

М.П. Пасинков, В.А. Турчина
Дніпропетровський національний університет ім. Олеся Гончара

ПРО ПАРАМЕТРИ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

Проведено ряд експериментів розв'язування задачі комівояжера за допомогою еволюційних алгоритмів.

Проведен ряд експериментів по решению задачи коммивояжера при помощи эволюционных алгоритмов.

A number of experiments to address the traveling salesman problem using evolutionary algorithms were run.

Ключові слова: генетичний алгоритм, задача комівояжера.

Вступ. Задача комівояжера є однією з класичних задач дискретної оптимізації. Простота її постановки приваблює дослідників, а належність її до класу NP-повних задач ставить перед ними ряд цікавих питань. Як оптимізаційна задача на графах, вона полягає у пошуку гамільтонового циклу найменшої довжини. До таких задач можуть зводитись проблеми пов'язані з розробкою логістичних маршрутів, комп'ютерних мереж, архітектури електронно-обчислювальних вузлів тощо. Для розв'язування задачі комівояжера виконано ряд експериментів із застосуванням еволюційних алгоритмів генетичного типу. Експерименти проводились із графами спеціального вигляду. Отримано оцінки ймовірності знаходження точного розв'язку залежно від розміру графу та критерію виходу з еволюційного процесу.

Постановка задачі. Агенту необхідно обійти n міст і повернутися у вихідний пункт відвідуючи кожне місто лише по одному разу та вибрати маршрут з мінімальною вартістю [6].

Дано: повнозв'язний граф $G = (V, U)$, де V – множина вершин графу, $|V| = n$ – кількість міст; U – множина ребер (характеризує шляхи між містами); матриця суміжності $R(i, j)$, де $i, j = 1, 2, \dots, n$, що характеризує відстані між містами.

Цільова функція для задачі має наступний вигляд:

$$Z(\varphi) = R(\varphi[n], \varphi[1]) + \sum_{i=1}^{n-1} R(\varphi[i], \varphi[i+1]) \rightarrow \min .$$

Класичні методи пошуку розв'язків. Класичним точним методом пошуку розв'язків задачі комівояжера є метод гілок та меж [3; 8], що полягає у побудові дерева пошуку розв'язків та підрахунку оцінок. Обчислювальна складність алгоритму гілок та меж росте з експоненціальною швидкістю. Оцінка складності методу гілок та меж є верхньою межею при проектуванні алгоритмів розв'язку задачі.

Існують наближені методи розв'язування поставленої задачі. Найвідомішими наближеними методами розв'язування є методи найближчого сусіднього міста, включення найближчого міста та метод найдешевшого включення. Алгоритми не є досконалими і не гарантують оптимальності маршруту.

Також для знаходження розв'язків застосовуються технології нейронних мереж, наприклад мережа Хопфілда [2; 7]. Широкого вжитку набули алгоритми на основі знаходження найменшого остовного дерева [5] та алгоритм імітації відпалу [4].

Генетичні алгоритми. Теорія еволюційних алгоритмів дала поштовх для створення нових підходів до розв'язування класичних задач. Не стала винятком і задача комівояжера. Існує безліч реалізацій генетичних алгоритмів та алгоритмів типу рою часток для згаданої задачі.

Для даної конкретної реалізації було використано OpenSource бібліотеку методів реалізації генетичних алгоритмів Genetic Algorithm Library Младена Янковіча [1].

Опис фреймворку. Використана бібліотека дозволяє конструювати генетичні алгоритми специфічного вигляду та виконувати обчислення у режимі кількох потоків команд.

Структура бібліотеки виділяє три великих блоки класів: класи для конструювання алгоритмів, класи хромосом та класи популяції.

До класу хромосоми включаються структури, що реалізують операції кросоверу і мутації, порівняльний оператор для фітнес-функцій та власне визначення вигляду хромосоми.

Клас популяції являє сукупність структур відбору, вибору особин для схрещування, формування нової популяції на наступному етапі еволюції.

Клас алгоритму включає критерій виходу з еволюційного процесу та власне алгоритм.

Бібліотека включає до себе не лише інтерфейси для конструювання алгоритму. Реалізовано також частовживані структури, такі як, наприклад, бінарна хромосома для класичного генетичного алгоритму (Binary-coded GA, BGA) та хромосома для генетичного алгоритму на дійсних числах (Real-coded GA, RGA); реалізовано ряд шаблонів для операцій кросоверу, мутації, шаблони поведінки, визначення часу, виходу з ітераційного процесу, селекції та формування популяції.

Опис алгоритму. Для пошуку оптимального маршруту задачі комівояжера використано генетичний алгоритм з критерієм виходу за незмінною впродовж деякої кількості епох найкращою хромосомою. Критерій відбору хромосом – менша фітнес функція.

