Журнал обчислювальної та прикладної математики

2024, № 2

Journal of Numerical & Applied Mathematics

УДК 519.1, 519.7

MSC 68Q17, 68Q25

COMPUTATIONAL EQUIVALENCE OF ONE-DIMENSIONAL QUOTA TSP VARIANT AND (MIN, +) CONVOLUTION

N. M. Skybytskyi, K. I. Denysov

Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, E-mail: n.skybytskyi@knu.ua, denisov_k@knu.ua

ЕКВІВАЛЕНТНІСТЬ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ ОДНОВИМІРНОГО ВАРІАНТА ЗАДАЧІ КОМІВОЯЖЕРА З КВОТОЮ ТА (MIN, +) ЗГОРТКИ

Н. М. Скибицький, К. І. Денисов

Факультет комп'ютерних наук та кібернетики, Київський національний університет імені Тараса Шевченка, Київ, Україна, E-mail: n.skybytskyi@knu.ua, denisov_k@knu.ua

Abstract. The paper considers a variant of the one-dimensional quota traveling salesman problem and its connection to the (min, +) convolution problem. We propose fine-grained reductions between these problems, resulting in a conditional lower bound on the computational complexity of the quota traveling salesman problem variant. We highlight the role of convexity in both problems and its connection to the proposed reductions.
Keywords: traveling salesman problem, min-plus-convolution, computational complexity, fine-grained reduction.

Анотація. У статті досліджено одновимірний варіант задачі комівояжера з квотою та її зв'язок із задачею про (min, +) згортку. Запропоновано ефективні способи звести ці задачі одна до одної. Отримано умовну нижню оцінку на обчислювальну складність варіанта задачі комівояжера з квотою. Окремо виділено практичне значення опуклості в обох задачах та її зв'язок із запропонованими способами зведення.
Ключові слова: задача комівояжера, min-плюс-згортка, обчислювальна складність, обчислювально ефективні зведення.

Вступ

Теорія точної обчислювальної складності займається отриманням верхніх та нижніх оцінок обчислювальної складності задач, для яких вже відомі алгоритми із поліноміально залежним від розміру вхідних даних часом роботи (поліноміальний клас складності). Зазначимо, що в літературі також зустрічається назва «теорія тонкої обчислювальної складності» (eng. *fine-grained complexity theory*).

Williams [1] наводить огляд деяких новітніх результатів у теорії точної обчислювальної складності. Здебільшого, для отримання оцінок пропонуються ефективні зведення до фундаментальних задач, для яких такі оцінки вже відомі. Такий підхід дозволяє пояснити практичну складність великих екземплярів деяких практичних задач, які належать до поліноміального класу складності.

Водночас багато задач у теорії точної обчислювальної складності залишаються нерозв'язаними. Williams [2] наводить огляд таких задач і обговорює їхнє значення для подальшого розвитку галузі.

Однією з таких фундаментальних задач є задача про (min, +) згортку. У цій задачі задано дві числові послідовності $a = (a_0, a_1, \ldots, a_{n-1})$ та $b = (b_0, b_1, \ldots, b_{n-1})$ однакової довжини n і потрібно обчислити послідовність $c = (c_0, c_1, \ldots, c_{2n-2})$, для якої

$$c_k = \min_{i+j=k} (a_i + b_j). \tag{1}$$

Умова $i + j = k$ на суму індексів нагадує операцію згортки функцій, звідки задача й отримала свою назву.

Наївний алгоритм для цієї задачі має обчислювальну складність $\Theta(n^2)$. Попри суттєві зусилля, досі не було знайдено алгоритму з обчислювальною складністю $O(n^{2-\varepsilon})$ для жодного $\varepsilon > 0$. Bremner та ін. [3] запропонували використовувати це для отримання умовних нижніх оцінок інших задач.

