

# ПОРІВНЯЛЬНИЙ АНАЛІЗ ГЕНЕТИЧНОГО ТА МУРАШИНОГО АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ МАРШРУТІВ У ЗАДАЧІ КОМІВОЯЖЕРА

Вінницький національний технічний університет

## Анотація

Задача комівояжера є однією з класичних оптимізаційних проблем. Значущість цієї задачі обумовлена її застосуванням в різноманітних областях, таких як: транспорт, логістика, виробництво та інші.

У даній роботі проведено порівняльний аналіз ефективності генетичного та мурашиного алгоритмів у контексті оптимізації маршрутів у задачі комівояжера. Здійснено детальний розгляд основних принципів функціонування обох алгоритмів та їхніх параметрів. Експериментально визначено вплив зазначених алгоритмів на якість знаходження оптимальних рішень для поставленої задачі. Результати дослідження дозволяють зробити висновки щодо переваг та обмежень використання генетичного та мурашиного підходів у вирішенні проблеми комівояжера.

**Ключові слова:** задача комівояжера, генетичні алгоритми, мурашині алгоритми, оптимізація маршрутів.

## Abstract

The traveling salesman problem is one of the classical optimization challenges. The significance of this problem is determined by its application in various fields, such as transportation, logistics, manufacturing, and others.

This paper presents a comparative analysis of the effectiveness of genetic and ant colony algorithms in the context of route optimization for the traveling salesman problem. A detailed examination of the fundamental principles of both algorithms and their parameters is conducted. The experimental determination of the impact of these algorithms on the quality of finding optimal solutions for the presented problem is performed. The research results allow drawing conclusions regarding the advantages and limitations of using genetic and ant colony approaches in solving the traveling salesman problem.

**Keywords:** travelling salesman problem, genetic algorithms, ant colony algorithms, route optimization.

## Вступ

Задача комівояжера (Travelling Salesman Problem, TSP) полягає у знаходженні найвигіднішого маршруту, що проходить через вказані міста хоча б по одному разу [1].

Задача була вперше сформульована в 1930 році і є однією з найбільш інтенсивно досліджуваних проблем оптимізації. Вона використовується як еталон для багатьох методів оптимізації. Незважаючи на те, що проблема складна з обчислювальної точки зору, відомо багато евристик і точних алгоритмів, тому деякі випадки з десятками тисяч міст можна розв'язати повністю [2].

TSP має декілька застосувань навіть у найчистішому вигляді, наприклад, планування, логістика та виробництво мікročіпів. Злегка змінений, він виглядає як підпроблема в багатьох сферах, однією з них є секвенування ДНК. У цих програмах концепція міста представляє, наприклад, клієнтів, точки спаявання або фрагменти ДНК, а концепція відстані представляє час у дорозі чи вартість, або міру подібності між фрагментами ДНК. TSP також з'являється в астрономії, оскільки астрономи, які спостерігають за багатьма джерелами, хочуть мінімізувати час, витрачений на переміщення телескопа між джерелами, у таких задачах TSP може бути вбудований у задачу оптимального керування. У багатьох програмах можуть бути накладені додаткові обмеження, такі як обмежені ресурси або часові вікна [2].

Генетичний алгоритм відомий своєю здатністю не тільки вирішувати і скорочувати перебір у складних завданнях, але й легко адаптуватися до зміни проблеми. Тоді як мурашиний алгоритм визначається своєрідністю підходу та можливістю використання в різних областях, включаючи транспорт та логістику.

Метою роботи є порівняльний аналіз генетичного та мурашиного алгоритмів для оптимізації маршрутів у задачі комівояжера.

### Результати дослідження

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, невідності) може служити сумарний час у дорозі, сумарна вартість дороги, або, в простому випадку, довжина маршруту [3].

Існує маса різновидів узагальненої постановки задачі, зокрема геометрична задача комівояжера (коли матриця відстаней відображає відстані між точками на площині), трикутна задача комівояжера (коли на матриці вартостей виконується нерівність трикутника), симетрична та асиметрична задачі комівояжера [1].

Прості методи розв'язання задачі комівояжера: повний лексичний перебір, жадібні алгоритми (метод найближчого сусіда), метод включення найближчого міста, метод найдешевшого включення, метод мінімального кістяка дерева. На практиці застосовують різні модифікації ефективніших методів: метод гілок і меж і метод генетичних алгоритмів, а також алгоритм мурашиної колонії.

