УДК 004.023 DOI https://doi.org/10.35546/kntu2078-4481.2025.2.2.6

І. Г. БУГАЄВА
кандидат технічних наук,
доцент кафедри технічної кібернетики
й інформаційних технологій ім. проф. Р. В. Мерката
Одеський національний морський університет
ORCID: 0000-0002-2839-9266

АНАЛІЗ ПАРАМЕТРІВ ГЕНЕТИЧНОГО АЛГОРИТМУ РОЗВ'ЯЗАННЯ ДВОВИМІРНОЇ ЗАДАЧІ УПАКОВКИ

Задача упаковки є актуальною і зустрічається у багатьох промислових виробництвах. Вона належить до класу NP-складних задач, тому використання точного перебірного алгоритму можливе лише для задач невеликої розмірності. Метаевристичні методи є популярним інструментом для наближеного вирішення складних комбінаторних оптимізаційних задач. У статті розглядається розв'язання задачі упаковки прямокутників у напівнескінченну смугу фіксованої ширини за допомогою генетичного алгоритму. Метою статті є дослідження впливу генетичних операторів схрещування та відбору на ефективність роботи алгоритму. У роботі наводиться постановка задачі, представлено генетичні оператори кросинговеру та відбору, які використовуються в алгоритмі. Розв'язання задачі упаковки ділиться на два етапи: визначення за допомогою генетичного алгоритму перестановки для послідовності пакування прямокутних елементів та застосування алгоритму декодування для їх розміщення у смузі. В якості декодеру використовувся алгоритм BLF. Проведено обчислювальний експеримент на тестових прикладах, відомих з літературних джерел, а також на двох згенерованих наборах прямокутників. Всі тестові задачі, що використовуються в цьому дослідженні, мають оптимальне рішення з мінімально допустимою довжиною зайнятої частини смуги. Програму реалізовано за допомогою мови програмування Python та пакету Deap. Аналіз результатів експериментів показав, що найкращі рішення задачі упаковки прямокутників у напівнескінченну смугу було отримано при використанні в генетичному алгоритмі оператора схрещування PMX та оператора турнірного відбору. Генетичний алгоритм показав найгірший результат при застосуванні оператора кросинговеру UPMX та відбору за правилом рулетки.

Ключові слова: *двовимірна задача упаковки, генетичний алгоритм, генетичні оператори, оптимізація.*

I. H. BUHAIEVA
Ph.D., Associate Professor at the Department of Technical Cybernetics
and Information Technologies named after Professor R.V. Merkt
Odesa National Maritime University
ORCID: 0000-0002-2839-9266

ANALYSIS OF PARAMETERS OF GENETIC ALGORITHM FOR SOLVING TWO-DIMENSIONAL PACKING PROBLEM

The task of packaging is a topical one and is encountered in many industrial productions. It belongs to the class of NP-hard problems, so the use of an exact enumeration algorithm is possible only for small-dimensional problems. Metaheuristic methods are a popular tool for the approximate solution of complex combinatorial optimization problems. The article considers solving the problem of packing rectangles into a semi-infinite strip of fixed width using a genetic algorithm. The purpose of the article is to study the influence of genetic operators of crossover and selection on the efficiency of the algorithm. The paper provides a statement of the problem, presents the genetic operators of crossover and selection used in the algorithm. The solution of the packing problem is divided into two steps: determining the permutation for the packing sequence of rectangular elements using a genetic algorithm, and applying a decoding algorithm to place them in the strip. The BLF algorithm was used as a decoder. A computational experiment was carried out on test examples known from literary sources, as well as on two generated sets of rectangles. All test problems used in this study have an optimal solution with the minimum admissible length of the occupied part of the strip. The program is implemented using the Python programming language and the Deap package. Analysis of the experimental results showed that the best solutions to the problem of packing rectangles into a semi-infinite strip were obtained when using the PMX crossover operator and the tournament selection operator in the genetic algorithm. The genetic algorithm showed the worst result when using the UPMX crossover operator and roulette selection.

