**СЕКЦІЯ КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ І СИСТЕМ**

**УДК 004.023:519**

*О.В. Скакаліна, к.т.н., доцент*

*Національний університет*

*«Полтавська політехніка імені Юрія Кондратюка»*

*А. О. Клочко, магістрантка Навчально-наукового інституту*

*Інформаційних технологій і механотроніки*

*Національний університет*

*«Полтавська політехніка імені Юрія Кондратюка»*

**ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ РОЗКРОЮ КОМПЛЕКТУЮЧИХ**

**ДЛЯ КОРПУСНИХ МЕБЛІВ**

Двовимірна упаковка є однією з найбільш затребуваних і актуальних варіацій завдання упаковки, так як сфера її застосування в промисловості дуже велика - ефективні алгоритми двовимірної упаковки необхідні для ведення складського господарства, розміщення і транспортування вантажів, розкроюванні матеріалів, і в багатьох інших областях. Завдання двовимірної упаковки може також використовуватися при вирішенні більш складних завдань, наприклад, при вирішенні завдань розподілу ресурсів, в задачах виділення обчислювальних кластерів, при побудові механізмів алокації і очищення пам'яті в віртуальних обчислювальних середовищах. Настільки висока гнучкість формулювання і виділення обмежуючих умов значно підвищує практичну цінність даного завдання, і підтримує її актуальність [1].

Складність вирішення завдання упаковки обумовлена ​​її приналежністю до класу NP-повних задач, для яких неможливе застосування точних методів в умовах реального виробництва через великі витрати часових ресурсів. У зв'язку з цим, одним з найбільш перспективних напрямків досліджень є розробка та вдосконалення різних наближених, а також евристичних методів вирішення завдань упаковки. Найбільш ефективними і такими, що добре зарекомендували себе при вирішенні таких завдань є еволюційні алгоритми як підклас евристичного класу оптимізаційних алгоритмів [2]. Широкий спектр застосування рішень завдання упаковки в різних сферах економічної діяльності робить завдання вдосконалення існуючих евристичних алгоритмів оптимізації рішення і створення нових ефективних алгоритмів конструювання упаковки актуальною[3].

В загальному вигляді задача двомірної ортогональної упаковки є окремим випадком задачі ортогональної упаковки розмірності D, що описується наступним чином: мається набір прямокутних контейнерів з габаритами {*W1J, W2J,…WDJ*} , j = *1,…,N* і набір прямокутних предметів {*w1i, wi2,…, wiD}, i = 1,…,n.* Тоді записуємо розташування об’єкта s в j-м контейнері в наступному вигляді : (x1ij; xij2; …xDij) .

Необхідно розташувати всі об’єкти в мінімальній кількості контейнерів при виконанні наступних умов:

* Ребра прямокутних об’єктів , що упаковані в контейнері, мають бути паралельні ребрам контейнера;
* Упаковані об’єкти не перетинаються один з одним;
* Упаковані об’єкти не перетинаються зі сторонами контейнера.

Вирішення будь-якої задачі пакування кодується стрічкою натуральних чисел довжиною *2n* :

*S = {A1, B1, A2, B2, …, AN, BN },*

де число *Ai* містить номер типу i-го об’єкту, що розташовується зі списку об’єктів; число *Bi* – інформацію щодо орієнтації цього об’єкту у просторі контейнеру. Цільовою функцією є відношення сумарного об’єму розміщених у контейнері об’єктів до об’єму *V\** описаного вокруг отриманої *D* – мірної упаковки контейнера.

Вхідними даними для розв’язування задачі розміщення довільних об’єктів на площині є масив лекал, які необхідно розмістити, і масив контурів, в яких потрібно розташувати лекала. Самі контури можуть бути як прямокутної, так і довільної форми, тобто матеріал міг використовуватися в попередніх задачах і залишився у відходах. Так само і лекала – можуть мати правильну і неправильну форму. Насамперед необхідно визначити, які лекала можна буде розташовувати і в якому контурі. У випадку, коли у деякому контурі неможливо розмістити жодне лекало, контур автоматично потрапляє у відходи, і ми його видаляємо з масиву контурів.

Для роботи генетичного алгоритму (ГА) насамперед нам потрібно визначити, за якими критеріями необхідно вибирати лекала і контури, в які вони будуть розміщуватися. Оскільки наперед невідомо, якої форми в нас будуть лекала і контури площин, на які ми будем розміщувати лекала, необхідно розглядати різні критерії. Спочатку треба розглядати критерії, які будуть опиратися на площу і форму лекал, хоча не треба відкидати і інші можливі критерії.

Опис реалізованого генетичного алгоритму:

Крок 1. Початок роботи алгоритму.

Крок 2. Отримання вхідних даних.

Отримується масив лекал, які необхідно розмістити, і масив контурів, в яких розміщуються лекала. Дані передаються з інтерактивної підсистеми.

Крок 3. Розраховуємо, які лекала потрапляють до певних контурів.

Аналізуємо, які лекала потрапляють до певних контурів і складаємо список. Якщо в певний контур неможливо розмістити жодного лекала – цей контур автоматично потрапляє у відходи, і ми його видаляємо з масиву контурів.

Крок 4. Для кожного генома проводимо ітерацію.

Для всіх можливих варіантів наборів критеріїв (генів) проводимо ітерацію (розташовуємо лекала).

Крок 5. Розраховуємо площу відходів і вибираємо m% найкращих результатів.

