УДК 519.87
DOI https://doi.org/10.26661/2786-6254-2023-1-02

АЛГОРИТМ ПЕРЕМІШАНИХ СТРИБАЮЧИХ ЖАБ У ЗАДАЧІ РОЗМІЩЕННЯ ВИРОБНИЦТВА

Козін І. В.
доктор фізико-математичних наук, професор,
професор кафедри економічної кібернетики
Запорізький національний університет
вул. Жуковського, 66, Запоріжжя, Україна
orcid.org/0000-0003-1278-8520
ainc00@gmail.com

Нарзуллаєв У. Х.
кандидат фізико-математичних наук, доцент,
декан факультету телекомунікаційних технологій та професійної освіти
Самаркандська філія Ташкентського університету інформаційних технологій імені Мухаммада Аль-Хорезмі
вул. Ібн Сіно, 2, Самарканд, Узбекистан
orcid.org/0009-0006-7851-0520
ulug1956_56@mail.ru

Алломов З. К.
магістрант факультету фізики та математики
Ургенчський державний університет
вул. Х. Олімжона, 14, Ургенч, Узбекистан
allomovzafar@gmail.com

Ключові слова: *дискретна оптимізація, метаевристика, орієнтована фрагментарна структура, алгоритм стрибаючих жаб, задача розміщення виробництва.*

Задача розміщення виробництва – одна з найвідоміших масових задач дискретної оптимізації. Є безліч варіантів постановки цієї задачі. Як правило, всі ці варіанти задачі розміщення виробництва належать до класу NP-важких задач, тобто для пошуку точного розв'язку такої задачі на сьогодні невідомі алгоритми поліноміальної трудомісткості. Досі не розроблено ефективних способів розрахунку нижніх границь оцінки цільової функції для такої задачі. Точні алгоритми для цієї задачі зводяться до перебору варіантів. У зв'язку з цим використання точних алгоритмів для вирішення задачі розміщення виробництва часто виявляється недоцільним і неможливим через великі витрати часу. Тому значний інтерес становить розробка та дослідження евристичних методів оптимізації. Одним із перспективних напрямів є розробка алгоритмів, заснованих на відомих метаевристичних підходах, які з успіхом використовуються для вирішення багатьох задач дискретної оптимізації. У цій роботі показано, що один з класів задач розміщення виробництва зводиться до задачі покриття повного графа зірками, що не перетинаються у вершинах. Доведено, що в такій постановці задачу можна розглядати як задачу оптимізації на орієнтованій фрагментарній структурі. Це дозволяє створювати гібридні алгоритми відшукання субоптимальних розв'язків задач такого класу на основі комбінації відомої метаевристики та фрагментарного алгоритму. В роботі розглянута одна з таких метаевристик, а саме алгоритм перемішаних стрибаючих жаб. Показано, що цей алгоритм може бути використаний для пошуку субоптимальних

рішень задачі оптимізації на множині перестановок. З іншого боку, за наявності орієнтованої фрагментарної структури задача дискретної оптимізації може бути зведена до задачі оптимізації на множині перестановок. Таким чином, отримано простий та досить ефективний метод відшукання субоптимальних розв'язків задачі розміщення виробництва. Метод може бути легко перенесений і на інші класи задач дискретної оптимізації, які можуть розглядатися як задачі оптимізації на орієнтованій фрагментарній структурі.

THE SHUFFLE FROG LEAPING ALGORITHM FOR THE PRODUCTION LOCATION PROBLEM

Kozin I. V.
Doctor of Physical and Mathematical Sciences, Professor,
Professor at the Department of Economic Cybernetics
Zaporizhzhia National University
Zhukovskoho str., 66, Zaporizhzhia, Ukraine
orcid.org/0000-0003-1278-8520
ainc00@gmail.com

Narzullaev U. H.
Candidate of Physical and Mathematical Sciences, Associate Professor,
Dean of the Faculty of Telecommunication Technologies and Professional Education
Samarkand branch of Tashkent University of Information Technologies
named after Muhammad Al-Khorezmi
Ibn Sino str., 2, Samarkand, Uzbekistan
orcid.org/0009-0006-7851-0520
ulug1956_56@mail.ru