Хромосома конструюється специфічним чином і являє собою маршрут комівояжера.

Для того, щоб уникнути некоректних маршрутів, що виникають із застосуванням класичних одноточкових чи багатоточкових кросоверів, що може суттєво вплинути на швидкість роботи алгоритму, використовується специфічно створений кросовер. Із хромосоми одного з батьків, вибирається випадковим чином місто і далі в хромосомах батьків вибирається те, яке є найближчим серед міст, пов'язаних з поточним. Якщо таке місто вже присутнє у маршруті-нащадку, то вибирається наступне, з найменшою відстанню. Якщо всі пов'язані міста вже присутні у маршруті, то вибирається випадкове місто, із тих, що до маршруту-нащадку ще не увійшли.

Алгоритм використовує оператор мутації, що міняє місцями міста у маршруті.

Розмір популяції 100 осіб. Вісім найкращих у кожній епосі потрапляють до наступної без операції відбору.

Результати експериментів. Було проведено ряд експериментів на графах специфічного вигляду. Граф формувався за мапою міст, що рівномірно розміщуються в квадраті. Відстань між сусідніми містами, що знаходяться на одній лінії (горизонтальній чи вертикальній) у квадраті складає дві одиниці виміру.

Таким чином, при парній кількості міст довжина оптимального маршруту складає $2 * n$ одиниць довжини, де n – кількість міст у заданій мапі. При непарній кількості міст оптимальна довжина маршруту визначається попередньо за допомогою методу гілок та меж.

Оскільки критерієм виходу з еволюційного процесу є досягнення певної кількості епох без зміни найкращого значення фітнес-функції, а відповідно і найкращої хромосоми, то необхідно визначитися з тою кількістю ітерацій, що мінімально необхідна для знаходження оптимального маршруту при заданій кількості міст.

Експеримент послідовно проводився для всіх квадратів починаючи від чотирьох міст і до ста. У циклі конструювалися еволюційні процеси з різними критеріями виходу. Виконувалась одна тисяча запусків алгоритму при кожній комбінації параметрів.

Після проведення експериментів для кожної кількості міст було побудовано графік на осі абсцис якого відкладається кількість епох, зазначена у критерії виходу, а на осі ординат – кількісний вираз відношення кількості успішних експериментів до неуспішних. Успішним називається експеримент, у ході якого було досягнуто оптимальної довжини маршруту для заданої кількості міст, неуспішним – експеримент за ходом якого було отримано довжину маршруту, що відрізняється від оптимальної.

Усі експерименти проводилися без використання багатопоточності.

Окрім зміни параметрів виходу з еволюційного процесу було також використано декілька конфігурацій оператора мутації. Для задачі комівояжера в даній конструкції алгоритму мутація має два основних параметри: ймовірність мутації та максимальна кількість позицій у маршруті, що можуть бути змінені. Наведено результати експериментів для ймовірності мутації 0.3 та максимальної кількості позицій – 2 (рис. 1) та параметрів 0.5 та 5 відповідно (рис. 2).

Результати експериментів показують, що із збільшенням кількості міст генетичний алгоритм із більшою ймовірністю мутації має гіршу збіжність і гірше відношення успішних експериментів до неуспішних. Окрім того збільшується час обчислень. Це відбувається за рахунок того, що часта мутація виводить популяцію за межі досягнутого мінімуму.

