УДК 621.382

Р. Базилевич, Р. Кутельмах, А. Томчук
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОМІВОЯЖЕРА ВЕЛИКИХ РОЗМІРНОСТЕЙ МЕТОДОМ СПІЛЬНИХ РЕБЕР

Досліджено евристичні алгоритми розв'язання задачі комівояжера: найближчого сусіда, 2-Opt, 3-Opt, Ліна-Кернігана та Ліна-Кернігана-Гельсгауна. Здійснено порівняльний аналіз часу роботи алгоритмів та якості отриманих розв'язків. Запропоновано декомпозиційний метод знаходження розв’язків задачі комівояжера на основі спільних ребер.

Ключові слова: задача комівояжераа, NN, 2-Opt, 3-Opt, алгоритм Ліна-Кернігана, алгоритм Ліна-Кернігана-Гельсгауна, евристика, велика розмірність.

Existing heuristic algorithms for solving traveling salesman problem, such as Nearest Neighbor, 2-Opt, 3-Opt, Lin-Kernighan and Lin-Kernighan-Helsgaun have been investigated in this work. The algorithms have been compared in terms of running time and solution quality. Decomposition approach, based on using common edges in multiple solutions, has been proposed.

Key words: traveling salesman problem, NN, 2-Opt, 3-Opt, Lin-Kernighan, LKH, heuristics, large-scale.

Вступ

Проблема побудови оптимальних маршрутів через задану множину точок на площині чи у просторі виникає у багатьох сферах людської діяльності, наприклад, задачах планування та логістики; при виробництві друкованих плат; мінімізації рухів у робототехніці; аналізі структури ДНК та ін. За суттю, усі ці проблеми зводяться до розв'язування задачі комівояжера. Великий інтерес до задачі пов'язаний з використанням її як платформи для дослідження загальних методів комбінаторної оптимізації [1]. Порівняно новим і мало вивченим практичним застосування задачі комівояжера є малювання неперервною лінією. Роберт Бош та Адріан Герман у 2003 році запропонували використання геометричної інтерпретації задачі комівояжера, яка полягає у створенні зображень з допомогою неперервної лінії [2].

Зважаючи на актуальність, існує багато алгоритмів розв'язування задачі комівояжера. Більшість з них незастосовні до задач великих розмірностей, оскільки складність цих алгоритмів зростає експоненційно. Часто доводиться обирати між часом роботи алгоритму та якістю отриманого розв'язку. Компроміс зазвичай знаходять, комбінуючи методи пошуку шляху та його послідовної оптимізації [3].

Постановка задачі

Задачу комівояжера формулюють так: дано множину міст, а також відстань між усіма можливими парами міст. Необхідно знайти шлях, який пролягає через усі міста, та повертається у початкове, окрім того сумарна довжина пройденого шляху має бути мінімальною.

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 d_{ij} та d_{ji} ваги однакові. В симетричному випадку кількість можливих маршрутів вдвічі менша. Симетрична задача моделюється неорієнтовним графом [1].

Розглянуто симетричну задачу комівояжера, де $V=\{v_1,\ldots,v_n\}$ – множина точок, $A=\{(r,s):r,s\in V\}$ – множина ребер, де d_{rs} = d_{sr} – вартість, що асоціюється з ребром. Для розв'язання задачі комівояжера слід знайти мінімальний замкнутий шлях, що проходить через усі вершини по одному разу [4]. Точки $v_i \in V$ представлені координатами (x_i, y_i) у двовимірному просторі і відстань d_{rs} між будь-якою парою точок представляється як евклідова відстань між r та s. Часто таку задачу ще називають *евклідовою задачею комівояжера*.

Огляд існуючих алгоритмів

Всі існуючі методи розв'язування задачі комівояжера можна поділити на дві групи:

- методи знаходження оптимальних маршрутів;
- наближені методи [5].

Серед алгоритмів пошуку оптимального розв'язку задачі комівояжера відомими є метод гілок та меж, прогресивні методи покращення на основі методів лінійного програмування, комбінації методів гілок та меж та відсікальних площин. Але всі їх поєднує той факт, що вони не застосовні до задач великої розмірності, оскільки обчислювальна складність цих алгоритмів зростає експоненційно. Тому алгоритми цього типу не розглядатимуться.

