УДК 004.02

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

Національний технічний університет України «Київський політехнічний інститут»

ТАБУ-ПОШУК ДЛЯ КВАДРАТИЧНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

Abstract

Yuri Zorin, Assoc. Prof., PhD; Sergey Podolsky, master student
Tabu search for the quadratic assignment problem

The paper presents a new fast tabu search approach for the quadratic assignment problem. The common method specifications and relevant core techniques are formulated along with a new move domain on solution candidates, tabu list exploitation, and innovative mathematical breakthrough to decrease an algorithmic complexity of some computations. The results of performance comparison with existing metaheuristic algorithms such as Robust Tabu Search have been presented.

Вступ

Квадратична задача про призначення вперше була сформульована Копмансом та Бекманом у 1957 р. і по сьогодні залишається однією з найбільш складних задач комбінаторної оптимізації [1]. Задача моделює наступну проблему з реального життя: дано множину з N об'єктів та множину з N місцеположень. Для кожної пари місцеположень задана відстань і для кожної пари об’єктів задана вага або потік між ними (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 – F (об’єкти) та L (місцеположення) рівних потужностей, на яких визначена функція ваги (потік) $f : F \times F \to R$ та функція відстані $d : L \times L \to R$. Необхідно знайти таке відображення $\pi : L \to F$ (призначення), що мінімізує цільову функцію

$$\sum_{a,b\in L} d(a,b)\cdot f\big(\pi(a),\pi(b)\big)$$

Задача є NP-складною. Наразі не існує точного алгоритму розв’язання задач розмірності N > 20. Алгоритмічна складність задачі становить $N^2N!$.

Постановка задачі

Метою роботи є розробка метаевристичного методу для розв'язання квадратичної задачі про призначення загального виду, ефективнішого за швидкодією та якістю розв'язків, ніж аналогічні відомі методи.

Загальні положення, опис та формулювання алгоритму

Запропонований алгоритм є ітеративним. Тривалість пошуку визначається критерієм зупинки: кількістю ітерацій, часом роботи, пороговим прийнятним значенням цільової функції (вартості розв'язку задачі) тощо.

Розроблений алгоритм складається з етапів ініціалізації та основної ітерації алгоритму. На етапі ініціалізації виконуються такі дії.

1. На початку задаються вхідні дані алгоритму: розмірність задачі N, а також матриці відстаней та потоків D і F відповідно розміру $N \times N$.
2. Випадковим чином конструюється початковий допустимий вектор-розв'язок π задачі. Обчислюється вартість C_π отриманого розв'язку.
3. Ініціалізуються вектор $\pi^* \coloneqq \pi$ та змінна $C_{\pi^*} \coloneqq C_\pi$, які відповідають найкращому відомому знайденому розв'язку та його вартості відповідно.
4. Для кожної можливої перестановки у векторі-розв'язку π двох елементів з індексами i та j, $i \neq j$, обчислюється величина Δ_{ij}, на яку змінюється значення вартості даного розв'язку внаслідок виконання цієї перестановки, за формулою
$$\Delta_{ij} \coloneqq (d_{ii} - d_{jj})\left(f_{\pi_j\pi_j} - f_{\pi_i\pi_i}\right) + (d_{ij} - d_{ji})\left(f_{\pi_j\pi_i} - f_{\pi_i\pi_j}\right) +$$
$$+ \sum_{k \neq i,j} \left[(d_{ki} - d_{kj})\left(f_{\pi_k\pi_j} - f_{\pi_k\pi_i}\right) + (d_{ik} - d_{jk})\left(f_{\pi_j\pi_k} - f_{\pi_i\pi_k}\right)\right]$$
Загальна кількість таких перестановок елементів i та j становить $\frac{N(N-1)}{2}$.
Після етапу ініціалізації переходимо до пунктів основної ітерації пошуку.
5. Якщо серед усіх перестановок елементів i та j у векторі-розв'язку π наявні перестановки, які зменшують найменшу знайдену вартість C_{π^*} розв'язку задачі, то серед них обрати таку перестановку i та j, яка має мінімальне значення Δ_{ij}, тобто якомога більше зменшує вартість C_{π^*} розв'язку.
6. Якщо в попередньому пункті не була знайдена жодна перестановка відповідно до зазначених вимог, то серед усіх перестановок елементів i та j, які не виконувались довше, ніж протягом $2N^2$ останніх ітерацій, обирається перестановка із мінімальним значенням Δ_{ij}.