Cygan, Mucha, Węgrzycki та Włodarczyk [4] отримали ряд зведень до та від задачі про (min, +) згортку, чим суттєво зміцнили статус цієї задачі як фундаментальної у теорії точної обчислювальної складності.

Іншою фундаментальною задачею у теорії складності алгоритмів є задача комівояжера. Багато її варіантів можуть бути розв'язані за поліноміальний час. Зокрема, de Berg, Buchin, Jansen та Woeginger [5] проводять точний аналіз складності декількох варіантів задачі комівояжера.

Важливо зазначити, що для певних варіантів задачі з додатковими властивостями існують швидші алгоритми. Зокрема Bringmann та Cassis [6] дослідили зв'язок задачі про (min, +) згортку та деяких варіацій задачі про рюкзак, а також субадитивних послідовностей. Властивості обмеженості та монотонності послідовностей a та b відіграють важливу роль у їхньому дослідженні. Уточнимо припущення Bremner та ін.:

Лема 1 (припущення)**.** *Для задачі про (min, +) згортку у постановці* (1) *з додатковою умовою* $-W \leq a_i, b_j \leq W$ *для усіх* $i, j = 0, 1, \ldots, n-1$ *не існує алгоритму з обчислювальною складністю* $O(n^{2-\varepsilon}\mathrm{polylog}(W))$ *для жодного* $\varepsilon > 0$. *Тут* $\mathrm{polylog}(W)$ *позначає поліном від логарифма* W.

Окрім обмеженості вхідних чисел, велике значення відіграє опуклість послідовностей a та b, або ж принаймні однієї з них. Eppstein, Galil та Giancarlo [7] запропонували методи загальної оптимізації для задач з опуклими послідовностями задовго до того, як були отримані більшість результатів у задачі про (min, +) згортку.

Втім, ці методи дозволяють розв'язувати задачу про згортку за час $O(n \log n)$ за умови, що хоча б одна із послідовностей a та b опукла. Bringmann та Cassis [8] використали це і дослідили значення опуклості для зв'язку задачі про (min, +) згортку та варіацій задачі про рюкзак.

Мета нашої роботи полягає в дослідженні зв'язку одновимірного варіанта задачі комівояжера з квотою та задачі про (min, +) згортку. Розділ 1 містить постановку задачі та основні позначення. У розділі 2 формулюються та доводяться основні результати.

1. Постановка задачі

Задача k-TSP, також відома як задача комівояжера з квотою, полягає у пошуку найкоротшого маршруту, який проходить через k з n заданих точок у метричному просторі. Найпоширенішим прикладом метричного простору є Евклідів d-вимірний простір. Найпростішому з Евклідових просторів відповідає $d = 1$, тобто сам простір є звичайною числовою прямою.

Одновимірна задача k-TSP (1D k-TSP) полягає у знаходженні найкоротшого маршруту, що проходить через k з n заданих точок на числовій осі. Задача розбивається на два кроки: (i) вибір підмножини з k точок та (ii) пошук найкоротшого маршруту, що проходить через усі вибрані точки.

Лема 2. *Оптимальний розв'язок відвідує обрані k точок у впорядкованому за незростанням чи неспаданням порядку.*

Доведення. Нехай координати обраних точок $x_0 \le x_1 \le \ldots \le x_{k-1}$, а розв'язок відвідує їх у порядку $x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}$. Позначимо j та l як індекси, для яких $i_j = 0$, $i_l = k-1$. Без обмеження загальності $j \le l$. Тоді виконуються нерівності

$$\begin{aligned}
&|x_{i_j} - x_{i_{j+1}}| + |x_{i_{j+1}} - x_{i_{j+2}}| + \ldots + |x_{i_{l-1}} - x_{i_l}| \ge \\
&\ge |x_{i_j} - x_{i_{j+2}}| + \ldots + |x_{i_{l-1}} - x_{i_l}| \ge \\
&\ge |x_{i_j} - x_{i_l}| = |x_0 - x_{k-1}|.
\end{aligned}$$

