ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

УДК 519.16

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ДЕКОМПОЗИЦІЙНОГО АЛГОРИТМУ СПІЛЬНИХ РЕБЕР ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОМІВОЯЖЕРА ВЕЛИКИХ РОЗМІРНОСТЕЙ

Кутельмах Р.К., Угриновський Б.В.
Національний університет «Львівська політехніка»

У статті описано та досліджено декомпозиційний алгоритм на основі спільних ребер для розв'язування задачі комівояжера великих розмірностей. Метод спільних ребер застосований в якості початкової процедури декомпозиційного алгоритму для знаходження розв'язків підзадач. Експериментальні дослідження показали, що алгоритм доцільно застосовувати для розв'язування задач великої розмірності – понад 100000 точок. Алгоритм показує стабільне значення якості отриманого розв'язку в межах 8%. Час розв'язування зростає лінійно-логарифмічно.
Ключові слова: задача комівояжера, алгоритм найближчого сусіда, 2-opt, алгоритм Ліна-Кернігана, метод спільних ребер, декомпозиційний алгоритм.

Постановка проблеми. Задача комівояжера полягає у знаходженні найвигіднішого маршруту, що проходить через вказані вершини тільки по одному разу з поверненням у початкову вершину. Задача комівояжера є *NP*-складною. Вона є однією з базових задач комбінаторної оптимізації, що має широке прикладне застосування [1] в транспортних задачах, задачах планування та логістики, мінімізації рухів у робототехніці, аналізі структури ДНК, системах телекомунікацій, виробництві друкованих плат, лазерній нарізці та ін.

Задачу можна представити у вигляді моделі на графі. Розрізняють різні варіанти задачі, найважливішими з яких є симетрична та асиметрична задачі. У випадку симетричної задачі всі пари ребер між тими самими вершинами мають однакову вагу, тобто для ребер dij та dji ваги однакові. В асиметричному випадку кількість можливих маршрутів вдвічі менша. Симетрична задача моделюється неорієнтовним графом.

В даній роботі розглядатиметься симетрична Евклідова задача комівояжера, де заданими вважають множину V з N точок ($|V|=N$), які описані їхніми координатами (x_i, y_i). Необхідно знайти маршрут T, що проходить по одному разу через кожну точку, довжина якого $L(T)$ є мінімальною.

Багато існуючих задач та проблем мають надвелику розмірність (понад 100000 точок) і потребу в зменшенні витрат на отримання оптимальних розв'язків. Оскільки складність задачі є експонентною, а вимоги до швидкості обчислень, якості та розмірності задач підвищуються, то виникає необхідність розроблення спеціалізованих алгоритмів для швидкого та якісного розв'язування задач великих розмірностей.

Аналіз останніх досліджень і публікацій. Сьогодні існує багато методів розв'язування задачі комівояжера. Відомі методи розв'язання можна поділити на дві групи: точні методи [2] та евристичні (наближені) методи [3].

Точні методи знаходять гарантовано оптимальний шлях. Вони потребують значних часових витрат і підходять лише для задач відносно малих розмірностей – до 1000 вершин, а з використанням розпаралелення – до 5000 вершин в графі.

Евристичні методи знаходять, часто за коротший час, наближені розв'язки. Наближені методи можна поділити на три групи: конструктивні алгоритми, алгоритми оптимізації та метаевристичні методи.

Конструктивні алгоритми – це найшвидші за часом обчислень методи, що забезпечують найгіршу якість розв'язків. За кожен крок роботи алгоритму до існуючої частини знайденого маршруту додається нове ребро. Найкращі конструктивні алгоритми забезпечують якість, на 10-15% гіршу від оптимального розв'язку. До конструктивних алгоритмів належать: алгоритм найближчого сусіда; жадібний алгоритм; алгоритм вставки.

Алгоритми оптимізації намагаються шляхом деяких змін скоротити вже обчислений маршрут. Прикладом такої евристики є *k-opt*-алгоритми, що систематично видаляють з маршруту групи з k ребер і замінюють їх іншими k ребрами отримуючи новий маршрут. Оскільки повний перебір можливих варіантів дорівнює кількості можливих маршрутів, на практиці обмежують k щонайбільше до 5. При цьому, випробовуються всі варіанти заміни двох та трьох ребер, уникаючи перебору з більшою кількістю ребер через значні витрати часу.

