

С. Д. ТОЧИЛІН

кандидат фізико-математичних наук, доцент,  
доцент кафедри інформаційної безпеки та наноелектроніки  
Національний університет «Запорізька політехніка»  
ORCID: 0000-0003-2010-6358

## КОМП'ЮТЕРНА ПРОГРАМА РІШЕННЯ РІЗНИХ ВАРІАНТІВ ЗАДАЧІ КОМІВОЯЖЕРА З ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ

У наш час інтенсивно розвивається транспортна логістика. Оптимальні маршрути руху транспортних засобів, для певного набору пунктів комунікації: транзитних, вихідного та кінцевого пункту, можуть бути знайдені при рішенні різних варіантів задачі комівояжера.

Задача комівояжера – класична задача оптимізації. У випадку її замкнутого варіанта, комівояжер, рухаючись із вихідного пункту (ВП), однократно відвідує всі транзитні пункти та вертається у ВП. У випадку незамкнутого варіанта, комівояжер рухається з ВП, однократно відвідує всі транзитні пункти та прибуває у кінцевий пункт.

Маршрути руху для замкнутого та незамкнутого варіанта задачі можуть бути визначені за допомогою комп'ютерних програм, які використовують для рішення генетичний алгоритм (ГА).

При застосуванні ГА для рішення задачі комівояжера, при великій кількості транзитних пунктів, можлива ситуація вибору оптимального маршруту з декількох, отриманих при різних параметрах генетичного алгоритму. У цьому випадку необхідно оперативна інтерактивна взаємодія із програмою для зміни параметрів ГА. Як відомо, інтерактивність мають комп'ютерні програми із графічним інтерфейсом користувача.

У роботі на мові програмування Java розроблена кросплатформна комп'ютерна програма рішення різних варіантів задачі комівояжера з допомогою генетичного алгоритму. Вона має графічний інтерфейс користувача та дозволяє визначити оптимальний маршрут руху, як для замкнутого, так і незамкнутого варіанту задачі комівояжера. У той час, графічний інтерфейс додатка дає можливість змінювати параметри ГА у процесі його роботи та відображати знайдені маршрути руху у графічному вигляді.

Попередньо дані про розташування пунктів руху, у вигляді їх назв та відповідних значень географічних координат широти та довготи, вводяться у програму з \*.csv файлу.

Програма при функціонуванні використовує генетичний алгоритм на основі методів, які для кожного варіанта задачі комівояжера реалізовували кросовер, мутацію хромосом, механізми елітизму та турнірного відбору особин, оцінку придатності популяції (методи реалізації кросовера і мутації хромосом для різних варіантів задачі відрізнялися). Для оцінки відстані між містами та довжини маршруту додаток використовує формулу заверсинусів.

**Ключові слова:** задача комівояжера, генетичний алгоритм, оптимальний маршрут, графічний інтерфейс користувача.

S. D. TOCHILIN

Candidate of Physical and Mathematical Sciences, Associate Professor,  
Associate Professor at the Department of Information Security  
and Nanoelectronics  
Zaporizhzhia Polytechnic National University  
ORCID: 0000-0003-2010-6358

## COMPUTER PROGRAM FOR SOLVING VARIOUS VERSIONS OF THE TRAVELING SALESMAN PROBLEM USING A GENETIC ALGORITHM

Currently, transport logistics is developing intensively. Optimal routes for vehicles for a certain set of communication points: transit, initial and final points, can be found by solving various versions of the traveling salesman problem.

The traveling salesman problem is a classical optimization problem. In the case of its closed version, the traveling salesman, moving from the initial point (IP), visits all transit points once and returns to the IP. In the case of an open version, the traveling salesman moves from the IP, visits all transit points once and arrives at the final point.

The routes for the closed and open versions of the problem can be determined using computer programs that use a genetic algorithm (GA) for solving.

When using GA to solve the traveling salesman problem, with a large number of transit points, a possible situation is choosing the optimal route from several obtained with different parameters of the genetic algorithm. In this case, prompt

interactive interaction with the program is necessary to change the GA parameters. As is known, computer programs with a graphical user interface have interactivity.

