \oplus B|$ має лінійну верхню оцінку відносно розмірів вихідних множин, що суттєво покращує наївну квадратичну оцінку. На основі цього теоретичного результату побудовано алгоритм, який дозволяє знаходити множину $A \oplus B$ за полі-логарифмічний час.

У галузі рядкових алгоритмів розроблено новий метод декомпозиції Ліндона, який використовує шарувату архітектуру структур даних та монотонний стек. Цей результат має важливе значення для задач лексикографічного упорядкування, зокрема для ефективного обернення перетворення Барроуза—Віллера, що застосовується в системах стиснення даних.

Встановлено зв’язок між задачами обчислення згорток послідовностей та комбінаторною оптимізацією. У роботі встановлено зведення задачі про $(\min, +)$ згортку до окремого випадку задачі комівояжера з квотою, що дозволило застосувати результати теорії тонкої обчислювальної складності для аналізу ефективності алгоритмів маршрутизації.

Для структур даних типу дерево розв’язано задачі пошуку та підрахунку зигзагоподібних послідовностей. Запропоновано ефективні алгоритми, що базу-

ються на методі динамічного програмування, використанні найменшого спільного предка (англ. *lowest common ancestor*) та техніці злиття “від меншого до більшого”. Ці методи переважають підходи, що ґрунтуються на декомпозиції центроїдів та декомпозиції “важкий—легкий” (англ. *heavy-light decomposition*).

Досліджено реберну зв’язність регулярних графів, зокрема циркулянтних. Запропоновано конструктивні методи пошуку та підрахунку мінімально зв’язних циркулянтних графів, що знаходить застосування у проектуванні надійних мережевих топологій, стійких до відмов окремих ланок.

У прикладній частині роботи побудовано ефективні алгоритми розв’язування задач комбінаторної оптимізації в умовах невизначеності. Запропоновано методи, що використовують структури даних, динамічне програмування, метод гілок та меж, кластеризацію та генетичні алгоритми. Проведено теоретичний аналіз констант апроксимації та ефективності.

Проведено ряд обчислювальних експериментів, що підтвердили перевагу розроблених методів над класичними евристичними для задач, які не допускають константної апроксимації за поліноміальний час. Показано, що хоча такі методи не є алгоритмами апроксимації з гарантованою оцінкою, вони мають практичне значення для задач робастної оптимізації.

У дисертації запропоновано нові перспективні алгоритми та отримані нові теоретичні результати про обчислювальну складність алгоритмів. Теоретичне значення мають методи доведень теорем про збіжність інтегральних характеристик та теорем про екстремальні оцінки. Розроблені алгоритми та програмні засоби можуть бути використані для планування маршрутів безпілотних літальних апаратів, оптимізації логістичних процесів, аналізу геномних послідовностей у біоінформатиці та в системах алгоритмічної торгівлі на фінансових ринках.

Ключові слова: задача комівояжера, динамічне програмування, безпілотний літальний апарат, чисельні методи, математичне моделювання, комп’ютерне моделювання, генетичний алгоритм, невизначеність, топологічна структура.

ABSTRACT

Skybytskyi N.M. Robust algorithms for combinatorial problems of routing and trading — Qualification scientific work in the form of manuscript.

Thesis for degree of doctor of philosophy, speciality 113 — “Applied mathematics” (11 — “Mathematics and statistics”). — Taras Shevchenko National University of Kyiv, Kyiv, 2026.

The work presents new theoretical results in five domains of discrete mathematics, which serve as a foundation for constructing robust optimization methods. Specifically, the algebraic properties of the bitwise addition operation are investigated within the context of additive combinatorics. For interval sets of natural numbers A and B , it is proven that the size of their bitwise sum $|A \oplus B|$ has a linear upper bound relative to the sizes of the initial sets, which significantly improves upon the naive quadratic bound. Based on this theoretical result, an algorithm has been constructed that allows finding the set $A \oplus B$ in polylogarithmic time.

In the field of string algorithms, a new method for Lyndon decomposition has been developed, utilizing a layered data structure architecture and a monotonic stack. This result is significant for lexicographical ordering problems, particularly for the efficient inversion of the Burrows-Wheeler Transform, which is widely applied in bioinformatics for genome analysis and in data compression systems.

A deep connection has been established between problems of sequence convolution computation and combinatorial optimization. Thesis investigates the $(\min, +)$ -convolution, for which, unlike classical convolution, no strongly subquadratic algorithm exists. A reduction of this problem to the special case of the quota traveling salesman problem is established, enabling the application of fine-grained computational complexity theory results to analyze the efficiency of routing algorithms.

For tree-based data structures, the problems of searching for and counting zigzag

sequences have been solved. Effective algorithms are proposed based on dynamic programming, the lowest common ancestor (LCA) concept, and the “small-to-large” merging technique. These methods outperform approaches based on centroid or heavy-light decomposition (HLD).

In the context of graph theory, the edge connectivity of regular graphs, specifically circulant graphs, has been investigated. Constructive methods for searching and counting minimally connected circulants are proposed, which is crucial for designing reliable network topologies resilient to individual link failures.

The objective of the applied part of the work is to construct effective algorithms for solving combinatorial optimization problems under uncertainty. Methods utilizing data structures, dynamic programming, the branch-and-bound method, clustering, and genetic algorithms are proposed. A theoretical analysis of approximation constants and efficiency has been conducted.

A series of computational experiments was performed, confirming the superiority of the developed methods over classical heuristics for problems that do not admit constant-factor approximation in polynomial time. It is demonstrated that although such methods are not formally approximation algorithms with guaranteed bounds, they possess significant practical value for robust optimization problems.

The work holds both theoretical and practical significance. The obtained analytical results contribute to fine-grained computational complexity theory, the theory of algorithms, and robust optimization. Results of the experimental part demonstrate the practical value of the developed software tools, which can be used for UAV route planning, logistics process optimization, and decision-making under uncertainty.

Key words: traveling salesman problem, dynamic programming, unmanned aerial vehicle, numerical methods, mathematical modeling, computer modeling, genetic algorithm, uncertainty, topological structure.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Статті у наукових фахових виданнях:

1. Skybytskyi N. M. and Denysov K. I. *Computational Equivalence of One-Dimensional Quota TSP Variant and $(\min, +)$ Convolution* // Journal of Numerical and Applied Mathematics. 2024. no. 2. P. 62–67. Access mode: <http://dx.doi.org/10.17721/2706-9699.2024.2.04>.
2. Skybytskyi N. M. *Robust Time to Buy and Sell Stock* // Journal of Numerical and Applied Mathematics. 2025. no. 1. P. 90–100. Access mode: <http://dx.doi.org/10.17721/2706-9699.2025.1.08>.
3. Skybytskyi N. M. *Two-Level Algorithm for the UAV Routing Problem with Moving Targets* // Cybernetics and Computer Technologies. 2025. Dec. no. 4. P. 29–36. Access mode: <http://dx.doi.org/10.34229/2707-451X.25.4.3>.

Публікації, що засвідчують апробацію матеріалів дисертації:

1. Skybytskyi N. M. *UAVRP With Moving Targets* // Proceedings of the XXII International Scientific-Practical Conference “Shevchenkivska Vesna”, April 11, 2024, Kyiv, Ukraine. P. 103.
2. Skybytskyi N. M. and Denysov K. I. *Equivalence of 1D K -TSP Variant and $(\min, +)$ Convolution* // Proceedings of the XVII International Scientific and Practical Conference “Information Technologies and Automation — 2024”. October 31–November 1, 2024, Odesa, Ukraine. P. 103.
3. Skybytskyi N. M. *Linear Upper Bound on a Segment XOR Cardinality* // Proceedings of the XXIII International Scientific-Practical Conference “Shevchenkivska Vesna”. April 10, 2025, Kyiv, Ukraine. P. 121.

4. Скибицький Н. М. *Робастна комбінаторна оптимізація у задачі планування відключень електроенергії* // Матеріали XII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених “Об’єднані наукою: перспективи міждисциплінарних досліджень”. 20–21 листопада 2025, Київ, Україна. С. 327.

ЗМІСТ

Вступ	12
Розділ 1. Огляд літератури	18
1.1. Фундаментальні засади оптимізації	19
1.1.1. Місце оптимізації у прикладній математиці	19
1.1.2. Неперервна та дискретна оптимізація	21
1.1.3. Оптимізація в умовах невизначеності	22
1.2. Комбінаторна оптимізація та складність алгоритмів	24
1.2.1. Теоретико-графові основи	25
1.2.2. Класи складності та NP-повнота	26
1.2.3. Точні, наближені та евристичні алгоритми	27
1.2.4. Кластеризація та попередні перетворення	28
1.3. Відкриті питання робастної апроксимації	29
1.3.1. $(\min, +)$ згортки	29
1.3.2. Маршрутизація БПЛА	29
1.4. Методологія проведення обчислювальних експериментів	30
1.4.1. Тестові набори даних	30
1.4.2. Метрики робастності та ефективності	31
1.5. Висновки до розділу 1	31
Розділ 2. Фундаментальна теорія алгоритмів	33
2.1. Вступ	33
2.1.1. Бітові операції	34
2.1.2. Рядкові алгоритми	36
2.1.3. Згортки послідовностей	38
2.1.4. Послідовності у деревах	40

2.1.5. Регулярні графи	42
2.2. Побітова поелементна сума множин	44
2.3. Лексикографічна декомпозиція рядка	49
2.4. Мінімум-плюс згортка	53
2.5. Зигзагоподібні послідовності у деревах	62
2.6. Реберна зв'язність регулярних графів	66
2.7. Висновки до розділу 2	73
Розділ 3. Маршрутизація та кластеризація БПЛА	75
3.1. Вступ	75
3.1.1. Маршрутизація літальних апаратів	75
3.1.2. Кластеризація рухомих об'єктів	77
3.2. Основна частина	79
3.2.1. Впорядковуюча сила	79
3.2.2. Формалізація циклічних переваг та неможливість задання повного порядку	81
3.2.3. Кластеризація та TLGA	83
3.2.4. Близькість рухомих об'єктів	84
3.2.5. Інтегральна метрика близькості	86
3.2.6. Алгоритми кластеризації на основі інтегральної метрики .	89
3.2.7. Чисельні експерименти та результати	90
3.3. Висновки до розділу 3	91
Розділ 4. Робастна торгівля акціями	93
4.1. Вступ	93
4.2. Основна частина	94
4.2.1. Класичні задачі	94
4.2.2. Класичні розв'язки	96
4.2.3. Робастні задачі	98
4.2.4. Робастні розв'язки	101

4.3. Висновки до розділу 4	106
Висновки	107
Список використаних джерел	109
Додаток А. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації	125
A.1. Статті у наукових фахових виданнях:	125
A.2. Матеріали конференцій:	125
A.3. Апробація результатів дисертації	126

ВСТУП

Актуальність теми. Сучасні прикладні задачі в галузях логістики, фінансового аналізу та обробки великих даних все частіше вимагають розв’язання проблем комбінаторної оптимізації в умовах невизначеності. Класичні методи оптимізації, орієнтовані на детерміновані вхідні дані, часто виявляються нестійкими до збурень, що призводить до суттєвого погіршення якості рішень на практиці. Водночас, зростання обсягів даних вимагає розробки нових, ефективніших алгоритмів для базових дискретних структур.

Актуальність теми зумовлена необхідністю подолання розриву між теоретичними оцінками обчислювальної складності та практичними потребами в робастних алгоритмах. Зокрема, важливим є дослідження алгебраїчних властивостей побітових операцій, рядкових алгоритмів та структур даних на деревах і графах, які слугують фундаментом для побудови ефективних методів маршрутизації БПЛА та фінансових стратегій. Розробка методів, що поєднують гарантовану точність з обчислювальною ефективністю, є критично важливою для систем підтримки прийняття рішень.

Мета і задачі дослідження. Метою роботи є розробка та теоретичне обґрунтування ефективних наближених та робастних алгоритмів для розв’язання задач комбінаторної оптимізації, а також дослідження фундаментальних властивостей дискретних структур, що лежать в їхній основі.

Задачі дослідження:

- Дослідити алгебраїчні властивості операції побітового додавання для інтервальних множин та розробити ефективний алгоритм обчислення її результату.
- Розробити новий метод лексикографічної декомпозиції рядків (розкладу Ліндона) для оптимізації обробки текстових даних.

- Встановити зв'язок між задачами обчислення згорток послідовностей та комбінаторною оптимізацією, зокрема звести задачу $(\min, +)$ -згортки до задачі комівояжера.
- Побудувати ефективні алгоритми пошуку та підрахунку зигзагоподібних послідовностей у деревоподібних структурах даних.
- Дослідити реберну зв'язність регулярних графів (циркулянтів) та розробити методи пошуку мінімально зв'язних топологій.
- Розробити математичні моделі та алгоритми кластеризації рухомих об'єктів для задач маршрутизації БПЛА в умовах невизначеності.
- Побудувати та дослідити робастні стратегії торгівлі акціями, що мінімізують жаль при коливаннях ринку.

Об'єкт дослідження. Задачі комбінаторної оптимізації на дискретних структурах (графах, деревах, рядках) та процеси прийняття рішень в умовах невизначеності.

Предмет дослідження. Методи побудови, аналізу складності та оцінки ефективності наближених і робастних алгоритмів для задач дискретної математики та оптимізації.

Методи дослідження. У роботі використано апарат теорії алгоритмів та обчислювальної складності (зокрема, теорії тонкої складності), методи динамічного програмування, теорії графів, адитивної комбінаторики, метод гілок і меж, а також евристичні підходи (генетичні алгоритми) та методи математичного моделювання.

Наукова новизна отриманих результатів. Основними результатами, які визначають наукову новизну та виносяться на захист, є такі результати:

- *Фундаментальні алгоритми:*

1. Вперше доведено лінійну верхню оцінку розміру побітової суми інтервальних множин та запропоновано полілогарифмічний алгоритм для її знаходження.

2. Розроблено новий алгоритм розкладу Ліндона на основі шаруватої архітектури даних та монотонного стека, що є важливим для задач стиснення даних (обернення перетворення Барроуза-Вілера).
3. Встановлено звідність задачі обчислення $(\min, +)$ -згортки до спеціального випадку задачі комівояжера (Quota TSP), що дозволяє застосувати результати теорії детальної складності до аналізу маршрутизації.
4. Запропоновано нові ефективні алгоритми пошуку зигзагоподібних послідовностей у деревах та конструктивні методи аналізу реберної зв'язності циркулянтних графів.
- *Прикладна оптимізація:*
5. Розроблено інтегральну метрику близькості для рухомих об'єктів та алгоритми їхньої кластеризації, що дозволяє оптимізувати маршрути груп БПЛА.
6. Запропоновано робастні методи для задачі торгівлі акціями, які, на відміну від класичних підходів, забезпечують стійкість розв'язків до флуктуацій вхідних даних.
7. Експериментально доведено перевагу розроблених методів над класичними евристиками для задач, що не допускають апроксимації зі сталим множником за поліноміальний час.

Практичне значення отриманих результатів. Розроблені алгоритми та програмні засоби можуть бути використані для планування маршрутів безпілотних літальних апаратів, оптимізації логістичних процесів, аналізу геномних послідовностей у біоінформатиці та в системах алгоритмічної торгівлі на фінансових ринках. Результати теоретичних досліджень можуть бути впроваджені в навчальні курси з теорії алгоритмів та дискретної математики.

Особистий внесок здобувача. Постановку завдань дослідження здійснив науковий керівник роботи доктор фізико-математичних наук, професор Семенов Володимир Вікторович.

Автор залучив магістра Університету Денисова Костянтина Ігоровича до дослідження згорток для здобуття останнім досвіду наукових досліджень. У спільних публікаціях авторів дисертації належать [1].

Всі інші теоретичні результати, доведення теорем, розробка алгоритмів, їхня програмна реалізація та проведення обчислювальних експериментів виконані здобувачем самостійно.

Публікації. Результати дисертації опубліковано в наукових працях [1–3].

Апробація результатів дисертації. Результати дисертації представлені на конференціях:

- XXII International Scientific-Practical Conference “Shevchenkivska Vesna” (Київ, 11 квітня 2024); [4]
- XVII International Scientific and Practical Conference “Information Technologies and Automation” (Одеса, 31 жовтня–1 листопада 2024); [5]
- XXIII International Scientific-Practical Conference “Shevchenkivska Vesna” (Київ, 10 квітня 2025); [6]
- XXI International Workshop “Dynamical System Modeling and Stability Investigations” (Київ, 8–10 травня 2025); [7]
- XII Всеукраїнська наукова-практична конференція студентів, аспірантів та молодих вчених “Об’єднані наукою: перспективи міждисциплінарних досліджень” (Київ, 20–21 листопада 2025); [8]
- XIV International Scientific and Practical Conference “Modern Cybernetics: Glushkov Readings” (Київ, 30 грудня 2025). [9]

Структура та обсяг дисертації. Дисертація складається з двох анотацій українською та англійською мовами, вступу, чотирьох розділів, висновків, списку використаних джерел та одного додатку. Повний обсяг дисертації становить 128 сторінок, з них основний текст займає 113 сторінок. Робота містить 29 рисунків та 7 таблиць. Список використаних джерел налічує 137 найменувань.

Перший розділ присвячено огляду літератури за темою дисертації. Проана-

лізовано фундаментальні засади неперервної та дискретної оптимізації, місце оптимізації в умовах невизначеності. Розглянуто теоретико-графові основи комбінаторної оптимізації, класи складності (зокрема NP-повноту) та існуючі підходи до розв’язання задач: точні, наближені та евристичні алгоритми. Окремлено відкриті питання робастної апроксимації, задачі маршрутизації БПЛА та методологію проведення обчислювальних експериментів.

Другий розділ містить фундаментальні результати в теорії алгоритмів. Досліджено бітові операції над множинами, рядкові алгоритми (лексикографічна декомпозиція), згортки послідовностей (зокрема мінімум-плюс згортка), а також алгоритми на деревах (зигзагоподібні послідовності) та графах (реберна зв’язність регулярних графів). Ці результати створюють теоретичну базу для побудови більш складних оптимізаційних методів.

Третій розділ присвячено прикладним задачам маршрутизації та кластеризації безпілотних літальних апаратів (БПЛА). Запропоновано математичну формалізацію циклічних переваг, яка унеможливорює задання повного порядку, та розроблено алгоритмічні підходи TLGA. Введено інтегральну метрику близькості рухомих об’єктів та на її основі побудовано алгоритми кластеризації. Наведено результати чисельних експериментів.

Четвертий розділ розглядає задачу визначення найкращого часу для торгівлі акціями в контексті робастної оптимізації. Проведено порівняльний аналіз класичних задач та розв’язків із робастними постановками, що враховують невизначеність ринкових даних. Запропоновано методи побудови робастних розв’язків, які мінімізують жаль.

Подяка. Автор щиро вдячний:

- своєму науковому керівнику, доктору фізико-математичних наук, професору Семенову Володимирі Вікторовичу за всебічну підтримку, цінні поради та формування високої математичної культури під час підготовки дисертації;

- магістру Київського національного університету імені Тараса Шевченка Костянтину Денисову за спільний розвиток у сфері спортивного програмування та плідні дискусії, що сприяли розширенню алгоритмічного світогляду;
- аспіранту Київського національного університету імені Тараса Шевченка Андрію Уразовському за постійну дружню підтримку та створення сприятливого академічного середовища;
- аспіранту Бостонського університету Драґосу Рістаче та випускнику ОНУ імені І. І. Мечникова Дмитру Плюсову за натхненну співпрацю в межах міжнародних змагань з програмування та обмін цінним практичним досвідом;
- випускнику Індійського технологічного інституту у Мадрасі Нішанку Сурешу та випускнику Індійського технологічного інституту у Харагпурі Рею Шреяну за конструктивний обмін ідеями щодо оптимізації алгоритмів та обговорення найкращих практик розробки програмного забезпечення.

РОЗДІЛ 1

ОГЛЯД ЛІТЕРАТУРИ

Цей огляд розпочинається з визначення місця задач оптимізації у математиці. В огляді виділено задачі комбінаторної оптимізації та проведено порівняльний аналіз із задачами неперервної оптимізації. Комбінаторна оптимізація також відома як дискретна завдяки скінченній природі множини можливих розв'язків відповідних задач.

Аналіз ключових праць підкреслює важливість робастної оптимізації і дозволяє виділити її основні відмінності від класичної оптимізації. Робастна оптимізація також відома як оптимізація в умовах невизначеності вхідних даних, що прямо протиставляється класичній оптимізації в умовах визначеності.

Наступною метою огляду є класифікація алгоритмів пошуку розв'язків робастних задач. Скінченність множини можливих розв'язків дозволяє побудову алгоритму повного перебору для пошуку оптимального розв'язку, що породжує потребу в поділі алгоритмів за швидкодією. В огляді проаналізовані основні результати теорії складності обчислень та підкреслена потреба у побудові алгоритмів з поліноміальним часом роботи.

Водночас, деякі задачі робастної оптимізації є занадто складними для побудови оптимального розв'язку за поліноміальний час. Одним зі способів спрощення таких задач є перехід до задач пошуку наближеного розв'язку, що дозволяє подальшу побудову алгоритмів апроксимації. Такі алгоритми знаходять розв'язок, що несуттєво відрізняється від оптимального. Вони відрізняються як від точних, так і від евристичних алгоритмів, проте мають спільні елементи з обома класами. Синтезом результатів цієї частини огляду є наукові роботи присвячені побудові схем апроксимації з поліноміальним часом роботи.

Проведений аналіз дозволяє виділити задачі робастної оптимізації, для яких

ще не розроблено схеми апроксимації з поліноміальним часом роботи. Огляд ідентифікує такі задачі в галузях побудови оптимальних маршрутів, розкладів та бюджетних планів, та окреслює перспективні підходи до їхнього пошуку.

Завершується огляд науковими роботами щодо організації обчислювальних експериментів та інтерпретації отриманих результатів.

1.1. Фундаментальні засади оптимізації

Перший підрозділ огляду визначає місце оптимізації у сучасній прикладній математиці, порівнює та протиставляє неперервну та дискретну оптимізацію, а також підкреслює важливість розгляду робастних постановок оптимізаційних задач в умовах невизначеності.

1.1.1. Місце оптимізації у прикладній математиці. Не буде перебільшенням стверджувати, що будь-яка задача прикладної математики має справу з певною спрощеною моделлю об'єктивної фізичної реальності, яка постійно уточнюється шляхом порівняння теоретичних та експериментальних результатів. Такий підхід поділяють як іноземні, так і українські дослідники. Зокрема, фундаментальна праця Vox та співавторів “Statistics for experimenters” [10] заклала основи статистичного підходу до моделювання, а його афоризм “всі моделі хибні, але деякі з них корисні” став крилатим у науковій спільноті, підкреслюючи наближений характер будь-якої математичної абстракції.

1.1.1.1. Моделювання та коректність задач. В українській математичній школі питання загальної теорії оптимальних алгоритмів та їх застосування до реальних моделей детально висвітлено у монографії академіків І. В. Сергієнка, В. К. Задіраки та О. М. Литвина [11]. Автори наголошують на важливості врахування похибок вхідних даних та обчислень при побудові алгоритмів, що є критичним для практичної реалізації чисельних методів на ЕОМ.

Не менш важливо розуміти, що розв'язки, отримані з математичних моде-

лей, використовуються для прийняття рішень та керування складними системами. Значущість питань існування та єдиності розв'язків підкреслюють ґрунтовні наукові дослідження. Класична монографія Jacques-Louis Lions [12] з теорії оптимального керування системами, що описуються диференціальними рівняннями з частинними похідними, розглядає фундаментальні питання коректності постановок задач керування та існування оптимальних розв'язків у функціональних просторах.

У вітчизняній літературі питання моделювання складних динамічних процесів, зокрема задач динаміки з розподіленими параметрами, глибоко досліджено у праці В. В. Скопечького, В. А. Стояна та Ю. Г. Кривоноса [13]. Автори приділяють значну увагу оберненим задачам, які часто виникають при ідентифікації параметрів фізичних моделей, де оптимізаційні методи є ключовим інструментом знаходження розв'язку.

1.1.1.2. Ітераційні методи та їх збіжність. Питання збіжності методів розв'язування оптимізаційних задач, зокрема варіаційних нерівностей, активно розробляються школою В. В. Семенова. У роботах з Ю. В. Маліцьким та О. С. Харьковим [14–16] досліджено умови сильної збіжності проксимальних та екстраградієнтних методів. Розроблені авторами підходи дозволяють будувати ефективні алгоритми для задач, де класичні градієнтні методи можуть не збігатися або працювати повільно.

Крім того, для забезпечення обчислювальної стійкості розв'язків широко застосовуються спеціалізовані ітераційні процедури. Дослідження збіжності методів типу проекції та скорочення, а також методів для нелінійних задач комплементарності представлені у роботах Kanzow [17], Pang і Gabriel [18]. Подальший розвиток ці методи отримали у працях She-feng і Pu-sheng [19] та Tseng [20], які запропонували модифікації з покращеними характеристиками збіжності.

Окремий інтерес становлять методи, що поєднують оптимізацію з теорією нейродинамічних систем. Питання стійкості процесів навчання у таких систе-

мах, зокрема в моделях з післядією, досліджували А. В. Шатирко та співавтори [21]. Для задач великої розмірності з обмеженнями ефективним виявився метод ADMM з прискоренням Нестерова, модифікації якого розробляли В. А. Григоренко та ін. [22], а також методи L-BFGS-B для задач з простими обмеженнями [23].

1.1.2. Неперервна та дискретна оптимізація. Фундаментальна відмінність між неперервною та комбінаторною оптимізацією полягає у природі множини допустимих розв'язків.

1.1.2.1. Опукла та неперервна оптимізація. У неперервній оптимізації ключовими є поняття опуклості та диференційовності. Класичним підручником, що систематизує знання з опуклої оптимізації, є книга Boyd і Vandenberghe [24]. Вона стала настільною книгою для дослідників, які працюють з методами, що гарантують знаходження глобального екстремуму. Важливим розширенням цього апарату є напіввизначене програмування (SDP), огляд якого наведено у роботі Boyd і Vandenberghe [25], що відкриває шляхи до релаксації складних дискретних задач.

Вагомий внесок у розвиток теорії опуклого аналізу та теорії операторів у гільбертових просторах зробили Bauschke і Combettes [26], чия монографія є важливим джерелом для розуміння збіжності ітераційних алгоритмів нерухомої точки. Окрім того, С. І. Ляшко та В. В. Семенов [27] запропонували економічні модифікації методів типу Корпелевич для монотонних задач рівноваги, а Ю. Маліцький [28, 29] розвинув методи проектованого відбитого градієнта та алгоритми золотого перетину для варіаційних нерівностей.

Сучасні дослідження у неперервній оптимізації виходять за рамки евклідових просторів. Наприклад, робота Chen та співавторів [30] розглядає прискорення оптимізації у просторі ймовірнісних мір, що є актуальним для задач машинного навчання та транспорту мас. Також розробляються ефективні методи проектування на ℓ_1 -кулю для задач навчання у високих розмірностях [31]. Окрему

увагу слід приділити узагальненому методу еліпсоїдів, уніфіковане представлення якого надали П. Стецюк та співавтори [32].

1.1.2.2. Дискретна оптимізація на графах. На противагу цьому, комбінаторна оптимізація оперує на дискретних структурах, таких як графи. Багато задач тут формулюються у термінах цілочисельного програмування або аналізу ефективності рішень, як-от методи DEA (data envelopment analysis). Проблеми застосування DEA до неточних даних розглядали Lee і Zhu [33]. Також важливим є розгляд багатокритеріальних задач теорії прийняття рішень, детальний огляд яких надає Sayin [34].

Базові поняття теорії графів, необхідні для формулювання задач на мережах, викладено у класичних підручниках Bollobas [35] та Diestel [36, 37]. Ці праці описують властивості зв'язності, парувальності та екстремальні властивості графів, які є фундаментом для побудови алгоритмів. Фундаментальні питання перелічувальної комбінаторики, що визначають потужність простору пошуку та складність перебору, висвітлено у праці Stanley [38].

Українська школа робить значний внесок і в дискретну оптимізацію. Наприклад, підхід до розв'язання дискретних задач векторної оптимізації на комбінаторних множинах перестановок запропоновано у роботі Н. В. Семенової, Л. М. Колечкіної та А. Н. Нагірної [39]. Цей напрямок дозволяє ефективно вирішувати задачі впорядкування та розміщення без повного перебору варіантів.

1.1.3. Оптимізація в умовах невизначеності. Класичні методи оптимізації часто припускають точність вхідних даних. Проте на практиці дані можуть бути неточними або зашумленими. Це призвело до виникнення напрямку робастної оптимізації.

1.1.3.1. Теоретичні основи робастності. Основоположною працею у цій галузі є книга Ben-Tal і Nemirovski [40], а також їхня пізніша робота спільно з El Ghaoui [41]. Вони розробили методологію розв'язання задач, де параме-

три належать певним множинам невизначеності (еліпсоїдальним, поліедральним тощо), гарантуючи допустимість розв'язку для будь-якої реалізації невизначеності. Важливим результатом є дослідження Bertsimas і Sim [42], які запропонували підхід “price of robustness”. Цей метод дозволяє контролювати рівень консерватизму розв'язку, балансуючи між надійністю та вартістю цільової функції.

Сучасні підходи інтегрують методи машинного навчання в оптимізацію. Так, Goerigk і Kurtz [43] розглядають робастну оптимізацію, керовану даними, з використанням глибоких нейронних мереж для побудови множин невизначеності. Оглядова стаття Aissi та співавторів [44] систематизує підходи до мінімакських версій задач комбінаторної оптимізації, які є природним формулюванням робастних задач на графах. Зокрема, мінімакські підходи часто застосовуються для задач вибору елементів з інтервальними даними, як це описано у Kasperski [45].

1.1.3.2. Конфліктно-керовані системи та стійкість. Українські вчені також активно працюють з задачами в умовах невизначеності та конфлікту. Монографія А. А. Чикрія [46] присвячена конфліктно-керованим процесам, що є узагальненням ігрових задач динаміки, де невизначеність моделюється як дія супротивника. У контексті диференціальних ігор, які є близькими до робастного керування, варто відзначити класичні роботи Isaacs [47] та Friedman [48]. Питання стійкості систем, що є ключовим для робастності, досліджуються у роботах В. В. Пічура [49, 50], де розглядається практична стійкість диференціальних включень та технологічних процесів.

Статистичні аспекти моделювання невизначеності, зокрема для економічних систем, розглядаються у роботах Golodnikov та співавторів [51]. Також важливим є коректний вибір ймовірнісних розподілів для моделювання стохастичних даних, наприклад, розподілу фон Мізеса, детальний аналіз якого наведено у роботі Benlakhdar та ін. [52].

1.1.3.3. Застосування у фінансах та аналізі часових рядів. Окремий пласт прикладних задач з високим рівнем невизначеності становлять задачі фінансової математики. Тут робастність інтерпретується через призму адекватності моделей реальним динамічним процесам.

Дослідники розглядають моделі стохастичної волатильності та їх вплив на ціноутворення активів, а також застосування мультифрактального броунівського руху для моделювання цін акцій [53]. Для прогнозування часових рядів широко використовуються моделі ARIMA, ефективність яких досліджено Jarrett [54] та Mondal [55].

Важливим аспектом сучасних досліджень є інтеграція гетерогенних даних. Наприклад, Nuij та ін. [56] розробляють фреймворки для включення новинного фону в торгові стратегії. Технічний аналіз лімітних книг заявок з використанням згорткових нейромереж для прогнозування цін розглянуто у роботі Tsantekidis та ін. [57].

Аналіз ефективності торгових стратегій на різних ринках наведено у роботах Vijn [58] та Fifield [59]. Також Wu та ін. [60] пропонують адаптивні стратегії на основі глибокого навчання з підкріпленням.