Всі ефективні (такі, що скорочують повний перебір) методи розв'язання задачі комівояжера — евристичні. У більшості евристичних методів знаходиться не найефективніший маршрут, а наближений розв'язок. Користуються популярністю так звані any-time алгоритми, тобто алгоритми, що поступово покращують деякий поточний наближений розв'язок [1].

Генетичний алгоритм — це метод розв'язання задач як обмеженої, так і необмеженої оптимізації, який ґрунтується на природному відборі — процесі, який рухає біологічну еволюцію. Генетичний алгоритм неодноразово змінює популяцію окремих рішень [4]. Алгоритм використовує аналоги генетичного представлення (бітові рядки), придатності (оцінки функцій), генетичної рекомбінації (перехрещування бітових рядків) і мутації (перевертання бітів). Алгоритм працює так, що спочатку створюється популяція випадкових бітових рядків фіксованого розміру. Основний цикл алгоритму повторюється протягом фіксованої кількості ітерацій або доти, доки не буде видно подальше покращення найкращого рішення за задану кількість ітерацій [5]. На кожному кроці генетичний алгоритм вибирає особин із поточної популяції, які стануть батьками, і використовує їх для народження дітей для наступного покоління. Протягом наступних поколінь популяція «еволюціонує» до оптимального рішення. На рисунку 1.1 наведено блок-схему, де описано основні кроки алгоритму [4].

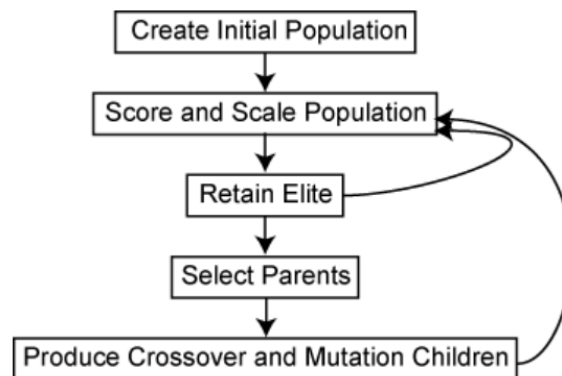


Рисунок 1.1 – блок-схема з основними кроками генетичного алгоритму

Генетичний алгоритм використовує три основні типи правил на кожному кроці для створення наступного покоління з поточної популяції:

1. Правила відбору відбирають особин, які називаються батьками, які вносять внесок у популяцію наступного покоління. Вибір, як правило, є стохастичним і може залежати від результатів окремих осіб;

2. Правила кросинговеру поєднують двох батьків для формування дітей для наступного

покоління;

3. Правила мутації застосовують випадкові зміни до окремих батьків для формування дітей.

Техніка оптимізації мурашиних колоній повністю натхненна поведінкою колоній мурах у пошуку їжі, яку вперше представив Марко Доріго в 1990-х роках. Мурахи є еусоціальними комахами, які віддають перевагу виживанню та підтримці спільноти, а не як окремі види. Вони спілкуються один з одним за допомогою звуку, дотику та феромону. Феромони — це органічні хімічні сполуки, що виділяються мурахами, які викликають соціальну реакцію у представників того ж виду. Це хімічні речовини, здатні діяти як гормони поза тілом особини, яка виділяє секрет, впливаючи на поведінку особин, які їх приймають. Оскільки більшість мурах живе на землі, вони використовують поверхню ґрунту, щоб залишати сліди феромонів, за якими можуть стежити (відчувати запах) інші мурахи [6].

Мурахи живуть у гніздах, і основний принцип АСО полягає в спостереженні за рухом мурах із їхніх гнізд, щоб шукати їжу найкоротшим можливим шляхом. Спочатку мурахи починають безладно пересуватися в пошуках їжі навколо своїх гнізд. Цей випадковий пошук відкриває кілька маршрутів від гнізда до джерела їжі. Тепер, виходячи з якості та кількості їжі, мурахи повертають частину їжі з необхідною концентрацією феромонів на зворотному шляху. Залежно від цих виділень феромонів, ймовірність вибору певного шляху наступними мурахами буде керівним фактором для джерела їжі. Очевидно, ця ймовірність базується на концентрації, а також на швидкості випаровування феромону. Також можна помітити, що оскільки швидкість випаровування феромону також є вирішальним фактором, довжину кожного шляху можна легко порівняти [6].