Allomov Z. K.
Master's Student of the Faculty of Physics and Mathematics
Urgench State University
H. Olimzhona str., 14, Urgench, Uzbekistan
allomovzafar@gmail.com

Key words: *discrete optimization, metaheuristics, oriented fragmentary structure, jumping frog algorithm, production placement problem.*

The production location problem is one of the most well-known discrete optimization problems. There are many options for setting this task. As a rule, all these variants of the problem of location of production belong to the class of NP-hard problems, that is, for its exact solution of such a problem, algorithms of polynomial complexity are currently unknown. So far, no effective methods have been developed for calculating the lower bounds for this problem, which make it possible to assess the achievement of the optimum. Exact algorithms for this problem are reduced to a complete enumeration of options. In this regard, the use of exact algorithms for solving the problem of production location often turns out to be inappropriate and impossible due to the large time costs. Therefore, the development and study of heuristic optimization methods is of considerable interest. One of the promising areas is the development of algorithms based on well-known metaheuristic approaches that are successfully used to solve many discrete optimization problems.
In this paper, we show that one of the classes of production location problems is reduced to the problem of covering a complete graph with vertex-disjoint stars. It is proved that in this formulation the problem can be considered as an

optimization problem on an oriented fragmentary structure. This allows you to create hybrid algorithms for finding suboptimal solutions to problems of this class based on a combination of well-known metaheuristics and a fragmentary algorithm. The mixed jumping frogs algorithm was chosen as a metaheuristic. It is shown that this algorithm can be used to find suboptimal solutions on a set of permutations. On the other hand, in the presence of an oriented fragmentary structure, the discrete optimization problem can be reduced to an optimization problem on a set of permutations. Thus, a simple and rather effective method for finding suboptimal solutions to the production location problem has been obtained. The method can be easily transferred to other classes of discrete optimization problems, which can be considered as problems on an oriented fragmentary structure.

Вступ

Більшість класів задач дискретної оптимізації належать до NP-важких задач [1]. Для цих задач невідомі точні алгоритми пошуку оптимального розв'язку, складність яких обмежена поліном від довжини умови задачі. Зокрема, до таких класів належать різноманітні варіанти задачі розміщення виробництва [2–5]. Задачі такого типу мають як теоретичний інтерес, так і практичну спрямованість. Тому необхідні методи, які дозволяють отримати хоча б субоптимальні (наближені до оптимального в будь-якому сенсі) пошуки розв'язків цих задач. Одним з інструментів пошуку субоптимальних розв'язків подібних задач за прийнятний час є метаевристики. Відносна простота метаевристик [6–8], обмежений обсяг пам'яті і висока швидкість алгоритмів роблять їх незамінними для вирішення прикладних задач, в яких потрібно отримати хоча б допустиме рішення, більш-менш відповідальне вимогою замовника. Метаевристики не мають теоретичного обґрунтування. Єдиним підтвердженням їхньої якості є практика, тобто результати, одержані на тестових прикладах, факти вирішення реальних прикладних задач тощо. Але в задачах практики зазвичай є досить багато обмежень, які не дозволяють створити універсальний метод розв'язку для великої кількості прикладних задач. У цій роботі пропонується підхід до дискретних задач із використанням орієнтованих фрагментарних структур. Такий підхід, хоч і не є універсальним, дозволяє просто будувати гібридні алгоритми на основі наявних метаевристик для великого класу дискретних оптимізаційних задач. У роботі пропонується гібридний алгоритм на основі комбінації фрагментарного алгоритму та алгоритму перемішаних стрибаючих жаб для одного класу задач розміщення виробництва.

Огляд наявної літератури

Навіть у найпростішій постановці, без обмежень на обсяги виробництва, задача розміщення виробництва є NP-важкою [1; 18]. Тому актуальними є дослідження різних варіантів цієї здачі з метою відшукання як точних, так і наближених алгоритмів. Варто зазначити, що для деяких спеціальних підзадач можлива побудова точних алгоритмів поліноміальної трудомісткості. Наприклад, у [9] для задачі з однаковими обмеженнями на обсяги виробництва на колійному графі (коли множина пунктів виробництва та попиту є вершинами, що лежать на одному шляху) з n вершинами та m ребрами запропоновано точний алгоритм з трудомісткістю $O(m^5n^2 + m^3n^3)$. В основному для різних варіантів задачі пропонуються алгоритми на основі метаевристик [10]. Задача розміщення виробництва розглядається як у безперервному варіанті [11], так і в дискретному як задача покриття графа зірками [12]. Сучасний огляд різних типів задач розміщення виробництва та методів їх розв'язку наведено у роботі [13].