In this work, a cross-platform computer program for solving various versions of the traveling salesman problem using a genetic algorithm was developed in the Java programming language. It has a graphical user interface and allows you to determine the optimal route for both closed and open versions of the traveling salesman problem. At the same time, the graphical interface makes it possible to change the GA parameters in the process of working with the application and display the found routes in graphical form.

Previously, data on the location of the traffic points, in the form of their names and corresponding values of geographic coordinates of latitude and longitude, are entered into the program from a \*.csv file.

The program uses a genetic algorithm based on methods that implemented crossover, chromosome mutation, elitism and tournament selection mechanisms, and population fitness assessment for each version of the traveling salesman problem (the methods for implementing crossover and chromosome mutation of the population differed for different versions of the problem). The application uses the haversine formula to estimate the distance between cities and the length of the route.

**Key words:** traveling salesman problem, genetic algorithm, optimal route, graphical user interface.

### Постановка проблеми

У наш час інтенсивно розвивається транспортна логістика [1]. У цьому зв'язку актуальною задачею є розробка кросплатформних комп'ютерних програм, які автоматизують процес знаходження оптимальних транспортних маршрутів руху та мають графічний інтерфейс користувача (Graphical User Interface – GUI).

Оптимальні маршрути руху транспортних засобів, для певного набору пунктів комунікації: транзитних, вихідного і кінцевого пункту, можуть бути знайдені при рішенні різних варіантів задачі комівояжера. У випадку замкнутого варіанта (ЗВ), комівояжер, рухаючись із вихідного пункту (ВП), однократно відвідує всі транзитні пункти та вертається в ВП. У випадку незамкнутого варіанта (НВ), комівояжер рухається з ВП, однократно відвідує всі транзитні пункти і прибуває в кінцевий пункт.

Маршрути руху для ЗВ та НВ задачі можуть бути визначені за допомогою комп'ютерних програм, що використовують при рішенні генетичний алгоритм. При цьому актуальною проблемою є розробка комп'ютерних програм, які автоматизують процес знаходження оптимальних маршрутів комунікації на основі рішення задачі комівояжера за допомогою генетичного алгоритму та мають GUI.

### Аналіз останніх досягнень і публікацій

До теперішнього часу розроблена велика кількість комп'ютерних програм, які вирішують задачу комівояжера за допомогою генетичного алгоритму, наприклад додатки створені у роботах [2–5]. Програми у [2–3] не мають графічного інтерфейсу та знаходять рішення лише для замкнутого варіанту задачі комівояжера. У той час, тільки незамкнутий варіант задачі вирішують додатки створені у [4, 5], але лише програма роботи [5] має GUI.

### Формулювання мети дослідження

У роботі було поставлене завдання розробки кросплатформного Java-додатка з GUI з можливістю знаходження оптимального маршруту руху при рішенні, як замкнутого, так і незамкнутого варіанта задачі комівояжера за допомогою генетичного алгоритму.

### Викладення основного матеріалу дослідження

Для рішення поставленого завдання був створений Java-додаток ST\_Travelling із графічним інтерфейсом користувача.

При розробці програми використовувалася бібліотека jfreechart-1.0.19.jar, яка дозволяла відображати маршрути руху знайдені при роботі додатка у графічному вигляді. Процес компіляції та запуску програми здійснювався за допомогою, як архівних наборів класів і утиліт Java (наприклад, jdk-6u16), так і сучасних (наприклад, jdk-8.0.452.9-hotspot).

Програма при функціонуванні використовувала генетичний алгоритм на основі методів, які для кожного варіанта задачі комівояжера реалізовували кросовер, мутацію хромосом, механізми елітизму та турнірного відбору особин, оцінку придатності популяції (методи реалізації кросовера і мутації хромосом для різних варіантів задачі відрізнялися). При розрахунку довжини маршрутів, відстані між містами визначалися на основі значень їх широти та довготи за допомогою формули гаверсинусів [6].

