

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-55-13>

УДК 004.7

Коляда Костянтин Вячеславович, к.т.н., старший викладач

<https://orcid.org/0000-0002-3962-5791>

Рожко Дмитро Вікторович, магістрант

<https://orcid.org/0009-0008-3444-8598>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна.

ВИКОРИСТАННЯ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗКУ ЗАДАЧІ КОМІВОЯЖЕРА

Коляда К.В., Рожко Д.В. Використання метаевристичних алгоритмів для розв'язку задачі комівояжера. У статті розглядається та проводиться аналіз існуючих метаевристичних алгоритмів. Можливість їх використання при побудові та передачі даних всередині комп'ютерної системи. Також їх застосування у локальних чи глобальних мережах об'єднаних комп'ютерів. Згадано принципи, які лягли в основу формування поняття алгоритм у галузі інформаційних технологій. Описано актуальність використання метаевристичних алгоритмів в проектуванні мереж передачі даних. Встановлено зміст поняття алгоритм та його складність, використані у контексті даної статті. Визначено метаевристичний алгоритм та його відмінності від інших. Розглянуто особливості конкретних видів даних алгоритмів знаходження оптимального шляху. Визначено концепції метода рою часток, які використовують колективний інтелект для реалізації гравітаційного пошуку закладених в основу даного підходу. Розглянуто окремі види метаевристичних алгоритмів, зокрема мурашиний та бджолиний алгоритми. Описано метод гравітаційного пошуку, закладений в основу евристичних алгоритмів оптимізації, згаданих раніше. Встановлено визначення колективного інтелекту для даного контексту, що ліг в основу даного метода рішення завдання комівояжера. Описано актуальність використання конкретних, раніше згаданих, алгоритмів. Демонстрація систем, у яких можливе використання метаевристичних алгоритмів. Вказано можливість їх прикладного застосування у прикладних сферах діяльності людини та суспільства. Запропоновано методи ефективного застосування мурашиного та бджолиного алгоритмів. Виділено переваги та недоліки їх використання. Сформульовано можливості подальшої модернізації даних конкретних методів рішення завдання комівояжера. Описано переваги бджолиного та мурашиного алгоритмів над генетичними алгоритм. Виділення переваг даної належності для практичного застосування даних алгоритмів для рішення задач оптимізації.

Ключові слова: метаевристичний алгоритм, мурашиний алгоритм, бджолиний алгоритм, евристичний алгоритм, метод рою часток, алгоритм гравітаційного пошуку, колективний інтелект.

Kolyada K., Rozhko D. Using metaheuristic algorithms to solve the Traveling Salesman Problem. The article considers and analyzes the existing metaheuristic algorithms. The possibility of their use in the construction and transfer of data within the computer system. Also their application in local or global networks of united computers. The principles that formed the basis of the formation of the concept of algorithm in the field of information technologies are mentioned. The relevance of using metaheuristic algorithms in the design of data transmission networks is described. The content of the concept of algorithm and its complexity, used in the context of this article, are established. The metaheuristic algorithm and its differences from others are defined. The peculiarities of specific types of these algorithms for finding the optimal path are considered. Concepts of the particle swarm method, which use collective intelligence to implement gravitational search, are defined as the basis of this approach. Some types of metaheuristic algorithms are considered, in particular, ant and bee algorithms. The gravitational search method, which is the basis of the previously mentioned heuristic optimization algorithms, is described. The definition of collective intelligence for this context was established, which formed the basis of this method of solving the traveling salesman's task. The relevance of the use of specific, previously mentioned, algorithms is described. Demonstration of systems in which the use of metaheuristic algorithms is possible. The possibility of their applied application in applied spheres of human and social activity is indicated. Methods of effective application of ant and bee algorithms are proposed. Advantages and disadvantages of their use are highlighted. Possibilities of further modernization of these specific methods of solving the traveling salesman's task are formulated. Advantages of bee and ant algorithms over genetic algorithms are described. Highlighting the advantages of this property for the practical application of these algorithms for solving optimization problems.

Keywords: polynomial algorithm, ant colony optimization, artificial bee colony, heuristic algorithm, particle swarm optimization, graphiti-zation search algorithm, swarm intelligence.

Постановка наукової проблеми.

