Issue 17, 2025

https://doi.org/10.23939/sisn2025.17.138

УДК 004.02

АНАЛІЗ ШВИДКОДІЇ МЕТОДУ K-MEANS ДЛЯ ДЕКОМПОЗИЦІЇ ЗАДАЧІ КОМІВОЯЖЕРА ВЕЛИКИХ РОЗМІРНОСТЕЙ

Роман Кутельмах

Національний університет "Львівська політехніка",
кафедра програмного забезпечення, Львів, Україна
E-mail: roman.k.kutelmakh@lpnu.ua, ORCID: 0009-0008-6499-1161

Декомпозиція задачі базується на кластеризації вхідної множини точок відомим методом k-means та алгоритмі розширення часткового розв'язку у кластерах. Саме k-means застосовано для поділу множини вхідних даних для задачі комівояжера великих розмірностей на менші підзадачі. Обгрунтовано доцільність його використання для зменшення розмірності. На основі проведених експериментів запропоновано застосування ієрархічної версії алгоритму для задач розмірністю більше мільйона точок, проаналізовано доцільність застосування алгоритму k-means для відомих тестових задач комівояжера великих та дуже великих розмірностей. Вхідними даними служать задачі з колекцій TSPLIB та DIMACS TSP Challenge. Проведено експерименти для задач розмірністю від 100 тисяч до 10 мільйонів точок. Згідно з одержаними результатами експериментів (час роботи алгоритму кластеризації задачі на 10 мільйонів точок становить трохи більше 8 хвилин), зроблено висновок про можливість застосування такої кластеризації для декомпозиції задачі.

Ключові слова: кластеризація, алгоритм, велика розмірність, k-means, задача комівояжера, поділ графу

Постановка проблеми

Статтю присвячено дослідженню можливостей декомпозиції задачі комівояжера великих розмірностей, що є важливою та актуальною задачею на сьогодні. Будучи теоретичною задачею, вона знаходить своє застосування на практиці. 3D-друк та аерофотозйомка дронами — це нові сфери її застосування, які можуть призводити до появи задач дуже великих розмірностей. Обчислювальні затрати для отримання високоякісних розв'язків у таких випадках повинні бути дуже обмеженими. Інші сфери застосування охоплюють секвенування ДНК і рентгенівську кристалографію. Задачу комівояжера, що передбачає "географічні" елементи (міста і дороги), можна використати як модель для пошуку мінімальної довжини контактів, які з'єднують виводи у надвеликій інтегральній схемі НВІС (VLSI).

Для багатьох задач великих розмірностей у галузі дослідження операцій, таких як зокрема задачі транспорту та логістики, доцільним є розбиття вхідної задачі на менші підзадачі. Зазвичай для цих цілей застосовуються різні методи кластеризації. При наявності вимог роботи в реальному часі, важливість швидкодії розбиття вхідної задачі на менші підзадачі зростає. Вибір конкретного методу кластеризації залежить від самого характеру задачі та заданих обмежень. У статті досліджено швидкодію відомого алгоритму кластеризації k-means (Arthur & Vassilvitskii, 2007; Fränti & Virmajoki, 2006) для задачі комівояжера (ЗК) (Кутельмах, 2024; Applegate et al., 2006) великих розмірностей (Кутельмах, 2024; Star Tours, 2023; Gaia DR2, 2023; Drori et al., 2020) у двовимірному просторі.

Проаналізовано швидкодію алгоритму для цих задач та обґрунтовано доцільність застосування ієрархічної версії алгоритму k-means для задач розмірністю більше мільйона точок з метою зменшення часу роботи алгоритму.

Метою даного дослідження є визначення швидкодії алгоритму k-means для розбиття великих наборів даних. Алгоритм кластеризації розглядається як початкова процедура при декомпозиції ЗК великих розмірностей на менші підзадачі (рис. 1). В якості тестових даних обрано задачі з бібліотеки TSPLIB (1995) та DIMACS TSP Challenge (2001).

