УДК 004.04

С.Д. Погорілий, д-р техн. наук, проф.,
А.В. Потебня, студент
Київський національний університет імені Тараса Шевченка
sdp77@i.ua, poteba@yandex.ru

Новітній метод розв'язання задач комбінаторної оптимізації великої розмірності

Вперше запропоновано ефективний метод розв'язання складних задач оптимізації в режимі м'якого реального часу. Формування розв'язків відбувається за допомогою розподілу множини вершин графа на сукупність оболонок та їх сполучення. Виконано низку експериментальних досліджень алгоритму та показано придатність його використання для розв'язання задачі комівояжера. Наведено рекомендації щодо застосування методу при розв'язанні прикладних задач.

Ключові слова: задачі комбінаторної оптимізації, мінімальна опукла оболонка, задача комівояжера, картографічні сервіси, алгоритм Джарвіса, м'який реальний час, розбиття графів, NP-складні задачі, граф.

Вступ

NP-складні задачі оптимізації є надзвичайно розповсюдженими у багатьох сферах сучасних наукових досліджень [1]. Без розробки та застосування методів оптимізації неможливе керування ректифікаційними колонами в промисловості, установками крекінгу нафти, конверторами при виробництві сталі та ін. До відомих транспортних задач зводяться проблеми економічної кібернетики та керування організацією виробництва [2].

При розробці надвеликих інтегральних схем (VLSI), виникає необхідність з'єднати їх компоненти у такий спосіб, щоб одержаний пристрій був компактним, споживав мало потужності та швидко проводив сигнали [3]. Подібні вимоги враховуються при проектуванні топології локальних мереж, визначенні розподілу задач між процесорними елементами, а також при синтезі ланцюжків ДНК та плануванні руху робота [4, 5].

До важливих геометричних задач оптимізації належать задача комівояжера *(travelling salesman problem)*, проблема пошуку мінімального дерева Штейнера *(minimum Steiner tree)*, задача розбиття графів *(graph partition problem)* та багато інших [6].

Класичні алгоритми не застосовні до задач такого типу, що належать до класу NP-складних. Обчислювальні затрати на їх розв'язання експоненційно зростають зі збільшенням розмірності оброблюваних графів [7].

В останні роки завдяки розвитку картографічних сервісів *Google Maps* та *Яндекс Карти* надзвичайного поширення набули засоби пошуку найкоротших маршрутів між заданими точками [8, 9]. Відомо, що такі операції здійснюються на основі алгоритму Дейкстри, запропонованого понад 50 років тому. Однак, зростання популярності цих сервісів призвело до збільшення вимог користувачів щодо такого функціоналу. Зокрема, важливим його недоліком є відсутність засобів побудови кільцевих маршрутів, що потребують розв'язання задачі комівояжера. Крім того, розроблені інструменти повинні працювати в режимі м'якого реального часу та забезпечувати можливість динамічної перебудови маршруту.

Нині запропоновано декілька пакетів відповідного програмного забезпечення, найвідомішим серед яких є *OptiMap* [10]. Однак, спроби його застосування для побудови маршрутів наштовхнулися на ряд труднощів, до яких належать обмеження кількості вхідних точок, відсутність можливості динамічного редагування побудованих шляхів і значні часові витрати.

Згадані недоліки та ряд інших спричинюють непридатність такого засобу для практичного використання та підтверджують потребу розробки новітнього програмного забезпечення для розв'язання поставленої задачі.

Задача комівояжера широко застосовується для перевірки обчислювальних алгоритмів комбінаторної оптимізації. Тому її використано для відстеження ефективності новітнього методу, запропонованого у цій статті. Існує небагато алгоритмів, здатних забезпечити одержання якісних циклів комівояжера для великих графів, що обумовлює потребу впровадження нових засобів розв'язання цієї задачі [11].

<u>Метою дослідження</u> є розробка новітнього високошвидкісного методу розв'язання складних задач оптимізації для графів великої розмірності. Виконано його реалізацію на прикладі задачі комівояжера з метою встановлення залежностей якості одержаних циклів від різних наборів вхідних даних. На основі сформованого методу за-

пропоновано програмний продукт для побудови картографічних маршрутів у режимі м'якого реального часу.

Основоположні поняття

Відомо, що в загальному випадку розв'язання задачі оптимізації передбачає знаходження мінімуму заданої функції [12]. Наприклад, проблема Штейнера потребує відшукання найкоротшої мережі, яка з'єднує скінченний набір точок. Задача розбиття графів вимагає виділення окремих груп вузлів відповідно до значень заданої вагової функції [6, 13].

Задача комівояжера передбачає одержання закритого циклу найменшої ваги, що містить всі вершини, наявні у графі. Важливою є умова їх одноразових відвідин при побудові маршруту.

Нехай, неорієнтований граф G задано за допомогою множини вершин V, множини ребер E та вагової функції $\omega: E \to R.$ Відповідно, $c_{i,j}$ – вага ребра $e_{i,j}$, яка дорівнює довжині ділянки (i, j) вихідного графа. В цій роботі для дослідження обрано симетричну евклідову задачу комівояжера, для якої виконуються умови $c_{i,j} = c_{j,i}$ та $c_{i,j} + c_{j,k} \ge c_{i,k}$, де $i, j \in V$ [14].

Нехай, H – множина всіх допустимих циклів, які відповідають наведеним умовам, а M' – деякий розв'язок, що належить множині H:

$$M' = \langle m_1, m_2, \ldots, m_n \rangle,\ m_i \in V,\ M' \subset H.$$

Відповідно, якщо L' – функція довжини маршруту, то задача комівояжера вимагає знаходження циклу найменшої довжини M^*, для якого виконується наступне співвідношення:

$$L^* = len\ M^* \to \min len\ M',\ \forall M' \subset H.$$

Групою вчених розроблено пакет програмного забезпечення для точного розв'язання задачі комівояжера – Concorde TSP Solver [15]. Він забезпечив одержання оптимальних розв'язків для всіх тестів з бібліотеки TSPLIB [16], зокрема, для графа, що містить 85900 вершин. Однак, пошук маршруту зайняв майже 136 років процесорного часу на кластері з процесорами Intel Xeon 2,8 GHz та AMD Opteron [1].

Розв'язання задач оптимізації великої розмірності

Для розв'язання NP-складних задач оптимізації запропоновано чимало алгоритмів та штучних інтелектуальних систем. До них належать нейронні мережі, евристичні алгоритми, методи бджолиної колонії, алгоритми штучних бактерій, генетичні алгоритми, штучні річкові потоки, алгоритми імунної системи та багато інших [5].

Наприклад, одним з найбільш ефективних засобів наближеного розв'язання задачі комівояжера є алгоритм Ліна-Кернігана-Гельсгауна [1, 17]. Однак, значна обчислювальна складність обумовлює непридатність його застосування при обробці графів великої розмірності.

Тому, в працях [18, 19] запропоновані декомпозиційні алгоритми розв'язання NP-складних задач оптимізації. Вони передбачають відстеження у вихідному графі кластерів з високою концентрацією вершин, формування відповідних частинних розв'язків та зшивання їх до загального результату. Разом з тим, застосування алгоритмів декомпозиції спричинює появу низки важливих недоліків:

1. Виділення кластерів на основі густини розташування вершин графа, запропоноване в роботі [18], може виявитися марним у випадку рівномірного розподілу вузлів. Крім того, потужність кожної з груп вершин є непередбачуваною, що спричинює неможливість контролю за розподілом навантаження.
2. Натомість у праці [19] запропоновано виділення кластерів з фіксованою кількістю вершин. Однак, такий алгоритм потребує знаходження тріангуляції – розбиття площини на плоскі фігури – трикутники. При цьому задача побудови оптимальної тріангуляції також є NP-повною. Відомі алгоритми можуть забезпечити лише наближене розв'язання даної проблеми [20].
3. Сформовані за допомогою цих методів маршрути мають відповідні вади, часткове усунення яких є можливим лише за рахунок застосування допоміжних оптимізаційних алгоритмів [5].

