УДК 519.857
DOI https://doi.org/10.32689/maup.it.2025.3.21

Марія СЕМАНЬКІВ
кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних систем,
Карпатський національний університет імені Василя Стефаника,
mariia.semankiv@pnu.edu.ua
***ORCID:** 0000-0002-1314-8923*

ВИКОРИСТАННЯ АЛГОРИТМУ ВЕЛЬЦЛЯ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

Анотація. *Стаття присвячена вирішенню однієї з NP-задач, а саме пошуку оптимального маршруту комівояжера для відвідання кожного із n заданих міст. Ефективні алгоритми розв'язання даної задачі дозволяють знаходити оптимальні маршрути навіть для великих наборів міст, що зменшує витрати часу, пального та ресурсів. Поєднання високої точності з малою обчислювальною складністю робить такі методи придатними для використання в реальних системах, де рішення потрібно приймати швидко.*

Мета роботи *– вдосконалення методу гілок та меж для розв'язання задачі комівояжера за рахунок застосування алгоритму Вельцля та використання мінімального охоплюючого кола (Minimum Enclosing Circle, MEC) як евристики для підсилення відсікаючих правил у даному методі.*

Методологія. *Алгоритм Вельцля – це класичний приклад ефективного геометричного методу з відсіками та евристикою, який можна легко включати до більш складних алгоритмів. У задачах розміщення, пакування, колізій, кластеризації тощо, алгоритм Вельцля може бути використаний як частина оціночної або відсікаючої функції: обчислити мінімальне коло, що містить підмножину об'єктів-міст, щоб оцінити обсяг/простір і якщо коло з новою точкою виходить за дозволені межі – відсікати гілку.*

Наукова новизна. *Автором запропоновано використання алгоритму Вельцля для прискорення точного алгоритму розв'язання задачі комівояжера. Вигода у знаходженні мінімального кола, що охоплює множину точок-міст на площині, полягає у отриманні більш тісної нижньої межі гілки в методі гілок та меж, у збільшенні кількості гілок у дереві пошуку, які відсічуться раніше, і як наслідок метод запрацює швидше. MEC відсікає ті гілки, у яких невідвідані міста лежать на великій відстані одне від одного, і навіть найоптимальніший добудований маршрут буде занадто дорогим. Це зменшує простір пошуку і прискорює алгоритм.*

Висновки. *Метод гілок та меж із використанням мінімального охоплюючого кола MEC має широкий спектр практичного застосування у задачах, де необхідно швидко отримати якісний маршрут з мінімальними обчислювальними витратами, якщо допускається невелике відхилення від оптимального розв'язку. Таким чином, використання MEC у поєднанні з методом гілок та меж є універсальним підходом, що поєднує точність математичної оптимізації та швидкість евристичних методів і може бути впроваджене в будь-якій сфері, де маршрутизація має велике практичне значення.*

Ключові слова: *задача комівояжера (TSP), алгоритм Вельцля, метод гілок та меж.*

Mariia SEMANKIV. THE USE OF WELZL'S ALGORITHM FOR SOLVING THE TRAVELING SALESMAN PROBLEM

Abstract. *This article is devoted to solving one of the NP-hard problems, namely finding the optimal traveling salesman route for visiting each of the n given cities. Efficient algorithms for solving this problem make it possible to find optimal routes even for large sets of cities, which reduces time, fuel, and resource costs. Combining high accuracy with low computational complexity makes such methods suitable for real-world systems where decisions must be made quickly.*

The purpose of this work *is to improve the branch-and-bound method for solving the traveling salesman problem by applying Welzl's algorithm and using the Minimum Enclosing Circle (MEC) as a heuristic to enhance the pruning rules in the branch-and-bound method.*

Methodology. *Welzl's algorithm is a classical example of an efficient geometric method with pruning and heuristics, which can be easily integrated into more complex algorithms, including the branch-and-bound method. In problems of placement, packing, collision detection, clustering, and others, Welzl's algorithm can be used as part of an evaluation or pruning function: compute the minimum circle that contains a subset of city objects to estimate the space/volume, and if the circle with a new point goes beyond the allowed limits, prune the branch.*

