УДК 519.16

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

ОСОБЛИВОСТІ ДЕКОМПОЗИЦІЇ ЗАДАЧІ КОМІВОЯЖЕРА

Розроблено алгоритми декомпозиції задачі комівояжера. Запропоновано виділити чотири етапи розв’язання задачі: кластеризація множини вхідних точок, розв’язання часткових задач у виділених кластерах, зшивання часткових розв’язків у загальний розв’язок та його оптимізація. Метод дає змогу зменшити затрати часу на пошук розв’язку з незначними втратами якості.

Ключові слова: задача комівояжера, комбінаторна оптимізація, декомпозиція.

Decomposition algorithms for TSP are developed. The paper proposes to split the full problem into four stages: decomposition into clusters, finding the partial solutions for each cluster, merging partial solution into full solution and its optimization. Approach could reduce the computation time with small quality losses.

Keywords: traveling salesman problem, combinatorial optimization, decomposition

Вступ

Задача комівояжера – одна з найвідоміших задач комбінаторної оптимізації, яка полягає в пошуку найкоротшого маршруту через всі точки з поверненням в його початок. Задача належить до комбінаторних класу *NP*, оскільки має обчислювальну складність *O(n!)*. Для великих та надвеликих розмірностей неможливо віднайти точний розв’язок за прийнятний час. Із зростанням розмірності множини точок істотно зростає час пошуку розв’язку. Одним з найефективніших для розв’язання задачі є алгоритм Ліна–Кернігана–Гельсгауна (LKH) [1]. Найбільша задача комівояжера, для якої відомий точний розв’язок, містить 85900 точок. Пошук розв’язку зайняв майже 136 років процесорного часу на кластері з ПК з процесорами Intel Xeon 2,8 GHz та AMD Opteron [2]. Сьогодні не існує ефективних алгоритмів розв’язання задачі для великих та надвеликих розмірностей. Для таких задач виникає необхідність у декомпозиції з метою утворення підзадач із заданою розмірністю. Це дає змогу за допустимий час знаходити оптимальні часткові розв’язки.

Існуючі алгоритми мають високу обчислювальну складність (не менше $O(n^2)$), що робить їх малопридатними для розв’язування задач великих розмірностей. Ці методи переважно реалізують послідовний підхід і погано піддаються розпаралеленню для реалізації на суперкомп’ютерах та розподілених GRID системах. У зв’язку з цим необхідним є розроблення методологій, алгоритмічного та програмного забезпечення для розв’язання задач великих розмірностей (понад 10^6 точок) з високою якістю та близькою до лінійної обчислювальною складністю. Для побудови ефективних алгоритмів розв’язання такого класу задач запропоновано декомпозиційний підхід з використанням заданої множини точок.

Методологія розв’язання задачі

Пошук розв’язку для задач великих та надвеликих розмірностей пропонується виконувати в чотири етапи. На першому етапі відбувається розбиття множини точок на підмножини із заданою обмеженою розмірністю. Необхідно виділити кластери точок, для яких можливе розв’язання задачі за невеликий час з високою якістю. Параметрами алгоритму є обмеження на максимальну та мінімальну кількість точок у кластері. Другий етап містить пошук часткових розв’язків задачі для кожного з утворених кластерів. Використовується один з відомих алгоритмів. 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. Третій етап – зшивання часткових розв’язків у загальний за допомогою методу "кільця" [3]. Отриманий так розв'язок – гірший, ніж зі розв'язання задачі для повної множини точок. Проте значно зменшується час розв'язання задачі. Для послідовної оптимізації розв'язку доцільно використати один або декілька оптимізаційних алгоритмів [4, 5]. Псевдокод алгоритму наведено в табл. 1.

Таблиця 1

Псевдокод алгоритму розв'язання задачі

Алгоритм: Пошук розв'язку задачі комівояжера
Вхідні дані: Множина точок *S*
Результат: Маршрут
1. Кластеризація множини даних 2. Пошук розв'язку для кожного з кластерів окремо 3. Зшивання часткових розв'язків в загальний 4. Оптимізація розв'язку

Кластеризація робочого поля

Кластеризація виконується на попередньо побудованій тріангуляції Делоне [6]. У такій тріангуляції жодна з точок не входить в середину іншого з описаних навколо трикутників кіл. Результатом тріангуляції є множина трикутних граней, для кожної з яких визначено три суміжні трикутники. Трикутники, які розміщуються на границі робочого поля, мають лише два суміжні трикутники.

Розроблено два алгоритми кластеризації робочого поля: приєднанням найближчого суміжного трикутника та приєднанням трикутника з найменшою площею. Псевдокоди кластеризації зображено в табл. 2, 3 та 4.

Таблиця 2

Псевдокод алгоритму кластеризації

Алгоритм: Кластеризація
Вхідні дані: Множина точок S, $N = \|S\|$, максимальна та мінімальна кількість точок в кластері N_{max} та N_{min}
Результат: Множина кластерів P
1. Побудова тріангуляції Делоне 2. Впорядкування множини трикутників 3. Чи утворено всі кластери? а. Утворення нового кластера з розмірністю на більше N_{max} б. Перехід на п.3 4. Пошук всіх точок, які не належать до жодного кластера і приєднання їх до найближчого кластера 5. Пошук кластерів, кількість точок в яких менша за N_{min} та приєднання їх до сусідніх кластерів