У першому переході ми використали нерівність трикутника, а у другому – принцип математичної індукції. Рівність в усіх нерівностях трикутника досягається лише за умов $x_{i_j} \le x_{i_{j+1}} \le \ldots \le x_{i_l}$ або $x_{i_j} \ge x_{i_{j+1}} \ge \ldots \ge x_{i_l}$. Також зрозуміло, що зайвих точок бути не повинно, а тому $j = 0$ і $l = k-1$ (точки відвідані за неспаданням) або ж $j = k-1$ і $l = 0$ (точки відвідані за незростанням). □

Лема 3. *Оптимальний розв'язок вибирає k послідовних точок.*

Доведення. Впорядкуємо всі точки за неспаданням: $x_0 \le x_1 \le \ldots \le x_{n-1}$. Нехай були вибрані точки $x_{i_0} \le \ldots \le x_{i_{k-1}}$. З попередньої леми випливає, що оптимальний шлях між ними має довжину $x_{i_{k-1}} - x_{i_0}$. Позначимо $i_0 = j$, $i_{k-1} = l$, тоді ми мінімізуємо $x_l - x_j$ за умови $l - j \ge k-1$. З впорядкованості точок випливає нерівність $x_l - x_j \ge x_{j+k-1} - x_j$, а останній вираз якраз-таки позначає довжину шляху між k послідовними точками. □

Отже, задачу можна розв'язати, упорядкувавши точки та обчисливши значення

$$y_k = \min_{i=0..n-k} (x_{i+k-1} - x_i). \tag{2}$$

Часова складність цього алгоритму визначається етапом сортування. Надалі припустимо, що точки вже впорядковані. Тоді оптимальна часова складність алгоритму становить $\Theta(n)$.

Надалі ми розглянемо такий варіант одновимірної задачі k-TSP: за впорядкованою числовою послідовністю $x = (x_0, x_1, \ldots, x_{n-1})$, необхідно вивести числову послідовність $y = (y_1, y_2, \ldots, y_n)$, тобто розв'язати задачу k-TSP для всіх значень $k = 1, \ldots, n$. Нову задачу ми назвали 1D all k TSP. Наївне застосування попереднього алгоритму окремо для кожного k призводить до алгоритму з часовою складністю $\Theta(n^2)$ для нового варіанта.

У наступному розділі ми опишемо алгоритм для цієї задачі з часовою складністю $o(n^2)$ та покажемо, що у припущенні леми 1 не існує алгоритму з часовою складністю $O(n^{2-\varepsilon}\text{polylog}(W))$ для жодного $\varepsilon > 0$. Ми досягнемо цих результатів побудувавши спосіб звести цю задачу до (min, $+$) згортки та навпаки за майже лінійний час.

2. Ефективні зведення між задачами

Теорема 1. *Існує алгоритм зі складністю $o(n^2)$ для задачі 1D all k TSP.*

Доведення. Зведемо задачу 1D all k TSP до задачі про (min, $+$) згортку. Нам задана числова послідовність $x = (x_0, x_1, \ldots, x_{n-1})$ і потрібно побудувати послідовності a та b. Визначимо їх як $a = (x_{n-1}, \ldots, x_1, x_0)$ та $b = (-x_0, -x_1, \ldots, -x_{n-1})$. Іншими словами, $a_i = x_{n-1-i}$, $b_j = -x_j$. Таке перетворення займає час $\Theta(n)$.

Розв'яжемо задачу про (min, $+$) згортку для послідовностей a та b. Отримаємо послідовність c зі значеннями:

$$c_k = \min_{i+j=k} (a_i + b_j) = \min_{i+j=k} (x_{n-1-i} - x_j).$$

Зробимо заміну $i = k - j$ та отримаємо:

$$c_k = \min_{j=0..k} (x_{j+n-k-1} - x_j).$$

У вступі ми ввели позначення y_{n-k} для цього виразу. Це означає, що обчисливши послідовність $c = (c_0, c_1, \ldots, c_{n-1})$ ми можемо відновити значення $y_{n-0} = y_n, y_{n-1}, \ldots, y_{n-(n-1)} = y_1$, тобто усю числову послідовність $y = (y_1, y_2, \ldots, y_n)$. Це перетворення займає час $\Theta(n)$.

