УДК 004.02

К.т.н., доцент Зорін Ю.М., магістрант Подольський С.В.

Національний технічний університет України «Київський політехнічний інститут»

АЛГОРИТМ ПЕРЕТИНУ ЛОКАЛЬНИХ ОПТИМУМІВ ДЛЯ ОПТИМІЗАЦІЇ ЗАДАЧІ ПРО КВАДРАТИЧНІ ПРИЗНАЧЕННЯ

Abstract

Yuri Zorin, Assoc. Prof., PhD; Sergey Podolsky, master student

An algorithm for quadratic assignment problem based on local optima intersection

The paper describes a new metaheuristic algorithm for the quadratic assignment problem optimization based on local optimum search. A new technique for fast locally optimal solution search is proposed. To prevent premature convergence the storing of solution hashes and diversification procedure are introduced. The comparative analysis of the algorithm efficiency is carried out. The prospects for further research and modification are proposed.

Вступ

Задача про квадратичні призначення є однією із фундаментальних задач комбінаторної оптимізації з категорії задач про розташування об'єктів [1]. Задача моделює наступну проблему із реального життя: дано множину із N об'єктів та множину із N місцеположень. Для кожної пари місцеположень задана відстань і для кожної пари об'єктів задана вага або потік (тобто кількість поставок, що транспортуються між двома об'єктами). Задача полягає в розташуванні всіх об'єктів у різних місцеположеннях з метою мінімізації суми відстаней, помножених на відповідні потоки.

Формальна постановка задачі наступна: дано дві множини – P (об'єкти) та L (призначення) рівних потужностей, на яких визначена функція ваги $w : P \times P \to R$ та функція відстані $d : L \times L \to R$. Необхідно знайти таке відображення $f : P \to L$ (призначення), що мінімізує цільову функцію

$$\sum_{a,b \in P} w(a,b) * d(f(a), f(b))$$

Задача є NP-складною, тому не існує алгоритму її розв'язання за поліноміальний час. Алгоритмічна складність задачі становить $N^2 N!$.

Постановка задачі

Метою роботи є розробка ефективного метаевристичного алгоритму оптимізації задачі про квадратичні призначення за заданими вхідними матрицями потоків між об'єктами та відстаней між місцеположеннями.

Загальні положення, опис та формулювання алгоритму

Запропонований алгоритм є ітеративним. В основі роботи алгоритму лежить використання локальних оптимумів, що не повторюються. Пошук локального оптимуму ґрунтується на тому, що за поліноміальний час [2] можна перевірити доцільність всіх $N(N-3)/2$ можливих перестановок пар елементів в деякому векторі, що є кандидатом на розв'язок задачі розміром N.

Очевидно, що при перевірці, або навіть здійсненні, деякої перестановки розглядати її знову недоцільно. Однак дана перестановка може виявитися актуальною повторно, якщо після її розгляду хоча б одна інша перестановка виявиться ефективною і буде виконана. При цьому можна зробити евристичне припущення, що перестановки, які були здійснені пізніше, є оптимізуючими з більшою ймовірністю, ніж ті, що були здійснені нещодавно. Це припущення базується на тому факті, що чим більше перестановок було здійснено після деякої конкретної перестановки, тим більше добутків відстаней на відповідні потоки пари об'єктів цієї перестановки могло потенційно зменшитись за абсолютним значенням. Виходячи з цього, розроблено такий алгоритм знаходження локального оптимуму.

1. Створити список допустимих перестановок та заповнити його випадковою послідовністю всіх можливих перестановок пар індексів масиву розміру N.
2. Створити порожній список заборонених перестановок.
3. Якщо список допустимих перестановок порожній, то закінчити пошук.
4. Із ймовірністю, пропорційною номеру перестановки, нумеруючи з кінця списку допустимих, вибрати із нього випадкову перестановку пар індексів.
5. Якщо вибрана перестановка є оптимізуючою, то виконати її й додати список заборонених перестановок в кінець списку допустимих. Список заборонених перестановок після цього встановлюється порожнім.
6. Додати вибрану на 4-му кроці перестановку в кінець списку заборонених.

В запропонованому алгоритмі введені поняття «інтенсивність» та матриця інтенсивностей M, елементи M_{ij} якої відповідають бажаності призначення j-го об'єкта i-му місцеположенню. Використовуючи запропоновану матрицю інтенсивностей, на кожній основній ітерації алгоритму виконується конструювання розв'язків, які потім і приводяться до локальних оптимумів за зазначеним вище алгоритмом. У випадку, якщо внаслідок конструювання отримується локальний оптимум, очевидним є факт передчасної збіжності алгоритму, що потребує виконання диверсифікації.

