

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем**

ДОПУЩЕНО ДО ЗАХИСТУ

В.о. завідувача кафедри інтелектуальних
інформаційних систем, канд. техн. наук, доц.

_____ Є. В. Сіденко

« _____ » _____ 202_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**ОПТИМІЗАЦІЯ ЛОГІСТИЧНИХ ОПЕРАЦІЙ
ПІДПРИЄМСТВ МАЛОГО БІЗНЕСУ НА ОСНОВІ
ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.2710816

Виконав студент 6-го курсу, групи 607

_____ *Б. А. Москальський*

« _____ » _____ 202_ р.

Керівник: канд. техн. наук, доцент

_____ *О. В. Козлов*

« _____ » _____ 202_ р.

Миколаїв – 2023

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

Спеціальність **124 «Системний аналіз»**
(шифр і назва)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри інтелектуальних
інформаційних систем, канд. техн. наук,
доцент

_____ Є. В. Сіденко
« _____ » _____ **20** р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Москальському Богдану Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Оптимізація логістичних операцій підприємств малого бізнесу на основі інтелектуальних технологій».
- Керівник роботи Козлов Олексій Валерійович, канд. техн. наук, доцент.
- Затв. наказом Ректора ЧНУ ім. Петра Могили від «03» листопада 2022 р. № 197
2. Строк подання студентом роботи 17 лютого 2023 р.
3. Вхідні (початкові) дані до роботи: результати роботи класичного мурашиного алгоритму; задачі комівояжера; задачі транспортної логістики; 20 пунктів призначення; не більше ніж за 100 ітерацій. Очікуваний результат: система побудови оптимізованого маршруту за допомогою покращеного мурашиного алгоритму.
4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):
 - аналіз сучасного стану проблеми оптимізації логістичних операцій підприємств малого бізнесу;

- дослідження особливостей застосування мурашиних алгоритмів для оптимізації транспортних маршрутів;
- дослідження покращеного мурашиного алгоритму 3-opt;
- порівняльний аналіз результатів застосування базового мурашиного алгоритму та покращеного мурашиного алгоритму 3-opt.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: Електробезпека приміщень.

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці		
Методична частина		

Керівник роботи: канд. техн. наук, доцент О. В. Козлов
(наук. ступінь, вчене звання, прізвище та ініціали)

 (підпис)

Завдання прийнято до виконання Москальський Б.А.
(прізвище та ініціали)

 (підпис)

Дата видачі завдання « 07 » _____ листопада _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН Виконання магістерської кваліфікаційної роботи

Тема: Оптимізація логістичних операцій підприємств малого бізнесу на основі інтелектуальних технологій

№	Найменування роботи	Початок	Закінчен ня	Примітк и
1	Визначення керівника і теми МКР.	01.09.202 2	30.09.202 2	
2	Подання заяви на затвердження теми МКР	03.10.202 2	12.10.202 2	
3	Отримання завдання на виконання МКР	13.10.202 2	10.11.202 2	
4	Складання календарного плану на період виконання МКР	11.11.202 2	15.11.202 2	
5	Огляд літератури за темою дослідження	16.11.202 2	04.12.202 2	
6	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	05.12.202 2	25.12.202 2	
7	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	26.12.202 2	12.01.202 3	
8	Опис фахової частини МКР, зокрема: дослідження сучасного стану проблеми оптимізації логістичних операцій підприємств малого бізнесу; дослідження особливостей застосування мурашиних алгоритмів для оптимізації транспортних маршрутів, розробка системи планування та оптимізації маршрутів на основі покращеного алгоритму мурашиних колоній 3-орт.	13.01.202 3	25.01.202 3	
9	Розробка спеціальної частини з охорони праці та методичної частини	26.01.202 3	02.02.202 3	
10	Попередній захист МКР на засіданні комісії кафедри	03.02.202 3	03.02.202 3	
11	Коригування роботи за результатами	04.02.20	06.02.20	