Одним з найпростіших евристичних алгоритмів є **алгоритм найближчого сусіда** (англ. nearest neighbor). Цей алгоритм, а також йому подібні інколи називають конструктивними, оскільки як їх часто використовують для побудови початкового маршруту, який може піддаватися подальшій оптимізації. Основними перевагами цього алгоритму є його відносно мала обчислювальна складність, простота розуміння і "природність" за принципом дії. Кроки алгоритму:

1. Обрати довільну початкову точку.
2. Знайти найближчу невідвідану точку і перейти до неї.
3. Якщо лишаються невідвідані точки, то перейти до кроку 2.
4. Повернутися до початкової точки.

Обчислювальна складність алгоритму – $O(n^2)$. Якість отриманого розв'язку може бути гіршою від якості оптимального на 25–30 %, проте розв'язок буде отримано з мінімальними часовими затратами [6].

Найчастіше застосовуються такі алгоритми *оптимізації* маршруту, як 2-Opt та 3-Opt. Це сімейство алгоритмів часто називають k-Opt алгоритмами (рис. 1).

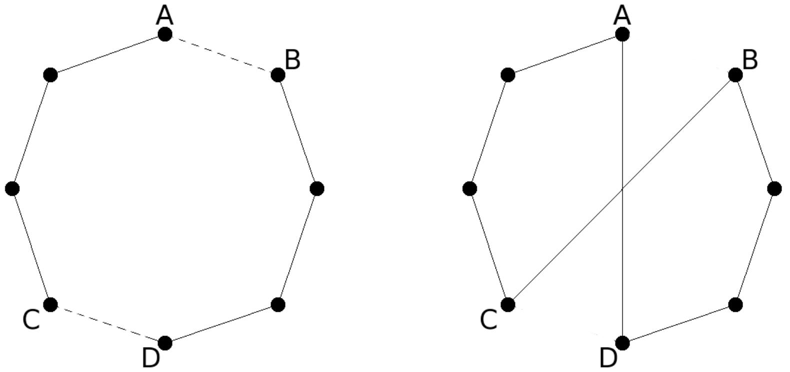


Рис. 1. Перестановка ребер при використанні алгоритму 2-Opt

Алгоритм 2-Opt полягає у видаленні двох ребер з вже існуючого шляху (отриманого за допомогою конструктивного алгоритму) і заміні їх двома новими за умови, що буде отримано новий коротший шлях. При видаленні ребра AB та DC і заміною їх новими ребрами AD та BC утвориться новий цикл, і якщо довжина нового шляху менша ніж у попереднього, то перебудовується існуючий шлях. У протилежному випадку аналізується наступне ребро, не змінюючи існуючого маршруту.

Ця дія повторюється для кожної пари ребер. Слід зазначити, що замінити ці два ребра можна лише одним способом, як показано на рис. 1, інакше буде утворено два цикли (рис. 2). Отриманий у результаті оптимізації розв'язок буде гіршим від оптимального на 4–20 %. Обчислювальна складність алгоритму – $O\left(n^2\right)$.

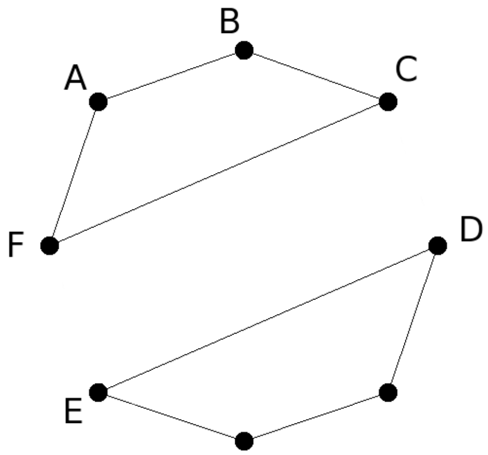


Рис. 2. Можливі утворення циклів у процесі застосування алгоритму 2-Opt

Аналогічно працює алгоритм 3-Opt. Здійснюється заміна трьох ребер. У такому випадку можна утворити дві можливі заміни, як показано на рис. 3. На практиці цей алгоритм часто реалізовується як два або три послідовні застосування алгоритму 2-Opt. Отриманий результат є гіршим від оптимального на 3–4 %. Обчислювальна складність алгоритму $O\left(n^3\right)$.