У наш час з розповсюдженням мережевих цифрових технологій особливої важливості набувають підходи взаємодії між елементами мережі. Величина та складність системи у багато разів виросла у порівнянні з першими поколіннями інтернету. За даними DataReportal зараз всього 5.35 мільярдів людей користується інтернетом, що становить 66.2% всього 8.08 мільярдного населення нашої планети [1]. Дана величина постійно зростає. Ще у 2022 році людей онлайн було 63% від загальної кількості населення, відповідно до даних із згаданого раніше аналітичного джерела. Вказана статистичної інформація нашої нашої планети, що проблема пошуку оптимального шляху проходження повідомлень мережею є актуальною для ефективною передачі даних. Дана задача

комівояжера успішно вирішується існуючими метаевристичними алгоритмами. Проте їх удосконалення є актуальним для проходження повідомлень оптимальнішим шляхом, що дозволить не задіювати деякі вузли та ребра графа системи для передачі даних, що в свою чергу покращить ефективність передачі повідомлень мережею. Ефективне використання ресурсів мережевого апаратного забезпечення також є важливим при налаштуванні взаємодії між елементами комп'ютерної системи.

Аналіз досліджень.

Спочатку опису методів рішення поставленої задачі варто визначити з використанні термінами у контексті даного дослідження. Згадана ще в темі, задача комівояжера полягає у знаходженні найкоротшого маршруту, що проходить через всі вказані міста хоча б по одному разу. Існує багато різновидів цієї задачі, які вимагають знаходження оптимального маршруту з різними критеріями. Відомі дві групи методів розв'язання - точні і евристичні. Точні методи забезпечують оптимальні рішення за достатньо великий час, тоді як евристичні методи можуть знаходити ефективні шляхи, які не обов'язково є оптимальним рішенням за короткий час. На практиці частіше застосовуються більш складні методи, такі як алгоритм мурашиної колонії. Задача комівояжера є NP-повною, що означає, що для її розв'язання потрібно велика кількість обчислень [2]. Для вирішення проблеми комівояжера можливе її представлення у вигляді математичної моделі на графі. Зазвичай припускається, що кожна вершина графа відповідає місту, а ребро між вершинами показує зв'язок між містами. У рамках даного дослідження вершини являтимуть собою мережеві вузли, а ребра – з'єднання між ними. Кожному ребру можна призначити вагу, яка може відображати відстань, час або вартість подорожі між містами. Для спрощення задачі та забезпечення існування маршруту зазвичай припускають, що модельний граф задачі є повним, тобто між усіма парами вершин існує ребро. Якщо між деякими вузлами немає безпосереднього сполучення, то можна ввести ребра з максимальною вагою, щоб такі ребра не потрапляли до оптимального маршруту через їх велику довжину або вартість. Залежно від того, які ваги накладаються на ребра, існують різні варіанти задачі. Найважливіші серед них - симетрична та метрична задачі. Маршрут, який проходить по всіх вершинах графа лише один раз, який також називається гамільтоновим маршрутом. Метою задачі комівояжера є знайти найкоротший маршрут, який пройде через всі міста.

Метаевристичні методи рішення задачі комівояжера поєднують в собі пошук локальних та глобальних розв'язків для оптимізації задач. Багато з них спочатку знаходять початковий розв'язок і потім його покращують іншими методами. Для уникнення глухих кутів використовуються стратегії, такі як табу-пошук або симуляція відпалу. Деякі методи, наприклад мурашиний алгоритм або генетичні алгоритми, базуються на природніх процесах. Зазначені методи можуть знаходити як якісні, так і віддалені від оптимальних розв'язки. Результат та тривалість обчислення залежать від конкретних елементів, що використовуються. У даному випадку евристичний алгоритм може знайти краще рішення, яке не обов'язково є оптимальним. Часто евристичний алгоритм може бути найкращим рішенням для вирішення складних задач, навіть якщо він не завжди працює правильно [3]. Такі алгоритми широко використовуються на практиці, оскільки вони дозволяють отримувати швидкі та прийнятні результати в обмежені строки не залежно від оптимальності знайденого шляху. Гравітаційний пошук, закладений у метод оптимізації мурашиними та бджолиними колоніями, являє собою ефективний метод знаходження оптимального рішення завдання оптимізації, який базується на моделюванні гравітаційної взаємодії між рішеннями [4]. Дані алгоритми добре підходять для рішення задач з великою кількістю змінних та складними умовами завдань комбінаторної та багатокритерійної оптимізації. У реаліях мережевої чи комп'ютерної системи кількість ітерацій не завжди може бути константною для всіх випадків використання. Конфігурація мережі може змінюватись, тому для предметної області даної статі розглядатиметься поліноміальна версія метаевристичного мурашиного та бджолиного алгоритмів.