1.1.3.4. Робастність мережевих структур. У контексті мережевих систем робастність часто розглядається як здатність мережі зберігати зв'язність та працездатність при атаках або випадкових відмовах вузлів. Li та ін. [61] пропонують стратегії оцінки та підсилення мереж проти каскадних відмов. Dashti та ін. [62] досліджують стійкі самоорганізовані мережі в мультиагентних системах. Для специфічних застосувань, таких як Інтернет речей, аналіз робастності з використанням регулярних графів проведено у роботі Dhuli та ін. [63].

1.2. Комбінаторна оптимізація та складність алгоритмів

Цей розділ присвячений методам розв'язання задач комбінаторної оптимізації, класифікації алгоритмів за їхньою точністю та складністю, а також аналізу

структур, на яких ці задачі визначаються.

1.2.1. Теоретико-графові основи. Оскільки багато задач комбінаторної оптимізації формуються мовою графів, властивості топології мережі є визначальними для побудови та аналізу алгоритмів.

1.2.1.1. Випадкові та складні мережі. Класичні роботи Erdős і Rényi [64] заклали основи теорії випадкових графів, показавши умови виникнення гігантської компоненти зв'язності. Albert і Barabási [65] розвинули статистичну механіку складних мереж, запропонувавши моделі безмасштабних мереж (англ. *scale-free networks*), що більш точно описують реальні комунікаційні та соціальні структури.

1.2.1.2. Симетричні графічні структури. Особливу увагу в літературі приділено регулярним та симетричним структурам, які часто виступають як екстремальні випадки або бажані топології для проектування робастних мереж. Voesch і Tindell [66] досліджували зв'язність циркулянтних графів, що є важливим для надійності мереж. Gatt [67] та Лісковець [68] незалежно займалися переліком таких графів, що дозволяє оцінити різноманітність доступних топологій. Huang та ін. [69] показали оптимальність циркулянтних графів як топологій з низькою затримкою передачі даних.

Крім того, досліджуються властивості сильно регулярних графів. Brouwer і Mesner [70], а також Cioabă [71] вивчали зв'язок між власними числами та реберною зв'язністю регулярних графів. Ganesan [72] аналізував умови існування зв'язних регулярних графів, а Wu та ін. [73] досліджували робастність регулярних ґратчастих структур на основі поняття природної зв'язності.

1.2.1.3. Зв'язність та динамічні процеси. З погляду вразливості мереж, критичними параметрами є реберна зв'язність та наявність паруваль. Harary [74] надав класичні результати щодо максимальної зв'язності графів. Nagamochi і Ibaraki [75] систематизували алгоритми обчислення зв'язності. Су-

часні дослідження розширюють ці результати: Henning і Yeo [76] вивчають парування у графах із заданим максимальним степенем, а Suil і West [77] — у регулярних графах. Škovičera і Varša [78] довели NP -повноту задачі про індекс досконалого парування для кубічних графів.

Також розглядаються динамічні процеси на графах. Šubelj [79] запропонував методи обчислення добре збалансованих кістякових дерев, що важливо для розподілених алгоритмів. Yoneda та ін. [80] встановили нижні оцінки зв'язності мережі, що гарантують синфазну синхронізацію осциляторів.

1.2.1.4. Алгоритми для потокових задач. Класична книга Cormen та співавторів [81] описує базові алгоритми на графах. Особливе місце займають поточкові задачі. Варто відзначити піонерські роботи Ford і Fulkerson [82], які сформулювали теорему про максимальний потік та мінімальний розріз. Edmonds і Karp [83] покращили ці методи, забезпечивши поліноміальну складність. Dinitz [84] запропонував концепцію шаруватих мереж. Tardos [85] розробили перший сильно поліноміальний алгоритм для циркуляції мінімальної вартості. Більш сучасні підходи проптовхування перед-потіку Goldberg і Tarjan [86] та недавній прорив Chen та ін. [87], які досягли майже лінійного часу виконання для задач про максимальний потік, демонструють постійний прогрес у цій галузі.

Специфічні точні методи для задачі комівояжера розвивалися з середини XX століття, починаючи з робіт Croes [88]. Сучасний аналіз дрібнозернистої складності варіантів TSP наведено у роботі de Berg та ін. [89].

1.2.2. Класи складності та NP -повнота. Центральним поняттям у теорії алгоритмів є поняття обчислювальної складності. Задачі комбінаторної оптимізації часто належать до класу NP -складних задач.

Канонічним джерелом з теорії NP -повноти є книга авторів Garey та Johnson “Computers and Intractability” [90]. Вона містить визначення класів P , NP , NP -complete, а також розлогий каталог відомих складних задач. Johnson, Papadi-

mitriou і Steiglitz [91] також надають широкий огляд алгоритмів та локального пошуку у комбінаторній оптимізації.

Фундаментальну роль у становленні теорії відіграла стаття Karp [92], який продемонстрував поліноміальні зведення між 21 комбінаторною задачею, довівши їхню NP -повноту. Це, зокрема, стосується задачі комівояжера та задачі про рюкзаки. Останніми роками набуває популярності так звана тонка складність, яка дозволяє розрізняти складність задач всередині поліноміального класу або доводити нижні оцінки, базуючись на сильних гіпотезах, про що детально пише Williams [93, 94].

1.2.3. Точні, наближені та евристичні алгоритми. Існує три основні підходи до розв’язання складних комбінаторних задач: точні методи, алгоритми апроксимації та евристики.

1.2.3.1. Точні алгоритми та структури даних. Ці методи гарантують знаходження глобального оптимуму, але часто мають експоненційну складність у найгіршому випадку. Монографія Korte і Vygen [95] дає вичерпний огляд теорії та точних алгоритмів комбінаторної оптимізації, включаючи методи лінійного програмування та цілочисельної оптимізації. Для покращення швидкодії методів динамічного програмування та роботи з графами критично важливими є ефективні структури даних, такі як купи Фібоначчі, запропоновані Fredman і Tarjan [96], та техніки прискорення динамічного програмування, розроблені Eppstein та ін. [97].

1.2.3.2. Алгоритми апроксимації. Оскільки пошук точних розв’язків NP -складних задач для великих розмірностей є непрактичним, важливу роль відіграють алгоритми, що знаходять розв’язок за поліноміальний час із гарантованою похибкою. Знаковим результатом є робота Christofides [98], який запропонував алгоритм для метричної задачі TSP з гарантованим коефіцієнтом апроксимації $3/2$. Цей результат залишається одним із найкращих для загаль-

ного випадку метричної TSP протягом десятиліть. Для геометричних постановок задачі Arora [99] розробив поліноміальну схему апроксимації, що дозволяє знаходити розв'язок із довільною точністю $(1 + \varepsilon)$.

1.2.3.3. Евристичні алгоритми. Для практичних застосувань часто використовуються методи, що не дають теоретичних гарантій, але демонструють високу ефективність “у середньому”. Сюди відносяться генетичні алгоритми, ідеологія яких була започаткована Holland [100]. Іншим потужним методом є імітація відпалу, застосована до комбінаторних задач Kirkpatrick [101]. Окремий клас становлять алгоритми мурашиних колоній, детально описані у книзі Dorigo і Stützle [102]. Л. Ф. Гуляницький досліджував питання диверсифікації пошуку в таких алгоритмах [103], що покращує їхню здатність виходити з локальних екстремумів.

1.2.4. Кластеризація та попередні перетворення. Для задач маршрутизації великої розмірності часто застосовують декомпозицію через кластеризацію. Огляд алгоритмів ієрархічної кластеризації наведено у роботі Ran та ін. [104]. Також важливими є підходи до кластеризації мереж, що еволюціонують, описані Hartmann та ін. [105], які дозволяють враховувати динаміку зміни структури графу.

У контексті задачі комівояжера, Guttmann-Beck та ін. [106] дослідили апроксимаційні алгоритми для кластеризованої TSP, де відвідування міст згруповано. Вплив врахування внутрішньокластерних та міжкластерних відстаней на ефективність маршрутизації розглянуто у роботах Sheng та ін. [107], які фокусуються на міжкластерних зв'язках, та Ding та ін. [108], які пропонують дво-рівневий генетичний алгоритм.

1.3. Відкриті питання робастної апроксимації

Незважаючи на значний прогрес, існує клас робастних задач комбінаторної оптимізації, для яких питання ефективної апроксимації залишається відкритим. Зокрема, це стосується задач, що можуть бути зведені до обчислення згорток або їх узагальнень.

1.3.1. $(\min, +)$ згортки. У роботі Н. М. Скибицького та К. І. Денисова [1] показано обчислювальну еквівалентність одновимірної задачі комівояже-ра з квотами та $(\min, +)$ -згортки. Це відкриває нові можливості для перенесення результатів складності між цими областями. Також досліджено робастні постановки задач фінансової математики, зокрема задачу про оптимальний час купівлі та продажу акцій в умовах невизначеності [2], що зводиться до пошуку шляхів на специфічних графах.

Ці задачі вимагають розробки нових схем апроксимації, оскільки існуючі точні алгоритми для згорток та споріднених задач про рюкзак мають обмеження. Як показано у роботах Bringmann і Cassis [109, 110], а також Cygan та ін. [111], існують бар'єри складності, що унеможливають створення суттєво швидших точних алгоритмів. Дослідження у цій галузі також тісно пов'язані з комбінаторними задачами на намистах та згортками, які вивчали Bremner та ін. [112]. Побудова ефективних наближених алгоритмів для таких структур є актуальним напрямком досліджень.

1.3.2. Маршрутизація БПЛА.

1.3.2.1. Загальні підходи та специфіка БПЛА. Важливим класом прикладних задач є маршрутизація безпілотних літальних апаратів (БПЛА). Це динамічна галузь, де класичні моделі VRP (англ. *vehicle routing problem*) адаптуються до специфічних обмежень дронів, таких як заряд батареї та вантажопідіймальність. Загальний огляд методів маршрутизації транспортних засобів наведено у статті Laporte [113]. Однак специфіка БПЛА вимагає враху-

вання нових факторів. Зокрема, Arsie і Frazzoli [114] досліджували ефективну маршрутизацію декількох транспортних засобів без прямої комунікації, покладаючись на розподілені протоколи. Murayama [115] аналізував компроміс між робастністю зв'язності мережі роботів та їх зоною покриття. Також важливим є врахування керованості динамічних мереж зі змінною топологією, як у системах “лідер-послідовник” [116], та моделювання намірів агентів у робототехніці [117].

1.3.2.2. Огляд національних та закордонних досліджень. Українські дослідники Ю. Ф. Кучеренко та ін. [118] аналізують досвід застосування груп БПЛА у мережецентричних операціях, акцентуючи на координації дій. В. А. Лупандін та ін. [119] розглядають тенденції створення груп БПЛА та перспективи їх розвитку. Л. Ф. Гуляницький разом із колегами [120] досліджує конкретні алгоритмічні аспекти задачі маршрутизації дронів, а В. П. Горбулін та ін. [121] пропонують методи оптимізації маршрутів команд БПЛА за наявності альтернативних та динамічних депо.

Серед закордонних досліджень варто виділити роботу Savuran і Karakaya [122] щодо оптимізації маршрутів БПЛА, що запускаються з рухомого носія. Сучасні логістичні підходи включають інтеграцію вантажівок та дронів для операцій “останньої милі”, як це описано у роботі Long та ін. [123]. Також набувають актуальності “зелені” маршрути: Куо та ін. [124] застосовують генетичні алгоритми (NSGA-II) для мінімізації викидів вуглецю при доставці дронами.

1.4. Методологія проведення обчислювальних експериментів

1.4.1. Тестові набори даних. Для верифікації теоретичних оцінок ефективності алгоритмів необхідне проведення чисельного моделювання. Методологія експериментального аналізу алгоритмів ґрунтується на принципах, викладених у класичних працях з алгоритмів та структур даних. Важливим аспектом є використання стандартизованих наборів даних для забезпечення відтворюваності результатів. У контексті потокових задач, фундаментальна монографія

Ahuja, Magnanti і Orlin [125] надає не лише теоретичний базис, але й приклади тестування алгоритмів на мережах. Для задач комівояжера та їх варіацій стандартом де-факто є використання бібліотек типу TSPLIB, що дозволяє порівнювати результати з відомими бенчмарками, як це робиться у роботах з алгоритмів апроксимації (наприклад, [106, 108]).

1.4.2. Метрики робастності та ефективності. Оцінка робастності мереж та алгоритмів вимагає специфічних метрик, відмінних від простого часу виконання. Abbas та ін. [126] пропонують методи покращення зв'язності мереж за допомогою довірених вузлів, що може бути використано як модель для генерації тестових графів з різним рівнем стійкості. Wandelt та ін. [127] розглядають оцінку робастності транспортних мереж через використання спільнот, що також дає інструментарій для створення реалістичних тестових сценаріїв, наближених до реальних інфраструктурних мереж. Експерименти повинні оцінювати не лише час виконання, але й якість отриманих розв'язків, використовуючи метрики на кшталт price of robustness [42], на великих вибірках даних.

1.5. Висновки до розділу 1

Позитивними рисами існуючих рішень є:

- Висока гнучкість моделювання — розроблені методи здатні працювати з різними типами множин невизначеності (еліпсоїдальні, поліедральні, сценарні) та різною топологією мереж (циркулянтні, випадкові графи), що дозволяє адаптувати їх до широкого класу задач.
- Математична обґрунтованість — більшість підходів супроводжуються строгими теоретичними результатами, що включають доведення повноти (класи P та NP), оцінки коефіцієнтів апроксимації (схеми PTAS) та аналіз збіжності ітераційних процедур.
- Прикладна спрямованість — алгоритми адаптовані до конкретних завдань реального світу: маршрутизація дронів з урахуванням заряду бата-

реї, оптимізація фінансових портфелів в умовах стохастичної волатильності, забезпечення стійкості комунікаційних мереж.

- Алгоритмічне забезпечення — наявність стандартизованих бенчмарків (наприклад, TSPLIB) та розвинених бібліотек для лінійного програмування полегшує порівняння та впровадження методів.

Водночас виявлені і суттєві обмеження:

- Обчислювальна складність — багато ключових задач (зокрема, варіанти TSP та рюкзака) є NP -складними, що унеможливорює використання точних методів для задач великої розмірності у реальному часі.
- Консерватизм розв'язків — класичні мінімаксні підходи до робастності часто призводять до надмірно обережних рішень, що характеризується високою ціною робастності.
- Залежність від вхідних припущень — ефективність методів критично залежить від коректності вибору множини невизначеності; порушення цих припущень може суттєво знизити якість розв'язку.
- Проблеми апроксимації — для певних класів задач, таких як робастні згортки, побудова ефективних схем апроксимації з поліноміальним часом роботи залишається відкритим питанням через існуючі бар'єри складності.

Підсумовуючи, хоча галузь робастної комбінаторної оптимізації є розвинутою та багатою на підходи, універсального методу, який би одночасно поєднував поліноміальну складність, гарантовану точність та низький рівень консерватизму, наразі не існує. Це залишає простір для подальших досліджень, результати яких, зокрема побудова нових апроксимаційних схем та експериментальна перевірка їх ефективності, будуть представлені у наступних розділах роботи.

РОЗДІЛ 2

ФУНДАМЕНТАЛЬНА ТЕОРІЯ АЛГОРИТМІВ

2.1. Вступ

Сфера теоретичної інформатики зазнала суттєвих змін зумовлених глобальними технологічними трендами та локальними українськими викликами останніх років. Наразі фундаментальні дослідження вже не сприймаються виключно як абстрактні а розглядаються крізь призму їхнього застосування у галузі національної безпеки, захисту критичної інфраструктури, та кібербезпеки.

Ефективні алгоритми пов'язані зі зв'язність графів, криптографічними примітивами, структурними властивостями таких об'єктів як дерева та рядки стають інструментами подвійного призначення. Водночас в Україні зростає значення теорії прийняття рішень в умовах невизначеності та розширення теоретичних основ алгоритмів апроксимації.

В Україні дослідженнями у цьому напрямку займаються передусім Київський національний університет імені Тараса Шевченка, Інститут кібернетики імені В.М. Глушкова НАН України та Інститут математики НАН України. Саме створені в межах цих закладів наукові школи відіграють провідну роль в розробці алгоритмів в умовах невизначеності та строгих бюджетних обмежень.

В цьому розділі представлено нові теоретичні результати у п'яти галузях теорії алгоритмів та дискретної математики: властивостях операції побітового додавання, рядкових алгоритмах, алгоритмах на деревах, теорії тонкої обчислювальної складності та реберної зв'язності регулярних графів.

Отримані результати є підґрунтям для дослідження робастних версій розглянутих задач та побудови алгоритмів апроксимації для них у розділі 3 та розділі 4 дисертації.

2.1.1. Бітові операції. Одним з класичних будівельних блоків сучасної інформатики є операція побітового виключного або, яка позначається символом $\oplus$ та читається як *ксор* англійською мовою. Її прикладне значення складно переоцінити, адже її застосування охоплюють криптографічні примітиви, коди виправлення помилок, методи генерації псевдовипадкових чисел, але не обмежуються переліченими галузями.

На противагу класичній арифметичній операції додавання $+$, ксор виконується порозрядно без переносів у наступний розряд. Приклад такого додавання зображено на рис. 2.1.

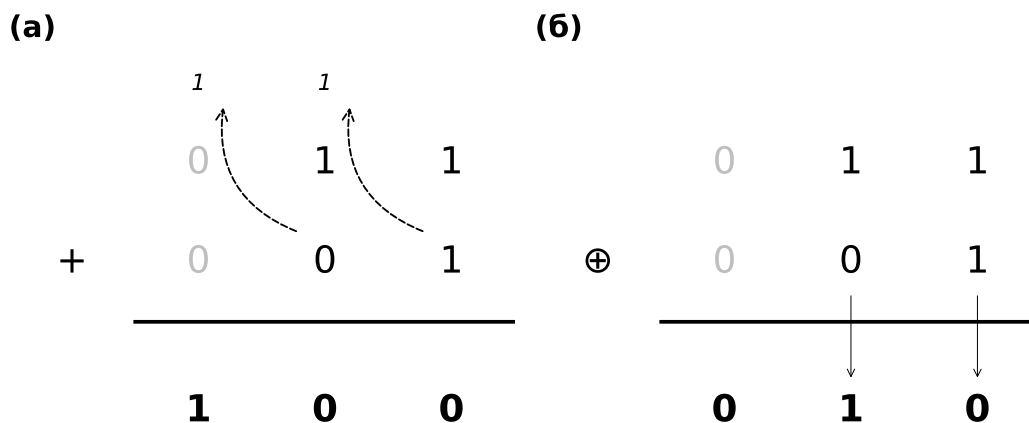


Рис. 2.1: Порівняння результатів додавання (а) та побітового додавання (б).

Така особливість породжує багато унікальних алгебраїчних властивостей. Зокрема, $(\mathbb{N}_0, \oplus)$ можна розглядати як комутативну (абелеву) групу ізоморфну прямій сумі зліченної кількості $\mathbb{Z}_2$. Така структура часто породжує складні закономірності попри свою простоту на рівні окремих цифр (бітів).

В галузі адитивної комбінаторики вивчаються множини вигляду

$$A + B = \{a + b \mid a \in A, b \in B\}, \quad (2.1.1)$$

де $A, B \subset \mathbb{N}_0$, або $A, B \subset \mathbb{Z}_p$ для простого p та було отримано багато класичних результатів, як-то

Теорема 1 (Коші—Девенпорта). Для довільного простого p та непорожніх $A, B \subset \mathbb{Z}_p$ виконується нерівність

$$|A + B| \geq \min(p, |A| + |B| - 1). \quad (2.1.2)$$

Більше подібних результатів можна знайти у відомій книзі Стенлі [38]. Такий результат спонукає дослідження властивостей поелементної суми множин в інших групах, зокрема у вже згаданих $(\mathbb{N}_0, \oplus)$. Вводиться поняття:

Означення 1. Побітовою поелементною сумою множин $A, B \subset \mathbb{N}_0$ будемо називати множину

$$A \oplus B = \{a \oplus b | a \in A, b \in B\}, \quad (2.1.3)$$

що прямо відповідає рівнянню (2.1.1).

Для множин A, B загального вигляду питання пошуку структури у множині $A \oplus B$ та оцінки її розміру є досить складним, тож відповідний підрозділ на- томість зосереджений на множинах спеціального вигляду. А саме, розглянуто A, B що є півінтервалами $[a, c)$ та $[b, d)$ натуральних чисел. Приклад взаємодії інтервалів під дією ксору зображено на рис. 2.2.

Центральними результатами відповідного підрозділу є аналог теореми 1:

Теорема 2. Для непорожніх інтервальних множин $A, B \subset \mathbb{N}_0$ виконується співвідношення

$$|A \oplus B| \leq C(|A| + |B|), \quad (2.1.4)$$

де $C > 0$ — певна дійсна константа.

та побудова алгоритму, що дозволяє за полі-логарифмічний час знайти множину $A \oplus B$ для заданих інтервалів A, B .

Зауваження 1. На відміну від теореми 1, оцінка зверху а не знизу, оскільки оцінка знизу в рівнянні (2.1.4) є тривіальною та випливає з ін'єктивності ксору: $|\{a\} \oplus B| = |B|$ для довільних $a \in \mathbb{N}_0$ та $B \subset \mathbb{N}_0$.

Зауваження 2. Така лінійна верхня оцінка суттєво краща за наївну квадратичну оцінку $|A \oplus B| \leq |A| \cdot |B|$.

		$Y \in [2, 6)$			
		2	3	4	5
$X \in [4, 8)$	4	6	7	0	1
	5	7	6	1	0
	6	4	5	2	3
	7	5	4	3	2

Рис. 2.2: Матриця взаємодії відрізків $[2, 6)$ та $[4, 8)$ при операції ксор.

Ці результати були отримані автором одноосібно і апробовано на конференції Шевченківська весна 2025.

2.1.2. Рядкові алгоритми. У галузі рядкових алгоритмів важливе місце посідають задачі лексикографічного упорядкування та декомпозиції рядків. До числа прикладних застосувань таких алгоритмів входять пошуки повторів у геномах в біоінформатиці, ефективне стиснення даних та побудова вдалих індексів для швидкого пошуку інформації в інтернеті.

Як правило, задачі із рядками потребують створення масивних структур даних, як-то суфіксні масиви, масив найдовших спільних префіксів, або навіть суфіксні автомати. Більш детальний опис суфіксних структур даних та їхніх

застосувань міститься у роботах Манбера [128] та Крочемора [129].

У багатьох прикладних задачах одним із перших кроків є певна декомпозиція або перетворення тексту на зручніший для подальшої обробки. Одним з найвідоміших подібних кроків є перетворення Барроуза—Віллера (BWT):

Означення 2. *Перетворенням Барроуза—Віллера* рядка s називається рядок із символів останнього стовпчика матриці Барроуза—Віллера рядка s .

Означення 3. *Матрицею Барроуза—Віллера* рядка s називається матриця, рядками якої є циклічні зсуви рядка s відсортовані лексикографічно.

Ефективний алгоритм побудови BWT, як правило, використовує суфіксний масив рядка s .

Означення 4. *Суфіксним масивом* рядка s називається масив індексів, які є позиціями початку суфіксів s у лексикографічно відсортованому порядку.

Для ототожнення циклічних зсувів та суфіксів до рядка s достатньо дописати певний символ що в ньому не зустрічається та вважати його більшим за усі справжні символи s . Оскільки існують лінійні за часом алгоритми побудови суфіксного масиву, то й BWT теж можна побудувати за лінійний час $O(n)$, де n позначає довжину рядка s , тобто кількість символів.

BWT було б не дуже корисним якби не існувало способу його обернути. Для відновлення рядка s за його BWT зазвичай використовують декомпозицію (факторизацію) Ліндона. Спершу дається означення слова Ліндона:

Означення 5. Непорожній рядок w називається словом Ліндона, якщо він є строго меншим у лексикографічному порядку за всі свої власні суфікси.

Приклад 1. Словами Ліндона є, зокрема, a , b , ab , aab , abb . Натомість, aba (суфікс $a < aba$), ba не є словами Ліндона.

Означення 6. Декомпозицією Ліндона рядка s називається його розбиття $s = w_1|w_2|\dots|w_k$, де усі w_i є словами Ліндона, а також $w_1 \geq w_2 \geq \dots \geq w_k$.

Ліндон з колегами довели що таке розбиття єдине, а згодом Дюваль [130] винайшов алгоритм який дозволяє знайти такий розклад за лінійний час $O(n)$.

Зауважимо, що тут і надалі під О-великим розуміється складність в найгіршому випадку, а не в середньому.

У відповідному підрозділі означається альтернативний спосіб декомпозиції рядка на слова Ліндона та запропоновано ефективний алгоритм побудови такої декомпозиції з використанням монотонного стеку.

Забігаючи наперед анонсовано, що розроблений алгоритм використовує шарувату архітектуру з кількох структур даних зображених на рис. 2.3.

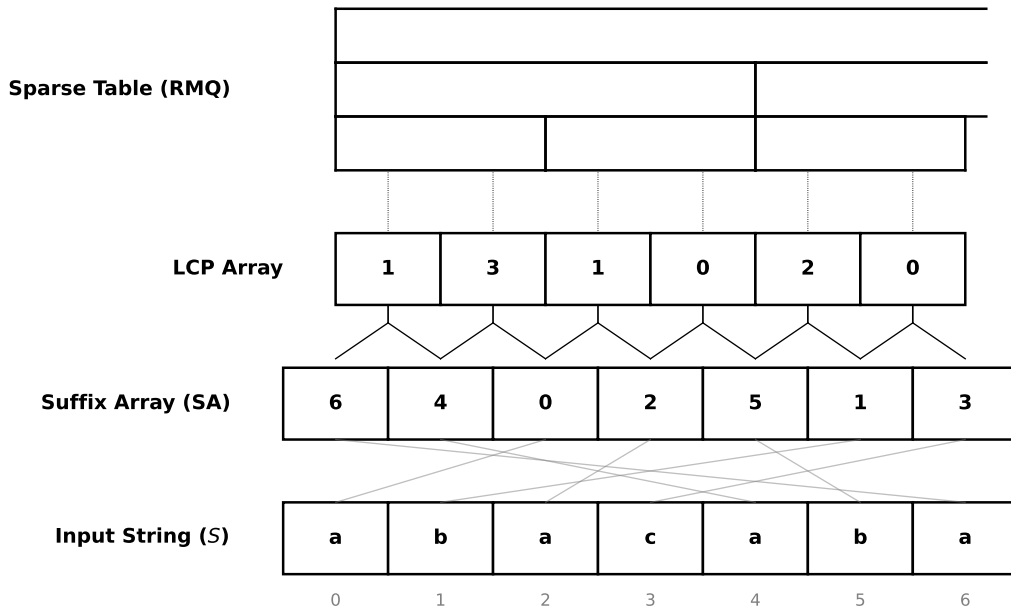


Рис. 2.3: Ілюстрація рівневої архітектури алгоритму.

Ці результати були отримані у співавторстві з Нішанком Сурешем, випускником Індійського технологічного інституту у Мадрасі. Результати лягли в основу однієї з задач міжнародного відкритого змагання з програмування.

2.1.3. Згортки послідовностей. Операція згортки є фундаментальним будівельним блоком не лише в дискретній оптимізації, але й в неперервній. Щоправда, розуміння згортки тут суттєво відрізняється. Так, у галузі математичного аналізу як правило означають $(+, \cdot)$ згортку:

Означення 7. Неперервною $(+, \cdot)$ згорткою двох функцій $f, g : \mathbb{R} \rightarrow \mathbb{R}$ називають функцію $h : \mathbb{R} \rightarrow \mathbb{R}$ визначену співвідношенням

$$h(z) = (f * g)(z) = \int_{-\infty}^{\infty} f(x)g(z - x)dx. \quad (2.1.5)$$

В дискретній математиці поняття $(+, \cdot)$ згортки теж існує і знаходить необмежені практичні застосування. Зрештою, самих лише згорткових штучних нейронних мереж достатньо аби увіковічити поняття згортки у колективній пам'яті людства. Наведемо означення для повноти:

Означення 8. Дискретною $(+, \cdot)$ згорткою двох послідовностей $f, g : \mathbb{Z} \rightarrow \mathbb{R}$ називають послідовність $h : \mathbb{Z} \rightarrow \mathbb{R}$ визначену співвідношенням

$$h[n] = (f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n - m]. \quad (2.1.6)$$

Такий запис має прояснити назву плюс-помножити згортки, адже операція додавання не зустрічалася явно у рівнянні (2.1.5).

Добре відомі алгоритми швидкого обчислення згортки скінченних послідовностей довжини n , більшість яких засновані на швидкому перетворенні Фур'є та мають лінеаритмічний час роботи $O(n \log n)$. Дисертація не зупиняється на внутрішніх механізмах перетворення Фур'є та способі його обчислення, адже вони не стали у пригоді при побудові відповідного алгоритму.

Натомість дається означення мінімум-плюс згортки:

Означення 9. $(\min, +)$ згорткою двох послідовностей $f, g : \mathbb{Z} \rightarrow \mathbb{R}$ називають послідовність $h : \mathbb{Z} \rightarrow \mathbb{R}$ визначену співвідношенням

$$h[n] = \min_{m=-\infty}^{\infty} (f[m] + g[n - m]). \quad (2.1.7)$$

Напротивагу швидкому перетворенню Фур'є для рівняння (2.1.6), для скінченної версії рівняння (2.1.7) поки що не знайдено суттєво суб-квадратного алгоритму. А саме, Циган та інші [111] висунули гіпотезу про мін-плюс згортку:

Гіпотеза 1. Для жодного $\varepsilon > 0$ не існує алгоритму який за час $O(n^{2-\varepsilon})$ обчислює $(\min, +)$ згортку довільних послідовностей довжини n .

Зауваження 3. *Гіпотеза допускає існування несуттєво суб-квадратних алгоритмів. Бремнер з колегами [112] запропонували алгоритм з обчислювальною складністю $O\left(\frac{n^2(\log \log n)^3}{\log^2 n}\right)$.*

Більше про відмінність між виразами $O(n^{2-\epsilon})$ та $O\left(\frac{n^2(\log \log n)^3}{\log^2 n}\right)$ можна прочитати у роботах з теорії тонкої обчислювальної складності, наприклад у Вільямса [93, 94]. В цій галузі встановлюють нижні оцінки складності алгоритмів з класу P , деякі з яких є точними.

Зауваження 4. *Також існує багато спеціальних випадків вхідних даних для яких $(\min, +)$ згортку можна обчислити за час $O(n \log n)$. Сюди відносяться як задачі з монотонними та опуклими послідовностями, так і псевдополіноміальні алгоритми час роботи яких залежить від вхідних значень а не лише їхнього розміру, тобто кількості бітів необхідних для запису.*

У підрозділі встановлено зв'язок задачі про мінімум-плюс згортку з окремим випадком задачі комівояжера з квотою. Отримані зведення достатньо швидкі і дозволяють використовувати результати теорії тонкої обчислювальної складності та переносити їх з однієї задачі на іншу.

Ці результати отримано у співавторстві з К.І. Денисовим, студентом факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка. Результати були апробовані на конференції ІТІА 2024 та опубліковані у 2-ому випуску Журналу обчислювальної та прикладної математики за 2024-ий рік.