Наведені недоліки, властиві для відомих алгоритмів, додатково демонструють необхідність впровадження новітнього надшвидкого методу для розв'язання задач оптимізації.

Новітній швидкий метод розв'язання складних задач оптимізації великої розмірності

Нехай, задано неорієнтований граф G = (V, E). Ідея нового методу полягає у розбитті множини вершин вихідного графа V на сукупність оболонок $O = \langle O_1, O_2, \ldots, O_n \rangle, O_i \subseteq V$ за відповідними правилами. При цьому всі вузли повинні бути розподілені за сформованими підмножинами, тобто $\bigcup_{i=1}^{n} O_i = V$ та $O_i \ne \varnothing$. Додатково забезпечується умова ортогональності розбиття, тобто одна вершина не може входити до складу різних оболонок: $O_i \bigcap O_j = \varnothing, \forall i \ne j$.

Для виділення відповідних підмножин застосовано поняття мінімальної опуклої оболонки (МОО). Відомо, що опуклою оболонкою *(convex hull)* множини вершин графа $Conv V$ називається найменша опукла множина, яка містить V. Існує чимало способів побудови мінімальних опуклих оболонок. До них належать алгоритм Джарвіса *(Jarvis march)*, Грехема *(Graham Scan)*, швидкої оболонки *(QuickHull)* та багато інших [21]. У цій статті для формування МОО застосовано алгоритм Джарвіса. Наприклад, виділені вершини у графі на рис. 1а належать до його мінімальної опуклої оболонки.

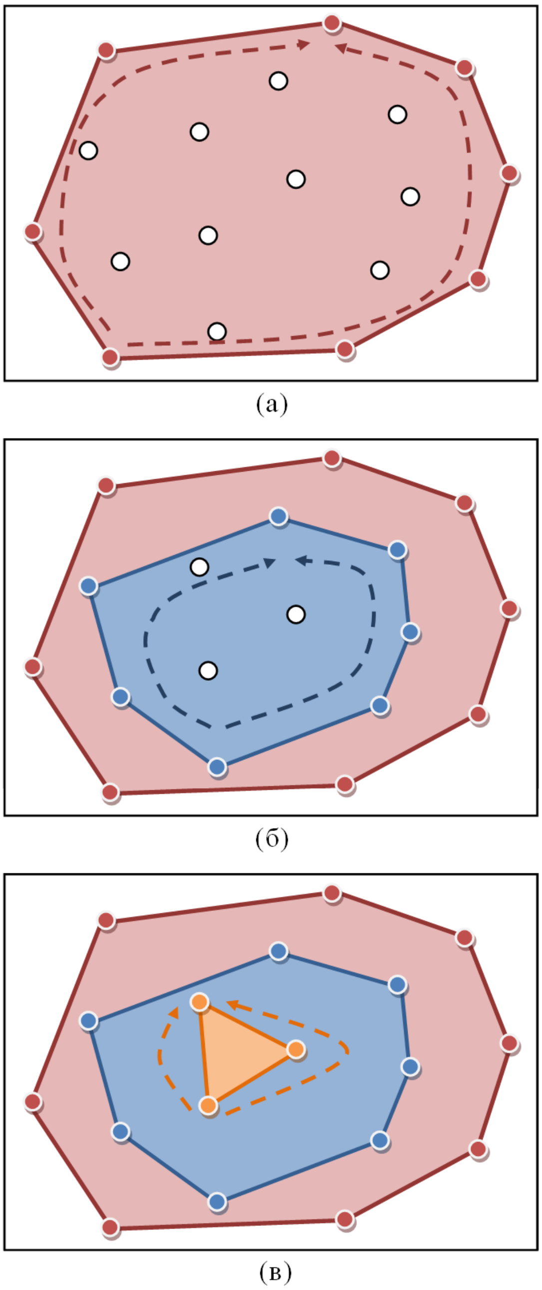


Рисунок 1 – Приклад розбиття множини вершин графа на сукупність оболонок

Алгоритм Джарвіса або «метод загортання подарунка» починає роботу зі знаходження точки $v_1 \in V$, яка гарантовано входить до складу МОО. Наприклад, в якості такої вершини може бути обрана найнижча точка вихідного графа. Якщо задано декілька вузлів з найменшою ординатою, то обирається вершина, розташована ліворуч від інших [22].

Надалі визначається точка $v_2 \in V$, яка має найменший полярний кут відносно v_1 як початку координат. Знайдена вершина включається до складу МОО. Визначення наступних точок відбувається за аналогічним принципом. Якщо чергова знайдена вершина v_i є початковою точкою v_1, то обхід графа завершується і побудова МОО вважається закінченою.

Важливою перевагою алгоритму Джарвіса є можливість його розпаралелювання, оскільки обробка графа може одночасно здійснюватися за двома напрямками, як показано на рис. 1а.

Таким чином, новітня процедура виділення оболонок може бути представлена за допомогою наступного виразу:

$$O_i = Conv\,(V \setminus O),\ O = \langle O_1, O_2, \ldots, O_{i-1} \rangle.$$

Тобто, на кожному етапі формування чергової оболонки відбувається шляхом пошуку МОО множини вузлів, які належать до V, але не входять до складу попередніх оболонок. Алгоритм закінчується, якщо всі вершини розподілені за оболонками, тобто при $V \setminus O = \varnothing$.

Таблиця 1 – Придатність нового методу до розбиття множини вершин графа на сукупність оболонок O

кількість вершин графа, |V|	середня кількість виділених підмножин, |O|
500	30,7
1000	48,1
1500	64,4
2000	77,4
2500	88,8
3000	101,5
3500	112,8
4000	122,4

Розглянемо приклад застосування цього методу. Формування зовнішньої оболонки O_1 здійснюється шляхом знаходження МОО множини вершин вихідного графа (рис. 1а). За аналогією відбувається утворення наступних наборів вузлів (рис. 1б). Знайдена система оболонок міститься на рис. 1в. Таким чином, при |V| = 17 множина вершин графа зазнала розподілу на 3 оболонки.

Таблиця 1 демонструє здатність розробленої процедури до виділення оболонок за різних значень розмірності графа. Отримані результати підтверджують доцільність використання запропонованого методу для розподілу набору вершин за підмножинами.

На рис. 2 сформовано залежність кількості вузлів утворених оболонок від розмірності вихідних графів. Випробування проводилися для початкових наборів вершин, одержаних випадковим чином. З отриманих результатів випливає, що зі збільшенням розмірності задачі середня кількість вершин у сформованих оболонках зростає. Наприклад, якщо при |V| = 500 більшість оболонок містять 15 – 20 вузлів, то при |V| = 2000 цей показник досягає 35 – 40 вершин.