Метаевристичні методи комбінують методи пошуку локальних та глобальних розв'язків у абстрактні стратегії евристичної оптимізації задач. Багато методів цього типу базуються на методах пошуку локальних розв'язків, тобто, вони обчислюють деякий початковий розв'язок (наприклад, застосовуючи метод найближчого сусіда) та покращують його іншими методами, наприклад, методом k-opt-евристики, доти, поки неможливо буде знайти кращий маршрут.

Групою вчених [4-5] розроблено пакет програмного забезпечення для точного розв'язування задачі комівояжера – Concorde TSP Solver [6]. В цьому пакеті реалізовані точні методи, зокрема метод гілок і меж, що можуть бути використані для задач невеликих розмірностей. Він забезпечив одержання оптимальних розв'язків для усіх тестів із бібліотеки TSPLIB, розмірністю включаючи 85900 точок.

Відомі евристичні методи мають обчислювальну складність $O(N^2)$ і вище, що є також неефективним для задач великих та надвеликих розмірностей (близько мільйона точок і більше). Найкращий на даний час евристичний метод Ліна-Кернігана-Гельсгауна має обчислювальну складність, близьку до $O(N^{2,2})$.

Дієвими підходами до розв'язування задачі комівояжера великих та надвеликих розмірностей можна назвати декомпозиційні методи, методи локальної та послідовної оптимізації.

Декомпозиційні методи передбачають розбиття всієї множини точок на підмножини обмеженої розмірності (500-2000 точок), для яких можна отримати високоякісні часткові розв'язки з невеликими часовими затратами; зшивання часткових розв'язків у початковий розв'язок та наступне його покращення розробленими оптимізаційними методами. Обчислювальна складність методів є близькою до лінійно-логарифмічної, що обґрунтовано теоретично та підтверджено експериментально. Прикладом декомпозиційного алгоритму є *алгоритм приєднання часткових розв'язків у підмножинах при декомпозиції задачі комівояжера* [7].

Методи локальної оптимізації або як їх ще називають методи локального пошуку, перебирають можливі розв'язки шляхом виконання локальних змін, доки результат не зведеться до оптимального або не буде вичерпано певний ліміт часу чи кількість спроб.

Існує метод знаходження розв'язку задачі на основі спільних ребер [8], що полягає в об'єднанні розв'язків, отриманих за допомогою різних алгоритмів і з меншою обчислювальною складністю. Кроки алгоритму:

1. Знаходження N початкових розв'язків з допомогою N різних евристичних алгоритмів;
2. Знаходження всіх спільних ребер серед N розв'язків;
3. Об'єднання спільних ребер з допомогою іншого евристичного алгоритму.

Виділення невирішених раніше частин загальної проблеми. Експериментальні дослідження методу спільних ребер з використанням алгоритму 2-opt довели, що він є застосовний для задач великої розмірності і дозволяє покращити розв'язок задачі. Метод спільних ребер може бути застосований в якості початкової процедури для знаходження розв'язків підзадач декомпозиції. Поведінка методу спільних ребер із попередньою декомпозицією невідома і потребує експериментальних досліджень та аналізу.

Мета статті. Метою даної роботи є підвищення ефективності розв'язування задачі комівояжера. Для досягнення поставленої мети пропонується реалізувати та дослідити метод знаходження розв'язку задачі комівояжера з допомогою декомпозиції задачі на підзадачі та поступовим розв'язуванням підзадач методом спільних ребер.

Виклад основного матеріалу. Декомпозиційний метод включає три головні етапи: декомпозицію задачі, розв'язування підзадач та об'єднання розв'язків підзадач. Алгоритм із застосуванням методу спільних ребер:

1. Декомпозиція вхідної задачі на N підзадач.
2. Для кожної N_i ($i=\overline{1,N}$) підзадачі виконати кроки 3 і 4.
3. Розв'язування N_i підзадачі з допомогою алгоритму спільних ребер.
4. Об'єднання N_i розв'язку із загальним розв'язком.