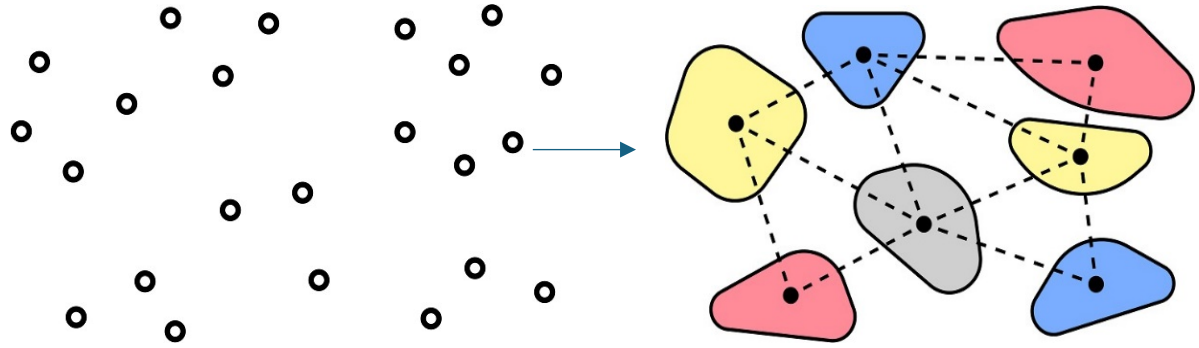

Рис. 1. Розбиття вхідної множини точок на кластери та формування кластерного графу

Досліджено відомі тестові задачі розмірністю від 10^5 до 10^7 точок з різним характером розподілу. ЗК формулюють наступним чином. Задано множину точок V, де $|V| = N \geq 3$ та кожна точка характеризується координатами в k-вимірному просторі ($k \geq 2$), наприклад: $V = (v_1, v_2, \ldots, v_N)$, де $v_i = (a_1, a_2, \ldots, a_k)$ та функцією відстані *distance*: V x $V \rightarrow$ R, де *distance*(v_i, v_j) для $1 \leq i \leq N$, $1 \leq j \leq N$, $i \neq j$ є метрикою (наприклад, евклідовою). Необхідно знайти найкоротший замкнутий маршрут T, який відвідує кожну точку з V лише один раз і повертається в початкову точку. Тобто, необхідно знайти T, такий що *length* T є мінімальною, де $T = <v_1, v_2, \ldots, v_N>$ і $v_i \in V$; $|T| = N$, та *length* $T = \sum_{i=1}^{N-1} distance(v_i, v_{i+1}) + distance(v_N, v_1)$ є функцією довжини маршруту. Задача вважається симетричною, якщо *distance*(v_i, v_j) = *distance*(v_j, v_i) для $1 \leq i \leq N$, $1 \leq j \leq N$, $i \neq j$.

Аналіз останніх досліджень та публікацій

Декомпозиції ЗК присвячено багато наукових праць (Kutelmakh, 2024; Applegate et al., 2006). Відомо, що задача, для яких розв'язок можна перевірити за поліноміальний час є складною і основною проблемою при її декомпозиції є те, що втрачається можливість отримання оптимального розв'язку (optimal solution). Це зумовлено тим фактом, що суттєво звужується простір (search space) можливих розв'язків (feasible solutions). До того ж, відомі евристичні алгоритми переважно теж обмежують множину ребер-кандидатів (candidate set) для забезпечення кращої швидкодії евристичного алгоритму.

Відомо, що для розбиття вхідної множини точок для ЗК доцільним є застосування різних методів кластеризації та поділу графу (Taillard & Helsgaun, 2019). До них належать алгоритми, що базуються на мінімальному зв'язному дереві (Jothi et al., 2018; Gagolewski et al., 2024), методи розбиття графу (Kernighan & Lin, 1970), розбиття маршруту на сегменти, підмножини тріангуляції Делоне (Guibas & Stolfi, 1985), декомпозиція Карпа (Darte & Vivien, 1995), k-d-дерево (Bentley, 1990) та дерево квадрантів (Drori et al., 2020). Альтернативним підходом до розв'язування ЗК великих розмірностей є застосування технологій нейронної комбінаторної оптимізації (Khamis, 2024).