Key words: *two-dimensional packing problem, genetic algorithm, genetic operators, optimization.*

Постановка проблеми

У статті розглядається двовимірна задача упаковки прямокутників у напівнескінченну смугу. Задача є актуальною і зустрічається у промислових виробництвах у таких галузях, як, наприклад, паперова, скляна, текстильна.

Вона полягає в тому, що необхідно розмістити деяку кількість невеликих прямокутних об'єктів у напівнескінченній смузі фіксованої ширини таким чином, щоб їхні сторони були паралельні сторонам смуги та довжина зайнятої смуги була мінімальною.

Для вирішення цієї задачі існує безліч методів: точні методи, запропоновані Гілмором та Гоморі, різні евристики та метаевристики.

Задача упаковки є NP-складною. Зі збільшенням кількості прямокутників, які пакуються у смугу, різко зростає кількість можливих варіантів розміщення прямокутників і розв'язати точними методами задачу за поліноміальний час неможливо. Популярним інструментом для наближеного вирішення складних комбінаторних оптимізаційних задач є метаевристики – генетичні алгоритми, метод імітації відпалу, пошук з заборонами та інші.

Генетичні алгоритми для розв'язання оптимізаційних задач були запропоновані Джоном Холландом у 1975 році та широко використовуються в даний час.

Формулювання мети дослідження

Метою статті є дослідження впливу генетичних операторів кросинговеру та відбору на ефективність роботи генетичного алгоритму розв'язання двовимірної задачі упаковки прямокутників у напівнескінченну смугу.

Аналіз останніх досліджень і публікацій

У статті [1] наведено огляд двовимірних задач розкрою та упаковки. Автори розглядають основні формулювання та методи рішення для двовимірних ортогональних задач розрізання та пакування, де і предмети, і контейнери є прямокутниками. У роботі описано основні методи попередньої обробки даних та релаксації, коротко розглянуто евристичні та апроксимаційні алгоритми для чотирьох основних завдань з літератури: пошук упаковки з мінімальною висотою, пакування елементів у мінімальну кількість контейнерів, знаходження упаковки з максимальною цінністю та визначення існування допустимої упаковки.

У роботі [2] розглядаються двовимірні задачі упаковки в контейнери з шириною W і висотою H (2BP) і контейнер з шириною W і нескінченною висотою, званий смугою (2SP). Обговорюються математичні моделі двовимірних задач упаковки, класичні алгоритми наближення, а також евристичні та метаевристичні методи.

Задачі розміщення набору прямокутних елементів на смузі мінімальної висоти присвячена стаття [3]. Автори представляють алгоритм табу-пошуку, генетичний алгоритм та гібридний алгоритм, який поєднує два підходи. Продуктивність запропонованих алгоритмів оцінюється за допомогою обчислювальних експериментів на прикладах з літератури та на випадково згенерованих прикладах.

У статті [4] запропоновано генетичний алгоритм, який використовує як декодер BL алгоритм розміщення прямокутників у напівнескінченну смугу. Наведено результати експериментів на двох наборах прямокутників. У першому наборі J1 25 прямокутників різних розмірів, набір J2 включає 50 прямокутних елементів. Порівнюються шаблони упаковки, отримані з використанням евристики – алгоритму BL та із застосуванням метаевристики – генетичного алгоритму, де як декодер використовується BL-алгоритм. Застосування генетичного алгоритму забезпечує зменшення висоти пакування прямокутників на 19 %.

У роботі [5] на семи тестових задачах C1–C7 порівнюються результати роботи евристик BL та BLF, а також результати застосування різних метаевристик, у тому числі генетичних алгоритмів, комбінованих з BL та BLF алгоритмами. Результати експериментів показали, що використання в генетичному алгоритмі в якості декодера евристики BLF дає кращі результати у порівнянні з алгоритмом BL, особливо зі збільшенням кількості прямокутників у тестовому наборі.