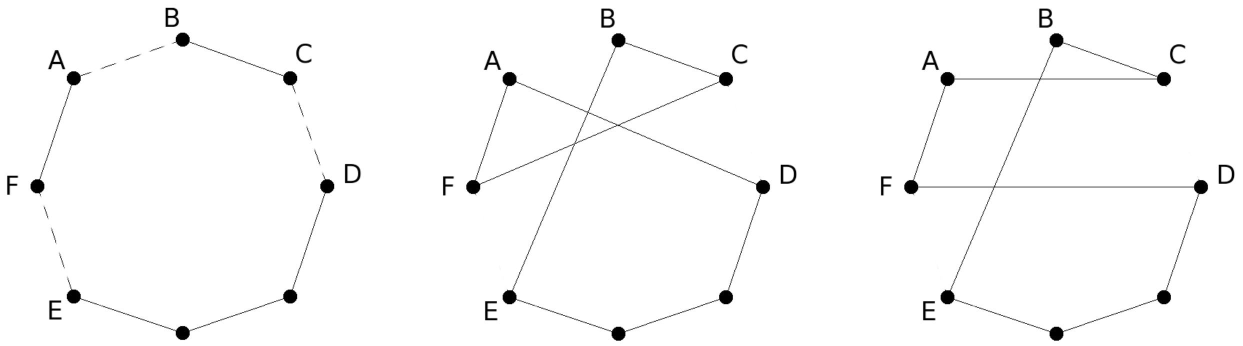


Рис. 3. Можливі перестановки при використанні алгоритму 3-Opt

Логічним продовження описаних методів оптимізації маршруту є використання “більших *k*”, тобто 4, 5 і більше. Зі збільшенням кількості робер, що видаляються, збільшується кількість можливих перестановок, що, своєю чергою, збільшує час роботи алгоритму і робить його неприйнятним для задач великої розмірності. Вдосконаленням методів типу *k-opt* є метод Ліна–Кернігана. Цей метод передбачає використання початкового субоптимального маршруту та його оптимізацію. Кількість ребер (число *k*), які будуть замінені на кожному кроці алгоритму, змінюється динамічно. Особливістю цього алгоритму є те, що він гарантує, що отриманий розв'язок для задачі розмірністю N точок буде не довшим у $4\sqrt{N}$ разів від оптимального, а також ніколи не є гіршим за початковий маршрут. Метод запропоновано в 1973 році Ліном та Керніганом. Більшість аспектів реалізації описано в оригінальній праці.

Обчислювальна складність методу становить $O\left(n^{2.2}\right)$. Отриманий результат є гіршим від оптимального на 1–3 %. Сьогодні розроблено багато модифікацій алгоритму Ліна–Кернігана: багаторівневий алгоритм Ліна–Кернігана [8], ланцюговий алгоритм Ліна–Кернігана [9], паралельна модифікація алгоритму Ліна–Кернігана [10]. Найефективнішою модифікацією алгоритму вважається метод Ліна–Кернігана–Гельсгауна, що полягає в удосконаленні правил вибору ребер, що будуть замінені [7].

Експериментальні дослідження методів найближчого сусіда, 2-opt, 3-opt, LK та LKH

З метою порівняння роботи алгоритмів здійснено експериментальні дослідження. Тестові дані містили задачі розмірністю 100, 500, 1000, 5000, 10000, 50000, 100000 точок. Тестові дані було отримано на основі модифікації вхідних даних задачі mona-liza100k. Дослідження проводили на ПК з процесором Intel Celeron M220.

Як еталонний було обрано LKH, оскільки отриманий за його допомогою результат є найближчим до оптимального. Алгоритм Ліна–Кернігана дозволяє отримати результат, що гірший від еталонного менш як на 1%, при цьому максимальне відхилення від цього результату спостерігається під час розв'язання задач розмірністю від 5000 до 50000 точок. Алгоритм 3-Opt дозволяє отримати розв'язок гірший від еталонного у межах до 10 %, залежно від розмірності задачі. Результат роботи алгоритму 2-Opt гірший від еталонного у межах 15 %, і найгірший результат було одержано за допомогою алгоритму найближчого сусіда – він гірший від еталоннного у межах 30%. Результати наведено на рис. 4 та в табл. 1.