	попереднього захисту	23	23	
12	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	07.02.20 23	09.02.20 23	
13	Подання МКР рецензенту	09.02.20 23	10.02.20 23	
14	Рецензування МКР	11.02.202 3	12.02.20 23	
15	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.02.20 23	17.02.20 23	
16	Захист МКР перед екзаменаційною комісією (ЕК)	24.02.20 23	24.02.20 23	

Розробив студент Москальський Б. А. _____ (підпис)
 (прізвище та ініціали)

Керівник роботи канд. техн. наук, доцент О. В. Козлов _____ (підпис)
 (наук. ступінь, вчене звання, прізвище та ініціали)

«14» листопада 2022 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 607 ЧНУ ім. Петра Могили

Москальського Богдана Андрійовича

на тему: **“ОПТИМІЗАЦІЯ ЛОГІСТИЧНИХ ОПЕРАЦІЙ ПІДПРИЄМСТВ
МАЛОГО БІЗНЕСУ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ”**

Актуальність даного дослідження полягає у необхідності підвищення ефективності планування і оптимізації транспортних маршрутів із урахуванням часових обмежень, розробці програмного забезпечення з використанням сучасного мета-евристичного метода покращеної мурашиної колонії 3-орт для вирішення задачі комівояжера, що призведе до пришвидшення прокладання маршруту системою, зменшення витрат на транспортування продукту та підвищення якості обслуговування клієнтів.

Об'єктом роботи є процеси оптимізації логістичних операцій підприємств малого бізнесу.

Предметом роботи є методи, моделі та програмні засоби для оптимізації логістичних операцій підприємств малого бізнесу.

Метою роботи є підвищення ефективності системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму.

Для досягнення поставленої мети були визначені такі задачі:

- аналіз особливостей та характеристик сучасних систем оптимізації логістичних операцій підприємств малого бізнесу;
- аналіз та узагальнення існуючих методів оптимізації різнотипних логістичних операцій;

- дослідження різних модифікацій мурашиного алгоритму та особливостей їх застосування для вирішення задачі комівояжера;
- розробка системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt;
- дослідження ефективності системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt;

В результаті виконання роботи було досліджено і порівняно два мета-евристичних методи оптимізації класичний мурашиний алгоритм і покращений мурашиний алгоритм 3-opt, проаналізовано вплив їх внутрішніх параметрів на роботу алгоритмів, визначені основні їх переваги та недоліки, а також розроблено програмне забезпечення, в якому реалізовані відповідні методи.

Дана робота складається з п'яти розділів. Кожен розділ відповідно присвячений: аналіз сучасного стану проблеми оптимізації логістичних операцій підприємств малого бізнесу, дослідження особливостей застосування мурашиних алгоритмів для оптимізації транспортних маршрутів, покращений мурашиний алгоритм і переваги його застосування, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи — 109 сторінок. Магістерська кваліфікаційна робота містить 0 додаток, 17 рисунків, 12 таблицю і посилання на 51 літературних джерел.

Ключові слова: задача комівояжера, оптимізація транспортних маршрутів, TSP, АСО-метод, 3-opt.

ABSTRACT

to the master's qualification work by the student of the group 607 of Petro Mohyla
Black Sea National University

Moskalskyi Bohdan

“OPTIMIZATION OF LOGISTICS OPERATIONS OF SMALL BUSINESS ENTERPRISES BASED ON INTELLIGENT TECHNOLOGIES”

The relevance of this study lies in the need to increase the efficiency of planning and optimization of transport routes taking into account time constraints, software development using the modern meta-heuristic method of the improved ant colony 3-opt for solving the traveling salesman problem, which will lead to faster routing by the system, reducing costs for product transportation and improving the quality of customer service.

The **object** of the work is the optimization processes of logistics operations of small business enterprises.

The **subject** of the work is methods, models and software tools for optimizing logistics operations of small business enterprises.

The **purpose** of the work is to improve the efficiency of the system for optimizing logistics operations of small businesses based on an improved ant algorithm.