Так само в АСО у нас є група штучних мурах, які намагаються знайти найкраще вирішення проблеми. Наприклад, якщо ми хочемо знайти найкоротший шлях між двома точками, штучні мурахи випадковим чином переміщуються з однієї точки в іншу, залишаючи за собою феромонний слід. Феромонний слід вказує на якість шляху, який пройшла мураха. Мурахи вважають за краще йти за сильнішим феромонним слідом, який відповідає кращому шляху [7].

На рисунку 1.2 для простоти розглянуто два можливі шляхи між джерелом їжі та гніздом мурах.

Етапи можна проаналізувати таким чином:

1. Етап 1: усі мурахи у своєму гнізді. У середовищі відсутній вміст феромонів. (Для алгоритмічного проектування залишкову кількість феромону можна враховувати без впливу на ймовірність);

2. Етап 2: Мурахи починають пошук з рівною (0,5 кожна) ймовірністю вздовж кожного шляху. Зрозуміло, що вигнутий шлях довший, а отже, час, який потрібен мурахам, щоб дістатися до джерела їжі, більше, ніж інший;

3. Етап 3: Мурахи коротшим шляхом досягають джерела їжі раніше. Зараз, очевидно, вони стикаються з подібною дилемою відбору, але цього разу завдяки феромонному сліду вздовж коротшого шляху, який уже доступний, ймовірність відбору вища;

4. Етап 4: більше мурах повертається коротшим шляхом, і згодом концентрація феромонів також зростає. Крім того, через випаровування концентрація феромонів на довшому шляху зменшується, зменшуючи ймовірність вибору цього шляху на подальших етапах. Тому вся колонія поступово використовує коротший шлях із вищою ймовірністю. Отже, оптимізація шляху досягнута.

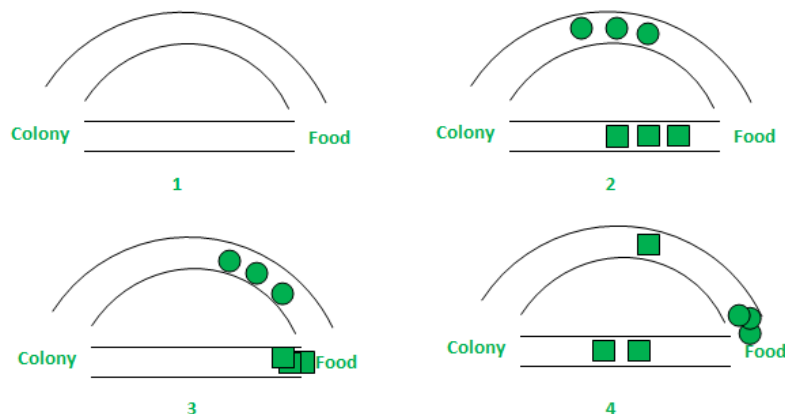


Рисунок 1.2 – простий приклад мурашиного алгоритму

У даній роботі, для генетичного алгоритму основні дослідження проведені з параметрами схрещування та мутації. За замовчуванням були обрані наступні параметри:

1. Кількість індивідів (особин) у популяції на кожній ітерації алгоритму дорівнює 300;
2. Кількість найкращих індивідів, які вибираються з популяції на кожній ітерації для збереження у наступному поколінні без змін дорівнює 10.

Для мурашиного алгоритму основні дослідження проведені з параметрами ваги феромонів для вибору мурахами наступного міста, ваги евристичної інформації (відстані між містами) та коефіцієнтом випаровування. За замовчуванням були обрані наступні параметри:

1. Кількість мурах у колонії дорівнює 10;
2. Вага феромону, яка додається до кращого шляху або маршруту (елітного) для підвищення ймовірності вибору цього маршруту мурахами дорівнює 1;
3. Мінімальний коефіцієнт масштабування для феромонів дорівнює 0.001;
4. Коефіцієнт, який визначає, наскільки багато феромону додається мурахою на шляху під час внесення внеску феромону дорівнює 1.0;
5. Початковий рівень феромону, присвоєний усім ребрам мережі дорівнює 1.0;
6. Кількість ітерацій або кроків, які виконує мурашиний алгоритм дорівнює 150.