Поняття алгоритм за визначенням являє собою послідовність точно визначених кроків, які дозволяють вирішити певну задачу або проблему за скінченну кількість дій, закінчення яких обумовлюється логічними або апаратними обмеженнями електронно обчислювальної машини [5].

Популярним прикладом метаевристичних алгоритмів є мурашиний алгоритм, також відомий як алгоритм оптимізації мурашиної колонії. Він є ефективним методом для знаходження наближених розв'язків задачі комівояжера та схожих завдань на графах. Основна ідея полягає в тому, щоб мурахи маркували успішні шляхи великою кількістю феромону [6].

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^N l_k^q \cdot f_k^p}$$

де P_i – ймовірність проходження шляхом i , l_i – обернена до ваги i -того переходу величина, f_i – кількість феромона на i переході, q – жадібність алгоритму, p – стабільність алгоритму так, що $q + p = 1$.

Результат використання даного підходу може бути неточним з першого проходження або навіть гіршим, проте повторне застосування алгоритму може дати достатньо точний висновок через ймовірність правильного рішення.

У природі мурахи ходять довільно, але після знаходження їжі повертаються до колонії, залишаючи за собою феромонний слід. Інші мурахи, знаходячи цей шлях, починають за ним слідувати, підсилюючи його. Проте з часом феромони випаровуються, і довші шляхи втрачають привабливість. Як результат, короткі маршрути використовуються частіше, і феромони на них стають все більш концентрованими. Якби феромони не зникали, перші обрані мурахою шляхи були б пріоритетними для інших та розвідка нових шляхів була б обмежена. Даний принцип ігнорує вибір найбільш очевидного шляху і дозволяє мурахам знаходити найкращі шляхи до їжі. Мурашиний алгоритм полягає в наслідуванні цієї стратегії для пошуку оптимального рішення в задачах.

На рисунку 1 продемонстровано початкову концепцію, яка виникла під впливом спостережень за мураками, які, не дивлячись на свою індивідуальну обмеженість, спроможні колективно знаходити найкоротший шлях до джерела їжі в гніздо.

Наступні кроки описують логіку прийняття рішення:

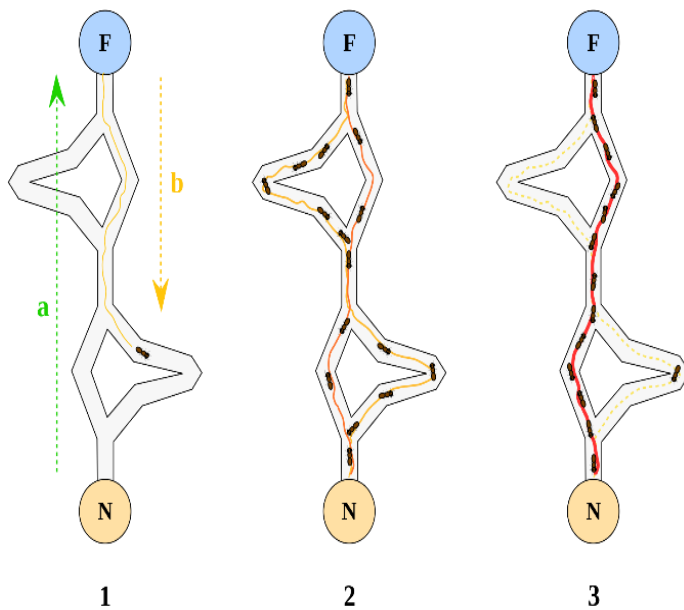


Рис. 1 – прийняття рішень у мурашиних алгоритмах.

1. Перша досліджувана мураха слідує з мурашника (F) в напрямку (b) до джерела їжі (N). Потім, шляхом (a) повертається до мурашника (F), залишаючи за собою феромонний слід.

2. Мурахи випадково обирають всі шляхи руху між мурашником (F) та джерелом їжі (N).

3. Мурахи ідуть коротшим серед усіх раніше випробуваних шляхів через втрату щільності феромонних слідів на коротших відрізках шляху.

Для формування математичної моделі для використання мурашиним алгоритмом вважатимемо, що середовище для мурах – це неорієнтовний граф, де кожне ребро має вагу, що вказує на відстань між вершинами. Мурахи можуть проходити по графу у будь-якому напрямку, оскільки граф двох скерований. Для вибору шляху мурахи використовують феромони на ребрах: чим більше феромону на ребрі, тим вища ймовірність, що мураха

обере конкретний шлях. Маршрут, який мурахи виберуть, визначає кількість феромону, яку вони залишають на ребрах. Даний чинник забезпечує пошук найкоротшого шляху. При використанні тільки додатній зворотного зв'язку мурашки можуть застрягнути в локально-оптимальному маршруті. Утворення зациклень можна уникнути, використовуючи від'ємний зворотний зв'язок, який озчатиме випаровування феромону. Правильне встановлення часу випарування може допомогти колонії знайти глобально оптимальне рішення.