Таблиця 3

Утворення кластера приєднанням найближчого суміжного трикутника

Алгоритм: Утворення нового кластера приєднанням найближчого суміжного трикутника
Вхідні дані: Множина точок S, $N = \|S\|$, максимальна та мінімальна кількість точок в кластері N_{max} та N_{min}
Результат: Множина кластерів P
1. Пошук найближчого суміжного трикутника, який не належить до жодного з кластерів 2. Чи кількість точок в кластері не перевищує N_{max}? а. Додавання трикутника до кластера б. Встановлення поточного кластера для трикутника в. Перехід на п.1 3. Кластер сформовано

Рис. 1. Кластеризація приєднанням трикутника з найменшою площею

Рис. 2. Загальний розв'язок задачі

Таблиця 4

Утворення кластера приєднанням найменшого трикутника

Алгоритм: Утворення нового кластера приєднанням найменшого суміжного трикутника
Вхідні дані: Множина точок S, $N = \|S\|$, максимальна та мінімальна кількість точок в кластері N_{max} та N_{min}, найбільше допустиме відношення між площами трикутників в кластері W
Результат: Множина кластерів P
1. Пошук суміжного трикутника з найменшою площею, який не належить до жодного з кластерів 2. Чи кількість точок в кластері не перевищує N_{max} і відношення площі трикутника до площі першого трикутника кластера не перевищує W? а. Додавання трикутника до кластера б. Встановлення поточного кластера для трикутника в. Перехід на п.1 3. Кластер сформовано

У результаті кластеризації можливим є утворення кластерів, розмірність яких значно менша за задану в параметрах алгоритму. Постає необхідність усунення таких кластерів. Їх приєднують до суміжного кластера.

Таблиця 5

Приєднання кластерів з малою розмірністю

Алгоритм: Приєднання малих кластерів
Вхідні дані: Множина кластерів P, мінімальна кількість точок в кластері N_{min}
Результат: Множина кластерів P
1. Пошук кластера з розмірністю, меншою за N_{min} 2. Кластер знайдено? а. Пошук всіх суміжних кластерів б. Визначення середньої площі трикутника для кожного з суміжних кластерів в. Пошук кластера, середня площа трикутника якого найменш відрізняється від середньої площі трикутника поточного г. Приєднання поточного кластера до знайденого д. Перехід на п.1. 3. Об'єднання кластерів завершено

Процедура приєднання передбачає пошук всіх суміжних кластерів. Обчислюється середня площа трикутників для кожного з них. Кластер приєднується до суміжного з найбільш подібною середньою площею трикутників. У табл. 5 зображено псевдокод алгоритму.

Пошук загального розв'язку задачі

Етап кластеризації забезпечує можливість знаходити розв'язки для кожного з кластерів незалежно. Для розв'язання задачі в межах утворених кластерів можна застосувати будь-який з відомих алгоритмів, що забезпечують високу якість результату (наприклад, LKH [1]). Час знаходження розв'язку та його якість залежатимуть від вибраного алгоритму.

Отримані часткові розв'язки піддаються зшиванню за допомогою методу "кільця" [3]. Псевдокод алгоритму зображено в табл. 6.

Таблиця 6

Зшивання часткових розв'язків

Алгоритм: Зшивання часткових розв'язків
Вхідні дані: Множина часткових розв'язків, глибина внутрішньої зони кільця G_{in} та глибина зовнішньої зони G_{out}
Результат: Загальний розв'язок задачі
1. Для кожного з кластерів а. Формування внутрішньої зони шляхом поширення хвилі на G_{in} трикутників всередину кластера б. Формування зовнішньої зони шляхом поширення хвилі на G_{out} трикутників назовні кластера в. Пошук фіксованих ребер г. Пошук розв'язку задачі для утвореного "кільця" з врахуванням фіксованих ребер д. Заміна фіксованих ребер частинами загального шляху е. Приєднання кластера до загального розв'язку завершено

Після звивання розв'язок задачі піддається оптимізації. Для одержання якісного розв'язку необхідною умовою є послідовне застосування декількох оптимізаційних алгоритмів. Деякі з алгоритмів запропоновано в [4, 5].

Висновки

Запропоновано та деталізовано алгоритми деокмпозиції задачі комівояжера для реалізації методу "кільця". Метод розбиває задачу на етапи кластеризації, знаходження часткових розв'язків, їх зшивання в загальний розв'язок та оптимізації. Всі часткові розв'язки для виділених кластерів можуть бути знайдені незалежно. Запропоновані алгоритми дозволять зменшити час розв'язання задачі на суперкомп'ютерах та GRID системах за рахунок розпаралелення на етапах пошуку часткових розв'язків та їх зшивання.

1. K. Helsgaun, "An effective implementation of the Lin–Kernighan Traveling Salesman Heuristic", 2002. 2. Applegate D., Bixby R., Chvátal V., Cook W., Espinoza D., Goycoolea M., and Helsgaun K. Certification of an Optimal TSP Tour Through 85,900 Cities // Operations Research Letters – 37 (2009). – pp. 11—15. 3. Базилевич Р.П., Кутельмах Р.К., Кузь Б.О. Алгоритм розв'язання задачі комівояжера великої розмірності методом "тора" // Вісник Нац. ун-ту "Львівська політехніка" – 2010. – № 686. – С. 179–182. 4. Базилевич Р.П., Кутельмах Р.К. Алгоритм оптимізації розв'язків задачі комівояжера у локальній області" // Наук.-техн. журнал "Радіоелектронні і комп'ютерні системи" 2009,№7 (41). – С. 41–45. 5. Базилевич Р.П., Кутельмах Р.К. Оптимізація розв'язків задачі комівояжера методом послідовного сканування // Вісник Нац. ун-ту "Львівська політехніка" 2009. – № 638. – С. 254–-260. 6. Делоне Б.Н. О пустоте сферы // Изв. АН СССР. ОМЕН. 1934. № 4. – С. 793–800.