Викладення основного матеріалу дослідження

Постановка задачі. Розглядається двовимірна задача ортогональної упаковки прямокутників в напівнескінченну смугу. Даний набір із n прямокутних елементів, кожен із яких визначається шириною w_i і довжиною l_i ($i = 1, \ldots, n$). Дана область упаковки (напівнескінченна смуга) шириною W. Необхідно так розташувати набір прямокутних елементів на напівнескінченній смузі, щоб їх w-ребра були паралельні W-ребру смуги та щоб довжина зайнятої частини смуги була мінімальною.

Введемо прямокутну систему координат XOY. Припускаємо, що вісь Ox збігається з верхньою необмеженою стороною смуги, Oy – з бічною стороною смуги. При цьому прямокутники не повинні накладатися один на одного і виходити за межі смуги:

$$((x_i \geq x_j + l_j) \vee (x_j \geq x_i + l_j)) \vee ((y_i \geq y_j + w_j) \vee (y_j \geq y_i + w_i), \quad i, j = 1, 2, \ldots, n, i \neq j;$$

$$(x_i \geq 0) \vee (y_i \geq 0) \vee (y_i + w_i \leq W), \quad i = 1, 2, \ldots, n,$$

де (x_i, y_i) – координати лівого нижнього кута i-го прямокутника на смузі. Повороти прямокутників не допускаються.

Мета задачі – отримати схему упаковки, яка мінімізує довжину зайнятої частини смуги L, тобто знайти такий набір координат (x_i, y_i), $i = 1, 2, \ldots, n$, щоб $L = \max_i (x_i + l_i) \to \min$.

Алгоритми декодування. Схема упаковки прямокутників у напівнескінченну смугу представляється перестановкою, яка відповідає послідовності, в якій прямокутники повинні розміщуватись у смузі. Розв'язання задачі упаковки ділиться на два етапи: визначення перестановки для послідовності розміщення прямокутних елементів

та застосування алгоритму їх розміщення у смузі (алгоритма декодування). Існує безліч алгоритмів декодування, наприклад, такі як BL (bottom-left), BLF (bottom-left-fill), IBL (improved bottom-left).

Алгоритм BL (нижній лівий) [6] упаковує прямокутники в заданому порядку в міру їхнього послідовного вилучення зі списку. Прямокутник спочатку розміщується у верхньому правому куті основного об'єкта, переміщається вниз наскільки можливо, а потім вліво.

В алгоритмі IBL (удосконалений нижній лівий) [7] прямокутний елемент переміщається вниз і ліворуч, поки він не зможе рухатися далі. Різниця між алгоритмом BL та покращеним алгоритмом IBL полягає в тому, що прямокутник в алгоритмі IBL переміщається з перевіркою нижнього порожнього простору.

Алгоритм BLF (нижній лівий із заповненням) [8] намагається помістити прямокутник у найнижчу доступну позицію основного об'єкта та вирівняти її по лівому краю. У порівнянні з алгоритмами BL та IBL, алгоритм BLF має більш щільні шаблони компонування.

Генетичний алгоритм. В теорії про генетичні алгоритми структура даних називається генотипом, а її декодування – фенотипом [9]. У задачі упаковки результатом декодування є шаблон упаковки прямокутників. Генотип представляє собою перестановку цілих чисел, яка задає послідовність упаковки прямокутників у смугу.

У даній роботі для розв'язання задачі упаковки генетичний алгоритм використовується для отримання перестановок, а алгоритм BLF – для розміщення прямокутників у смузі відповідно до певної перестановки.

