Алгоритми апроксимації розв'язків робастних задач комбінаторної оптимізації

Дисертація на здобуття наукового ступеня доктора філософії

Нікіта Максимович Скибицький
Науковий керівник: д.ф.–м.н., професор
Володимир Вікторович Семенов

Кафедра обчислювальної математики
Факультет комп'ютерних наук та кібернетики
Київський національний університет імені Тараса Шевченка
n.skybytskyi@gmail.com, n.skybytskyi@knu.ua

23 грудня 2025 р.
Науковий семінар «Теорія оптимальних рішень»
Інститут кібернетики імені В.М. Глушкова НАН України

Мета, об'єкт, предмет та методи 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.
- **Об'єктом дослідження** є задачі комбінаторної оптимізації на дискретних структурах та процеси прийняття рішень в умовах невизначеності.
- **Предметом дослідження** є методи побудови, аналізу складності та оцінки ефективності наближених і робастних алгоритмів для задач дискретної математики та оптимізації.
- **Методами дослідження** є апарат теорії алгоритмів та обчислювальної складності, методи динамічного програмування, теорії графів, адитивної комбінаторики, метод гілок і меж, а також генетичні алгоритми та методи математичного моделювання.

План

Розділ 2. Максимум-плюс згортка

Розділ 3. Маршрутизація та кластеризація БПЛА

Розділ 4. Робастна торгівля акціями

Постановка задачі

Нехай задано дві послідовності цілих чисел $a = (a_0, \dots, a_{n-1})$ та $b = (b_0, \dots, b_{n-1})$.

Означення

Максимум-плюс згорткою послідовностей a та b називається послідовність c довжини $2n-1$, елементи якої визначаються як:

$$c_k = (a * b)_k = \max_{i+j=k}(a_i + b_j), \quad k = 0, \dots, 2n-2. \tag{1}$$

Застосування

Команда Marek Cygan'a [2] показала, що наступні та інші задачі мають еквівалентну складність (є швидкі $\iff$ зведення):

- Задача *мінімум-плюс згортки* $c_k = \min_{i+j=k}(a_i + b_j)$ тривіально зводиться до вихідної заміною знаків $x \mapsto -x$.
- Перевірка *суперадитивності* послідовності, тобто чи виконується $a_{i+j} \geq a_i + a_j$ для всіх i, j вимагає обчислення згортки $a * a$ послідовності a із собою.
- Задача про *0-1 рюкзак*, де операція згортки використовується для ефективного об'єднання множин часткових розв'язків.
- *Обмежена задача 3SUM*: дано три множини цілих чисел $A, B, C \subseteq \{-n, \ldots, n\}$. Необхідно визначити, чи існують $a \in A, b \in B, c \in C$, такі що $a + b = c$.

Складність

- Існує очевидний алгоритм перебору зі складністю $O(n^2)$.
- David Bremner та ін. [1] запропонували нетривіальний метод зі складністю:
$$O\left(\frac{n^2 \log\log^3 n}{\log^2 n}\right) = o(n^2). \tag{2}$$
- Наразі невідомо про існування суттєво субквадратного алгоритму з часом роботи $O(n^{2-\varepsilon})$ для будь-якого $\varepsilon > 0$.
- Для багатьох важливих частинних випадків (опуклі або монотонні послідовності) відомі швидкі алгоритми зі складністю $O(n \log n)$ або навіть $O(n)$.
- Для звичайної $(+, \times)$-згортки існують алгоритми на основі швидкого перетворення Фур'є зі складністю $O(n \log n)$.

Апроксимація

Oliver Serang [4] запропонував швидкий наближений метод:

- Заміна півкільця $(\mathbb{R} \cup \{-\infty\}, \max, +)$ кільцем $(\mathbb{R}, +, \times)$ через експоненційне відображення з параметром $p > 0$:

$$(a \oplus b)_k \approx \frac{1}{p} \log \left(\sum_{i+j=k} e^{pa_i} \cdot e^{pb_j} \right). \tag{3}$$

  Це дозволяє застосувати FFT за час $O(n \log n)$.

