УДК 519.8

В.Б. Крижанівський, к.ф.-м.н., доц.
Г.П. Шавурська, магістр
Житомирський державний технологічний університет

ЕВОЛЮЦІЙНИЙ ПІДХІД ДО РОЗВ'ЯЗАННЯ ЗАДАЧІ УПАКОВКИ ТРИВИМІРНИХ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ

Розглянуто проблему оптимального розміщення тривимірних геометричних об'єктів. Положення геометричних об'єктів визначається за допомогою розв'язання задачі лінійного програмування. Задачі лінійного програмування перебираються за допомогою генетичного алгоритму. Розроблено та реалізовано програмний продукт, який дає можливість досліджувати поведінку генетичних алгоритмів залежно від розміру популяції, оператору відбору батьківської пари, типу схрещування, оператору вибору особин у нову популяцію тощо.

Постановка проблеми. Актуальність дослідження та стан розробленості проблематики. Взагалі, задачі розкрою та упаковки (cutting and packing problems) полягають у відшуканні раціонального розміщення в деякому об'ємі певного набору об'єктів [1]. З практичної точки зору це і лазерна 3D різка; і завантаження контейнерів у морських, авіа та інших перевезеннях; і компоновочні задачі при проектуванні будівель, комп'ютерів, суден; і задачі розкрою в швейній промисловості та машинобудуванні тощо. В кожному конкретному випадку математична постановка має свої особливості. Це, в першу чергу, стосується функції цілі та обмежень, які накладаються на можливі положення об'єктів. Наприклад, попарний неперетин об'єктів та невихід за задані межі. Критерієм якості розміщення може бути, зокрема, коефіцієнт заповнення заданого об'єму. В крупносерійному виробництві за рахунок розв'язання подібних задач досягається значна економія матеріалів.

Складність задач і методи для їх розв'язання залежать від виду об'єктів і обмежень, які накладаються. Наприклад, задача про упаковку заданого набору паралелепіпедів в контейнери полягає у мінімізації кількості зайнятих контейнерів [2]. Різновиди цієї задачі виникають у різних галузях, крім, власне, розміщення в контейнерах. Це і завантаження вантажівок із обмеженням по вазі, і створення резервних копій на зйомних носіях і, навіть, оптимізація митних платежів (http://tamsbor.far.ru). Оскільки дана задача є NP-складною, то для розв'язання використовуються евристичні та метаевристичні алгоритми [3, 4], а також методи штучного інтелекту, наприклад нейронні мережі [5].

В зв'язку зі своєю практичною значущістю, велику увагу серед задач упаковки привертають задачі упаковки довільних багатогранників. Зрозуміло, що задача упаковки паралелепіпедів є частковим випадком задачі упаковки багатогранників. Тому складність розв'язання задачі упаковки багатогранників є, в усякому разі, не меншою, ніж складність задачі упаковки паралелепіпедів. Серед підходів до розв'язання можна виділити два напрямки. Перший ґрунтується на еволюційних алгоритмах та моделювання відпалу [6, 7, 8, 9], а другий на розробці зручної математичної моделі, її дослідженні, створенні алгоритмів для пошуку локальних оптимумів та пропонуванні методів їх ефективного перебору [10, 11, 12].

На основі результатів, які отримані в науковій школі чл.-кор. НАН України Стояна Ю.Г. [12], у даній роботі пропонується гібридний генетичний алгоритм для пошуку глобального оптимуму задачі упаковки опуклих багатогранників у напівнескінченній призмі. Для невеликої кількості об'єктів глобальний оптимум можна знайти методом «грубої сили». Але в практичних задачах кількість об'єктів є такою, що подібні техніки є неефективними. Оскільки час роботи таких алгоритмів росте експоненціально залежно від кількості об'єктів, тому в даному напрямку пропонується використовувати евристичні стратегії пошуку. Наприклад, генетичні алгоритми. Вони забезпечують задовільний результат за прийнятний час.

Викладення основного матеріалу. Змістовна постановка завдання. Нехай маємо n опуклих багатогранників і паралелепіпед, один з розмірів якого, наприклад, висота є змінною величиною. Необхідно розмістити усі n багатогранників у паралелепіпеді таким чином, щоб висота його набула найменшого значення. На можливі положення багатогранників накладаються природні обмеження, що полягають у тому, що багатогранники попарно не перетинаються і не виходять за межі паралелепіпеда.