Для кожного можливого розташування лекала обчислюємо площу контурів, в які вже не потрапить жодне лекало, тобто ті контури, які підуть у відходи. Після цього з отриманих результатів вибираємо m% результатів (селекціонуємо найкращі гени), де площа відходів найменша, враховуючи результати, які наближаються до тих m%.

Крок 6. Розраховуємо, які лекала потрапляють до нових контурів.

Аналізуємо, які лекала потрапляють до нових контурів і складаємо новий список лекал.

Крок 7. Чи є для утворених нових контурів лекала?

Якщо в нових контурах не можливо розмістити жодного лекала, переходим до кроку 8, в іншому випадку переходимо до кроку 9.

Крок 8. Утворені контури відносимо до відходів.

Контури потрапляють до відходів і ми їх видаляємо з масиву контурів.

Крок 9. Чи є ще лекала для розміщення?

Перевіряємо, чи є у вихідному масив ще не розміщені лекала. Якщо немає, переходимо до кроку 11. В іншому випадку переходимо до кроку 10.

Крок 10. Схрещуємо найкращі результати і отримуємо нові можливі геноми.

За допомогою генетичних операторів (кросинговеру, мутації та інших) схрещуємо отримані покоління для можливого зародження досконалішого гена. Надалі для виконання генетичних операцій не використовуємо тих критеріїв, які дали негативні результати. Отримуємо нові можливі геноми (нову популяцію). Переходимо до кроку 4.

Крок 11. Виведення результатів.

На цьому кроці виводимо такі результати:

* послідовність розташування лекал у контурах з координатами розміщення лекал ;
* процент відходів;
* які лекала не розміщені (якщо такі є).

Крок 12. Кінець роботи алгоритму.

У результаті роботи ГА отримаємо наступну діаграму розкрою (рис. 1)

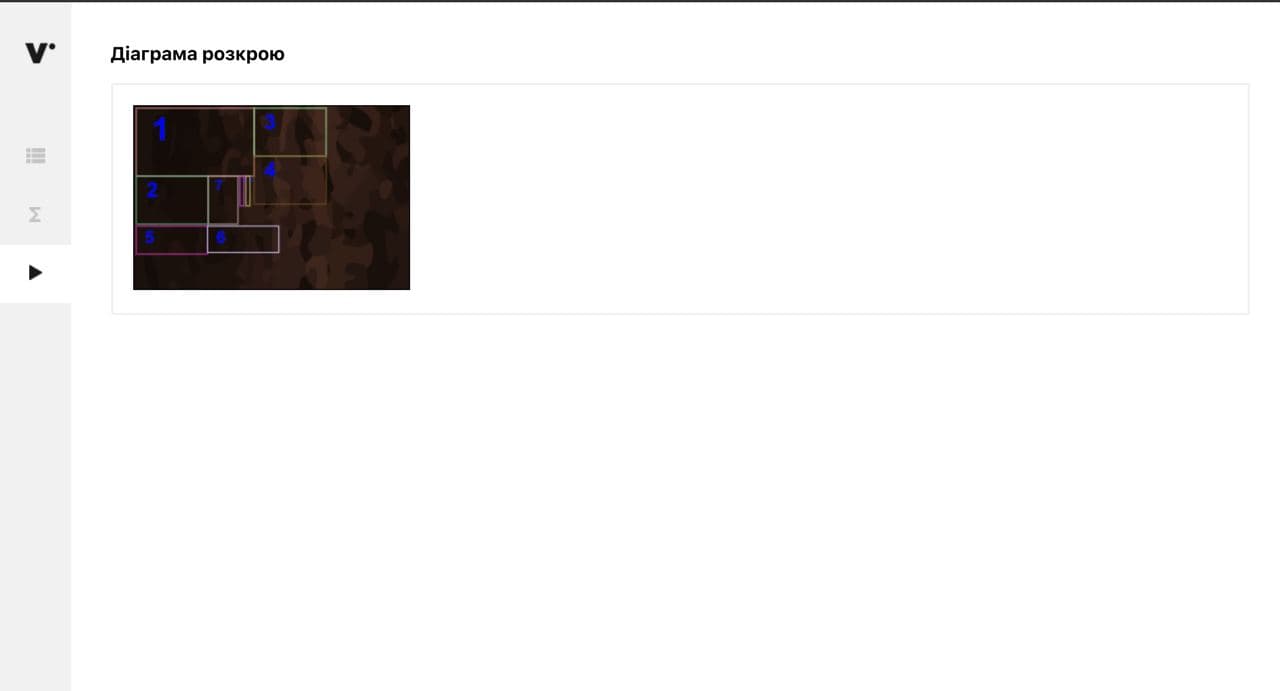


Рисунок 1. Діаграма розкрою комп’ютерного столу

Підтверджена ефективність застосування ГА при рішенні задач двомірної упаковки об’єктів. Визначені оптимальні параметри ГА для рішення задач двомірної упаковки об’єктів. Запропонований критерій останова ГА з проведенням зовнішнього контролю попадання поточної популяції до локального мінімуму.

ЛІТЕРАТУРА

1. *Chekanin V.A., Kovshov E.E., Hué N.N. Increasing the efficiency of evolutionary algorithms when solving optimization problems of packing objects. Chekanin, E.E. Kovshov, N.N. Hue // Control systems and information technologies. 2009. - No. 3. S. 63-67.*
2. *Скакаліна О.В. Інформаційні технології оптимізації управління складними територіально-розподіленими системами : монографія / Скакаліна О.В. - Полтава: ПолтНТУ, 2017.-223 c.*
3. *Bortfeldt A., Wascher G. Constraints in container loading: a state-of-the-art review.*

*European Journal of Operational Research. 2013. V. 229.*