- Наприклад:

$$\max_{p=1}(2,5) = \log(e^2 + e^5) \approx 5.05, \tag{4}$$

$$\max_{p=10}(2,5) = \frac{1}{10} \log(e^{20} + e^{50}) \approx 5.000000000000009. \tag{5}$$

- Точність зростає при збільшенні p, проте швидко призводить до переповнення. У стандарті IEEE 754 не можна представити e^{710}.

Поширення похибки

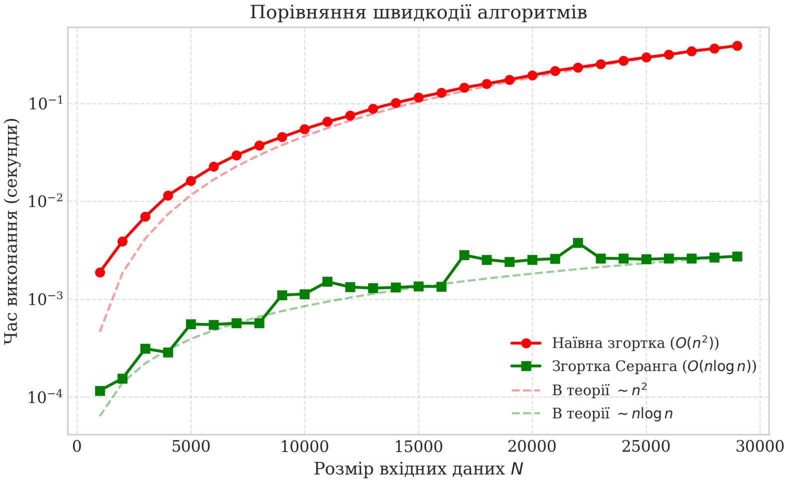
Нехай існує алгоритм, що обчислює $(\max, +)$-згортку з адитивною похибкою, яка не перевищує δ:

$$|\tilde{c}_k - c_k| \leq \delta, \quad \forall k. \tag{6}$$

Для задач отримано наступні теоретичні оцінки точності розв'язків:

Задача	**Глибина зведення**	**Результуюча похибка**
$(\min, +)$-згортка	$O(1)$	δ
Супераддитивність	$O(1)$	δ
0-1 рюкзак	$O(\log N)$	$\delta \cdot \lceil \log_2 N \rceil$ [1]
Обмежена 3SUM	$O(1)$	δ
Найдовший шлях	$O(N)$	$\delta \cdot N$

[1] Для одержання такої оцінки необхідний адаптивний вибір δ, інакше похибка буде $\delta \cdot N$.


Порівняння швидкодії алгоритмів
Час виконання (секунди)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
5000
10000
15000
20000
25000
30000
Розмір вхідних даних N
Наївна згортка ($O(n^2)$)
Згортка Серанга ($O(n \log n)$)
В теорії $\sim n^2$
В теорії $\sim n \log n$

Точність алгоритму Серанга та межа переповнення
Похибка апроксимації MAE
Параметр масштабування p
MAE
IEEE 754
10^{1}
10^{0}
10^{-1}

План

Постановка задачі

Вхідні дані

- K БПЛА з максимальною швидкістю V.
- $M \gg K$ цілей з відомими законами руху $z_j(t)$.
- Швидкість цілей $v_j(t) < V$ і відділена від V.

Задача оптимізації

Розбити множину цілей на K маршрутів $S_1, \ldots, S_K$ та знайти траєкторії БПЛА, щоб мінімізувати час завершення місії:

$$T^* = \max_{k=1..K} T(S_k) \to \min, \tag{7}$$

де $T(S_k)$ — час, необхідний k-му БПЛА для послідовного перехоплення всіх цілей зі своєї підмножини S_k.

Фундаментальні результати щодо задач переслідування (7) зі складним рухом загалом та динамічної задачі комівояжера зокрема отримані в роботах А.О. Чикрія, його учнів та колег [8].

Обчислювальна складність

Hammar та Nilsson довели таке:

Теорема

Навіть для одного БПЛА $K = 1$ і лінійного руху $z_j(t) = v_j \cdot t + z_j(0)$ задача NP-складна. Більше того, NP-складною є навіть задача побудови апроксимації з мультиплікативною константою $C < 2$.