Математична постановка завдання. Розглянемо тривимірний дійсний арифметичний простір R^3. Тоді паралелепіпед P, в якому розміщуються багатогранники подамо таким чином:

$$P = \{(x, y, z) \in R^3 : 0 \le x \le l, 0 \le y \le \omega, 0 \le z \le h\}, \tag{1}$$

де l, ω – константи; h – змінна.

Багатогранники $P_i, i = 1,...,n$ задамо координатами вершин $v_{ik} = (x_{ik}, y_{ik}, z_{ik}), k = 1,...,n_i$. Тоді точки багатогранника – це точки опуклої оболонки його вершин:

$$P_i = \{v \in R^3 : v = \sum_{k=1}^{n_i} \lambda_k v_{ik}, \sum_{k=1}^{n_i} \lambda_k = 1, \lambda_k \ge 0, k = 1,...,n_i\}. \tag{2}$$

Трансляцію P_i на вектор $u_i = (x_i, y_i, z_i)$ позначимо так:

$$P_i(u_i) = \{v \in R^3 : v = u_i + \bar{v}, \bar{v} \in P_i\}. \tag{3}$$

Можливість обертання багатогранників не розглядається. Тому вектор u_i будемо називати вектором параметрів розміщення багатогранника P_i. Отже тепер може бути сформульована <u>Задача оптимізації</u>. Знайти вектори параметрів розміщення багатогранників $u_1, u_2,..., u_n$ такі, що усі багатогранники $P_i, i = 1,...,n$ знаходяться в P, попарно не перетинаються і h досягає мінімального значення.

Умови знаходження багатогранників у паралелепіпеді та їх попарний неперетин визначають множину $\Omega \subset R^{3n+1}$ припустимих значень вектора змінних $(u_1, u_2,..., u_n, h)$ поставленої задачі. Для формалізації даних умов зручно скористатися апаратом Φ-функцій [13]. В результаті множину Ω можна подати у вигляді об'єднання поліедрів:

$$\Omega = \bigcup_{e=1}^{M} \Omega_e. \tag{4}$$

Введемо множину індексів S для усіх можливих невпорядкованих пар багатогранників $P_i, i = 1,...,n$:

$$S = \{1, 2,..., n(n-1)/2\}. \tag{5}$$

Тоді кожний поліедр Ω_e в представленні (4) можна записати у вигляді системи лінійних нерівностей:

$$\begin{cases} a_{sp}^T (u_i - u_j) - b_{sp} \ge 0, s \in S, 1 \le i < j \le n, \\ x_i + x_i^- \ge 0, \\ x_i + x_i^+ \le l, \\ y_i + y_i^- \ge 0, i = 1, \ldots, n, \\ y_i + y_i^+ \le \omega, \\ z_i + z_i^- \ge 0, \\ z_i + z_i^+ \le h, \end{cases} , \tag{6}$$

де p – індекс грані f_p будь-якого з об'єктів пари $\{i, j\}$; a_{sp}^T – транспонований вектор зовнішньої нормалі до f_p, $x_i^- = \min\{x_{ik} \mid k = 1, \ldots, n_i\}$, $x_i^+ = \max\{x_{ik} \mid k = 1, \ldots, n_i\}$, $y_i^- = \min\{y_{ik} \mid k = 1, \ldots, n_i\}$, $y_i^+ = \max\{y_{ik} \mid k = 1, \ldots, n_i\}$, $z_i^- = \min\{z_{ik} \mid k = 1, \ldots, n_i\}$, $z_i^+ = \max\{z_{ik} \mid k = 1, \ldots, n_i\}$; b_{sp} – вільний член, що визначається з геометричних розмірів об'єктів пари $\{i, j\}$.