Серед метаевристик, які використовуються під час пошуку субоптимальних розв'язків задачі розміщення виробництва, найчастіше трапляються варіації генетичного алгоритму та алгоритму мурашиної колонії [14], а також наближені алгоритми на основі ітераційного випадкового пошуку або локального пошуку в метричному просторі.

На жаль, всі наявні підходи до проблеми на основі метаевристиків не є універсальними. Для кожного типу задач необхідно розробляти свій варіант метаевристики та, відповідно, свої комп'ютерні реалізації. Крім того, якщо в задачі розміщення виробництва з'являються додаткові умови (наприклад, обмеження на обсяги поставок або кількість променів зірок покриття), то наявні алгоритми не спрацьовують і необхідно будувати нові варіанти метаевристик.

У роботі [16] запропоновано ідею підходу до задач дискретної оптимізації на основі використання фрагментарних структур. Такий підхід дозволяє розбити алгоритм розв'язання задачі на дві частини, причому перша з них (задача оптимізації на перестановках) є універсальною стосовно цільової функції. Фрагментарний алгоритм використовує знаходження значень цільової функції і зазвичай має поліноміальну трудомісткість.

У цій статті запропоновано використовувати цей підхід для дискретної задачі розміщення виробництва. Для цього пропонується модифікувати алгоритм перемішаних жаб, що стрибають, для пошуку субоптимальних розв'язків на множині всіх перестановок з довільною цільовою функцією. А потім привести задачу розміщення виробництва до задачі оптимізації на фрагментарній структурі.

Алгоритм перемішаних стрибаючих жаб для пошуку оптимальної перестановки

Алгоритм методу перемішаних стрибаючих жаб простий для розуміння та реалізації, має невелику кількість параметрів, успішно застосовувався для вирішення задач комбінаторної та безперервної оптимізації [8; 9].

Нехай задана функція $F(s)$ на множині перестановок n елементі S_n в. Задача полягає у пошуку перестановки $s^* \in S_n$, для якої значення функції $F(s)$ максимальне.

Будемо називати жабами перестановки з S_n . Суть алгоритму перемішаних стрибаючих жаб для пошуку оптимальної перестановки зводиться до такої послідовності кроків.

Крок 1. Ініціалізувати початкову популяцію жаб як множину точок простору перестановок S_n з метрикою Кендалла.

Крок 2. Обчислити значення критерію оптимальності кожної перестановки з початкової популяції.

Крок 3. Упорядкувати розв'язки (перестановки) у порядку зменшення значення критерію оптимальності.

Крок 4. Розділити віртуальних жаб (розв'язки) на блоки таким чином, що перша у відсортованому списку віртуальна жаба потрапляє до першого блоку, друга заноситься до другого блоку тощо.

Так триває доти, поки всі жаби не будуть розподілені у вказану кількість блоків.

Крок 5. У кожному блоці з номером $k \in \{1,2,...,K\}$ знайти найкращий s_{k1} та найгірший s_{k2} розв'язок.

Крок 6. Спробувати поліпшити становище найгіршої віртуальної жаби шляхом випадкового переміщення її у напрямку кращої жаби. Це відбувається шляхом застосування оператора кросоверу $s = Cross(s_{k2}, s_{k1})$.

Крок 7. Якщо попередня операція не покращує рішення, то спробувати покращити становище найгіршої віртуальної жаби шляхом переміщення її у напрямку глобально найкращої жаби $s = Cross(s_{k2}, s_{11})$.

Крок 8. Якщо й остання операція не призводить до поліпшення позиції віртуальної жаби, то замість неї випадковим чином створити в області пошуку нову жабу – перестановку.

Крок 9. Об'єднати віртуальні жаби всіх блоків в одну групу.

Крок 10. Якщо умови завершення алгоритму не виконані, то перехід до кроку 3.

Крок 11. Остання глобально найкраща віртуальна жаба відповідає субоптимальному розв'язку задачі.

Опишемо тепер цей алгоритм формально з урахуванням параметрів.