Рівняння знаходження кількості феромону [6]:

$$\Delta \tau_{ij}^k = \frac{Q}{L^k(t)}$$

Рівнянням можливо обрахувати кількість феромону, яка лишилась на кожному ребрі графа шляху мурахою k інтенсивності Q на шляху L . Як результат, отримуємо короткий шлях, що характеризується великою концентрацією феромону.

Довший шлях містить його значення нижче. У подальшому обрахований результат можна використати у рівнянні для збільшення кількості феромону ребер пройдених шляхів мурахою [6].

$$\tau_{ij}(t) = \Delta \tau_{ij}(t) + (\tau_{ij}^k \cdot \rho)$$

У даному рівнянні важливо, що кожне ребро шляху позначається феромоном, який пропорційний довжині шляху. Потрібно зачекати, доки мураха обійде весь шлях і тільки потім оновити рівні феромону, щоб точно визначити довжину маршруту. Константа ρ має значення від 0 до 1.

Спочатку шляху кожне ребро може обиратися. Для поступового видалення ребер довшого шляху використовується процес випаровування з ребра графу феромону [6]. З рівняння маємо наступну його модифікацію описану формулою:

$$\tau_{ij}(t) = \tau_{ij}(t) \cdot (1 - \rho)$$

Для коректного використання рівняння за випаровування береться обернений коефіцієнт обрахунку шляху.

Одною із варіацій рішення задачі комівояжера метаевристичним алгоритмом є штучна бджолина сім'я, основний принцип якої структурно схожий медоносним бджолам. Бджолиний алгоритм є відносно новим методом для пошуку глобальних екстремумів складних функцій у багатьох вимірах. Алгоритм поєднує сусідній та випадковий пошук і може також використовуватися для оптимізації як комбінаторних, так і функціональних завдань.

Бджолиний алгоритм базується на принципах пошуку оптимального рішення за допомогою колективного інтелекту. Бджоли в алгоритмі шукають найбільш придатні джерела для збирання нектару, використовуючи випадкові початкові координати і швидкості. Кожна бджола може запам'ятати місце, де вона знайшла багато квітів і порівнювати їх з іншими місцями, які відкрили інші бджоли. Коли бджола вибирає між поверненням до свого власного місця або дослідженням місця, визначеного іншими, вона вирішує, на якому місці шукати квіти, керуючись особистим досвідом або соціальними сигналами. Іноді бджола може виявити нове місце з більшою кількістю квітів, яке варто відвідати. Інші бджоли потім дізнаються про це і також направляються туди, спираючись на бджолиний танець. Процес, де бджоли досліджують поле, переходячи до місць з великою кількістю квітів і сповільнюючи рух, щоб перевірити їх. Вони шукають найбільшу концентрацію квітів, порівнюючи з раніше відвіданими місцями. Потім бджола рухається до місця з найбільшою кількістю квітів, і інші бджоли робітничі починають збирати нектар там [6].

У початковому етапі алгоритму методу бджолиної сім'ї встановлюють параметри для оптимізації і визначають допустимий інтервал для пошуку оптимальних значень. Потім бджоли розташовуються випадковим чином в допустимій області, а їх вектори і швидкості руху визначаються, при чому кожна частка повинна переміщатися через простір рішень, якщо вона була б бджолою в сім'ї. Далі проводиться обробка даних для кожної частинки, оцінюючи її придатність порівнюючи з персональною найкращою позицією та глобальною національною позицією. Якщо знайдено значення краще, ніж попереднє, то відповідні позиції замінюються. Коригування швидкості частинки є важливим елементом процесу оптимізації, де важливо правильно визначити швидкість залежно від розміщення позицій. Частинка стежить за найбільшою придатністю у напрямку цих позицій згідно з певним рівнянням:

