МОДЕЛЮВАННЯ ПРОЦЕСІВ І СИСТЕМ

УДК 519.16

Р. Базилевич, Р. Кутельмах, Б. Кузь
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

АЛГОРИТМИ КЛАСТЕРИЗАЦІЇ РОБОЧОГО ПОЛЯ З ОБМЕЖЕННЯМИ ДЛЯ ЗАДАЧІ КОМІВОЯЖЕРА

Описано три підходи до кластеризації робочого поля для задачі комівояжера, що забезпечує поділ множини точок на частини з заданими обмеженнями. Один із відомих алгоритмів використовується для отримання розв’язків в кожному кластері з подальшим зшиванням часткових розв’язків.

Ключові слова: задача комівояжера, кластеризація.

Article describes three approaches to clustering set of points of TSP into subsets with given constraints. One of the well-known basic algorithms is used for solutions at every cluster with further joining of partial solutions.

Keywords: traveling salesman problem, clustering.

Вступ

Для розв’язання широкого кола прикладних проблем використовується як базова задача комівояжера. Одним із ефективних алгоритмів пошуку наближеного розв’язку задачі є алгоритм Ліна–Кернігана–Гельсгауна [1]. Обчислювальна складність алгоритму становить $O(n^{2,2})$, що робить його, як і інші евристичні алгоритми, малопридатними для задач великих та надвеликих розмірностей. Для зменшення затрат часу на пошук розв’язку запропоновано декомпозиційний алгоритм, в якому використовується розбиття множини даних на частини з подальшим пошуком часткових розв’язків для кожної з них та зшиванням їх у загальний розв’язок. Постає проблема вибору алгоритму кластеризації множини вхідних даних з метою виділення множини точок, для яких будуть знаходитись часткові розв’язки. Від розв’язання цієї задачі залежатиме якість остаточного рішення.

Формулювання задачі

Для задачі комівояжера заданими вважають множину $P=\{p_1, p_2, \ldots, p_N\}$ з N точок, які описані їх координатами $p_i=(x_i, y_i)$, $\forall i \in \{1, 2, \ldots, N\}$ та функцію віддалі $dist: P\times P \to R$, яку оцінюємо евклідовою метрикою:

$$r_{ij} = dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\,,\ \forall i, j \in \{1, 2, \ldots, N\}.$$

Нехай H – множина всіх допустимих розв’язків, тобто можливих варіантів замкнутих маршрутів, що обходять всі точки з множини P лише по одному разу; M' – деякий маршрут, що належить множині H та є розв’язком задачі:

$$M' = \langle m_1, m_2, \ldots, m_N \rangle,\ m_i \in P,\ M' \subset H,$$

та L' – функція довжини маршруту:

$$L' = len\ M' = r_{N,\ 1} + \sum_{i=1}^{N-1} r_{i,\ i+1}.$$

Необхідно знайти маршрут M^* мінімальної довжини:

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

Задача належить до комбінаторних класу NP, оскільки має обчислювальну складність 0(N!). Знаходження точного розв'язку для задач цього класу у випадку їх великої розмірності є практично неможливим, тому актуальним є розроблення наближених методів, які забезпечують добру якість та мають близьку до лінійної обчислювальну складність.

Методика розв'язання

Знаходження розв'язку для задачі великої розмірності поділяється на такі етапи:

1. Кластеризація робочого поля.
2. Пошук розв'язків задачі комівояжера окремо для кожного з кластерів.
3. Зшивання часткових розв'язків в єдиний для всієї задачі.
4. Оптимізація розв'язку.

Кластеризація робочого поля передбачає поділ множини точок на підмножини. Розв'язок для кожного з кластерів шукають за допомогою одного із відомих алгоритмів для задачі комівояжера, наприклад, Ліна–Кернігана–Гельсгауна [1]. Сумарні затрати часу на пошук окремих розв'язків для всіх кластерів є значно меншими, ніж затрати часу на пошук розв'язку для всієї множини точок. Часткові розв'язки задачі для кожного з кластерів зшиваються в загальний розв'язок для цілої задачі. Отриманий так розв'язок буде гіршим, ніж результат розв'язання задачі для всієї множини точок, але час розв'язання задачі – значно меншим. Можлива подальша оптимізація розв'язку, зокрема алгоритмами, запропонованими у [2–3].

Розглянемо алгоритми для першого етапу – кластеризації робочого поля. Результатом є множина кластерів, кожен із яких містить обмежену кількість точок. Оскільки якість кінцевого результату значно залежить від таких параметрів, як розмір кластерів, кількість включених в них точок, їх форми та інших, доцільно розробити алгоритми кластеризації та дослідити їх вплив на якість та час розв'язування задачі.

