СИСТЕМНИЙ АНАЛІЗ

І.В. СЕРГІЄНКО, В.П. ШИЛО, С.В. ЧУПОВ, П.В. ШИЛО

УДК 519.854

ПРО РОЗВ'ЯЗАННЯ КВАДРАТИЧНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

Анотація. Запропоновано дві модифікації повторюваного ітерованого алгоритму табу розв'язання квадратичної задачі про призначення (з технологією виділення ядра і без неї). Проведено дослідження цих модифікацій порівняно з кращими сучасними алгоритмами розв'язання цієї задачі. Показано ефективність розроблених алгоритмів, зокрема, для задач великої розмірності, для яких з їхньою допомогою знайдено нові рекорди.

Ключові слова: квадратична задача про призначення, повторюваний ітерований табу-пошук, технологія виділення ядра розв'язку, випадкове збурення компонент розв'язку, ефективність алгоритмів.

ВСТУП

Квадратична задача про призначення (Quadratic Assigning Problem, QAP) є важливою у практичному сенсі. До неї зводяться задачі розміщення логічних модулів на мікросхемі, розподіл медичних служб у великому госпітальному центрі, балансування турбін, аналізу зображень та ін. Розв'язання QAP потребує здійснення значного обсягу обчислень, зокрема одне значення її цільової функції знаходиться з трудомісткістю $O(n^2)$. Причому обсяг обчислень експоненціально зростає зі зростанням розмірності задачі. Практичне значення та обчислювальна складність цієї задачі підтверджується хронологією дослідження. Задачу було сформульовано у 1957 р. [1]. У 1961 р. запропоновано набір із трьох задач розмірністю 36 [2]. У 1968 р. представлено тестові задачі, розмірність яких коливається від 12 до 30 [3]. У 1976 р. було доведено [4], що ця задача належить класу NP-складних задач. У роботі [5] в 1984 р. було відзначено, що не можна побудувати точних методів для ефективного розв'язання QAP розмірністю, більшою за 20. І тільки з кінця 80-х років минулого сторіччя розпочався бурхливий розвиток наближених методів знаходження якісних розв'язків квадратичної задачі про призначення. Це пов'язано, зокрема, зі стрімким розвитком технічних можливостей обчислювальної техніки. Водночас були побудовані тестові задачі, що мають розмірність, значно більшу ніж розмірність наявних задач. Нині максимальна розмірність тестових QAP у бібліотеці $OR-Lib$ [6] становить 150.

Сьогодні більшість ефективних наближених алгоритмів для QAP ґрунтується на використанні різноманітних схем методу табу-пошуку (Tabu Search, TS), запропонованого у 1989–1990 рр. [7, 8]. Варто згадати такі версії методу, як надійний табу-пошук [9] (Robust Tabu Search, RoTS), реактивний табу-пошук [10] (Reactive Tabu Search, ReTS), ітерований табу-пошук [11] (Iterated Tabu Search, ITS). Протягом останніх років найбільш якісні результати отримано такими наближеними алгоритмами як Breakout Local Search (BLS) [12], Memetic Algorithm for QAP (BMA) [13], Improved Hybrid Genetic Algorithm (IHGA) [14], ITS for QAP [15] та багатьма іншими.

У цій статті набули подальшого розвитку результати розв’язання QAP із [16]. Розроблено дві модифікації повторюваного ітерованого алгоритму табу (з технологією виділення ядра розв’язку [17] та випадковим збуренням певної кількості його компонент). Проведене порівняльне експериментальне дослідження кращих відомих для цієї задачі та розроблених алгоритмів підтвердило ефективність останніх.