- Зведення до класичної задачі X3C про точне покриття 3-елементними множинами.
- Використовується конструкція GGJ (Garey, Graham та Johnson).

Теорема

Для одного БПЛА $K = 1$ і спільного лінійного руху $z_j(t) = v \cdot t + z_j(0)$ існує схема апроксимації з поліноміальним часом роботи.

Дворівневий алгоритм

Зводимо задачу до K незалежних підзадач:

1. Кластеризація цілей: $S = S_1 \sqcup \cdots \sqcup S_K$.
2. Маршрутизація $K = 1$: для кожного кластера S_k наближено розв'язується задача комівояжера з рухомими цілями.

Як кластеризувати?

- Друга теорема спонукає кластеризувати за напрямком v_j.
- Статична задача комівояжера спонукає кластеризувати за позицією $z_j(0)$.
- В роботі: $\int_0^\infty d(z_i(t), z_j(t))\rho(t)dt$.

Робастна кластеризація

Метод K-середніх

$$\sum_{k=1}^{K} \sum_{x \in S_k} d(x, \mu_k)^2 \to \min, \tag{8}$$

де $\mu_k = \frac{1}{|S_k|} \sum_{x \in S_k} x$ — середнє, яке може не належати множині S.

Поняття *середньої траєкторії* μ_k аналітично складне.

Метод K-медоїдів

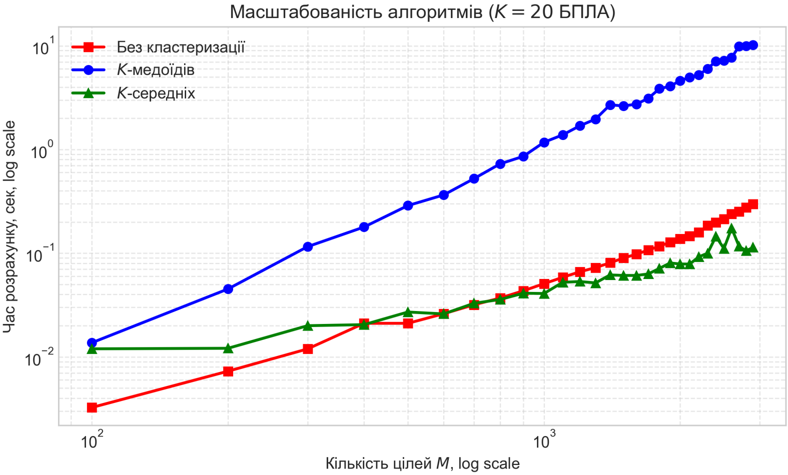
$$\sum_{k=1}^{K} \sum_{x \in S_k} d(x, m_k) \to \min, \tag{9}$$

де $m_k \in S_k$ — реальний об'єкт із вихідної множини.

Задача (9) є задачею негладкої оптимізації. Kaufman та Rousseeuw [3] розробили для неї наближений метод PAM.

Вплив 1% викидів ($M = 2000$, $K = 20$)

Показник	**Без класт.**	**K-середніх**	**K-медоїдів**
Середній час	320.4	286.1	**281.5**
Середня довжина	16528.1	9541.1	**9266.9**
Максимальний час	523.1	437.4	**384.2**
Максимальна довжина	22844.8	10371.1	**9932.8**


Масштабованість алгоритмів (K = 20 БПЛА)
Без кластеризації
K-медоїдів
K-середніх
Час розрахунку, сек, log scale
Кількість цілей M, log scale
10^1
10^0
10^-1
10^-2
10^2
10^3

План

Розділ 2. Максимум-плюс згортка

Розділ 3. Маршрутизація та кластеризація БПЛА

Розділ 4. Робастна торгівля акціями

Постановка задачі I

Розглядається задача алгоритмічної торгівлі одним активом на дискретному проміжку часу $i = 1, \ldots, n$.

Інтервальна невизначеність

- Точні ціни p_i невідомі, але належать заданим довірчим інтервалам:
$$p_i \in [l_i, r_i]. \qquad (10)$$
- Стан портфеля $s_i \in \{0, 1\}$ (0 — гроші, 1 — акція).