Алгоритм кластеризації k-means формує k кластерів з використання середніх (mean) значень для визначення близькості між точками. Центроїд кожного кластера є середнім усіх його точок, обчисленим за кожною ознакою та слугує точкою відліку для визначення відстані до інших точок.

Алгоритм має малу обчислювальну складність, оскільки використовує простий ітеративний процес, у якому центроїди постійно оновлюються і модифікуються з метою мінімізації відстаней. Ця простота робить k-means ефективним з точки зору швидкодії для великих наборів даних.

Формулювання цілі статті

Метою статті є аналіз алгоритму k-means для декомпозиції ЗК великих розмірностей. Основним параметром дослідження є час поділу вхідної множини точок на менші підзадачі (кластери). Стрімке зростання розмірності задачі в таких галузях, як транспортні задачі, 3D-друк, аерофотозйомка та інші сфери з додатковими обмеженнями обчислювальних ресурсів потребує ефективної та швидкої декомпозиції. Очікується, що результати експериментів підтвердять можливість використання методу k-means (зокрема, ієрархічної версії) на етапі кластеризації великих наборів даних (від сотень тисяч до десятків мільйонів точок). Це стане основою для наступного етапу - кластеризації ЗК надвеликих розмірностей (більше мільярда точок) в тривимірному просторі, дослідження інших методів кластеризації з метою їх поєднання, та розв'язання часткових ЗК евристичними, наближеними або точними методами.

Виклад основного матеріалу

У запропонованій методології декомпозиції ЗК вхідна область поділяється на підмножини меншої розмірності (кластери). Суть декомпозиції в даній роботі полягає у поділі вхідної множини точок на кластери (підмножини) суттєво меншого розміру, що дасть змогу розв'язувати задачу окремо в кластерах та певним чином об'єднувати розв'язки (на наступному етапі). Для утворених підмножин формується кластерний граф. За допомогою спеціальних процедур приєднання часткових маршрутів у сусідніх кластерах (Kutelmakh et al., 2023) формується повний розв'язок задачі. Алгоритми кластеризації дозволяють розбити вхідну задачу на менші підзадачі. В даному дослідженні, результатом етапу кластеризації є сформована множина $C = (C_1, C_2, \ldots, C_K)$ з k кластерів, де кожен кластер в цій множині є підмножиною $V : C_i \in C, C_i \subset V, i \in \{1, \ldots, k\}, 1 \leq |C_i| < N, C_i = \{p_1, p_2, \ldots, p_n\}, p_i \in V$, а n є максимальною кількістю точок у кластері.

На рис. 2 показано результат кластеризації тестової задачі u1060 розмірністю 1060 точок (TSPLIB, 1995), а на рис. 3 показано результат кластеризації тестової задачі pla33810 розмірністю 33000 точок (TSPLIB, 1995). Рис. 4 ілюструє 30 кластерів для задачі Mona Lisa TSP Challenge (2008), що є прикладом конвертування зображення в набір точок з подальшим розв'язанням ЗК (Бош, 2022). На рис. 5 показано 4000 кластерів, сформованих алгоритмом k-means для задачі E10M.0 (DIMACS TSP Challenge, 2001).