Система лінійних нерівностей (6) разом із функцією цілі $h \to \min$ формують задачу лінійного програмування (ЗЛП). Кожна ЗЛП відрізняється від іншої обраними гранями f_p для кожної пари багатогранників. Отже, для того щоб розв'язати поставлену задачу оптимізації слід розв'язати усі задачі лінійного програмування. Найкращий з розв'язків буде глобальним оптимумом поставленої оптимізаційної задачі розміщення багатогранників. Очевидно, що кількість таких ЗЛП навіть для невеликої кількості багатогранників буде астрономічно великою. Наприклад, підрахуємо кількість ЗЛП, які необхідно розв'язати для розміщення двох тетраедрів (P_1, P_2) та одного куба (P_3). Невпорядковані пари, які з них можна утворити: $\{P_1, P_2\}$, $\{P_1, P_3\}$, $\{P_2, P_3\}$. Для кожної пари сумарні кількості граней складають 8, 10, 10 відповідно. Отже кількість ЗЛП виду (6), які слід розв'язати, дорівнює 800. А для трьох ікосаедрів – 64000 ЗЛП.

На сьогоднішній день не існує підходів, що дозволяють отримати точний розв'язок поставленої задачі. Тому пропонуються і досліджуються різні евристичні техніки. Однією з ефективних евристичних технік є генетичні алгоритми. Саме цей підхід і застосовується в даній роботі до розв'язання поставленої задачі.

Пропонований алгоритм. Для опису алгоритму введемо поняття індивіду, функції фітнесу та генетичних операторів.

Представлення індивіду. Кожен індивід – це задача лінійного програмування (6), що кодується у вигляді цілочисельного вектора довжиною $|S|$, де S – множина пар багатогранників (5). Компоненти даного вектора містять номери граней відповідної пари при наскрізній нумерації граней для кожної пари.

Для подання багатогранника скористаємось таким представленням [14]:

1. Множина граней :: множина елементів типу грань.
2. Множина ребер :: множина елементів типу ребро.
3. Функція R3 (вхід : вершина) : точка дійсного арифметичного простору R^3.

Грань = множина елементів типу ребро.

Ребро = запис (початок, кінець : вершина).

Вершина = Z^+; // Z^+ – цілі додатні числа.

Точка R^3 = запис (x, y, z : число).

Для прикладу запишемо в даній нотації чотиригранник з множиною вершин $V = \{1, 2, 3, 4\}$.

1. Множина граней = {
{{1, 4}, {4, 2}, {1, 2}}, // грань 1,
{{1, 2}, {2, 3}, {1, 3}}, // грань 2,
{{2, 4}, {4, 3}, {2, 3}}, // грань 3,
{{1, 4}, {4, 3}, {1, 3}}, // грань 4,
}
2. Множина ребер = {
{1, 2}, {2, 3}, {1, 3}, {1, 4}, {4, 3}, {2, 4}
}
3. Функція R3 (вхід : вершина)

Аргумент (вершина)	*Значення функції R3*
1	(1, 0, 0)
2	(0, 0, 0)
3	(0, 2, 0)
4	(0, 0, 3)

Тоді, наприклад, для розміщення трьох чотиригранників індивід формується таким чином. Множина індексів пар чотиригранників $S = \{1, 2, 3\}$, де 1 індексує пару {1, 2}, 2 індексує пару {1, 3}, а 3 пару {2, 3}. Для кожної пари вводимо наскрізну індексацію граней об'єктів. Для двох чотиригранників індексуємо вісім граней: {1, 2, 3, 4, 5, 6, 7, 8}. Тоді, наприклад, цілочисельний вектор (3, 2, 4) кодує ЗЛП для розміщення трьох об'єктів, де 3 – кодує третю грань пари {1, 2}, 2 – кодує другу грань пари {1, 3} та 4 – кодує четверту грань пари {2, 3}. Задача лінійного програмування (6) в даному випадку набуває вигляду:

$$\begin{cases} a_{13}^T(u_1 - u_2) - b_{13} \ge 0, \\ a_{22}^T(u_1 - u_3) - b_{22} \ge 0, \\ a_{34}^T(u_2 - u_3) - b_{34} \ge 0, \\ x_i + x_i^- \ge 0, \\ x_i + x_i^+ \le l, \\ y_i + y_i^- \ge 0, i = 1, 2, 3. \\ y_i + y_i^+ \le \omega, \\ z_i + z_i^- \ge 0, \\ z_i + z_i^+ \le h, \end{cases}$$