Тобто, кожен алгоритм для задачі про (min, $+$) згортку може бути перетворений на алгоритм для задачі 1D all k TSP за лінійний час. Враховуючи, що алгоритми для загальної задачі про (min, $+$) згортку мають більшу часову складність, це означає, що кожен алгоритм для задачі про (min, $+$) згортку може бути перетворений на алгоритм для задачі 1D all k TSP з такою ж часовою складністю.

Зокрема, у [3] було запропоновано алгоритм для задачі про (min, $+$) згортку зі складністю $O(n^2(\log\log n)^3/\log^2 n) = o(n^2)$. □

Теорема 2. *У припущенні леми 1 для задачі 1D all k TSP не існує алгоритму з часовою складністю $O(n^{2-\varepsilon}\mathrm{polylog}(W))$ для жодного $\varepsilon > 0$.*

Доведення. Зведемо задачу про (min, +) згортку до 1D all k TSP. Нам задані числові послідовності $a = (a_0, a_1, \ldots, a_{n-1})$ та $b = (b_0, b_1, \ldots, b_{n-1})$ і потрібно обчислити послідовність c визначену в (1). Розглянемо послідовність x визначену як

$$x = (-a_0, \ldots, -a_{n-1}, b_{n-1}, \ldots, b_0).$$

Іншими словами, $x_i = -a_i$ для $i = 0, 1, \ldots, n-1$ та $x_{n+j} = b_{n-1-j}$ для $j = 0, 1, \ldots, n-1$.

Розв'яжемо задачу 1D all k TSP для цієї послідовності. Отримаємо послідовність y зі значеннями визначеними в (2). Ми отримали $2n$ значень, проте надалі будемо використовувати лише n з них, а саме $y_{n+1}, y_{n+2}, \ldots, y_{n+n} = y_{2n}$. Для них виконується рівність

$$y_{n+k} = \min_{i=0..n-k}(x_{i+n+k-1} - x_i).$$

Використаємо визначення x та отримаємо:

$$y_{n+k} = \min_{i=0..n-k}(b_{n-k-i} + a_i).$$

Зауважимо, що $n - k - i + i = n - k$, тобто $y_{n+k} = c_{n-k}$ за означенням (1).

Щоправда, у формулюванні (2), задача 1D all k TSP вимагає впорядкованості послідовності x, а побудована нами послідовність не обов'язково має таку властивість. Модифікуємо послідовність, визначивши $x'_i = x_i + 2iW$. Нагадаємо, що абсолютні значення a_i та b_j обмежені зверху числом W для усіх $i, j = 0, 1, \ldots, n-1$. Це забезпечує впорядкованість x', оскільки:

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

Використане перетворення також призводить до зміни

$$y'_k = y_k + 2(k-1)W,$$

що дозволяє відновити початкову послідовність y за заданими y' та W.

Щоправда, у конструкції x' є ще одна проблема: її елементи не обов'язково належать сегменту $[-W, W]$. Натомість можемо записати нерівності

$$x'_i = x_i + 2iW \geq (-W) + 0W = -W \geq -4nW,$$
$$x'_j = x_j + 2jW \leq (+W) + 2(2n-1)W = (4n-1)W \leq 4nW,$$

з яких випливає, що x'_i належить сегменту $[-4nW, 4nW]$ для усіх $i = 0, 1, \ldots, 2n-1$.