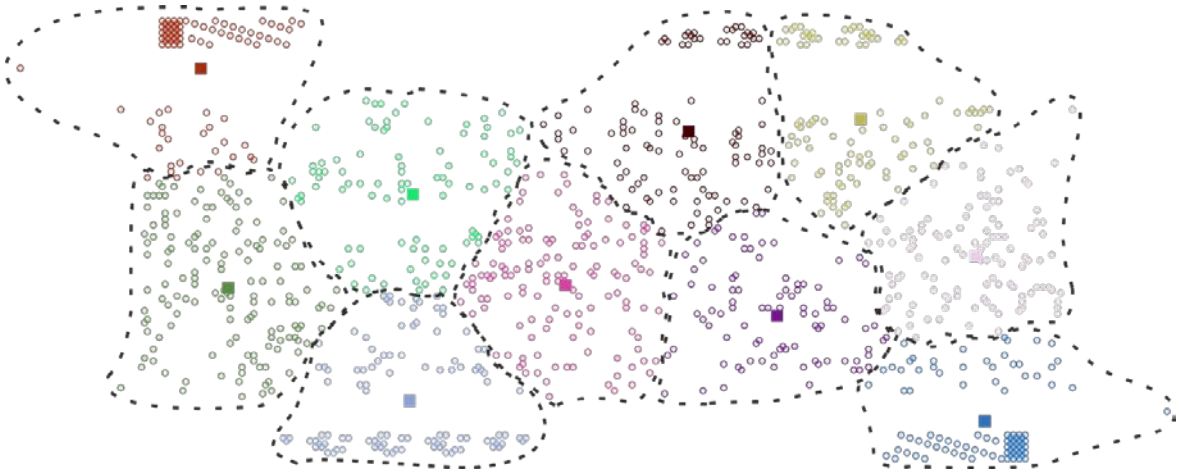

Рис. 2. 10 кластерів, одержаних алгоритмом k-means для задачі u1060 розмірністю 1060 точок

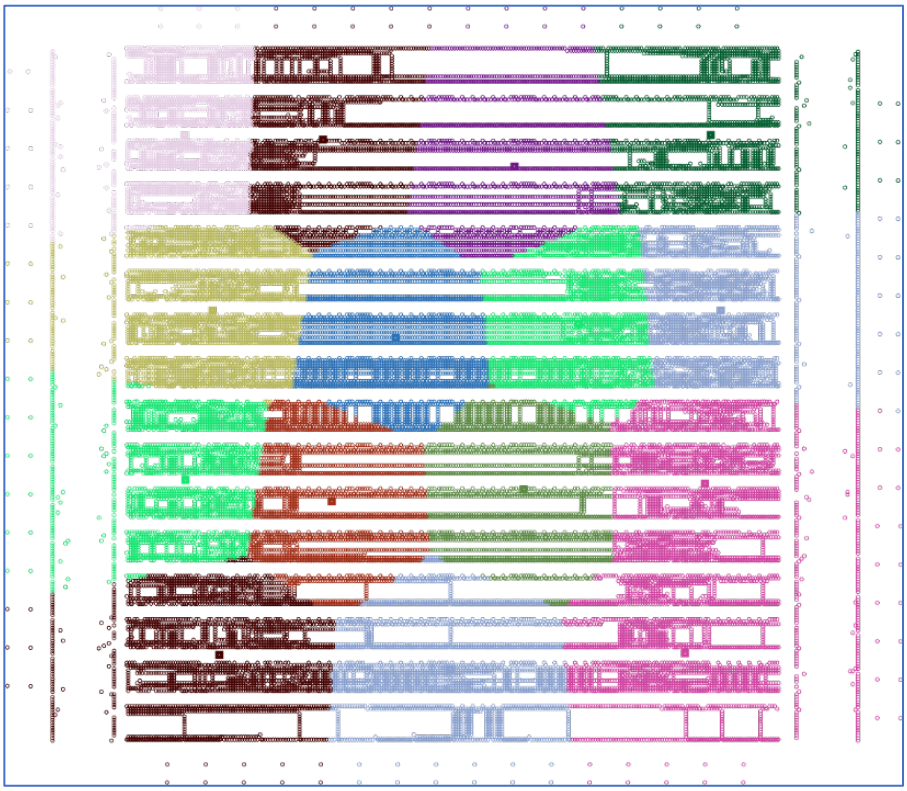

Рис. 3. 15 кластерів, сформованих алгоритмом k-means для задачі pla33810 розмірністю 33810 точок