Приклад такого розв'язування зображено на рисунку 1.

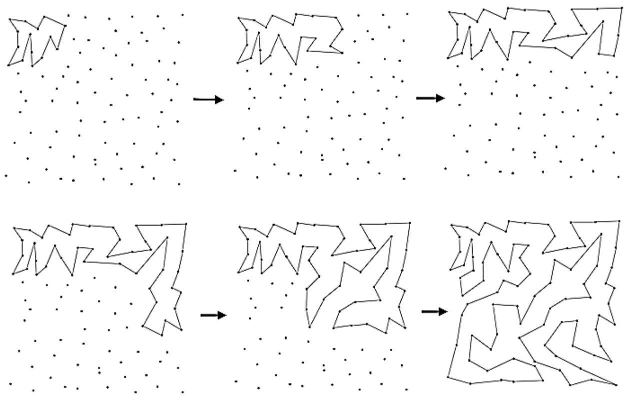

Рис. 1. Знаходження розв'язку задачі комівояжера на основі його нарощення

Кожен етап розв'язування задачі реалізується конкретними алгоритмами чи групою алгоритмів.

Оскільки випробовування алгоритму буде відбуватися на тестових випадково згенерованих задачах комівояжера, які характеризуються рівномірним розміщенням точок, то реалізація декомпозиції полягатиме у розбитті вхідної задачі сіткою, що розділятиме всю множину точок на підзадачі. Параметри декомпозиції включатимуть мінімальну та максимальну кількість точок підзадачі, кількість горизонтальних та вертикальних ліній поділу і т. д.

Виконується поступове нарощення розв'язку з допомогою методу спільних ребер. Процес поступового нарощення починається із пошуку розв'язку у деякій підмножині N_0. Далі вибирається наступна підмножина, наприклад, N_1 (рис. 2).

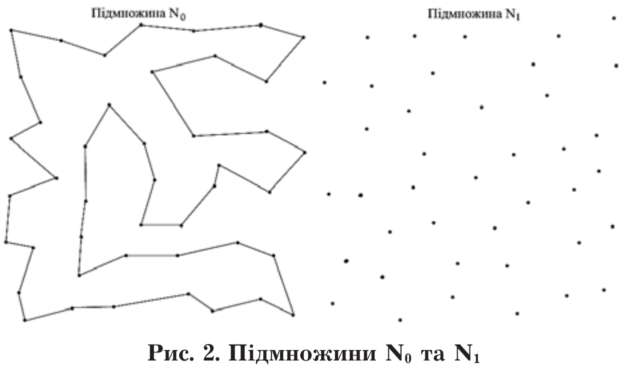

Рис. 2. Підмножини N_0 та N_1

Алгоритм приєднання часткових розв'язків має такий параметр, як величина області «перетину» α. Він може задаватися або у числовому вигляді, або в процентному відносно розміру підмножини. Рекомендоване значення параметра – у межах 20-70% від розміру підмножини.

На рисунку 3 зображено підмножини N_0 та N_1 з областю перетину.