Кластеризація даних

На першому кроці множина точок піддається тріангуляції, яка розбиває робоче поле на плоскі фігури, одна з яких – зовнішня, а всі інші – внутрішні, що утворюють трикутники. Для тріангуляції Делоне жодна з точок не входить всередину іншого з описаних навколо трикутників кіл [4]. Вперше таку тріангуляцію було запропоновано в 1934 році [5]. Результатом тріангуляції є множина трикутних граней, для кожної з яких визначено три суміжні трикутники (крім тих, які розміщуються на границі робочого поля, вони мають лише два суміжні трикутники).

Кожен кластер включає деяку множину точок. Приналежність трикутника до одного з кластерів визначається входженням до нього всіх його трьох точок. Кластер має визначену множину суміжних трикутників, до якої входять ті, що не належать до нього, мають одну або дві точки, які до нього входять.

Приєднання трикутника до кластера – це включення всіх його точкок, які ще не належать кластерові та відповідне коригування множини суміжних трикутників. Кожний з трикутників додається до списку суміжних за умови, що вершини, протилежні до спільного ребра для даного трикутника та суміжного, не належать до жодного з інших кластерів. В іншому випадку трикутник не додається до списку суміжних.

Розроблено три алгоритми кластеризації множини точок шляхом поширення хвилі:

- приєднанням найменшого суміжного трикутника;
- приєднанням суміжного трикутника;
- приєднанням трикутника з найменшим ребром.

Основним параметром для обмеження розмірів кластерів є кількість їх точок. Проте можуть бути обрані також і інші, наприклад обмеження на площу приєднаного трикутника порівняно з площею першого, що утворює кластер.

Алгоритм кластеризації приєднанням найменшого суміжного трикутника

Кластеризація складається з таких етапів:

1. Побудова тріангуляція Делоне. Формування множини всіх трикутників.

2. Визначення площі для кожного з трикутників та впорядкування їх за площею.

3. Розширення вже утвореного фрагмента кластера додаванням найменшого за площею суміжного трикутника. На першому кроці вибирається найменший серед усіх трикутників утворених в процесі тріангуляції. Вилучення його з множини утворених в п. 1 трикутників.

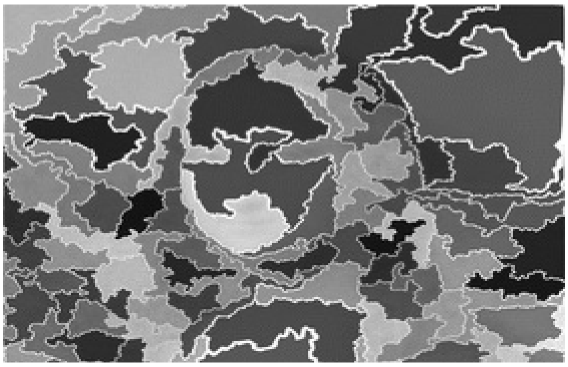

Рис. 1. Кластеризація приєднанням найменшого суміжного трикутника

4. Закінчення процесу кластеризації, якщо множина утворених в п. 1 трикутників є порожньою. У протилежному випадку – перехід на п. 5.

5. Формування наступного кластера, якщо число точок вже утвореного фрагмента відповідає заданому обмеженню. У протилежному випадку – повернення на п. 3.

Можливе виникнення ситуації, коли після побудови всіх кластерів залишаються точки, які не належать до жодного з кластерів. Для таких точок виконується пошук найближчого кластера та додання точки до нього.

Після виконання описаних операцій утворюються множина кластерів. Можливе утворення кластерів із значно меншою кількістю точок, ніж задана в параметрі алгоритму. Це пов’язано з тим, що деяка множина вершин може опинитися замкненою всередині одного або більшого числа кластерів, що унеможливлює утворення кластера необхідного розміру. Постає необхідність введення додаткового параметра – найменшої кількості точок у кластері. Для цього параметра можливе використання значення, яке становить деяку частину від заданої параметром алгоритму кількості точок у кластері.

Для кожного з кластерів, кількість точок в яких є меншою за задану мінімально допустиму, виконується пошук усіх суміжних кластерів та визначення числа ребер, які з’єднують кожен з суміжних кластерів із поточним. Кластер об’єднується із суміжним кластером, який з’єднується з поточним найбільшою кількістю ребер. У результаті утворюється множина кластерів із кількістю точок в кожному кластері, не меншою за задану мінімально допустиму.