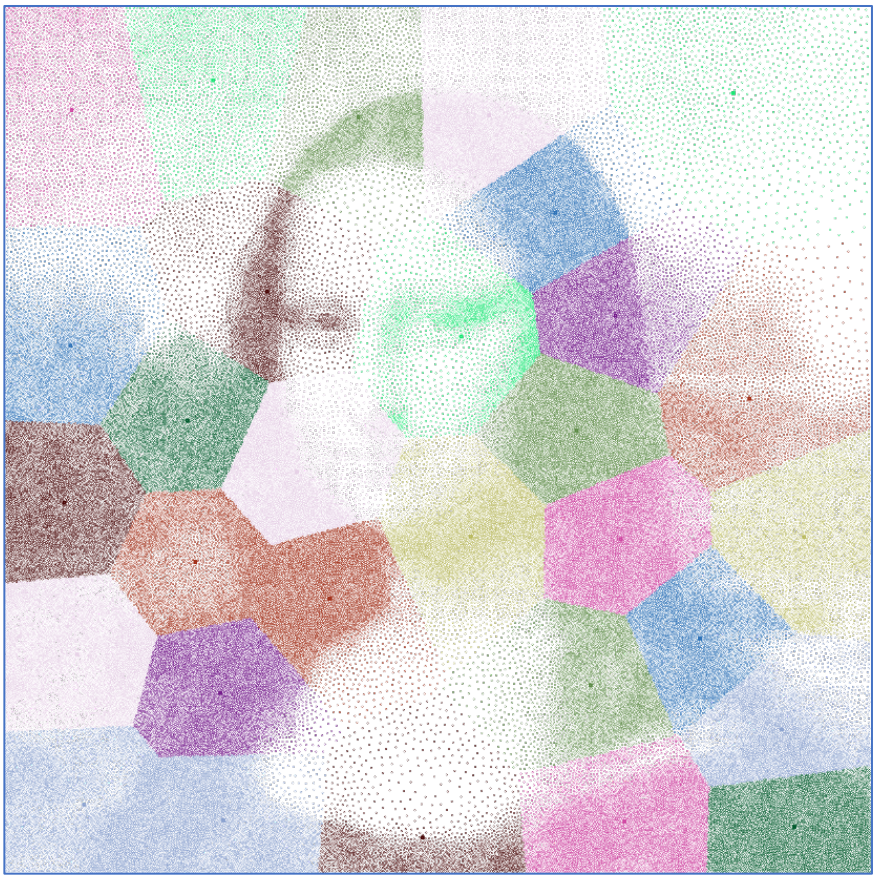

Рис. 4. 30 кластерів, сформованих алгоритмом k-means для задачі Mona Lisa розмірністю 10^5 точок