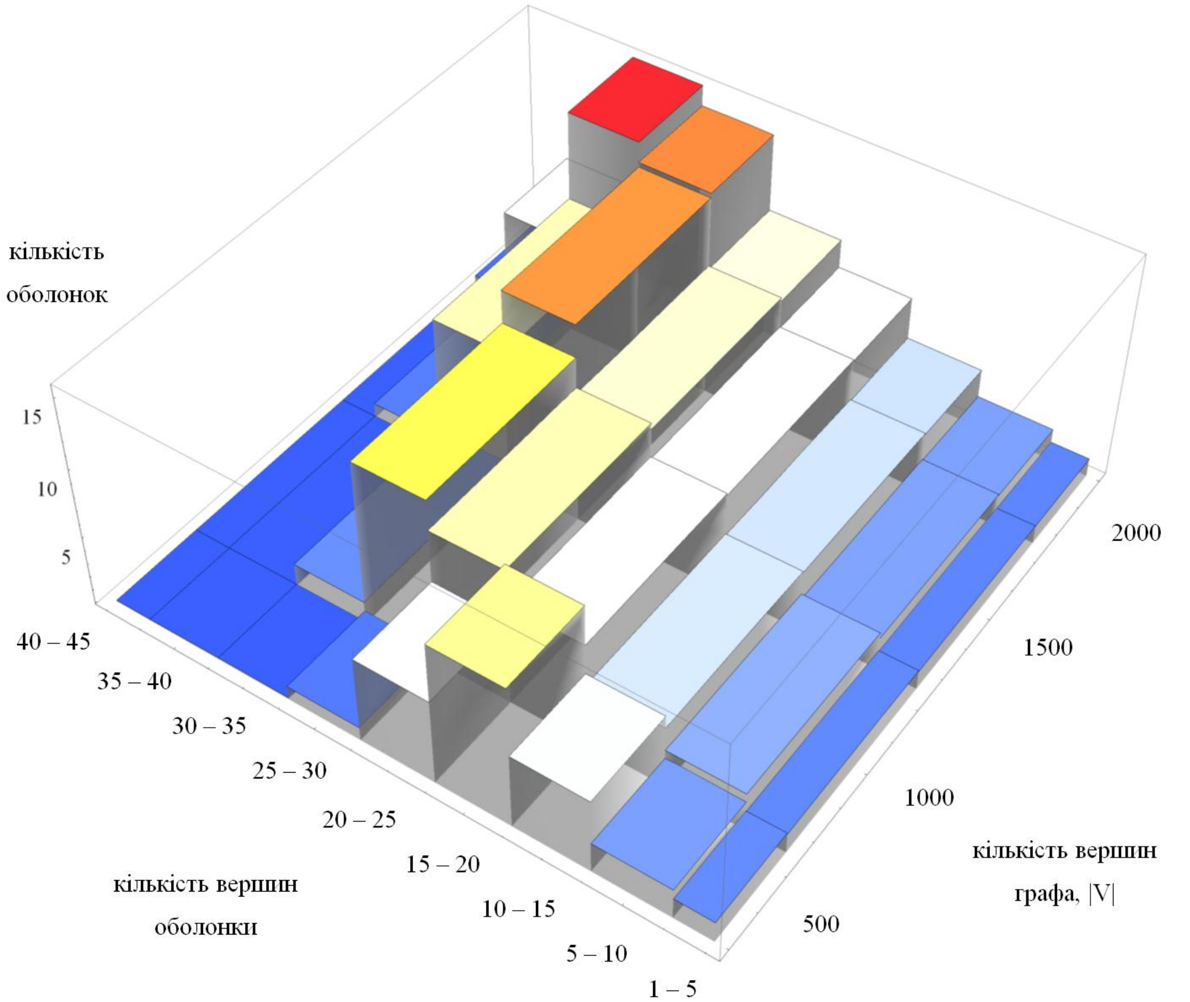


Рисунок 2 – Залежність кількості вершин виділених оболонок від розмірності графа

Наступний етап запропонованого методу полягає у здійсненні зшивання сусідніх оболонок з метою формування розв'язку вихідної задачі оптимізації. Розглянемо виконання відповідних операцій на прикладі знаходження циклу комівояжера.

Відомо, що зовнішнє та внутрішнє кільця O_i та O_{i+1} є найменшими за довжиною для відповідних наборів вершин. Метою процедури зшивання є формування об'єднаної оболонки $O_i \bigcup O_{i+1}$ зі збереженням відповідних властивостей.

Розроблений з використанням цієї умови метод сполучення має наступний вигляд:

1. Відбувається включення до зовнішнього кільця O_i вершин внутрішньої оболонки O_{i+1} для формування початкового наближеного розв'язку. Ця операція здійснюється шляхом заміни ребер зовнішньої оболонки $(k, k+1)$ на відповідні з'єднання з внутрішніми вузлами (k, j) та $(j, k+1)$ при $k, k+1 \in O_i$, $j \in O_{i+1}$. Такі модифікації виконуються для вершин, які задовольняють умові «найдешевшого включення»: $\min(c_{k,j} + c_{j,k+1} - c_{k,k+1})$, де $c_{k,j}$ – довжина ділянки (k, j) вихідного графа. Проблема заміни одного ребра на з'єднання з кількома внутрішніми ву-

злами вирішується шляхом додавання до O_i відповідної ділянки внутрішнього кільця.

2. Наступні операції забезпечують необхідні вдосконалення початкового маршруту. Під час послідовного сканування внутрішньої оболонки O_{i+1} здійснюються спроби додавання її ребер до сформованого розв'язку. Такі операції потребують перебудови системи з'єднань між сусідніми оболонками. Відповідні зміни фіксуються при умові скорочення довжини маршруту.

3. Застосування допоміжних оптимізаційних алгоритмів *(local search optimization algorithm)* забезпечує суттєве покращення отриманих маршрутів [23]. Наприклад, у роботі [16] подібні способи оптимізації використано з метою вдосконалення розв'язків, одержаних за допомогою методів декомпозиції. Слід зазначити також про їх застосування у праці [24] для збільшення якості маршрутів, запропонованих генетичними алгоритмами.