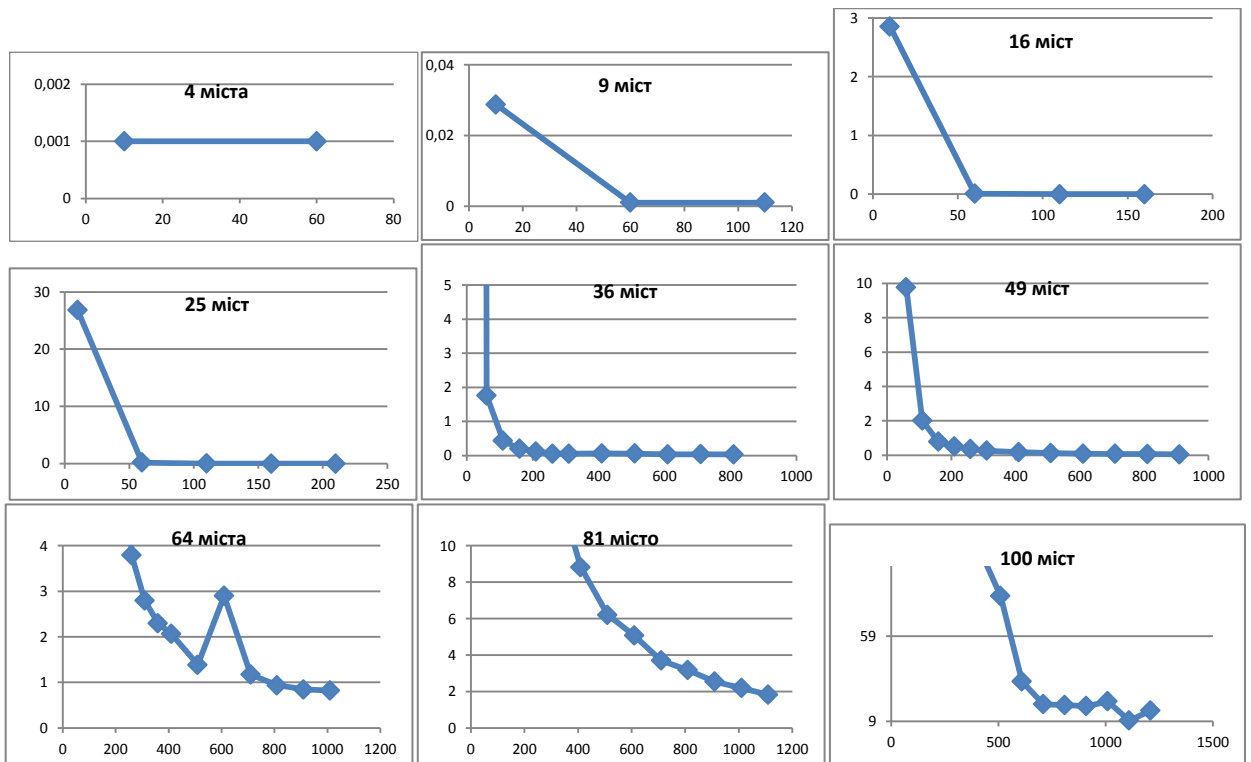


Рис. 1. Результати експерименту для параметрів мутації (0.3; 2)

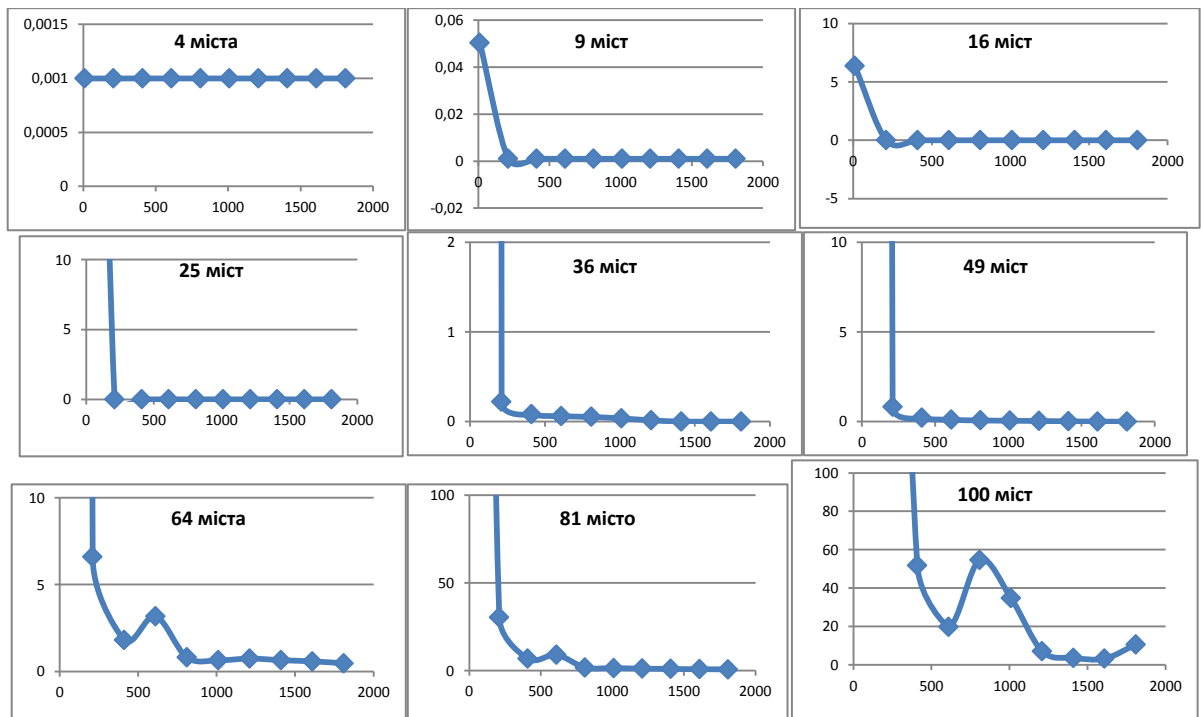


Рис. 2. Результати експерименту для параметрів мутації (0.5; 5)

Було проведено ще один експеримент: при зазначених вище параметрах оператора мутації було виконано вимірювання часу виконання еволюції для мап від чотирьох до 1225 міст. На графіку (рис. 3) по горизонтальній осі відкладено кількість міст, по вертикальній – час перебігу еволюції у секундах. Кількість ітерацій при незмінному найкращому значенні фітнес-функції вибрано 15000.