For the set goal, the following tasks were achieved:

- analysis of features and characteristics of modern systems for optimizing logistics operations of small businesses;
- analysis and generalization of existing optimization methods of various types of logistics operations;
- research of various modifications of the ant algorithm and the features of their application to solve the tasks of the traveling salesman;

- development of a system for optimizing logistics operations of small businesses based on the improved 3-opt ant algorithm;
- study of the effectiveness of the system for optimizing logistics operations of small business enterprises based on the improved 3-opt ant algorithm;

As a result of the work, two meta-heuristic methods of optimization, the classic ant algorithm and the improved 3-opt ant algorithm, were investigated and compared, the influence of their internal parameters on the operation of the algorithms was analyzed, their main advantages and disadvantages were determined, and software was developed in which the appropriate methods.

This work consists of five sections. Each section is respectively dedicated to: analysis of the current state of the problem of optimizing logistics operations of small business enterprises, research on the peculiarities of the use of ant algorithms to optimize transport routes, an improved ant algorithm and the advantages of its application, labor protection, the methodological part of the master's work. The total volume of work is 109 pages. The master's thesis contains 0 appendix, 17 figures, 12 tables and references to 51 literary sources.

Key words: traveling salesman's problem, optimization of transport routes, TSP, ACO-method, 3-opt.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП	5
1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ88 ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ОПЕРАЦІЙ ПІДПРИЄМСТВ МАЛОГО БІЗНЕСУ	7
1.1 Аналіз сучасних задач транспортної логістики на підприємствах різних галузей	7
1.2 Існуючі системи оптимізації логістичних операцій підприємств малого бізнесу	11
1.3 Особливості та перспективи застосування інтелектуальних методів та алгоритмів для оптимізації логістичних операцій	13
Висновки до розділу 1	15
2 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ МУРАШИНИХ АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ МАРШРУТІВ	17
2.1 Дослідження задачі пошуку оптимального шляху при плануванні транспортних маршрутів	17
2.2 Аналіз математичних методів оптимізації транспортних маршрутів	27
2.3 Особливості застосування базового мурашиного алгоритму для вирішення задачі пошуку оптимального шляху	29
2.4 Модифікації мурашиного алгоритму	37
Висновки до розділу 2	40
3 РОЗРОБКА СИСТЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ МАРШРУТІВ НА ОСНОВІ АЛГОРИТМУ ПОКРАЩЕНОЇ МУРАШИНОЇ КОЛОНІЇ	41
3.1 Особливості алгоритму графа згорткової мережі покращеної мурашиної колонії	41
3.2 Порівняння результатів роботи базового і покращеного мурашиних алгоритмів	61
3.3 Проектування системи та її інтерфейсу користувача	67
Висновки до розділу 3	71
4 МЕТОДИЧНИЙ РОЗДІЛ	73
5 СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ ТА БЕЗПЕКИ У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	83
ВИСНОВКИ	101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	104

ПЕРЕЛІК СКОРОЧЕНЬ

TSP	– задача комівояжера
ACO	– мурашина колонія
LP	– лінійне програмування
IP	– цілочисельне програмування
NFO	– оптимізація мережевого потоку
GCNIACO	– граф згорткової мережі покращеної мурашиної колонії
TSP	– traveling salesman problem
ACO	– ant colony optimization
LP	– linear programming
IP	– integer programming
NFO	– network flow optimization
GCNIACO	– graph convolutional network improved ant colony optimization

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«ОПТИМІЗАЦІЯ ЛОГІСТИЧНИХ ОПЕРАЦІЙ ПІДПРИЄМСТВ МАЛОГО БІЗНЕСУ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ»

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.2710816

Виконав студент 6-го курсу, групи 607

_____ *Б. А. Москальський*

«___»_____ 202_ р.

Керівник: канд. техн. наук, доцент

_____ *О. В. Козлов*

«___»_____ 202_ р.

