

# ГЕНЕТИЧНИЙ АЛГОРИТМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

А. М. Господінов<sup>1</sup>, С. А. Смирнов<sup>1</sup>

<sup>1</sup> Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»,  
Фізико-технічний інститут

## Анотація

Досліджена робота генетичного алгоритму на прикладі задачі комівояжера. Запропоновано свій варіант налаштування алгоритму. Наведене порівняння ефективності розв'язку задачі комівояжера за допомогою генетичного алгоритму з відомими методами розв'язку на однакових наборах даних.

**Ключові слова:** генетичний алгоритм, оптимізація, селекція, схрещування

## Вступ

Генетичні алгоритми виникли в результаті спостереження і спроб копіювання природних процесів, що відбуваються в світі живих організмів, зокрема, еволюції, пов'язаної з нею селекції (природного відбору) популяції живих істот. У даній роботі розглядаються генетичні алгоритми як інструмент оптимізації. Моделлю, що розглядається, була обрана задача комівояжера, оскільки проблематика NP-повних задач зараз як ніколи актуальна. Інструментарієм для практичного дослідження було обрано середовище MATLAB, так як воно має значну кількість вбудованих функцій і панелей інструментів для розв'язку завдань генетичного програмування.

## 1. Постановка задачі

Комівояжер - бродячий торговець. Задача комівояжера - важливе завдання транспортної логістики, галузі, що займається плануванням транспортних перевезень. Комівояжеру, щоб розпродати потрібні і не дуже потрібні в господарстві товари, потрібно об'їздити  $n$  пунктів і врешті-решт повернутися в початковий пункт. Потрібно визначити найбільш вигідний маршрут об'їзду. В якості міри вигідності маршруту (точніше кажучи, не вигідності) може служити сумарний час у дорозі, сумарна вартість дороги, або, в простому випадку, довжина маршруту.

## Математичне формулювання

Задачу можна представити у вигляді множини ребер  $V$ . Кожному ребру  $i, j$  співставляється двійкова змінна  $x_{ij} \in \{0, 1\}$ , яка дорівнює 1, якщо ребро належить маршруту, і 0 – у іншому випадку. Кожна вершина повинна сполучатися через пару ребер з рештою вершин, тобто, через вхідне і вихідне ребро:

$$\forall i \in V \sum_{j \in V \setminus \{i\}} x_{ij} = 2$$

Так, кожен вектор з елементами, рівними 0 і 1, визначає коректний маршрут, який є розв'язком задачі комівояжера:

$$\min \left\{ \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} \mid x_{\text{valid}}(1)(2), x_{ij} \in \{0, 1\} \right\}$$

## 2. Робота генетичного алгоритму

Генетичний алгоритм - евристичний алгоритм пошуку, який використовується для розв'язку задач оптимізації та моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Є різновидом еволюційних обчислень. Основною особливістю генетичного алгоритму є використання оператора «схрещування», який виробляє операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування в живій природі. Далі розберемо структуру роботи генетичного алгоритму.

### Початкова популяція

Початкова популяція генерується зазвичай випадково. Єдиний критерій – достатня різноманітність особин, щоб уникнути потрапляння популяції в найближчий локальний екстремум, що не є глобальним.

### Оцінка пристосованості

Оцінювання популяції необхідно для того, щоб виявити більш пристосованих і менш пристосованих особин. Для підрахунку пристосованості кожної особини використовується функція пристосованості (цільова функція)  $f_i = f(G_i)$ , де  $G_i = \{g_{ik} \mid k = 1, 2, \dots, N\}$  – хромосома  $i$ -ї особини,  $g_{ik}$  – значення  $k$ -го гена  $i$ -ї особини,  $N$  – кількість генів в хромосомі.

## Селекція

Селекція (відбір) необхідна, щоб вибрати більш пристосованих особин для схрещування. З існуючих варіантів селекції розглянемо рулеточну селекцію. В даному варіанті селекції ймовірність  $p_i$   $i$ -ї особини взяти участь в схрещуванні пропорційна значенню її пристосованості  $f_i$  і дорівнює  $p_i = \frac{f_i}{\sum_j f_j}$

## Схрещування

Відібрані в результаті селекції особини (батьківські) схрещуються і дають потомство. Хромосоми нащадків формуються в процесі обміну генетичною інформацією (із застосуванням оператора кросовера) між батьківськими особинами. Створені таким чином нащадки складають популяцію наступного покоління.