Параметри методу такі:

1) кількість класів жаб Q $(Q \ge 2)$;

2) кількість елементів r у кожному класі (передбачається, що розміри класів однакові та $r \ge 2$);

3) максимальна кількість кроків K алгоритму;

4) кількість D найкращих жаб у класі, причому $0 < D < r$.

Відповідно до заданих параметрів розмір N популяції жаб (множини допустимих рішень) визначається формулою $N = Qr$. На початковому етапі алгоритму створюється вихідна популяція жаб $P^{(0)}$ шляхом генерації випадкових перестановок $s^j = (i_{j1}, i_{j2}, .., i_{jn})$, $j = 1,2,..., N$.

Вибирається найкраща за функцією мети перестановка вершин, яка задає елемент $s^* = (i_1, i_2, ..., i_n) \in S_n$, та обчислюється значення цільової функції $F(x^*)$ на цій перестановці.

Крок k $(1 \le k \le K)$. Впорядковується множина $P^{(k-1)}$ за значенням цільової функції, тобто $F(s^k) \ge F(s^{k-1})$, $k = 2,3,..., N-1$. Розбивається популяція $P^{(k-1)}$ на Q класів однакової потужності r

$$P_q^{(k-1)} = \{s^{(qi)} \mid s^{(qi)} = x^j,\ j = q + (i-1)Q,$$

$$i = 1,2,..,r,\ q = 1,2,...,q\}.$$

Визначається найкращий розв'язок $s^* = s^1$ за значенням цільової функції на всій популяції. У кожному класі $P_q^{(k-1)}$ визначаються «кращий» $s^{(q1)}$ та «гірший» $s^{(qr)}$ за значенням цільової функції розв'язок. У кожному класі $P_q^{(k-1)}$ змінюються положення (послідовності обходу вершин графа) жаб із номерами від $D+1$ до r. Для кожного значення індексу $i \in \{D+1, 2, ..., r\}$ визначається нове положення i-ї жаби (послідовність обходу вершин) у класі з номером q за таким правилом:

Обчислюється випадкова перестановка s^c з відрізка між перестановками $s^{(q1)}$ та $s^{(qr)}$ в метриці Кендала. Для визначення перестановки s^c з відрізка між перестановками $s^{(q1)}$ і $s^{(qr)}$ використовується оператор кросоверу $s^c = Cross(s^{(q1)}, s^{(q2)})$. Перестановка на відрізку між $s^{(q1)}$ і $s^{(qr)}$ будується за таким правилом: послідовності $s^{(q1)}$ і $s^{(qr)}$ проглядаються зліва направо. На черговому кроці випадково вибирається один із перших елементів послідовностей і додається до нової перестановки – нащадку. Потім цей елемент видаляється з перестановок $s^{(q1)}$ і $s^{(qr)}$. Наприклад, можливий результат застосування цієї операції до перестановок (2, 4, 7, 6, 1, 3, 5, 8) та (5, 8, 1, 3, 4, 2, 6, 7) дає перестановку (2, 4, 5, 7, 6, 1, 3, 8).

Якщо $F(s^c) < F(s^{(qi)})$, то вважаємо $s^{(qi)} = s^c$. Якщо $F(s^c) \geq F(s^{(qi)})$, то вибирається випадкова перестановка s^c у відрізку між s^* і $s^{(qr)}$. Якщо $F(s^c) < F(s^{(qi)})$, то вважаємо $s^{(qi)} = s^c$. В іншому випадку вибираємо $s^{(qi)}$ як випадково згенеровану перестановку.

Вважаємо $P^{(k)} = \bigcup_{q=1}^{Q} P_q^{(k-1)}$ і переходимо до чергового кроку алгоритму.

Алгоритм закінчує роботу, коли проведено задану кількість кроків. Поточна перестановка s^*, визначена на останньому кроці, береться як оптимальний розв'язок задачі.

Зауважимо, що описаний вище алгоритм вирішує задачу відшукання оптимальної перестановки з n елементів на множині всіх перестановок S_n з цільовою функцією $F(s)$, яка задана на цій множині. Причому конкретний вид цільової функції не має значення. Тому розглянутий вище алгоритм може використовуватися для відшукання субоптимальних розв'язків задач оптимізації на множині всіх перестановок з довільними цільовими функціями. Далі буде показано, що цей алгоритм може бути узагальнено на широкий клас дискретних оптимізаційних задач, а саме задач, що мають орієнтовану фрагментарну структуру.