ПОСТАНОВКА ЗАДАЧІ

Квадратична задача про призначення полягає в оптимальному розміщенні n об’єктів на n позиціях або локаціях. Більш точно її формулюють у такий спосіб. Нехай $A=[a_{ij}]$ та $B=[b_{ij}]$ — дві квадратні матриці порядку n з невід’ємними елементами. Матриця A задає величини потоків, які циркулюють між об’єктами, та називається матрицею потоків (flow matrix), а матриця B визначає відстані між усіма об’єктами і має назву матриці відстаней (distance matrix). Потрібно у такий спосіб розподілити об’єкти по локаціях, щоб сума відстаней, помножених на відповідні потоки, була мінімальною.

Нехай π_i — номер локації об’єкта i, а Π_n — множина усіх перестановок $(1, \ldots, n)$.

Математична модель цієї задачі має такий вигляд: знайти

$$\min_{\pi \in \Pi_n} \left\{ f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j} \right\}.$$

Зазначимо, що в одних джерелах A — матриця відстаней, а B — матриця потоків (наприклад [9]), а в інших — навпаки (наприклад [13]). Це стосується і тестових задач з бібліотеки $OR-Lib$ [6]. Наприклад, у тестовій задачі $tai100a$ перша за порядком матриця — матриця відстаней, друга — матриця потоків, а в задачі $tai80a$ — все навпаки. Це пояснюється справедливістю співвідношення

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\overline{\pi}_i \overline{\pi}_j} b_{ij} \text{ для QAP,}$$

де $\overline{\pi} = (\pi_n, \ldots, \pi_1)$ — інверсійна до π перестановка.

ПОВТОРЮВАНИЙ ІТЕРОВАНИЙ ТАБУ-ПОШУК (RITSR, RITSK)

Авторами розроблено та експериментально досліджено дві модифікації наближеного алгоритму розв’язання QAP, названого RITS (Repeated Iterated Tabu Search) [16]: алгоритми RITSR та RITSK. На рис. 1 представлено формальну схему одного кроку цих алгоритмів.

Зазвичай схеми сучасних наближених методів розв’язання задач оптимізації містять дві фази: інтенсифікація пошуку та диверсифікація пошуку. На першій фазі здійснюється пошук локального екстремуму задачі, а на другій — робиться спроба вийти з його околу та перейти в іншу, ще не досліджену, область множини допустимих розв’язків задачі. У поданій на рис. 1 блок-схемі алгоритмів RITSR і RITSK першу фазу пошуку представляють блоки $\boxed{f \leftarrow detailed\_ITS\_search(\pi)}$ та $\boxed{f \leftarrow investigative\_ITS\_search(\pi)}$. Обидва ці блоки реалізують однакову схему ітерованого локального табу-пошуку (ITSL). Відрізняються вони лише параметрами пошуку. Під час детального пошуку локального мінімуму кількість ітерацій становить $9n^2$, а під час дослідницького — $3n^2$. Дослідницький пошук