### 2.1. Налаштування алгоритму

Для отримання результатів був використаний пакет прикладних програм для вирішення задач технічних обчислень – MATLAB. Він надає вбудовану панель інструментів OptimToolbox, яка призначена для розширення функціональних можливостей пакета алгоритмами оптимізації, зокрема, генетичними алгоритмами.

#### Запропоновані налаштування

- На вході хромосома подана у вигляді бінарного вектора.
- Розмір популяції стандартний – 50 особин.
- Ймовірність кросинговеру – 0.8. Використовуємо метод генерації перестановок.
- Так як розмір популяції відносно невеликий, є ймовірність потрапляння в локальний мінімум. Тому приймаємо досить велике значення параметра мутації – 0.2.
- Нова популяція створюється після розрахунку функції пристосованості.
- Критерієм зупинки виконання алгоритму є незмінність значень функції пристосованості.

## Результати

Порівняємо ефективність розв'язку задачі комівояжера за допомогою генетичного алгоритму з ефективними методами вирішення на однакових наборах даних з критеріями часу та точності використання. В якості методу побудови мінімального кістяка був використаний алгоритм Прима.

Табл. 1. Порівняння точності розв'язку задачі комівояжера за допомогою генетичних алгоритмів з ефективними методами вирішення на однакових наборах даних (розмір мінімального за довжиною гамільтонового циклу в графі)

Кількість міст	Генетичний алогитм	Рішення за допомогою мінімального кістяка	Оптимізація рішення за допомогою мінімального кістяка
5	1.42	1.46	1.46
15	3.22	3.5	3.28
50	5.89	7.94	6.68
100	8.73	10.79	10.05
200	11.98	15.14	13.78
400	16.33	20.61	18.8

Табл. 2. Порівняння часу розв'язку задачі комівояжера за допомогою генетичних алгоритмів з ефективними методами розв'язання на однакових наборах даних (час в мс), процесор Intel(R) Core(TM) i5-3230M CPU @2.60GHz

Кількість міст	Генетичний алогитм	Рішення за допомогою мінімального кістяка	Оптимізація рішення за допомогою мінімального кістяка
5	82	15	15
15	153	32	36
50	543	95	102
100	2530	105	130
200	6735	953	1703
400	15847	2186	3873

## Висновки

Аналізуючи табл. 1 і 2, очевидно, що генетичний алгоритм знаходить більш точне рішення в порівнянні з рішенням задачі за допомогою мінімального кістяка навіть з оптимізацією. Однак при вирішенні задачі комівояжера за допомогою генетичного алгоритму при великій кількості міст (велика довжина хромосоми) виникли проблеми зі збіжністю, тому був збільшений параметр мутації; так само генетичний алгоритм має не дуже хороші показники по часу. На основі отриманих результатів, виділимо переваги і недоліки генетичних алгоритмів.

#### Переваги:

- Велика кількість вільних параметрів, що дозволяє ефективно вбудовувати евристики;

- Ефективне розпаралелювання;
- Ефективний в рішенні задач зі складною математичною моделлю.

#### Недоліки:

- Можливі складності в питанні збіжності;
- У простих цільових функціях генетичний алгоритм завжди програє по швидкості простим алгоритмам пошуку.

Специфікою алгоритму є те, що при великих вхідних даних (велика довжина хромосоми, велика популяція, тощо) бажане додаткове налаштування алгоритму для більш продуктивної роботи.

#### Перелік використаних джерел

1. Батищев Д. И., Неймарк Е. А., Старостин Н. В. Применение генетических алгоритмов к решению задач дискретной оптимизации — 3 изд. — 2007. — 85 с. — Режим доступа: <http://www.unn.ru/pages/e-library/aids/2007/15.pdf>.
2. Вороновский Г. К., и др. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности — 1997. — 112 с. — Режим доступа: <https://bookini.ru/geneticheskie-algoritmy-iskusstvennye-nejronnye-seti->
3. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы — М.: Горячая линия, 2006. 452 с. —
4. Панченко Т.В. Генетические алгоритмы: учебно-методическое пособие — 2007. — Режим доступа: [http://window.edu.ru/catalog/pdf2txt/394/39394/17112?p\\_page=1](http://window.edu.ru/catalog/pdf2txt/394/39394/17112?p_page=1)
5. De Jong, K.A. Introduction to the second special issue on genetic algorithms. — Machine Learning — p. 351-353.