$$v_n^{i+1} = w \cdot v_n^i + c_1 r(p_n - x_n) + c_2 r(g_n - x_n)$$

де v_n^i – швидкість бджоли а n вимірі попереднього кроку, x_n – координата бджоли n вимірі, p_n – найкраща персональна позиція, g_n – найкраща глобальна позиція, r – довільне число вірогідності від -1 до 1. Обчислення проводяться для кожного n кроку-виміру. Дане рівняння показує, що нова швидкість формується шляхом збільшення або зменшення початкової швидкості в w разів, а також додавання векторів швидкості глобальної і поточної глобальної позиції для конкретного напрямку. c_1 та c_2 являють собою пізнавальні та соціальні коефіцієнти визначення взаємного тяжіння до глобальної та персональної найкращої позиції. Вказані значення визначають вплив на бджолі і визначає її пам'ять про найкраще рішення вибору найкращої персональної позиції та інших членів сім'ї відповідно. Збільшення цих коефіцієнтів сприяє дослідженню простору рішень шляхом руху кожної частинки в напрямку своєї позиції та дослідження глобального максимуму. w вважається

інерціальною вагою від 0 до 1 та відображає міру бджоли, яка характеризує вірність своєму первинному курсові з ігноруванням впливу глобальних та персональних найкращих позицій.

Можливості врахування меж в задачах можуть бути різними. Один із способів – зробити їх поглинаючими. Даний підхід визначає, що бджола у випадку виходу за дозволену область буде повернена в заданий простір рішень. Інший варіант рішення полягає у відображенні швидкості частинки при наближенні до межі. Однак найефективнішим виявилось використання "невидимих стін", де бджола може вилітати за межі дозволеної області. Отримані значення у даному випадку не враховуються, поки вона не повернеться назад.

Метою роботи є аналіз існуючих метаевристичних алгоритмів з подальшим формулюванням нового підходу для рішення задачі комівояжера при відправленні повідомлень між елементами комп'ютерної системи чи мережі.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Результатом аналізу опису мурашиного алгоритму є те, даний метод рішення задачі комівояжера є метаевристичним методом оптимізації, інспірованим поведінкою мурах при пошуку шляху джерела їжі. Основна ідея алгоритму полягає в тому, що мурахи залишають феромони на своєму шляху, інші мурахи використовують цю інформацію для вибору найбільш оптимального шляху. Чим більше феромонів залишено на певному шляху, тим ймовірніше, що інші мурахи оберуть саме цей шлях. Переваги мурашиного алгоритму включають його здатність знаходити оптимальні рішення в складних просторах пошуку, а також його здатність до адаптації до змін в навколишньому середовищі. Крім того, цей алгоритм відмінно підходить для задач комбінаторної оптимізації. Однак у мурашиного алгоритму є недоліки. Наприклад, він може сходитися до локального оптимуму, якщо не вживати достатніх заходів для різноманітності шляхів пошуку. Також даний алгоритм може бути неефективним у випадках, коли мурахи не можуть комунікувати між собою або коли інформація про феромони швидко втрачається. У цілому нині, мурашиний алгоритм демонструє хороші результати у вирішенні складних завдань оптимізації, але його ефективного застосування необхідно враховувати специфіку завдання та особливості довкілля.

Результатом аналізу опису бджолиного алгоритму є те, даний метод рішення задачі комівояжера є метаевристичний алгоритм оптимізації, інспірований поведінкою бджіл під час пошуку їжі. Він заснований на ідеї множинного руху рою бджіл навколо квітів, збираючи їжу та обмінюючись інформацією про місцезнаходження квітів.

Принцип роботи бджолиного алгоритму полягає в наступному:

1. Ініціалізація популяції – початкове розподіл бджіл з різних рішень простору пошуку.
2. Визначення критерію якості – функція, яку потрібно оптимізувати.
3. Ітераційний процес – бджоли переміщуються по простору рішень, обмінюючись інформацією та оновлюючи своє положення відповідно до кращих рішень.
4. Конвергенція – алгоритм сходиться до раціонального рішення, враховуючи обмін інформацією та ітераційне оновлення популяції.

Перевагами бджолиного алгоритму являються простота реалізації та ефективність у вирішенні багатьох завдань оптимізації, багаторазове використання практичної інформації при ухваленні рішень про наступний крок, здатність знаходити оптимальне рішення за наявності кількох локальних екстремумів.

Недоліками алгоритму бджолиної колонії є нестійкість до вибору параметрів та функцій адаптації, висока обчислювальна складність зі збільшенням розмірності простору пошуку, необхідність ретельного налаштування параметрів для конкретного завдання.

Таким чином, бджолиний алгоритм є потужним інструментом для вирішення завдань оптимізації, який ефективний у вирішенні різних завдань, проте вимагає ретельного налаштування та вибору параметрів для досягнення оптимальних результатів.