Таблиця 1

Час роботи алгоритмів найближчого сусіда, 2-opt, 3-opt, LKH для задач розмірністю 10000–100000 точок (секунди)

Розмірність	10000	50000	100000
Алгоритм NN	0,01	0,09	0,2
Алгоритм 2-opt	0,04	0,37	1,3
Алгоритм 3-opt	0,09	0,72	2,5
Алгоритм LKH	15,6	90,3	237,3

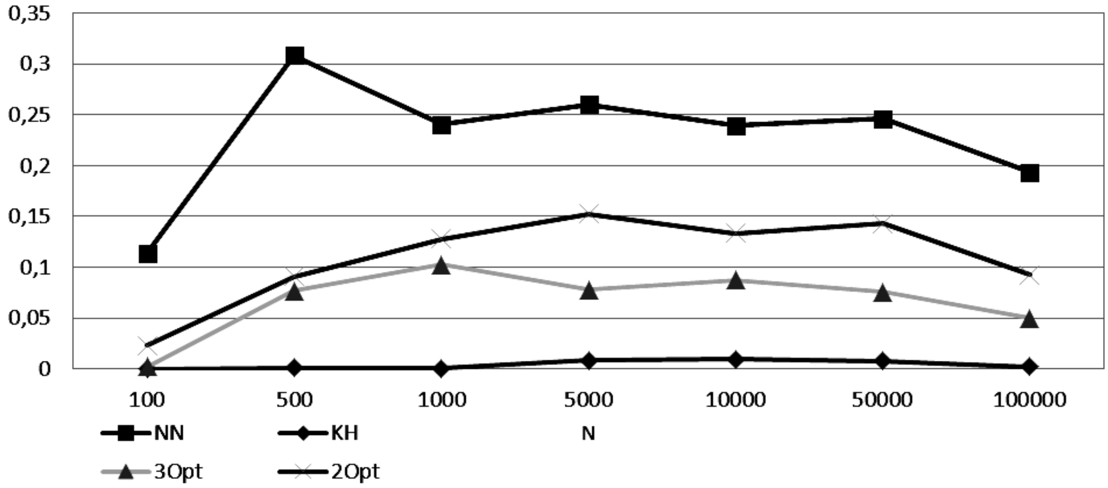


Рис. 4. Якість отриманих розв'язків задач розмірністю 100–100000 точок

Вдосконалення методів знаходження розв’язку задачі на основі спільних ребер

Пропонується метод знаходження розв’язку, що полягає в об'єднанні розв’язків, отриманих за допомогою різних алгоритмів і з меншою обчислювальною складністю. Кроки алгоритму:

1. Знаходження N розв'язків задачі комівояжера за допомогою одного чи різних методів. Допускається застосування методів з різною обчислювальною складністю. В результаті отримується N шляхів S_i (рис. 6).
2. Маючи множину з N шляхів, виділяється множина ребер R, які належать до всіх шляхів, отриманих на кроці 2 (рис. 7).
3. Зшивання отриманих послідовностей ребер у розв’язок задачі.

Для прикладу припустимо, що перед початком роботи алгоритму є деяка множина точок (рис. 5) та декілька розв’язків для цієї множини точок, отримані за допомогою деякого евристичного методу.

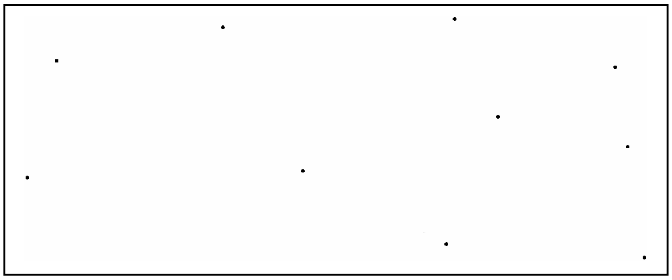
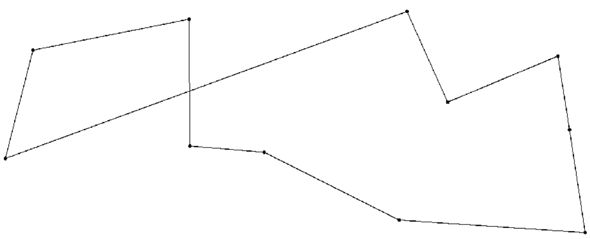
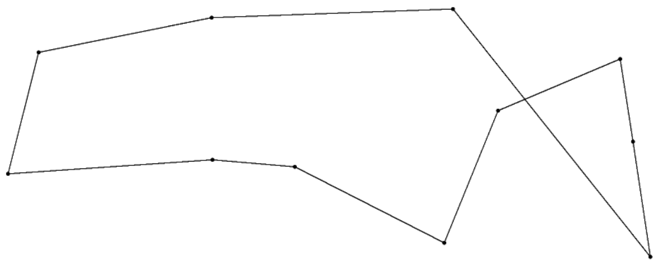