Задачі з орієнтованою фрагментарною структурою

Орієнтованою фрагментарною структурою [10] (X, E) на кінцевій множині X називається сімейство впорядкованих наборів $E = \{E_1, E_2, ..., E_n\}$ його елементів таких, що для будь-якої непустої послідовності $E_i = (x_1, x_2, ..., x_k) \in E$, будь-яка її початкова підпослідовність $(x_1, x_2, ..., x_{k'})$, $k' < k$ також належить E.

Елементи з множини E називатимемо допустимими фрагментами. Елементарним фрагментом називатимемо допустимий фрагмент, що складається з одного елемента. Максимальний фрагмент – допустимий фрагмент, який не є підмножиною іншого фрагменту.

Максимальний фрагмент може бути побудований за допомогою наступного «жадібного» алгоритму:

а) елементи множини X лінійно впорядковуються;

б) на початковому кроці вибирається порожня множина $X_0 = \varnothing$;

в) на кроці з номером $k + 1$ вибирається перший по порядку елемент $x \in X \setminus X_k$, такий, що $X_k \bigcup \{x\} \in E$;

г) алгоритм закінчує роботу, якщо на черговому кроці не вдалося знайти елемент $x \in X \setminus X_k$ із необхідною властивістю.

Результат роботи алгоритму визначається заданим лінійним порядком на множині X. Таким чином, будь-який максимальний фрагмент може бути описаний деякою перестановкою елементів множини X. Нехай $A \in E$. Умову для елемента $x \in X$, за якої $A \bigcup \{x\} \in E$, називатимемо умовою приєднання елемента x.

Нехай тепер кожному фрагменту приписано вагу, тобто задана функція $\rho : E \to R^1$. Припускатимемо, що функція ρ монотонна за включенням (зростаюча або спадна). Якщо $A, B \in E$ і $A \subseteq B$, то $\rho(A) \leq (\geq) \rho(B)$. Задача оптимізації на орієнтованій фрагментарній структурі – це задача відшукання допустимого фрагменту максимальної (мінімальної) ваги. Очевидно, що для монотонно зростаючих ваг оптимальне рішення буде максимальним фрагментом.

Будь-який максимальний фрагмент визначається заданим лінійним порядком перегляду елементарних фрагментів. Цей порядок визначає результат роботи фрагментарного алгоритму, який побудує необхідний максимальний фрагмент.

Кожен лінійний порядок визначається деякою перестановкою $s \in S_n$ елементарних фрагментів (n – кількість елементарних фрагментів). Зіставимо у кожній перестановці максимальний фрагмент, який їй породжує. Позначимо це відображення через $\varphi : S_n \to E$. Таким чином, має місце природна комутативна діаграма відображень:

$$\begin{array}{ll} S_n & \\ \varphi\downarrow & \searrow F \circ \varphi, \\ E \to R^1 & \end{array}$$

яка перетворює задачу оптимізації на орієнтованій фрагментарній структурі в задачу оптимізації на множині перестановок. Причому будь-яка перестановка є допустимою. Для великих значень n задача пошуку оптимальної перестановки, як правило, є важкою у обчислювальному сенсі. Тож для таких задач виправдано застосування метаевристик. Зокрема, алгоритм перемішаних стрибаючих жаб, який був описаний вище, автоматично переноситься на будь-яку задачу з орієнтованою фрагментарною структурою. Змінюється лише правило пошуку значень цільової функції. Для отримання цих значень використовується фрагментарний алгоритм, який є індивідуальним для кожного класу дискретних оптимізаційних задач з орієнтованою фрагментарною структурою. Як приклад використання алгоритму перемішаних стрибаючих жаб розглянемо задачу про розміщення виробництва.

Задача розміщення виробництва

Розглянемо найпростішу постановку задачі розміщення виробництва [11]. Нехай на евклідовій площині виділено множину N точок $X = \{M_1, M_2, ..., M_N\}$, кожна з яких характеризується двома евклідовими координатами. Кожна з точок є або місцем розташування майбутнього споживача продуктів виробництва, або можливої точки розміщення виробництва.