Для тестування алгоритму, який було реалізовано за допомогою мови програмування Python та пакету Deap, використовувалося кілька наборів даних: відомі з літературних джерел тести J1, J2 [4], набори даних P1 кожного з тестів C1–C6 [5], а також два випадково згенерованих набору елементів T1, T2 з 61 і 31 прямокутниками відповідно. Всі тестові задачі в цьому дослідженні мають оптимальні рішення з мінімально допустимою довжиною зайнятої частини смуги.

Наприклад, схема упаковки з оптимальним розміщенням прямокутників у смузі шириною $W = 120$ для тестового завдання T1 представлена на рисунку 1. Довжина зайнятої частини смуги $L_{opt} = 150$.

Оператори відбору. Серед операторів відбору найбільш поширеними є два імовірнісних оператори:

– турнірний відбір: у кожному раунді турнірного відбору з популяції випадково вибирається певна кількість особин, і та, у якої пристосованість більша, відбирається у наступне покоління;

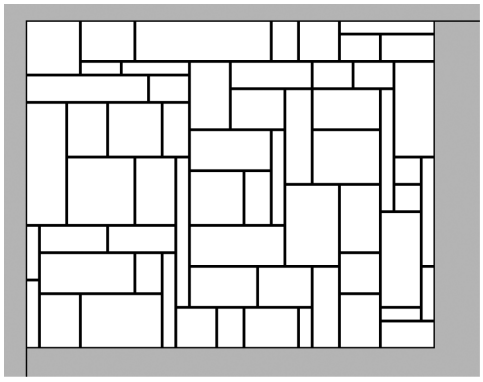


Рис. 1. Оптимальна схема упаковки для задачі T1, $L_{opt} = 150$

– відбір пропорційно пристосованості (відбір за правилом рулетки): кожній особині призначається ймовірність, що дорівнює відношенню її пристосованості до сумарної пристосованості популяції; потім відбувається відбір всіх особин популяції для подальшої генетичної обробки згідно з величиною ймовірності. При такому відборі члени популяції з більш високою пристосованістю з більшою ймовірністю частіше вибиратимуться, ніж особини з низькою пристосованістю.

Оператори схрещування. Оскільки цілочисельні хромосоми представляють індекси впорядкованого списку, використовувалися оператори схрещування, що гарантують, що завжди створюється допустимий нащадок [10]:

– Order Crossover (OX);
– Partially Mapped Crossover (PMX);
– Uniform Partial Mapped Crossover (UPMX).

При упорядкованому схрещуванні (OX) випадково вибираються дві точки розрізу хромосом-батьків. Гени, розташовані між двома точками в обох хромосомах, обмінюються між собою. Потім у кожному нащадку заповнюються гени, для чого використовуються гени відповідного батька, починаючи з наступного за другою точкою розрізу і переносяться тільки ті значення, яких ще немає в нащадку.

У методі схрещування з частковим зіставленням (PMX) дві точки схрещування вибираються випадковим чином у хромосомах-батьках. Гени першої хромосоми, розташовані між цими точками, обмінюються з генами

другої хромосоми, що бере участь у схрещуванні. Відображення значень генів при обміні використовується для заповнення інших генів обох хромосом. При цьому виключається дублювання значень генів.

Рівномірне схрещування із частковим зіставленням (UPMX) генерує двох нащадків шляхом зіставлення пар значень, що відповідають індексам, обраним випадковим чином у двох батьках, та обміну значеннями кожної з пар у нащадках, які спочатку є копіями батьків.

Оператор мутації. В генетичному алгоритмі використовувався оператор мутації Shuffle, який перетасовує елементи вхідної послідовності із заданою ймовірністю переміщення кожного елемента.

Результати досліджень. В роботі досліджувався вплив операторів схрещування OX, PMX, UPMX та операторів турнірного відбору та відбору за правилом рулетки на ефективність роботи генетичного алгоритму розв'язання задачі упаковки прямокутників у напівнескінченну смугу.

Використовувалися такі значення параметрів алгоритму: розмір популяції – 100, ймовірність мутації – 0,1, ймовірність схрещування – 0,9, кількість поколінь – 100. Програма виконувалася з однаковими параметрами для кожного тестового набору даних по десять разів.