Також для дослідження було обрано наступну схему розташування міст (графів) для задачі комівояжера зображену на рисунку 1.3.

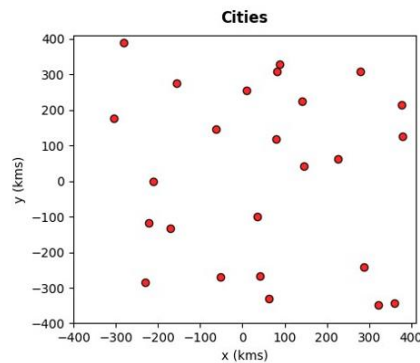


Рисунок 1.3 – схема розташування міст (графів)

Після запуску одного з алгоритмів з певними параметрами на екрані відображається оптимальний шлях, який вдалось знайти алгоритму (рисунок 1.4), та графік, на якому видно найкращу дистанцію знайдену на певній ітерації (рисунок 1.5).

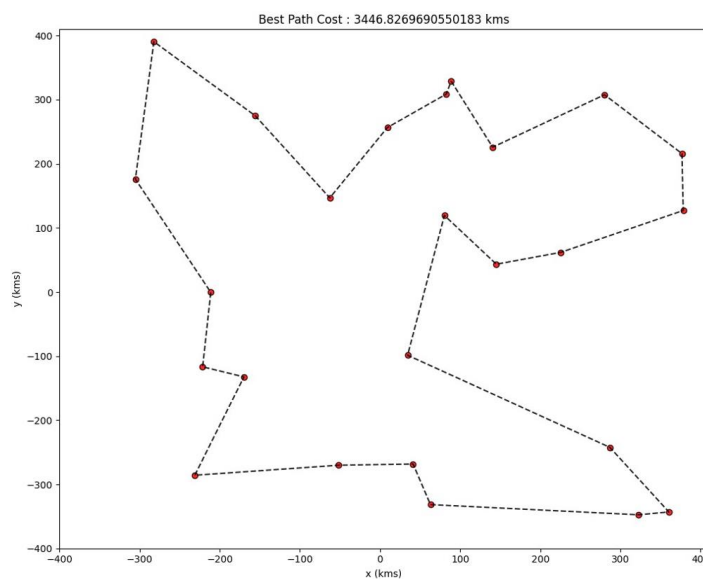


Рисунок 1.4 – приклад оптимального шляху, знайденого алгоритмом

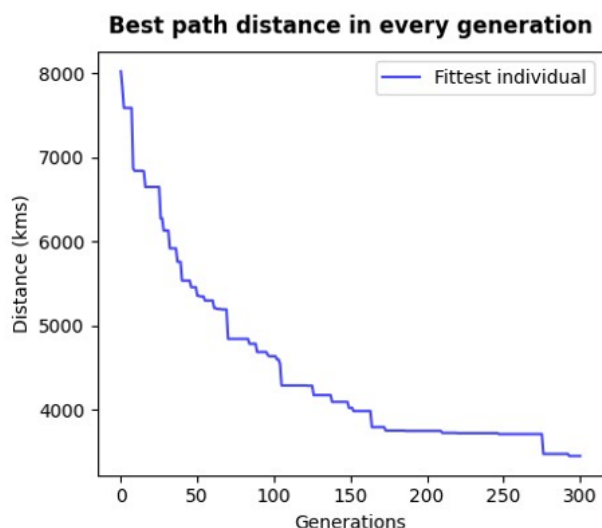


Рисунок 1.5 – приклад графіку з відображенням найкращої дистанції на певній ітерації алгоритму

Загалом було проведено по 10 запусків кожного алгоритму, параметри яких та результати подано в таблиці 1.

Таблиця 1 – Результати дослідження

Генетичний алгоритм		Мурашиний алгоритм			Оптимальний маршрут, км	
Мутації	Схрещення	Alpha (вага феромона у виборі наст. міст)	Beta (евристична інформація (відстані між містами))	Rho (коефіцієнт випаровування)	Генетичний	Мурашиний
0.1	0.9	0.1	1	1	3628.98	7150.31
0.2	0.9	0.2	1.5	0.2	3446.83	4516.26
0.2	0.8	0.3	2	0.3	3446.83	3818.62
0.3	0.2	0.3	5	0.4	3807.73	3446.83
0.3	0.5	0.5	3	0.1	4103.79	3517.98
0.4	0.9	0.6	2	0.6	3446.83	3450.01
0.7	0.2	0.7	4	0.7	4106.94	3446.83
0.6	0.8	0.8	4.5	0.2	3941.57	3446.83
0.8	0.4	0.9	1	0.1	3978.15	3717.32
0.9	0.1	1	5	0.1	4254.73	3446.83

З таблиці видно, що найкращий результат в генетичному алгоритмі досягається при великих значеннях схрещення (0.8-0.9) та низьких значеннях мутації (0.1-0.4). Це може вказувати на те, що обмін генами є більш ефективним за мутації в даному випадку.