2.1.4. Послідовності у деревах. Окрім бітових операцій, масивів та рядків в теорії алгоритмів зачасту можна зустріти графи. Найпростішими графами є дерева, тож більшість досліджень доцільно розпочинати саме з них. Зрештою, множина дерев міститься у множині графів, тож складність алгоритму для знаходження певного об'єкту у загальному графі не може бути меншою за складність для дерева. Водночас у математиці часто зустрічаються знакомі змінні послідовності. Дамо формальне

Означення 10. Послідовність $(a_n)_{n=1}^{\infty}$ називається *знакозмінною* якщо або

$$a_1, a_3, \dots > 0 \quad \wedge \quad a_2, a_4, \dots < 0, \quad (2.1.8)$$

або

$$a_1, a_3, \dots < 0 \quad \wedge \quad a_2, a_4, \dots > 0. \quad (2.1.9)$$

Вони тісно пов'язані з зигзагоподібними послідовностями. Дамо формальне

Означення 11. Послідовність $(a_n)_{n=1}^{\infty}$ називається *зигзагоподібною* якщо

$$a_1 > a_2 < a_3 > a_4 < \dots \quad \vee \quad a_1 < a_2 > a_3 < a_4 > \dots \quad (2.1.10)$$

Зв'язок між ними визначає

Лема 1. *Префіксні суми кожної знакозмінної послідовності є зигзагоподібною послідовністю, а різницеvim масивом кожної зигзагоподібної послідовності є знакозмінна послідовність.*

Для повноти наводяться означення:

Означення 12. Послідовність *префіксних сум* b послідовності a визначається співвідношеннями

$$b_n = \sum_{k=1}^n a_k. \quad (2.1.11)$$

Означення 13. *Різницеvim масив* d послідовності c визначається співвідношеннями

$$d_n = c_{n+1} - c_n. \quad (2.1.12)$$

Окрім пошуку зигзагоподібних послідовностей важливі прикладні задачі аналізу геномів та високомолекулярних сполук потребують вміти нумерувати або ж принаймні рахувати кількість таких послідовностей.

У масиві задача пошуку найдовшої знакозмінної підпослідовності розв'язується за лінійний час $O(n)$ методами динамічного програмування. Узагальнення цієї задачі на дерева є складнішим через нелінійну структуру.

У відповідному підрозділі послідовно розв'язано дві задачі підрахунку зигзагоподібних послідовностей у деревах з бінарними та довільними числовими

мітками. Побудовані алгоритми застосовують методи динамічного програмування на деревах, поняття найменшого спільного предка та метод злиття від меншого до більшого.

Також було розглянуто підходи, що ґрунтуються на центроїдній та heavy-light декомпозиціях введених Бендером [131] та Слейтором [132], проте отримані алгоритми виявилися громіздкими та менш ефективними.

Ці результати були отримані у співавторстві з Нішанком Сурешем, випускником Індійського технологічного інституту у Мадрасі та Реєм Шреяном, випускником Індійського технологічного інституту у Харагпурі. Результати лягли в основу однієї з задач міжнародного відкритого змагання з програмування.

2.1.5. Регулярні графи. Останній підрозділ теоретичної частини присвячений реберній зв'язності регулярних графів. Такі графи з однаковими степенями усіх вершин знаходять застосування у проєктуванні реальних мереж з фізичними обмеженнями, як-то енергомережі, дорожні мережі, та мережеві комутатори.

Нехай $G = (V, E)$ — простий неорієнтований граф, де V — множина вершин, а $E \subseteq V \times V$ — множина ребер. Порядок графа позначається як $n = |V|$, а кількість ребер як $m = |E|$.

Означення 14. Граф G називається *k-регулярним*, якщо ступінь кожної вершини $v \in V$ дорівнює k :

$$\forall v \in V : \deg v = k \quad (2.1.13)$$

Важливим підкласом регулярних графів є циркулянтні графи.

Означення 15. Для натуральних чисел $0 < s_1 < \dots < s_l \leq n/2$ граф $C_n^{s_1, s_2, \dots, s_l} = (\{0, 1, \dots, n-1\}, \{(u, v) | \exists i : v = u + s_i \pmod{n}\})$ називається *циркулянтним графом* або просто *циркулянтною*.

Їхня структура тісно пов'язана з групою $(\mathbb{Z}_n, +)$, де n позначає кількість вершин у графі.

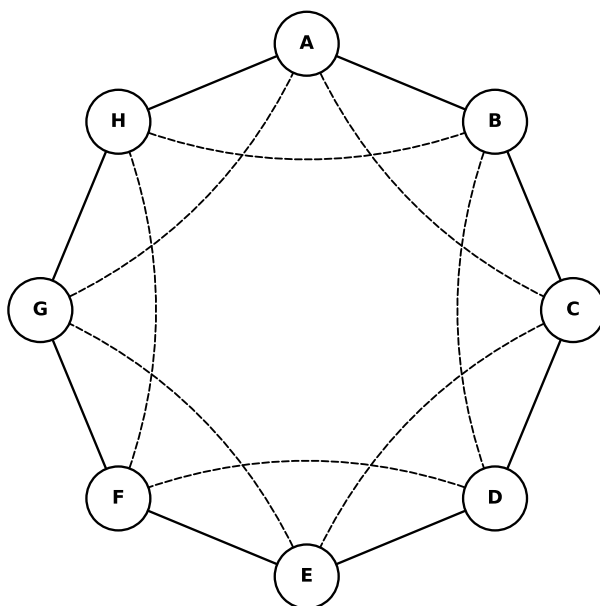


Рис. 2.4: Приклад циркулянти з парними n і k .

Для багатьох застосувань важливою характеристикою графа є його реберна зв'язність — мінімальна кількість ребер які необхідно видалити з графа так, щоб він перестав бути зв'язним. Така підмножина ребер також називається (глобальним) мінімальним розрізом.

Основними класичними результатами тут є

Теорема 3. *Реберна зв'язність k -регулярного циркулянтного графа з кроками $s_1, \dots, s_l$ для яких $\gcd(s_1, \dots, s_l, n) = 1$ дорівнює k .*

Гіпотеза 2. *Довільна k -регулярна циркулянта або незв'язна або має реберну зв'язність k .*

Відповідний підрозділ натомість зосереджено на погано зв'язаних циркулянтах. У ньому запропоновано конструктивні способи пошуку таких мінімально зв'язних графів та обговорено питання їхнього підрахунку.

Ці результати були отримані у співавторстві з науковим керівником В.В. Семеновим, доктором фізико-математичних наук та професором Київського національного університету імені Тараса Шевченка. Результати були апробовані на конференції DSMSI 2025.

2.2. Побітова поелементна сума множин

Перш ніж переходити до формулювання та доведення основних результатів, введемо строгі математичні означення об'єктів нашого дослідження.

Означення 16. Нехай $\mathbb{N}_0$ — множина невід'ємних цілих чисел. Для $x, y \in \mathbb{N}_0$ операція *побітового додавання* $x \oplus y$ визначається наступним чином: якщо $x = \sum_{i=0}^{\infty} x_i 2^i$ та $y = \sum_{i=0}^{\infty} y_i 2^i$, де $x_i, y_i \in \{0, 1\}$, то

$$x \oplus y = \sum_{i=0}^{\infty} (x_i + y_i \pmod{2}) 2^i. \quad (2.2.1)$$

Ми розглядаємо розширення цієї операції на множини.

Означення 17. Для двох множин $X, Y \subseteq \mathbb{N}_0$ їх *побітовою сумою* називається множина:

$$X \oplus Y = \{x \oplus y \mid x \in X, y \in Y\}. \quad (2.2.2)$$

Основним об'єктом нашої уваги є відрізки послідовних цілих чисел. В теорії алгоритмів та обчислень такі відрізки часто представляють собою блоки пам'яті, вікна даних або часові інтервали.

Означення 18. Відрізком довжини k з початком у точці x називається множина $[x, x + k)$.

Гіпотеза 3. Для будь-якого натурального $k > 1$ та будь-яких невід'ємних цілих x, y :

$$|[x, x + k) \oplus [y, y + k)| \leq 4(k - 1). \quad (2.2.3)$$

Одержане доведення гіпотези $4(k - 1)$ є громіздким, тож натомість наведемо просте доведення дещо слабшої, але асимптотично еквівалентної оцінки.

Теорема 4. Для будь-якого натурального $k \geq 5$ та будь-яких невід'ємних цілих x, y :

$$|[x, x + k) \oplus [y, y + k)| \leq 5(k - 2). \quad (2.2.4)$$

Доведення Теорема 4. Доведемо, що $f(n) \leq 5(n - 2)$ для $n \geq 5$ методом математичної індукції.

База індукції: у роботі [6] було доведено, що функція f володіє властивістю періодичності, що дозволило знайти $f(5) = 12$ та $f(6) \leq 16$ перебором з відтинанням. Перевіримо нерівність для $n = 5$ та $n = 6$: для $n = 5$ формула дає $5 \cdot 3 = 15 \geq 12 = f(5)$, а для $n = 6$, формула дає $5 \cdot 4 = 20 \geq 16 \geq f(6)$.

Крок індукції: Припустимо, що твердження вірне для всіх $m < n$. Розглянемо n . У роботі [6] також було доведено, що $f(2k), f(2k + 1) \leq 2f(k + 1)$.

1. n парне, $n = 2k$.

Маємо $f(2k) \leq 2f(k + 1)$. Оскільки $k \geq 3$ (бо $n \geq 6$), то $k + 1 < 2k$, і можна застосувати припущення індукції до $f(k + 1)$.

$$f(k + 1) \leq 5((k + 1) - 2) = 5(k - 1). \quad (2.2.5)$$

Тоді $f(2k) \leq 2 \cdot 5(k - 1) = 10(k - 1) = 10k - 10$.

Нам потрібно довести, що ця величина не перевищує нашу цільову оцінку для $2k$: $5(2k - 2) = 10k - 10$.

Отримуємо $10k - 10 \leq 10k - 10$. Нерівність перетворюється на рівність, отже твердження вірне.

2. n непарне, $n = 2k + 1$.

$$f(2k + 1) \leq 2f(k + 1) \leq 2 \cdot 5(k - 1) = 10k - 10. \quad (2.2.6)$$

Цільова оцінка: $5((2k + 1) - 2) = 5(2k - 1) = 10k - 5$.

Перевіряємо: $10k - 10 \leq 10k - 5$. Це еквівалентно $-10 \leq -5$, що очевидно вірно.

Таким чином, крок індукції доведено для всіх n . Теорема доведена. $\square$

Цей результат є важливим кроком вперед, оскільки він математично строго обґрунтовує лінійність, відкидаючи можливість поганих конфігурацій, які могли б дати надлінійний ріст.

Розглянемо узагальнення задачі на випадок відрізків різної довжини та дослідили внутрішню структуру множини XOR-сум.

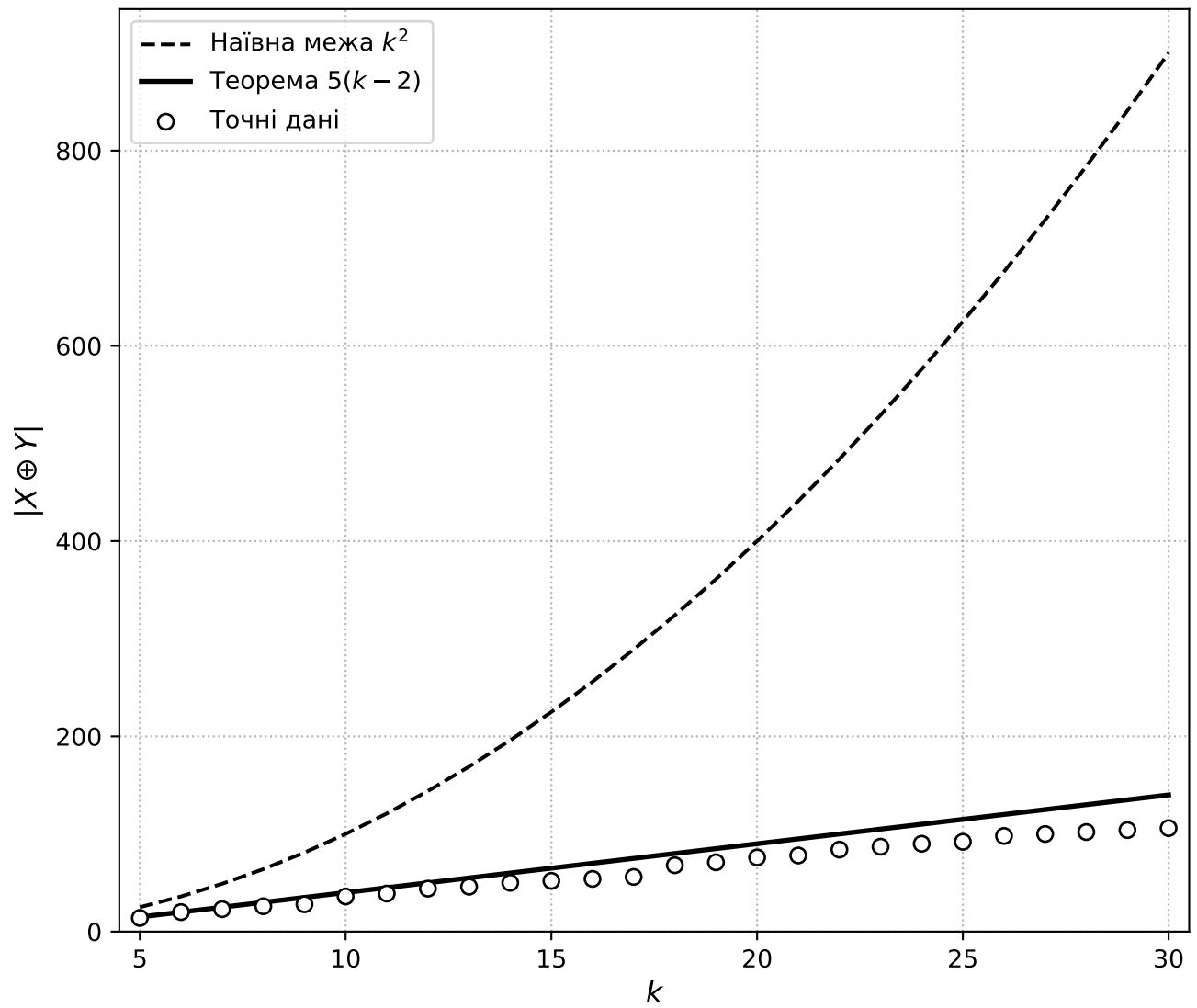


Рис. 2.5: Порівняння лінійної верхньої оцінки $5(k-2)$ з квадратичною оцінкою k^2 та реальними значеннями $f(k)$ для k до 30.

Нехай маємо два відрізки довжинами k_1 та k_2 , де $k_1 \leq k_2$. Позначимо $g(k_1, k_2) = \max_{x,y} |[x, x + k_1) \oplus [y, y + k_2)|$.

Твердження 1. Для $k_1 \leq k_2$, $g(k_1, k_2) = O(k_2)$.

Тобто верхня межа залежить лінійно від довжини *довшого* відрізка.

Доведення. Довший відрізок $[y, y + k_2)$ можна подати як об'єднання $\lceil k_2/k_1 \rceil$ відрізків довжини k_1 . Нехай $Y = \bigcup_{i=0}^{m-1} Y_i$, де Y_i — відрізок довжини k_1 , а $m \approx k_2/k_1$. Тоді $X \oplus Y = X \oplus (\bigcup Y_i) = \bigcup (X \oplus Y_i)$. Потужність об'єднання не перевищує суми потужностей: $|X \oplus Y| \leq \sum_{i=0}^{m-1} |X \oplus Y_i|$. За основною теоремою, кожен доданок $|X \oplus Y_i| \leq 5(k_1 - 2) \approx 5k_1$. Тоді загальна сума $\leq \frac{k_2}{k_1} \cdot 5k_1 = 5k_2$. Отже, $|X \oplus Y| \leq 5k_2$. $\square$

Цей результат має цікаву інтерпретацію: якщо довгий потік даних сканується коротким вікном за допомогою XOR, ентропія результату обмежена довжиною потоку, а не добутком довжин.

Множина $[x, x + k) \oplus [y, y + k)$, як правило, не є суцільним відрізком. Множина містить пропущені числа. Виявлено, що структура дірок нагадує структуру фракталу, що виникає при побудові трикутника Паскаля за модулем 2 (трикутник Серпінського). Це не дивно, оскільки операція XOR тісно пов'язана з додаванням в полі $\mathbb{F}_2$.

На основі рекурсивної леми можна побудувати алгоритм обчислення $f(k)$ значно швидший за перебір. Замість перевірки всіх x, y , можна використовувати динамічне програмування на дереві бітових представлень. Це відкриває шлях до обчислення точних значень $f(k)$ для $k \approx 10^9$ і вище, що неможливо методом прямого перебору. Складність такого алгоритму оцінюється як $O(k^{\log_2 3})$ або навіть полілогарифмічна від k при використанні техніки цифрового динамічного програмування (англ. *digit dynamic programming*).

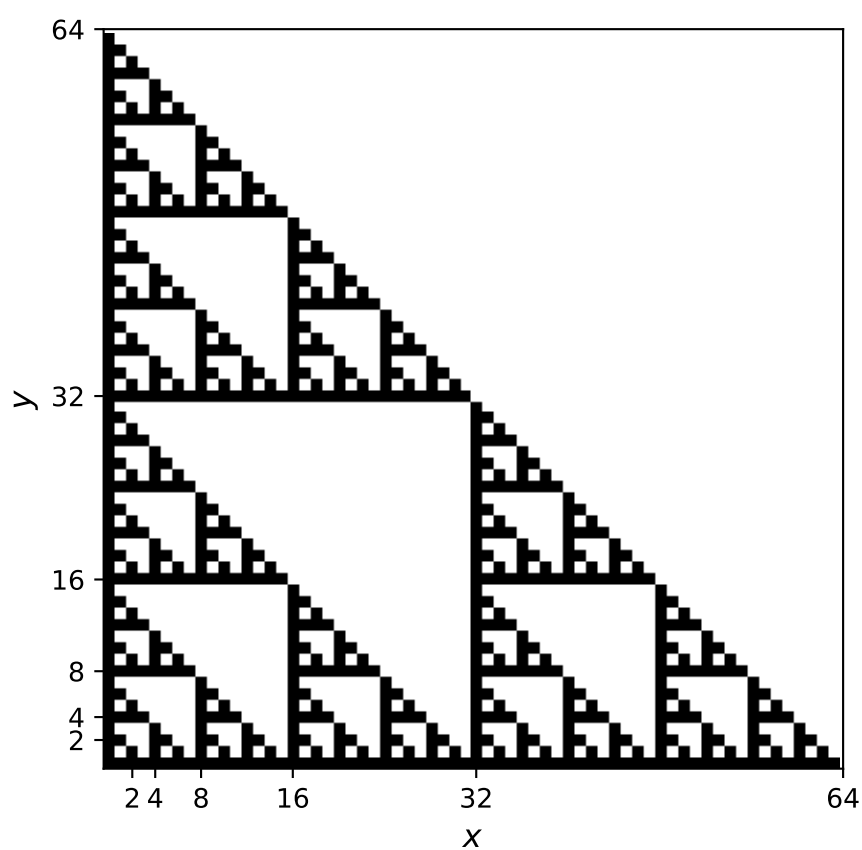


Рис. 2.6: Чорні клітинки позначають пари x, y для яких $x \& y = 0$. Ця структура нагадує трикутник Серпінського.

2.3. Лексикографічна декомпозиція рядка

Нехай S — рядок довжини n над скінченним впорядкованим алфавітом. Нехай A — спочатку порожній рядок. Визначимо операцію над S наступним чином:

1. Знайти підрядок $S[l \dots r]$, який є лексикографічно найбільшим серед усіх підрядків S . У разі неоднозначності обрати будь-який.
2. Приєднати $S[l \dots r]$ до кінця рядка A .
3. Видалити $S[l \dots r]$ з S , об'єднавши частини, що залишилися:

$$S := S[1 \dots l] + S(r \dots n).$$

Такі операції повторюються доки S не стане порожнім. Необхідно знайти кінцеве значення A .

Для побудови ефективного алгоритму спочатку доведемо ключову властивість максимального підрядка.

Лема 2. *Лексикографічно найбільший підрядок рядка S завжди є суфіксом.*

Доведення. Припустимо від супротивного, що максимальний підрядок $M = S[l \dots r]$ не є суфіксом, тобто $r < n$. Розглянемо підрядок $M' = S[l \dots r + 1]$. Оскільки M' має M як префікс, і M' довший за M , то в лексикографічному порядку $M' > M$. Це суперечить припущенню, що M є максимальним. Отже, $r = n$, і M є суфіксом. $\square$

З леми 2 випливає, що на кожному кроці видаляється суфікс поточного рядка. Це означає, що ліва частина рядка залишається незмінною, і початковий рядок S фактично розбивається на блоки справа наліво. Нехай S розбивається на підрядки $B_k, B_{k-1}, \dots, B_1$, де B_1 — перший знайдений максимальний суфікс, B_2 — максимальний суфікс залишку і так далі. Тоді відповідь $A = B_1 + B_2 + \dots + B_k$. Зауважимо, що в термінах індексів початкового рядка це відповідає вибору послідовності індексів розбиття $1 = i_k < i_{k-1} < \dots < i_1 \leq n$.

Будемо обробляти рядок справа наліво та підтримувати стек індексів L , що

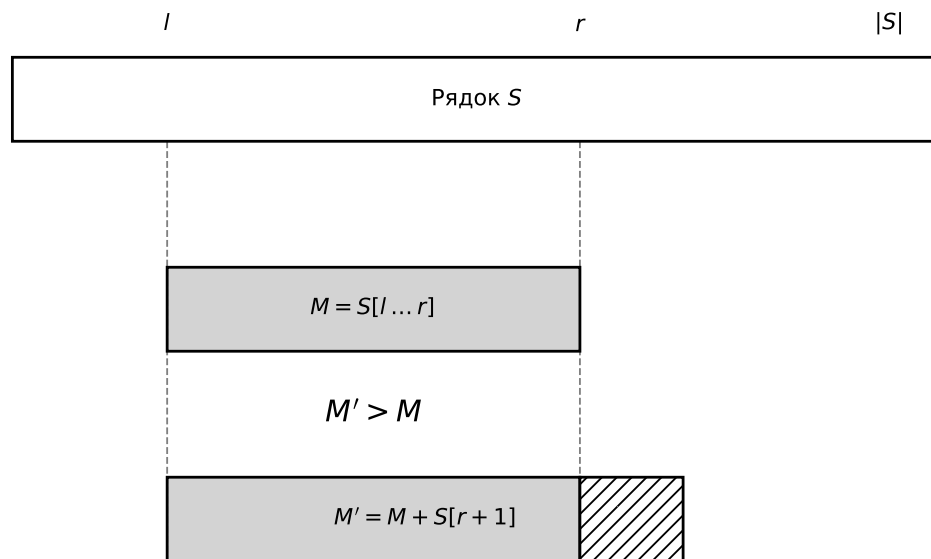


Рис. 2.7: Ілюстрація леми 2. Якщо максимальний підрядок не є суфіксом, можна розширити його на один символ праворуч, отримавши більший підрядок.

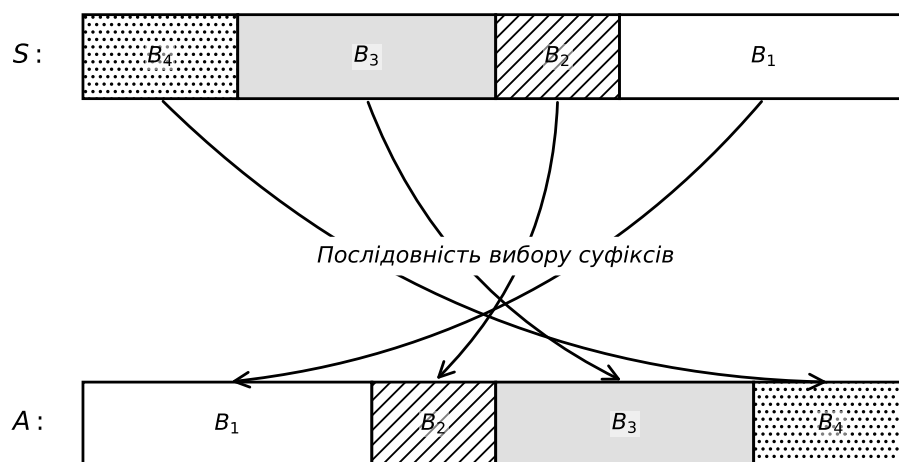


Рис. 2.8: Ілюстрація розбиття рядка на блоки за допомогою максимальних суфіксів. Результат A утворюється конкатенацією блоків у зворотному порядку.

представляють початки потенційних оптимальних суфіксів.

Algorithm Solve(S):

1. Побудувати масив найдовших спільних префіксів (LCP)
2. Побудувати розріджену таблицю над LCP
3. Ініціалізувати стек $L = [n + 1]$.
4. for i від n до 1 :
 - while $L.size() \geq 2$:
 - $x = L.top()$
 - $y = L.second_top()$
 - if CompareSubstrings(i, y, x, y):
 - $L.pop()$
 - else:
 - break
 - $L.push(i)$
5. Відновити відповідь A , використовуючи індекси зі стеку L .

Функція CompareSubstrings порівнює $S[i_1 \dots j_1]$ та $S[i_2 \dots j_2]$:

- Знаходимо довжину найдовшого спільного префікса

$$l = lcp(S[i_1 \dots], S[i_2 \dots]) \quad (2.3.1)$$

- Обмежуємо l мінімальною довжиною порівнюваних підрядків:

$$l = \min(l, j_1 - i_1, j_2 - i_2).$$

- Якщо l більше або дорівнює довжині коротшого підрядка, то довшим вважається той, що має більшу довжину. У алгоритмі завжди порівнюємо $S[i \dots y]$ та $S[x \dots y]$, де $i < x$, тому перший рядок завжди довший і містить другий як префікс. Отже, якщо $l \geq y - x$, перший рядок лексикографічно більший.
- Якщо $l < y - x$, порівнюємо символи $S[i_1 + l]$ та $S[i_2 + l]$.

Теорема 5. Запропонований алгоритм має часову складність $O(n)$.

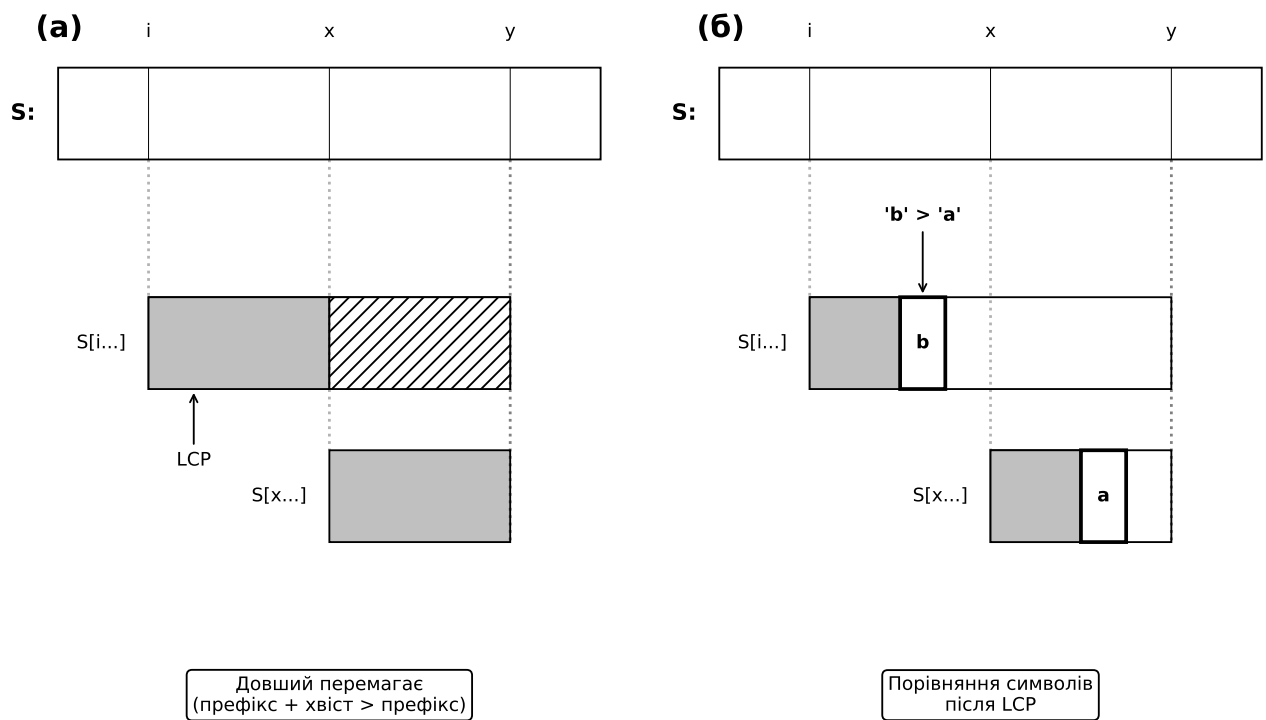


Рис. 2.9: Ілюстрація порівняння підрядків за допомогою масиву найдовших спільних префіксів. Використання розрідженої таблиці дозволяє швидко знаходити довжину спільного префікса.

Доведення. Побудова суфіксного масиву (англ. *suffix array*) та масиву найдовших спільних префіксів (англ. *longest common prefix*) займає $O(n)$ [128].

Кожен індекс i додається до стеку рівно один раз. Кожен індекс може бути видалений зі стеку не більше одного разу. Отже, кількість операцій зі стеком $O(n)$. Використання розрідженої таблиці (англ. *sparse table*) для масиву найдовших спільних префіксів дозволяє виконувати порівняння за $O(1)$.

Сумарна складність $O(n)$ при використанні лінійних алгоритмів побудови суфіксного масиву. Просторова складність становить $O(n)$ для зберігання рядка та допоміжних масивів. $\square$

Для порівняння, наївний алгоритм, що шукає максимальний суфікс шляхом прямого перебору на кожному кроці, має складність $O(n^2)$ у найкращому випадку і $O(n^3)$ з урахуванням порівняння рядків, що є неприйнятним.

2.4. Мінімум-плюс згортка

Нехай задано дві цілочисельні послідовності $A = (a_0, a_1, \dots, a_{n-1})$ та $B = (b_0, b_1, \dots, b_{n-1})$. Задачею про $(\min, +)$ згортку є обчислення послідовності $C = (c_0, c_1, \dots, c_{2n-2})$, де кожен елемент визначається як:

$$c_k = \min_{i+j=k} (a_i + b_j), \quad \text{для } 0 \leq k \leq 2n - 2. \quad (2.4.1)$$

Тут індекси i та j пробігають значення від 0 до $n - 1$.

Гіпотеза 4. У стандартній моделі обчислень не існує алгоритму, який розв'язує задачу $(\min, +)$ згортки для послідовностей довжини n з елементами порядку $O(\text{poly}(n))$ за час $O(n^{2-\varepsilon})$ для будь-якого фіксованого $\varepsilon > 0$.

Зауважимо, що на відміну від класичної згортки, яка обчислюється за лінеаритмічний час $O(n \log n)$ через швидке перетворення Фур'є (FFT), $(\min, +)$ згортка визначена над тропічним напівкільцем $(\mathbb{R} \cup \{\infty\}, \min, +)$, для якого невідомо аналогів теореми про згортку, що дозволяли б таку ж ефективність. Найкращий відомий алгоритм, запропонований Bremner et al., має складність

Треїсінг алгоритму для $S = abacaba$

i	S[i]	Дія / Порівняння	Стан Стеку
7	'a'	Стек порожній. Push 7.	7
6	'b'	$ba > a \rightarrow$ Pop 7. Push 6.	6
5	'a'	$aba < ba \rightarrow$ Push 5.	6 5
4	'c'	$caba > ba \rightarrow$ Pop 6. $caba > aba \rightarrow$ Pop 5. Push 4.	4
3	'a'	$acaba < caba \rightarrow$ Push 3.	4 3
2	'b'	$bacaba > acaba \rightarrow$ Pop 3. $bacaba < caba \rightarrow$ Push 2.	4 2
1	'a'	$abacaba < bacaba \rightarrow$ Push 1.	4 2 1

Фінальний результат (індекси в стеку: 1, 2, 4):

B_3 (a)		B_2 (ba)		B_1 (caba)		
a	b	a	c	a	b	a
1	2	3	4	5	6	7

Рис. 2.10: Приклад роботи алгоритму для рядка $S = abacaba$. Показано вибір максимальних суфіксів на кожному кроці та формування відповіді A .