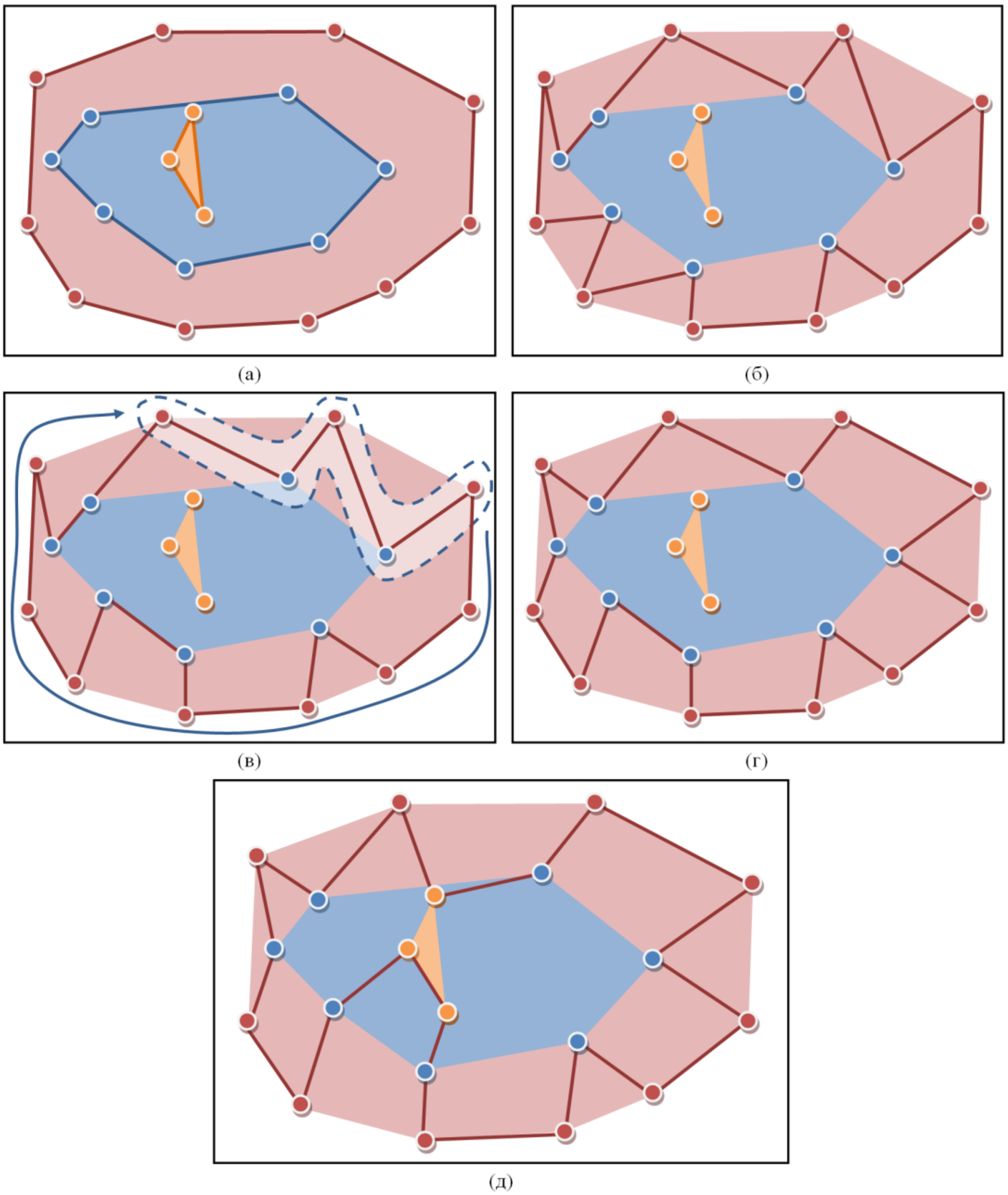


Рисунок 3 – Приклад зшивання оболонок графа для розв'язання задачі комівояжера

Для здійснення локальної оптимізації сформованих розв'язків у даній роботі використано один з найпоширеніших способів – 2-opt. Під час послідовного сканування початкового маршруту цей алгоритм здійснює заміну двох ребер з метою збільшення якості запропонованих розв'язків [25].

Разом з тим, значна обчислювальна складність цієї процедури обумовлює непридатність її застосування при обробці маршрутів для графів великої розмірності. Тому, новий метод передбачає виділення в утвореному циклі ланцюжків, в межах яких відбувається сканування за допомогою алгоритму 2-opt. Послідовний зсув таких ділянок вздовж кільця дозволяє здійснювати перевірку потенційно «небезпечних» ділянок графа, як показано на рис. 3в.

Розглянемо приклад виконання цього алгоритму. Нехай, множина вершин вихідного графа зазнала розбиття на 3 оболонки (рис. 3а). Зшивання зовнішніх кілець потребує формування наближеного розв'язку, наведеного на рис. 3б. Наступні вдосконалення відбуваються шляхом додавання ребра внутрішньої оболонки та відповідної перебудови маршруту (рис. 3в). Здійснення послідовного сканування спричинює додаткові покращення розв'язку, сформовані на рис. 3г. Об'єднання наступних оболонок відбувається за аналогічним принципом. Знайдений цикл, що є найменшим для даного графа, міститься на рис. 3д.

Експериментальні дослідження запропонованого методу

Формування загального циклу шляхом об'єднання складових оболонок дозволяє істотно збільшити швидкість одержання розв'язків. Однак, неправильний вибір параметрів сканування може призвести до виникнення низки важливих порушень. Накопичуючись з приєднанням додаткових оболонок, ці спотворення призводять до катастрофічної деградації розробленої системи. Особливого значення ця проблема набуває при розв'язанні задачі комівояжера для графів великої розмірності.

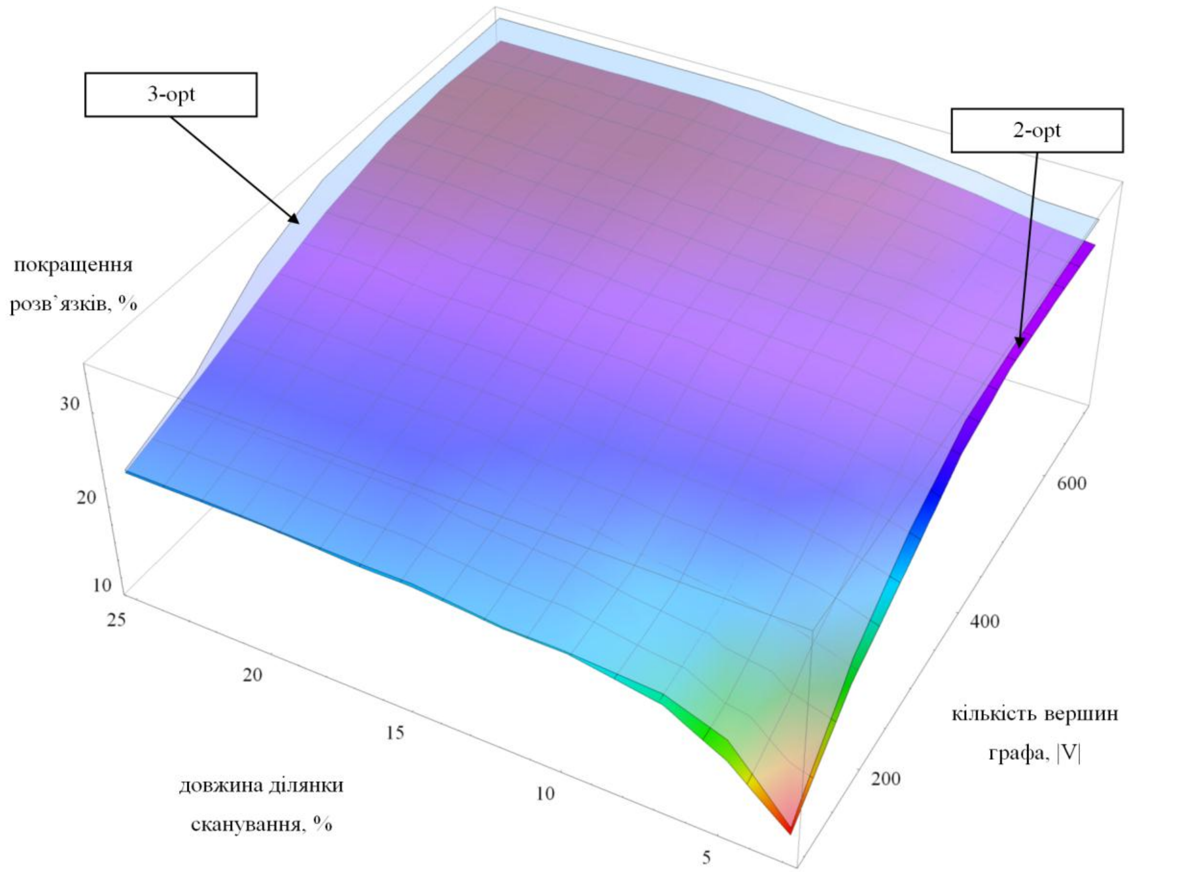


Рисунок 4 – Залежність ефективності послідовного сканування від довжини застосованих ланцюжків та розмірності графа для різних алгоритмів оптимізації

На рис. 4 сформовано залежність ефективності послідовного сканування від довжини виділених ланцюжків та розмірності графа. Отримані результати підтверджують важливість визначення оптимальних параметрів області сканування. Вибір малих значень довжини ланцюжків (2 – 5%) суттєво знижує якість сформованих циклів. В цьому випадку пошкодження зовнішніх оболонок хаотично поширюються при з'єднанні з наступними кільцями. Проте, встановлення великої довжини ланцюжків (вище 15%) майже не впливає на збільшення якості розв'язків, зменшуючи при цьому швидкодію алгоритму. Таким чином, найбільш стабільна робота розробленого методу спостерігається за проміжних значень довжини області оптимізації (7 – 15%).