Щодо мурашиного алгоритму, то вага alpha (вага феромонів у виборі наступного міста) та beta (евристична інформація) мають значний вплив на результат. Високі значення alpha та більш помітне значення beta супроводжуються кращими результатами. Коефіцієнт випаровування (rho) також має вплив. Низьке значення rho супроводжується кращими результатами.

Загалом, враховуючи обидва алгоритми, важливо знайти баланс між параметрами для досягнення оптимальності. У генетичного алгоритму це високе схрещення та низькі мутації, а у мурашиного алгоритму - підходящі значення alpha, beta і низьке rho. В цілому результати показують, що обидва алгоритми - генетичний та мурашиний - при доцільному виборі параметрів можуть досягти однакового мінімального значення відстані між містами, яке становить 3446.83 км.

## Висновки

У даній роботі було досліджено генетичний та мурашиний алгоритми в контексті оптимізації маршрутів у задачі комівояжера. Отримані результати дозволяють зробити наступні висновки:

1. Обидва алгоритми виявили здатність до знаходження оптимального маршруту, досягаючи однакового мінімального значення відстані між містами (3446.83) при підходящій параметризації;

2. Генетичний алгоритм демонструє ефективність при високому рівні схрещення та низькому рівні мутацій. Мурашиний алгоритм, з іншого боку, виявляється більш ефективним при високих значеннях ваги феромонів ( $\alpha$ ), великій вазі евристичної інформації ( $\beta$ ), та низькому коефіцієнті випаровування ( $\rho$ ).

3. Обидва методи є перспективними для використання в задачах оптимізації маршрутів комівояжера. Вибір конкретного методу може залежати від конкретних вимог завдання та характеристик вхідних даних.

Отже, подальший розвиток та дослідження цих алгоритмів можуть призвести до вдосконалення їхньої ефективності та застосування у різних областях, де вирішення задачі комівояжера є актуальним завданням.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Задача комівояжера [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Задача\\_комівояжера](https://uk.wikipedia.org/wiki/Задача_комівояжера) – Назва з екрану;
2. Travelling salesman problem [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem) – Назва з екрану;
3. Господінов, А. М., Смирнов, С. А. (рік видання). Генетичний алгоритм для розв'язання задачі комівояжера. [Назва роботи]. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Фізико-технічний інститут. [Електронний ресурс] Режим доступу до ресурсу: <https://ela.kpi.ua/bitstream/123456789/25202/1/S.19-21.pdf>;
4. What Is the Genetic Algorithm? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html> – Назва з екрану;
5. Simple Genetic Algorithm From Scratch in Python [Електронний ресурс] – Режим доступу до ресурсу: <https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/> – Назва з екрану;
6. Introduction to Ant Colony Optimization [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/> – Назва з екрану;
7. Implementing Ant colony optimization in python- solving Traveling salesman problem [Електронний ресурс] – Режим доступу до ресурсу: <https://induraj2020.medium.com/implementation-of-ant-colony-optimization-using-python-solve-traveling-salesman-problem-9c14d3114475> – Назва з екрану.

**Лобода Максим Олександрович** — студент групи ІСТ-206, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [loboba12392@gmail.com](mailto:loboba12392@gmail.com)

**Кулик Ярослав Анатолійович** – к.т.н., доцент кафедри Автоматизації та інтелектуальних інформаційних технологій, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: [kulyk.y.a@vntu.edu.ua](mailto:kulyk.y.a@vntu.edu.ua)

**Loboda Maksym Oleksandrovych** — Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: [loboba12392@gmail.com](mailto:loboba12392@gmail.com)

**Kulyk Yaroslav Anatoliyovych** – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: [kulyk.y.a@vntu.edu.ua](mailto:kulyk.y.a@vntu.edu.ua)