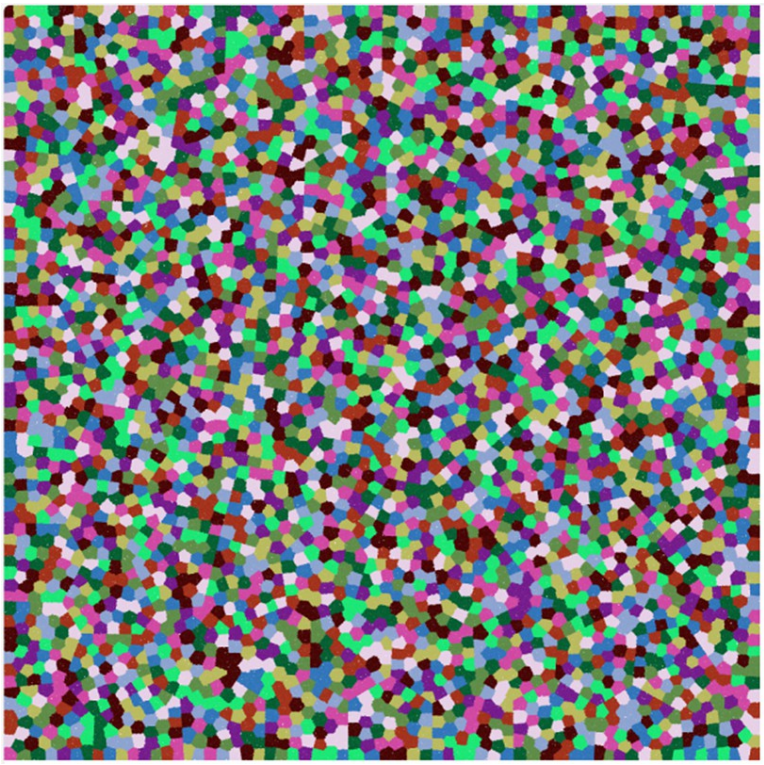

Рис. 5. 4000 кластерів, сформованих алгоритмом k-means для задачі E10M.0 розмірністю 10^7 точок

Для дослідження швидкодії алгоритму кластеризації обрано задачі з колекції DIMACS TSP Challenge (2001) розмірністю від 10^5 до 10 мільйонів точок. Також досліджено найбільшу задачу з бібліотеки TSPLIB – pla85900 (TSPLIB, 1995). Усі експерименти проводилися на ПК з Windows 11 (тактова частота 4,2 ГГц, 96 ГБ оперативної пам'яті). Код алгоритму k-means взято у (KmeansPlusPlus, 2017). Алгоритм k-means адаптовано до задач великих розмірностей завдяки реалізації автором ієрархічної версії (два рівні). Кількість кластерів задавалась таким чином, щоб розмір кластеру (кількість точок) був однаковим для кожної задачі. Таблиці 1 та 2 містять результати експериментів. Рис. 6 ілюструє графік залежності часу роботи алгоритму від розмірності для досліджених задач до 1 мільйона точок.

Таблиця 1

Результати дослідження алгоритму k-means для задач розмірністю 85900 – 3×10^6 точок

Тестова задача	Розмірність	Час роботи алгоритму, с	Кількість кластерів
pla85900	85900	0,69	26
monasisa100K	100000	1,3	30
E100k.0	100000	1,3	30
usa115475	115475	1,1	35
earring200K	200000	4,7	60
ara238025	238025	5,4	71
E316k.0	316000	11,6	95
lra498378	498378	26,5	150
lrb744710	744710	65	223
E1M.0	10^6	142	300
E3M.0	3×10^6	3618	900