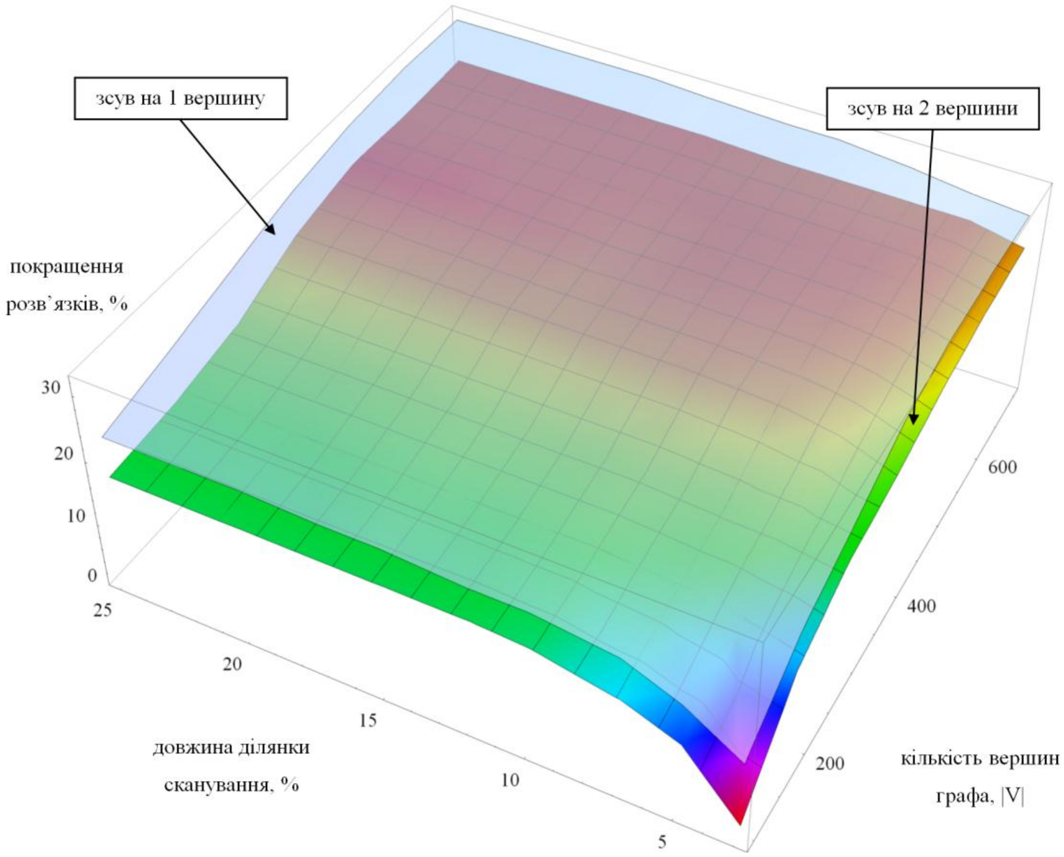


Рисунок 5 – Залежність ефективності вдосконалення циклів від значення зсуву ділянок оптимізації

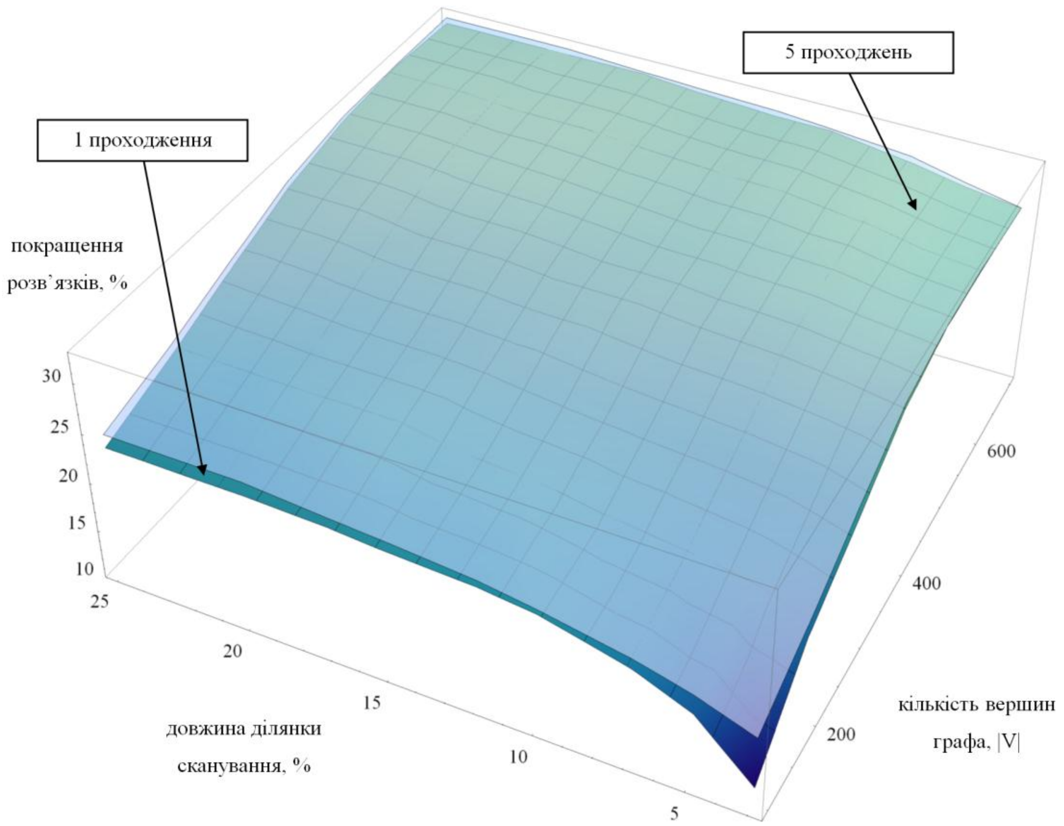


Рисунок 6 – Вплив додаткових проходжень на збільшення якості запропонованих розв'язків

Слід зазначити, що застосування в ділянках сканування алгоритму 3-opt забезпечує незначне підвищення якості сформованих розв’язків. Однак, у зв’язку з надзвичайно високою складністю, використання цієї процедури є обмеженим за великої довжини ділянок сканування.

При обробці циклів для графів великої розмірності зростає важливість вибору оптимального значення зсуву ділянок оптимізації. Відповідна залежність міститься на рис. 5. Отримані результати демонструють, що утворення найбільш якісних циклів відбувається при зміщенні області оптимізації лише на один вузол. Збільшення цього показника до двох вершин неминуче призводить до зменшення ефективності алгоритму на 5%.

Додатковим засобом збільшення якості одержаних маршрутів є виконання повторних проходжень при здійсненні послідовного сканування. У випадку використання невеликих ланцюжків додаткові операції забезпечують скорочення циклів на 5 – 7%. Однак, за протилежних умов ефективність таких сканувань знижується до 1 – 3%.

В разі дотримання наведених рекомендацій застосування розробленого методу забезпечує формування маршрутів, на 0,5 – 2% довших відносно оптимальних розв’язків. Для визначення точності алгоритму були застосовані графи бібліотеки TSPLIB.