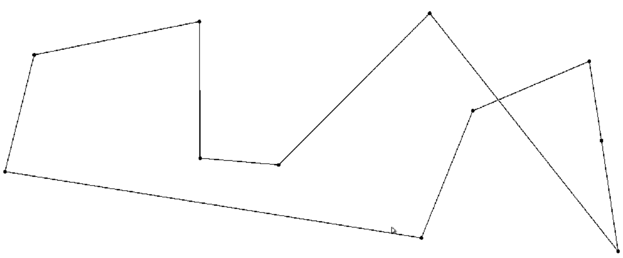
Рис. 5. Початковий набір точок



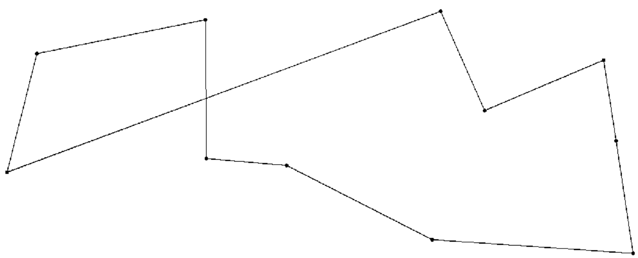
а



б



В



г

Рис. 6. Розв’язки для даної множини точок

Множину ребер, що входять до всіх розв’язків, показано на рис. 7. Наступним кроком є пошук найкоротших відстаней між не з'єднаними послідовностями ребер (рис. 8).