Усі об'єкти у цій постановці вважаються точковими. Як цільову функцію задачі приймаються витрати на відкриття виробництва в заданій точці або за доставку продукції до певного клієнта. Усі точки-споживачі розподіляються між точками виробництва таким чином, що:

а) кожен споживач приписаний одній і лише одній точці виробництва;

б) розподіл споживачів не змінюється;

в) кожній точці приписано вартість відкриття виробництва у цій точці;

г) вартість доставки продукції з точки виробництва до точки-споживача прямо пропорційна евклідовій відстані між точками.

Будь-який допустимий розв'язок задачі може бути представлений на площині графом, який має вигляд об'єднання зірок, що не перетинаються у вершинах. Пункти виробництва є центрами зірок, а промені пов'язують пункти виробництва з точками – споживачами продукту.

Задача, що розглядається, є *NP*-важкою [12]. Тому застосування метаевристики для пошуку наближених оптимальних розв'язків є виправданим.

Покажемо насамперед, що задача належить до класу оптимізаційних задач на орієнтованій фрагментарній структурі. Як множина елементарних фрагментів вибираються ребра повного графа з вершинами в точках множини X.

Кожен допустимий фрагмент будемо будувати, дотримуючись такої умови приєднання. Чергове ребро приєднується до вибраного набору ребер, якщо після приєднання отриманий підграф є об'єднанням зірок, що не перетинаються у вершинах. Якщо чергове ребро приєднати не вдається, переходимо до наступного по порядку ребра. Алгоритм закінчує роботу, якщо список ребер вичерпано. Множина ребер, які послідовно будуть побудовані в результаті роботи такого алгоритму (множина E), утворюють орієнтовану фрагментарну структуру. Центри одержаних у результаті роботи алгоритму зірок та ізольовані вершини (якщо в них локалізовані одержувачі продукції) є точками розміщення виробничих потужностей. Цільова функція задачі – вартість локалізації організації виробництва у вибраних точках розміщення виробництва плюс вартість доставки продукції споживачам по проміннях зірок. Очевидно, цільова функція є монотонною, а допустимий розв'язок обов'язково є максимальним фрагментом (за виключенням ізольованих точок, що не є точками розташування користувачів).

Будь-який максимальний фрагмент визначається заданим лінійним порядком перегляду елементарних фрагментів. Цей порядок визначає результат роботи фрагментарного алгоритму, який побудує необхідний максимальний фрагмент.

Оскільки задача розміщення виробництва може бути приведена до задачі оптимізації на множині перестановок розмірності $N(N-1)/2$, то до неї практично без змін може бути застосований алгоритм перемішаних стрибаючих жаб, який був описаний вище для пошуку оптимальної перестановки. Це дозволяє досить ефективно (поліноміальне обмеження від кількості вершин графу на кожній ітерації за заданої кількості числа ітерацій) отримувати субоптимальні розв'язки задачі розміщення виробництва.

Висновки

У роботі показано, що відома метаевристика, що заснована на алгоритмі перемішаних стрибаючих жаб, який добре зарекомендував себе у низці прикладних задач, може з успіхом застосовуватися для різних класів дискретних оптимізаційних задач, що мають орієнтовану фрагментарну структуру. Для пошуку субоптимального розв'язку задачі розміщення виробництва наведено простий гібридний алгоритм, заснований на комбінації алгоритму перемішаних стрибаючих жаб і фрагментарного алгоритму.

Для перевірки якості алгоритму використовувалася програма «Фрагментарні структури та еволюційні алгоритми» (свідоцтво про реєстрацію авторського права на твір № 111008). Було згенеровано сто тестових завдань у вигляді графів із щільністю ребер 0,5–0,7. Ваги ребер графів у тестових задачах рівномірно розподілені на проміжку [10, с. 500]. Порівняння проводилося з алгоритмом ітеративного випадкового пошуку та алгоритмом локального пошуку. У першому випадку задавалася кількість ітерацій (популяцій в алгоритмі жаб, що стрибають). У другому випадку робота алгоритму обмежувалася заданим часом розрахунку. Результати чисельного експерименту показали, що у 100% тестових задач запропонований алгоритм показував кращі результати, ніж алгоритм ітеративного випадкового пошуку. Приблизно на 80% тестових задач алгоритм методу жаб, що стрибають, випереджав алгоритм локального пошуку.