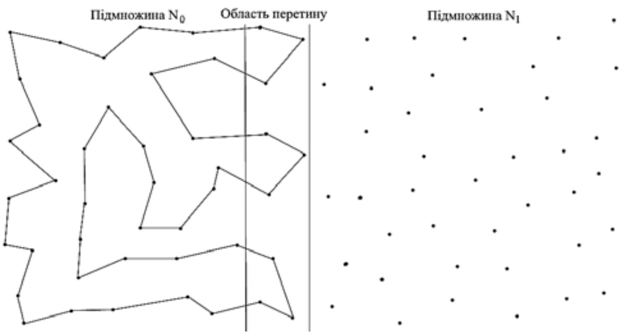

Рис. 3. Підмножини N$_0$ та N$_1$ із областю перетину

Для знаходження єдиного розв'язку задачі комівояжера для двох підмножин N_0 та N_1 розглянемо множину N^*, що містить усі точки з множини N_1 та точки з множини N_0, які належать ребрам з області перетину α. Розв'язок задачі комівояжера для N^* міститиме розв'язок для всієї множини N_1 та тієї частини множини N_0, яка є найближчою до підмножини N_1. Для того, щоб на наступному кроці можна було об'єднати розв'язки задачі комівояжера, у підмножинах N_0 та N^* застосовуються фіксовані ребра, які встановлюються між першою відвіданою точкою та останньою, а також між парами вхідних та вихідних точок у послідовності їх відвідування. Маючи усі необхідні вхідні дані (точки та умовні ребра), можемо розв'язати задачу комівояжера для підмножини N^* з допомогою алгоритму спільних ребер. Рисунок 4 ілюструє розв'язок задачі комівояжера для неї.

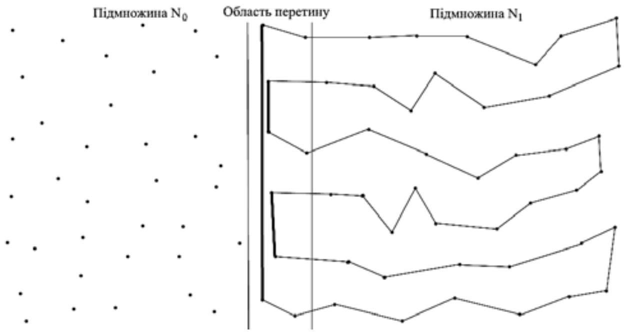

Рис. 4. Розв'язок задачі комівояжера для підмножини N* з допомогою алгоритму спільних ребер

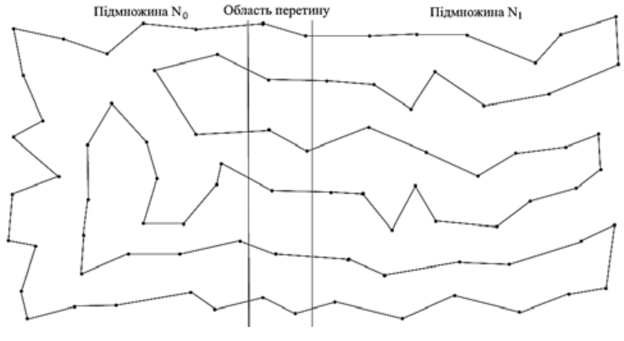

Рис. 5. Об'єднання розв'язків у підмножинах N$_0$ та N* в розв'язок задачі для підмножин N$_0$ та N$_1$

Останній крок – вилучення фіксованих ребер зі знайденого розв'язку та об'єднання його з розв'язком для підмножини N_0. У результаті одержуємо розв'язок задачі комівояжера для усіх точок підмножин N_0 та N_1. Рисунок 5 показує результат об'єднання розв'язків у підмножинах N_0 та N_1.

Алгоритм триває доти, доки всі підмножини не приєднаються до повного розв'язку вхідної множини точок.

Критеріями оцінювання роботи алгоритму для розв'язування задачі комівояжера є тривалість розв'язування задачі та якість отриманого розв'язку. Для оцінки якості розв'язку задачі використовується функція $f_{quality}$:

$$f_{quality}(M, M^*) = \frac{(lenM - lenM^*)}{lenM^*}, \tag{1}$$

де M^* – оптимальний розв'язок задачі, M – розв'язок, знайдений певним алгоритмом. При $f_{quality}(M, M^*) = 0$ маршрут M (розв'язок задачі) вважається оптимальним, інакше – наближеним. Якщо M^* є невідомим, то порівняння здійснюється з найкращим на даний час відомим розв'язком. Функція оцінки якості розв'язку може приймати від'ємне значення, що свідчить про отримання кращого розв'язку, ніж відомий.