Обчислення функції фітнесу. Фактично, обчислення функції фітнесу полягає у розв'язанні задачі лінійного програмування (6). Але слід зазначити, що кількість нерівностей в ЗЛП (6) росте залежно від кількості багатогранників n за формулою $(n^2 - n)/2 + 6n$. У той же час кількість стовпчиків ЗЛП росте лінійно: $3n + 1$. Тому в даному випадку зручніше перейти до ЗЛП, яка є двоїстою до ЗЛП виду (6). Отриману двоїсту ЗЛП зручно розв'язувати модифікованим симплекс методом [15]. У результаті на кожній ітерації доведеться перераховувати лише матрицю CARRY розмірністю $(3n+2)\times(3n+2)$.

Генетичні оператори. Основними генетичними операторами в даній роботі є кросинговер, мутація та оператор вибору батьківської пари. Принцип роботи одноточкового кросинговеру в даній задачі полягає в наступному. Нехай обрано батьківську пару індивідів $A = (a_1, a_2, ..., a_{|S|})$ та $B = (b_1, b_2, ..., b_{|S|})$. Випадковим чином визначаємо ціле число в межах від 1 до $|S| - 1$. Дане число визначає точку розділення індивідів на частини для обміну частинами між батьками. Нехай випадкове число дорівнює i, тоді нащадки мають вигляд $D_1 = (a_1, a_2, ..., a_i, b_{i+1}, \ldots, b_{|S|})$, $D_2 = (b_1, b_2, ..., b_i, a_{i+1}, \ldots, a_{|S|})$. Також реалізовані двоточковий, однорідний кросинговери та кросинговер зі зменшенням заміни.

Особливість мутації в даній задачі полягає в тому, що індивід кодується не бінарним вектором, як у більшості генетичних алгоритмів, а цілочисельним. Причому кожна координата вектора взаємно однозначно відповідає парі багатогранників. Значення координати вектора індексує одну з граней пари багатогранників, щодо якої будується обмеження на неперетин. Отже, процедура мутації полягає у випадковому обранні координати вектора (гену) з наступним випадковим обранням його значення в межах від 1 до сумарної кількості граней пари об'єктів, яку кодує даний ген.

Алгоритм.

1. Випадковим чином створити початкову популяцію $P(0)$.
2. **While** (*Умова зупинки == false*) **do.**
3. $t = 0$.
4. $O = \varnothing$; // O – множина нащадків.
5. **For** k = 1 **to** *Розмір Популяції* **do.**
6. p_1 = Обрати ($P(t)$, оператор вибору); // Одним з операторів вибору (панміксія, аутбридинг, інбридинг тощо) обрати одного індивіда з популяції.
7. p_2 = Обрати ($P(t)$, оператор вибору); // Обрати другого індивіда.
8. Кросинговер (p_1, p_2, o_1, o_2); // Процедура кросинговеру для батьківської пари p_1 та p_2, яка повертає нащадків o_1, o_2.
9. Додати (O, o_1, o_2); // Помістити обох нащадків у множину O.
10. **End For.**
11. Мутація (O); // Випадковим чином із заданою ймовірністю провести мутацію нащадків.
12. $t = t + 1$.
13. $P(t)$ = Відбір (O, оператор відбору); //Одним з операторів відбору (витісненням, випадково) відібрати індивідів у нову популяцію.
14. **End While.**

Програмна реалізація. Концептуально програма поділяється на два модулі: модуль взаємодії з користувачем «Модуль інтерфейсу» та модуль розрахунків «Функціональний модуль». Взаємодія модулів представлена на рисунку 1.