Таким чином, запропонована методика побудови гібридних алгоритмів показала ефективність на розглянутому класі задач. Вона легко може бути розширена і на інші варіанти метаевристик, які розроблені для пошуку субоптимальних розв'язків задачі оптимізації на множині перестановок. У перспективі передбачається розширити цей метод на більшість класів задач розміщення виробництва з різними обмеженнями і навіть на суміжні типи задач, такі як задачі покриття графів типовими підграфами.

ЛІТЕРАТУРА

1. Garey M.R. and Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco, CA, 1979.
2. Eusuff M.M., Lansey K.E. Optimization of water distribution network design using the shuffled frog leaping algorithm *J. Water Resour. Planning Mgmt*. 2003. Vol. 129. P. 210–225.
3. Khumawala B.M. An Efficient Branch-Bound Algorithm for the Warehouse Location Problem. *Management Science*. V. 18. 1972. P. 718 D. S. 731.
4. Krarup J., Pruzan P.M. The simple plant location problem: Survey and synthesis *European Journal of Operational Research*. V. 12. 1983. P. 36 D. S. 81.
5. Karen Aardal, Jaroslaw Byrka, and Mohammad Mahdian. Facility location. In Encyclopedia of Algorithms. Springer, 2016. P. 717–724. DOI: 10.1007/978-1-4939-2864-4_139.
6. Sean Luke. Essentials of Metaheuristics, Lulu. 2009. URL: http://cs.gmu.edu/~sean/book/metaheuristics/.
7. Щербина О.А. Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор). *Таврический вестник информатики и математики*. 2014. № 1. С. 56–72.
8. Blum C., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison *ACM Computing Surveys* (CSUR). 35 (3). P. 268–308.
9. Ageev A.A. A polynomial-time algorithm for the facility location problem with uniform hard capacities on path graphs. Proc 2nd Int. Workshop Discrete Optimization Methods in Production and Logistics (DOM'2004). Omsk : Nasledie. Dialog-Sibir. 2004. P. 28–32.
10. Herminia I. Calvete, Carmen Galé, José A Iranzo, José-Fernando CamachoVallejo, and Martha-Selene Casas-Ramírez. A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Computers & Operations Research.* 124. 2020. URL: https://doi.org/10.1016/j.cor.2020.105066.
11. Sutanto G.R., Kim S., Kim D., Sutanto H. A heuristic approach to handle capacitated facility location problem evaluated using clustering internal evaluation. *IOP Conf. Series: Materials Science and Engineering.* 332. 2018. P. 1–8. DOI: 10.1088/1757-899X/332/1/012023.
12. Edward Gimadi, Alexandr Shtepa, and Oxana Tsidulko. Improved Exact Algorithm for the Capacitated Facility Location Problem on a Line Graph. In 2019 *15th International Asian School-Seminar Optimization Problems of Complex Systems* (OPCS). 2019. P. 53–57. URL: https://doi.org/10.1109/OPCS.2019.8880248.
13. Derya C Turkoglu and Mujde E Genevois. A comparative survey of service facility location problems. *Annals of Operations Research.* 292. Issue 1. 2019. P. 399–468. URL: https://doi.org/10.1007/s10479-019-03385-x.
14. Mohd Jani, Nurul Hafiza, Mohd Radzi, Nor Haizan, Ngadiman, Mohd Salihin. Ant colony optimization for solving university facility layout problem. *Proceedings of the 20th National Symposium on Mathematical Sciences: Research in Mathematical Sciences: A Catalyst for Creativity and Innovation*. AIP Conference Proceedings, Volume 1522. Issue 1. 2013. P. 1355–1359. URL: https://doi.org/10.1063/1.4801286.
15. Narimani M.R. A New Modified Shuffle Frog Leaping Algorithm for NonSmooth Economic Dispath World. *Applied Sciences Journal.* 2011. P. 803–814.
16. Kozin I.V., Maksyshko N.K., Perepelitsa V.A. Fragmentary Structures in Discrete Optimization Problems. *Cybernetics and Systems Analysis.* November 2017. V. 53. P. 931–936. DOI: https://doi.org/10.1007/s10559-017-9995-66.
17. Козин И.В., Полюга С.И., Сардак В.И. Фрагментарная модель размещения производства. Сборник «*Математичне та комп'ютерне моделювання». Серія «Фізико-математичні науки».* 2019. Вип. 19. Кам'янець-Подільський національний університет ім. Івана Огієнка. С. 35–40.
18. Christos H. Papadimitriou, Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall. 1982. ISBN 0-13-152462-3.