Спільними рисами розглянутих алгоритмів є те, що і мурашині, і бджолині є евристичними ройовими алгоритмами гравітаційного пошуку з використанням колективного інтелекту. Вони мають випадкову поведінку, але був розроблений детермінований алгоритм на основі гравітаційної кінематики під назвою "центральна оптимізація сил" та починають пошук з однієї точки. Розвідка полягає в розширенні простору пошуку, а використання - у знаходженні оптимального розв'язку. Для ефективного пошуку потрібно знаходити баланс між цими двома аспектами. Щоб уникнути підтримання в локальному оптимумі, алгоритм повинен розглядати результати розвідки на

```
point:
| num | x | y |
+-----+-----+
| 0 | 12.6656 | 34.3967 |
| 1 | 58.3999 | 47.5795 |
| 2 | -50.2695 | 86.1596 |
| 3 | -78.4910 | 16.5287 |
| 4 | -24.0765 | 53.6886 |
| 5 | 40.6567 | -21.2777 |
| 6 | -10.8025 | -91.8239 |
| 7 | -83.2807 | -6.0268 |
| 8 | -39.3477 | -42.7264 |
| 9 | -32.1911 | 75.3474 |
```

Рис. 2 – задані точки з координатами для симуляції мурашиної колонії

```
best distance = 532.533
path:
| id from | id to | distance |
+-----+-----+
| 3 | 4 | 65.8923 |
| 4 | 2 | 41.7186 |
| 2 | 9 | 21.0650 |
| 9 | 0 | 60.7378 |
| 0 | 1 | 47.5963 |
| 1 | 5 | 71.1065 |
| 5 | 6 | 87.3202 |
| 6 | 8 | 56.7925 |
| 8 | 7 | 57.2448 |
| 7 | 3 | 23.0584 |
```

Рис. 4 – найкращий маршрут знайдений мурашиною колонією.

```
iter | min | max | best
+-----+-----+-----+-----+
| 1 | 666.987 | 1052.95 | 666.987 |
| 2 | 661.275 | 971.391 | 661.275 |
| 3 | 532.533 | 1098.29 | 532.533 |
| 4 | 586.978 | 979.913 | 532.533 |
| 5 | 585.27 | 997.433 | 532.533 |
| 6 | 566.662 | 1038.35 | 532.533 |
| 7 | 532.533 | 978.047 | 532.533 |
| 8 | 621.485 | 1054.17 | 532.533 |
| 9 | 534.693 | 989.837 | 532.533 |
| 10 | 633.867 | 942.93 | 532.533 |
| 11 | 610.697 | 953.805 | 532.533 |
| 12 | 621.291 | 1005.77 | 532.533 |
| 13 | 586.978 | 1020.95 | 532.533 |
| 14 | 666.169 | 1017.43 | 532.533 |
| 15 | 601.881 | 995.576 | 532.533 |
| 16 | 594.415 | 1020.02 | 532.533 |
| 17 | 554.668 | 976.186 | 532.533 |
| 18 | 554.522 | 1031 | 532.533 |
| 19 | 620.638 | 1049.04 | 532.533 |
| 20 | 564.309 | 966.198 | 532.533 |
| 21 | 564.309 | 1019.38 | 532.533 |
| 22 | 619.242 | 999.078 | 532.533 |
| 23 | 659.674 | 950.696 | 532.533 |
```

Рис. 3 – обходи точок мурашиною колонією з знайденим найкращим результатом на кожному проході

```
iter | min | max | best
+-----+-----+-----+-----+
| 1 | 522.257 | 1087.47 | 522.257 |
| 2 | 580.395 | 1081.39 | 522.257 |
| 3 | 557.754 | 1047.5 | 522.257 |
| 4 | 603.507 | 980.905 | 522.257 |
| 5 | 503.944 | 999.468 | 503.944 |
| 6 | 540.502 | 959.941 | 503.944 |
| 7 | 532.533 | 958.67 | 503.944 |
| 8 | 557.256 | 987.511 | 503.944 |
| 9 | 503.944 | 985.896 | 503.944 |
| 10 | 634.692 | 1041.03 | 503.944 |
| 11 | 569.059 | 971.638 | 503.944 |
| 12 | 634.949 | 999.326 | 503.944 |
| 13 | 576.659 | 1018.68 | 503.944 |
| 14 | 543.837 | 954.769 | 503.944 |
| 15 | 627.237 | 980.047 | 503.944 |
| 16 | 616.752 | 1041.28 | 503.944 |
| 17 | 566.662 | 972.093 | 503.944 |
| 18 | 566.662 | 971.183 | 503.944 |
| 19 | 522.257 | 952.629 | 503.944 |
| 20 | 572.834 | 1024.71 | 503.944 |
| 21 | 540.502 | 950.626 | 503.944 |
| 22 | 564.502 | 963.866 | 503.944 |
| 23 | 569.059 | 1025.25 | 503.944 |
| 24 | 623.793 | 1037.15 | 503.944 |
| 25 | 622.167 | 1020.78 | 503.944 |
```