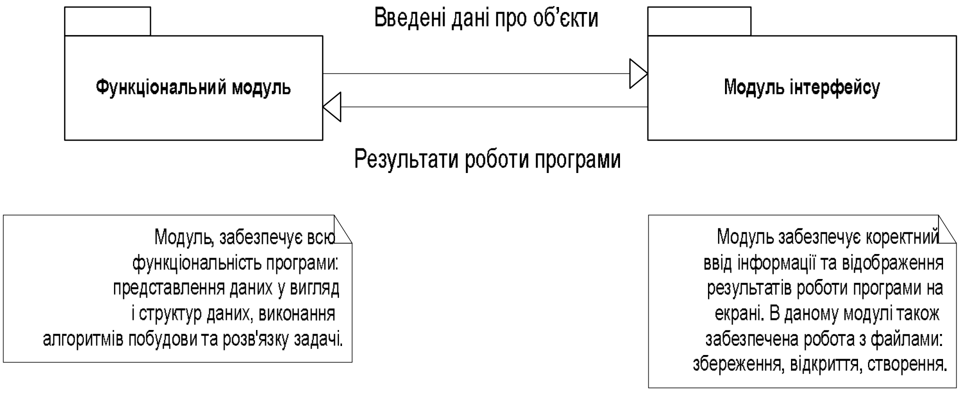

Рис. 1. Взаємодія модулів програми

Модулі представлені класами, що до них належать. Їх робота виконується в двох асинхронних потоках. Взаємодія між модулями забезпечується системою подій.

На діаграмі компонентів (рис. 2) наведено залежності внутрішніх і зовнішніх просторів імен програми.

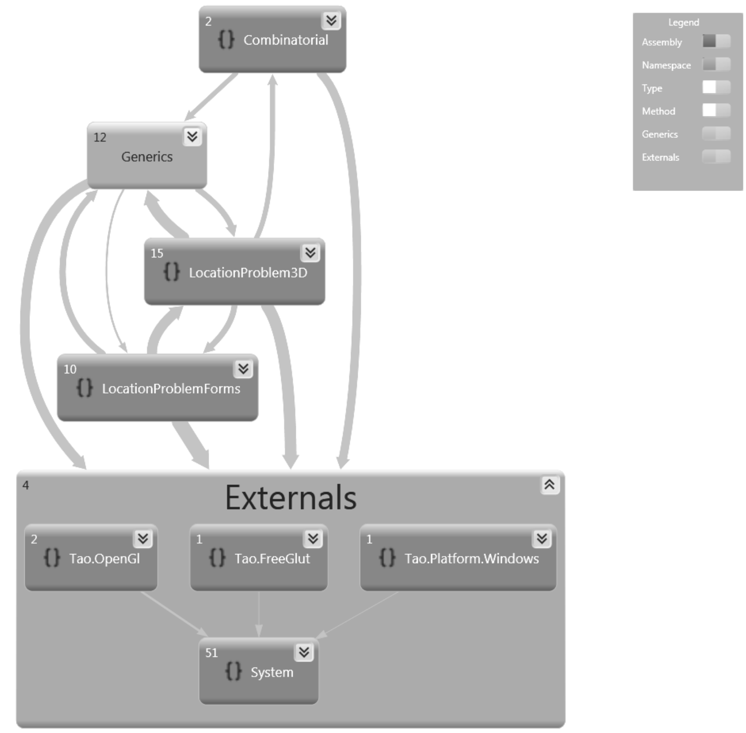

Рис. 2. Діаграма компонентів у розрізі просторів імен

Внутрішні простори імен. LocationProblem3D – простір імен, в якому описані класи математичної (функціональної) частини програми. LocationProblemForms – простір імен, що містить класи взаємозв’язку програми з користувачем. Generics – дозволяють визначати типізовані структури даних без прив’язки до стандартних типів даних.

Зовнішні простори імен. Tao.OpenGL – відповідає за реалізацію бібліотеки OpenGL, що використана для візуалізації тривимірних сцен із простих примітивів. Tao.FreeGlut – відповідає за реалізацію бібліотеки утиліт Glut, що відповідає за системний рівень операцій вводу-виводу при роботі з операційною системою, була використана для управління вікном візуалізації, для виведення літер.

Tao.Platform.Windows – відповідає за підтримку елементів візуалізації безпосередньо на платформі Windows.