Scientific novelty. *The author proposes using Welzl's algorithm to speed up the exact solution of the traveling salesman problem. The benefit of finding the minimum circle covering a set of city points on the plane lies in obtaining a tighter lower bound for a branch in the branch-and-bound method, increasing the number of branches in the search tree that are pruned earlier, and consequently making the branch-and-bound method faster. MEC prunes branches in which the unvisited cities lie far apart, and even the most optimal extended route would be too costly. This reduces the search space and accelerates the algorithm.*

Conclusions. *The branch-and-bound method with the use of the Minimum Enclosing Circle has a wide range of practical applications in problems where it is necessary to quickly obtain a high-quality route with minimal computational cost, even if a slight deviation from the optimal solution is acceptable. Thus, using MEC in combination with the branch-and-bound method is a universal approach that combines the precision of mathematical optimization with the speed of heuristic methods and can be implemented in any field where routing has significant practical importance.*

Key words: *traveling salesman problem (tsp), Welzl's algorithm, branch and bound method.*

Постановка проблеми. Задача комівояжера (Traveling Salesman Problem, TSP) залежно від формулювання відноситься до різних класів у сімействі *NP*-задач. TSP як задача розпізнавання є *NP*-повною задачею, оскільки будь-яке розв'язання можна перевірити за поліноміальний час (пройти по циклу, підрахувати суму ваг і перевірити ≤ *K*). TSP як оптимізаційна задача є *NP*-складною задачею, адже ця задача складніша, ніж клас *NP*, бо немає поліноміального алгоритму перевірки оптимальності (можна перевірити, що цикл має певну довжину, але довести, що він мінімальний – важче) [2; 10].

Задача комівояжера полягає у пошуку оптимального маршруту для відвідання кожного із *n* міст. Комівояжеру необхідно побувати у кожному місті рівно один раз, і повернутися у вихідне, з якого була розпочата мандрівка. Відомо що переміщення із міста *i* у місто *j* зазначається вартістю *c(i,j)* гривень. А також можливе опрацювання задачі відносно відстані між містами. Якщо згадати теорію графів, то можна так описати задачу: потрібно відшукати гамільтонів цикл у визначеному графі із найменшою вартістю (сума вартості кожного ребра циклу буде його загальною ціною). У відповідність можна поставити задачу вирішення, яка звучатиме так: чи є у графі *G* Гамільтонів цикл, вартістю меншою або такою ж як значення *k*. Мова даної проблеми математично описується таким чином: $G=(V,E)$ – зв'язний неорієнтовний граф з множиною вершин $|V|=n$, кожному ребру $e \in E$ приписана вага (вартість) $w(e)\geq 0$.

$$TSP_{\leq k}=\{\, (G, w, k) \mid \exists\ C \text{ Гамільтонів цикл: } f(C) \leq k \},$$

де $f(C)$ – це функція вартості Гамільтонового циклу C, $k \in Z$, у графі G наявний цикл Гамільтона із вартістю на більшою ніж k [2].

Більшість прийнятних по часу та кількості вхідних даних методів для розв'язку даної задачі є евристичними (дають не оптимальний, а наближений до нього результат) із доволі значною похибкою. Окрім них існують і точніші методи, проте хоч їх виконання і входить у клас *P*, проте все ж займає значний проміжок часу для достатньо великої вибірки.

Одним з найвідоміших точних методів розв'язання задачі комівояжера є метод гілок та меж, основою якого є пошук маршруту завдяки матриці коефіцієнтів. Даний метод має ряд переваг, а саме, він гарантовано знаходить оптимальний маршрут, тобто це точний метод, як і повний перебір, тільки містить умову відкидання частини розв'язків. За рахунок цього він швидший за повний перебір (завдяки нижнім межам і відсіченню «невигідних» гілок, часто розглядається лише фрагмент усіх можливих шляхів. У кращих випадках – у десятки або навіть тисячі разів швидше за повний перебір). Метод гілок і меж підходить для середніх розмірів задачі (у практиці до 12–14 міст може працювати в реальному часі, при хороших нижніх межах – навіть більше) [6].