Таблиця 1

Порівняння роботи алгоритмів

Алгоритм	**Найближчого сусіда**	**2-opt**	**Ліна – Кернігана**	**Спільних ребер**	**Декомпозиційний**
Розмір задачі	500				
Час виконання, с.	0,0001	0,14	1	0,63	**0,42**
Довжина шляху	10433	9045	8245	8957	**8919**
Оцінка якості, %	26,6	9,7	0,0001	8,6	**8,2**
Розмір задачі	1 000				
Час виконання, с.	0,01	0,58	4	3	**1,62**
Довжина шляху	28651	24930	22956	24479	**24877**
Оцінка якості, %	25,2	8,9	0,3	6,9	**8,7**
Розмір задачі	5 000				
Час виконання, с.	0,33	14,5	41	91,4	**12**
Довжина шляху	313317	272881	253447	274125	**274129**
Оцінка якості, %	24,6	8,1	0,4	8,6	**8,6**
Розмір задачі	10 000				
Час виконання, с.	1,33	80,9	121	447	**24,3**
Довжина шляху	882812	774812	721340	769672	**781281**
Оцінка якості, %	23	7,9	0,5	7,2	**8,8**

Для дослідження декомпозиційного алгоритму виконано порівняння цього алгоритму із іншими відомими евристичними алгоритмами та випробовування на задачах великої розмірності.

Вхідні параметри декомпозиційного алгоритму спільних ребер під час досліджень:

– Розмір підзадач декомпозиції – 400-600 точок;

– Величина області перетину – 30%;

– Базові алгоритми – 4 алгоритми 2-opt для отримання початкових розв'язків та алгоритм 2-opt для об'єднання розв'язків.

Декомпозиційний алгоритм порівнюється із алгоритмом найближчого сусіда, 2-opt, Ліна-Кернігана та спільних ребер. Оптимальні розв'язки знайдено з допомогою точного алгоритму програмного засобу Concorde TSP Solver [6]. Результати порівняння роботи алгоритмів для задач розмірністю 500, 1000, 5000 та 10000 точок показано в таблиці 1 та графіках на рисунках 6 і 7.

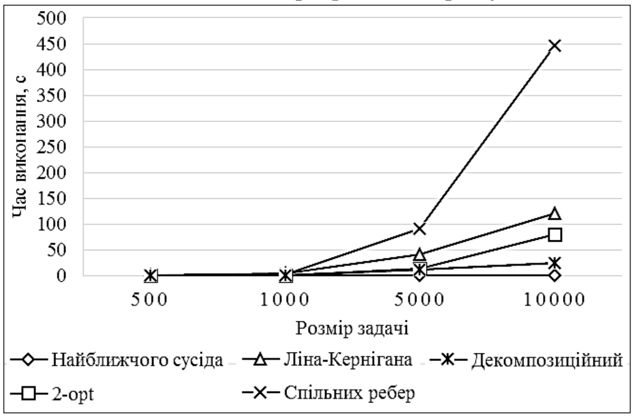

Рис. 6. Графік порівняння часу виконання алгоритмів

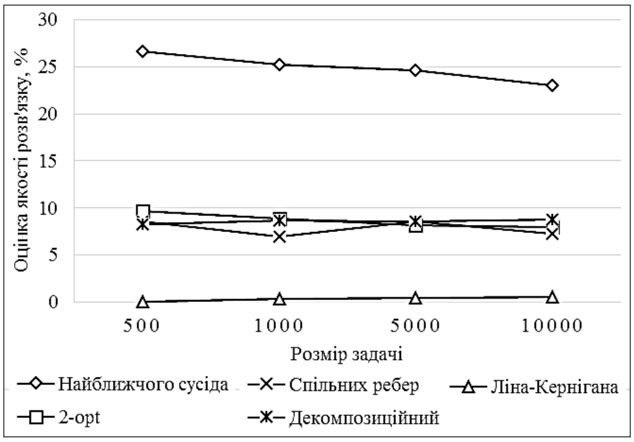

Рис. 7. Графік порівняння оцінки якості розв'язків

На основі проведеного порівняння зроблено висновок, що час розв'язування задачі декомпозиційним алгоритмом спільних ребер значно менший за відомі алгоритми – обчислювальна складність близька до лінійної. Для задачі розміром 10 000 точок декомпозиційний алгоритм виконується швидше за алгоритм Ліна-Кернігана в 5 разів, а в порівнянні з базовим алгоритмом спільних ребер – в 19 разів. Спостерігається збереження якості розв'язків із зростанням розміру задачі в межах 8,2-8,8%, що в різних випадках краще чи гірше від якості розв'язків інших алгоритмів на 1-2%.