$O\left(\frac{n^2(\log \log n)^3}{\log^2 n}\right)$ [112], що є лише асимптотично трохи кращим за тривіальний $O(n^2)$.

Задача k -TSP полягає у пошуку найкоротшого маршруту, що відвідує щонайменше k точок із заданої множини V у метричному просторі. У одновиірному випадку множина точок $X = \{x_0, x_1, \dots, x_{n-1}\}$ розташована на дійсній прямій. Без обмеження загальності вважатимемо, що точки відсортовані: $x_0 \leq x_1 \leq \dots \leq x_{n-1}$. Задача **1D all k -TSP** полягає в обчисленні вартості оптимального маршруту y_k для кожного можливого розміру квоти $k \in \{1, \dots, n\}$.

Лема 3. Для 1D k -TSP оптимальний набір з k точок завжди є неперервним підвідрізком відсортованої послідовності точок. Тобто, існує індекс i ,

такий що оптимальний набір точок є $\{x_i, x_{i+1}, \dots, x_{i+k-1}\}$. Довжина найкоротшого маршруту, що відвідує ці точки, дорівнює $x_{i+k-1} - x_i$.

Доведення. Розглянемо довільний набір з k точок $S \subset X$. Нехай $x_{\min}$ — мінімальна координата в S , а $x_{\max}$ — максимальна. Будь-який маршрут, що відвідує всі точки з S , повинен відвідати $x_{\min}$ і $x_{\max}$. Оскільки рух відбувається на прямій, відстань між цими точками є нижньою межею довжини маршруту. Маршрут, що просто йде від $x_{\min}$ до $x_{\max}$, досягає цієї межі і має довжину $x_{\max} - x_{\min}$. Щоб мінімізувати цю величину, нам необхідно вибрати k точок так, щоб різниця між максимальною та мінімальною була найменшою. Нехай зафіксовано $x_{\min} = x_i$ та $x_{\max} = x_j$ ($i < j$). Кількість точок, доступних у інтервалі $[x_i, x_j]$, дорівнює $j - i + 1$. Щоб задовольнити квоту, повинно виконуватися $j - i + 1 \geq k$. Оскільки функція цілі $x_j - x_i$ монотонно зростає зі збільшенням j (при фіксованому i), оптимальним є вибір мінімально можливого j , тобто $j = i + k - 1$. Це відповідає вибору рівно k послідовних точок: $x_i, \dots, x_{i+k-1}$. $\square$

Сценарій А (Неоптимальний)



Сценарій Б (Оптимальний)

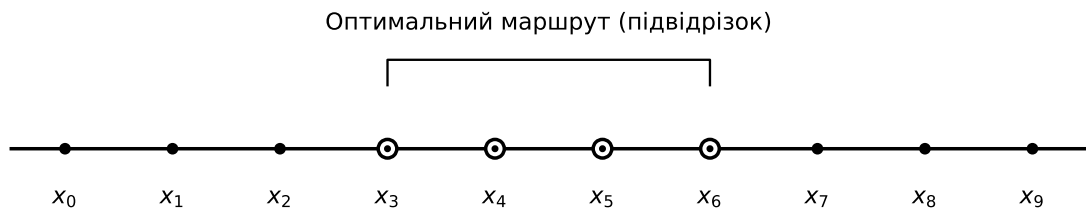


Рис. 2.11: Ілюстрація до Лема 3.

Отже, задачу 1D all k -TSP можна формалізувати як обчислення послідовності $Y = (y_1, \dots, y_n)$, де:

$$y_k = \min_{i=0}^{n-k} (x_{i+k-1} - x_i). \quad (2.4.2)$$

Наївне обчислення за цією формулою вимагає $O(n)$ операцій для кожного k , що сумарно дає $O(n^2)$.

У цьому підрозділі наведено конструктивні доведення зведення однієї задачі до іншої, уточнюючи алгоритмічні кроки та аналіз складності.

Теорема 6. *Задача 1D all k -TSP лінійно зводиться до задачі $(\min, +)$ згортки. Тобто, якщо $T_{conv}(n)$ — час розв'язання згортки, то 1D all k -TSP розв'язується за $T_{tsp}(n) = O(n) + T_{conv}(n)$.*

Доведення. Нехай $x = (x_0, \dots, x_{n-1})$ — вхідний відсортований масив точок. Наша мета — обчислити y_k згідно з (2.4.2). Сформуємо дві послідовності A та B довжини n :

$$a_i = x_{n-1-i}, \quad i = 0, \dots, n-1, \quad (2.4.3)$$

$$b_j = -x_j, \quad j = 0, \dots, n-1. \quad (2.4.4)$$

Це перетворення вимагає $O(n)$ часу. Нехай C — $(\min, +)$ згортка послідовностей A та B :

$$c_m = \min_{i+j=m} (a_i + b_j).$$

Розкриємо вираз для c_m , підставляючи означення A та B :

$$c_m = \min_{i+j=m} (x_{n-1-i} - x_j).$$

Виразимо i через m та j : $i = m - j$. Підставимо у індекс при x :

$$n - 1 - i = n - 1 - (m - j) = n - 1 - m + j.$$

Тоді:

$$c_m = \min_j (x_{j+(n-1-m)} - x_j).$$

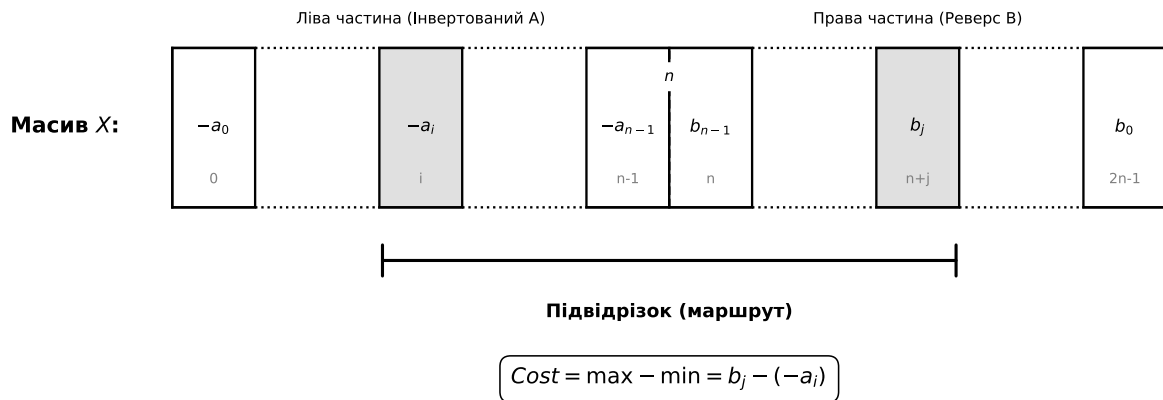


Рис. 2.12: Ілюстрація до Теорема 6.

Порівняємо отриманий вираз із цільовою формулою (2.4.2):

$$y_k = \min_j (x_{j+k-1} - x_j). \quad (2.4.5)$$

Вирази під знаком мінімуму стають ідентичними, якщо покласти розмір вікна $(n - 1 - m)$ рівним $(k - 1)$.

$$k - 1 = n - 1 - m \implies k = n - m.$$

Або, еквівалентно, $m = n - k$. Таким чином, k -й елемент розв'язку задачі TSP (де k — кількість точок) точно відповідає $(n - k)$ -му елементу результату згортки:

$$y_k = c_{n-k}. \quad (2.4.6)$$

Оскільки індекси згортки m пробігають значення від 0 до $2n - 2$, покриваємо всі необхідні значення k від 1 до n (при m від $n - 1$ до 0). Решта значень згортки можуть бути відкинуті. Отже, алгоритм для згортки повністю розв'язує задачу 1D all k -TSP. $\square$

Зворотнє зведення є більш нетривіальним через вимогу задачі TSP щодо впорядкованості вхідного масиву.

Теорема 7. *Задача $(\min, +)$ згортки для послідовностей з елементами в діапазоні $[0, W]$ зводиться до задачі 1D all k -TSP. Якщо $T_{tsp}(N, W)$ — час розв’язання 1D all k -TSP для N точок з координатами до W , то згортка обчислюється за $O(n) + T_{tsp}(2n, O(nW))$.*

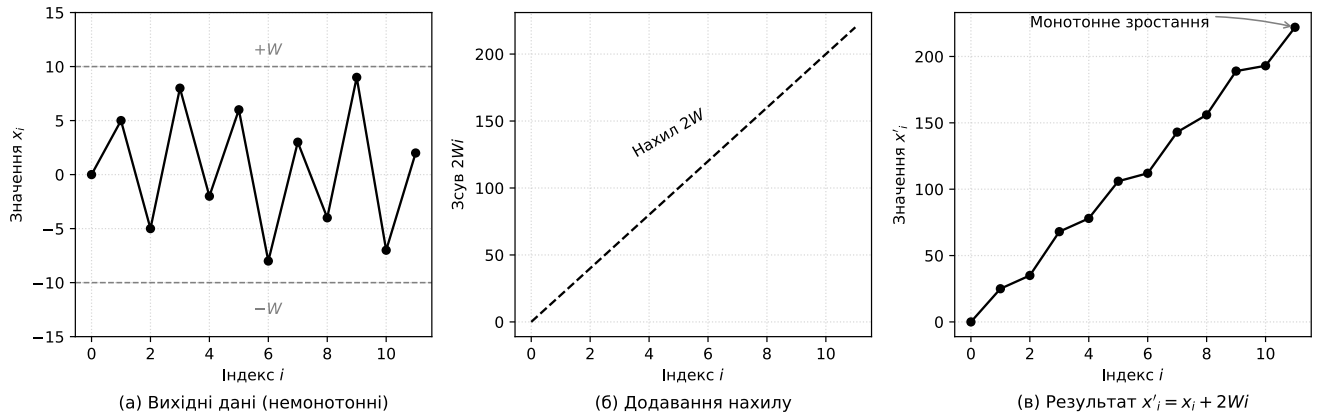


Рис. 2.13: Ілюстрація до Теорема 7.

Доведення. Нехай задано довільні послідовності A та B довжини n .

Крок 1: Конструювання об’єднаного масиву. Розглянемо масив X довжини $2n$, побудований наступним чином:

$$X = (-a_0, -a_1, \dots, -a_{n-1}, b_{n-1}, b_{n-2}, \dots, b_0). \quad (2.4.7)$$

Формально:

$$x_i = \begin{cases} -a_i, & 0 \leq i < n, \\ b_{2n-1-i}, & n \leq i < 2n. \end{cases} \quad (2.4.8)$$

Розглянемо вираз для 1D k -TSP на цьому масиві для конкретного “зсуву” (або квоти). Нехай шукаємо мінімум різниці $x_{\text{right}} - x_{\text{left}}$, де x_{right} береться з другої половини масиву (частина B), а x_{left} — з першої (частина $-A$). Нехай $x_{\text{left}} = x_i = -a_i$ ($0 \leq i < n$). Нехай $x_{\text{right}} = x_j = b_{2n-1-j}$ ($n \leq j < 2n$). Тоді різниця:

$$x_j - x_i = b_{2n-1-j} - (-a_i) = a_i + b_{2n-1-j}.$$

Хочемо, щоб це відповідало $a_i + b_u$ при фіксованій сумі індексів $i + u = m$.

Покладемо $u = 2n - 1 - j$. Тоді умова $i + u = m$ перетворюється на:

$$i + (2n - 1 - j) = m \implies j - i = 2n - 1 - m.$$

Величина $j - i$ у задачі TSP відповідає $K - 1$, де K — кількість точок у вікні.

Тобто, поклавши квоту $K = 2n - m$, отримаємо:

$$y_K = \min_{j-i=K-1} (x_j - x_i) = \min_{i+(2n-1-j)=m} (a_i + b_{2n-1-j}) = \min_{i+u=m} (a_i + b_u) = c_m.$$

Крок 2: Забезпечення монотонності. Масив X , побудований на першому кроці, не обов'язково є відсортованим, що порушує умову вхідних даних для 1D k -TSP. Для виправлення цього застосуємо техніку лінійного зсуву. Визначимо модифікований масив X' :

$$x'_i = x_i + 2i \cdot W. \quad (2.4.9)$$

Перевіримо умову сортування $x'_{i+1} \geq x'_i$:

$$x'_{i+1} - x'_i = (x_{i+1} + 2(i+1)W) - (x_i + 2iW) = (x_{i+1} - x_i) + 2W.$$

Оскільки за умовою вхідні дані обмежені, $a_k, b_k \in [0, W]$, то будь-який елемент $x_k \in [-W, W]$. Максимально можлива різниця між двома довільними елементами становить $W - (-W) = 2W$. Отже, $x_{i+1} - x_i \geq -2W$. Підставляючи це, отримуємо:

$$x'_{i+1} - x'_i \geq -2W + 2W = 0.$$

Таким чином, послідовність X' є неспадною.

Крок 3: Відновлення результату. Застосуємо алгоритм 1D all k -TSP до

X' , отримавши значення y'_k .

$$\begin{aligned}
y'_k &= \min_{j-i=k-1} (x'_j - x'_i) \\
&= \min_{j-i=k-1} ((x_j + 2jW) - (x_i + 2iW)) \\
&= \min_{j-i=k-1} (x_j - x_i + 2W(j - i)) \\
&= \min_{j-i=k-1} (x_j - x_i) + 2W(K - 1) \\
&= y_k + 2W(K - 1).
\end{aligned}$$

Звідси $y_k = y'_k - 2W(K - 1)$. Знаючи y_k , отримуємо c_m (де $m = 2n - k$). $\square$

Зауваження: максимальне значення елемента в X' становить приблизно $W + 2(2n)W \approx 4nW$. Це збільшення діапазону значень є платою за зведення. Воно вносить логарифмічний множник у складність.

Нехай є алгоритм для 1D all k -TSP, що дає наближений розв'язок. Постає питання перенесення наближення на задачу $(\min, +)$ згортки.

1. Припустимо, алгоритм для TSP повертає $\tilde{y}_k$ таке, що $y_k \leq \tilde{y}_k \leq y_k + \delta$.

Оскільки $c_m = y_{2n-m} - \text{const}$ (у зворотньому зведенні, після корекції на $2W(K - 1)$), адитивна похибка зберігається без змін.

$$\tilde{c}_m = \tilde{y}_{2n-m} - \text{shift} \leq y_{2n-m} + \delta - \text{shift} = c_m + \delta.$$

Отже, зведення є адитивно-стійким.

2. Припустимо, що маємо $(1 + \varepsilon)$ -наближення для TSP на масиві X' :

$$\tilde{y}'_k \leq (1 + \varepsilon)y'_k.$$

Розкриємо вираз через вихідні змінні згортки:

$$\tilde{c}_m + \text{shift} \leq (1 + \varepsilon)(c_m + \text{shift}).$$

$$\tilde{c}_m \leq (1 + \varepsilon)c_m + \varepsilon \cdot \text{shift}.$$

Величина зсуву $\text{shift} = 2W(k - 1) \approx 2W \cdot n$ для $k = n$. Це значення є дуже великим порівняно з цільовим значенням згортки c_m (яке може

бути порядку W). Фактична похибка становить $\varepsilon \cdot c_m + \varepsilon \cdot 2nW$. Навіть при малому ε , доданок $\varepsilon \cdot 2nW$ може домінувати над самим значенням розв'язку.

Це призводить до важливого негативного результату:

Наслідок 1. *Стандартне зведення через лінійний зсув $x'_i = x_i + 2iW$ не зберігає коефіцієнт мультиплікативної апроксимації. Поліноміальна схема наближення (англ. polynomial-time approximation scheme) для 1D all k -TSP на великих координатах не трансформується автоматично у PTAS для $(\min, +)$ згортки.*

Опуклі послідовності відіграють ключову роль у швидких алгоритмах динамічного програмування [109]. Послідовність A називається опуклою, якщо $a_{i+1} - a_i \geq a_i - a_{i-1}$ для всіх i .

Теорема 8. *Лінійна трансформація $x'_i = x_i + C \cdot i$, використана для сортування масиву у Теоремі 7, зберігає властивість опуклості вихідної послідовності.*

Доведення. Друга скінченна різниця для X' виглядає так:

$$\begin{aligned} \Delta^2 x'_i &= x'_{i+1} - 2x'_i + x'_{i-1} \\ &= (x_{i+1} + C(i+1)) - 2(x_i + Ci) + (x_{i-1} + C(i-1)) \\ &= (x_{i+1} - 2x_i + x_{i-1}) + C(i+1 - 2i + i - 1) \\ &= \Delta^2 x_i. \end{aligned}$$

Отже, X' є опуклою тоді й лише тоді, коли X є опуклою. □

Наслідок. Відомо, що $(\min, +)$ згортка двох опуклих послідовностей (або навіть однієї опуклої та однієї довільної) може бути обчислена за час $O(n)$ за допомогою алгоритму SMAWK або методів Eppstein et al. [97]. Завдяки нашому зведенню, це означає, що задача **1D all k -TSP** може бути розв'язана за $O(n)$, якщо множина точок X має властивість “прискореного зростання” (відстані між сусідніми точками зростають), що відповідає опуклості координат. Це

суттєве покращення порівняно з загальним випадком.

2.5. Зигзагоподібні послідовності у деревах

Нехай $T = (V, E)$ — зв'язний ациклічний граф (дерево) з множиною вершин V , $|V| = N$. Кожній вершині $v \in V$ зіставлено мітку $A_v \in \{1, 2\}$. Послідовність вершин $S = (v_1, v_2, \dots, v_k)$ довжини $k \geq 2$ називається *зигзагоподібною*, якщо виконуються дві умови:

1. **Топологічна умова:** Вершини $v_1, \dots, v_k$ лежать на простому шляху в дереві, причому порядок їх слідування в S відповідає порядку їх відвідування при русі від v_1 до v_k .
2. **Знакова умова:** Значення міток суворо чергуються. Для випадку бінарних міток $\{1, 2\}$ це означає, що для всіх $1 \leq i < k$, $A_{v_i} \neq A_{v_{i+1}}$.

Необхідно знайти кількість унікальних зигзагоподібних послідовностей. Дві послідовності вважаються різними, якщо вони відрізняються довжиною або набором вершин. Послідовність $(v_1, \dots, v_k)$ та $(v_k, \dots, v_1)$ вважаються ідентичними (неорієнтовані шляхи).

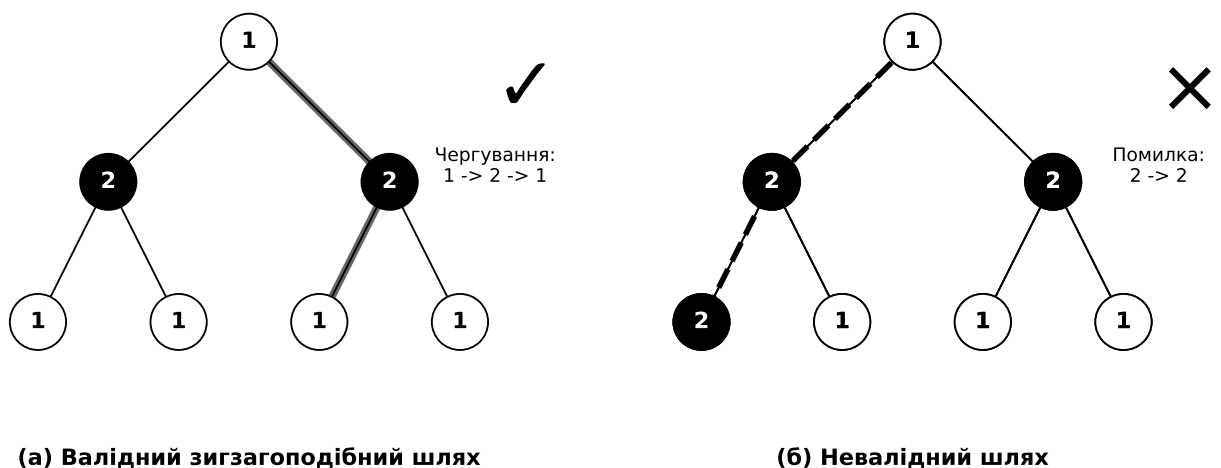


Рис. 2.14: Приклад дерева з мітками та зигзагоподібними послідовностями.

Для розв'язання задачі використаємо метод динамічного програмування на дереві. Вкоренимо дерево у вершині 1.

Стани динамічного програмування. Для кожної вершини u визначимо $DP[u]$ як кількість зигзагоподібних послідовностей, що починаються в u і повністю лежать у піддереві u . Для зручності обчислень будемо вважати одиночну вершину u послідовністю довжини 1. Оскільки мітки набувають лише значень 1 та 2, введемо допоміжні величини для агрегації інформації з дітей:

- $S_1[u]$ — сума $DP[v]$ для всіх v у піддереві u , таких що $A_v = 1$.
- $S_2[u]$ — сума $DP[v]$ для всіх v у піддереві u , таких що $A_v = 2$.

Алгоритм складається з одного обходу в глибину. Для кожної вершини u виконуємо наступні дії:

1. Рекурсивно обробляємо всіх дітей $v \in children(u)$.
2. Обчислюємо $S_1[u]$ та $S_2[u]$ як суму відповідних значень дітей.
3. Обчислюємо $DP[u]$:
 - Якщо $A_u = 1$, то продовжити шлях можна лише у вершини з міткою 2. Отже, $DP[u] = 1 + \sum_{v \in children(u)} S_2[v]$.
 - Якщо $A_u = 2$, аналогічно $DP[u] = 1 + \sum_{v \in children(u)} S_1[v]$.
4. Оновлюємо $S_{A_u}[u]$ додаванням $DP[u]$.
5. Підраховуємо відповідь, розглядаючи u як найменшого спільного предка для кінців шляху.

Будь-який шлях у дереві має унікальну вершину u , яка є LCA для його кінців. У цій точці можливі два випадки формування зигзагоподібної послідовності:

Випадок А: Вершина u входить у послідовність. Це означає, що ми з'єднуємо два шляхи, що йдуть вниз від u у різні піддерева. Оскільки u є точкою з'єднання, сусідні до неї вершини в обох гілках повинні мати мітку, відмінну від A_u . Кількість таких пар гілок обчислюється як сума добутків відповідних сум S для пар дітей. Нехай $target = 3 - A_u$ (протилежне значення). Тоді кількість

комбінацій:

$$Count_A = \sum_{v_i, v_j \in children(u), i < j} S_{target}[v_i] \cdot S_{target}[v_j]$$

Це можна обчислити за $O(deg(u))$ використовуючи префіксні суми або формулу квадрата суми:

$$\sum_{i < j} x_i x_j = \frac{1}{2} \left(\left(\sum x_i \right)^2 - \sum x_i^2 \right)$$

Випадок А: вершина u з'єднує дві гілки

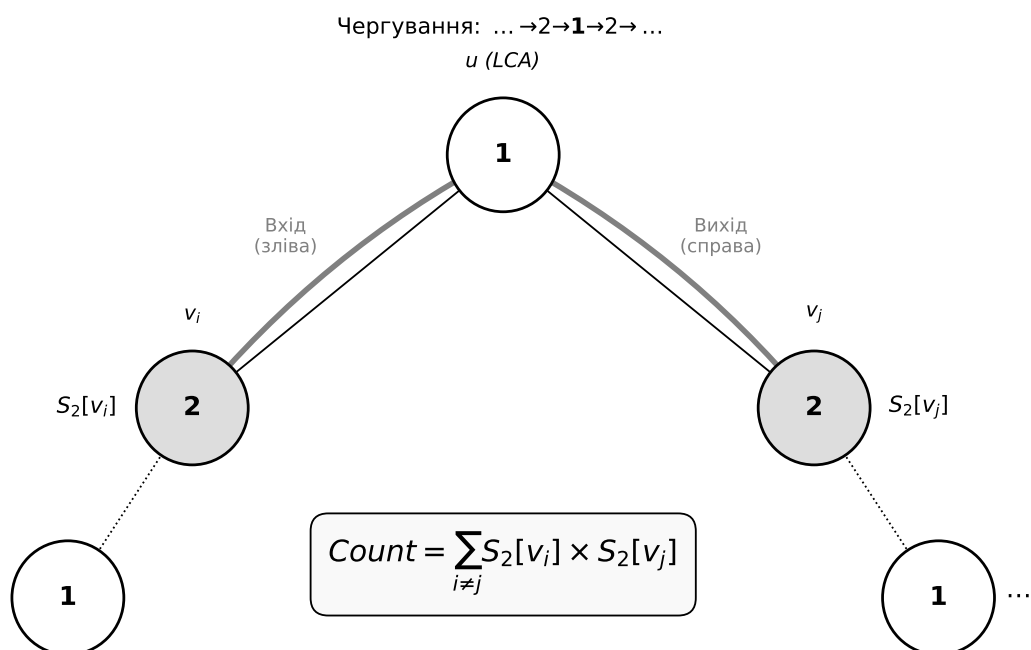


Рис. 2.15: Ілюстрація Випадку А: Вершина $u \in LCA$ і входить у послідовність.

Випадок Б: Вершина $u \in LCA$, але не входить у послідовність. Це можливо, якщо шлях проходить “через” u транзитом, але сама u не задовольняє умові чергування відносно сусідів, або ми просто пропускаємо її. Але згідно з умовою задачі (easy version), це еквівалентно з’єднанню гілки, що починається з 1, та гілки, що починається з 2, які походять з різних піддерев дітей u .

$$Count_B = \sum_{v_i, v_j \in children(u), i \neq j} S_1[v_i] \cdot S_2[v_j]$$

Це значення дорівнює:

$$\left(\sum S_1[v] \right) \cdot \left(\sum S_2[v] \right) - \sum (S_1[v] \cdot S_2[v])$$

Випадок Б: З'єднання 1→2 без участі u

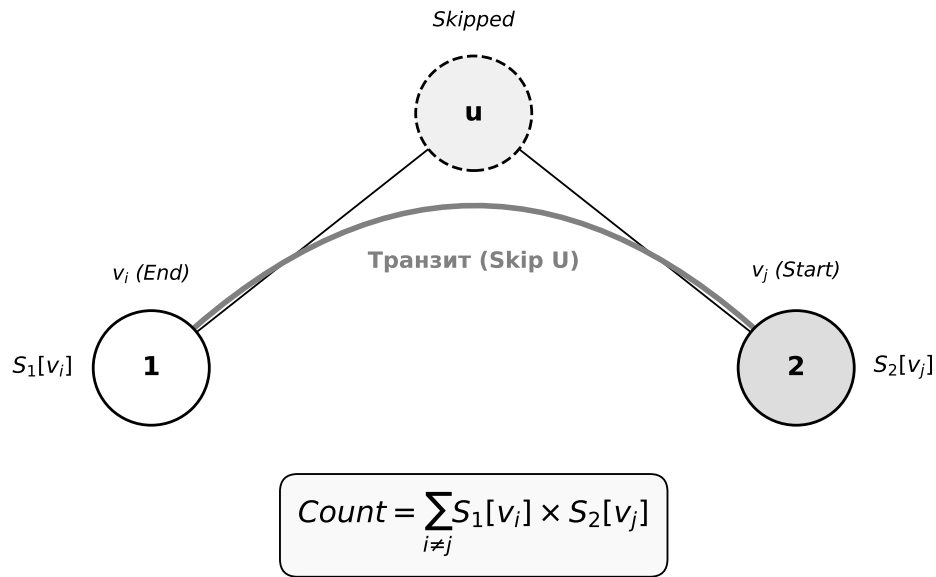


Рис. 2.16: Вершина $u \in LCA$, але не входить у послідовність.

Загальна відповідь є сумою $Count_A + Count_B$ по всіх вершинах u , плюс внесок вертикальних шляхів (врахованих у DP). Оскільки ми враховували шляхи довжини 1, у кінці віднімаємо N .

Теорема 9. *Запропонований алгоритм коректно обчислює кількість зигзагоподібних послідовностей за час $O(N)$.*

Доведення. Коректність. Алгоритм розглядає кожну можливу послідовність рівно один раз у точці її найменшого спільного предка (LCA). Розділення на випадки (LCA входить у послідовність чи ні) покриває всі можливі топологічні конфігурації шляхів. Використання динамічного програмування гарантує, що ми коректно підраховуємо кількість продовжень шляху вглиб дерева.

Складність. Алгоритм відвідує кожну вершину один раз. У кожній вершині виконується фіксована кількість арифметичних операцій над агрегованими значеннями від дітей. Обчислення комбінаторних сум оптимізовано до $O(1)$ за допомогою попередньо обчислених сум та сум квадратів (або за один прохід по дітях). Отже, часова складність є лінійною — $O(N)$. Просторова складність

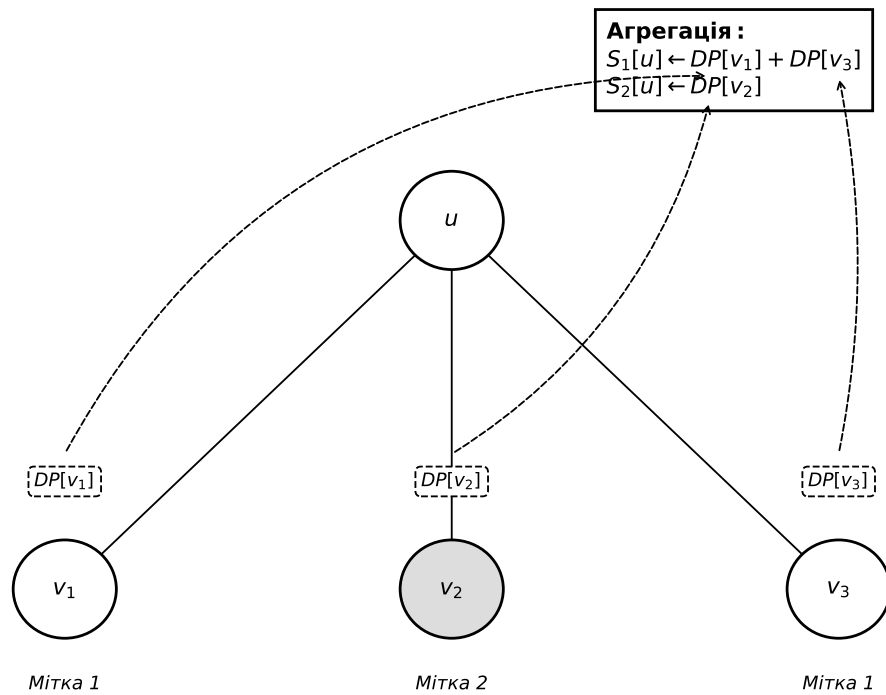


Рис. 2.17: Схема роботи алгоритму DFS для обчислення кількості зигзагоподібних послідовностей.

$O(N)$ для зберігання стеку рекурсії та масивів міток. □

Для порівняння, наївний перебір усіх пар вершин (u, v) з перевіркою шляху займав би $O(N^3)$ або $O(N^2)$ часу, що неприпустимо для $N = 10^5$.

2.6. Реберна зв'язність регулярних графів

Існування регулярних графів підпорядковується фундаментальному закону, відомому як Лема про рукостискання.

Лема 4 (Про рукостискання). У будь-якому скінченному неорієнтованому графі сума ступенів усіх вершин дорівнює подвоєній кількості ребер:

$$\sum_{v \in V} \deg(v) = 2|E| \quad (2.6.1)$$

Для k -регулярного графа це означає: $n \cdot k = 2|E|$.

Звідси випливає важлива необхідна умова існування: добуток nk повинен

бути парним числом. Це означає, що неможливо побудувати k -регулярний граф непарного порядку n , якщо k також є непарним.