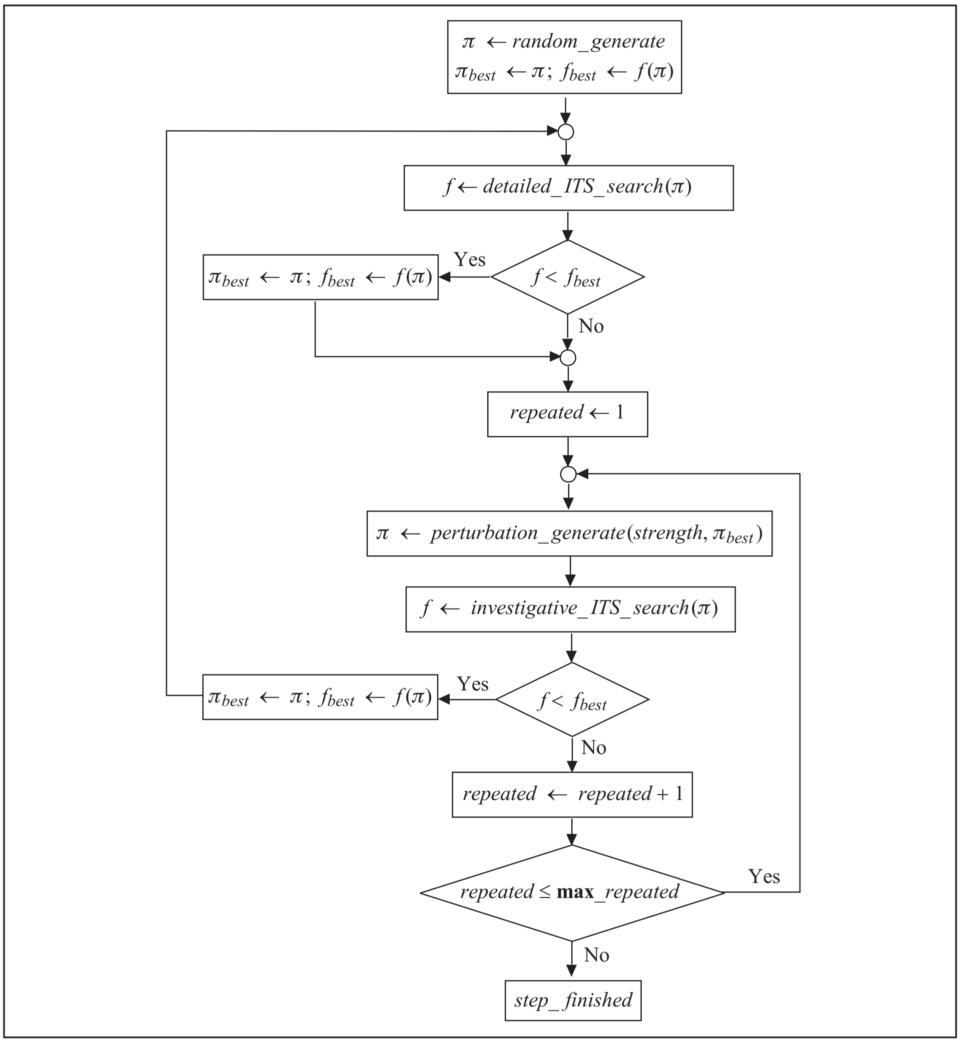

Рис 1. Схема одного кроку алгоритмів RITSR та RITSK

призначений для виявлення кращих розв'язків, що містяться на незначній відстані від межі поточного околу локального мінімуму. Процедура дослідницького пошуку повторюється певну кількість разів, що визначається параметром алгоритму $\mathbf{max}\_\ repeated$. Якщо отримано кращий розв'язок, він запам'ятовується та починається детальний пошук нового, кращого за знайдений, розв'язку в околі знайденої перестановки. Після цього цикл повторень у режимі дослідницького пошуку розпочинається заново. Друга фаза пошуку — фаза диверсифікації — виконується щоразу перед початком нового дослідницького пошуку у циклі повторень. На рис. 1 вона представлена блоком $\boxed{\pi \leftarrow perturbation\_\,generate(strength, \pi_{best})}$.

У цій фазі збурюється поточний локальний мінімум у такий спосіб, щоб наступний дослідницький пошук відбувався на певній відстані від околу поточного розв'язку. Величина такого збурення задається параметром *strength*. Цей параметр визначає кількість елементів у перестановці, яка буде збурюватись. У процесі експериментальних досліджень з'ясувалося, що найбільш придатне значення величини збурення становить 33% від розмірності n задачі. Але значення *strength* не завжди вибирається наперед і є фіксованим. У процесі досліджень використано схеми, згідно з якими значення *strength* або циклічно змінювалося від певної мінімальної до певної максимальної величини, або змінювалося навпаки, або послідовно набувало значень з наперед заданого масиву.