Коротка характеристика класів програми. OrderedCollection – впорядкована множина. Відповідає за зберігання координат кожної з вершин об'єкта. Figure3D – зберігає список координат вершин об'єкта, списки вершин, що входять до кожної з граней об'єкта, а також його ім'я. Становить інтерфейс для роботи з фігурою: ініціалізація, копіювання, перевірку на опуклість, визначення крайніх точок фігури. FigureList – містить список фігур. Забезпечує додавання та видалення фігур. CombinatorialBase, Combinations, Calculations – допоміжні класи, що реалізують функціональність для генерації векторів розміщень, знаходження кількості можливих розміщень, факторіалу числа та ін. LocationProblem – клас містить усі необхідні дані для побудови та розв'язку задачі розміщення. Основними методами є: побудова обмежень на невихід та на неперетин, побудова двоїстої задачі, методи, що забезпечують роботу модифікованого симплекс-алгоритму. ExchaustiveSearch – повний перебір усіх можливих розміщень. GeneticAlgorithm – клас, що містить перелік операторів генетичного алгоритму, дозволяє, залежно від отриманих параметрів, виконувати той чи інший тип оператору ГА. Individual – клас кодує особину в термінах генетичного алгоритму. ResultEventArgs – тип, об'єкт якого використовує подія-повідомлення про кращий розв'язок.

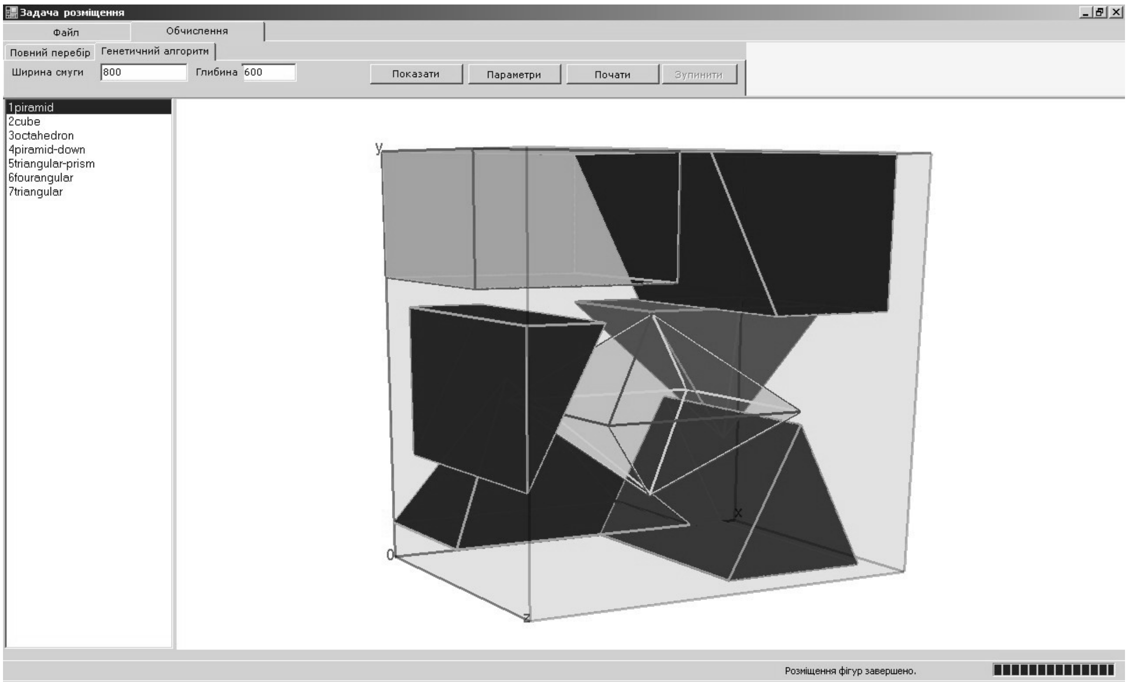

Рис. 3. Результат роботи програми з рекордним значенням функції фітнесу

Обчислювальний експеримент. Особливістю генетичних алгоритмів є те, що вони мають багато параметрів для налаштування, що впливають на якість отриманих рішень. Для дослідження даних впливів проведено ряд обчислювальних експериментів. Головна увага приділена залежностям якості рішення від:

- розміру популяції;
- кількості поколінь.

Також важливим є з'ясування наявності ефекту успадкування при схрещуванні в популяціях індивідів.

Для проведення експерименту було обрано 7 опуклих багатогранників $P_i, i=1,\ldots,7$ з такими кількостями граней: P_1 – 5 шт., P_2 – 6 шт., P_3 – 8 шт., P_4 – 5 шт., P_5 – 5шт., P_6 – 6 шт., P_7 – 5 шт. Параметри паралелепіпеду, в якому розміщуються багатогранники: l = 600, ω = 800. В даному випадку загальна кількість ЗЛП, які слід розв'язати, щоб знайти глобальний оптимум повним перебором складає порядку 10^{21}. Це зайвий раз підтверджує необхідність розробки підходів, здатних отримувати прийнятні розв'язки для задач подібної розмірності.