На початку роботи із програмою, за допомогою меню «File», назви міст (пунктів маршруту) та їх географічні координати, завантажувалися з \*.csv файлу і заповнювали таблицю «Cities». При цьому географічне розташування пунктів руху, у зменшеному масштабі, відображалось у вікні програми у графічному вигляді.

Потім установлювалися параметри необхідні для роботи додатка. Значення, розташовувані в полях вводу «Population», «pCrossover», «pMutation», «Elite», «Competitors», «Ngen», ініціалізували розмір популяції, імовірності схрещування та мутації неелітних хромосом у поколінні, число елітних особин, кількість хромосом у турнірі відбору, число поколінь хромосом, відповідно. Прапорці «TSP» і «OTSP» дозволяли вибрати замкнутий або незамкнутий варіант рішення задачі комівояжера, відповідно.

Запуск програми здійснювався при натисканні на кнопку «Run». При цьому додаток розраховував та відображав у графічному вигляді маршрут руху. Крім того, розміщав його довжину у полі вводу «Distance, km».

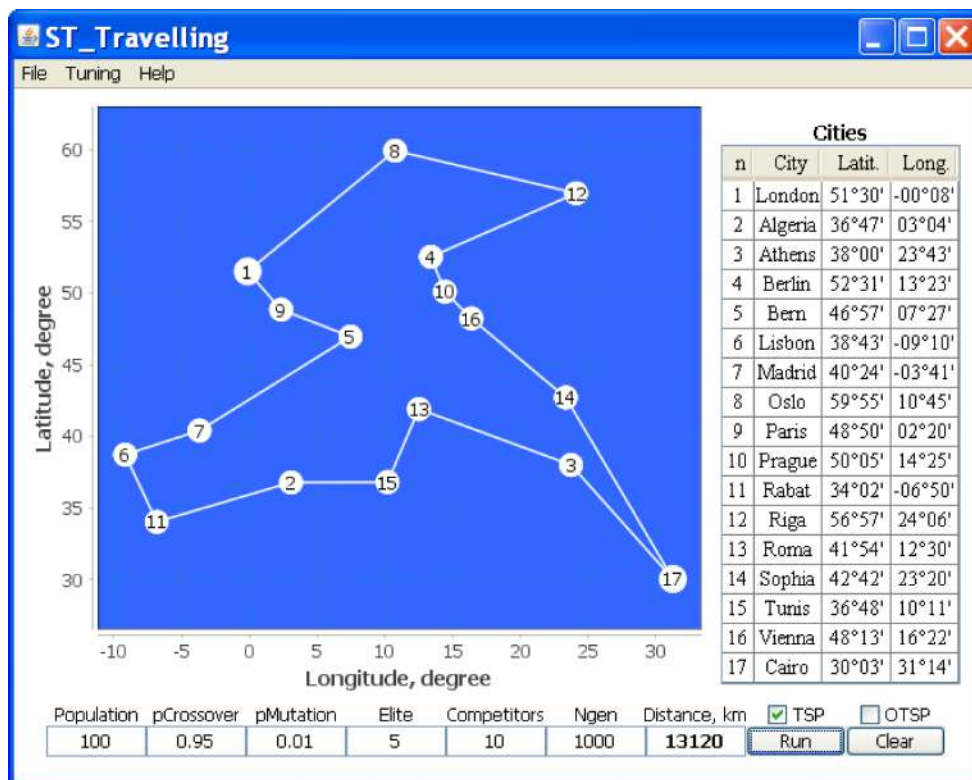


Рис. 1. Вікно програми з маршрутом руху