У цій статті досліджено два варіанти збурення: випадкове збурення (алгоритм RITSR) та збурення на основі виділення ядра розв’язку (алгоритм RITSK). Випадкове збурення полягає у перестановці *strength* елементів π_{best}, що знаходяться на випадково вибраних позиціях. У результаті експериментальних досліджень з’ясувалося, що структури якісних розв’язків конкретної задачі оптимізації схожі між собою. Стосовно QAP це означає, що у різних перестановках, для яких значення цільової функції близькі до оптимального (якісні розв’язки), на певних позиціях знаходяться елементи з однаковими значеннями. Збурення за технологією виділення ядра [17] полягає у визначенні тих елементів перестановки π_{best}, для яких воно найбільш імовірно має перспективи знаходження кращих розв’язків. Знайдені елементи впорядковуються за спаданням таких ймовірностей, після чого останні *strength* елементів випадковим чином збурюються. Більш детальний опис алгоритмів RITSR та RITSK буде представлено у наступній статті.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ РОЗРАХУНКІВ

Під час експериментального дослідження ефективності алгоритмів RITSR та RITSK основну увагу було приділено розв’язанню тестових задач типу *taiXXa* [9], де $XX = 50$, 60, 80, 100 — розмірність задачі, оскільки вони є найбільш важкими з погляду обсягу обчислень. У роботі [9] представлено алгоритм для генерування задач типу *taiXXa* довільної розмірності. На його основі розроблено процедуру побудови задач такого типу.

У тестових задачах *taiXXa* матриці відстаней та потоків є симетричними. Їхні елементи — цілі випадкові числа, рівномірно розподілені на проміжку [0, 99] (*uniform_distribution(0,99)*). Особливістю цих задач є наявність великої кількості локальних мінімумів різної якості, більшість з яких не задовольняють користувача. Якщо ступінь збурення поточного розв’язку на фазі диверсифікації вибирати невеликим (меншим за 30%), то отримаємо довготривалий процес переходу від одного локального мінімуму до іншого, який може бути як хорошим, так і неякісним. У більшості випадків розв’язок є неякісним. В іншому випадку (для ступеня збурення, що перевищує 40%), певні області можуть бути не досліджені, а отже, якісний розв’язок задачі може бути пропущений. На рис. 2 представлено наближене розміщення розв’язків задач *tai*100*a*. Надзвичайно висока щільність розташування локальних мінімумів різної якості на всій множині перестановок робить задачі цього типу дуже складними з погляду отримання хороших розв’язків за прийнятний час.

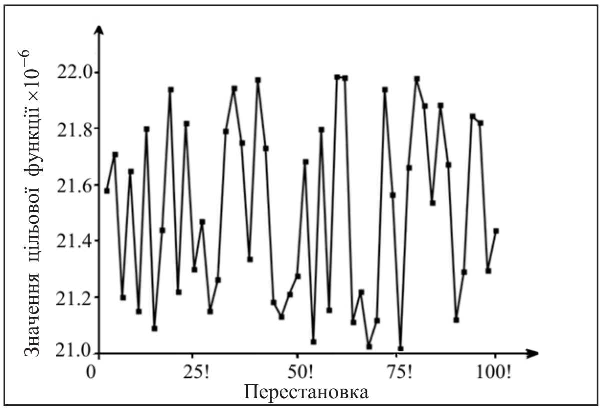

Рис. 2. Схематичне розміщення розв’язків задач *tai*100*a*

Усі зазначені алгоритми реалізовано на мові C++ у середовищі Microsoft Visual Studio 2017. Обчислювальні експерименти проведено з використанням PC Intel® Core™ i7-3820 CPU 3.5 GHz і 8.0 GB оперативної пам’яті. Для порівняння ефективності алгоритмів RITSK та RITSR з іншими алгоритмами було вибрано кращі сучасні алгоритми ITS [11,15], BLS [12], BMA [13]. Тексти програм алгоритмів BLS та BMA взято із загальнодоступних джерел [http://www.info. univ-angers.fr/~hao/ BLS_code.html; http://www.info.univ-angers.fr/pub/ hao/BMA.html]).