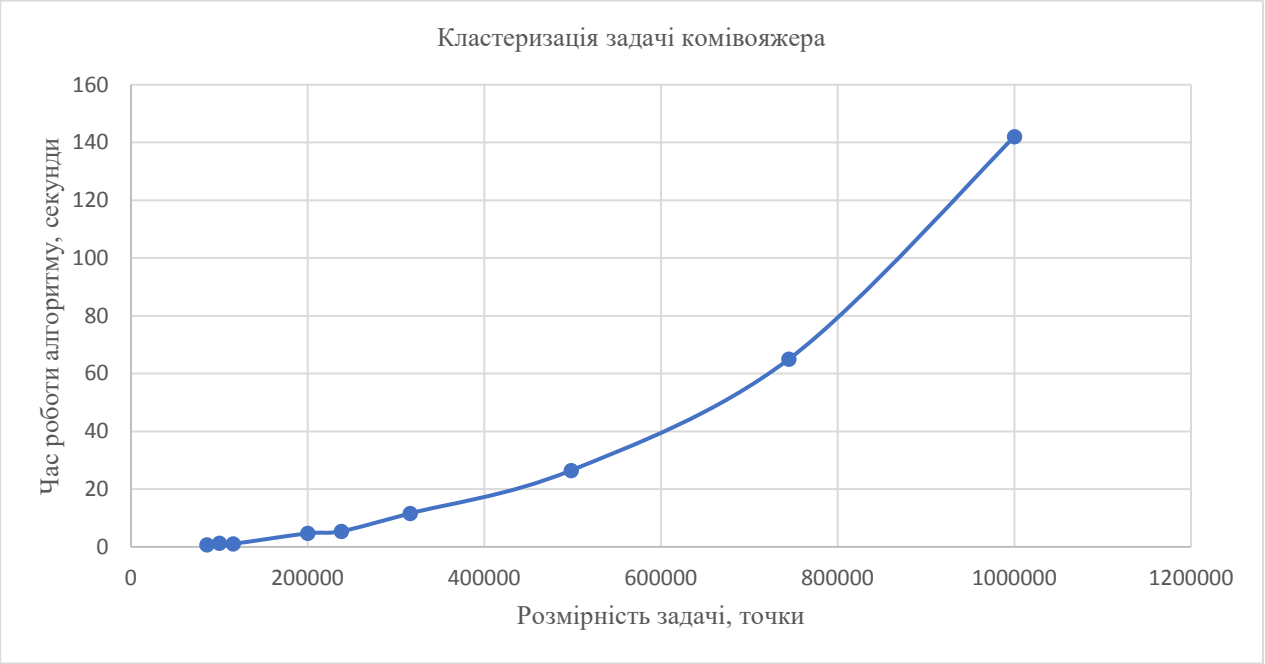

Рис. 6. Час роботи алгоритму для досліджених задач розмірністю 85900 – 10^6 точок

Таблиця 2

Результати дослідження ієрархічної версії алгоритму k-means для задач розмірністю 3 та 10 мільйонів точок

Тестова задача	Розмірність	Час роботи алгоритму, секунди	Кількість кластерів
E3M.0	3×10^6	102	900
E10M.0	10^7	521	3000

З результатів експериментів видно, що алгоритм кластеризації k-means доцільно застосовувати для декомпозиції ЗК великих розмірностей у двовимірному просторі. Для задач розмірністю від мільйона до 10 мільйонів час роботи ієрархічної версії алгоритму залишається прийнятним (до декількох хвилин). Це є важливим показником для можливості роботи в середовищах з обмеженими обчислювальними ресурсами. Збільшення кількості рівнів ієрархічної кластеризації k-means дасть змогу застосувати запропонований підхід для задач розмірністю сотні мільйонів та мільярди точок. При цьому очікується прийнятний час обчислень. У майбутньому передбачається дослідження паралельних версій алгоритму, а також інших алгоритмів кластеризації, зокрема тих, що передбачають реалізацію на графічних процесорах.

Висновки

Досліджено швидкодію алгоритму k-means при декомпозиції задачі комівояжера великих розмірностей у двовимірному просторі. Для задач розмірністю мільйон точок час роботи алгоритму складав 2,4 хвилини. Для задач розмірністю 10 мільйонів точок час роботи ієрархічного підходу склав 8,7 хвилини, що вказує на доцільність його застосування у середовищах з обмеженими обчислювальними ресурсами. Для задач розмірністю більше сотень мільйонів точок доцільним є застосування багаторівневої (більше 2 рівнів) ієрархічної кластеризації k-means.

Розглянутий алгоритм кластеризації не забезпечить оптимальний поділ з точки зору якості отриманого маршруту. Необхідно дослідити інші методи кластеризації (DBSCAN, OPTICS) та поділу графу (наприклад, алгоритм Кернігана-Ліна, вершинний сепаратор і т. д.). В майбутньому планується проведення цих досліджень.