Але метод гілок та меж не гарантує швидкий результат у найгіршому випадку, адже при поганому виборі меж або даних він працює майже як метод повного перебору, тобто *O(n!)*). Слід зауважити, що метод потребує складної реалізації, необхідна реалізація зниження матриці, обчислення нижньої межі, черга з пріоритетами, структура вузлів тощо. Він складніший в реалізації ніж жадібний алгоритм або метод повного перебору. Зокрема він чутливий до вибору нижньої межі (якщо нижня межа слабка, тобто погано наближує найменші можливі витрати – відсікання неефективне). Також слід згадати, що він не масштабується до великих задач (для 25+ міст час роботи може бути занадто довгим навіть при хороших оптимізаціях). Але незважаючи на вказані недоліки метод гілок та меж – золотий стандарт для точного вирішення TSP, коли задача невелика або середня за розміром, побудовано матрицю відстаней між містами [6].

Аналіз останніх досліджень і публікацій. Науковці намагалися покращити метод гілок і меж, зокрема шляхом розумнішого розгалуження (вибір вузлів з найкращою потенційною користю), використання сильніших і швидших меж через релаксації та евристики, впровадження жадібних і локальних алгоритмів для швидкого знаходження початкових розв'язків [4], скорочення дерева за допомогою відсікання вузлів і домінування, а також застосування паралельних та розподілених обчислень і комбінування з іншими методами оптимізації, такими як метод відсічних площин або врахування симетрії задачі [3–5]. В кожному з даних випадків покращення однієї характеристики призводило до ускладнення або погіршення інших параметрів [9; 10].

Метою роботи є вдосконалення методу гілок та меж для розв'язання задачі комівояжера за рахунок використання алгоритму Вельцля та обчислення мінімального охоплюючого міста кола для оцінки геометричних меж та побудови евристичного початкового маршруту.

Алгоритм Вельцля використовується для знаходження мінімального кола, що охоплює множину точок на площині. Алгоритм Вельцля – рекурсивний алгоритм для знаходження радіусу мінімального описуючого (охоплюючого) кола, яке накриває всі задані точки. Він працює у середньому за лінійний час. Алгоритм працює «знизу–вгору» рекурсивно: випадково переставляємо точки, потім по черзі «додаємо» точки і підтримуємо мінімальне коло для вже розглянутої підмножини. Якщо нова точка лежить уже в поточному колі – нічого не змінюємо. Якщо не лежить – ця точка має бути однією з опорних для нового кола; тоді ми шукаємо мінімальне коло для попереднього підмножини з примусовим включенням цієї точки у множину опорних *R*. Оскільки опорних не більше трьох, рекурсія по *R* завершується швидко [7; 8].

Даний алгоритм можна використати як евристику для обмеження пошуку маршруту комівояжера. Мінімальне коло дає верхню межу діаметра області з точками-містами. Це може допомогти в методах відсікання методу гілок та меж чи оцінці початкових маршрутів.

Новизна. Алгоритм Вельцля швидко визначає мінімальне описане коло (MEC, Minimum Enclosing Circle – центр кола *C*, радіус кола *r*) для множини точок-міст. MEC сам по собі не дає жорсткої нижньої межі для довжини оптимального TSP-циклу, але дає корисну геометричну інформацію (просторовий масштаб, центр масового скупчення та діаметр $d = 2r$), яку можна використати як:

- джерело евристики для побудови якісного початкового розв'язку (щоб знизити глобальну верхню межу (найкраще знайдене на даний момент допустиме рішення для задачі). Це допоможе алгоритму не витрачати час на явно гірші гілки і поступово звужує область пошук, тобто посилити відсікання;
- для просторового розбиття (кластеризації) у методі гілок та меж (розгалуження за географічними кластерами). Це дозволить розбити простір можливих рішень на кластери або області, які можна оцінювати окремо, і визначати межі для кожного кластеру;
- для швидких геометричних перевірок/відсікань (наприклад, якщо поєднати з MST, нижнею межею у гілці);
- для прискорення обчислення оцінок, бо радіус *r* визначається лише опуклою оболонкою (внутрішні точки можна тимчасово відкинути).

Постановка задачі. Границі (або межі) у методі гілок та меж – це ключовий інструмент для ефективного відсікання невигідних підзадач. Якість і швидкість роботи методу напряму залежать від того, як добре побудовані ці межі. Розглянемо вузли в методі гілок та меж та оцінимо їх перспективність. Для цього знайдемо нижню межу (*LB*) вартості розширення маршруту. У вузлі є частковий маршрут *P*, що проходить через множину вже відвіданих вершин *V*, решта вершин – це *U*, ще не відвідані.

$$LB = \mathrm{length}(P) + LB_{connect}(V,U),$$

де *length(P)* – вартість уже пройденого часткового маршруту,

$LB_{connect}$ – мінімальна додаткова довжина щоб "покрити" *U* і повернутися до маршруту.