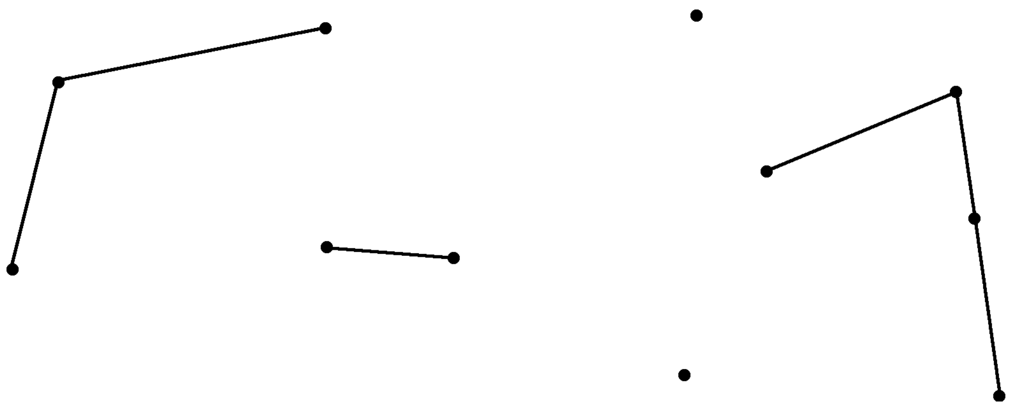


Рис. 7. Ребра, що входять до усіх шляхів

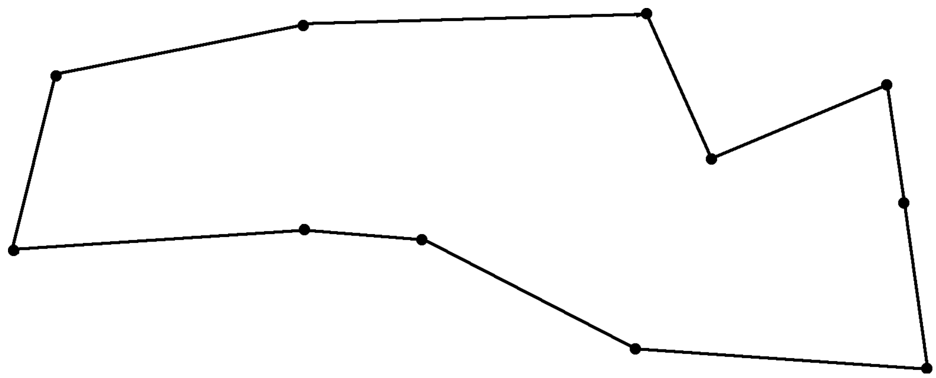


Рис. 8. Результат об'єднання

Проведено експериментальні дослідження запропонованого методу. Як тестову обрано задачу розмірністю 264 точки. Початковими розв'язками задачі було обрано 6 різних маршрутів, одержаних за допомогою методу 2-opt (довжина кожного з яких на 13 % більша від оптимального). Також з метою порівняння з відомими алгоритмами цю задачу було розв'язано за допомогою алгоритмів найближчого сусіда та LKH. Результати наведено в табл. 2.

Таблиця 2

Довжина маршрутів для задачі розмірністю 264 точки, отриманих за допомогою алгоритму спільних ребер та відомих методів відносно оптимального

Алгоритм найближчого сусіда	Алгоритм 2-opt	Алгоритм на основі спільних ребер	Алгоритм LKH
28.7 %	13 %	6,6 %	Оптимальний розв'язок

Отриманий за допомогою алгоритму спільних ребер розв'язок гірший від оптимального на 6,6 % при використанні алгоритму 2-opt для об'єднання фрагментів шляхів. На рис. 9 показано спільні ребра та розв'язок досліджуваної задачі, отриманий за допомогою методу спільних ребер.

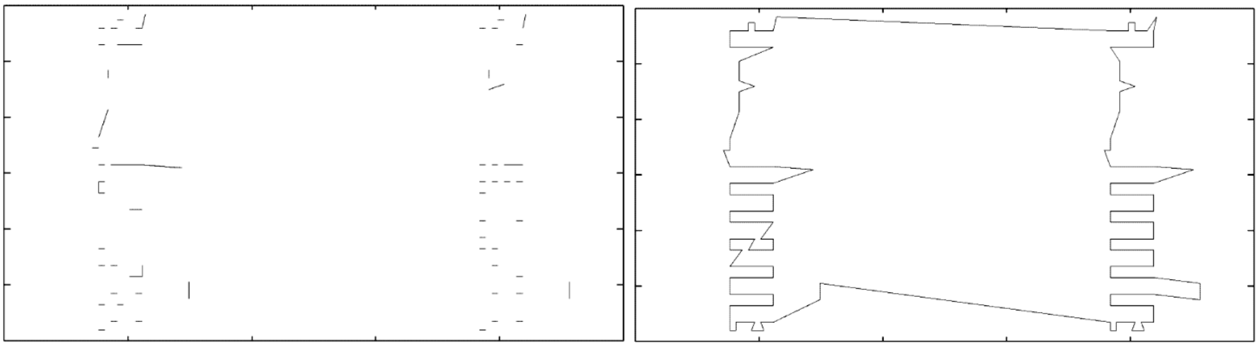


Рис. 9. Спільні ребра та розв'язок, знайдений на їх основі для задачі розмірністю 264 точки

Цю методику доцільно застосувати для розв'язування задач великих розмірностей, зокрема в областях знаходження локальних розв'язків задачі (як додаткову початкову процедуру в методі знаходження розв'язку на основі його нарощення, рис. 10).