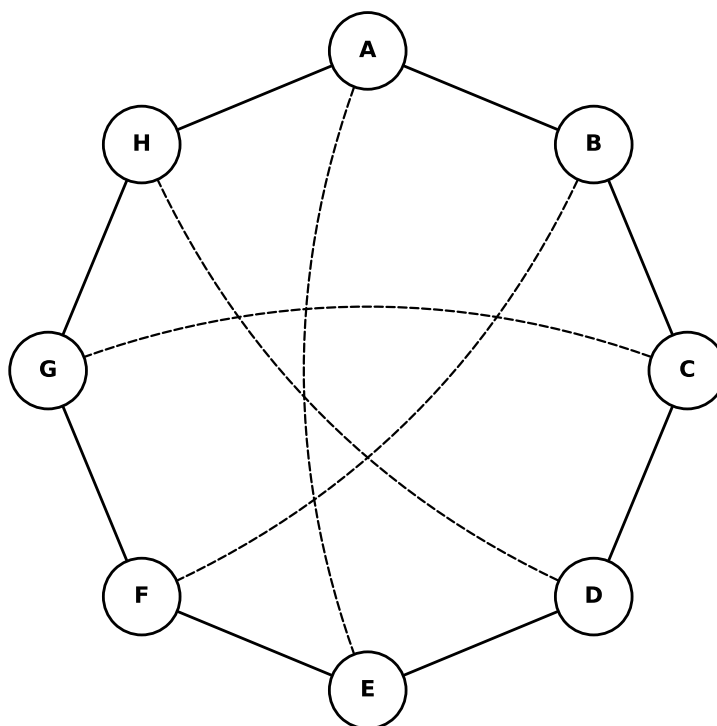


Рис. 2.18: Парне n і непарне k

У випадках, коли умова парності nk не виконується (тобто n і k непарні), розглядається клас майже регулярних графів.

Означення 19. Граф називається майже k -регулярним, якщо ступені всіх вершин дорівнюють k , за винятком однієї вершини, яка має ступінь $k - 1$ (або $k + 1$, залежно від контексту).

Ці графи відіграють критичну роль у нашому аналізі, оскільки вони часто виникають як індуковані підграфи при розрізанні регулярного графа.

Ключовим об'єктом нашого дослідження є реберна зв'язність.

Означення 20. Для підмножини вершин $S \subset V$ ($S \neq \emptyset, S \neq V$), розрізом $\partial(S)$ називається множина ребер, що з'єднують вершини з S з вершинами з доповнення $T = V \setminus S$:

$$\partial(S) = \{(u, v) \in E \mid u \in S, v \in T\} \quad (2.6.2)$$

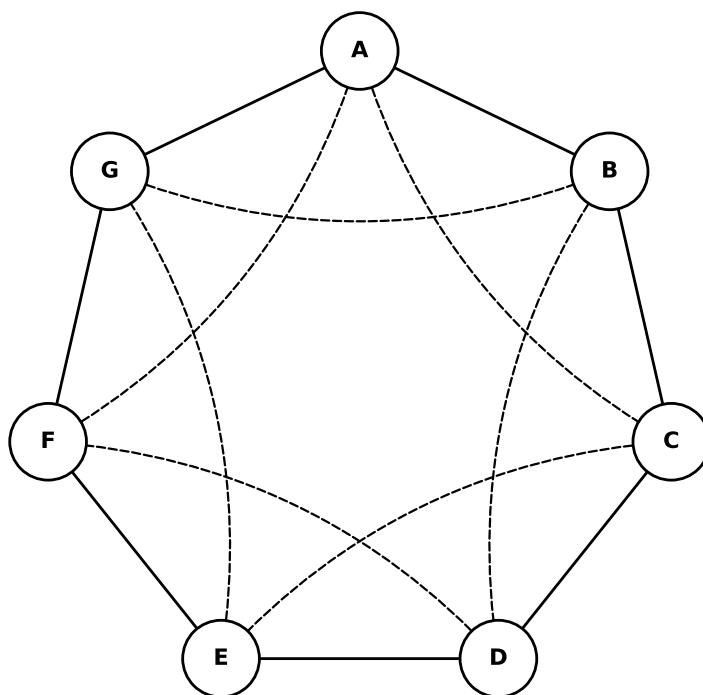


Рис. 2.19: Непарне n і парне k

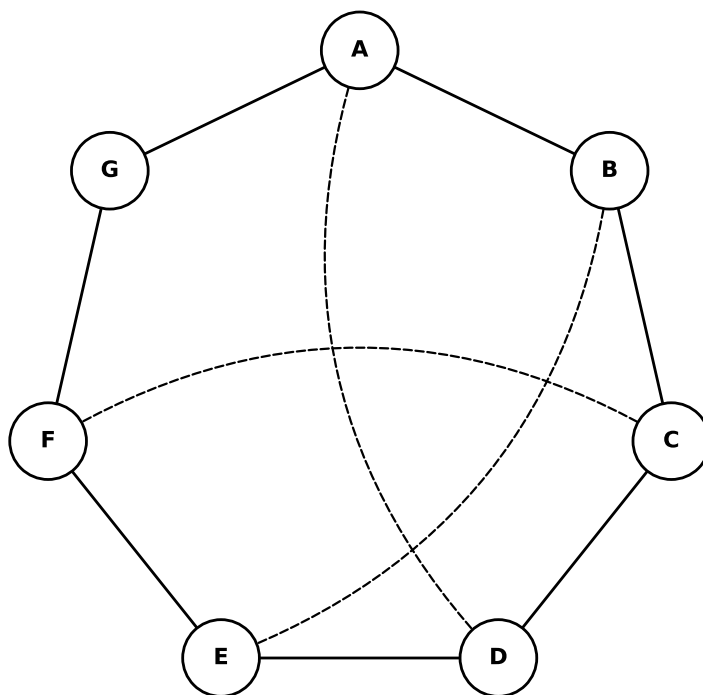


Рис. 2.20: Майже регулярний граф для непарних n і k

Розмір розрізу позначається як $|\partial(S)|$.

Означення 21. Реберною зв'язністю $\lambda(G)$ графа G називається мінімальний розмір розрізу серед усіх можливих підмножин S :

$$\lambda(G) = \min_{\emptyset \subsetneq S \subsetneq V} |\partial(S)| \quad (2.6.3)$$

Для будь-якого k -регулярного графа очевидно, що $\lambda(G) \leq k$, оскільки розріз, що відділяє одну вершину від решти графа, завжди містить рівно k ребер.

Теорема 10. Нехай G — зв'язний k -регулярний граф порядку n . Мінімумом можливого значення реберної зв'язності $\lambda_{\min}(n, k)$ визначається такими умовами:

1. **Випадок парного k :**

- Якщо $k < n < 2k + 2$, то $\lambda_{\min} = k$.
- Якщо $n \geq 2k + 2$, то $\lambda_{\min} = 2$.

2. **Випадок непарного k ($k > 1$):**

- Якщо $k < n < 2k + 2$, то $\lambda_{\min} = k$.
- Якщо $n = 2k + 2$, то $\lambda_{\min} = 2$.
- Якщо $n > 2k + 2$, то $\lambda_{\min} = 1$.

Зауваження: Умова $n > k$ є обов'язковою для існування будь-якого регулярного графа без кратних ребер та петель.

Для доведення теореми нам знадобиться оцінка кількості ребер у розрізі через розмір меншої компоненти розбиття.

Лема 5 (Квадратична оцінка розрізу). Нехай (S, T) — розбиття вершин k -регулярного графа, і нехай $|S| = x$. Тоді розмір розрізу задовольняє нерівність:

$$|\partial(S)| \geq x(k + 1 - x) \quad (2.6.4)$$

Доведення. Розглянемо суму ступенів вершин у множині S . Вона дорівнює $k \cdot x$. Ця сума складається з двох частин:

1. Ребра, що повністю лежать всередині S (позначимо їх кількість $e(S)$).

Кожне таке ребро враховується двічі.

2. Ребра розрізу $\partial(S)$, що з'єднують S з T . Кожне таке ребро враховується один раз.

Отже, маємо балансове рівняння: $kx = 2e(S) + |\partial(S)|$. Звідси $|\partial(S)| = kx - 2e(S)$. Оскільки граф простий, $e(S) \leq \binom{x}{2} = \frac{x(x-1)}{2}$. Підставивши це в рівняння, отримуємо: $|\partial(S)| \geq kx - 2 \cdot \frac{x(x-1)}{2} = x(k+1-x)$. $\square$



Рис. 2.21: Ілюстрація леми про ємність розрізу.

Проведено аналіз, розбиваючи простір параметрів (n, k) на області.

Випадок 1: Малі графи ($n < 2k + 2$). Нехай S — менша компонента, $|S| \leq k$. Функція $f(x) = x(k+1-x)$ на інтервалі $x \in [1, k]$ досягає мінімуму на кінцях (k). Отже, будь-який розріз має розмір щонайменше k . Таким чином, $\lambda_{\min} = k$.

Випадок 2: Парне k та великі графи ($n \geq 2k + 2$). Розріз розміру 1 неможливий через обмеження парності (сума ступенів у компоненті S стала б непарною $k|S| - 1$). Розріз розміру 2 можливий і може бути сконструйований шляхом “зшивання” двох регулярних графів.

Випадок 3: Непарне k та критичний порядок $n = 2k + 2$. Для існування 1-розрізу необхідно, щоб $|S|$ було непарним (для збереження парності суми ступенів). З Леми 3.2, для малого розрізу необхідно $|S| > k$. При $n = 2k + 2$,

єдиний варіант $|S| = k + 1$, але це число парне (бо k непарне). Суперечність. Тому мінімальний розріз — 2.

Випадок 4: Непарне k та дуже великі графи ($n > 2k + 2$). Тут можна обрати $|S|$ так, щоб задовольнити умови парності та уникнути обмежень Лема 3.2. Можна побудувати міст між двома майже регулярними графами.

Конструктивна побудова та перелік екстремальних графів. Далі деталізовано процедуру генерації графів, що реалізують нижні межі. Метод злиття зі зміною ребер дозволяє створювати графи з розрізом розміру 2.

Алгоритм 1 (Генерація 2-зв'язного регулярного графа).

1. *Вхід:* Парне k , $n \geq 2k + 2$.
2. *Базові компоненти:* Згенеруйте випадкові k -регулярні графи G_1 та G_2 .
3. *Операція Swap:* Видалити $e_1 \in G_1, e_2 \in G_2$ та додати перехресні ребра між кінцями.
4. *Результат:* Пара нових ребер утворює глобальний мінімальний розріз.

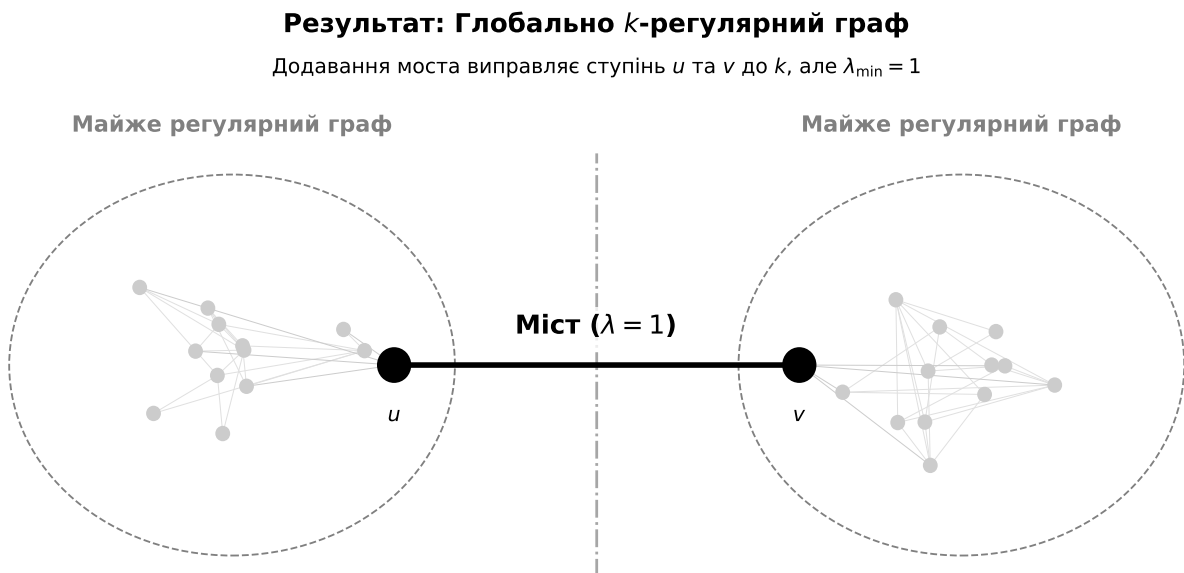


Рис. 2.22: Екстремальний граф для $n > 2k + 2$ та непарного k .

Унікальність структури для $n = 2k + 2$ (Непарне k). Екстремальний граф

для цього випадку є унікальним з точністю до ізоморфізму. Він складається з двох повних графів K_{k+1} , у кожному з яких видалено по одному ребру, і кінці цих ребер з'єднані навхрест.

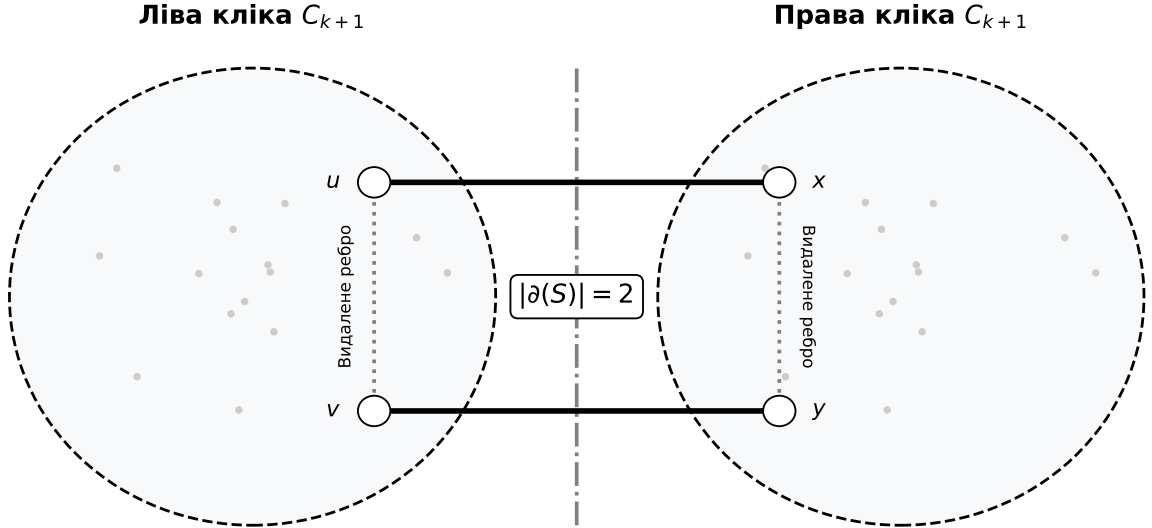


Рис. 2.23: Екстремальний граф для $n = 2k + 2$ та непарного k .

Гіпотеза 5. Нехай $G = C_n^S$ — зв'язний циркулянтний граф ступеня k . Тоді його реберна зв'язність $\lambda(G)$ завжди дорівнює k .

Експериментальна верифікація на вибірці з понад 10^6 графів не виявила жодного контрприкладу. Результати перевірки розміщені на GitHub [133].

Запропоновано евристичний алгоритм для підвищення надійності мережі: ідентифікувати компоненти розрізу та виконати серію операцій Swar між ребрами, що лежать глибоко всередині компонент, та ребрами розрізу.

Наші екстремальні графи (два кластери, з'єднані малою кількістю ребер) будуть мати алгебраїчну зв'язність $\mu_2 \approx 0$. Запропоновано використовувати спектральний моніторинг як ранню систему попередження для мережевих адміністраторів.

Таблиця 2.1

Зведена таблиця екстремальних значень зв'язності

Ступінь k	Порядок n	$\lambda_{\min}$
Парний	$k < n < 2k + 2$	k
Парний	$n \geq 2k + 2$	2
Непарний	$k < n < 2k + 2$	k
Непарний	$n = 2k + 2$	2
Непарний	$n > 2k + 2$	1

2.7. Висновки до розділу 2

У розділі здійснено комплексне дослідження алгоритмічних та комбінаторних властивостей фундаментальних дискретних структур. Отримані результати дозволили суттєво покращити часові оцінки для низки класичних задач та встановити нові теоретичні межі для графів і числових послідовностей.

Основні наукові та практичні здобутки розділу полягають у наступному:

1. Доведено, що потужність множини XOR-суми двох відрізків довжини k обмежена зверху величиною $5(k - 2)$ для всіх $k \geq 5$. Цей фундаментальний результат дозволив здійснити редукцію складності задачі з квадратичної до лінійної. Отримана оцінка надає суворі гарантії щодо обмеженості простору значень у системах, що використовують інтервальні XOR-операції, а запропонований метод доведення на основі бітової рекурсії демонструє ефективність поєднання експериментального аналізу з математичною індукцією.
2. Для задачі конкатенації лексикографічно максимальних підрядків формалізовано її зведення до послідовного вибору суфіксів. Розроблено алгоритм із часовою складністю $O(n)$, що базується на використанні суфіксного масиву та монотонного стеку. Теоретичний аналіз та порівняння з наївними методами підтверджують значну перевагу запропонованого

підходу.

3. Встановлено алгоритмічну еквівалентність між задачами 1D k -TSP та $(\min, +)$ згорткою, що дозволяє переносити результати оптимізації між цими доменами. Обґрунтовано гіпотезу про відсутність алгоритму зі складністю $O(n^{2-\varepsilon})$ для 1D all k -TSP. Показано, що для практично важливого випадку опуклих даних складність зменшується до лінійної, попри технічні виклики, пов'язані з розширенням діапазону значень до $O(nW)$.
4. Представлено лінійний алгоритм підрахунку знакозмінних шляхів у деревах з мітками, що використовує декомпозицію за найменшим спільним предком. Продемонстровано універсальність техніки об'єднання “від меншого до більшого”, яка дозволяє звести квадратичну складність до лінійної у задачах підрахунку шляхів з обмеженнями.
5. Виведено точні формули для мінімальної реберної зв'язності $\lambda_{\min}$ регулярних графів та систематизовано спектр “найгірших” топологій. На основі отриманих меж надано рекомендації щодо використання циркулянтних топологій для підвищення надійності критичних систем.

Розроблений у цьому розділі математичний апарат та алгоритмічні конструкції складають теоретичне підґрунтя для розв'язання прикладних задач у подальших частинах роботи:

- Ефективні методи розв'язання k -TSP та аналіз зв'язності графів будуть адаптовані для побудови алгоритмів керування роєм дронів у третьому розділі.
- Алгоритми швидкої обробки рядків та інтервальних оцінок (XOR-суми) знайдуть застосування при моделюванні стратегій на фінансових ринках у четвертому розділі.

РОЗДІЛ 3

МАРШРУТИЗАЦІЯ ТА КЛАСТЕРИЗАЦІЯ БПЛА

3.1. Вступ

3.1.1. Маршрутизація літальних апаратів. У розділі розглядається проблема оптимізації траєкторії групи безпілотних літальних апаратів (БПЛА), яким доручено перехоплення заданого масиву з n рухомих цілей у d -вимірному евклідовому просторі, де $d \in \{2, 3\}$. Наші зусилля зосереджені саме на розв’язанні комбінаторних аспектів цієї задачі, таких як розбиття цілей між переслідувачами та впорядкування цілей. Також ми застосовуємо ряд спрощень, які дозволяють легко розв’язувати локальні задачі переслідування “один-на-один”.

Ця задача є природним варіантом задачі маршрутизації БПЛА (UAVRP) з динамічними депо [121]. Її практичне застосування є критично важливим для військової стратегії, морського патрулювання та логістичних операцій, що вимагають взаємодії з активами під час їх транзиту. Проблема класифікується як NP-важка, оскільки зводиться до класичної задачі комівояжера (TSP) у граничному випадку нульових швидкостей цілей.

Особливу складність вносить нелокальність функції вартості: зміна порядку відвідування цілей нелінійно трансформує часові ряди прибуття та геометрію подальших сегментів маршруту, що робить конвенційні евристики локального пошуку (наприклад, 2-opt) неефективними. У розділі формалізовано концепцію *впорядковуючої сили* цільової функції та доведено теорему про обмеженість простору допустимих значень довжин маршрутів розмірністю конфігураційного простору задачі. Наведено формалізований приклад системи з трьох цілей, що демонструє виникнення нетранзитивних (циклічних) переваг у виборі маршруту, що фундаментально відрізняє динамічну задачу від статичної.

Класична задача комівояжера та її узагальнення — задача маршрутизації транспортних засобів — є фундаментальними проблемами комбінаторної оптимізації. У класичній постановці задано набір статичних вузлів, і метою є знаходження перестановки $P = (p_1, p_2, \dots, p_n)$, що мінімізує сумарну вартість переміщення, яка зазвичай виражається через евклідову відстань або час подорожі.

Проте сучасні сценарії застосування безпілотних систем, описані Горбуліним, Гуляницьким та Сергієнком [121], виходять за межі статичних графів. Задача маршрутизації БПЛА з рухомими цілями (англ. *moving target TSP*, MT-TSP) вимагає не просто відвідування фіксованих координат, а просторово-часового перехоплення. Це означає, що координати цілі i є функцією часу $x_i(t)$. Відповідно, вартість ребра графа c_{ij} перестає бути константою і стає функцією $c_{ij}(t_{\text{arrival at } i})$, що залежить від моменту завершення попереднього етапу місії. Ця залежність породжує ефект пам'яті у цільовій функції: рішення, прийняте на початку маршруту, впливає на положення всіх наступних цілей у момент їхнього майбутнього перехоплення. Це явище, яке ми називаємо нелокальністю функції вартості, унеможливорює пряме застосування класичних методів декомпозиції та локального пошуку.

У статичній TSP широко застосовується евристика 2-opt, запропонована Croes [88]. Вона базується на локальному покращенні маршруту шляхом усунення перетинів ребер. Наприклад, для маршруту $A \rightarrow B \rightarrow C \rightarrow D$, якщо сума довжин $|AB| + |CD| > |AC| + |BD|$, то вигідніше змінити порядок на $A \rightarrow C \rightarrow B \rightarrow D$.

У контексті рухомих цілей цей принцип не працює. Заміна ребра (A, B) на (A, C) змінює час прибуття в точку C , а отже і її координати. Це, в свою чергу, змінює час вильоту з C до B , що змінює координати B у момент перехоплення. Отже, локальна зміна топології маршруту викликає глобальні збурення всієї траєкторії, що вимагає повного перерахунку розкладу для кожної ітерації оптимізації.

3.1.2. Кластеризація рухомих об'єктів. У підрозділі представлено новий метод кластеризації об'єктів, що рухаються лінійно, розроблений для підвищення ефективності розв'язання задач маршрутизації. Метод базується на модифікації існуючого алгоритму k -медоїдів із використанням альтернативного поняття близькості рухомих об'єктів, що враховує їхню просторово-часову динаміку.

Обговорюється застосування розробленого підходу в дворівневому алгоритмі для задачі маршрутизації групи безпілотних літальних апаратів (БПЛА), метою яких є перехоплення декількох динамічних цілей. Проведено порівняльний аналіз різних стратегій кластеризації на основі існуючого набору даних випадково згенерованих сценаріїв. Чисельні експерименти, наведені у статті, встановлюють практичні переваги запропонованої стратегії кластеризації над класичним методом k -середніх.

Сучасна логістика та управління транспортними потоками стикаються з численними викликами, які можуть бути формалізовані як складні задачі маршрутизації. Такі задачі часто включають обмежену або багатокритеріальну оптимізацію і, отже, можуть слугувати універсальною моделлю для різноманітних промислових логістичних завдань, від доставки вантажів до управління міським трафіком.

Проте, навіть найновіші дослідження в галузі задач маршрутизації, такі як роботи Zhou, що стосуються модульних транспортних засобів [134] або врахування викидів вуглецю в контексті екологічної стійкості у роботах Lou [135], зосереджуються переважно на стаціонарних цілях. Ці обмеження, хоч і спрощують математичне моделювання, надаючи дослідникам прямий доступ до великого інструментарію класичної комбінаторної оптимізації, також накладають суттєві обмеження на практичну застосовність отриманих результатів у складніших і більш динамічних сценаріях.

На відміну від комерційних областей логістики, де статичність є допустимим спрощенням, оборонна промисловість та сектори реагування на надзвичайні си-

туації вже давно зіткнулися з практичною важливістю врахування плинних, високодинамічних та невизначених ситуацій. Теорія диференціальних ігор була розроблена саме для врахування складної динаміки цілей, де кожен агент системи може змінювати свою поведінку у відповідь на дії інших.

На перетині цих напрямків досліджень лежить задача маршрутизації групи БПЛА для перехоплення групи динамічних цілей. Як задача типу “багато-до-багатьох”. Вона має очевидні комбінаторні елементи, як-то розподіл цілей між перехоплювачами і визначення порядку обслуговування. Саме на них зосереджується ця робота. Водночас, на останньому етапі “один-на-один” доречно використати високоефективні інструменти переслідування та перехоплення з теорії конфліктно-керованих процесів, розроблені в [46].

Як динамічна задача, де положення цілей змінюється безперервно, багато існуючих комбінаторних методів маршрутизації не можуть бути застосовані до неї безпосередньо без суттєвої модифікації або дискретизації часу, що може призвести до втрати точності.

Раніше ми запропонували новий дворівневий алгоритм для UAVRP з лінійно рухомими цілями. Початковий крок складався з кластеризації цілей для зведення вихідної задачі великої розмірності до кількох менших задач типу “один-на-багатьох”. Цей крок декомпозиції довів свою ефективність за певних умов, зокрема коли цілі утворювали щільні, повільно змінні групи. Однак він не зміг покращити евристику найближчого сусіда в загальному випадку, особливо коли цілі мали високі відносні швидкості.

Причиною такої поведінки був невідповідний метод кластеризації, який ігнорував вектори швидкості цілей та їхнє прогнозоване положення у часі, спираючись виключно на миттєвий знімок системи. У цьому розділі ми вдосконалили метод кластеризації, щоб він явно враховував положення цілей у часі. Ми відтворили чисельні експерименти та провели детальне порівняння наївної кластеризації за початковою позицією з покращеною версією, яка використовує зважений інтеграл відстані як альтернативну міру близькості цілей.

3.2. Основна частина

3.2.1. Впорядковуюча сила. Спершу розглянемо один БПЛА. Нехай задано множину його цілей $T = \{1, \dots, n\}$ та початкову позицію БПЛА p_0 . Стан i -ї цілі в момент часу t описується вектором $p_i(t) \in \mathbb{R}^d$. Для лінійного руху $p_i(t) = p_{i,0} + v_i t$, де v_i — вектор швидкості. БПЛА має максимальну швидкість $V_{\max}$. Задача полягає у знаходженні перестановки $\sigma \in S_n$ та послідовності моментів часу $t_1, \dots, t_n$, таких що $t_k \geq t_{k-1}$ та:

$$\|p_{\sigma(k)}(t_k) - p_{\sigma(k-1)}(t_{k-1})\| \leq V_{\max}(t_k - t_{k-1}) \quad (3.2.1)$$

з мінімізацією функціоналу t_n (англ. *makespan*) або сумарної пройденої відстані. Для лінійного випадку час перехоплення Δt між точкою вильоту A (в момент t_{start}) і ціллю B визначається з квадратного рівняння:

$$\|(p_{B,0} + v_B(t_{start} + \Delta t)) - A\|^2 = (V_{\max} \Delta t)^2 \quad (3.2.2)$$

Рівняння ілюструє нелінійну залежність часу перехоплення від параметрів руху.

Одним із ключових теоретичних результатів даного дослідження є аналіз інформаційної ємності простору рішень.

Означення 22. Нехай S — множина допустимих екземплярів задачі, а $C = \{c_1, \dots, c_n\}$ — множина всіх можливих перестановок (маршрутів). Визначимо відображення $g : S \rightarrow P_n!$, де $P_n!$ — множина перестановок індексів розв'язків, таке що $g(x)$ впорядковує розв'язки за зростанням вартості цільової функції. Величину $\log |g(S)|$ назвемо *впорядковуючою силою* функції f .

Розглянемо відображення $g : S \rightarrow \mathbb{R}^{n!}$, де $g(x) = (L_1(x), \dots, L_{n!}(x))$ — вектор довжин усіх можливих маршрутів.

Теорема 11. Для всіх $n > \Gamma^{-1}(2d)$, де Γ — гамма-функція, існує комбінація з $n!$ дійсних чисел, яка не може бути реалізована як набір довжин маршрутів для жодного екземпляра евклідової задачі MT-TSP.

Доведення. Екземпляр з n лінійно рухомими цілями у d -вимірному просторі повністю визначається початковими координатами та векторами швидкостей

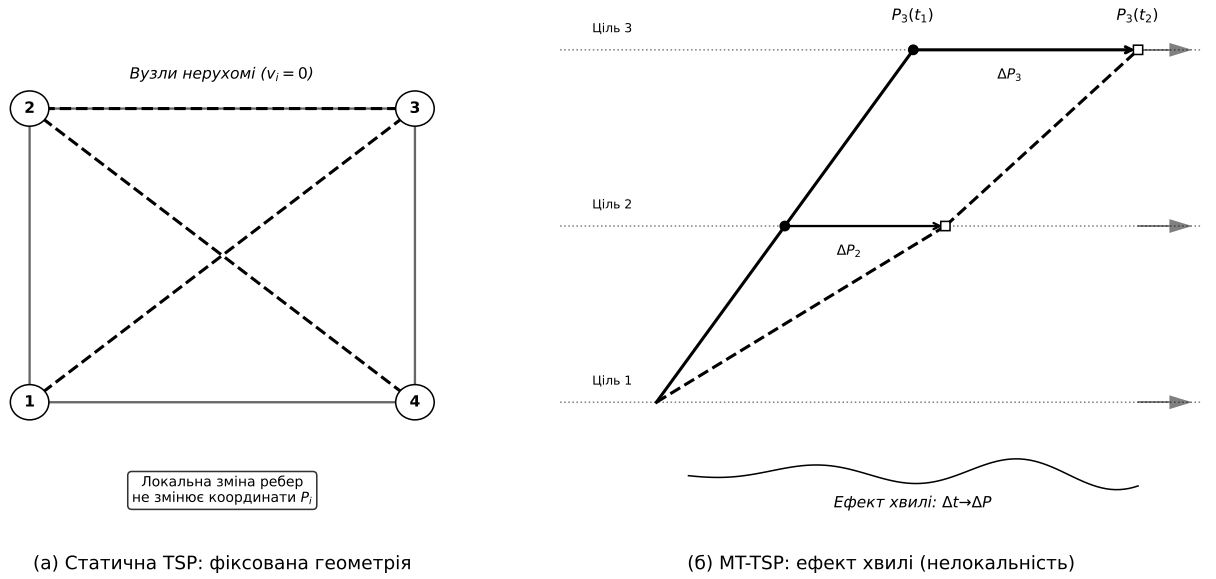


Рис. 3.1: Схематична ілюстрація ефекту хвилі: зміна часу прибуття до цілі впливає на її позицію в момент перехоплення, що змінює вартість наступного сегмента маршруту.

кожної цілі. Це дає $2d$ параметрів на одну ціль, отже розмірність простору вхідних даних $\dim(S) = 2dn$. Довжина конкретного маршруту є неперервною функцією від цих $2dn$ змінних. Таким чином, ми маємо відображення з простору $\mathbb{R}^{2dn}$ у простір $\mathbb{R}^{n!}$.

Оскільки факторіал зростає значно швидше за лінійну функцію ($n! \gg 2dn$ для великих n), то згідно з теоремою про інваріантність області та властивостями розмірності Хаусдорфа, образ цього відображення є многовидом розмірності щонайбільше $2dn$ у просторі розмірності $n!$. Це означає, що “майже всі” точки у $\mathbb{R}^{n!}$ є недосяжними.