Миколаїв – 2023

ВСТУП

На сьогодні в Україні розвивається безліч різних видів бізнесу: служби доставки, спеціалізовані магазини, ресторани, бари, пошти та інші сервіси. Крім того, впродовж війни, у місті Миколаїв та в інших прифронтових містах багато людей потребують гуманітарної допомоги, у тому числі доставки питної і технічної води для мешканців різних районів. Наведені послуги і сервіси потребують координації доставки продуктів, що може здійснюватися власноруч або користуючись послугами інших перевізників.

Доставка відбувається за різних умов, за різними адресами, різною кількістю товарів і транспорту різних характеристик. Міжміські перевезення мають чіткий, здебільшого, прямий маршрут по автостраді, і виділеним смугам для важкого вантажного транспорту, у той час як у місті, маленькі транспортні засоби можуть переміщатись за будь якими дозволеними напрямками, які можуть мати різну вартість, важкість та інші характеристики.

Подібна робота полегшується завдяки відповідним технічним рішенням, застосункам та сервісам, що систематизують логістичні операції, такі як: перевезення, розподіл вантажу, підбір транспортних засобів, регулювання навантаження транспортної мережі тощо.

Однією з найбільш складних та важливих задач транспортної логістики є задача пошуку оптимального маршруту. Такі сервіси як ANT Logistics використовують мурашині алгоритми для розв'язання даної задачі, однак, вони мають ряд недоліків. Таким чином, існує нагальна потреба у покращенні вже розроблених систем та застосунків для пошуку оптимальних маршрутів за рахунок розробки та впровадження удосконалених інтелектуальних алгоритмів оптимізації.

Об'єктом дослідження є процеси оптимізації логістичних операцій підприємств малого бізнесу.

Предметом дослідження є методи, моделі та програмні засоби для оптимізації логістичних операцій підприємств малого бізнесу.

Метою роботи є підвищення ефективності процесів оптимізації логістичних операцій підприємств малого бізнесу за рахунок розробки системи планування маршрутів на основі покращеного мурашиного алгоритму.

Для досягнення поставленої мети були визначені такі задачі:

- аналіз особливостей та характеристик сучасних систем оптимізації логістичних операцій підприємств малого бізнесу;
- аналіз та узагальнення існуючих методів оптимізації різнотипних логістичних операцій;
- дослідження різних модифікацій мурашиного алгоритму та особливостей їх застосування для вирішення задачі комівояжера;
- розробка системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt;
- дослідження ефективності системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt.

Апробація результатів магістерської роботи. Основні результати магістерських досліджень доповідались та обговорювались на Науково-практичній конференції «Могилянські читання – 2022», підсекція «Інтелектуальні інформаційні системи» (2022 р., ЧНУ, ім. Петра Могили, Миколаїв).

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ОПЕРАЦІЙ ПІДПРИЄМСТВ МАЛОГО БІЗНЕСУ

1.1 Аналіз сучасних задач транспортної логістики на підприємствах різних галузей

Наш світ давно функціонує за ринковими відносинами, продаж і купівля, товари і послуг тощо, це функціонування потребує налагодженої системи доправлення, постачання, зберігання тощо. Саме логістика відповідає за ці функції, координування шляхів, перевезення продуктів, розподілення їх по складах, підбір транспортних маршрутів, контроль потоку транспортування. Ці задачі логістики можна перетворити на математичні моделі, які зможе розв'язати програма написана людиною.

Для успішного керування логістикою підприємства необхідно вдало вирішувати логістичні задачі які пов'язані з наступними аспектами:

- транспортування;
- розподілення;
- призначення;
- потік транспортної мережі;
- маршрути.

Наведені вище задачі вирішуються теорією оптимізації. Найбільш розвинутою областю теорії оптимізації є лінійна оптимізація чи лінійне програмування (Linear Programming), де розглядаються оптимізаційні задачі із лінійними цільовими функціями і областю пошуку, що задається лінійними обмеженнями, тобто є багатогранником чи багатогранною областю. Це пояснюється тим, що лінійне програмування має надзвичайно широку сферу

практичного використання у теорії прийняття рішень, дослідженні операцій і оптимальному плануванні [1].