Рис. 5 – обходи точок мурашиною колонією з обміном інформацією про знайдений найкращий результат на кожному проході.

```
best distance = 503.944
path:
| id from | id to | distance |
+-----+-----+-----+
| 1 | 5 | 71.1065 |
| 5 | 6 | 87.3202 |
| 6 | 8 | 56.7925 |
| 8 | 7 | 57.2448 |
| 7 | 3 | 23.0584 |
| 3 | 2 | 75.1327 |
| 2 | 9 | 21.0650 |
| 9 | 4 | 23.1290 |
| 4 | 0 | 41.4989 |
| 0 | 1 | 47.5963 |
```

Рис. 6 – найкращий маршрут знайдений мурашиною колонією з обміном інформацією між мурахами.

початкових ітераціях. Вони надають хороші результати у цільових сферах використання, тому що немає одного універсального алгоритму, який вирішить всі оптимізаційні завдання. Мурашині та бджолині алгоритми симулюють активність методом рою часток. Даний метод оптимізації не вимагає знання градієнта оптимізованої функції. Він був створений для імітації соціальної поведінки, але виявився більш ефективним для оптимізації. За допомогою нього оптимізується функція, утримуючи популяцію часток, які переміщуються в просторі розв'язків за певною формулою [8]. Відповідно до даного методу, кожна частинка, являє собою потенційне рішення задачі, яке рухається в просторі з цілю знаходження оптимального рішення.

Для наочного аналізу мурашиного алгоритму створено симуляцію мурашиної колонії для знаходження оптимального шляху проходження через всі точки зазначені на рисунку 2. Позиції

точок задані величинами координатної площини x, y . Враховано також ситуації, коли координати є від'ємними величинами для коректності дослідження. Дані координати можуть бути також представлені у вигляді неорієнтовного графа з однаковими вагами на всіх ребрах. У рамках демонстрації та для практичного аналізу мурашиного алгоритму дані спрощення не є критичними. На рисунку 4 продемонстровано результати роботи оптимізації мурашиної колонії на всіх ітераціях проходження маршруту. Із отриманої таблиці можна спостерігати, що найкращий шлях було отримано ще при 3 проходженні через усі точки. Наступні 20 кроків отримання найкращих шляхів є рівними між собою. У випадку прикладного використання рішення завдання комівояжера для вибору короткого маршруту з'єднання згадані однакові шляхи можуть використовуватись при паралельній передачі даних, коли, наприклад, 20 комп'ютерів одночасно відправляють повідомлення в спільною мережею в одному отримувачу. Рисунок 4 демонструє конкретні вузли останнього знайденого короткого маршруту передачі даних від вузла 3 до вузла 7 та назад. Слід зазначити, що в отриманому результаті не з'являється жодного повторного проходження вузлами, тобто знайдено гамільтонів шлях.

Після аналізу опису бджолиного алгоритму з'являється гіпотеза для покращення продемонстрованої раніше мурашиної колонії. Особливістю бджолиної колонії є обмін інформацією між частинками рою, чого не роблять мурахи, виходячи з аналізу опису відповідного алгоритму. Як результат, існуючу мурашину колонію удосконалено обміном даними про знайдений шлях іншими частинками рою. Як результат отримано результат з рисунка 5, що демонструє дещо кращі показники величини найкоротшого шляху. Інші риси мурашиної колонії залишаються незмінними. У даному досліді також є 20 рівних між собою коротких шляхи для можливості відправки відповідної кількості повідомлень мережею альтернативними маршрутами. Відповідно до рисунка 6, в удосконаленій версії мурашиного алгоритму також не з'являється жодного повторювального вузла. У даному випадку також знайдено гамільтонів цикл.

Висновки та перспективи подальшого дослідження.

1. З описаних описів алгоритм оптимізації мурашиної колонії може використовуватися для розв'язання складних завдань оптимізації, які передбачають пошук найбільш відповідного рішення для оптимізації цільової функції. Це може бути пошук мінімуму або максимуму ціни, точності, часу, відстані та інших параметрів з дискретної множини можливих рішень.