Результати експериментів наведено у вигляді значень відносних відхилень обчислених довжин зайнятої смуги L від оптимальної довжини L_{opt}:

$$dev = \frac{L - L_{opt}}{L_{opt}} \cdot 100\ \%.$$

В таблиці 1 представлені значення відносних відхилень dev (%) для кращих рішень тестових задач.

Таблиця 1

Результати експериментів – мінімальні значення dev (%)

Тестові приклади	Оператор схрещування					
	OX		PMX		UPMX	
	Оператор відбору					
	турнір	рулетка	турнір	рулетка	турнір	рулетка
C1/P1	0,00	0,00	0,00	0,00	0,00	0,00
C2/P1	6,67	6,67	0,00	6,67	6,67	6,67
C3/P1	3,33	3,33	6,67	10,00	10,00	10,00
C4/P1	5,00	6,67	6,67	8,33	6,67	8,33
C5/P1	0,00	1,11	0,00	0,00	0,00	2,22
C6/P1	5,83	7,5	5,83	7,50	8,33	10,83
J1	6,67	6,67	6,67	6,67	6,67	6,67
J2	6,67	6,67	6,67	6,67	6,67	6,67
T1	3,33	3,33	3,33	3,33	3,33	6,67
T2	6,67	6,67	3,33	6,67	3,33	6,67

Середні значення відносних відхилень dev (%), які обчислені по результатах десяти прогонів алгоритму для кожної тестової задачі, наведено в таблиці 2.

Таблиця 2

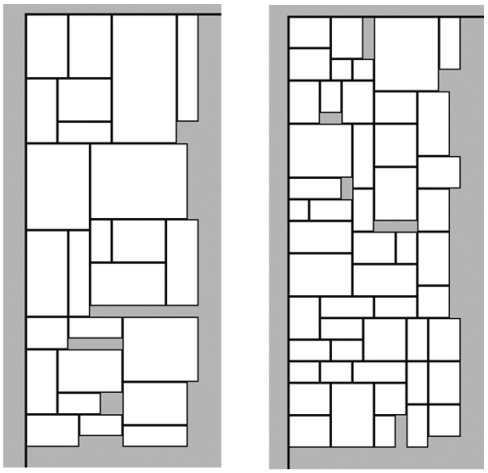
Результати експериментів – середні значення dev (%)

Тестові приклади	Оператор схрещування					
	OX		PMX		UPMX	
	Оператор відбору					
	турнір	рулетка	турнір	рулетка	турнір	рулетка
C1/P1	0,50	0,50	0,50	2,00	4,50	4,00
C2/P1	6,67	6,67	6,00	6,67	6,67	6,67
C3/P1	6,00	6,33	7,33	10,33	11,00	12,00
C4/P1	8,33	9,17	8,33	9,83	9,33	11,50
C5/P1	0,44	0,88	0,22	1,11	0,67	2,89
C6/P1	7,25	8,33	6,11	8,89	9,17	11,39
J1	7,33	7,33	7,33	8,00	9,33	10,67
J2	6,67	6,67	6,67	9,33	6,67	10,67
T1	6,00	5,55	4,45	5,55	4,45	6,67
T2	6,67	7,00	6,33	6,67	7,00	8,33
Середнє	5,59	5,84	5,33	6,84	6,88	8,48

Як видно з таблиці 2, найкращі рішення тестових задач (середнє відхилення 5,33 %) були отримані при використанні оператора схрещування PMX та оператора турнірного відбору. Застосовуючи в генетичному алгоритмі оператор кросинговеру OX та турнірний відбір або відбір за правилом рулетки були отримані близькі значення середніх відхилень (5,59 % та 5,84 % відповідно).

Найгірший результат середнього відхилення 8,48 % було отримано при використанні оператора схрещування UPMX та відбору за правилом рулетки.