REFERENCES

1. Garey, M.R. and Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, San Francisco, CA.
2. Eusuff, M.M., Lansey, K.E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm J. Water Resour. Planning Mgmt. Vol. 129. Pp. 210–225.
3. Khumawala, B.M. (1972). An Efficient Branch-Bound Algorithm for the Warehouse Location Problem. Management Science. V. 18. Pp. 718–731.
4. Krarup J., Pruzan P.M. (1983). The simple plant location problem: Survey and synthesis. European Journal of Operational Research. V. 12. Pp. 36–81.

5. Karen Aardal, Jaroslaw Byrka, and Mohammad Mahdian (2016). Facility location. In Encyclopedia of Algorithms, Springer, pp. 717–724.
6. Sean Luke (2009). Essentials of Metaheuristics, Lulu. Retrieved from: http://cs.gmu.edu/~sean/book/metaheuristics/.
7. Scherbina, O.A. (2014). Metaheuristic Algorithms for Combinatorial Optimization Problems (Review) [Metaevristicheskie algoritmyi dlya zadach kombinatornoy optimizatsii (obzor)]. Tavricheskiy vestnik informatiki i matematiki. № 1. Pp. 56–72.
8. Blum, C., Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison ACM Computing Surveys (CSUR), 35 (3), 268–308.
9. Ageev, A.A. (2004). A polynomial-time algorithm for the facility location problem with uniform hard capacities on path graphs. Proc 2nd Int. Workshop Discrete Optimization Methods in Production and Logistics (DOM'2004). Omsk: Nasledie; Dialog-Sibir, pp. 28–32.
10. Herminia, I. Calvete, Carmen Galé, José A Iranzo, José-Fernando CamachoVallejo, and Martha-Selene Casas-Ramírez. (2020). A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. Computers & Operations Research 124.
11. Sutanto G.R., Kim S., Kim D., Sutanto H. (2018). A heuristic approach to handle capacitated facility location problem evaluated using clustering internal evaluation, IOP Conf. Series: Materials Science and Engineering, 332, pp. 1–8.
12. Edward Gimadi, Alexandr Shtepa, and Oxana Tsidulko. (2019). Improved Exact Algorithm for the Capacitated Facility Location Problem on a Line Graph. In 2019 15th International Asian School-Seminar Optimization Problems of Complex Systems (OPCS). Pp. 53–57.
13. Derya C Turkoglu and Mujde E Genevois (2019). A comparative survey of service facility location problems. Annals of Operations Research 292. Issue 1. Pp. 399–468.
14. Mohd Jani, Nurul Hafiza, Mohd Radzi, Nor Haizan, Ngadiman, Mohd Salihin (2013). Ant colony optimization for solving university facility layout problem. Proceedings of the 20th National Symposium on Mathematical Sciences: Research in Mathematical Sciences: A Catalyst for Creativity and Innovation. AIP Conference Proceedings, V. 1522. Issue 1. Pp. 1355–1359.
15. Narimani, M.R. (2011). A New Modified Shuffle Frog Leaping Algorithm for NonSmooth Economic Dispath. World Applied Sciences Journal. Pp. 803–814.
16. Kozin, I.V., Maksyshko, N.K., Perepelitsa, V.A. (2017). Fragmentary Structures in Discrete Optimization Problems, Cybernetics and Systems Analysis November, V. 53, Issue 6, pp. 931–936.
17. Kozin, I.V., Polyuga, S.I., Sardak, V.I. (2019). Fragmented production location model [Fragmentarnaya model razmescheniya proizvodstva]. Sb. «Matematichne ta komp'yuterne modelyuvannya». Seriya «Fiziko-matematichni nauki». V. 19. Kam'yanets-Podilskiy natsionalniy universitet im. Ivana Ogienka, pp. 35–40.
18. Christos H. Papadimitriou, Kenneth Steiglitz (1982). Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall. ISBN 0-13-152462-3.