УДК 519.854
DOI https://doi.org/10.24144/2616-7700.2026.49(2).300-307

С. В. Чупов[1], О. І. Кузка[2], В. М. Дуран[3]

[1] ДВНЗ «Ужгородський національний університет»,
доцент кафедри системного аналізу та теорії оптимізації,
кандидат фізико-математичних наук, доцент
serhii.chupov@uzhnu.edu.ua
ORCID: https://orcid.org/0000-0002-7715-3924

[2] ДВНЗ «Ужгородський національний університет»,
доцент кафедри системного аналізу та теорії оптимізації,
кандидат фізико-математичних наук, доцент
oleksandr.kuzka@uzhnu.edu.ua
ORCID: https://orcid.org/0000-0002-7556-3057

[3] ДВНЗ «Ужгородський національний університет»,
аспірант кафедри системного аналізу та теорії оптимізації
valentyn.duran@uzhnu.edu.ua
ORCID: https://orcid.org/0009-0001-6409-3061

СИМУЛЯЦІЯ ВІДПАЛУ ДЛЯ КВАДРАТИЧНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ НА ОСНОВІ ЛЕКСИКОГРАФІЧНО ВПОРЯДКОВАНИХ ПЕРЕСТАНОВОК

Квадратична задача про призначення (QAP) є однією з найбільш вивчених комбінаторних задач оптимізації з різними практичними застосуваннями. У цій статті ми представляємо адаптивний алгоритм симуляції відпалу (ASAQAP) для розв'язання QAP, що використовує спеціальний запис перестановок, названий позиційним представленням. Це дозволяє здійснювати арифметичні операції над перестановками поданими у позиційному виді. ASAQAP досліджує простір пошуку шляхом використання адаптивної симуляції відпалу, та локального табу пошуку на окремих етапах розв'язання задачі. Експериментальні оцінки на множині тестових задач типу "*Taie*" розмірності 27, 45 та 75 показують, що запропонований підхід здатний досягти відомі на даний момент результати для всіх екземплярів із придатним середнім часом обчислення. Також надаються порівняння, щоб показати конкурентоспроможність запропонованого підходу відносно алгоритму CPTSQAP.

Ключові слова: Квадратична задача про призначення, евристичний алгоритм пошуку, метод симуляції відпалу, локальний алгоритм наближеного пошуку, позиційне представлення перестановок, ефективність алгоритмів.

1. Вступ. Задача квадратичного призначення (*Quadratic Assignment Problem*, QAP) є однією з найскладніших задач комбінаторної оптимізації та має широке застосування в інженерії, логістиці та проектуванні. Вона належить до класу NP-складних задач, що унеможливлює ефективне знаходження точного розв'язку для задач великої розмірності. У зв'язку з цим значна увага приділяється евристичним та метаевристичним методам, серед яких важливе місце займає метод *Adaptive Simulated Annealing* (ASA) [4].

Основні особливості ASA:

- адаптивна схема охолодження;
- індивідуальне налаштування параметрів для кожної змінної;
- використання спеціальних розподілів для генерації кроків;
- можливість ефективної роботи у задачах великої розмірності.

Найкращі результати були досягнуті гібридними евристичними методами, у яких перегляд різних областей простору пошуку (диверсифікація) здійснювався за однією схемою, а уточнення розв'язку, або локальний пошук (інтенсифікація) – за іншою. Як правило, локальний пошук у найефективніших алгоритмах відбувався за одним з варіантів табу пошуку (надійний табу, RTS [7], реактивний табу, RETS [1]). Перегляд простору пошуку також відбувається за різноманітними схемами: випадкове збурення поточного розв'язку (Ітеративний заборонений пошук (ITS) [6]), паралельний перегляд різних областей (Cooperative parallel tabu search, CPTS [5]), механізм генетичного пошуку (Наближений гібридний алгоритм для розв'язання QAP, HBMA [3]), механізм імітації відпалу (Глобальний рівноважний пошук, GES [8], повторюваний ітерований алгоритм табу, RITSR [9]).