Автори висловлюють велику вдячність А. Місевічусу (A. Misevicius) за надані тексти програм для алгоритму ITS. Це дало змогу провести експериментальні дослідження в однакових умовах та більш об'єктивно здійснити порівняння ефективності алгоритмів під час розв'язання одних і тих самих задач.

У табл. 1 наведено деякі результати порівняльного дослідження цих алгоритмів з алгоритмами RITSK (з виділенням ядра) та RITSR (з випадковим збуренням). У другому стовпці таблиці BKS — відомий рекорд. Для кожного алгоритму наведено відхилення (у відсотках) від BKS середнього значення цільової функції, отриманого за допомогою цього алгоритму після виконання 40 спроб розв'язання задачі. Напівжирним шрифтом виділено найкращі результати. Для усіх алгоритмів обмеження за часом становило 2 години.

Для задачі *tai*100*a* алгоритмом RITSK знайдено кращий за відомий на цей момент розв'язок
π_{best} = (89, 63, 41, 47, 84, 25, 91, 21, 8, 14, 0, 58, 32, 26, 75, 19, 86, 68, 64, 48, 49, 57, 42, 80, 88, 13, 78, 45, 24, 5, 97, 59, 82, 54, 38, 67, 31, 43, 6, 3, 23, 18, 74, 90, 28, 16, 36, 34, 12, 4, 1, 40, 65, 10, 44, 2, 96, 30, 85, 35, 7, 79, 69, 11, 83, 56, 50, 15, 37, 76, 22, 29, 81, 60, 95, 73, 53, 93, 70, 66, 51, 55, 20, 9, 17, 92, 87, 33, 62, 72, 99, 71, 46, 39, 61, 52, 27, 98, 94, 77) із значенням цільової функції $f_{best} = 21\,044\,752$.

З теоретичної точки зору наведений вище розв'язок π_{best} не є перестановкою, оскільки відлік позицій у ньому починається з нуля. Такий спосіб представлення перестановки є дуже зручним з обчислювальної точки зору. Проте, якщо до кожного значення π_{best} додати одиницю, то отримаємо класичну перестановку.

Таблиця 1

Задача	BKS	RITSK	RITSR	ITS	BLS	BMA
tai50a	4938796	0.050	**0.048**	0.050	0.102	0.090
tai60a	7205962	0.144	0.154	**0.135**	0.295	0.193
tai80a	13499184	0.350	**0.342**	0.368	0.564	0.458
tai100a	21044752	**0.277**	0.284	0.311	0.569	0.432

ВИСНОВКИ

Отримані результати експериментальних досліджень свідчать, що алгоритми RITSK та RITSR здатні конкурувати з найкращими алгоритмами розв'язання квадратичної задачі про призначення. Подальше використання технологій паралельних обчислень, зокрема портфелів та команд алгоритмів, дасть змогу пришвидшити процес відшукання якісних розв'язків QAP.

СПИСОК ЛІТЕРАТУРИ

1. Koopmans T.C., Beckmann M.J. Assignment problems and the location of economic activities. *Econometrica*. 1957. Vol. 25, N 1. P. 53–76.
2. Steinberg L. The backboard wiring problem: a placement algorithm. *SIAM Review*. 1961. Vol. 3, N 1. P. 37–50.
3. Nugent Ch.E., Vollmann Th.E., Ruml J. An experimental comparison of techniques for the assignment of facilities to locations. *Oper. Res*. 1968. Vol. 16. P. 150–173.
4. Sahni S., Gonzalez T. P-complete approximation problems. *J. of the ACM*. 1976. Vol. 23, N 3. P. 555–565.
5. Burkard R.E. Quadratic assignment problems. *European J. of Oper. Res*. 1984. Vol. 15, N 3. P. 283–289.
6. Beasley J.E. OR-library; distributing test problems by electronic mail. *J. of Oper. Res. Society*. 1990. Vol. 41, N 11. P.1069–1072.
7. Glover F. Tabu Search — Part I. *ORSA J. on Computing*. 1989. Vol. 1, N 3. P. 190–206.
8. Glover F. Tabu Search — Part II. *ORSA J. on Computing*. 1990. Vol. 2, N 1. P. 4–32.