Алгоритм кластеризації приєднанням суміжного трикутника

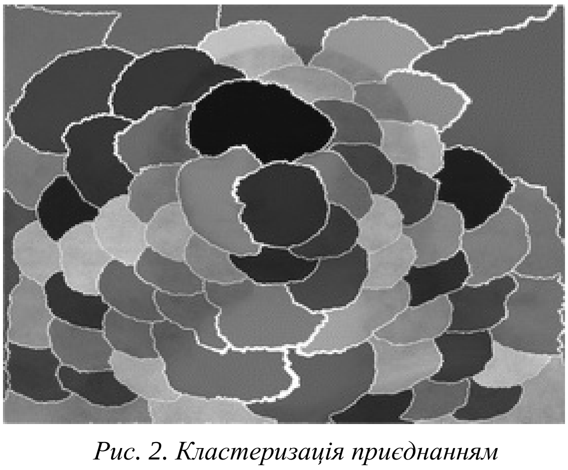

Рис. 2. Кластеризація приєднанням суміжного трикутника

Кроки 1, 4 та 5 для наступних двох алгоритмів є аналогічні, як і в попередньому. Для алгоритму кластеризації приєднанням суміжного трикутника відмінними є тільки кроки:

2. Визначення всіх суміжних до вже утвореного фрагмента трикутників. На першому кроці як початковий фрагмент береться деякий зі середини робочого поля трикутник.

3. Розширення хвилею вже утвореного фрагмента кластера з деякого суміжного трикутника з множини, визначених на попередньому кроці.

Алгоритм кластеризації приєднанням трикутника з найменшим ребром

Для алгоритму кластеризації приєднанням трикутника з найменшим ребром відмінними є такі кроки:

2. Визначення множини всіх ребер, утворених в п. 1 трикутників та впорядкування їх за довжиною. Визначення всіх суміжних до вже утвореного фрагмента трикутників та виділення серед них трикутника з найменшим ребром. На першому кроці береться трикутник з найменшим ребром.

3. Розширення вже утвореного фрагмента кластера додаванням виділеного на попередньому кроці трикутника.

Результати експериментів

Проведено кластеризацію робочого поля розміром 100000 точок з використанням трьох описаних алгоритмів. Експерименти здійснені на задачі Mona Lisa [6]. Використовувались такі параметри кластеризації: розмір кластера – 1500 точок, для алгоритму додавання найменшого суміжного трикутника мінімальний розмір кластера становив 300 точок. Результат кластеризації цим алгоритмом наведено на рис. 1, алгоритмом приєднання суміжного трикутники – на рис. 2, алгоритмом приєднання трикутника з найменшим ребром – на рис. 3.

Кластери, утворені в результаті використання методу поширення хвилі трикутниками мають найбільш круглу форму; форма кластерів, утворених в результаті використання другого та третього алгоритму залежить від щільності розміщення точок, утворені кластери виділяють окремі зони робочого поля з врахуванням їх густини.

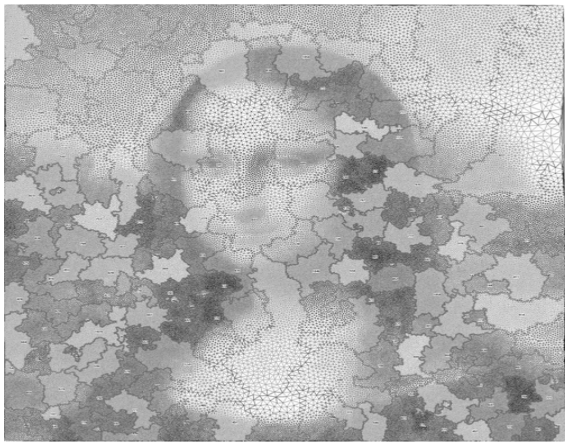

Рис. 3. Кластеризація приєднанням трикутника з найменшим ребром

Висновки

Кластери, утворені за допомогою описаних алгоритмів, можуть бути використані для розв'язування задачі комівояжера для кожного з них окремо з наступним об'єднанням часткових розв'язків в спільний для всієї множини точок робочого поля. Такий підхід значно зменшує затрати часу на розв'язання задачі для великих та надвеликих розмірностей та дозволяє широко розпаралелити цей процес. Очікується незначне погіршення якості одержаного результату порівняно з розв'язком без кластеризації, проте цей вплив може бути послаблений застосуванням додаткових алгоритмів оптимізації [2, 3].

1. Helsgaun K. An effective implementation of the Lin–Kernighan Traveling Salesman Heuristic, 2002. 2. Базилевич Р.П., Кутельмах Р.К. Алгоритм оптимізації розв'язків задачі комівояжера у локальній області // Наук.-техн. журнал "Радіоелектронні і комп'ютерні системи". – Харків, 2009. – №7 (41). – С. 41–45. 3. Базилевич Р.П., Кутельмах Р.К. Оптимізація розв'язків задачі комівояжера методом послідовного сканування // Вісник Нац. ун-ту "Львівська політехніка" № 638, Львів, 2009. – С. 254–260. 4. Скворцов А.В. Триангуляция Делоне и её применение. – Томск: Изд-во Том. ун-та, 2002. 5. Делоне Б.Н. О пустоте сферы // Изв. АН СССР. ОМЕН, 1934. – № 4. – С. 793–800. 6. Kaplan C.S., Bosch R. TSP Art // Proceedings of Bridges 2005: Mathematical Connections in Art, Music and Science.