Нижня межа вузла (MST bound) у задачі комівояжера ґрунтується на тому, що будь-який тур повинен утворювати зв'язну структуру: для оцінки будується мінімальне основне дерево на множині ще невідвіданих вершин, після чого до нього додають два найкоротші ребра, що з'єднують дерево з початком та кінцем часткового маршруту; сума вартості вже пройденого шляху, MST та цих двох ребер дає нижню межу, яка обчислюється швидко і є популярною релаксацією у методі гілок та меж.

Запропоновано перед побудовою MST запустити на виконання алгоритм Вельцля над *U*, отримати центр *C* і радіус *r* (MEC). Додавання MEC у нижню оцінку вузла в методі гілок і меж дає сильнішу (тіснішу) нижню межу.· MEC же одразу показує геометричний мінімальний діаметр підмножини точок *U* (рис.1). Якщо всі невідвідані вершини лежать у великому колі радіуса r, то для будь-якого туру, що їх відвідає, довжина маршруту не може бути меншою за *2r* (щоб хоча б перетнути діаметр). Таким чином, додаючи MEC-оцінку, «підтягуємо» нижню межу, змушуючи її враховувати просторове розташування точок.

Оцінка складності. Алгоритм методу гілок та меж із редукцією матриці відстаней на кожному кроці робить редукцію матриці ($O(n^2)$) та вставляє вузол у чергу ($O(\log k)$, де *k*- кількість вузлів). В гіршому випадку алгоритм пройде всі *n!* перестановок. Отже, загальна складність становить $O(n!\cdot n^2)$, але в реальності значно менша за рахунок відсікання гілок [1].

Включене в алгоритм обчислення MEC працює в середньому лінійно, $O(m)$ для *m* точок, де $m = |U|$ – кількість ще не відвіданих вершин у вузлі. Тобто кожен вузол тепер додатково має $O(|U|)$. В гіршому випадку, коли MEC викликається у всіх вузлах, складність становить $O(n!\cdot(n^2+n))=O(n!\cdot n^2)$,·тобто той самий порядок, бо n^2 домінує над *n.* Теоретично порядок складності не змінюється: і без MEC, і з ним – це $O(n!\cdot n^2)$. Але на практиці MEC значно підсилює нижню межу, що дає більше вузлів на відсікання, а це, в свою чергу, призводить до того, що дерево пошуку суттєво скорочується. Отже, час виконання у середньому значно менший, особливо для «розкиданих» геометрично точок.

Практична реалізація. Для оцінки практичної реалізації запропонованого рішення обрано мову програмування Python та розроблено код для вирішення задачі комівояжера для вибірки міст України на основі їх координат з використанням методу гілок та меж в класичному вигляді та з відсіканням гілок на основі алгоритму Вельцля та обчислення MEC.

ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

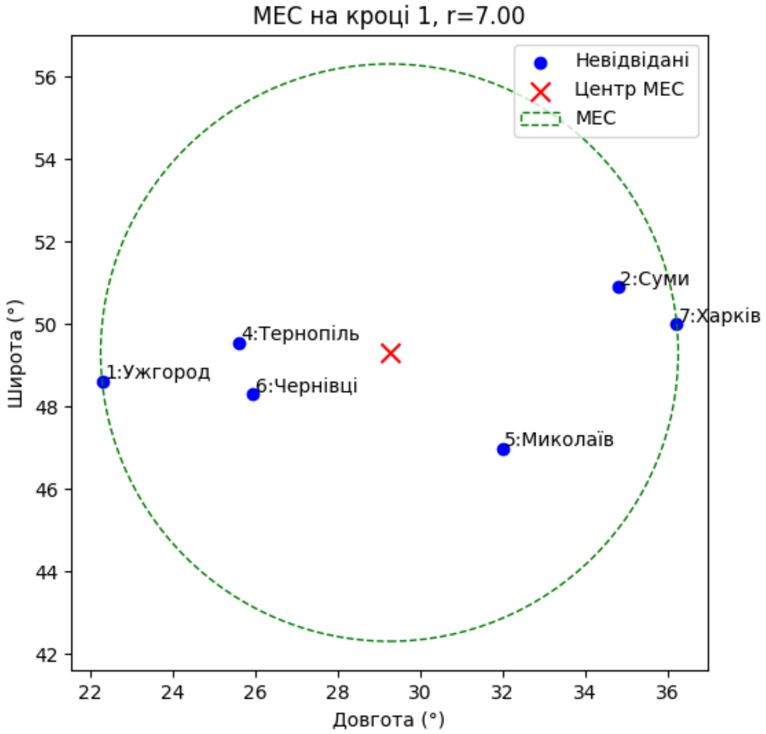

Рис. 1. Візуалізація використання мінімального охоплюючого кола для міст, що входять у нього при певному кроці обчислення на одному з вузлів

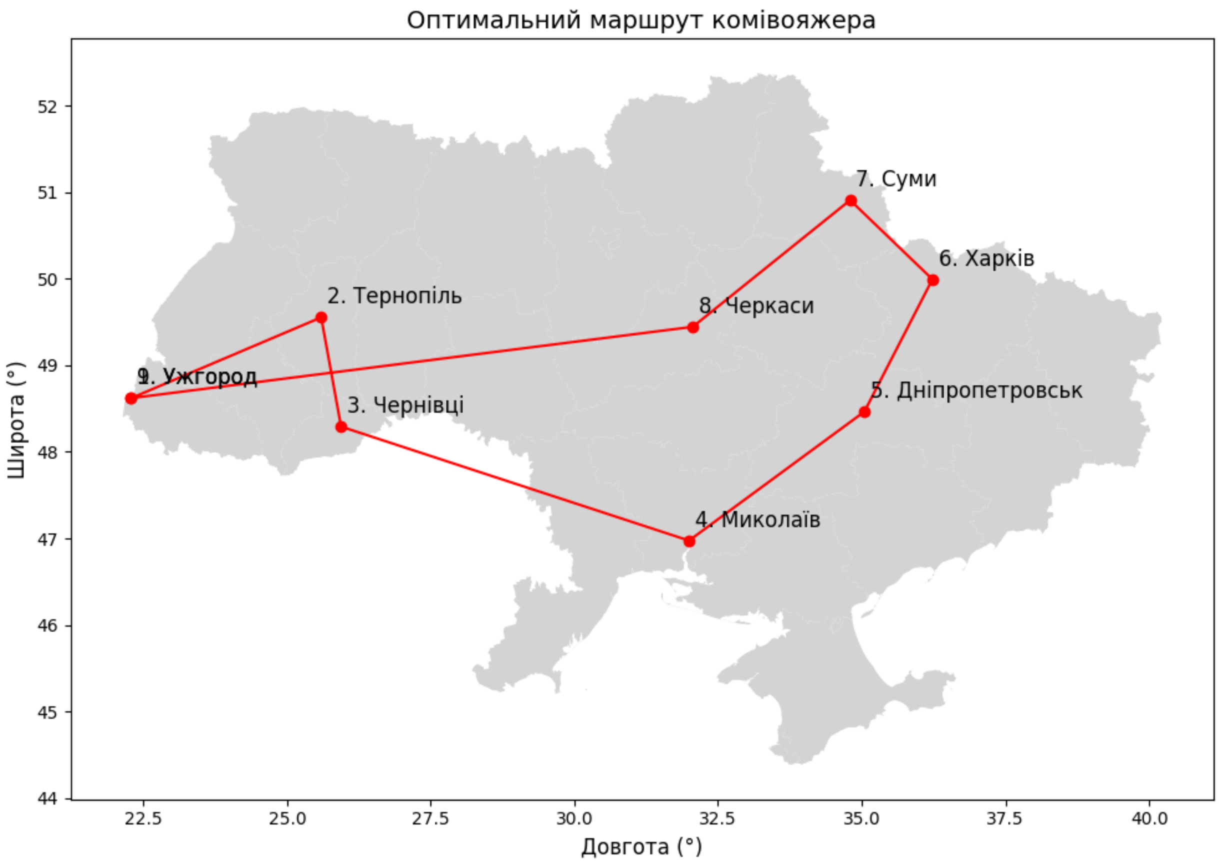

Рис. 2. Графічна візуалізації розв'язку задачі комівояжера для восьми міст України на основі методу гілок та меж із використанням мінімального охоплюючого кола

В (табл. 1) внесено час виконання програм на основі обох кодів: метод гілок і меж, метод з використанням мінімального охоплюючого кола.