Іншими словами, структура евклідового простору накладає жорсткі алгебраїчні залежності між довжинами різних маршрутів. Наприклад, неможливо незалежно змінювати довжину одного маршруту, не впливаючи на довжини інших, що мають спільні ребра. Вираз $\Gamma^{-1}(2d)$ у формулюванні теореми є апроксимацією точки перетину функцій $n!$ та $2dn$, показуючи, що навіть для відносно малих n задача є перевизначеною топологічними обмеженнями. $\square$

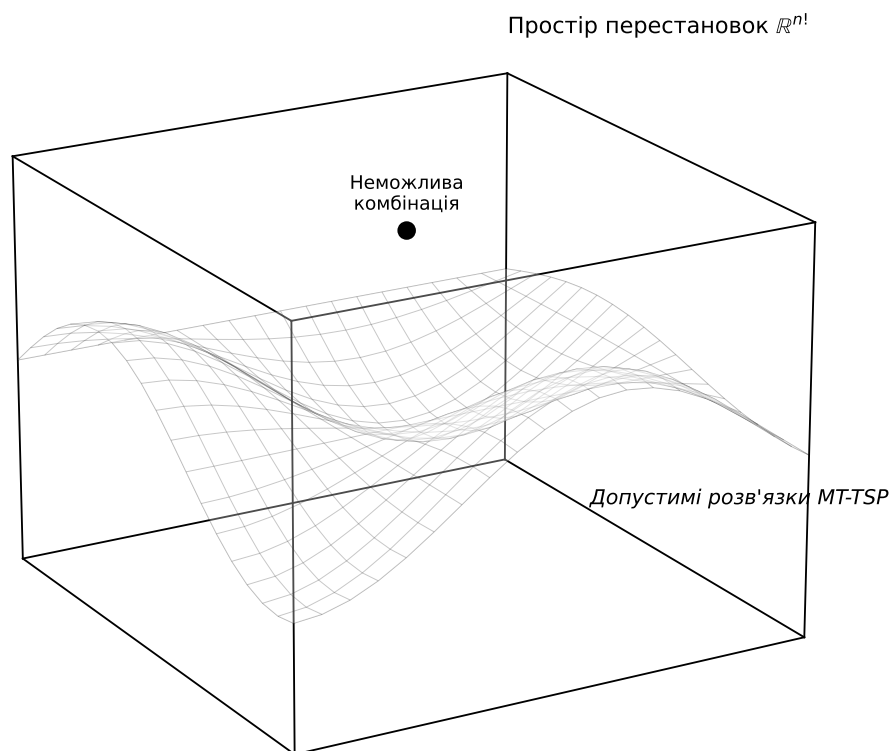


Рис. 3.2: Схематична ілюстрація: простір можливих довжин маршрутів утворює многовид низької розмірності у високовимірному просторі всіх перестановок.

Цей результат є фундаментальним, оскільки він вказує на те, що простір пошуку MT-TSP, попри свою величезну комбінаторну складність, має внутрішню структуру низької розмірності, яку потенційно можна використовувати для побудови ефективних алгоритмів.

3.2.2. Формалізація циклічних переваг та неможливість задання повного порядку. Однією з найбільш контрінтуїтивних властивостей MT-TSP є порушення транзитивності переваг, що унеможливорює просте сортування цілей за близькістю.

Розглянемо систему з трьох цілей A, B, C , що знаходяться у вершинах рівно-

стороннього трикутника зі стороною L , зображену на рис. 3.3. Нехай у момент часу $t = 0$ цілі починають рух уздовж сторін трикутника проти годинникової стрілки зі швидкістю v . БПЛА знаходиться в центрі трикутника. Позначимо $f(X \rightarrow Y)$ як час, необхідний для перехоплення цілі X , а потім цілі Y .

Аналіз:

1. Якщо БПЛА спочатку летить до A , він витрачає час Δt_1 . За цей час цілі B зміщується вздовж сторони BC , віддаляючись від вершини B , але потенційно наближаючись до точки перехоплення, якщо БПЛА рухатиметься від A .
2. Однак, критичним є взаємне розташування. Рух цілей створює ситуацію, аналогічну грі “камінь-ножиці-папір”.
 - Перехоплення A робить B зручнішою наступною ціллю (вона рухається назустріч або під зручним кутом).
 - Але якщо почати з B , то C стає зручнішою.
 - Якщо почати з C , то A стає зручнішою.

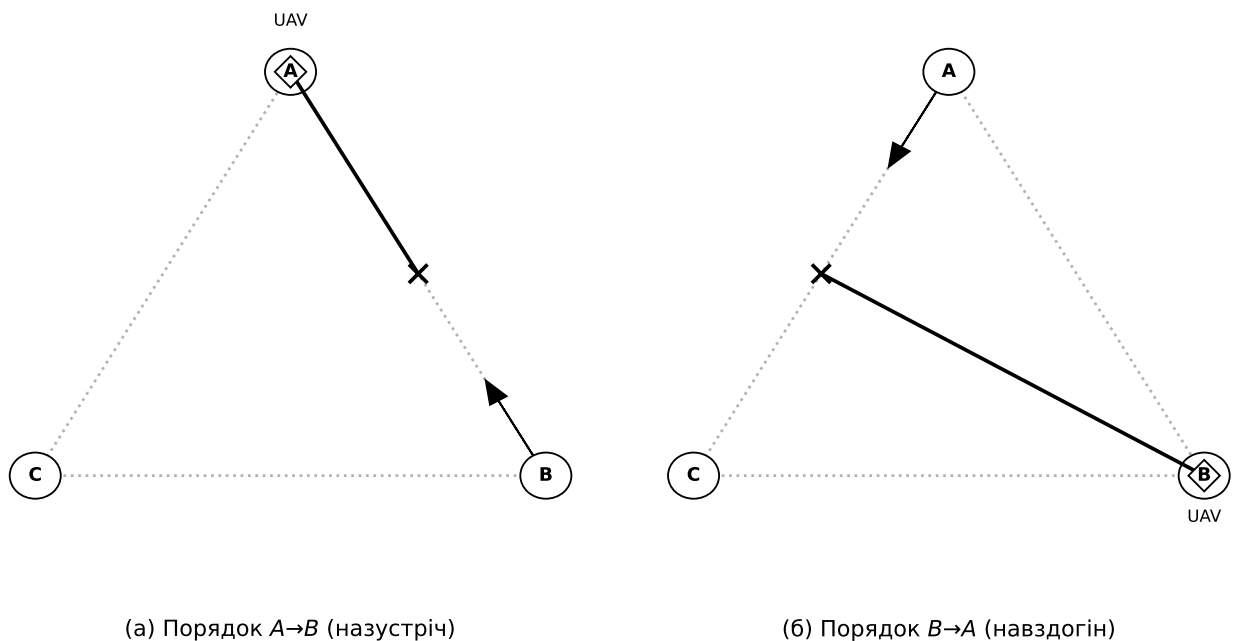


Рис. 3.3: Ілюстрація трикутного парадоксу: рух цілей створює циклічні переваги у виборі порядку перехоплення.

Це призводить до циклічної нерівності вартостей маршрутів:

$$t_n(A \rightarrow B) < t_n(B \rightarrow A)$$

$$t_n(B \rightarrow C) < t_n(C \rightarrow B)$$

$$t_n(C \rightarrow A) < t_n(A \rightarrow C)$$

Більше того, у повному циклі може виникнути ситуація:

$$t_n(A \rightarrow B \rightarrow C \rightarrow A) < t_n(A \rightarrow C \rightarrow B \rightarrow A) \quad (3.2.3)$$

Це означає, що порядок обходу $A \prec B \prec C \prec A$ є оптимальним у локальному сенсі (між парами), але глобально створює цикл. Це доводить неможливість задання лінійного порядку на множині рухомих цілей, який би коректно відображав їх вартість для маршрутизації.

3.2.3. Кластеризація та TLGA. Враховуючи описані вище труднощі, застосування точних методів обмежене малими розмірностями. Для знаходження розв’язків задач практичного масштабу доцільно використовувати багаторівневі підходи.

Дворівневий генетичний алгоритм (англ. *two-level genetic algorithm*, TLGA): Як зазначено у Горбуліна та ін. [121] та підтверджено Ding [108], ефективним є підхід “розділяй і володарюй”:

1. Кластеризація: групування рухомих об’єктів. Оскільки об’єкти рухаються, класичний k-means модифікується для врахування векторів швидкості. Liu та ін. запропонували об’єкти з подібними векторами швидкості та просторовою близькістю об’єднуються в кластери [136].
2. Глобальна маршрутизація: знаходження оптимального порядку відвідування кластерів. Одним з підходів тут є наближене розв’язання TSP, де вузлами є центроїди груп.
3. Локальна маршрутизація: побудова Гамільтонового шляху всередині кожного кластера. Тут вже можна врахувати точні кінематичні рівняння,

оскільки всередині кластера відстані менші, і похибки лінеаризації менш критичні.

3.2.4. Близькість рухомих об'єктів. Ми моделюємо цілі нашої задачі маршрутизації як кортежі їхніх початкових позицій $z_i^{(0)} = (x_i^{(0)}, y_i^{(0)})$ та векторів швидкості $\dot{z}_i = c_i \cdot w_i$, де c_i представляє скалярну швидкість цілі, а $w_i = (u_i, v_i)$ — одиничний вектор, що задає напрямок руху. Тут $1 \leq i \leq n$, де n позначає загальну кількість цілей у системі. Припускається, що рух є рівномірним і прямолінійним на розглядуваному часовому горизонті, що є стандартним припущенням для короткострокового тактичного планування.

У нашій попередній роботі ми застосували стандартний алгоритм кластеризації k -середніх. Метод вирішує задачу оптимізації мінімізації суми квадратів відстаней всередині кластера (англ. *within-cluster sum of squares*, WCSS). Формально, маючи набір об'єктів $x_1, x_2, \dots, x_n$, необхідно розбити їх на k неперетинних кластерів $S_1, S_2, \dots, S_k$ з метою мінімізації функціоналу:

$$\sum_{i=1}^k \sum_{x \in S_i} |x - m_i|^2 \rightarrow \min, \quad (3.2.4)$$

де m_i — центроїд (середнє геометричне) кластера i .

На практиці використання цього методу вимагає, щоб базовий простір, з якого походять об'єкти $x_1, x_2, \dots, x_n$, був d -вимірним евклідовим простором для деякого натурального d . В іншому випадку визначення центроїда m_i у рівнянні (3.2.4) стає нетривіальним завданням. У просторі траєкторій, де об'єктами є функції часу, поняття “середнього” може не мати фізичного змісту або призводити до траєкторій, які неможливо реалізувати (наприклад, усереднення двох траєкторій, що розходяться в протилежні боки, дасть нерухому точку, що не відповідає динаміці жодної з цілей).

Раніше ми розглядали лише часткову кластеризацію цілей за їхніми початковими позиціями $z_i^{(0)}$. Це інтуїтивно зрозумілий підхід, оскільки початкова позиція є первинною евклідовою компонентою опису цілі. Однак, наші чисельні

експерименти переконливо показали, що такий підхід не дає прийнятних результатів у певних практичних сценаріях, особливо коли швидкості об'єктів є значними порівняно з відстанню між ними.

Тому виникає необхідність інтеграції векторів швидкості у процес поперечної обробки та кластеризації. Вектор швидкості також представляється парю дійсних чисел, тому може виникнути спокуса запустити алгоритм k -means у $2d$ -вимірному евклідовому просторі (у нашому випадку $d = 4$, простір станів $\mathbb{R}^4$). Однак, швидкості додатково обмежені областю $B_{\max(c_i)}$, тобто кулею з центром у початку координат із радіусом, що дорівнює максимальній швидкості будь-якої цілі. Штучне визначення близькості як евклідової відстані між векторами їхніх компонент, вирівняних по осях (координати плюс швидкості), є методологічно некоректним через різну фізичну розмірність величин (відстань вимірюється в метрах, швидкість — у метрах за секунду). Це створює проблему масштабування: зміна одиниць вимірювання часу або відстані може кардинально змінити результати кластеризації.

3.2.4.1. Фундаментальний контрприклад незалежності компонент.

Навіть якщо відкинути проблему розмірності, трактування початкових позицій і швидкостей як незалежних компонент у сенсі:

$$d((z_i, w_i), (z_j, w_j)) = f(d_z(z_i, z_j), d_w(w_i, w_j)) \quad (3.2.5)$$

неминуче призведе до логічних суперечностей незалежно від вибору функції агрегації f та метрик d_z і d_w . Більше того, це твердження справедливе навіть у одновимірному просторі $\mathbb{R}^1$, не кажучи вже про довільний $\mathbb{R}^d$ з $d > 1$.

Для ілюстрації розглянемо дві цілі $(z_i, w_i) = (0, 1)$ та $(z_j, w_j) = (1, -1)$. У даному випадку цілі знаходяться на відстані 1 одиниця одна від одної і рухаються назустріч зі швидкістю 1. Це означає, що через час $t = 0.5$ вони зустрінуться в точці 0.5. Така конфігурація надає ідеальну можливість для перехоплення обох цілей одним агентом за короткий проміжок часу.

Тепер розглянемо іншу пару цілей $(z_k, w_k) = (0, -1)$ та $(z_l, w_l) = (1, 1)$. Геометрично початкові позиції ті самі (0 і 1), і вектори швидкостей мають ті самі модулі, але протилежні напрямки розходження. Вони віддаляються одна від одної, і відстань між ними з часом лише зростає ($D(t) = 1 + 2t$). Очевидно, що стратегія перехоплення для другої пари має бути принципово іншою.

Незважаючи на цю очевидну семантичну та тактичну різницю, формальне застосування метрики (3.2.5) дає ідентичні результати:

$$\begin{aligned}
 d((z_i, w_i), (z_j, w_j)) &= f(d_z(z_i, z_j), d_w(w_i, w_j)) \\
 &= f(d_z(0, 1), d_w(1, -1)) \\
 &= f(d_z(0, 1), d_w(-1, 1)) \quad (\text{в силу симетрії метрики}) \\
 &= f(d_z(z_k, z_l), d_w(w_k, w_l)) \\
 &= d((z_k, w_k), (z_l, w_l)).
 \end{aligned} \tag{3.2.6}$$

Тут ми неодноразово використовували властивість симетрії, якою повинна володіти будь-яка коректна метрика d_w . Цей простий контрприклад чітко демонструє, що надзвичайно важко — якщо взагалі можливо — підібрати такий вибір функцій f , d_z та d_w , який зробив би результуючу метрику d значущою для динамічних задач.

(а) Рух назустріч

(б) Рух віддаляючись

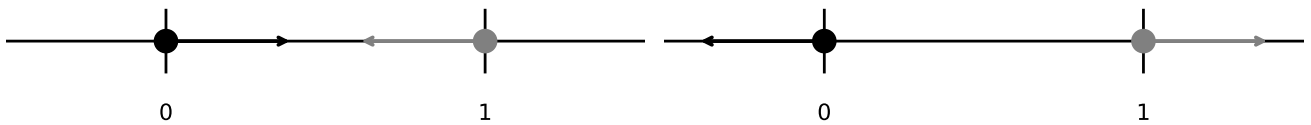


Рис. 3.4: Приклад демонструє відсутність семантично змістовної метрики

3.2.5. Інтегральна метрика близькості. Зазначений вище контрприклад дає підказку для вирішення проблеми. А саме, пара цілей (i, j) у сцена-

рії зближення стає ідентичною парі (k, l) через певний проміжок часу, якщо розглядати їх миттєві положення. Це спостереження спонукає нас визначити близькість рухомих цілей не як статистику, а як середню евклідову відстань між ними протягом певного часового інтервалу. Оскільки часовий горизонт є безперервним, природно визначити це середнє через визначений інтеграл за часом:

$$d(i, j) = \int_0^T d_z(z_i(t), z_j(t)) dt. \quad (3.2.7)$$

Таке пряме визначення можливе за умови відповідного вибору T . Однак, завдання апріорного вибору T є досить складним. Більше того, для розбіжних траєкторій відстань зростає лінійно з часом, тому інтеграл буде домінуватися значеннями на кінці інтервалу. Тому доцільно ввести вагову функцію, яка б надавала більшої значущості близькості на ранніх етапах. Це формулюється математично як невластний інтеграл:

$$d(i, j) = \int_0^\infty d_z(z_i(t), z_j(t)) g(t) dt, \quad (3.2.8)$$

де $g(t) \geq 0$ — вагова функція, що відображає відносну вагу моменту часу t .

Нижче ми наводимо суворе математичне доведення, яке встановлює необхідні та достатні умови для вагової функції $g(t)$. Нехай рух двох довільних цілей i та j описується рівняннями лінійної динаміки:

$$z_i(t) = z_i^{(0)} + w_i t, \quad z_j(t) = z_j^{(0)} + w_j t,$$

де $w_i = c_i w_i$ — вектори швидкості. Миттєва відстань між ними:

$$D(t) = d_z(z_i(t), z_j(t)) = \|(z_i^{(0)} - z_j^{(0)}) + (w_i - w_j)t\|.$$

Введемо позначення: $\Delta z = z_i^{(0)} - z_j^{(0)}$ та $\Delta w = w_i - w_j$. Тоді $D(t) = \|\Delta z + \Delta w t\|$.

Лема 6 (про лінійну оцінку росту). *Для будь-яких двох лінійно рухомих об'єктів функція відстані $D(t)$ має лінійний порядок зростання, тобто існують такі додані константи $C_{\min}, C_{\max} > 0$ та час T_0 , що для всіх $t > T_0$:*

$$C_{\min} t \leq D(t) \leq C_{\max} t, \quad \text{якщо } \|\Delta w\| \neq 0.$$

Доведення. Верхня оцінка впливає безпосередньо з нерівності трикутника:

$$D(t) = \|\Delta z + \Delta w t\| \leq \|\Delta z\| + \|\Delta w\|t.$$

При $t \rightarrow \infty$, доданок з t домінує. Можемо вибрати $C_{\max} = \|\Delta w\| + \varepsilon$. Для нижньої оцінки використаємо зворотну нерівність трикутника:

$$D(t) = \|\Delta w t + \Delta z\| \geq \left| \|\Delta w\|t - \|\Delta z\| \right|.$$

Для $t > \frac{\|\Delta z\|}{\|\Delta w\|}$, вираз під модулем є додатним. При великих t , $D(t) \approx \|\Delta w\|t$. $\square$

Теорема 12 (про збіжність). *Інтеграл $I = \int_0^\infty D(t)g(t)dt$ збігається тоді і лише тоді, коли збігаються моменти нульового та першого порядку вагової функції:*

$$\int_0^\infty g(t)dt < \infty \quad \wedge \quad \int_0^\infty t \cdot g(t)dt < \infty.$$

Доведення. Розглянемо вираз для $D(t)$.

$$I = \int_0^\infty \|\Delta z + \Delta w t\|g(t)dt.$$

1. Достатність. Нехай $\int_0^\infty tg(t)dt < \infty$ та $\int_0^\infty g(t)dt < \infty$. Використовуючи верхню оцінку з леми 6 ($D(t) \leq \|\Delta z\| + \|\Delta w\|t$):

$$I \leq \int_0^\infty (\|\Delta z\| + \|\Delta w\|t)g(t)dt = \|\Delta z\| \int_0^\infty g(t)dt + \|\Delta w\| \int_0^\infty tg(t)dt.$$

Оскільки обидва інтеграли у правій частині збіжні, то I також збігається.

2. Необхідність. Нехай $\Delta w \neq 0$. Використовуючи нижню оцінку

$$D(t) \geq \|\Delta w\|t - \|\Delta z\| \tag{3.2.9}$$

для великих $t > T^*$:

$$\int_{T^*}^\infty D(t)g(t)dt \geq \|\Delta w\| \int_{T^*}^\infty tg(t)dt - \|\Delta z\| \int_{T^*}^\infty g(t)dt.$$

Якщо інтеграл I збігається, то це вимагає збіжності домінуючого члена

$$\|\Delta w\| \int_{T^*}^\infty tg(t)dt. \tag{3.2.10}$$

$\square$

Наслідок 2. Функція $g(t)$ повинна спадати на нескінченності швидше, ніж $1/t^2$. Ми пропонуємо використовувати

$$g_1(t) = \frac{1}{(t+1)^3} \quad (3.2.11)$$

або

$$g_2(t) = \frac{1}{(t+1)^2 \log^2(t+1)}. \quad (3.2.12)$$

3.2.6. Алгоритми кластеризації на основі інтегральної метрики.

Використовуючи введене визначення метрики, ми можемо обчислити матрицю попарних відстаней $A = \{a_{i,j}\}$, де $a_{i,j} = d(i,j)$.

Оптимальним вибором для метричних просторів, де визначена лише відстань, є метод k -медоїдів. Він вирішує задачу мінімізації суми відмінностей всередині кластера:

$$\sum_{i=1}^k \sum_{x \in S_i} d(x, x_{c_i}) \rightarrow \min, \quad (3.2.13)$$

де $x_{c_i} \in S_i$ — обраний представник кластера i .

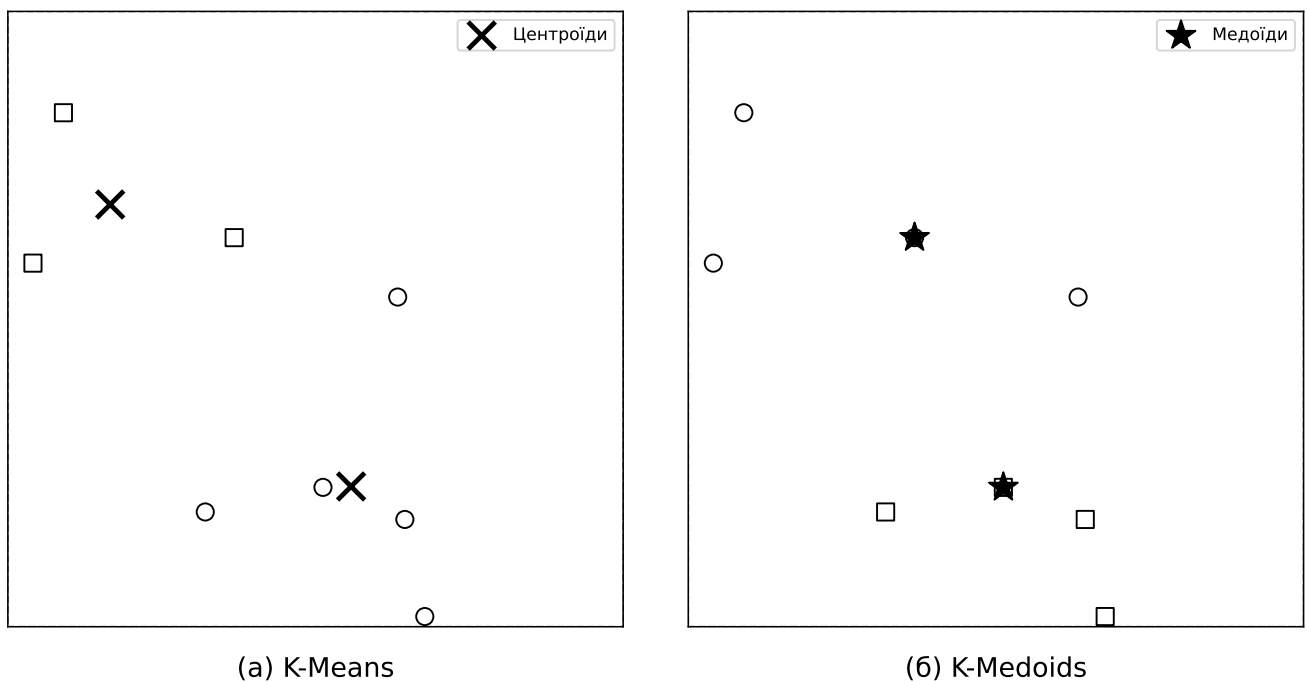


Рис. 3.5: Порівняння кластеризації k -середніх та k -медоїдів

3.2.7. Чисельні експерименти та результати. Опишемо методологію проведення обчислювальних експериментів. Всього було проведено чотири експерименти. Кожен з них складався з 30 сценаріїв (наборів вхідних даних). Отримані результати у кожному експерименті групували та візуалізували кількома способами.

У кожному з сценаріїв частина даних була сталою, а решта обиралися випадково з певного ймовірнісного розподілу. В усіх експериментах: $n = 30$, $m = 4$, $c_i \in [5, 25]$, $c'_j = 30$. Для відтворюваності експериментів, ми опублікували використані генератори випадкових чисел та отримані набори даних на GitHub [137].

Порівняно три стратегії: без кластеризації, k -середніх та k -медоїдів.

У першому сценарії початкові позиції та напрямки руху цілей випадкові. Усі розподіли у цьому експерименті рівномірні:

- початкові позиції цілей обираються з $[-1000, 1000] \times [-1000, 1000]$;
- кут руху вибирається з напів-інтервалу $[0, 2\pi)$.

Таблиця 3.1

Результати для випадкових позицій та швидкостей

Стратегія	avg. len	avg. makespan
None	32,774	13,412
k -means	55,654	30,034
k -medoids	38,676	17,746

У другому сценарії початкові позиції скупчені а напрямки руху випадкові:

- обираються m випадкових центрів скупчення зі згаданого квадрата;
- для кожного центра обирається декілька точок з кулі з цим центром та радіусом 150.

У третьому сценарії початкові позиції випадкові а напрямки руху скупчені:

- обираються m випадкових кутів зі згаданого напів-інтервалу;
- для кожного кута обирається декілька напрямків руху розподілу фон Мізеса з цим кутом як центром та параметром $\kappa = 8$.

Результати для кластеризованих позицій

Стратегія	avg. len	avg. makespan
None	30,226	10,936
k -means	35,062	15,973
k -medoids	29,730	14,006

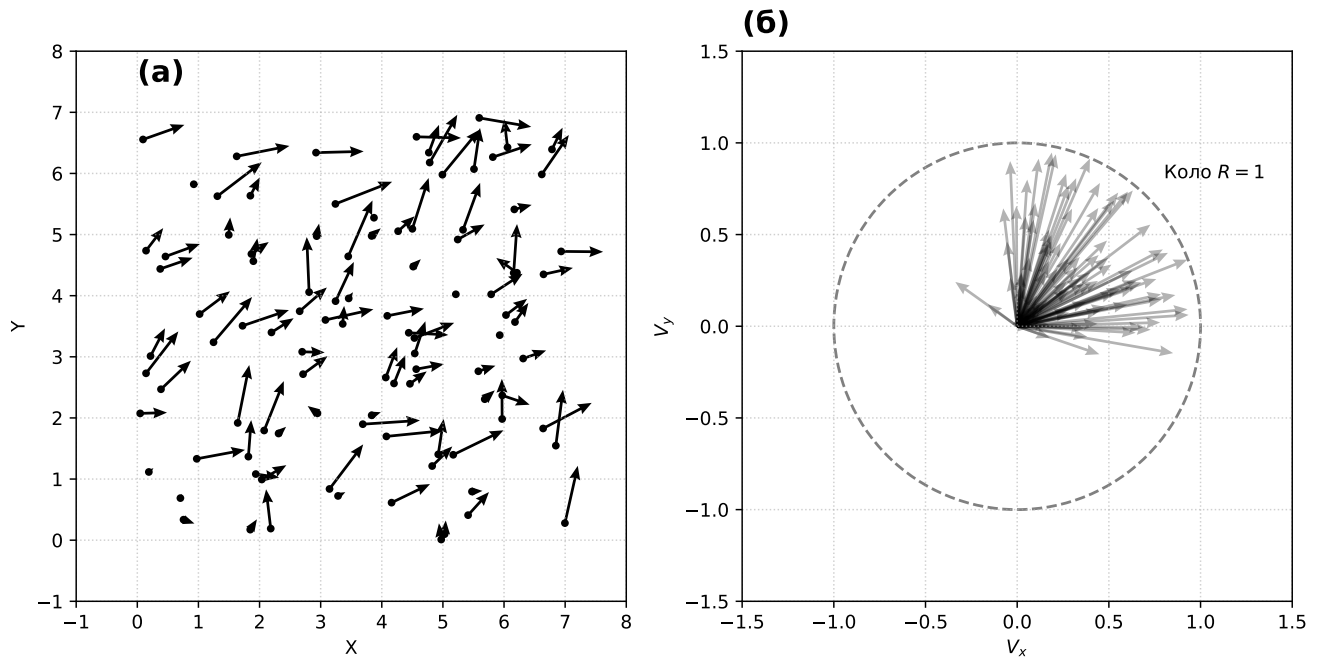


Рис. 3.6: Приклад (нормалізованих) векторів з розподілу фон Мізеса.

3.3. Висновки до розділу 3

Задача маршрутизації БПЛА для перехоплення рухомих цілей є суттєво складнішою за класичні варіації VRP через залежність метрики від часу. Доведено, що простір розв'язків має специфічні топологічні обмеження, а динаміка руху породжує ефекти нетранзитивності переваг. Показано, що врахування нелінійних факторів є критично необхідним для побудови реалістичних планів польоту. Запропоновано ієрархічний підхід із використанням кластеризації який дозволив ефективно знайти наближені рішення для задач великої розмірності.

У розділі всебічно розглянуто проблему кластеризації динамічних об'єктів у

Таблиця 3.3

Результати для кластеризованих швидкостей (роїв)

Стратегія	avg. len	avg. makespan
None	35,701	14,462
k -means	48,818	28,109
k -medoids	34,694	16,239

контексті маршрутизації БПЛА. Доведено, що незалежний розгляд позицій та швидкостей призводить до логічних помилок. Запропоновано нову інтегральну метрику та доведено теорему про її збіжність. Чисельні експерименти підтвердили перевагу методу k -medoids з інтегральною метрикою, особливо в сценаріях зі складною кінематикою цілей.

РОЗДІЛ 4

РОБАСТНА ТОРГІВЛЯ АКЦІЯМИ

4.1. Вступ

У цьому розділі дисертації розглянуто декілька варіантів задачі керування портфелем акцій за різних моделей фінансового ринку. Одразу зазначимо, що ми застосовуємо ряд спрощень для зведення задачі до суто комбінаторної. Втім, навіть одержана спрощена модель може бути застосована на деяких сильно подільних ринках, таких як криптовалюти.

Класична література найчастіше моделює акції за допомогою методів стохастичності. На цьому етапі дослідники стикаються з проблемою вибору моделі реальності, яка була б достатньо простою для аналізу, але водночас відображала б ключові властивості ринку, як-то волатильність та важкі хвости розподілів.

Ціна певної акції в розрізі часу, як правило, моделюється випадковим процесом. Традиційним прикладом є броунівський рух. Він же лежить в основі моделі ціноутворення опціонів Блека—Шоулза. Однак, Bianchi та інші [53] з'ясували, що реальні ринкові дані часто демонструють фрактальні властивості не притаманні стандартними гауссівськими моделями.

Щойно певна модель обрана постає питання передбачення поведінки окремих акцій та ринку в цілому в майбутньому. Прогнозування цін акцій є окремою задачею, основні методи розв'язання якої наразі можна розділити на два великі класи. Класичні економетричні методи запропоновані Mondal та Jarrett конкурують тут із новітніми підходами штучного інтелекту.

Наприклад, Kim та Tsantekidis продемонстрували ефективність нейронних мереж для передбачення цін за історичними даними. Втім, жоден з методів наразі не володіє здатністю точно передбачити точкові значення $p(t)$ цін у час t у

нетривіальних моделях. Натомість усі методи надають або певний ймовірнісний розподіл, або ж довірчі інтервали $[l(t), r(t)]$ з певним рівнем довіри.

Третім етапом є вибір торгової стратегії. Він же і є фокусом цього розділу. Перш за все тут потрібно обрати метрику для оптимізації. Класичною стратегією є максимізація очікуваного прибутку, тобто прибутку у середньому. До такої філософії науки відносяться, зокрема, недавні роботи Wu [60]. Другою стратегією є максимізація прибутку в найгіршому випадку. Вона ліквідує недоліки усереднення, проте створює нові проблеми. Нашою метою є мінімізація так званого жалю відносно оптимального рішення, яке можна було б прийняти в умовах визначеності.

4.2. Основна частина

Основні результати цього підрозділу було опубліковано у статті [2].