На рис. 2 наведено схеми упаковки прямокутників у напівнескінченну смугу, які були отримані у результаті роботи генетичного алгоритму для тестових задач J1, J2. Оптимальна довжина зайнятої смуги $L_{opt} = 15$. Обчислена довжина в обох задачах $L = 16$. Відносне відхилення обчисленої довжини зайнятої смуги від оптимальної довжини дорівнює 6,66 %.



а б

Рис. 2. Шаблони упаковки, створені при розв'язанні задач: *а*) J1, *б*) J2

Результат роботи генетичного алгоритму з використанням кросинговеру PMX та турнірного відбору для тестової задачі T1 наведено на рис. 3, де представлено графіки мінімальних та середніх значень функції пристосованості, обчислених протягом 100 ітерацій генетичного алгоритму.

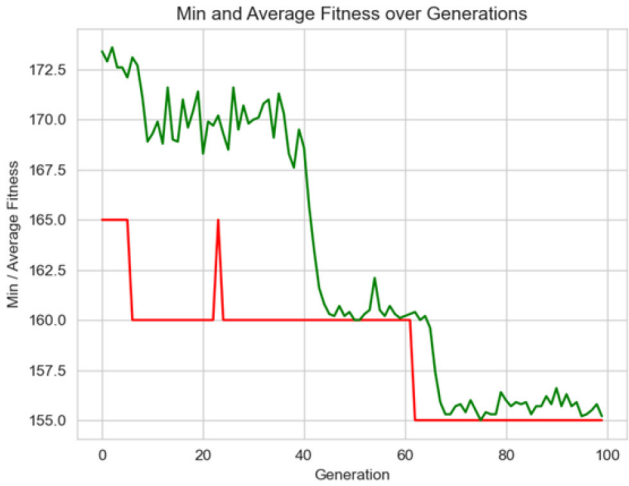


Рис. 3. Мінімальні та середні значення функції пристосованості

Схема упаковки, яка була отримана при розв'язанні задачі T1, представлена на рис. 4. Обчислена довжина $L = 155$.

Величина відносного відхилення обчисленого значення довжини зайнятої смуги від оптимальної довжини для задачі T1 становить 3,33 %.

Висновки

У дослідженні проаналізовано вплив операторів схрещування OX, PMX, UPMX та операторів турнірного відбору й відбору за правилом рулетки на ефективність роботи генетичного алгоритму при розв'язанні задачі упаковки прямокутників у напівнескінченну смугу.

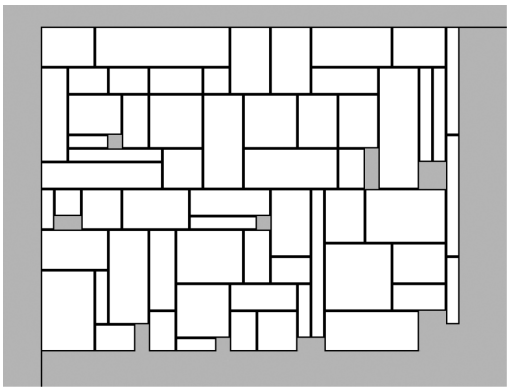


Рис. 4. Шаблон упаковки, створений при розв'язанні задачі T1

Результати досліджень показали, що використання генетичного алгоритму з оператором кросинговеру PMX та турнірним відбором дає найкращі рішення у більшості тестових задач.

Слід зазначити, що на деяких наборах тестових даних найкращий результат роботи генетичного алгоритму був отриманий при застосуванні оператора схрещування OX та турнірного відбору.

Для розв'язання задачі упаковки прямокутників у напівнескінченну смугу фіксованої ширини не рекомендується використовувати в генетичних алгоритмах оператор кросинговеру UPMX.