На рис. 7 сформовано залежність часу виконання новітнього алгоритму від розмірності графа та довжини ділянок оптимізації. При цьому складність запропонованого методу визначається реалізацією наступних складових етапів:

1. Розподілення множини вершин графа за оболонками.

2. Об’єднання утворених кілець та формування загального розв’язку.

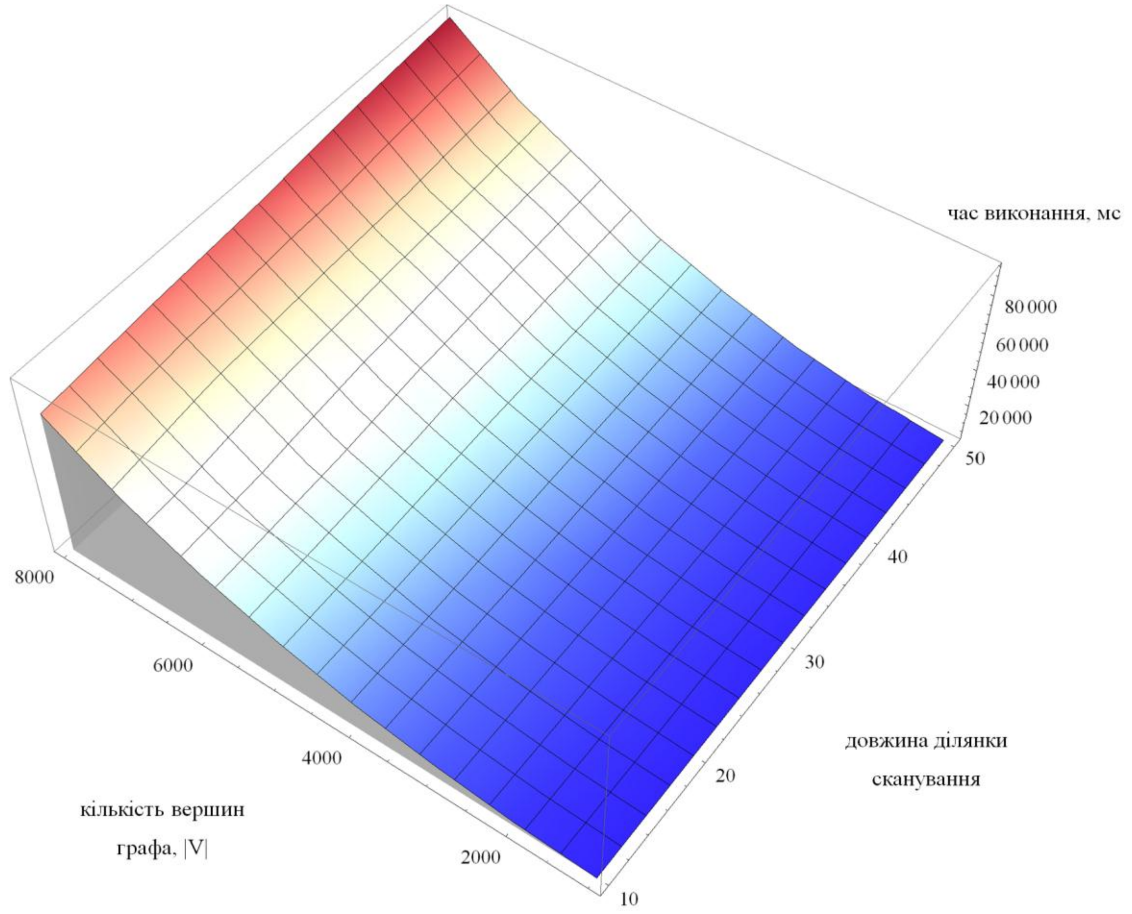


Рисунок 7 – Залежність часу виконання алгоритму від розмірності графа та довжини ланцюжків оптимізації

Відомо, що час виділення кожної оболонки O_i визначається складністю алгоритму Джарвіса $O(nh)$, де n – розмірність графа та h – кількість вузлів, включених до O_i. Однак, за умовами методу всі вершини мають бути розподілені за кільцями, тобто $\bigcup_{i=1}^{n} O_i = V$. Таким чином, етап виділення оболонок має квадратичну складність $O(n^2)$.

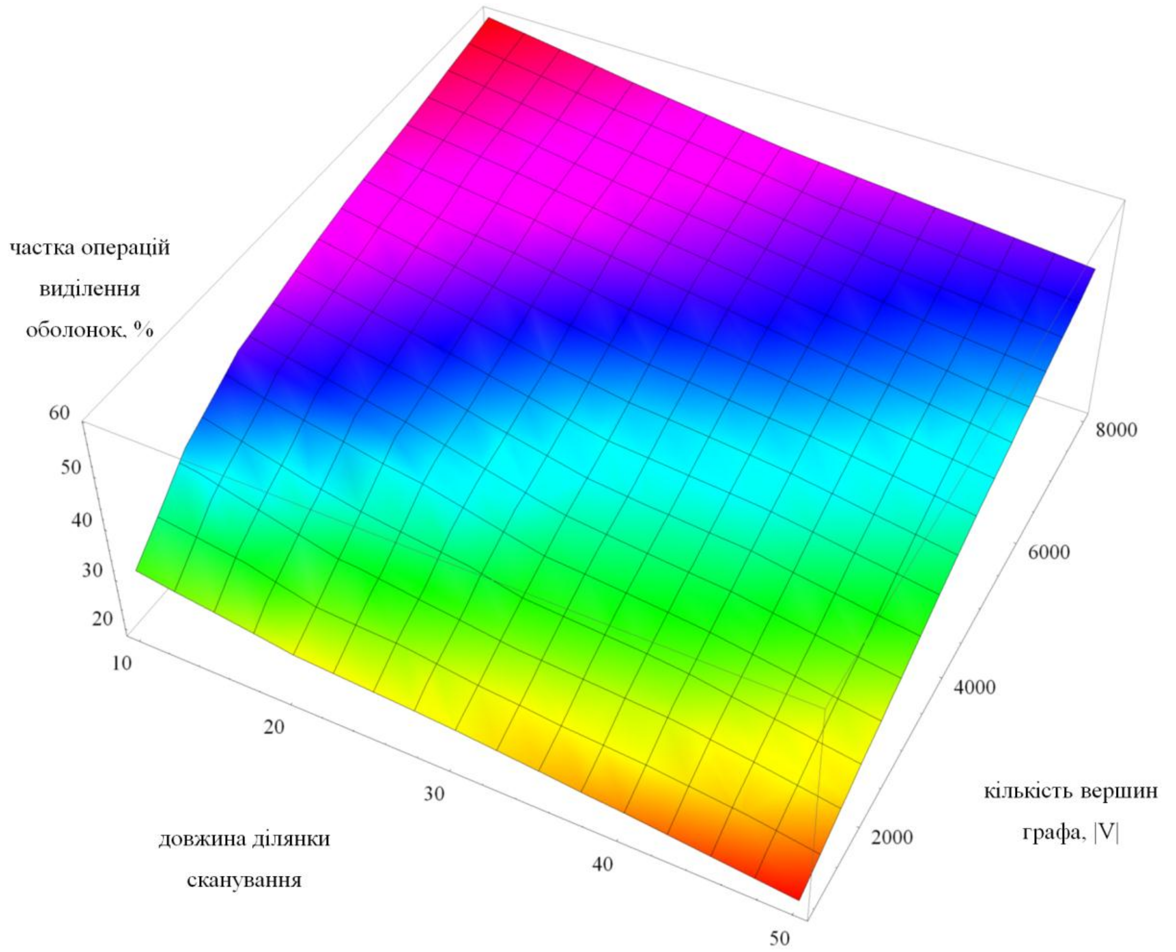


Рисунок 8 – Залежність частки операцій виділення оболонок від розмірності графа та довжини ланцюжків оптимізації

Сполучення утворених кілець потребує здійснення послідовного сканування шляхом виділення та зсуву ділянок оптимізації. Зі збільшенням розмірності оброблюваних графів, кількість таких ланцюжків зростає лінійно. Відповідною є також складність цього етапу роботи алгоритму.