REFERENCES

1. 1,331,906,450 stars Gaia DR2 (2023). https://www.math.uwaterloo.ca/tsp/star/gaia2.html
2. Applegate, D., Bixby, R., Chvátal, V., & Cook, W. (2006). *The traveling salesman problem: A computational study.* Princeton University Press.
3. Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). Society for Industrial and Applied Mathematics.
4. Bentley, J. L. (1990). Experiments on traveling salesman heuristics. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms* (pp. 91–99).
5. Bosch R. (2022), Opt Art. From mathematical optimization to visual design. Kharkiv. Fabula.
6. Darte, A., & Vivien, F. (1995). Revisiting the decomposition of Karp, Miller, and Winograd. *Parallel Processing Letters, 5*(1), 13–25. https://doi.org/10.1109/ASAP.1995.522901
7. DIMACS TSP Challenge (2001). http://archive.dimacs.rutgers.edu/Challenges/TSP/
8. Drori, I., Kates, B. J., Sickinger, W. R., Kharkar, A. G., Dietrich, B., Shporer, A., & Udell, M. (2020). GalaxyTSP: A new billion-node benchmark for TSP. *In Proceedings of the 1st Workshop on Learning Meets Combinatorial Algorithms @ NeurIPS 2020* (pp. 1–7). Vancouver, Canada.
9. Fränti, P., & Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition, 39*(5), 761–775.
10. Gagolewski, M., Cena, A., Bartoszuk, M., & Brzozowski, L. (2024). Clustering with minimum spanning trees: How good can it be? *Journal of Classification*, 1–23. https://doi.org/10.1007/s00357-024-09483-1
11. Guibas, L. J., & Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics, 4*(2), 75–123.
12. Jothi, R., Mohanty, S. K., & Ojha, A. (2018). Fast approximate minimum spanning tree based clustering algorithm. *Neurocomputing, 272*, 542–557. https://doi.org/10.1016/j.neucom.2017.07.038
13. Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal, 49*(2), 291–307. https://doi.org/10.1002/j.1538-7305.1970.tb01770.x
14. Khamis, A. (2024). *Optimization algorithms: AI techniques for design, planning, and control problems*. Manning.
15. KmeansPlusPlus (2017). *GitHub*. https://github.com/ieyjzhou/KmeansPlusPlus
16. Kutelmakh, R. K. (2024). *Solving NP-Hardness. Efficient Solution to the Traveling Salesman Problem* (Prof. Balylevych R. P., Ed.). Lviv. Lviv Polytechnic Publishing House.
17. Kutelmakh, R., Bazylevych, R., & Prasad, B. (2023). Solving large scale uniform traveling salesman problem by using partial solution expansion method. In *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)* (pp. 1–4). Lviv, Ukraine. https://doi.org/10.1109/CSIT61576.2023.10324172
18. Mona Lisa TSP Challenge (2008). https://www.math.uwaterloo.ca/tsp/data/ml/monalisa.html
19. Star Tours (2023). https://www.math.uwaterloo.ca/tsp/star/index.html
20. Taillard, É., & Helsgaun, K. (2019). Popmusic for the travelling salesman problem. *European Journal of Operational Research*, 272(2), 420–429. https://doi.org/10.1016/j.ejor.2018.06.039
21. TSPLIB (1995). http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

STUDY OF THE EFFECTIVENESS OF APPLYING THE K-MEANS METHOD TO DECOMPOSE LARGE-SCALE TRAVELING SALESMAN PROBLEMS

Roman Kutelmakh

Lviv Polytechnic National University,
Department of Software, Lviv, Ukraine
E-mail: roman.k.kutelmakh@lpnu.ua, ORCID: 0009-0008-6499-1161

The decomposition of the problem is based on clustering the input set of points using the well-known k-means method, combined with an algorithm for extending partial solutions within clusters. k-means clustering algorithm is examined for partitioning the input data set of large-scale TSP instances into smaller subproblems. The efficiency of using it to reduce problem size is substantiated. Based on

experiments, the application of a hierarchical version of the algorithm is proposed for problems with more than one million points. The paper analyzes the feasibility of using the k-means algorithm for well-known large and huge TSP test instances. The well-known collections of TSP instances (TSPLIB and the DIMACS TSP Challenge) are investigated. Experiments were conducted for problems ranging from 100 thousand to 10 million points. According to the experimental results (the clustering algorithm's runtime for a 10-million-point problem is slightly over 8 minutes), it was concluded that such clustering could indeed be used for problem decomposition. A hierarchical version of k-means clustering has been developed for very large-scale problems.

Keywords: clustering, algorithm, large scale, k-means, traveling salesman problem, subproblem, graph partitioning