2. Постановка задачі. Дано n об'єктів, які потрібно розташувати у n різних локаціях (місць призначень). Відомі величини потоків ресурсів a_{ij} між об'єктами i та j, де $i,\ j = 1,\ldots,n$, і відстані b_{rs} між локаціями (пунктами) r та s, де $r,\ s = 1,\ldots,n$. Потрібно знайти таке розподілення об'єктів по локаціях, щоб сума відстаней, помножена на відповідні потоки, була б мінімальною.

Отже, математичну постановку задачі QAP можна сформулювати наступним чином: знайти:

$$\min_{\pi\in\Pi^n} f(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{\pi_i\pi_j},$$

де $A = [a_{ij}]$ та $B = [b_{rs}]$ — квадратні матриці порядку n, Π^n — множина усіх перестановок розмірності n і π_i задає номер локації об'єкту i.

3. Позиційне представлення перестановок. Якщо усі перестановки однієї розмірності впорядкувати за лексикографічним зростанням, то можна помітити певну закономірність в їх розташуванні, яка дозволяє використовувати замість стандартного представлення перестановок інший вигляд [2].

Перестановку π можна записати так, що в результаті отримаємо цілочисловий вектор p, але на відміну від звичайної перестановки цей вектор буде складатися з позицій або індексів, що визначаються певним чином, яким відповідають елементи перестановки π. Схеми перетворення перестановки до позиційного виду представлено на лістингу 2.

Лістинг 2: Перетворення перестановки до позиційного виду.

```
function ToPosition(π, order:Permutation):Position;
begin
  p := $\left(\underbrace{0,0,\ldots,0}_{n}\right)$;
  for i:=0 to n-2 do begin
      $p_i$ := Index of $\pi_i$ in order;
      order := order {$\pi_i$};
  end;
  $p_{n-1}$ := 0;
  return p;
end;
```

Вектор цілих чисел $p = (p_0, p_1, \ldots, p_{n-1})$, отриманий в результаті роботи алгоритму $p := \mathit{ToPosition}(\pi,\ \mathit{order})$ називається *позиційним представленням* *перестановки* $\pi = (\pi_0, \pi_1, \dots, \pi_{n-1})$.

Як показано у [2], якщо перестановки розглядати в позиційному виді, тоді з'являється можливість проводити з ними різноманітні алгебраїчні дії так, що результуюча перестановка завжди буде займати правильне місце серед усіх перестановок, що впорядковано за базовою перестановкою *order*

4. Адаптивна симуляція відпалу для QAP. У запропонованому алгоритмі сусідні розв'язки будуються на основі позиційного представлення перестановок.

Після отримання випадкового зміщення до наступного розв'язку, яке залежить від поточної температури, воно перетворюється до цілого числа. Далі це значення множиться на позиційну одиницю $\bar{1}$ за алгоритмом *MulPositions* та додається до поточного розв'язку за алгоритмом *AddPositions* [2] (рядок 9, $\mathit{poffset} := \mathit{ConvetToPosition}(\mathit{offset})$).

Отриманий розв'язок використовується як початковий для здійснення локального пошуку покращення. Локальний пошук представляє собою варіант табу пошуку, представлений у [3].

Як і в класичному SA, використовується ймовірнісний критерій прийняття нового розв'язку: $P = e^{-\frac{\Delta Q}{T}}$, де ΔQ — зміна значення цільової функції. Це дозволяє алгоритму виходити з локальних мінімумів. Якщо новий розв'язок приймається, тоді він фіксується як поточний, рівень стагнації процесу скидається до 0. У запропонованому алгоритмі рівень стагнації це кількість кроків алгоритму після яких не було прийнято нового розв'язку. Якщо новий розв'язок не приймається, тоді рівень стагнації збільшується на одиницю. Коли рівень стагнації досягне критичної межі, тоді температура може тимчасово збільшуватися, що дозволяє уникнути застрягання в локальних мінімумах.