7. Якщо не було знайдено жодної перестановки, яка не виконувалась довше, ніж протягом $2N^2$ ітерацій, то серед усіх незаборонених на даній ітерації перестановок i та j обирається перестановка з мінімальним значенням Δ_{ij}.
8. Здійснити обрану перестановку елементів з індексами i та j у векторі π.
9. Оновити значення вартості C_π нового, модифікованого внаслідок перестановки, розв'язку π за формулою $C_\pi \coloneqq C_\pi + \Delta_{ij}$.
10. Якщо $C_\pi < C_{\pi^*}$, оновити найкращий відомий розв'язок та його вартість:
$$C_{\pi^*} \coloneqq C_\pi, \qquad \pi^* \coloneqq \pi$$
11. Заборонити зворотну перестановку i та j протягом наступних n ітерацій, де n – псевдовипадкове натуральне число, $1 \leq n \leq N$.
12. Для кожної можливої перестановки елементів p та q, $\{p, q\} \cap \{i, j\} = \{\emptyset\}$, $p \neq q$, скоректувати значення Δ_{pq} за формулою
$$\Delta_{pq} \coloneqq \Delta_{pq} + \left(d_{pi} - d_{pj} + d_{qj} - d_{qi}\right)\left(f_{\pi_p\pi_j} - f_{\pi_p\pi_i} + f_{\pi_q\pi_i} - f_{\pi_q\pi_j}\right) +$$
$$+ \left(d_{ip} - d_{jp} + d_{jq} - d_{iq}\right)\left(f_{\pi_j\pi_p} - f_{\pi_i\pi_p} + f_{\pi_i\pi_q} - f_{\pi_j\pi_q}\right)$$
13. Для кожної можливої перестановки k та i, $k \neq i$, $k \neq j$, обчислити нове значення Δ^*_{ik} за формулою із пункту 4 етапу ініціалізації.
14. Для кожної можливої перестановки у векторі-розв'язку елементів з індексами k та j, $k \neq i$, $k \neq j$, скорегувати значення Δ_{jk} за формулою
$$\Delta_{jk} \coloneqq \Delta_{jk} + \Delta_{ik} - \Delta_{ij} - \Delta^*_{ik} -$$
$$-\left(d_{ij} - d_{ik} - d_{ji} + d_{jk} + d_{ki} - d_{kj}\right)\left(f_{ij} - f_{ik} - f_{ji} + f_{jk} + f_{ki} - f_{kj}\right).$$
15. Оновити старе значення Δ_{ik} новим значенням Δ^*_{ik}: $\Delta_{ik} \coloneqq \Delta^*_{ik}$.
16. Скоректувати значення Δ_{ij} за формулою $\Delta_{ij} \coloneqq -\Delta_{ij}$.
17. Перевірити, чи виконується критерій зупинки. Якщо так, то повернути знайдений квазіоптимальний розв'язок π^* та його вартість C_{π^*}. В противному випадку перейти до пункту 5 етапу основної ітерації.

Окрім нового околу пошуку, який складається саме з перестановок елементів з індексами i та j, а не призначень кожному місцеположенню об'єкта, однією з найбільш важливих та вагомих змін з точки зору швидкодії табу-пошуку є застосування у пункті 14 основної ітерації алгоритму принципово нової формули алгоритмічної складності $O(1)$ для обчислення нового значення Δ_{jk} за відомими значеннями $\Delta_{jk}, \Delta_{ik}, \Delta_{ij}, \Delta^*_{ik}$, на відміну від формули з пункту 4 складності $O(N)$, яку досі застосовували у всіх відомих реалізаціях табу-пошуку для обчислення обох нових значень Δ_{ik} та Δ_{jk}.

Порівняння ефективності роботи алгоритму

Для оцінки ефективності алгоритму було проведено порівняння його швидкодії з найбільш відомими та ефективними метаевристичними алгоритмами Fast Ant System (FANT), Simulating Annealing (SA) та Robust Tabu Search (Ro-TS) Еріка Таілларда для розв'язання квадратичних задач про призначення загального вигляду. Методика оцінювання швидкодії передбачає заміри середнього часу знаходження глобального оптимуму низки стандартних відомих та вже досліджених квадратичних задач про призначення, загальнодоступних на сторінці "Problem Instances and Solutions" порталу QAPLIB, присвяченого дослідженням задачі [2].

Таблиця

Порівняння середнього часу знаходження глобального оптимуму

QAP Instance	Середній час пошуку			
	FANT	SA	Ro-TS	Даний табу-пошук
Tai20a	32238 мс	3283 мс	767 мс	419 мс
Tai20b	56 мс	3565 мс	46 мс	23 мс
Chr25a	2534 мс	15378 мс	16014 мс	13172 мс
Sko42	–	–	14769 мс	1507 мс

З результатів випробувань, вказаних у таблиці, видно, що розроблений нерозпаралелений табу-пошук має вищу швидкодію стосовно всіх вхідних задач, поступаючись часом лише алгоритму FANT при пошуку глобального оптимуму задачі Chr25a. Проте, в той же час, алгоритмам FANT та SA не вдалося знайти глобальний оптимум задачі Sko42 за прийнятний час, тоді як розроблений табу-пошук успішно впорався з даною задачею.

Висновки

Кількісні результати доводять практичну ефективність застосування розробленого табу-пошуку. При цьому значна частка підвищення швидкодії досягнута завдяки використанню виведеної математично формули складності $O(1)$ для виконання половини повних обчислень нових значень Δ_{ij}.

Література

1. The Quadratic Assignment Problem : Theory and Algorithms / E. Çela // Combinatorial Optimization. — 1998. —Vol. 1. — 304 p.

2. QAPLIB [Електронний ресурс] / R.E. Burkard, E. Çela, S.E. Karisch, F. Rendl. — Режим доступу : http://www.seas.upenn.edu/qaplib/.