Таким чином, час виконання запропонованого методу визначається часткою операцій виділення оболонок, яка неминуче зростає зі збільшенням розмірності оброблюваних графів (рис. 8).

При цьому для невеликих задач оптимізації швидкодія визначається переважно налаштуваннями послідовного сканування. Однак, при обробці великих графів зростає вплив першого етапу, що пояснює загальну квадратичну складність алгоритму.

На рис. 9 наведена залежність використання оперативної пам'яті від розмірності графа. З її вигляду можна зробити висновок про сублінійну просторову складність алгоритму. Форма цієї кривої спричинена розподіленням вузлів графа за оболонками, що пов'язано з необхідністю збереження утворених наборів вершин.

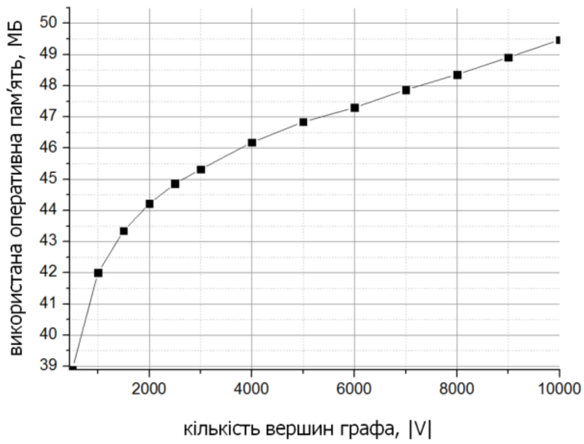
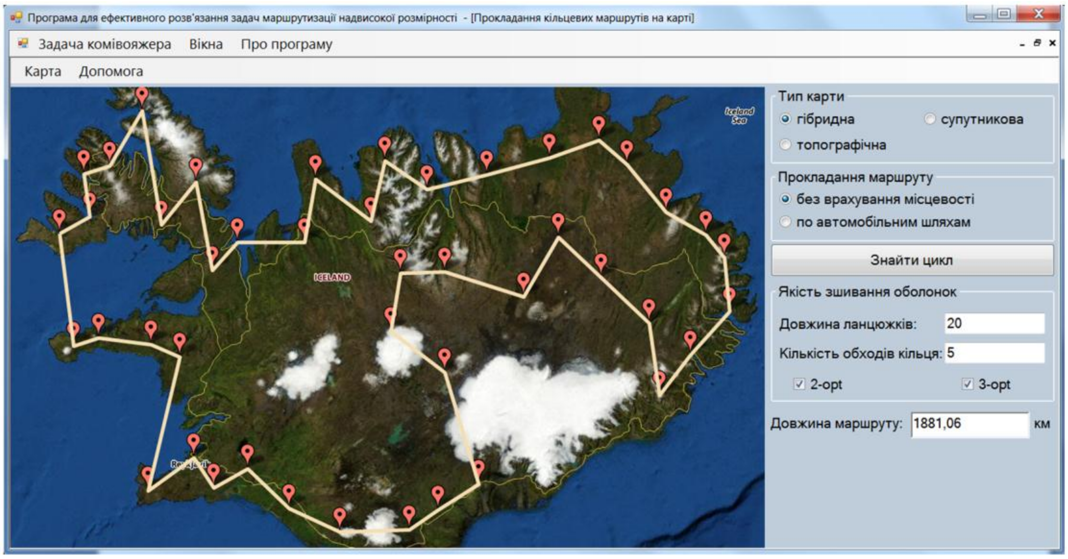


Рисунок 9 – Залежність використання оперативної пам'яті від розмірності графа

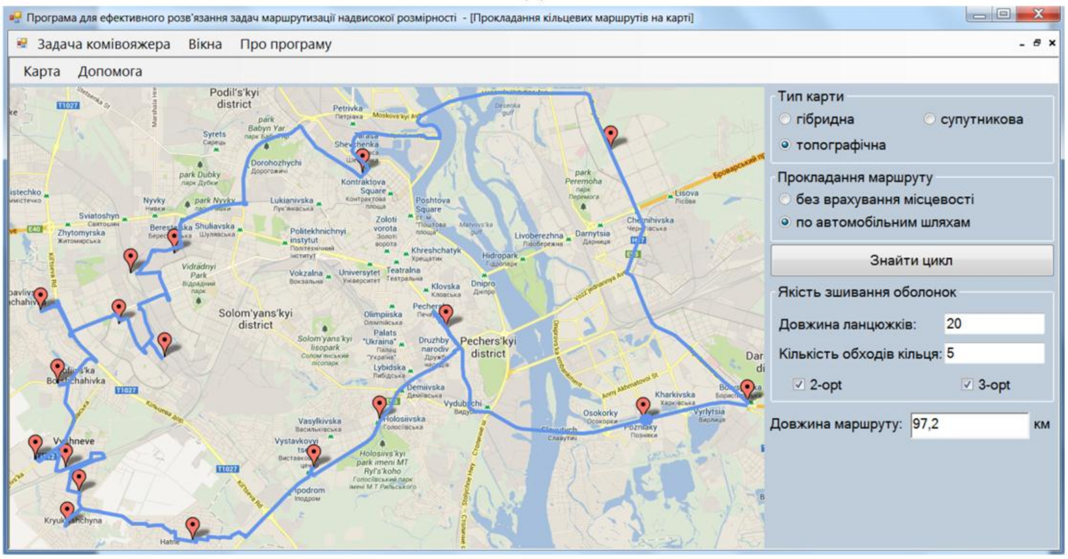
Важливою перевагою запропонованого алгоритму є можливість його розпаралелювання. При цьому необхідними є вдосконалення обох складових етапів формування загального розв'язку. Наприклад, при виконанні зшивання існує можливість розбиття оболонок на окремі сектори та паралельного здійснення в них сформованих операцій. Важливою є також розробка паралельної схеми алгоритму Джарвіса, яка полягає у здійсненні одночасного «загортання» графа за двома та більше напрямками.

На основі запропонованого методу розроблено програмне забезпечення для прокладання кільцевих маршрутів на географічних картах. Сформований продукт відрізняється високою швидкодією та можливістю динамічного редагування циклів. В ньому забезпечено варіанти прокладання маршрутів без врахування місцевості (рис. 10а) та по автомобільним шляхам (рис. 10б). Додатково надана можливість вибору параметрів сканування, що забезпечує контроль за якістю сформованих циклів та швидкістю їх утворення.

Прокладання маршрутів для 100 точок відбувається в режимі реального часу, для 1000 вузлів – потребує 7 с виконання, для 10000 – 130 с. Випробування проводилися за допомогою апаратної архітектури AMD Athlon II P320 – 2.1 GHz.



(а)



(б)

Рисунок 10 – Прокладання маршрутів на карті за допомогою розробленого програмного продукту

Висновки

У статті запропоновано новий підхід до розв'язання складних задач комбінаторної оптимізації. Розподілення вершин графа за оболонками дозволяє суттєво збільшити швидкість формування маршрутів. Експериментальні дослідження алгоритму підтверджують доцільність його застосування при обробці циклів для графів великої розмірності.