Як і в більшості наближених алгоритмах, критерієм зупинки роботи є наперед заданий максимальний час роботи алгоритму.

Схема роботи адаптивного методу симуляції відпалу для квадратичної задачі про призначення (ASAQAP) представлена на лістингу 3.

Лістинг 3: Формальна схема алгоритму ASAQAP.

```
1  function ASA_bounded(x^0: Permutation; T_0, T_min: double): Permutation;
2  begin
3      Position px := ToPosition(x^0);
4      x := x^0;  f := f(x);
5      x^best := x; F_best := f;
6      stagnantion = 0; k := 1;
7      while not stoppingCriteria do begin
8          T := T_0 / log(k+2); k := k+1;
9          offset := T*(Random(1.0)-0.5);
10         Position poffset := ConvetToPosition(offset);
11         Position px^new := px + poffset;
12         Permutation x^new := ToPermutation(px^new);
13         x^new := LocalSearch(x^new); f_new := f(x^new);
14         dE := f_new - f; P := E^(-dE/T);
15         if (dE < 0) or (P > Random(1.0)) then begin
16             x := x^new; px := ToPosition(x^new); f := f_new;
17             stagnantion := 0;
18         end else stagnantion := stagnantion + 1;
19         if f < F_best then begin
```

```
                F_best := f; x^best := x;
            end;
            if stagnantion > maxStagnation or T < T_min then begin
                x := Pertrube(x^best); px := ToPosition(x); f := f(x);
                stagnantion := 0; T := T_0;
            end;
    end;
    return x^best;
end;
```

5. Результати числових експериментів. Числові експерименти проводилися на тестових задачах типу *Taie* у яких назва має форму TaiXXeYY, де XX — визначає розмірність задачі, а YY — номер тестової задачі. У наступних таблицях використані такі позначення: *Problem* — назва задачі; *BKS* — відомий рекорд; #BKS — кількість розв'язаних задач з серії, що складалася з 10 спроб. Time – найменший час у хвилинах отримання найкращого розв'язку у серії. Результати розв'язання таких задач для алгоритму CPTS-QAP було взято з [5]. Максимальний час роботи алгоритму ASAQAP становив 5 хвилин.

Таблиця 1.

Задачі "***Tai27eYY***"

		CPTS-QAP		**ASAQAP**	
Problem	**BKS**	**#BKS**	**Time**	**#BKS**	**Time**
tai27e01	2558	10	6.2	10	0.02
tai27e02	2850	10	6.2	10	0.05
tai27e03	3258	10	6.1	10	0.01
tai27e04	2822	10	6.3	10	0.06
tai27e05	3074	10	6.3	10	0.04
tai27e06	2814	10	6.1	10	0.02
tai27e07	3428	10	6.4	10	0.07
tai27e08	2430	10	6.3	10	0.05
tai27e09	2902	10	6.7	10	0.07
tai27e10	2994	10	6.2	10	0.04
tai27e11	2906	10	6.2	10	0.03
tai27e12	3070	10	5.5	10	0.01
tai27e13	2966	10	6.6	10	0.06
tai27e14	3568	10	6.3	10	0.04
tai27e15	2628	10	6.3	10	0.02
tai27e16	3124	10	6.5	10	0.06
tai27e17	3840	10	6.4	10	0.03
tai27e18	2758	10	6.2	10	0.01
tai27e19	2514	10	6.4	10	0.03
tai27e20	2638	10	6.3	10	0.02

Таблиця 2.