4.2.1. Класичні задачі. У цьому підрозділі наведено огляд усіх класичних задач що розглядаються надалі. Проаналізовано їхні схожі риси та основні відмінності. Пояснене практичне значення всіх вхідних параметрів.

Означення 23. Розглядається задача із дискретним часом на *горизонті* в n послідовних днів; *ціна* певної акції у день $1 \leq i \leq n$ позначається p_i .

Вхідними даними кожної класичної задачі є послідовність $p_1, \dots, p_n$ цін в n послідовних днів.

Означення 24. *Транзакцією* називається пара індексів $1 \leq i < j \leq n$. Вона відповідає купівлі акції у день i та продажу в день j .

Означення 25. *Розкладом* I називається послідовність із $k \geq 0$ транзакцій. Для простоти комп'ютерного представлення усі дні записуються підряд: $1 \leq i_1 < i_2 < \dots < i_{2k} \leq n$.

Означення 26. У більшості задач *прибутком транзакції* i, j є значення виразу $\text{profit}(i, j) = p_j - p_i$. *Прибутком розкладу* I є значення знакозмінної

суми:

$$\begin{aligned}\text{profit}(I) &= -p_{i_1} + p_{i_2} - p_{i_3} + p_{i_4} - \dots - p_{i_{2k-1}} + p_{i_{2k}} \\ &= (p_{i_2} - p_{i_1}) + (p_{i_4} - p_{i_3}) + \dots + (p_{i_{2k}} - p_{i_{2k-1}}).\end{aligned}\quad (4.2.1)$$

Зауваження 5. Від'ємні члени відповідають купівлі акцій у дні непарні $i_1, i_3, \dots, i_{2k-1}$ та витраті грошей на це, тоді як додатні члени відповідають продажу акцій у парні дні $i_2, i_4, \dots, i_{2k}$ та отриманню за це грошей.

Вводяться також поняття комісії та охолодження:

Означення 27. Деякі задачі додають комісію f до кожної транзакції, тоді прибутком транзакції $i, j \in$ значення виразу $\text{profit}(i, j; f) = p_j - p_i - f$. Прибуток розкладу I змінюється відповідно:

$$\begin{aligned}\text{profit}(I; f) &= (p_{i_2} - p_{i_1} - f) + (p_{i_4} - p_{i_3} - f) + \dots + (p_{i_{2k}} - p_{i_{2k-1}} - f) \\ &= \text{profit}(i_1, \dots, i_{2k}) - k \cdot f.\end{aligned}\quad (4.2.2)$$

Зауваження 6. Комісія змушує оптимальний розв'язок містити лише високоприбуткові транзакції. Зокрема, вона нагадує метод регуляризації.

Задача з охолодженням накладає дещо суворіші обмеження на I .

Означення 28. Періодом охолодження d називається обмеження вигляду

$$i_{2j+1} > i_{2j} + d \quad \forall j = 1, \dots, k-1. \quad (4.2.3)$$

Зауваження 7. Така зміна відповідає наявності періоду охолодження, що триває d днів, між послідовними операціями продажу та купівлі у цьому порядку. Надалі $d = 1$. Зауважте, що немає необхідності чекати ще один день між послідовними операціями купівлі та продажу.

Метою кожної задачі є максимізація прибутку. Іншими словами, потрібно скласти розклад, що має максимально можливе значення відповідного визначення прибутку.

Зауваження 8. Може існувати кілька розв'язків з однаковим значенням прибутку; у цьому випадку немає необхідності перераховувати їх усі, і будь-який з них вважається допустимим значенням, що повертається.

Тепер ми готові сформулювати самі задачі:

Задача 1 (одна транзакція). За заданою послідовністю $p_1, \dots, p_n$ знайти розклад I з $k \leq 1$ що максимізує значення формули (4.2.1).

Задача 2 (дві транзакції). За заданою послідовністю $p_1, \dots, p_n$ знайти розклад I з $k \leq 2$ що максимізує значення формули (4.2.1).

Задача 3 (k транзакцій). За заданою послідовністю $p_1, \dots, p_n$ та заданим K знайти розклад I із $k \leq K$ що максимізує значення формули (4.2.1).

Задача 4 (безліміт). За заданою послідовністю $p_1, \dots, p_n$ знайти розклад I що максимізує значення формули (4.2.1).

Задача 5 (комісія). За заданою послідовністю $p_1, \dots, p_n$ знайти розклад I що максимізує значення формули (4.2.2).

Задача 6 (охолодження). За заданою послідовністю $p_1, \dots, p_n$ знайти розклад I з умовою формули (4.2.3) що максимізує значення формули (4.2.1).

Зауваження 9. Усі задачі з обмеженою кількістю транзакцій вводять лише обмеження нерівності замість обмежень рівності. Це запобігає включенню до оптимальних розв'язків збиткових транзакцій лише задля виконання обмеження рівності. Зокрема, виконання навіть однієї операції не має практичного сенсу, якщо послідовність p є спадною.

4.2.2. Класичні розв'язки. У цьому підрозділі оглядово наведено стандартні розв'язки всіх класичних задач. Спостереження щодо методів динамічного програмування використані в наступних розділах.

Усі задачі 1–6 можна розв'язати за допомогою методів динамічного програмування. Всі розклади складаються з окремих транзакцій і прибуток від транзакцій у формулі (4.2.1) не залежить від решти розкладу.

Добре відомо, що динамічне програмування є стандартним методом для багатьох задач з оптимальною підструктурою. Воно складається з двох основних частин: визначення станів — компактних представлень підзадач, та переходів

— відношень між значеннями станів. У решті цього підрозділу розглянуть різні види параметрів у межах стану.

Як вже зазначалося, стани всіх підзадач міститимуть день m , що позначає префікс $1, \dots, m$ днів. Повна задача матиме $m = n$.

Спочатку обговорюються стани тримати (hold), продано (sold) та охолодження (cold). Оскільки задачі 4, 6 та 5 не обмежують кількість транзакцій k , інформацію про кількість транзакцій не включається до стану підзадачі. Натомість стан залежить від того, чи утримуються акції в кінці дня m . Задача 6 додатково враховує можливість перебування в періоді охолодження, якщо акції щойно продано попереднього дня $m - 1$. Переходи в цих задачах є такими, де mp означає максимальний прибуток (max profit).

- необмежена кількість транзакцій із задачі 4:

$$\begin{aligned} mp(m, \text{hold}) &= \max(mp(m-1, \text{hold}), mp(m-1, \text{sold}) - p_m); \\ mp(m, \text{sold}) &= \max(mp(m-1, \text{sold}), mp(m-1, \text{hold}) + p_m). \end{aligned} \quad (4.2.4)$$

- комісія за транзакцію із задачі 5:

$$\begin{aligned} mp(m, \text{hold}) &= \max(mp(m-1, \text{hold}), mp(m-1, \text{sold}) - p_m); \\ mp(m, \text{sold}) &= \max(mp(m-1, \text{sold}), mp(m-1, \text{hold}) + p_m - f). \end{aligned} \quad (4.2.5)$$

Зауважте, що наступні формули є еквівалентними, оскільки кінцевою відповіддю є $mp(n, \text{sold})$:

$$\begin{aligned} mp(m, \text{hold}) &= \max(mp(m-1, \text{hold}), mp(m-1, \text{sold}) - p_m - f); \\ mp(m, \text{sold}) &= \max(mp(m-1, \text{sold}), mp(m-1, \text{hold}) + p_m). \end{aligned}$$

- період охолодження із задачі 6:

$$\begin{aligned} mp(m, \text{hold}) &= \max(mp(m-1, \text{hold}), mp(m-1, \text{sold}) - p_m); \\ mp(m, \text{sold}) &= \max(mp(m-1, \text{sold}), mp(m-1, \text{cold})); \\ mp(m, \text{cold}) &= mp(m-1, \text{hold}) + p_m. \end{aligned} \quad (4.2.6)$$

Тепер обговорюються стани кількість транзакцій. Задачі 1, 2 та 3 передбачають обмежену кількість транзакцій. Тому інформація про кількість транзакцій включається до стану підзадачі.

Зауваження 10. Можна зберігати явний параметр *hold/sold* або замінити його неявним індексом j у розкладі. Це можливо, оскільки парність j однозначно визначає тип операції (купівля або продаж). Переходи в усіх цих задачах є такими:

$$tr(m, j) = \max(tr(m-1, j), tr(m-1, j-1) + (-1)^j p_m). \quad (4.2.7)$$

Зауваження 11. Зауважте, що основна відмінність між трьома задачами полягає не в переходах. Натомість відрізняється розташування кінцевої відповіді: $\max_{j=0..K} tr(n, 2j)$. $K = 1$ у задачі 1, і $K = 2$ у задачі 2.

4.2.3. Робастні задачі. У цьому підрозділі представлено робастні версії всіх задач. Дано означення розв'язків з максимальним прибутком у найгіршому випадку та розв'язків з мінімальним жалем (англ. *regret*). Наведено спостереження, які дозволили узагальнити розв'язки з попереднього підрозділу для знаходження максимального прибутку в найгіршому випадку. Нарешті, показано, що розв'язок з мінімальним жалем може відрізнитися від розв'язку з максимальним прибутком у найгіршому випадку.

Означення 29. Усі робастні задачі замінюють послідовність p_i двома послідовностями l_i та r_i додатних чисел, що позначають *діапазони цін* певної акції протягом n послідовних днів. А саме, $l_i \leq p_i \leq r_i$ виконується для кожного $i = 1, \dots, n$.

Означення 30. Природно, визначення *прибутку* також потребує коригування. Однак це коригування є семантичним, а не символічним:

$$\begin{aligned} \text{profit}(I; p) &= -p_{i_1} + p_{i_2} - p_{i_3} + p_{i_4} - \dots - p_{i_{2k-1}} + p_{i_{2k}} \\ &= (p_{i_2} - p_{i_1}) + (p_{i_4} - p_{i_3}) + \dots + (p_{i_{2k}} - p_{i_{2k-1}}). \end{aligned} \quad (4.2.8)$$

Іншими словами, різниця полягає в тому, що значення прибутку тепер залежить від конкретної реалізації цін p , яка задовольняє заданим обмеженням (l, r) . Зауважте, що увесь розклад фіксується до того, як стають відомі ціни.

Зауваження 12. Зауважте, що класичну версію можна звести до робастної, поклавши $l_i = p_i = r_i$. Тому розв'язок робастної версії не може бути асимптотично швидшим за оптимальний розв'язок її класичної версії.

Означення 31. Мінімізуючи це визначення за реалізацією p , визначається *прибуток у найгіршому випадку* (англ. *worst-case*) розкладу I :

$$\begin{aligned} \text{worst-case profit}(I; p) &= \min_p \text{profit}(i_1, \dots, i_{2k}; p) \\ &= (l_{i_2} - r_{i_1}) + (l_{i_4} - r_{i_3}) + \dots + (l_{i_{2k}} - r_{i_{2k-1}}). \end{aligned} \quad (4.2.9)$$

Останнє перетворення можливе, оскільки обмеження є незалежними одне від одного. Цільова функція є лінійною за кожним зі своїх аргументів p_m . Отже, вона досягає свого мінімального значення в одній з граничних точок. Найгіршим сценарієм при купівлі акцій є r_m , а найгіршим сценарієм при продажу акцій є l_m .

Розв'язки за допомогою динамічного програмування, представлені в розділі про класичні розв'язки, можуть знайти розв'язок із максимальним прибутком у найгіршому випадку для робастної задачі. Щоб адаптувати ці розв'язки, потрібно замінити кожне входження p_m у (4.2.4)–(4.2.7) на l_m або r_m .

Означення 32. Щоб визначити *жаль* (англ. *regret*) розв'язку, спочатку потрібно визначити *втрати у найгіршому випадку* (англ. *worst-case loss*) розв'язку I відносно іншого розв'язку J .

$$\text{worst-case loss}(I; J) = \max_p (\text{profit}(J; p) - \text{profit}(I; p)). \quad (4.2.10)$$

Означення 33. *Жаль* розв'язку I — це максимальні втрати у найгіршому випадку серед усіх можливих розв'язків J , де $\arg \max_J$ часто називають *оптимальним розв'язком*:

$$\text{regret}(I) = \max_J \text{worst-case loss}(I; J). \quad (4.2.11)$$

Розв'язок з мінімальним жалем — це той, що має найменший можливий жаль. Важливо розуміти, що мінімізація жалю відрізняється від максимізації прибутку в найгіршому випадку.

Приклад 2. Розглянемо два дні з діапазонами цін (1, 3) та (2, 4) відповідно. Є два розв'язки: нічого не робити з $I = ()$, або купити і продати з $I = (1, 2)$. Їхні прибутки в найгіршому випадку становлять 0 та $\min_p(p_2 - p_1) = 2 - 3 = -1$ відповідно. Їхній жаль становить $\max_p(0 - 0, (p_2 - p_1) - 0) = \max_p(0, p_2 - p_1) = 4 - 1 = 3$ та $\max_p(0 - (p_2 - p_1), (p_2 - p_1) - (p_2 - p_1)) = \max_p(p_1 - p_2, 0) = 3 - 2 = 1$ відповідно.

У таблиці 4.1 наведено порівняльний аналіз розв'язків. Підкреслено оптимальний розв'язок за кожним критерієм (стовпчиком).

Таблиця 4.1

Порівняння прибутків у найгіршому випадку та жалю для двох розкладів на діапазонах цін (1, 3) та (2, 4).

розклад	прибуток	жаль
$I = ()$	<u>0</u>	3
$I = (1, 2)$	-1	<u>1</u>

Отже, розв'язком з максимальним прибутком у найгіршому випадку є нічого не робити, а розв'язком з мінімальним жалем — купити і продати. Результат інтерпретується на користь мінімізації жалю на ринку що зростає.

Наступний приклад демонструє, що розв'язки можуть збігатися, зокрема у випадку ринку що спадає.

Приклад 3. Розглянемо два дні з діапазонами цін (2, 4) та (1, 3) відповідно. Є два розв'язки: нічого не робити з $I = ()$, або купити і продати з $I = (1, 2)$. Їхні прибутки в найгіршому випадку становлять 0 та $\min_p(p_2 - p_1) = 1 - 4 = -3$ відповідно. Їхній жаль становить $\max_p(0 - 0, (p_2 - p_1) - 0) = \max_p(0, p_2 - p_1) = 3 - 2 = 1$ та $\max_p(0 - (p_2 - p_1), (p_2 - p_1) - (p_2 - p_1)) = \max_p(p_1 - p_2, 0) = 4 - 1 = 3$ відповідно.

У таблиці 4.2 наведено порівняльний аналіз розв'язків. Підкреслено оптимальний розв'язок за кожним критерієм (стовпчиком).

Таблиця 4.2

Порівняння прибутків у найгіршому випадку та жалю для двох розкладів на діапазонах цін $(2, 4)$ та $(1, 3)$.

розклад	прибуток	жаль
$I = ()$	<u>0</u>	<u>1</u>
$I = (1, 2)$	-3	3

Отже, як розв'язок з максимальним прибутком у найгіршому випадку, так і розв'язок з мінімальним жалем — це нічого не робити.

Перший приклад можна додатково модифікувати, щоб показати, що жаль двох розв'язків може відрізнятися як завгодно сильно. Зокрема, у випадку двох днів з діапазонами цін $(1, x+1)$ та $(x, 2x)$, розв'язок із максимальним прибутком у найгіршому випадку має жаль $2x - 1$, тоді як мінімальний жаль дорівнює 1. Зокрема, не існує константи C , такої що:

$$\text{regret}(\text{max worst-case profit solution}) \leq C \cdot \text{regret}(\text{min regret solution}).$$

Тому необхідний новий ефективний метод для знаходження або навіть наближення (з точністю до константного множника) мінімального жалю.

4.2.4. Робастні розв'язки. У цьому підрозділі розроблено ефективні алгоритми для знаходження розв'язків з мінімальним жалем для всіх обговорених задач.

Почнемо з аналізу визначення жалю. (4.2.11) та (4.2.10) разом дають такий вираз:

$$\text{regret}(I) = \max_J \text{worst-case loss}(I; J) = \max_{J,p} (\text{profit}(J; p) - \text{profit}(I; p)).$$

Отже, необхідно розглянути вираз

$$\min_I \max_{J,p} (\text{profit}(J; p) - \text{profit}(I; p)). \quad (4.2.12)$$

Розглянемо довільний день i . Розклади I та J можуть купувати акції в цей день, продавати акції або нічого не робити. Згідно з правилом добутку, всього існує 9 випадків.

1. Розклад I купує акції в день i .

- а) Обидва розклади купують акції в день i . Жаль не залежить від реалізації ціни p_i ;
- б) Розклад I купує акції в день i , а розклад J продає акції в день i . $\max_p$ у формулі жалю передбачає, що потрібно встановити $p_i = r_i$. Це означає, що розклад I заплатить більше, а розклад J заробить більше;
- в) Розклад I купує акції в день i , а розклад J нічого не робить. Встановлюємо $p_i = r_i$;

2. Розклад I продає акції в день i .

- а) Розклад I продає акції в день i , а розклад J купує акції в день i . Встановлюємо $p_i = l_i$, щоб розклад I заробив менше, а розклад J заплатив менше;
- б) Обидва розклади продають акції в день i . Жаль не залежить від реалізації ціни p_i ;
- в) Розклад I продає акції в день i , а розклад J нічого не робить. Встановлюємо $p_i = l_i$;

3. Розклад I нічого не робить в день i .

- а) Розклад I нічого не робить в день i , а розклад J купує акції в день i . Встановлюємо $p_i = l_i$;
- б) Розклад I нічого не робить в день i , а розклад J продає акції в день i . Встановлюємо $p_i = r_i$;
- в) Обидва розклади нічого не роблять в день i . Жаль не залежить від реалізації ціни p_i ;

Аналіз підсумовано в таблиці 4.3. Прочерк вказує на те, що жаль не залежить від значення p_i .

Таблиця 4.3

Кожна комірка містить значення p_i , що призводить до максимального жалю для заданої пари операцій у розкладах I та J .

$I \setminus J$	купівля	пропуск	продаж
купівля	—	r_i	r_i
пропуск	l_i	—	r_i
продаж	l_i	l_i	—

Отже, маючи фіксовану пару розкладів I та J , можна знайти ціни p , що призводять до максимального жалю. Зокрема, можна обчислити їх за лінійний час.

Залишається проаналізувати поведінку виразу виду $\min_I \max_J$. Це нагадує гру двох осіб. Як і раніше, частина $\min$ - $\max$ передбачає наявність певної оптимальної підструктури й існує багато підзадач, що перекриваються.

Це основні характеристики задачі динамічного програмування. Різниця полягає в тому, що цього разу потрібно включити поточний стан обох гравців.

Спочатку обговорюються стани утримувати, продано та охолодження. Цей підрозділ обговорює застосування підходу динамічного програмування до робастних задач 4, 6 та 5.

Як і раніше, стан залежить від того, чи утримує розклад I акції в кінці дня m . Тепер стан також залежить від того, чи утримує розклад J акції в кінці дня.

Робастна задача 6 додатково враховує можливість того, що I та J можуть перебувати в періоді охолодження, якщо вони щойно продали акції попереднього дня $m - 1$. Переходи в цих задачах є такими, де mr означає мінімальний жаль (англ. *minimum regret*).

- необмежена кількість транзакцій із робастної задачі 4:

$$\begin{aligned}
mr(m, \text{hold}, \text{hold}) &= \min(\max(mr(m-1, \text{hold}, \text{hold}), \\
&\quad mr(m-1, \text{hold}, \text{sold}) - l_m), \\
&\quad \max(mr(m-1, \text{sold}, \text{hold}) + r_m, \\
&\quad mr(m-1, \text{sold}, \text{sold}))); \\
mr(m, \text{hold}, \text{sold}) &= \min(\max(mr(m-1, \text{hold}, \text{hold}) + r_m, \\
&\quad mr(m-1, \text{hold}, \text{sold})), \\
&\quad \max(mr(m-1, \text{sold}, \text{hold}) + 2r_m, \\
&\quad mr(m-1, \text{sold}, \text{sold}) + r_m)); \\
mr(m, \text{sold}, \text{hold}) &= \min(\max(mr(m-1, \text{hold}, \text{hold}) - l_m, \\
&\quad mr(m-1, \text{hold}, \text{sold}) - 2l_m), \\
&\quad \max(mr(m-1, \text{sold}, \text{hold}), \\
&\quad mr(m-1, \text{sold}, \text{sold}) - l_m)); \\
mr(m, \text{sold}, \text{sold}) &= \min(\max(mr(m-1, \text{hold}, \text{hold}), \\
&\quad mr(m-1, \text{hold}, \text{sold}) - l_m), \\
&\quad \max(mr(m-1, \text{sold}, \text{hold}) + r_m, \\
&\quad mr(m-1, \text{sold}, \text{sold}))).
\end{aligned} \tag{4.2.13}$$

Як бачите, рівняння дещо громіздкі. Замість цього їх можна представити у вигляді матриці:

$$\begin{pmatrix} 0 & -l_m & r_m & 0 \\ r_m & 0 & 2r_m & r_m \\ -l_m & -2l_m & 0 & -l_m \\ 0 & -l_m & r_m & 0 \end{pmatrix} \tag{4.2.14}$$

- комісія за транзакцію із робастної задачі 5:

$$\begin{pmatrix} 0 & -l_m - f & r_m + f & 0 \\ r_m & 0 & 2r_m + f & r_m + f \\ -l_m & -2l_m - f & 0 & -l_m - f \\ 0 & -l_m & r_m & 0 \end{pmatrix} \quad (4.2.15)$$

- період охолодження із робастної задачі 6 має матрицю 9×9 , яку можна побудувати аналогічно.

Тепер обговорюються стани кількості транзакцій. Робастні задачі 1, 2 та 3 передбачають обмежену кількість транзакцій. Тому інформація про кількість транзакцій включається до стану підзадачі.

Можна зберігати явний параметр `hold/sold` або замінити його неявними індексами i та j у відповідному розкладі. Це можливо, оскільки парність i та j однозначно визначає типи операцій (купівля або продаж). Переходи в усіх цих задачах є такими:

$$\begin{aligned} mr(m, i, j) = \min(& \max(mr(m-1, i, j), \\ & mr(m-1, i, j-1) + f_{0,(-1)^j}(l_m, r_m)), \\ & \max(mr(m-1, i-1, j) + f_{(-1)^i,0}(l_m, r_m), \\ & mr(m-1, i-1, j-1) + f_{(-1)^i,(-1)^j}(l_m, r_m))). \end{aligned} \quad (4.2.16)$$

Тут f — це матриця функцій (з дещо нестандартними індексами $-1, 0$ та 1), що обчислюють зміну жалю на основі таблиці 4.3. Конкретні неважливі, адже основним результатом є те, що певний розв'язок методом динамічного програмування є можливим.

Зауваження 13. *Зауважте, що основна відмінність між трьома задачами полягає не в переходах. Натомість відрізняється розташування кінцевої відповіді: $\max_{i=0..K, j=0..K} mr(n, 2i, 2j)$. $K = 1$ у задачі 1, і $K = 2$ у задачі 2.*

Незважаючи на появу матриць 4×4 та 9×9 , їхні розміри не включаються до аналізу часової складності, оскільки вони є константними відносно кількості днів. Отже, отримані розв'язки для робастних задач 4, 6 та 5 мають час $O(n)$.

Складність розв'язку розробленого для робастної задачі з $O(nk^2)$, оскільки кількість станів зростає на множник $O(k)$. Зокрема, часова складність для робастних задач 1 та 2 все ще становить $O(n)$, оскільки $K = O(1)$.

Загалом, часова складність майже усіх розв'язків залишилася такою ж, як і у їхніх не робастних відповідників. Відкритим залишається питання, чи існують більш оптимальні розв'язки для робастної версії цієї задачі.

Гіпотеза 6. *Більш ефективними алгоритмами для цієї задачі будуть схеми апроксимації з певним константним множником, що залежить від потреб конкретного екземпляра задачі.*

4.3. Висновки до розділу 4

У цьому розділі було всебічно досліджено деякі класичні та робастні моделі дискретної торгівлі акціями. Сформульовано робастні аналоги усіх класичних задач для інтервальної моделі невизначеності майбутніх цін акцій. Це розширює інструментарій доступний майбутнім дослідникам для моделювання процесу керування портфелем.

Наведено конструктивні підтвердження того, що мінімаксна постановка кожної з розглянутих оптимізаційних задач відрізняється від постановки з мінімізацією жалю. Наведені приклади мають підказати практикантам коли доцільно використовувати кожен із досліджених цільових функцій.

Для усіх робастних постановок побудовано алгоритми мінімізації жалю з поліноміальними часом роботи. Така швидкодія є необхідною вимогою життєздатності методу у суворих практичних реаліях.

Закладено основи тонкого аналізу обчислювальної складності усіх розглянутих задач. Майбутні дослідження в цій галузі є другим пріоритетом після одержання поліноміальних алгоритмів. Також поглиблено та розширено фундаментальне розуміння ширини застосування робастної оптимізації у галузі дослідження операцій

ВИСНОВКИ

У дисертаційній роботі наведено теоретичне узагальнення і нове вирішення актуальної наукової задачі побудови ефективних наближених та робастних алгоритмів для проблем комбінаторної оптимізації, що характеризуються високою обчислювальною складністю або невизначеністю вхідних даних. Розроблені методи дозволяють подолати обмеження класичних підходів при роботі з фундаментальними дискретними структурами та прикладними задачами маршрутизації.

Основні наукові та практичні результати роботи полягають у такому:

1. **Вперше одержано** лінійну верхню оцінку потужності побітової суми інтервальних множин, що спростовує гіпотезу про квадратичну залежність. На основі цього теоретичного результату **розроблено** швидкий алгоритм обчислення побітової суми з полілогарифмічною часовою складністю, що суттєво підвищує ефективність обробки даних у задачах адитивної комбінаторики.
2. **Набули подальшого розвитку** алгоритми лексикографічного аналізу рядкових даних. **Запропоновано** новий підхід до побудови розкладу Ліндона, що базується на шаруватій архітектурі даних та використанні монотонного стека. Це дозволило оптимізувати процедури, критичні для біоінформатики та систем стиснення даних.
3. **Встановлено** фундаментальний зв'язок між задачами обчислення згорток числових послідовностей та задачами маршрутизації. Зокрема, **доведено звідність** задачі обчислення $(\min, +)$ -згортки до спеціального випадку задачі комівояжера, що дозволило застосувати результати теорії детальної складності для аналізу алгоритмів оптимізації.

4. **Удосконалено** методи аналізу деревоподібних структур даних. **Розроблено** ефективні алгоритми пошуку та підрахунку зигзагоподібних підпоследовностей у деревах, які, на відміну від підходів на основі декомпозиції центроїдів, забезпечили кращі показники часової складності.
5. **Дістало подальшого розвитку** дослідження топологічних властивостей регулярних графів. **Запропоновано** конструктивні методи визначення реберної зв'язності циркулянтних графів, що є необхідним для проєктування відмовостійких мережевих архітектур.
6. **Запропоновано алгоритм** маршрутизації групи безпілотних літальних апаратів в умовах невизначеності. **Обґрунтовано** введення інтегральної метрики близькості рухомих об'єктів та на її основі **побудовано** алгоритми кластеризації, які враховують динаміку змін взаємного розташування об'єктів, що дозволяє підвищити ефективність логістичних операцій.
7. **Розроблено** робастні методи оптимізації для задач фінансового ринку. **Побудовано** стратегії торгівлі, які мінімізують жаль в інтервальних сценаріях поведінки ринку, та експериментально підтверджено їхню перевагу над класичними евристичними методами в умовах високої волатильності.

Таким чином, у дисертаційній роботі досягнуто поставленої мети. Отримані результати у сукупності складають основу для створення нових інформаційних технологій робастної оптимізації та можуть бути використані при розробці спеціалізованого програмного забезпечення для систем підтримки прийняття рішень у логістиці, фінансовому секторі та обробці великих даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Skybytskyi N. M. and Denysov K. I. Computational Equivalence of One-Dimensional Quota TSP Variant and $(\min, +)$ Convolution // *Journal of Numerical and Applied Mathematics*. 2024. Vol. 2, no. 2. P. 62–67. Access mode: <http://dx.doi.org/10.17721/2706-9699.2024.2.04>.
- [2] Skybytskyi N. M. Robust Time to Buy and Sell Stock // *Journal of Numerical and Applied Mathematics*. 2025. Vol. 1, no. 1. P. 90–100. Access mode: <http://dx.doi.org/10.17721/2706-9699.2025.1.08>.
- [3] Skybytskyi N. M. Two-Level Algorithm for the UAVRP with Moving Targets // *Cybernetics and Computer Technologies*. 2025. Dec. Vol. 4, no. 4. P. 29–36. Access mode: <http://dx.doi.org/10.34229/2707-451X.25.4.3>.
- [4] Skybytskyi N. M. UAVRP With Moving Targets // Proceedings of the XXII International Scientific-Practical Conference “Shevchenkivska Vesna — 2024”. Kyiv, Ukraine. 2024. Apr. P. 103. Access mode: https://probability.knu.ua/shv2024/ShV_2024.pdf.
- [5] Skybytskyi N. M. and Denysov K. I. Equivalence of 1D K-TSP Variant and $(\min, +)$ Convolution // Proceedings of the XVII International Scientific and Practical Conference “Information Technologies and Automation—2024”. Odesa, Ukraine. 2024. Nov. P. 103. Access mode: <https://ontu.edu.ua/download/konfi/2024/Collection-of-abstracts-of-the-conference-ITIA-2024.pdf>.
- [6] Skybytskyi N. M. Linear Upper Bound on a Segment XOR Cardinality // Proceedings of the XXIII International Scientific-Practical Conference “Shevchenkivska Vesna — 2025”. Kyiv, Ukraine. 2025. Apr. P. 48–49. Access mode: https://probability.knu.ua/shv2025/ShV_2025.pdf.

- [7] Semenov V. V. and Skybytskyi N. M. Extreme Values of a Global Minimum Cut in Connected Regular Graphs // Proceedings of the XXI International Workshop “Dynamical System Modeling and Stability Investigations” (DSMSI-2025). Kyiv, Ukraine. 2025. May.
- [8] Скибицький Н. М. Робастна комбінаторна оптимізація у задачі планування відключень електроенергії // Матеріали XII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених “Об’єднані наукою: перспективи міждисциплінарних досліджень”. Київ, Україна. 2025.
- [9] Skybytskyi N. M. A Data Structure for APSP Queries in a Layered Graph // Modern Cybernetics: Glushkov Readings. Proceedings of the XIV International Scientific and Practical Conference. Kyiv, Ukraine : V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine. 2025.
- [10] Box G. E. P., Hunter W. H., Hunter S., et al. Statistics for Experimenters. John Wiley and sons New York, 1978. Vol. 664.
- [11] Sergienko I. V., Zadiraka V. K. i Lytvyn O. M. Elements of the General Theory of Optimal Algorithms. Springer Optimization and Its Applications. Springer International Publishing, 2021. ISBN: 9783030909086. Режим доступу: <http://dx.doi.org/10.1007/978-3-030-90908-6>.
- [12] Lions J. L. Optimal Control of Systems Governed by Partial Differential Equations. Springer, 1971. Vol. 170.
- [13] Скопєцький В. В., Стоян В. А. і Кривонос Ю. Г. Математичне моделювання прямих та обернених задач динаміки систем з розподіленими параметрами // К.: Наук. думка. 2002.
- [14] Semenov V. V. and Kharkov O. S. Strong Convergence of the Regularized Operator Extrapolation Algorithm For Variational Inequalities // *Cybernetics and Systems Analysis*. 2024. May. Vol. 60, no. 3. P. 392–402. Access mode: <http://dx.doi.org/10.1007/s10559-024-00680-9>.