Незмінні в рамках експерименту параметри генетичного алгоритму наведені в таблиці 1. На рисунку 4 показані залежності значення функції фітнесу від кількості поколінь при незмінному значенні розміру популяції, яка складалася з 2000 індивідів, та від розміру популяції при незмінній кількості поколінь, що дорівнювала 10. Кожне значення функції фітнесу є усередненням по 20 спробам запуску генетичного алгоритму. З графіків видно стійку тенденцію до поліпшення розв'язку при збільшенні кількості поколінь. Це фактично є експериментальним підтвердженням наявності властивості успадкування індивідами потрібних властивостей батьків. В контексті задачі розміщення це означає, що отримані від «хороших» батьківських ЗЛП нащадки, також в середньому дають «хороші» розв'язки. На рисунку 3 наведено розміщення багатогранників, яке дає рекордне значення функції фітнесу 1356. Це значення отримане при найбільшій кількості поколінь.

Таблиця 1

Параметри ГА, які не змінюються під час експерименту

Назва	Значення
Тип схрещування	Двоточковий
Оператор відбору батьківської пари	Аутбридинг за генотипом
Ймовірність мутації	0,03
Кількість елітних осіб відносно розміру популяції	5 %
Оператор відбору в нову популяцію	Витісненням

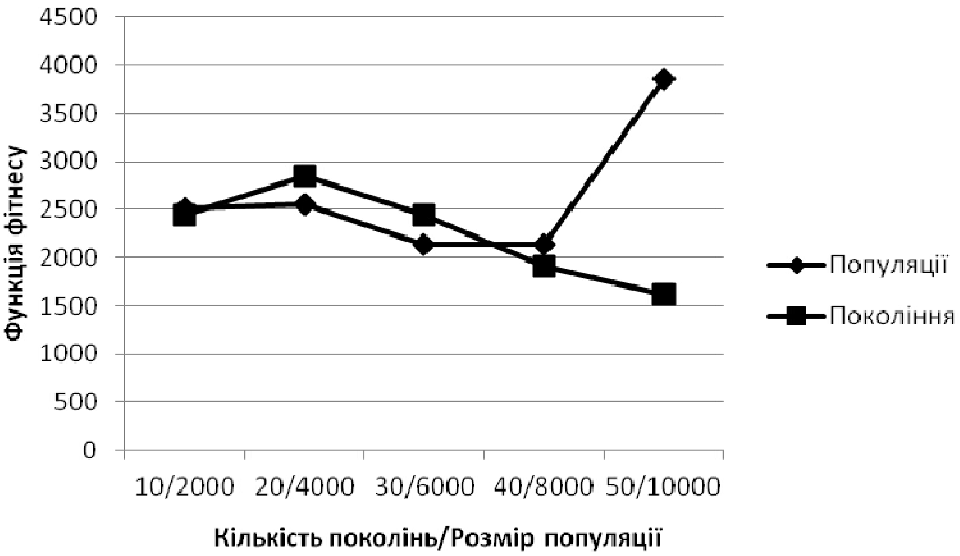

Рис. 4. Залежності якості розміщення від кількості поколінь та розміру популяції

Висновок. В роботі досліджується ідея використання ГА для розв'язання задач 3D упаковки довільних багатогранників. Запропоновано адаптацію операторів ГА для задач упаковки. Успішно застосовані властивості двоїстих задач та оптимізовано затрати часу та пам'яті за рахунок використання модифікованого симплекс-методу для обчислення функції фітнесу. Створено програмний продукт для розв'язання задач упаковки тривимірних геометричних об'єктів з використанням розроблених підходів. Проведено ряд обчислювальних експериментів для порівняння різних операторів генетичного алгоритму для дослідження впливу параметрів генетичного алгоритму на розв'язок задачі. З'ясовано, наявність явища успадкування індивідами потрібних властивостей їх батьків. Показано перспективність застосування ГА для задач великої розмірності. Подальші дослідження в цій області можуть бути зосереджені в таких напрямках: застосування інших моделей для побудови ГА, поєднання генетичного алгоритму з іншими класичними методами пошуку, розроблення алгоритмів, здатних до самоадаптації. Крім того, генетичний алгоритм може бути розпаралелений для ефективнішого пошуку результатів.