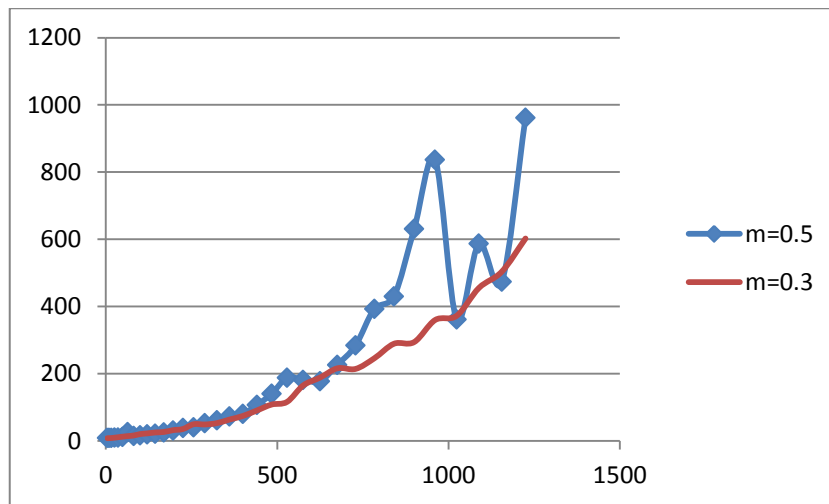


Рис. 3. Порівняння швидкості алгоритму при параметрах мутації (0.3; 2) та (0.5; 5)

Як результат, можна побачити, що при невеликій кількості міст час виконання еволюції майже не відрізняється, незалежно від параметрів мутації, проте із збільшенням кількості міст алгоритм із меншими параметрами мутації починає показувати кращий час.

Час перебігу еволюції з параметрами мутації (0.3; 2) складає від 8.192 сек. для чотирьох міст на мапі до 602.44 сек. для 1225 міст на мапі. Для параметрів (0.5; 5) аналогічні результати складають від 8.7 сек. до 962.047 сек.

Висновки. У генетичних алгоритмах при використанні критерію виходу з еволюційного процесу за кількістю ітерацій без зміни найкращого представника популяції, варто коригувати цей показник залежно від зміни кількості шуканих параметрів задачі, оскільки при збільшенні кількості параметрів кількість неоптимальних розв'язків збільшується.

Збалансований оператор мутації для задачі комівояжера дозволяє по-кращити швидкість збіжності алгоритму за рахунок відсутності виходів еволюційного процесу за межі глобального мінімуму.

Бібліографічні посилання

1. Genetic Algorithm Library by [Mladen Janković](http://www.codeproject.com/Articles/26203/Genetic-Algorithm-Library): <http://www.codeproject.com/Articles/26203/Genetic-Algorithm-Library>
2. **Hopfield J.J** and Tank D.W. Neural computation of decisions in optimization problems.
3. **Land, A.H.** und A.G. Doig (1960). An automatic method of solving discrete programming problems. In: *Econometrica* 28, S. 497-520
4. **Левитин А.В.** Глава 3. Метод грубой силы: Задача коммивояжера / А.В. Левитин // [Алгоритмы: введение в разработку и анализ](#). М., 2006 – С. 159-160.
5. **Мудров В.И.** Задача о коммивояжере / В.И. Мудров. – М., 1969. – С. 62.
6. **Кормен, Т.** Минимальные остовные деревья/ Т. [Кормен](#), Ч. [Лейзерсон](#). Р. [Ривест](#), К. [Штайн](#) // Алгоритмы: построение и анализ = Introduction to Algorithms – 2-е изд. – М., 2005. – 1296 с.
7. **Хайкин С.** Нейронные сети: полный курс. Издание 2. / С. Хайкин: пер. с англ. – М., 2006. – 1104 с.
8. **Кузнецов Ю.Н.** Математическое программирование: учебное пособие. 2-е изд. перераб. и доп./ Ю.Н. Кузнецов, В.И. Кузубов, А.Б. Волощенко. – М., 1980, 300 с.

Надійшла до редколегії 11.06.2012