Конструювання нового розв'язку виконується за таким алгоритмом.

1. Із списку місцеположень, яким ще не призначені об'єкти, обрати випадковим чином із однаковою ймовірністю довільне місцеположення i.
2. За допомогою колеса рулетки серед ще непризначених об'єктів вибрати для даного місцеположення об'єкт j із ймовірністю, що є пропорційною відповідному значенню елемента M_{ij} із матриці інтенсивностей M.
3. Якщо ще залишились непризначені об'єкти (або місцеположення), то перейти до кроку 1, інакше – завершити конструювання розв'язку.

Якщо зробити евристичне припущення, що кількість можливих локальних оптимумів на порядок менша, ніж кількість всіх можливих розв'язків ($N!$), то для запобігання передчасної збіжності алгоритму слід уникати повторного впливу одного й того ж локального оптимуму на матрицю інтенсивностей. Тобто доцільною є результуюча матриця, елементи якої формуються як комбінація або перетин різних локальних оптимумів. У зв'язку з цим необхідно вести список локальних оптимумів і виконувати в ньому пошук при спробі додавання нового. З цією метою для підвищення швидкодії доцільнішим є зберігання не самих векторів, а лише їх хеш-кодів.

Узагальнений остаточний алгоритм передбачає таку послідовність дій.

1. Ініціалізувати всю матицю інтенсивностей довільним додатним числом та створити порожній список хеш-кодів потенційних розв'язків задачі.
2. Сконструювати новий розв'язок задачі за допомогою колеса рулетки.
3. Якщо хеш-код сконструйованого розв'язку міститься в списку хеш-кодів, виконати диверсифікацію, скинувши в початкові відповідні значення матриці інтенсивностей для даного розв'язку та перейти до кроку 3.
4. Оптимізувати сконструйований розв'язок до локального оптимуму.
5. Якщо хеш-код отриманого на 3-му кроці локального оптимуму вже міститься в списку хеш-кодів, перейти до кроку 3.
6. Додати хеш-код отриманого локального оптимуму до списку хеш-кодів.
7. Помножити відповідні значення елементів M_{ij} матриці інтенсивностей M для призначень $i \rightarrow j$ отриманого оптимуму на параметр-множник q.
8. Якщо значення цільової функції для даного локального оптимуму є меншим, ніж для оптимального відомого, запам'ятати поточний.
9. Якщо виконується умова критерію зупинки, то завершити алгоритм і повернути відомий оптимальний розв'язок, інакше – перейти до кроку 3.

Із формулювання алгоритму видно, що він має лише один вхідний параметр – множник q, на який домножуються інтенсивності матриці. Було встановлено, що геометрична прогресія із значенням множника $q=1,1$ сприяє ефективнішій збіжності алгоритму. Очевидно, що початкові значення елементів матриці інтенсивностей при такому підході не мають значення.

Порівняння ефективності роботи алгоритму

Результати роботи запропонованого алгоритму було приведено до порівняння із результатами авторської реалізації алгоритму Fast Ant System (FANT) Еріка Таілларда [2]. Як вхідні дані було використано матриці розмірністю 20, заповнені випадковими числами у діапазоні від 1 до 99 включно. Кількість ітерацій для кожної реалізації алгоритму була підібрана таким чином, щоб забезпечити близький середній час виконання обох алгоритмів. У таблиці 1 приведені усереднені за 1000 запусків результати нерозпаралеленої роботи обох алгоритмів на ЕОМ однакової конфігурції.

Таблиця 1

Порівняння результатів роботи FANT та розробленого алгоритму

	Час виконання	Кількість ітерацій	Значення цільової функції
Fast Ant System (FANT)	581 мс	2065	863995
Розроблений алгоритм	574 мс	1000	863226

Висновки

Кількісні результати порівняння доводять, що запропонований алгоритм є ефективним та дозволяє отримати якісніші розв’язки за той самий час порівняно із алгоритмом FANT при довільних вхідних даних. В подальшому планується реалізація пошуку локальних оптимумів в околі перестановок більшої, ніж одна пара, кількості об’єктів заданого розв’язку. Перспективною також є розробка складнішого механізму диверсифікації при повторенні локальних оптимумів, а також впровадження інтенсифікації.

Література

1. The Quadratic Assignment Problem : Theory and Algorithms / E. Cela // Combinatorial Optimization. — 1998. —Vol. 1. — 304 p.
2. FANT : Fast Ant System / E. D. Taillard // Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale. — 1998. — (Technical Report)