Оцінимо тепер складність такого зведення. Ми звели задачу про (min, +) згортку з параметрами (n, W) до задачі 1D all k TSP з більшими параметрами $(2n, 4nW)$. Проведемо асимптотичний аналіз складності отриманої задачі:

$$\begin{aligned}
&O((2n)^{2-\varepsilon}\mathrm{polylog}(4nW)) = O(2^{2-\varepsilon}n^{2-\varepsilon}\mathrm{polylog}(4nW)) = \\
&= O(n^{2-\varepsilon}\mathrm{polylog}(4nW)) = O(n^{2-\varepsilon}(\mathrm{polylog}(4n) + \mathrm{polylog}(W))) = \\
&= O(n^{2-\varepsilon}(o(n^{\varepsilon/2}) + \mathrm{polylog}(W))) = O(n^{2-\varepsilon/2}\mathrm{polylog}(W)).
\end{aligned}$$

де у першому рядку ми винесли константу, а у другому та третьому скористалися асимптотичними властивостями полі-логафмічних функцій.

Отже, кожен алгоритм зі складністю $O(n^{2-\varepsilon}\text{polylog}(W))$ для задачі 1D all k TSP може бути перетворений на алгоритм для задачі про (min, $+$) згортки зі складністю $O(n^{2-\varepsilon/2}\text{polylog}(W))$. Проте існування останнього суперечить припущенню леми 1. $\square$

Зауваження. Зазначимо, що перетворення $x'_i = x_i + 2iW$ зберігає опуклість послідовності x. Справді,

$$\begin{aligned}
&x'_{i+1} - 2x'_i + x'_{i-1} = \\
&= (x_{i+1} + 2(i+1)W) - 2(x_i + 2iW) + (x_{i-1} + 2(i-1)W) = \\
&= (x_{i+1} - 2x_i + x_{i-1}) + (2(i+1)W - 4iW + 2(i-1)W) = \\
&= x_{i+1} - 2x_i + x_{i-1},
\end{aligned}$$

тобто другі різниці послідовностей x та x' однакові. Нагадаємо, що одним з критеріїв опуклості послідовності є невід'ємність других різниць.

Також опуклість зберігається у зведенні задачі 1D all k TSP до задачі про (min, $+$) згортку, що дозволяє використати техніки запропоновані Eppstein, Galil та Giancarlo [7] до опуклої задачі 1D all k TSP для побудови алгоритму з часовою складністю $O(n \log n)$.

Зауваження. Якщо застосувати наближений алгоритм, а не точний, то точність першого зведення залишиться незмінною, а ось точність другого зведення необхідно оцінити додатково. Ми розглянемо це питання більше детально у наступних роботах.

Література

1. Williams V. On some fine-grained questions in algorithms and complexity. *Proceedings of the ICM.* 2018. P. 3447–3487.
2. Williams V. Some Open Problems in Fine-Grained Complexity. *SIGACT News.* 2018, Vol. 49, No. 4, P. 29–35.
3. Bremner D. et al. Necklaces, Convolutions, and X+Y. *Algorithmica.* 2012, Vol. 69.
4. Cygan M., Mucha M., Węgrzycki K., Włodarczyk M.. On Problems Equivalent to (min,+)-Convolution. *ACM Trans. Algorithms.* 2019, Vol. 15, No. 1.
5. de Berg M., Buchin K., Jansen B., Woeginger G. Fine-grained Complexity Analysis of Two Classic TSP Variants. *ACM Trans. Algorithms.* 2021, Vol. 17, No. 1.
6. Bringmann K., Cassis A. Faster Knapsack Algorithms via Bounded Monotone Min-Plus-Convolution. *arXiv:1212.4771.* 2022.
7. Eppstein D., Galil Z., Giancarlo R. Speeding up dynamic programming. *In Proceedings of the 29th Annual SFCS.* 1988, P. 488–496.
8. Bringmann K., Cassis A. Faster 0-1-Knapsack via Near-Convex Min-Plus-Convolution. *arXiv:2305.01593.* 2023.

Надійшла: 11.11.2024 / Прийнята: 13.12.2024