Випробовування алгоритму на великих розмірностях проводилося для декількох наборів тестових задач однієї і тієї ж розмірності. Результати випробовування подані в таблиці 2 та графіку на рисунку 8.

Таблиця 2

Результати роботи декомпозиційного алгоритму спільних ребер для задач великої розмірності

Розмір задачі	Час виконання, хв.		
№ задачі	1	2	3
1 000 000	55,25	56,13	54,01
500 000	23,38	24,59	23,98
250 000	10,44	10,41	9,51
100 000	4,2	4	3
50 000	1,7	1,5	1,74

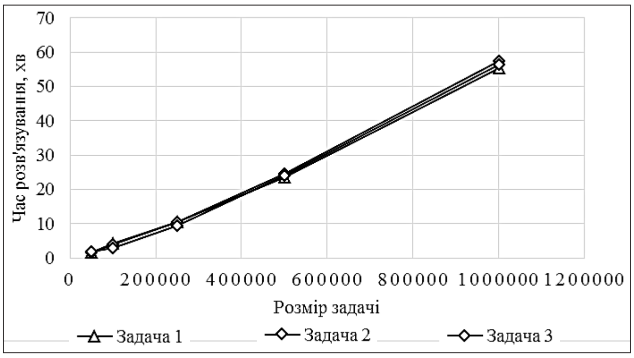

Рис. 8. Графік зростання часу розв'язування задачі із зростанням розміру задачі

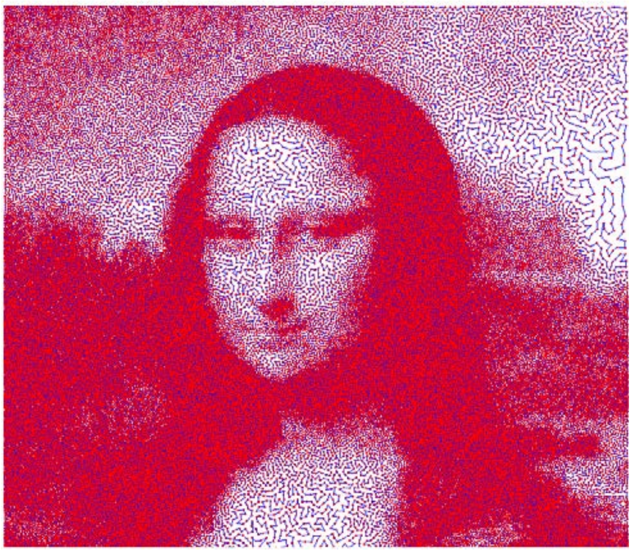

Рис. 9. Розв'язок задачі mona-lisa100K

Дослідження декомпозиційного алгоритму на задачах великої розмірності показує, що складність алгоритму близька до лінійно-логарифмічної. На основі виконаних досліджень можна зробити припущення, що алгоритм дозволяє розв'язувати задачі розмірністю понад 1 мільйон точок із збереженням якості розв'язку близькою до 8%.

З допомогою декомпозиційного алгоритму спільних ребер знайдено розв'язок відомої задачі комівояжера у форматі TSPLIB mona-lisa100K [9] (рис. 9) розмірністю 100000 вершин за 5 хвилин 18 секунд, а розв'язок задачі на 4,4% гірший за оптимальний.

Вхідні параметри декомпозиційного алгоритму спільних ребер:

– Розмір підзадач декомпозиції – 800-1200 точок;

– Величина області перетину – 30%;

– Базові алгоритми – 4 алгоритми 2-opt для отримання початкових розв'язків та алгоритм 2-opt для об'єднання розв'язків.