Список використаної літератури

1. Iori M., de Lima V. L., Martello S., Miyazawa F., Monaci M. Exact Solution Techniques for Two-dimensional Cutting and Packing. *European Journal of Operational Research*. 2020. Vol. 289(2). P. 399–415. DOI: 10.1016/j.ejor.2020.06.050

2. Lodi A., Martello S., Monaci M. Two-dimensional Packing Problems: A survey. *European Journal of Operational Research*. 2002. Vol. 141(2). P. 241–252. https://doi.org/10.1016/S0377-2217(02)00123-6

3. Iori M., Martello S., Monaci M. Metaheuristic Algorithms for the Strip Packing Problem. *Optimization and Industry: New Frontiers. Applied Optimization*. 2003. Vol. 78. P. 159–179. DOI:10.1007/978-1-4613-0233-9_7

4. Jakobs S. On genetic algorithms for the packing of polygons. *European Journal of Operational Research*. 1996. Vol. 88(1). P. 165–181. https://doi.org/10.1016/0377-2217(94)00166-9

5. Hopper E., Turton B. C. H. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*. 2001. Vol. 128(1). P. 34–57. DOI:10.1016/S0377-2217(99)00357-4

6. Baker B. S., Coffman E. G., Rivest R. L. Orthogonal packings in two dimensions. *SIAM Journal on Computing*. 1980. Vol. 9(4). P. 846–855. DOI:10.1137/0209064

7. Liu D., Teng H. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*. 1999. Vol. 112(2). P. 413–420. https://doi.org/10.1016/S0377-2217(97)00437-2

8. Chazelle B. J. The bottom-left bin-packing heuristic: an efficient implementation. *IEEE Transactions on Computers*. 1983. Vol. 32, No.8. P. 697–707. https://doi.org/10.1109/TC.1983.1676307

9. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer-Verlag. 1992. 388 p.

10. Cicirello V. A. A Survey and Analysis of Evolutionary Operators for Permutations. *Proceedings of the 15th International Joint Conference on Computational Intelligence ECTA*. 2023. Vol. 1. P. 288–299. DOI:10.5220/0012204900003595

References

1. Iori, M., de Lima, V. L., Martello, S., Miyazawa, F., & Monaci M. (2020). Exact Solution Techniques for Two-dimensional Cutting and Packing. *European Journal of Operational Research*, 289(2), 399-415. DOI:10.1016/j.ejor.2020.06.050

2. Lodi, A., Martello, S., & Monaci, M. (2002). Two-dimensional Packing Problems: A survey. *European Journal of Operational Research*, 141(2), 241–252. https://doi.org/10.1016/S0377-2217(02)00123-6

3. Iori, M., Martello, S., & Monaci, M. (2003). Metaheuristic Algorithms for the Strip Packing Problem. *Optimization and Industry: New Frontiers. Applied Optimization*, 78, 159–179. DOI:10.1007/978-1-4613-0233-9_7

4. Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88, 165–181. https://doi.org/10.1016/0377-2217(94)00166-9

5. Hopper, E. & Turton, B. C. H. (2001). An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*, 128, 34–57. DOI:10.1016/S0377-2217(99)00357-4

6. Baker, B. S., Coffman, E. G., & Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4), 846–855. DOI:10.1137/0209064

7. Liu, D. & Teng, H. (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112, 413–420. https://doi.org/10.1016/S0377-2217(97)00437-2

8. Chazelle, B. J. (1983). The bottomn-left bin-packing heuristic: an efficient implementation. *IEEE Transactions on Computers*, 32(8), 697–707. https://doi.org/10.1109/TC.1983.1676307

9. Michalewicz, Z. (1992). Genetic Algorithms + Data Structures = Evolution Programs. Berlin : Springer-Verlag.

10. Cicirello, V. A. (2023). A Survey and Analysis of Evolutionary Operators for Permutations. *Proceedings of the 15th International Joint Conference on Computational Intelligence ECTA*, 1, 288-299. DOI:10.5220/0012204900003595