- [15] Маліцький Ю. В. і Семенов В. В. Нові теореми сильної збіжності проксимального методу для задачі рівноважного програмування // *Журнал обчислювальної та прикладної математики*. 2010. Т. 102. С. 79–88.
- [16] Маліцький Ю. В. і Семенов В. В. Варіант екстраградієнтного алгоритму для монотонних варіаційних нерівностей // *Кибернетика і системний аналіз*. 2014. Т. 50. С. 125–131.
- [17] Kanzow Christian. Some equation-based methods for the nonlinear complementarity problem // *Optimization Methods and Software*. 1994. Jan. Vol. 3, no. 4. P. 327–340. Access mode: <http://dx.doi.org/10.1080/10556789408805573>.
- [18] Pang Jong-Shi and Gabriel Steven A. NE/SQP: A robust algorithm for the nonlinear complementarity problem // *Mathematical Programming*. 1993. June. Vol. 60, no. 1–3. P. 295–337. Access mode: <http://dx.doi.org/10.1007/BF01580617>.
- [19] She-feng Zhao and Pu-sheng Fei. Projection and contraction methods for nonlinear complementarity problem // *Wuhan University Journal of Natural Sciences*. 2000. Dec. Vol. 5, no. 4. P. 391–396. Access mode: <http://dx.doi.org/10.1007/BF02850762>.
- [20] Tseng Paul. On linear convergence of iterative methods for the variational inequality problem // *Journal of Computational and Applied Mathematics*. 1995. June. Vol. 60, no. 1–2. P. 237–252. Access mode: [http://dx.doi.org/10.1016/0377-0427\(94\)00094-H](http://dx.doi.org/10.1016/0377-0427(94)00094-H).
- [21] Шатирко А. В., Хусаїнов Д. Я. і Шакотько Т. І. Отримання умов збіжності процесів навчання у математичних моделях нейродинаміки з післядією // *International Scientific Technical Journal “Problems of Control and Informatics”*. 2022. Груд. Т. 67, № 5. С. 5–16. Режим доступу: <http://dx.doi.org/10.34229/2786-6505-2022-5-1>.

- [22] Hrygorenko V., Klyushin D., and Lyashko S. Multi-Block ADMM Method With Nesterov Acceleration // *International Scientific Technical Journal “Problems of Control and Informatics”*. 2021. Mar. Vol. 66, no. 4. P. 5–18. Access mode: <http://dx.doi.org/10.34229/1028-0979-2021-4-1>.
- [23] Byrd R. H., Lu P., Nocedal J., and Zhu C. A Limited Memory Algorithm for Bound Constrained Optimization // *SIAM Journal on Scientific Computing*. 1995. Sep. Vol. 16, no. 5. P. 1190–1208. Access mode: <http://dx.doi.org/10.1137/0916069>.
- [24] Boyd S. and Vandenberghe L. Convex Optimization. Cambridge University Press, 2004. Mar. ISBN: 9780511804441. Access mode: <http://dx.doi.org/10.1017/CBO9780511804441>.
- [25] Vandenberghe L. and Boyd S. Semidefinite Programming // *SIAM Review*. 1996. Mar. Vol. 38, no. 1. P. 49–95. Access mode: <http://dx.doi.org/10.1137/1038003>.
- [26] Bauschke H. H. and Combettes P. L. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, 2011.
- [27] Lyashko S. I., Semenov V. V., and Voitova T. A. Low-cost modification of Korpelevich’s methods for monotone equilibrium problems // *Cybernetics and Systems Analysis*. 2011. July. Vol. 47, no. 4. P. 631–639. Access mode: <http://dx.doi.org/10.1007/s10559-011-9343-1>.
- [28] Malitsky Yu. Projected Reflected Gradient Methods for Monotone Variational Inequalities // *SIAM Journal on Optimization*. 2015. Jan. Vol. 25, no. 1. P. 502–520. Access mode: <http://dx.doi.org/10.1137/14097238X>.
- [29] Malitsky Y. Golden Ratio Algorithms for Variational Inequalities // *Mathematical Programming*. 2019. July. Vol. 184, no. 1–2. P. 383–410. Access mode: <http://dx.doi.org/10.1007/s10107-019-01416-w>.
- [30] Chen Shi, Li Qin, Tse Oliver, and Wright Stephen J. Accelerating optimization over the space of probability measures. 2023. Access mode: <https://arxiv.org/abs/2310.04006>.

- [31] Duchi J., Shalev-Shwartz S., Singer Y., and Chandra T. Efficient Projections Onto the ℓ_1 -Ball for Learning in High Dimensions // ICML '08: Proceedings of the 25th International Conference on Machine Learning. 2008. P. 272–279. Access mode: <https://stanford.edu/~jduchi/projects/DuchiShSiCh08.pdf>.
- [32] Stetsyuk P., Fischer A., and Khomiak O. Unified Presentation of the Generalized Ellipsoid Method // Theory, Algorithms, and Experiments in Applied Optimization. Springer Nature Switzerland, 2025. P. 345–376. ISBN: 9783031913570. Access mode: http://dx.doi.org/10.1007/978-3-031-91357-0_16.
- [33] Lee H.-S. and Zhu J. Super-Efficiency Infeasibility and Zero Data in DEA // *European Journal of Operational Research*. 2012. Jan. Vol. 216, no. 2. P. 429–433. Access mode: <http://dx.doi.org/10.1016/j.ejor.2011.07.050>.
- [34] Sayin S. Multi-Objective Optimization and Decision Support Systems // Encyclopedia of Optimization. Springer US, 2008. P. 2460–2465. ISBN: 9780387747590. Access mode: http://dx.doi.org/10.1007/978-0-387-74759-0_422.
- [35] Bollobas B. Graph Theory: An Introductory Course. Heidelberg: Springer-Verlag, 1979.
- [36] Diestel R. Graph Theory. Graduate Texts in Mathematics. Springer Berlin Heidelberg, 2017. ISBN: 9783662536223. Access mode: <http://dx.doi.org/10.1007/978-3-662-53622-3>.
- [37] Diestel R. Extremal Graph Theory // Graph Theory. Berlin, Heidelberg : Springer Berlin Heidelberg, 2025. P. 179–226. ISBN: 978-3-662-70107-2. Access mode: https://doi.org/10.1007/978-3-662-70107-2_7.
- [38] Stanley R. P. Enumerative Combinatorics. Cambridge Studies in Advanced Mathematics. 2 ed. Cambridge University Press, 2011.

- [39] Semenova N. V., Kolechkina L. N., and Nagirna A. N. An Approach to Solving Discrete Vector Optimization Problems Over a Combinatorial Set of Permutations // *Cybernetics and Systems Analysis*. 2008. May. Vol. 44, no. 3. P. 441–451. Access mode: <http://dx.doi.org/10.1007/s10559-008-9016-x>.
- [40] Ben-Tal A. and Nemirovski A. Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. Society for Industrial and Applied Mathematics, 2001. Jan. ISBN: 9780898718829. Access mode: <http://dx.doi.org/10.1137/1.9780898718829>.
- [41] Ben-Tal A., El Ghaoui L., and Nemirovski A. Robust Optimization. Princeton University Press, 2009. Dec. ISBN: 9781400831050. Access mode: <http://dx.doi.org/10.1515/9781400831050>.
- [42] Bertsimas D. and Sim M. The Price of Robustness // *Operations Research*. 2004. Feb. Vol. 52, no. 1. P. 35–53. Access mode: <http://dx.doi.org/10.1287/opre.1030.0065>.
- [43] Goerigk M. and Kurtz J. Data-Driven Robust Optimization Using Deep Neural Networks // *Computers & Operations Research*. 2023. Mar. Vol. 151. P. 106087. Access mode: <http://dx.doi.org/10.1016/j.cor.2022.106087>.
- [44] Aissi H., Bazgan C., and Vanderpooten D. Min–Max and Min–Max Regret Versions of Combinatorial Optimization Problems: A Survey // *European Journal of Operational Research*. 2009. Sep. Vol. 197, no. 2. P. 427–438. Access mode: <http://dx.doi.org/10.1016/j.ejor.2008.09.012>.
- [45] Kasperski A. Minmax Regret Minimum Selecting Items // Discrete Optimization with Interval Data. Springer Berlin Heidelberg, 2008. P. 51–60. ISBN: 9783540784845. Access mode: http://dx.doi.org/10.1007/978-3-540-78484-5_5.
- [46] Чикрий А.А. Конфликтно-управляемые процессы. Киев : Наукова думка, 1992. ISBN: 5-12-003426-8.
- [47] Isaacs R. Differential Games. Dover Publications, 2012. P. 741.
- [48] Friedman A. Differential Games. Dover Publications, 2013. P. 368.

- [49] Pichkur V. V. Maximum Sets of Initial Conditions in Practical Stability and Stabilization of Differential Inclusions // *Modern Mathematics and Mechanics*. Springer International Publishing, 2018. Nov. P. 397–410. ISBN: 9783319967554. Access mode: http://dx.doi.org/10.1007/978-3-319-96755-4_20.
- [50] Pichkur V. V. and Sobchuk V. V. Mathematical Model and Control Design of a Functionally Stable Technological Process // *Journal of Optimization, Differential Equations and Their Applications*. 2021. May. Vol. 29, no. 1. P. 32. Access mode: <http://dx.doi.org/10.15421/142102>.
- [51] Golodnikov A. N., Ermol'ev Y. M., Ermol'eva T. Y., Knopov P. S., and Pepelyaev V. A. Integrated Modeling of Food Security Management in Ukraine. I. Model for Management of the Economic Availability of Food // *Cybernetics and Systems Analysis*. 2013. Jan. Vol. 49, no. 1. P. 26–35. Access mode: <http://dx.doi.org/10.1007/s10559-013-9481-8>.
- [52] Benlakhdar S., Rziza M., and Oulad Haj Thami R. In-depth Analysis of von Mises Distribution Models: Understanding Theory, Applications, and Future Directions // *Statistics, Optimization & Information Computing*. 2024. June. Vol. 12, no. 4. P. 1210–1230. Access mode: <http://dx.doi.org/10.19139/soic-2310-5070-1919>.
- [53] Bianchi S., Pantanella A., and Pianese A. Modeling stock prices by multifractional Brownian motion: an improved estimation of the pointwise regularity // *Quantitative Finance*. 2013. Aug. Vol. 13, no. 8. P. 1317–1330. Access mode: <http://dx.doi.org/10.1080/14697688.2011.594080>.
- [54] Jarrett J. E. and Kyper E. ARIMA Modeling With Intervention to Forecast and Analyze Chinese Stock Prices // *International Journal of Engineering Business Management*. 2011. Vol. 3, no. 3. P. 53–58.
- [55] Mondal P., Shit L., and Goswami S. Study of Effectiveness of Time Series Modeling (ARIMA) in Forecasting Stock Prices // *International Journal of Computer Science, Engineering and Applications*. 2014. Vol. 4, no. 2. P. 13.

- [56] Nuij W., Milea V., Hogenboom F., Frasincar F., and Kaymak U. An Automated Framework for Incorporating News Into Stock Trading Strategies // *IEEE Transactions on Knowledge and Data Engineering*. 2013. Vol. 26, no. 4. P. 823–835.
- [57] Tsantekidis A., Passalis N., Tefas A., Kannianen J., Gabbouj M., and Iosifidis A. Forecasting Stock Prices From the Limit Order Book Using Convolutional Neural Networks // *Proceedings of the IEEE 19th Conference on Business Informatics (CBI)*. 2017. Vol. 1. P. 7–12.
- [58] Vijh A. M. S&P 500 Trading Strategies and Stock Betas // *Review of Financial Studies*. 1994. Vol. 7, no. 1. P. 215–251.
- [59] Fifield S. G. M., Power D. M., and Sinclair C. D. An Analysis of Trading Strategies in Eleven European Stock Markets // *European Journal of Finance*. 2005. Vol. 11, no. 6. P. 531–548.
- [60] Wu X., Chen H., Wang J., Troiano L., Loia V., and Fujita H. Adaptive Stock Trading Strategies With Deep Reinforcement Learning Methods // *Information Sciences*. 2020. Vol. 538. P. 142–158.
- [61] Li J., Wang Y., Zhong J., Sun Y., Guo Z., Chen Z., and Fu C. Network Resilience Assessment and Reinforcement Strategy Against Cascading Failure // *Chaos, Solitons & Fractals*. 2022. Vol. 160. P. 112271.
- [62] Dashti Z. A. Z. S., Deplano D., Seatzu C., and Franceschelli M. Resilient Self-Organizing Networks in Multi-Agent Systems via Approximate Random k-Regular Graphs // *2022 IEEE 61st Conference on Decision and Control (CDC)*. 2022. P. 6448–6453.
- [63] Dhuli S., Kouachi S., Chhabra A., and Singh Y. N. Network Robustness Analysis for IoT Networks Using Regular Graphs // *IEEE Internet of Things Journal*. 2022. Vol. 9, no. 11. P. 8809–8819.
- [64] Erdős P. and Rényi A. On Random Graphs // *Publicationes Mathematicae*. 1959. P. 290–297.

- [65] Albert R. and Barabási A.-L. Statistical Mechanics of Complex Networks // *Reviews of Modern Physics*. 2002. P. 47–97.
- [66] Boesch F. and Tindell R. Circulants and Their Connectivities // *Journal of Graph Theory*. 1984. Dec. Vol. 8, no. 4. P. 487–499. Access mode: <http://dx.doi.org/10.1002/jgt.3190080406>.
- [67] Gatt V. On the Enumeration of Circulant Graphs of Prime-Power Order: the Case of p^3 . 2017. Access mode: <https://arxiv.org/abs/1703.06038>.
- [68] Liskovets V. Some Identities for Enumerators of Circulant Graphs // *Journal of Algebraic Combinatorics*. 2003. Nov. Vol. 18, no. 3. P. 189–209. Access mode: <http://dx.doi.org/10.1023/B:JACO.00000011937.70237.0b>.
- [69] Huang X., F. Ramos A., and Deng Y. Optimal Circulant Graphs as Low-Latency Network Topologies // *The Journal of Supercomputing*. 2022. Vol. 78, no. 11. P. 13491–13510.
- [70] Brouwer A. E. and Mesner D. M. The Connectivity of Strongly Regular Graphs // *European Journal of Combinatorics*. 1985. Vol. 6, no. 3. P. 215–216.
- [71] Cioabă S. M. Eigenvalues and Edge-Connectivity of Regular Graphs // *Linear Algebra and Its Applications*. 2010. Vol. 432, no. 1. P. 458–470.
- [72] Ganesan G. Existence of Connected Regular and Nearly Regular Graphs. 2018. Access mode: <https://arxiv.org/abs/1801.08345>.
- [73] Wu J., Barahona M., Tan Y.-J., and Deng H.-Z. Robustness of Regular Ring Lattices Based on Natural Connectivity // *International Journal of Systems Science*. 2011. Vol. 42, no. 7. P. 1085–1092.
- [74] Harary F. The Maximum Connectivity of a Graph // *Proceedings of the National Academy of Sciences*. 1962. Vol. 48, no. 7. P. 1142–1146.
- [75] Nagamochi H. and Ibaraki T. Algorithmic Aspects of Graph Connectivity. Cambridge University Press, 2008.

- [76] Henning M. A. and Yeo A. Matching and Edge-Connectivity in Graphs With Given Maximum Degree // *Discrete Mathematics*. 2021. Vol. 344, no. 8. P. 112438.
- [77] Suil O. and West D. B. Matching and Edge-Connectivity in Regular Graphs // *European Journal of Combinatorics*. 2011. Vol. 32, no. 2. P. 324–329.
- [78] Škoviera M. and Varša P. NP-Completeness of Perfect Matching Index of Cubic Graphs // 39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022) / Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2022. P. 56–1.
- [79] Šubelj L. Computing Well-Balanced Spanning Trees of Unweighted Networks // *Algorithms*. 2025. Vol. 18, no. 12, 760.
- [80] Yoneda R., Tatsukawa T., and Teramae J. The Lower Bound of the Network Connectivity Guaranteeing In-Phase Synchronization // *Chaos: An Interdisciplinary Journal of Nonlinear Science*. 2021. Vol. 31, no. 6.
- [81] Cormen T. H., Leiserson C. E., Rivest R. L., and Stein C. Introduction to Algorithms. MIT Press and McGraw-Hill, 2001.
- [82] Ford L. R. and Fulkerson D. R. Maximal Flow Through a Network // *Canadian Journal of Mathematics*. 1956. P. 399–404.
- [83] Edmonds J. and Karp R. M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems // *Journal of the ACM*. 1972. P. 248–264.
- [84] Dinitz Y. Algorithm for Solution of a Problem of Maximum Flow in a Network With Power Estimation // *Doklady Akademii Nauk SSSR*. 1970. P. 1277–1280.
- [85] Tardos E. A Strongly Polynomial Minimum Cost Circulation Algorithm. 1985.
- [86] Goldberg A. V. and Tarjan R. E. A New Approach to the Maximum Flow Problem // *Journal of the ACM*. 1988.
- [87] Chen L., Kyng R., Liu Y. P., Peng R., Gutenberg M. P., and Sachdeva S. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time // 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS) / IEEE. 2022. P. 612–623.

- [88] Croes G. A. A Method for Solving Traveling-Salesman Problems // *Operations Research*. 1958. Vol. 6, no. 6. P. 791–812. Access mode: <http://www.jstor.org/stable/167074> (online; accessed: 2025-12-12).
- [89] de Berg M., Buchin K., Jansen B., and Woeginger G. Fine-grained Complexity Analysis of Two Classic TSP Variants // *ACM Transactions on Algorithms*. 2021. Vol. 17, no. 1.
- [90] Garey M. R. and Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- [91] Johnson D. S., Papadimitriou C. H., and Steiglitz K. Combinatorial Optimization: Algorithms and Complexity // *The American Mathematical Monthly*. 1984. Mar. Vol. 91, no. 3. P. 209. Access mode: <http://dx.doi.org/10.2307/2322374>.
- [92] Karp R. M. Reducibility Among Combinatorial Problems // *Complexity of Computer Computations*. New York : Plenum Press, 1972. P. 85–103.
- [93] Williams V. On Some Fine-Grained Questions in Algorithms and Complexity // *Proceedings of the International Congress of Mathematicians (ICM)*. 2018. P. 3447–3487.
- [94] Williams V. Some Open Problems in Fine-Grained Complexity // *SIGACT News*. 2018. Vol. 49, no. 4. P. 29–35.
- [95] Korte B. and Vygen J. Combinatorial Optimization: Theory and Algorithms. Springer Berlin Heidelberg, 2018. ISBN: 9783662560396. Access mode: <http://dx.doi.org/10.1007/978-3-662-56039-6>.
- [96] Fredman M. L. and Tarjan R. E. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms // *ACM*. 1987. P. 596–615.
- [97] Eppstein D., Galil Z., and Giancarlo R. Speeding Up Dynamic Programming // *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (SFCS)*. 1988. P. 488–496.

- [98] Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem : Rep. : 388 / Graduate School of Industrial Administration, Carnegie Mellon University ; executor: Christofides N. : 1976.
- [99] Arora S. Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems // *Journal of the ACM*. 1998. Sep. Vol. 45, no. 5. P. 753–782. Access mode: <http://dx.doi.org/10.1145/290179.290180>.
- [100] Holland J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
- [101] Kirkpatrick S., Gelatt Jr C. D., and Vecchi M. P. Optimization by Simulated Annealing // *Science*. 1983. Vol. 220, no. 4598. P. 671–680.
- [102] Dorigo M. and Stützle T. Ant Colony Optimization. MIT Press, 2004.
- [103] Huliantskyi L. F. Search Diversification in ACO Algorithms and Its Application // *Cybernetics and Systems Analysis*. 2025. Jan. Vol. 61, no. 1. P. 21–33. Access mode: <http://dx.doi.org/10.1007/s10559-025-00741-7>.
- [104] Ran X., Xi Y., Lu Y., Wang X., and Lu Z. Comprehensive Survey on Hierarchical Clustering Algorithms and the Recent Developments // *Artificial Intelligence Review*. 2023. Vol. 56, no. 8. P. 8219–8264.
- [105] Hartmann T., Kappes A., and Wagner D. Clustering Evolving Networks // Algorithm Engineering. Springer International Publishing, 2016. P. 280–329. ISBN: 9783319494876. Access mode: http://dx.doi.org/10.1007/978-3-319-49487-6_9.
- [106] Guttmann-Beck N., Hassin R., Khuller S., and Raghavachari B. Approximation Algorithms With Bounded Performance Guarantees for the Clustered Traveling Salesman Problem // *Algorithmica*. 2000. Vol. 28, no. 4. P. 422–437.
- [107] Sheng W., Xi N., Song M., and Chen Y. Robot Path Planning for Dimensional Measurement in Automotive Manufacturing // *Transactions of the ASME*. 2005. Vol. 127, no. 2. P. 420–428.

- [108] Ding C., Cheng Y., and He M. Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs // *Tsinghua Science & Technology*. 2007. Vol. 12, no. 4. P. 459–465. Access mode: <https://www.sciencedirect.com/science/article/pii/S1007021407700688>.
- [109] Bringmann Karl and Cassis Alejandro. Faster Knapsack Algorithms via Bounded Monotone Min-Plus-Convolution. 2022. Access mode: <https://arxiv.org/abs/2205.08493>.
- [110] Bringmann Karl and Cassis Alejandro. Faster 0-1-Knapsack via Near-Convex Min-Plus-Convolution. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2023. Vol. 274. P. 24:1–24:16. Access mode: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2023.24>.
- [111] Cygan M., Mucha M., Węgrzycki K., and Włodarczyk M. On Problems Equivalent to $(\min, +)$ -Convolution // *ACM Transactions on Algorithms*. 2019. Jan. Vol. 15, no. 1. P. 25. Access mode: <https://doi.org/10.1145/3293465>.
- [112] Bremner D., Chan T. M., Demaine E. D., Erickson J., Hurtado F., Iacono J., Langerman S., and Taslakian P. Necklaces, Convolutions, and $X + Y$ // *Algorithms – ESA 2006* / ed. by Azar Y. and Erlebach T. Berlin, Heidelberg : Springer Berlin Heidelberg. 2006. P. 160–171.
- [113] Laporte G. The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms // *European Journal of Operational Research*. 1992. Vol. 59, no. 3. P. 345–358.
- [114] Arsie A. and Frazzoli E. Efficient Routing of Multiple Vehicles With No Explicit Communications // *International Journal of Robust and Nonlinear Control*. 2007. July. Vol. 18, no. 2. P. 154–164. Access mode: <http://dx.doi.org/10.1002/rnc.1258>.
- [115] Murayama T. Analysis of Trade-Off Between Network Connectivity Robustness Versus Coverage Area of Networked Multi-Robot System // *Artificial Life and Robotics*. 2022. Nov. Vol. 27, no. 4. P. 726–733. Access mode: <https://doi.org/10.1007/s10015-022-00794-3>.

- [116] Liu B., Chu T., Wang L., and Xie G. Controllability of a Leader–Follower Dynamic Network With Switching Topology // *IEEE Transactions on Automatic Control*. 2008. May. Vol. 53, no. 4. P. 1009–1013. Access mode: <http://dx.doi.org/10.1109/TAC.2008.919548>.
- [117] Jansen B. and Belpaeme T. A Computational Model of Intention Reading in Imitation // *Robotics and Autonomous Systems*. 2006. May. Vol. 54, no. 5. P. 394–402. Access mode: <http://dx.doi.org/10.1016/j.robot.2006.01.006>.
- [118] Кучеренко Ю. Ф., Науменко М. В. і Кузнецова М. Ю. Аналіз досвіду застосування безпілотних літальних апаратів та визначення напрямку їх подальшого розвитку при веденні мережецентричних операцій // *Системи озброєння і військова техніка*. 2018. Бер. № 1(53). С. 25–30. Режим доступу: <http://dx.doi.org/10.30748/soivt.2018.53.03>.
- [119] Лупандін В. А., Мегельбей Г. В., Мацько О. Й., Куртсеїтов Т. Л. і Міроненко П. О. Основні тенденції створення та застосування груп безпілотних літальних апаратів // *Наука і техніка Повітряних Сил Збройних Сил України*. 2019. Квіт. № 2(35). С. 88–96. Режим доступу: <http://dx.doi.org/10.30748/nitps.2019.35.11>.
- [120] Huliannytskyi L., Byshovets N., and Zhdanova O. About the Problem of Drone Routing // *Cybernetics and Computer Technologies*. 2024. Sep. no. 3. P. 34–47. Access mode: <http://dx.doi.org/10.34229/2707-451X.24.3.4>.
- [121] Horbulin V. P., Huliannytskyi L. F., and Sergienko I. V. Optimization of UAV Team Routes in the Presence of Alternative and Dynamic Depots // *Cybernetics and Systems Analysis*. 2020. Mar. Vol. 56, no. 2. P. 195–203. Access mode: <http://dx.doi.org/10.1007/s10559-020-00235-8>.
- [122] Savuran H. and Karakaya M. Route Optimization Method for Unmanned Air Vehicle Launched From a Carrier // *Lecture Notes on Software Engineering*. 2015. Jan. Vol. 3. P. 279–284.

- [123] Long Y., Xu G., Zhao J., Xie B., and Fang M. Dynamic Truck–UAV Collaboration and Integrated Route Planning for Resilient Urban Emergency Response // *IEEE Transactions on Engineering Management*. 2024. Vol. 71. P. 9826–9838. Access mode: <http://dx.doi.org/10.1109/TEM.2023.3299693>.
- [124] Kuo R. J., Edbert E., Zulvia F. E., and Lu S.-H. Applying NSGA-II to Vehicle Routing Problem With Drones Considering Makespan and Carbon Emission // *Expert Systems with Applications*. 2023. Vol. 221. P. 119777.
- [125] Ahuja R., Magnanti T., and Orlin J. Network Flows. Prentice-Hall, Inc., 1993.
- [126] Abbas W., Laszka A., and Koutsoukos X. Improving Network Connectivity and Robustness Using Trusted Nodes With Application to Resilient Consensus // *IEEE Transactions on Control of Network Systems*. 2018. Vol. 5, no. 4. P. 2036–2048.
- [127] Wandelt S., Shi X., and Sun X. Estimation and Improvement of Transportation Network Robustness by Exploiting Communities // *Reliability Engineering & System Safety*. 2021. Vol. 206. P. 107307. Access mode: <https://www.sciencedirect.com/science/article/pii/S0951832020308036>.
- [128] Manber Udi and Myers Gene. Suffix Arrays: A New Method for On-Line String Searches // *SIAM Journal on Computing*. 1993. Vol. 22, no. 5. P. 935–948. Access mode: <http://dx.doi.org/10.1137/0222058>.
- [129] Crochemore Maxime, Hancart Christophe, and Lecroq Thierry. Algorithms on Strings. Cambridge University Press, 2007. Apr. ISBN: 9781107670990. Access mode: <http://dx.doi.org/10.1017/CBO9780511546853>.
- [130] Pierre Duval Jean. Factorizing words over an ordered alphabet // *Journal of Algorithms*. 1983. Dec. Vol. 4, no. 4. P. 363–381. Access mode: [http://dx.doi.org/10.1016/0196-6774\(83\)90017-2](http://dx.doi.org/10.1016/0196-6774(83)90017-2).
- [131] Bender Michael A. and Farach-Colton Martín. The LCA Problem Revisited // *LATIN 2000: Theoretical Informatics*. Springer Berlin Heidelberg, 2000. P. 88–94. ISBN: 9783540464150. Access mode: http://dx.doi.org/10.1007/10719839_9.

- [132] Sleator Daniel D. and Tarjan Robert Endre. A data structure for dynamic trees // Proceedings of the thirteenth annual ACM symposium on Theory of computing - STOC '81. ACM Press. 1981. STOC '81. P. 114–122. Access mode: <http://dx.doi.org/10.1145/800076.802464>.
- [133] Skybytskyi Nikita. Connected Circulants. <https://gist.github.com/nika-skybytska/a47f37a8e62b27121250118ffc79a6e2>. 2025.
- [134] Zhou H., Li Y., Ma C., Long K., and Li X. Modular vehicle routing problem: Applications in logistics // *Transportation Research Part E: Logistics and Transportation Review*. 2025. Vol. 197.
- [135] Lou P., Zhou Z., Zeng Y., and Fan C. Vehicle routing problem with time windows and carbon emissions: a case study in logistics distribution // *Environmental Science and Pollution Research*. 2024. Vol. 31. P. 41600–41620.
- [136] Liu T., Yang K., and Moore A. W. The IOC algorithm: efficient many-class non-parametric classification for high-dimensional data // Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2004. Aug. KDD04. P. 629–634. Access mode: <http://dx.doi.org/10.1145/1014052.1014131>.
- [137] Skybytskyi Nikita. Clustered Pursuit: Generated Suite Dataset. https://github.com/nika-skybytska/clustered-pursuit/blob/f64acb39a5b23a902e2cae7f6f8d69ae316fd0fb/tests/generated_suite.json. 2026.

Додаток А

**Список публікацій здобувача за темою дисертації та відомості про
апробацію результатів дисертації**

А.1. Статті у наукових фахових виданнях:

1. Skybytskyi N. M. and Denysov K. I. *Computational Equivalence of One-Dimensional Quota TSP Variant and $(\min, +)$ Convolution* // Journal of Numerical and Applied Mathematics. 2024. no. 2. P. 62–67. Access mode: <http://dx.doi.org/10.17721/2706-9699.2024.2.04>.
2. Skybytskyi N. M. *Robust Time to Buy and Sell Stock* // Journal of Numerical and Applied Mathematics. 2025. no. 1. P. 90–100. Access mode: <http://dx.doi.org/10.17721/2706-9699.2025.1.08>.
3. Skybytskyi N. M. *Two-Level Algorithm for the UAV Routing Problem with Moving Targets* // Cybernetics and Computer Technologies. 2025. Dec. no. 4. P. 29–36. Access mode: <http://dx.doi.org/10.34229/2707-451X.25.4.3>.

А.2. Матеріали конференцій:

1. Skybytskyi N. M. *UAVRP With Moving Targets* // Proceedings of the XXII International Scientific-Practical Conference “Shevchenkivska Vesna”, April 11, 2024, Kyiv, Ukraine. P. 103.
2. Skybytskyi N. M. and Denysov K. I. *Equivalence of 1D K-TSP Variant and $(\min, +)$ Convolution* // Proceedings of the XVII International Scientific and Practical Conference “Information Technologies and Automation — 2024”. October 31–November 1, 2024, Odesa, Ukraine. P. 103.

3. Skybytskyi N. M. *Linear Upper Bound on a Segment XOR Cardinality* // Proceedings of the XXIII International Scientific-Practical Conference “Shevchenkivska Vesna”. April 10, 2025, Kyiv, Ukraine. P. 121.
4. Скибицький Н. М. *Робастна комбінаторна оптимізація у задачі планування відключень електроенергії* // Матеріали XII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених “Об’єднані наукою: перспективи міждисциплінарних досліджень”. 20–21 листопада 2025, Київ, Україна. С. 327.

А.3. Апробація результатів дисертації

Матеріали дисертаційної роботи доповідались та обговорювались на наукових конференціях:

- XXII International Scientific-Practical Conference “Shevchenkivska Vesna” (Київ, 11 квітня 2024);
- XVII International Scientific and Practical Conference “Information Technologies and Automation” (Одеса, 31 жовтня–1 листопада 2024);
- XXIII International Scientific-Practical Conference “Shevchenkivska Vesna” (Київ, 10 квітня 2025);
- XXI International Workshop “Dynamical System Modeling and Stability Investigations” (Київ, 8–10 травня 2025);
- XII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених “Об’єднані наукою: перспективи міждисциплінарних досліджень” (Київ, 20–21 листопада 2025);
- XIV International Scientific and Practical Conference “Modern Cybernetics: Glushkov Readings” (Київ, 30 грудня 2025).

А також доповідались та обговорювались на таких семінарах:

- Науковий семінар кафедри обчислювальної математики факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка (Київ, 2025);
- Науковий семінар “Теорія оптимальних рішень” інституту кібернетики імені В.М. Глушкова НАН України (Київ, 2025).