Висновки і пропозиції. Виконавши дослідження запропонованого і реалізованого декомпозиційного алгоритму на основі методу спільних ребер, можна зробити висновок, що його доцільно застосовувати для розв'язування задач великої розмірності – понад 100000 точок. Алгоритм показує стабільне значення якості отриманого розв'язку із зростанням розмірів задачі в межах 8%, а час розв'язування зростає лінійно. При певних налаштуваннях алгоритм дозволяє розв'язувати задачі з 1 мільйоном точок менше ніж за годину.

Декомпозиційний алгоритм спільних ребер має можливість широкого розпаралелення і може бути використаний на багатопроцесорних системах. В подальшому, важливо дослідити поведінку методу спільних ребер та декомпозиційного алгоритму на основі цього методу з використанням інших базових евристичних алгоритмів, які можуть дозволити знаходити значно якісніші розв'язки.

Список літератури:

1. Matai R. Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches / R. Matai, S. P. Singh, M. L. Mittal // Jaipur, India, 2010.
2. Land A. H. An automatic method of solving discrete programming problems / A. H. Land and A. G. Doig // Econometrica. – 1960. – 28. S. 497-520.
3. Johnson D. S. The Traveling Salesman Problem: A Case Study in Local Optimization / D. S. Johnson and L.A. McGeoch // November 20, 1995.
4. Applegate D. On the solution of traveling salesman problems / D. Applegate, R. E. Bixby, V. Chvatal and W. Cook // Documenta Mathematica, Extra Volume ICM III:645-656, 1998.
5. Applegate D. Chained Lin-Kernighan for large traveling salesman problems / D. Applegate, W. Cook, A. Rohe // INFORMS J. Computing, 15: 82-92, February 2002.
6. Concorde TSP Solver [Електронний ресурс]. – Режим доступу: http://www.math.uwaterloo.ca/tsp/concorde/index.html (2015).
7. Базилевич Р. Алгоритм приєднання часткових розв'язків у підмножинах при декомпозиції задачі комівояжера / Р. Базилевич, Р. Кутельмах // Вісн. Нац. ун-ту «Львівська політехніка». – 2009. – № 653. – С. 3-11.
8. Базилевич Р. Розв'язування задачі комівояжера великих розмірностей методом спільних ребер / Р. Базилевич, Р. Кутельмах, А. Томчук // Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. – 2014. – № 800. – С. 278-285.
9. Задача комівояжера monalisa [Електронний ресурс]. – Режим доступу: http://www.math.uwaterloo.ca/tsp/data/ml/monalisa.html (2012).

Кутельмах Р.К., Угриновский Б.В.
Национальный университет «Львовская политехника»

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ДЕКОМПОЗИЦИОННОГО АЛГОРИТМА ОБЩИХ РЕБЕР ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА БОЛЬШОЙ РАЗМЕРНОСТИ

Аннотация
В статье описано и исследовано декомпозиционный алгоритм на основе общих ребер для решения задачи коммивояжера больших размерностей. Метод общих ребер применен в качестве исходной процедуры декомпозиционного алгоритма для нахождения решений подзадач. Экспериментальные исследования показали, что алгоритм целесообразно применять для решения задач большой размерности – более 100 000 точек. Алгоритм показывает стабильное значение качества решений в пределах 8%. Время решения растет линейно-логарифмически.
Ключевые слова: задача коммивояжера, алгоритм ближайшего соседа, 2-opt, алгоритм Лина-Кернигана, метод общих ребер, декомпозиционный алгоритм.

Kutelmakh R.K., Uhrynovskyi B.V.
National University «Lviv Polytechnic»

INVESTIGATION OF THE EFFICIENCY OF COMMON EDGES DECOMPOSITION ALGORITHM FOR SOLVING LARGE-SIZE TRAVELING SALESMAN PROBLEM

Summary
The article describes and investigates decomposition algorithm for solving the large size traveling salesman problem based on common edges method. The common edges method is used as the initial procedure in decomposition algorithm for finding subproblem solutions. Experimental studies have shown that the algorithm is useful for solving large-size problems over 100 000 points. The algorithm shows a stable quality value of the solution within 8%. Solving time increases linearly-logarithmically.
Keywords: traveling salesman problem, nearest neighbor algorithm, 2-opt, Lin-Kernighan algorithm, common edges method, decomposition algorithm.