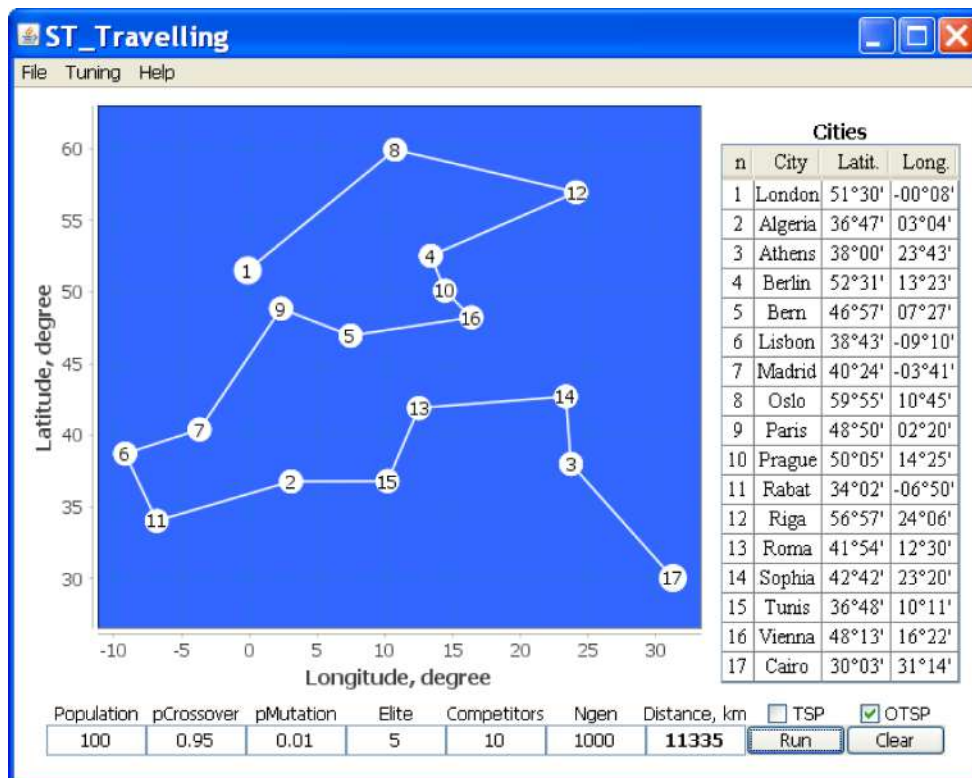


Рис. 2. Вікно програми з оптимальним маршрутом Лондон-Каїр

Для видалення знайденого маршруту використовувалася кнопка «Clear». Користувач програми за допомогою меню «Tuning», при необхідності, міг настроїти область відображення графіка. Інформація про додаток і особливості його роботи була доступна при використанні меню «Help».

На рис. 1 зображений маршрут авіасполучення для міст із таблиці «Cities», знайдений при рішенні замкнутого варіанта задачі комівояжера.

У той час, на рис. 2 зображений маршрут, знайдений при рішенні незамкнутого варіанта задачі комівояжера.

При цьому початковий і кінцевий пункт маршруту відповідає містам з першого та останнього рядка таблиці «Cities». При зміні записів у цих рядках, у випадку завантаження файлів з оновленими даними, відповідно міняється, як початковий, так і кінцевий пункт маршруту.

На рис. 3 зображений оптимальний маршрут руху між Парижем та Афінами, знайдений додатком ST\_Travelling при рішенні незамкнутого варіанта задачі комівояжера.