Задачі “***Tai45eYY***”

		CPTS-QAP		ASAQAP	
Problem	**BKS**	**#BKS**	**Time**	**#BKS**	**Time**
tai45e01	6412	10	51.0	10	1.1
tai45e02	5734	10	49.2	10	1.0
tai45e03	7438	10	51.0	10	1.0
tai45e04	6698	9	51.2	10	1.0
tai45e05	7274	7	50.8	10	1.0
tai45e06	6612	10	45.3	10	0.8
tai45e07	7526	10	48.4	10	0.8
tai45e08	6554	10	46.7	10	0.8
tai45e09	6648	7	48.1	10	0.8
tai45e10	8286	7	48.7	10	0.9
tai45e11	6510	10	47.0	10	0.8
tai45e12	7510	8	51.2	10	0.0
tai45e13	6120	10	52.4	10	0.8
tai45e14	6854	10	47.0	10	0.7
tai45e15	7394	8	49.8	10	0.8
tai45e16	6520	10	47.5	10	0.7
tai45e17	8806	8	49.0	10	0.9
tai45e18	6906	10	48.5	10	0.8
tai45e19	7170	9	48.3	10	0.9
tai45e20	6510	10	43.7	10	0.7

Таблиця 3.

Задачі “***Tai75eYY***”

		CPTS-QAP		ASAQAP	
Problem	**BKS**	**#BKS**	**Time**	**#BKS**	**Time**
tai75e01	14488	0	475.2	10	2.4
tai75e02	14444	0	472.4	10	2.5
tai75e03	14154	0	482.0	10	2.1
tai75e04	13694	0	464.7	10	2.8
tai75e05	12884	1	464.1	10	2.0
tai75e06	12554	0	485.2	10 (**12534**)	2.2
tai75e07	13782	0	431.7	10	3.2
tai75e08	13948	0	425.7	10	2.1
tai75e09	12650	0	485.9	10	3.0
tai75e10	14192	0	463.1	10	1.5
tai75e11	15250	0	442.5	10	1.2
tai75e12	12760	0	469.9	10	2.4
tai75e13	13024	0	450.0	10	2.0

tai75e14	12604	0	448.1	10	2.1
tai75e15	14294	0	461.7	10	2.6
tai75e16	14204	0	463.1	10	1.9
tai75e17	13210	0	483.5	10	3.4
tai75e18	13500	0	472.5	10	2.7
tai75e19	12060	1	458.2	10	2.1
tai75e20	15260	0	479.0	10	2.9

Якщо проаналізувати отримані результати, то слід відзначити, що запропонованим алгоритмом були розв'язані усі задачі у кожній серії (у задачі tai75e06 було отримано новий рекорд), на відміну від CPTS-QAP де для задач розмірності 75 тільки у двох було досягнуто відомого рекорду і тільки по одному разу у серії. Покращити відомі рекорди не вдалося напевно тому, що вони і є глобальними оптимумами.

6. Висновки та перспективи подальших досліджень. У роботі представлений гібридний евристичний алгоритм ASAQAP. Особливістю його реалізації є модифіковані схеми фази інтенсифікації та диверсифікації. На фазі диверсифікації використовується механізм адаптивної симуляції відпалу у якій наступна область пошуку визначається на основі позиційного представлення перестановок. Фаза інтенсифікації пошуку представляє собою модифіковану схему надійного пошуку табу.

Результати експериментів порівнювалися з відомими результатами, отриманими за алгоритмом CPTS-QAP. Отримані результати свідчать про належну ефективність алгоритму, як з точки зору досягнення відомих рекордів, так і з точки зору часу роботи.

У подальших дослідженнях планується:

– Провести експерименти для тестових задач розмірності 125, 175, 343, 729 та порівняти їх з результатами отриманими за алгоритмом IHBMA.
– Розробити механізм динамічної адаптації параметрів алгоритму.

Конфлікт інтересів

Автори заявляють, що не мають конфлікту інтересів щодо даного дослідження, включаючи фінансовий, особистий, авторський або будь-який інший, який міг би вплинути на дослідження, а також на результати, представлені в даній статті.