Таблиця 1

Час виконання алгоритмів для визначеної кількості міст України

Кількість міст	Час виконання методом гілок та меж (с)	Час виконання із використанням мінімального охоплюючого кола (с)
5	1.40	1.42
6	1.52	1.46
7	1.58	1.56
8	2.31	1.72
9	9.20	3.42
10	74.08	7.62
11	1083.53	26.33

Експериментальні результати показують, що використання мінімального описаного кола (MEC) у методі гілок та меж для задачі комівояжера практично не впливає на продуктивність при невеликій кількості міст (5–7), де накладні витрати на побудову MEC співставні з виграшем від відсікання. Однак починаючи з 8–9 міст MEC забезпечує суттєве скорочення часу роботи алгоритму завдяки ефективнішому відсіканню неперспективних гілок. Для більших задач (10–11 міст) спостерігається експоненційний виграш: час виконання з MEC зменшується у 10–40 разів у порівнянні з класичним підходом. Це підтверджує доцільність інтеграції MEC у схему оцінювання нижньої межі як засобу істотної оптимізації пошуку при зростанні розмірності задачі. На (рис. 3) побудовано графік залежності часу виконання програм від кількості міст.

Аналогічно були проаналізовані дані по кількості ітерацій, що здійснював кожен алгоритм для різної кількості міст. Результати показують, що використання MEC у методі гілок та меж істотно впливає на кількість ітерацій пошуку. Для невеликих задач (5–6 міст) спостерігаються коливання – іноді кількість ітерацій навіть більша, ніж у класичному підході, що пояснюється накладними витратами на перевірку гілок. Однак починаючи з 7 міст MEC різко зменшує кількість ітерацій: наприклад, при 9 містах від 109 601 зменшено до 2 325, а при 11 містах – від майже 10 млн до лише 17 620. Це демонструє, що MEC забезпечує значно ефективніше відсікання неперспективних гілок, завдяки чому простір пошуку скорочується на порядки, і метод гілок та меж стає практично застосовним для більших розмірностей задачі.

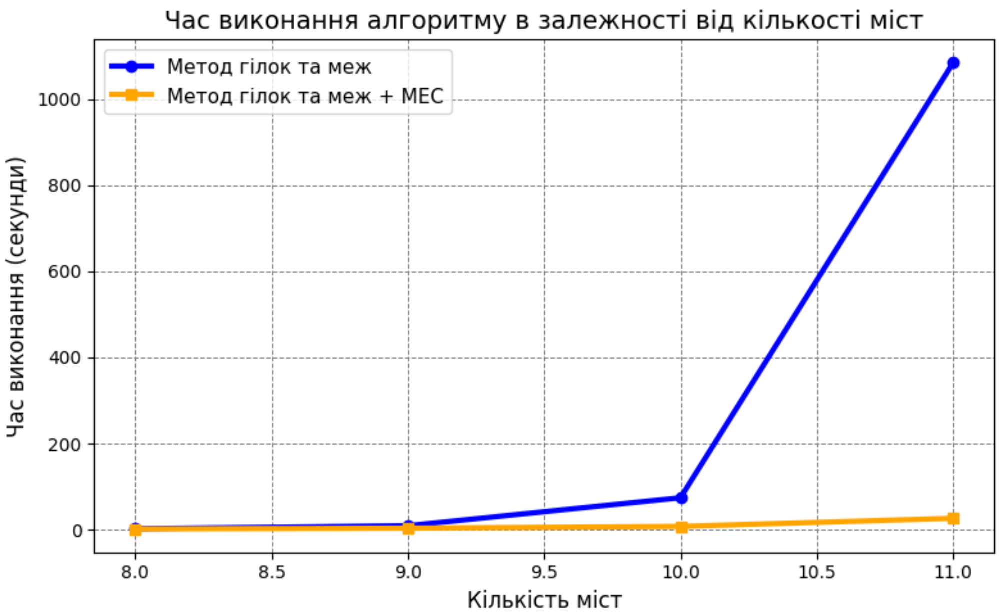

Рис. 3. Порівняльний графік залежності часу виконання від кількості міст методу гілок та меж без використання охоплюючого кола та з ним

ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

Але слід також відзначити, що алгоритм з МЕС стає не абсолютно оптимальний маршрут, а наближений, оскільки додаткова умова відсікання (МЕС) іноді «перерізає» гілки, які могли б привести до оптимального розв'язку. Водночас різниця між маршрутами невелика (у межах 3–5 %).·

Висновки. Алгоритм Вельцля є корисним допоміжним інструментом для оцінки геометричних меж або побудови евристичного початкового маршруту. Використання даного алгоритму при розв'язуванні задачі комівояжера полягає у застосуванні методу мінімального охоплюючого кола МЕС як евристики для підсилення відсікаючих правил у методі гілок та меж. Алгоритм Вельцля дозволяє за лінійний час побудувати найменше коло, що містить множину невідвіданих міст, і цим сформувати тісну нижню межу для вартості продовження часткового маршруту. Якщо мінімальне коло виходить за межі допустимого діапазону або його радіус гарантує надто велику додаткову відстань, така гілка пошуку відкидається ще до обчислення точних оцінок. Це істотно зменшує розмір дерева пошуку, прискорює знаходження розв'язку та робить можливим обробку більших за розміром задач комівояжера. Починаючи з 8 міст спостерігається суттєве прискорення: при 8 містах час зменшується на ≈25 %, при 9 – на ≈63 %, при 10 – на ≈90 %, а при 11 – більш ніж на 97 %. Отже, МЕС неефективний для малих розмірностей, але при збільшенні кількості міст він радикально скорочує час виконання алгоритму, демонструючи виграш на порядки. Також отримані результати свідчать, що використання МЕС у методі гілок та меж призводить до зростання сумарної довжини маршруту порівняно з класичним підходом, тобто отримані розв'язки є не абсолютно оптимальними, а наближеними. Різниця у вартості маршрутів становить у середньому 3–5 %, що можна вважати прийнятним відхиленням з огляду на значне скорочення часу роботи алгоритму та кількості ітерацій. Таким чином, МЕС виступає як ефективна евристика, яка дозволяє істотно зменшити обчислювальні витрати при розв'язанні задачі комівояжера за рахунок незначної втрати оптимальності.

Список використаних джерел:

1. Bang-Jensen J., Gutin G., Yeo A. When the greedy algorithm fails. *Discrete Optimization*. 2004. Vol. 1. P. 121–127.
2. Bendall G., Margot F. Greedy Type Resistance of Combinatorial Problems. *Discrete Optimization*. 2006. Vol. 3. P. 288–298.
3. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 2nd ed. MIT Press and McGraw-Hill, 2001. ISBN 0-262-53196-8.
4. Flemming J. A simple linear-time algorithm for the smallest enclosing circle, 2024. URL: https://arxiv.org/abs/2402.17853
5. Formella A., van Leeuwen E. A quasi-linear time heuristic to solve the Euclidean traveling salesman problem. 2024. arXiv:2401.12345. URL: https://arxiv.org/abs/2401.12345
6. Gutin A., Yeo A., Zverovich A. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*. 2002. Vol. 117. P. 81–86.
7. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation. 2nd ed. Addison-Wesley, 2000. ISBN 81-7808-347-7.
8. Kiran M. S., Beskirli M. A new approach based on collective intelligence to solve traveling salesman problems. *Biomimetics.* 2024. Vol. 9, No. 3. P. 95. DOI: https://doi.org/10.3390/biomimetics9030095.
9. Puerto J., Valverde-Martín C. The hampered travelling salesman problem with neighbourhoods. *Computers & Industrial Engineering*. 2024. Vol. 189. P. 109120. DOI: https://doi.org/10.1016/j.cie.2024.109120.
10. Smolík M., Vondrák I., Král J. Preprocessing techniques for the smallest enclosing circle problem. *Lecture Notes in Computer Science*. 2022. Vol. 13277. P. 312–324. DOI: https://doi.org/10.1007/978-3-031-10599-0_24.

Дата надходження статті: 07.09.2025
Дата прийняття статті: 20.10.2025
Опубліковано: 04.12.2025