Список використаної літератури:

1. Dyckhoff H. Typology of cutting and packing problem / H.Dyckhoff // European Journal of Operational Research. – 1990. – Vol. 44, № 2. – P. 145–159.
2. *Vazirani V.V.* Approximation Algorithms / *V.V. Vazirani.* – Berlin : Springer, 2003.

3. *Gehring P.* A genetic algorithm for solving the container loading problem / *P.Gehring, A.Bortfeld* // International Transaction in Operational Research. – 1997. – Vol. 4, № 5/6. – P. 401–418.
4. *George J.A.* Three-dimensional packing-solution approaches and heuristic development / *J.A. George, D.B. Robinson* // International Journal Production Research. – 1991. – Vol. 29. – P. 1673–1685.
5. *Wang Chunxi* Neural algorithms of two-dimensional packing / *Wang Chunxi, Cao Yuedong, Zha Jianzhong* // Intelligent Control and Automation, 2000 : proceedings of the 3rd World Congress. – 2000. – Vol. 2. – P. 1127–1131.
6. *Dowsland W.B.* Three-dimensional packing-solution approaches and heuristic development / *W.B. Dowsland* // International Journal Production Research. – 1991. – Vol. 29. – P. 1673–1685.
7. *Hopper E.K.* Application of Genetic Algorithms to Packing Problems. A Review / *E.K. Hopper, B.Turton* // Proceedings of the 2nd On-line World Conference on Soft Computing in Engineering and Manufacturing. – 1997. – P. 278–288.
8. Concept for a genetic algorithm for packing 3D objects of complex shape / *I.Ikonen, W.E. Biles, A.Kumar at ol.* // Proceedings of 1st Online Workshop on Soft Computing. – Nagoya University, 1996. – P. 211–215.
9. *Cagan J.* A simulated annealing-based algorithm using hierarchical models for general three-dimentional component layout / *J.Cagan, D.Degentesh, S.Yin* // Computer – Aided Design. – 1998. – Vol. 30, № 10. – P. 781–791.
10. Packing of convex polytopes into a parallelepiped / *Yu. Stoyan, N.Gil, G.Schiethauer at ol.* // Preprint, Technical University of Dresden, MATH-NM-04-2004. – 2003. – 22 p.
11. *Stoyan Yu.* Packing cylinders and rectangular parallelepipeds with distances between them into a given region / *Yu.Stoyan, A.Chugay* // European Journal of Operational Research. – 2009. – № 197. – P. 446–455.
12. *Chernov N.* Mathematical model and efficient algorithms for object packing problem / *N.Chernov, Yu.Stoyan, T.Romanova* // Computational Geometry. – 2010. – № 43. – P. 535–553.
13. Construction of a Φ-function fot two convex polytopes / *Yu.Stoyan, M.Gil, T.Romanova at ol.* // Preprint, Technical University of Dresden, MATH-NM-13-2000. – 2000. – 23 p.
14. *Кушниренко А.Г.* Программирование для математиков / *А.Г. Кушниренко, Г.В. Лебедев.* – М. : Наука, 1988. – 384 с.
15. *Пападимитриу Х.* Комбинаторная оптимизация. Алгоритмы и сложность / *Х.Пападимитриу, К.Стайглиц.* – М. : Мир, 1984. – 512 с.

КРИЖАНІВСЬКИЙ В’ячеслав Борисович — кандидат фізико-математичних наук, доцент кафедри програмного забезпечення обчислювальної техніки Житомирського державного технологічного університету.

Наукові інтереси:

—комп’ютерне моделювання;

—технологія машинобудування.

ШАВУРСЬКА Ганна Павлівна – магістр, керівник групи в компанії по розробці програмного забезпечення QAP Int.

Наукові інтереси:

—математичне моделювання та обчислювальні методи в наукових дослідженнях;

—програмна інженерія.

Стаття надійшла до редакції 18.02.2013