Фінансування

Дослідження здійснено в рамках кафедральної науково-дослідної роботи «Моделі і методи системного аналізу в міждисциплінарних дослідженнях» (державний обліковий номер 0125U003246).

Доступність даних

Усі дані доступні в цифровій або графічній формі в основному тексті рукопису.

Використання штучного інтелекту

Автори підтверджують, що при створенні даної роботи вони не використовували технології штучного інтелекту.

Внесок авторів

С. В. Чупов: концептуалізація, формальний аналіз, розробка алгоритмів, програмування, аналіз, написання — оригінальний проект. О. І. Кузка: програмування, написання, аналіз, проведення числових експериментів — оригінальний проект. В. М. Дуран: проведення числових експериментів, оформлення — оригінальний проект.

Список використаної літератури

1. Battiti, R., & Tecchiolli, G. (1994). The Reactive Tabu Search. *ORSA J. on Computing,* 6, 126–140. https://doi.org/10.1287/ijoc.6.2.126
2. Chupov S. V. (2016). Lexicographically ordered permutations. *Computer mathematics, Kyiv: Institute of Cybernetics of V. M. Glushkov NAS of Ukraine,* (2), 151–161.
3. Chupov, S. V., & Fedorishko, A. V. (2026). Improved hybrid algorithm for the quadratic assignment problem. *Scientific Bulletin of Uzhhorod University. Series "Mathematics and Computer Science* 48(1), 243–253. https://doi.org/10.24144/2616-7700.2026.48(1).243-253
4. Ingber, L. (1996). Adaptive Simulated Annealing (ASA): Lessons Learned. *Control and Cybernetics,* 25(1), 33–54. https://doi.org/10.48550/arXiv.cs/0001018
5. James, T., Rego, C., & Glover, F. (2009). A cooperative parallel tabu search algorithm for the quadratic assignment problem. *European Journal of Operational Research,* 195(3), 810–826. https://doi.org/10.1016/j.ejor.2007.06.061
6. Misevicius, A. (2012). An implementation of the iterated tabu search algorithm for the quadratic assignment problem. *OR Spectrum,* 34(3), 665–690. https://doi.org/10.1007/s00291-011-0274-z
7. Taillard, E. (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing,* 17, 443–455. https://doi.org/10.1016/S0167-8191(05)80147-4
8. Sergienko, I. V., & Shylo, V. P. (2003). *Discrete optimization problems. Problems, solution methods, research.* Kyiv: Naukova Dumka.
9. Sergienko, I. V., Shylo, V. P., Chupov, S. V., & Shylo, P. V. (2020). On solving the quadratic assignment problem. *Cybernetics and Systems Analysis,* (1), 64–69. https://doi.org/10.1007/s10559-020-00219-8 [in Ukrainian].

Chupov S. V., Kuzka O. I., Duran V. M. Annealing Simulation for A Quadratic Assignment Problem Based on Lexicographically Ordered Permutations.

The quadratic assignment problem (QAP) is one of the most studied combinatorial optimization problems with various practical applications. In this paper, we present an adaptive simulated annealing algorithm (ASAQAP) for solving QAP, which uses a special notation of permutations, called the positional representation. This allows performing arithmetic operations on permutations given in positional form. ASAQAP explores the search space by using adaptive simulated annealing, and local tabu search at separate stages of solving the problem. Experimental evaluations on a set of "Taie" type test problems of dimensions 27, 45 and 75 show that the proposed approach is able to achieve the currently known results for all instances with a suitable average computing time. Comparisons are also provided to show the competitiveness of the proposed approach with respect to the CPTSQAP algorithm.

Keywords: Quadratic assignment problem, heuristic search algorithm, simulated annealing method, local approximate search algorithm, positional representation of permutations, efficiency of algorithms.

Отримано: 30.03.2026 Прийнято: 15.04.2026 Опубліковано: 30.04.2026