Метод бджолиної сім'ї дозволяє ефективно розподілити процес на паралельні задачі, що значно прискорює його виконання. Порівняно з генетичним алгоритмом, де є кілька операторів, метод бджолиної сім'ї використовує лише один, який полягає у обчисленні швидкості, яка спрощує його використання. Даний підхід може мати схожу специфіку та сферу застосування як і метод оптимізації мурашиної колонії.

2. З аналізу алгоритму оптимізації за допомогою мурашиної колонії слідує те, що даний підхід може знаходити застосування у задачах календарного планування, маршрутизації транспорту, розподілу ресурсів та багатьох інших сферах. Дослідження показали, що метод мурашиної колонії може приносити результати, які перевершують результати генетичних мурашиних алгоритмів та нейронних мереж.

Методом бджолиної сім'ї легко визначається глобальний мінімум, на відміну від генетичних алгоритмах ця задача ускладнена. Ці методи базуються на різних природних процесах: метод бджолиної сім'ї використовує соціальну поведінку бджіл, а генетичний алгоритм імітує еволюцію та природний відбір. Є можливість об'єднати ці два методи для покращення результатів.

У випадку практичного застосування мурашиного та бджолиного алгоритмів для отримання оптимальних маршрутів для під'єднання у мережі «Інтернет» чи побудови оптимального логістичного шляху маємо важливим є також наявність набору оптимальних маршрутів, які формуються у процесі обрахунків. У випадку перевантаження мережевого чи логістичного шляху у залежності від поставленого завдання комівояжера є можливість використання менш оптимальних шляхів, які не є найвигіднішими серед оптимальних. Запропонований метод додаткового обміну даними між мурахами на кожній ітерації дозволяє деякою мірою покращити ефективність пошуку оптимального шляху. У подальшому даний підхід може застосовуватись для пошуку короткого маршруту відправки повідомлень мережею та покращуватись відповідно до вимог конкретної мережі.

Список бібліографічного опису:

1. S. Kemp, "Digital 2024: Global overview report," DataReportal – Global Digital Insights, Jan. 31, 2024. <https://datareportal.com/reports/digital-2024-global-overview-report>
2. E. L. Lawler, J. K. Lenstra, A H G Rinnooy Kan, and David Bernard Shmoys, The Traveling salesman problem : a guided tour of combinatorial optimization. Chichester West Sussex ; New York: Wiley, 1990.
3. M. J. Apter, The Computer Simulation of Behaviour. Routledge, 2018.
4. Sajad Ahmad Rather, Enhanced Gravitational Search Algorithms for Solving Global Optimization Problems. 2023.
5. S. S. Skiena, The algorithm design manual. London: Springer, 2012.
6. M. Dorigo and StützleT., Ant colony optimization / Ant colony optimization. Cambridge, Mass.: Mit Press, 2004.
7. Aboul Ella Hassanien and Eid Alamry, Swarm intelligence : principles, advances, and applications. Boca Raton, Florida ; London, England ; New York, New York: Crc Press, 2016.
8. R. C. Eberhart, Yuhui Shi, and J. F. Kennedy, Swarm intelligence c : y Russell C. Eberhart, Yuhui Shi, James Kennedy. Morgan Kaufmann, 2001.

References:

9. S. Kemp, "Digital 2024: Global overview report," DataReportal – Global Digital Insights, Jan. 31, 2024. <https://datareportal.com/reports/digital-2024-global-overview-report>
10. E. L. Lawler, J. K. Lenstra, A H G Rinnooy Kan, and David Bernard Shmoys, The Traveling salesman problem : a guided tour of combinatorial optimization. Chichester West Sussex ; New York: Wiley, 1990.
11. M. J. Apter, The Computer Simulation of Behaviour. Routledge, 2018.
12. Sajad Ahmad Rather, Enhanced Gravitational Search Algorithms for Solving Global Optimization Problems. 2023.
13. S. S. Skiena, The algorithm design manual. London: Springer, 2012.
14. M. Dorigo and StützleT., Ant colony optimization / Ant colony optimization. Cambridge, Mass.: Mit Press, 2004.
15. Aboul Ella Hassanien and Eid Alamry, Swarm intelligence : principles, advances, and applications. Boca Raton, Florida ; London, England ; New York, New York: Crc Press, 2016.
16. R. C. Eberhart, Yuhui Shi, and J. F. Kennedy, Swarm intelligence c : y Russell C. Eberhart, Yuhui Shi, James Kennedy. Morgan Kaufmann, 2001.