1.1.1 Транспортна задача

Транспортна задача полягає у пошуку найбільш вигідного плану перевезення однорідного продукту з пунктів виробництва (чи зберігання) до пунктів споживання, тобто від постачальників до споживачів, ефективність якого будемо оцінювати за критерієм найменшої вартості перевезення. Транспортна задача – це специфічна задача лінійного програмування.

Необхідно скласти такий план перевезення, щоб вивезти всю продукцію від постачальників, задовольнити потреби всіх споживачів і сумарна вартість перевезення при цьому має бути мінімальною. Окреслена постановка задачі вимагає виконання рівності загальної суми запасу вантажу загальній сумі потреб в ньому. Якщо в транспортній задачі умова виконується, то таку транспортну задачу називають закритою (з правильним балансом). Якщо ж рівність не виконується, то транспортну задачу називають відкритою (з неправильним балансом) [2].

1.1.2 Розподільча задача

Розподільча логістика – це управління транспортуванням, складуванням та іншими матеріальними і нематеріальними операціями, які здійснюються в процесі доведення готової продукції до споживача згідно з інтересами і вимогами останнього, а також передачі, зберігання й обробки відповідної інформації. Інакше її ще називають маркетинговою або збутовою логістикою. Доцільно все ж використовувати термін «розподільча логістика» як такий, що найбільш точно відображає наявність у логістичній системі керуючих впливів під час доведення готової продукції до кінцевих споживачів.

Принципова відмінність розподільчої логістики від традиційного розуміння збуту полягає насамперед у системному взаємозв'язку процесу розподілу з

процесами виробництва і закупівель під час управління матеріальними потоками, а також системному взаємозв'язку всіх функцій всередині самого розподілу [3].

1.1.3 Задача про призначення

Задача про призначення – це одне із завдань лінійного програмування, яка призначена для оптимального підбору n "пропозицій" до n "потребам", наприклад, для призначення виду роботи машині, призначення виду роботи виробничому відділу, призначення людини на посаду і тощо.

Метод призначень застосовується при вирішенні задач, що мають такі характеристики:

1. Є n "предметів", які повинні бути розподілені по n "пунктів призначення".
2. Кожен "предмет" повинен бути призначений єдиному "пункту призначення". У поняття "предмет" і "пункт призначення" може вкладатися різне смислове значення, яке визначається конкретним завданням менеджменту. Так як предмет може виступати певний вид діяльності (роботи), посада, людина і тощо.
3. Оптимальний підбір призначень повинен бути досягнутий за рахунок максимізації або мінімізації певної міри ефективності призначення: прибутку або вартості. Для кожного потенційного призначення оцінюється міра ефективності. Якщо мірою ефективності є прибуток, то в процесі рішення задачі про призначення вона максимізується, якщо мірою ефективності є вартість, вона мінімізується [4].

1.1.4 Задача про максимальний потік на транспортній мережі

Методи вирішення задачі про максимальний потік на мережі застосовуються на транспортних, комунікаційних, електричних мережах, при моделюванні різних процесів фізики і хімії, в деяких операціях над матрицями, для вирішення родинних завдань теорії графів, і навіть для пошуку Web-груп в WWW. У завданнях на максимальний потік виконується переміщення об'єктів будь-якої

природи через мережу. Розподіляються чи нафтогазові продукти, чи іграшки і одяг в магазини уцінених товарів по автомагістралях країни, чи біти інформації по мережах зв'язку для відображення на моніторах у всіх куточках світу – по суті, це одна і те ж завдання. Алгоритми їх розв'язання широко застосовуються як окремі продукти, так і в численних додатках [5].

При аналізі транспортних мереж часто виникає задача визначення максимального потоку, що може пропустити дана мережа, а також задача розподілу цього потоку по дугах мережі.

З математичної точки зору задача про максимальний потік формулюється в такий спосіб: при заданій конфігурації мережі і відомої пропускної здатності знайти не негативні значення, що задовольняють умовам і, що максимізують функцію.