Важливо зазначити, що така модель коректно моделює лише нескінченно подільні акції. Вона відносно добре працює для криптовалютних активів які діляться дуже сильно: в одному біткоїні 100 мільйонів сатоші.

Постановка задачі II

Жаль (regret)

Знайти стратегію ALG $= (s_1, \ldots, s_n)$, що мінімізує максимальну різницю між прибутком OPT та нашим прибутком:

$$\text{regret}(\text{ALG}) = \max_P (\text{profit}(\text{OPT}, P) - \text{profit}(\text{ALG}, P)) \to \min, \qquad (11)$$

де profit(ALG, P) — реалізований прибуток стратегії ALG на траєкторії цін P.

Комісія та охолодження

Комісія

Вводиться фіксована комісія $f \geq 0$ за кожну операцію. Для отримання прибутку розмах ціни має перевищувати поріг:

$$p_{sell} - p_{buy} > 2f. \tag{12}$$

Охолодження

Вводиться період охолодження τ (у нашому випадку $\tau = 1$) після продажу активу. Тепер три стани: $s_i \in \{\text{cash}, \text{hold}, \text{cool}\}$. Переходи:

- $\text{cash} \to \{\text{cash}, \text{hold}\}$ — дозвіл на покупку або очікування.
- $\text{hold} \to \{\text{hold}, \text{cool}\}$ — дозвіл на продаж або утримання.
- $\text{cool} \to \{\text{cash}\}$ — примусове очікування.

Динамічне програмування

Оскільки дії OPT нам невідомі, ми не можемо обчислити точне значення жалю. Тому ми відстежуємо вектор умовних жалів для можливих станів OPT.

Вектор жалю

Для кожної стратегії на кроці i визначаємо вектор $v_i \in \mathbb{R}^2$:

$$v_i = \begin{pmatrix} v_{i,0} \\ v_{i,1} \end{pmatrix} = \begin{pmatrix} \text{regret if } s_{\text{OPT},i} = \text{cash} \\ \text{regret if } s_{\text{OPT},i} = \text{hold} \end{pmatrix} \tag{13}$$

Фронт Парето

Зберігаємо множину векторів F_i. Вектор u *домінує* над v, якщо він менший за обома компонентами:

$$u \preceq v \iff u_0 \leq v_0 \land u_1 \leq v_1. \tag{14}$$

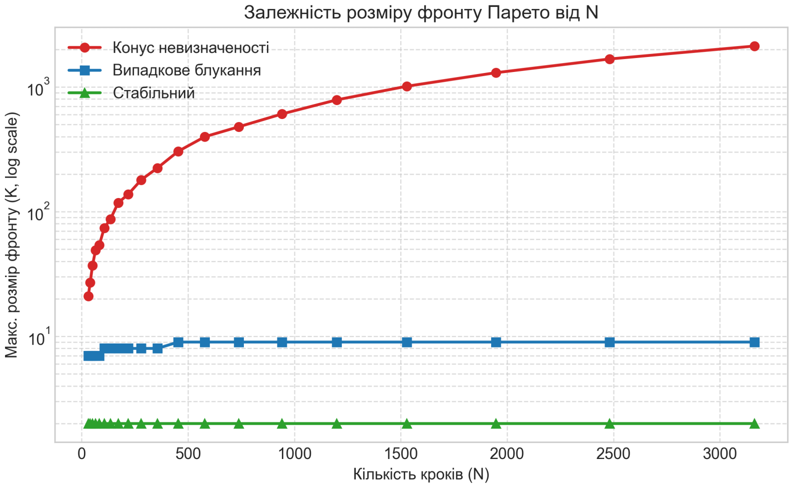
Стратегії, що домінуються іншими, видаляються.