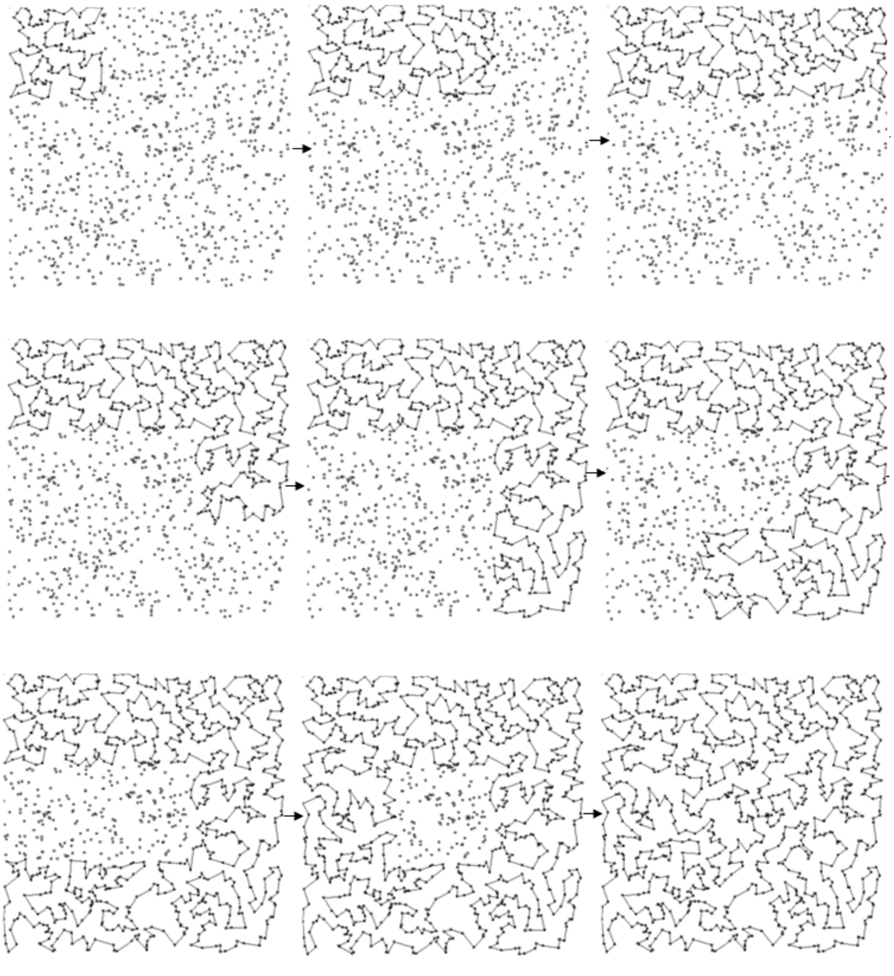


Рис. 10. Знаходження розв'язку задачі на основі нарощення часткового розв'язку

Висновки

У результаті проведених досліджень було проаналізовано алгоритми найближчого сусіда, 2-Opt, 3-Opt, Ліна–Кернігана та Ліна–Кернігана–Гельсгауна. Аналіз отриманих результатів дає змогу зробити припущення про те, що конструктивні алгоритми не є ефективними. У переважній більшості випадків алгоритм Ліна–Кернігана чи Ліна–Кернігана–Гельсгауна є найефективнішими.

Запропоновано методику знаходження розв’язку задачі на основі спільних ребер, що дає можливість широкого розпаралелення та знаходження високоякісних розв’язків задачі. Метод може бути застосованим до задач великих розмірностей. Отриманий за допомогою алгоритму спільних ребер розв'язок гірший від оптимального на 6,6 %. Подальшою роботою буде детальніше дослідження запропонованого методу та його вдосконалення.

1. R. Matai, S.P. Singh, M.L. Mittal, “Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches ”, Jaipur, India, 2010, 2. 2. R. Bosch and A. Herman, “Continuous line drawings via the traveling salesman problem,” Operations Research Letters 3 (2004) 302-303, 3. Р. Базилевич, Р. Кутельмах // Комп'ютерні науки та інформаційні технології [Текст] : [зб. наук. пр.] / відп. ред. Ю. М. Рашкевич. – Л. : Видавництво Національного університету “Львівська політехніка”, 2009. – 279 с. : іл. – (Вісник / Національний університет “Львівська політехніка”; № 650). – С. 235–244, 4.E. Nood and J. Been, An Efficient Transformation of the Generalized Travaling Salesman Problem, October 1991, 5. D.S. Johnson and L.A. McGeoch, “The Traveling Salesman Problem: A Case Study in Local Optimization”, November 20, 1995., 6. Christian Nilsson , Heuristics for the Traveling Salesman Problem, Linkoping Univercity, 7. Keld Helsgaun , General k-opt submoves for the Lin–Kernighan TSP veling heuristic Keld Helsgaun, 1 July 2009, 8. Chris Walsaw, A multilevel lin-kernigan-helsgaun algorithm for the treveling salesman problem, 9. D.Applegate, W.Cook, A.Rohe, Chained Lin-Kernigan for large traveling salesman problems, Algorithms and Optimization Departament, AT&T Labs-Research 10.Verhoeven, M.G.A. & Aarts, E.H.L. A parallel Lin-Kernighan algorithm for the traveling salesman problem, 1994.