Квадратична складність розробленого методу визначається часткою операцій виділення оболонок. Варто зауважити, що таку ж складність мають найбільш прості жадібні алгоритми, які здійснюють формування маршруту на основі вибору найближчого сусіда. Однак, якщо відплатою за «жадібність» є отримання спотворених розв'язків, то застосування нового методу дозволяє одержати якісні цикли за малі проміжки часу.

Розроблений програмний продукт відрізняється швидкодією та може бути включений до функціоналу популярних картографічних сервісів *Google Maps* та *Яндекс Карти*.

Подальший напрямок досліджень пов'язаний з розробкою та впровадженням паралельних схем запропонованого методу. Додаткового дослідження потребують алгоритми зшивання оболонок для формування розв'язків інших задач комбінаторної оптимізації.

Список літератури

1. The Traveling Salesman Problem: A Computational Study / [David L. Applegate, Robert E. Bixby, Vasek Chvátal, William J. Cook]. – Princeton: Princeton University Press, 2007. – 606 p.
2. Ding-Zhu Du, Panos Pardalos, Handbook of Combinatorial Optimization: Supplement Volume A. – Springer, 2010. – 648 p.
3. Погорілий С.Д. Формування та дослідження паралельної схеми алгоритму Крускала для систем зі спільною пам'яттю / С.Д. Погорілий, А.В. Потебня // Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка». – 2012 .– №16 (204). – С. 82 – 89.
4. TSP: Applications of the TSP [Електронний ресурс]. – Режим доступу: URL: http://www.tsp.gatech.edu/apps/index.html.
5. Погорілий С.Д. Розробка новітніх систем штучного інтелекту для розв'язання задачі комівояжера / С.Д. Погорілий, А.В. Потебня // Вісник Київського національного університету імені Тараса Шевченка. Серія фізико-математичні науки. – 2012 – №4. – С. 173 – 184.
6. Charles-Edmond Bichot, Patrick Siarry, Graph Partitioning. – Wiley-ISTE, 2011. – 320 p.
7. Richard M. Karp, Reducibility Among Combinatorial Problems // 50 Years of Integer Programming 1958 – 2008. – Springer-Verlag Berlin Heidelberg, 2010. – P. 219 – 241.
8. Google: API карт Google [Електронний ресурс]. — Режим доступу: https://developers.google.com/maps/?hl=ru.
9. Яндекс: API Яндекс карт. Маршрутизатор [Електронний ресурс]. — Режим доступу: http://api.yandex.ru/maps/doc/jsapi/1.x/dg/concepts/router.xml.
10. OptiMap: OptiMap – Fastest Roundtrip Solver [Electronic Resource]. – Mode of access: URL: http://www.gebweb.net/optimap/.
11. Solving Large-Scale TSP Using Adaptive Clustering Method / [Jin-Qiu Yang, Jian-Gang Yang, Gen-Lang Chen] // Second International Symposium on Computational Intelligence and Design. – 2009 – vol. 1. – P. 49 – 51.
12. Методы оптимизации. Примеры и задачи: Учеб. пособие / [Р.М. Ларин, А.В. Плясунов, А.В. Пяткин]. – Новосибирск, 2003. – 115 с.
13. M. W. Bern, R. L. Graham. The Shortest-Network Problem // Scientific American, 1989 – №3. – P. 64 – 70.
14. TSP: Concorde TSP Solver [Електронний ресурс]. – Режим доступу: URL: http://www.tsp.gatech.edu/concorde/.
15. Gerhard Reinelt. TSPLIB – A Travelling Salesman Problem Library // ORSA Journal on Computing. – 1991 – Vol. 3, № 4. – P. 376 – 384.
16. Decomposition and scanning optimization algorithms for TSP / [Bazylevych R., Prasad B., Kutelmakh R., Bazylevych L.] // Proceedings of the 2008 International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-08), Orlando, Florida, USA, 2008, P. 110 – 116.
17. K. Helsgaun, An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic // European Journal of Operational Research. – 2000 – 126 (1). – P. 106 – 130.
18. Ant colony algorithm for large scale TSP / [Xiaojiang Li, Jiapin Liao, Min Cai] // International Conference on Electrical and Control Engineering – ICECE, 2011. Yichang, 16 – 18 September. 2011. – P. 573 – 576.
19. Adam N. Letchford, Nicholas A. Pearson, Good triangulations yield good tours // Computers & Operations Research. – February, 2008 – Volume 35, Issue 2. – P. 638 – 647.
20. Скворцов А.В. Триангуляция Делоне и ее применение / А.В. Скворцов. – Томск: Изд-во Том. ун-та, 2002. – 128 с.

21. Погорілий С.Д. Новітній швидкий алгоритм знаходження мінімальних опуклих оболонок / С.Д. Погорілий, А.В. Потебня // Праці міжнародної конференції «Високопродуктивні обчислення HPC-UA 2013» (Україна, Київ, 7 – 11 жовтня, 2013). – С. 322 – 329.
22. Preparata, D. F., Shamos, M. Computational Geometry. An Introduction. – Berlin: Springer, 1993. – 390 p.
23. Погорілий С.Д. Дослідження та оптимізація архітектури еластичної нейронної мережі для розв'язання задачі комівояжера / С.Д. Погорілий, А.В. Потебня // Праці 9-ої міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем (TAAPSD'2012)» (Україна, Київ, 4 – 7 грудня, 2012). – С. 236 – 244.
24. Sengoku H., Yoshihara. A Fast TSP solver using GA on GAVA // Proc. Of 3rd Int'l Symp. On Artificial Life and Robotics. – 1998 – Vol. 1. – P. 283 – 288.
25. Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness / [Olaf Mersmann, Bernd Bischl, Jakob Bossek, Heike Trautmann, Markus Wagner, Frank Neumann] // Learning and Intelligent Optimization Lecture Notes in Computer Science. – Springer Berlin Heidelberg, 2012. – P. 115 – 129.

Надійшла до редакції 12.03.2014

С.Д. ПОГОРЕЛЫЙ, А.В. ПОТЕБНЯ
Киевский национальный университет имени Тараса Шевченко

НОВЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ БОЛЬШОЙ РАЗМЕРНОСТИ

Впервые предложен эффективный метод решения сложных задач оптимизации в режиме мягкого реального времени. Метод содержит этапы разбиения множества вершин графа на группы оболочек и процедуру их объединения. Выполнен ряд экспериментальных исследований алгоритма и показана возможность его применения для решения задачи коммивояжера. Приведены рекомендации по использованию метода при решении прикладных задач.

Ключевые слова: задачи комбинаторной оптимизации, минимальная выпуклая оболочка, задача коммивояжера, картографические сервисы, алгоритм Джарвиса, мягкое реальное время, разбиение графов, NP-сложные задачи, граф.

S.D. POGORILYY, A.V. POTEBNIA
Kyiv National Taras Shevchenko University, Ukraine

A NOVEL METHOD FOR SOLVING LARGE-SCALE COMBINATORIAL OPTIMIZATION PROBLEMS

In this paper an effective method is first suggested for soft real-time solving large instances of optimization problems. The new method recursively splits the input graph vertices set into hulls and merges them into final result. A set of algorithm investigations is conducted and its suitability for solving the travelling salesman problem is demonstrated. The developed method can produce good quality solutions quickly due to $O(n^2)$ time complexity and linear memory usage. Another important advantage of the new algorithm is its ability to parallel execution. The special program tool for real-time geographic routes calculation is designed based on the developed algorithm. Recommendations of method usage for applied tasks solving are provided.

Keywords: combinatorial optimization problems, minimal convex hull, travelling salesman problem, map services, Jarvis march, soft real-time, graph partition, NP-hard problems, graph.