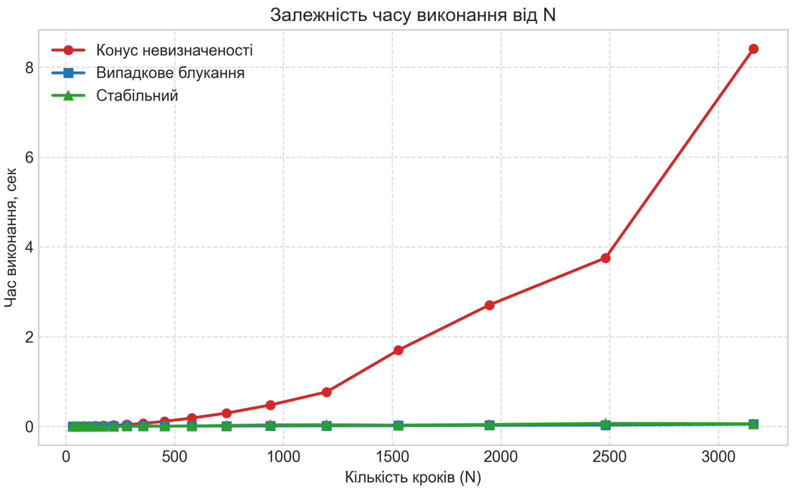
Розмір фронту Парето

Ключовим фактором швидкодії алгоритму є поточний розмір фронту Парето $K_i = |F_i|$.

Теоретичні оцінки

- Без відсікання кількість станів зростає експоненційно як $O(2^i)$. Для $n = 50$ розв'язати задачу неможливо.
- З відсіканням доведено, що фронт зростає лінійно, $K_i = O(i)$.
- Експериментально для декількох класів невизначеності встановлено $K_i = O(\log i)$.


Залежність розміру фронту Парето від N
Конус невизначеності
Випадкове блукання
Стабільний
Макс. розмір фронту (K, log scale)
10^3
10^2
10^1
0
500
1000
1500
2000
2500
3000
Кількість кроків (N)


Залежність часу виконання від N
Конус невизначеності
Випадкове блукання
Стабільний
Час виконання, сек
Кількість кроків (N)
0
2
4
6
8
0
500
1000
1500
2000
2500
3000

Основні наукові результати

1. **Вперше одержано** оцінки поширення похибки апроксимації $(\max, +)$-згортки на пов'язані задачі комбінаторної оптимізації.
2. **Експериментально визначено** практичні межі застосування алгоритму апроксимації Серанга для задач про 0-1 рюкзак та задачі про найдовший шлях. [7]
3. **Запропоновано новий метод** кластеризації рухомих об'єктів та успішно випробувано його на задачі маршрутизації БПЛА.
4. **Розроблено новий алгоритм** маршрутизації БПЛА та експериментально підтверджено його перевагу на 13%+ над класичними евристиками. [6]
5. **Вперше одержано** лінійні оцінки на розмір фронту Парето задачі про робастну торгівлю акціями з мінімальним жалем та її варіантів з комісією та кулдауном.
6. **Розроблено перший** поліноміальний алгоритм мінімізації жалю у робастній торгівлі акціями на основі метода динамічного програмування. [5]

Література і публікації I

D. Bremner, T. M. Chan, R. D. Demaine, J. Erickson, F. Hurtado, J. Iacono, S. Langerman, M. Patrascu, and P. Taslakian.
Necklaces, convolutions, and x+y, 2012.

M. Cygan, M. Mucha, K. Węgrzycki, and M. Włodarczyk.
On problems equivalent to (min,+)-convolution.
ACM Transactions on Algorithms, 15(1):25, January 2019.

L. Kaufman and P. J. Rousseeuw.
Partitioning around medoids (program pam).
In *Finding Groups in Data: An Introduction to Cluster Analysis*, chapter 2, pages 68–125. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1990.

O. Serang.
A fast numerical method for max-convolution and the application to efficient max-product inference in bayesian networks.
Journal of Computational Biology, 22(8):770–783, August 2015.

Література і публікації II

N. M. Skybytskyi.
Robust time to buy and sell stock.
Journal of Numerical and Applied Mathematics, 1(1):90–100, 2025.

N. M. Skybytskyi.
Two-level algorithm for the uavrp with moving targets.
Cybernetics and Computer Technologies, 4(4):29–36, December 2025.

N. M. Skybytskyi and K. I. Denysov.
Computational equivalence of one-dimensional quota tsp variant and (min, +) convolution.
Journal of Numerical and Applied Mathematics, 2(2):62–67, 2024.

А.А. Чикрий.
Конфликтно-управляемые процессы.
Наукова думка, Киев, 1992.

Дякую за увагу!