9. Taillard E. Robust taboo search for the quadratic assignment problem. *Parallel Computing*. 1991. Vol. 17, Iss. 4-5. P. 443–455.
10. Battiti R., Tecchiolli G. The reactive tabu search. *ORSA J. on Computing*. 1994. Vol. 6, N 2. P. 126–140.
11. Misevicius A., Lenkevicius A., Rubliauskas D. Iterated tabu search: An improvement to standard tabu search. *Information Technology and Control*. 2006. Vol. 35, N 3. P. 187–197.
12. Benlic U., Hao J.K. Breakout local search for the quadratic assignment problem. *Applied Math. And Computation*. 2013. Vol. 219, N 9. P. 4800–4815.
13. Benlic U., Hao J.K. Memetic search for the quadratic assignment problem. *Expert Systems with Applications*. 2015. Vol. 42, N 1. P. 584–595.
14. Misevicius A. An improved hybrid genetic algorithm: New results for the quadratic assignment problem. *Knowledge Based Systems*. 2004. Vol. 17, N 2–4. P. 65–73.
15. Misevicius A. An implementation of the iterated tabu search algorithm for the quadratic assignment problem. *OR Spectrum*. 2012. Vol. 34, N 3. P. 665–690.
16. Шило П.В. Повторяемый итерированный алгоритм табу для решения квадратичной задачи о назначениях. *Кибернетика и системный анализ*. 2017. Т. 53, № 2. С. 163–167.
17. Сергиенко И.В., Шило В.П. Технология ядра для решения задач дискретной оптимизации. *Кибернетика и системный анализ*. 2017. Т. 53, № 6. С. 73–83.

Надійшла до редакції 15.08.2019

И.В. Сергиенко, В.П. Шило, С.В. Чупов, П.В. Шило

О РЕШЕНИИ КВАДРАТИЧНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Аннотация. Предложены две модификации повторяемого итерированного алгоритма табу решения квадратичной задачи о назначениях (с технологией выделения ядра и без неё). Проведено исследование этих модификаций сравнительно с лучшими современными алгоритмами решения этой задачи. Показана эффективность разработанных алгоритмов, в частности, для задач большой размерности, для которых с их помощью найдены новые рекорды.

Ключевые слова: квадратичная задача о назначениях, повторяемый итерированный табу-поиск, технология выделения ядра решения, случайное возмущение компонент решения, эффективность алгоритмов.

I.V. Sergienko, V.P. Shylo, S.V. Chupov, P.V. Shylo

SOLVING THE QUADRATIC ASSIGNMENT PROBLEM

Abstract. This paper focuses on the development and improvement of the iterated repeated tabu algorithms for solving the quadratic assignment problem using core allocation technology. The comparative analysis of the empirical computational performance is presented, comparing the modern algorithms from the literature and the algorithms proposed in this paper. The results confirm the efficiency of the modified algorithms in terms of the solution quality and time especially for large-scale problems.

Keywords: quadratic assignment problem, repeated iterated tabu search, solution core allocation technology, random perturbation of solution components, algorithm efficiency.

Сергієнко Іван Васильович,
доктор фіз.-мат. наук, академік НАН України, професор, директор Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: incyb@incyb.kiev.ua.

Шило Володимир Петрович,
доктор фіз.-мат. наук, професор, провідний науковий співробітник Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: v.shylo@gmail.com.

Чупов Сергій Вікторович,
кандидат фіз.-мат. наук, доцент кафедри Державного вищого навчального закладу «Ужгородський національний університет», e-mail: serhii.chupov@uzhnu.edu.ua, sergey.chupov@gmail.com.

Шило Петро Володимирович,
кандидат фіз.-мат. наук, науковий співробітник Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: petershylo@gmail.com.