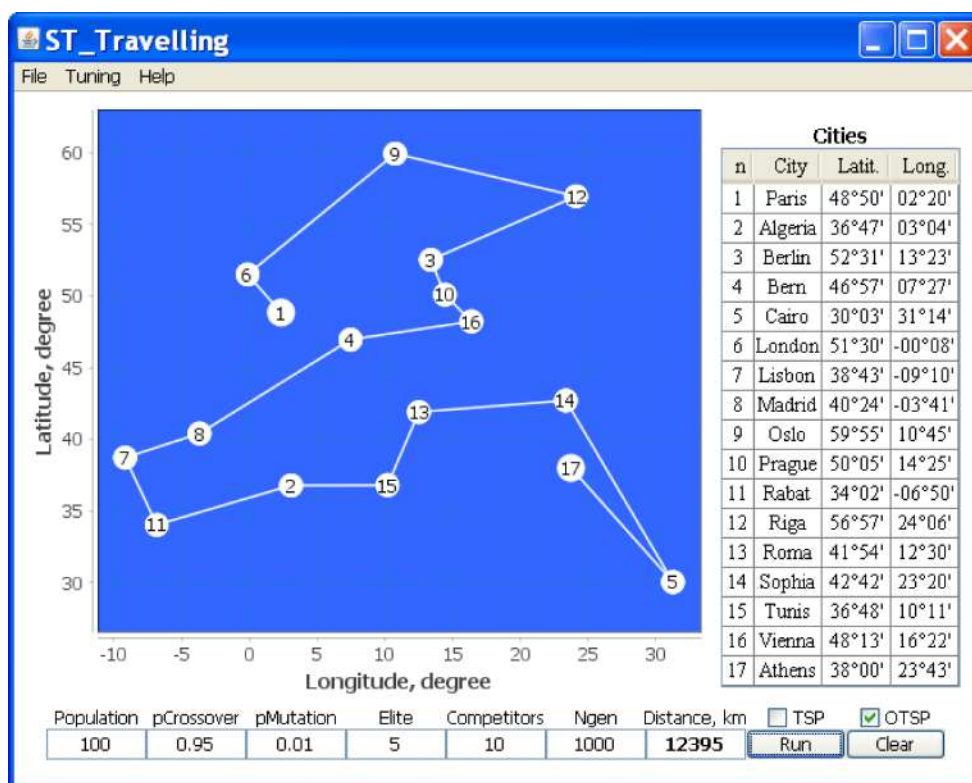


Рис. 3. Вікно програми з оптимальним маршрутом Париж-Афіни

### Висновки

У роботі був розроблений кросплатформний Java-додаток ST\_Travelling з GUI для знаходження оптимальних маршрутів руху на основі рішення, як замкнутого, так і незамкнутого варіанта задачі комівояжера з допомогою генетичного алгоритму.

Для знаходження маршруту, при рішенні незамкнутого варіанта задачі комівояжера, початковий та кінцевий пункт руху визначалися розташуванням інформації про них у вихідному файлі даних. Перший запис ініціалізував початковий пункт руху, останній – кінцевий.

При розрахунку довжини маршруту додаток використовував формулу гаверсинусів. Надалі передбачається модифікувати Java-додаток ST\_Travelling, зокрема, при знаходженні протяжності маршруту використовувати більш точні значення широти та довготи пунктів руху.

### Список використаної літератури

1. Смирнов И., Косарева Т. Транспортная логистика. Навчальний посібник. Київ : Центр навчальної літератури. 2019. 224 с.
2. Jacobson L., Kanber B. Genetic Algorithms in Java Basics. Apress. 2015. 154 p.
3. Wirsansky Eyal. Hands-On Genetic Algorithms with Python. Birmingham : Packt Publishing. 2020. 320 p.
4. Heaton J. Introduction to Neutral Network with Java. St. Louis: Heaton Research. 2008. 442 p.
5. Abubakar B. S. Developing a Java-based Genetic Algorithm to Solve the Travelling Salesman Problem. International Journal of Computer, Vol. 30, No 1, 2018, pp. 43–49.

6. Van Brummelen Glen. Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry. Princeton : Princeton University Press, 2013. 222 p.

#### References

1. Smirnov I., Kosareva T. (2019) Transportna lohistyka. Navchal'nyy posibnyk [Transport logistics. Manual]. Kyiv: Center for Educational Literature. 224 p. [in Ukrainian].
2. Jacobson L., Kanber B. (2015) Genetic Algorithms in Java Basics. Apress. 154 p.
3. Wirsansky Eyal. (2020) Hands-On Genetic Algorithms with Python. Birmingham : Packt Publishing. 320 p.
4. Heaton J. (2008) Introduction to Neural Network with Java. St. Louis : Heaton Research. 442 p.
5. Abubakar B. S. (2018) Developing a Java-based Genetic Algorithm to Solve the Travelling Salesman Problem. International Journal of Computer, Vol. 30, No 1, pp. 43–49.
6. Van Brummelen Glen. (2013) Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry. Princeton : Princeton University Press. 222 p.

*Дата першого надходження рукопису до видання: 28.09.2025*  
*Дата прийнятого до друку рукопису після рецензування: 23.10.2025*  
*Дата публікації: 28.11.2025*