Алгоритм для знаходження максимального потоку був запропонований Фордом і Фалкерсоном і полягає в поступовому збільшенні потоку, що пропускається по мережі, доти, поки він не стане найбільшим. Алгоритм заснований на теоремі Форда-фалкерсона: у будь-якій транспортній мережі максимальний потік із джерела в стік, дорівнює мінімальній пропускній здатності розрізу [6].

1.1.5 Задача комівояжера

Задача комівояжера – одна з основних задач комбінаторної оптимізації дослідження операцій і теоретичної інформатики що полягає у знаходженні найвигіднішого маршруту, який проходить через вказані точки хоча б один раз, після чого вертаючись у вихідну точку.

Проблема комівояжера зустрічається в чистому вигляді в багатьох практичних застосуваннях, наприклад, у плануванні маршрутів, у логістиці або при проектуванні мікросхем. Однак це трапляється ще частіше як підпроблема, наприклад, у розподілі товарів, у плануванні клієнтів або маршрутів

обслуговування поломок або в секвенуванні геному. Терміни «місто» та «відстань» не слід розуміти буквально, скоріше міста означають, наприклад, клієнтів, яких потрібно відвідати, свердловини або часткові ланцюги ДНК, тоді як відстань означає час у дорозі, витрати або ступінь, до якого дві ДНК пасма збігаються. У багатьох практичних застосуваннях також необхідно враховувати додаткові умови, такі як часові вікна або обмежені ресурси, що значно ускладнює вирішення проблеми.

1.2 Існуючі системи оптимізації логістичних операцій підприємств малого бізнесу

На даний момент існують сервіси які оптимізують логістичні зв'язки, серед яких обрано трьох представників:

TMS Planagy – система автоматизації транспорту, управління транспортом, управління ланцюгами транспортної логістики та управління доставками, яка може бути самостійною системою або входити у систему планування ресурсів підприємства (ERP) [7].

ANT Logistics – український хмарний TMS-сервіс для автоматизації транспортних перевезень [8].

OptimoRoute – онлайн-планувальник маршрутів для доставки та обслуговування на місцях [9].

У табл. 1.1 наведено порівняльний аналіз функціоналу перелічених вище сервісів, який демонструє, що розглянуті застосунки мають певні обмеження та потребують доопрацювання.

Таблиця 1.1 – Порівняння сервісів аналогів

Функціонал	ANT Logistics	TMS Planarry	OptimoRoute
Мобільний застосунок	ТАК	ТАК	ТАК
Маршрути за адресами	ТАК	ТАК	ТАК
Редагування активних маршрутів	НІ	НІ	ТАК
Розподіл за типом товарів	ТАК	НІ	НІ
Розрахунок вартості перевезення від типу палива	НІ	НІ	НІ
Розрахунок за умов зовнішнього впливу	НІ	НІ	НІ
Вибір різних алгоритмів	НІ	НІ	НІ

У табл. 1.2 наведено особливості застосунків, які розкривають розробники на офіційних сайтах. Усі наведені застосунки автоматично оптимізують маршрути і мають мобільні застосунки. Зокрема, застосунок TMS Planarry повністю виключає втручання людей у побудову маршрутів, коли ж інші дозволяють редагувати їх.

Таблиця 1.2 – Особливості сервісів аналогів

ANT Logistics	TMS Planarry	OptimoRoute
1. Маршрути за гео-зонами	1. Точне планування робочого часу	1. Автоматизоване планування
2. Маршрути за адресними групами	2. Точний розрахунок відстаней в маршрутних листах	2. Відстеження тривалості роботи водія
3. Багаторейсова маршрутизація	3. Мінімізація кількості використовуваних автомобілів	3. Мобільний додаток для водіїв
4. Далекомагістральна маршрутизація/кросс-докинг	4. Мобільний додаток	4. Відстеження замовлень у реальному часі
5. Групи сумісності товарів		5. Модифікація маршруту в реальному часі
6. Мобільний додаток		

Сервіс ANT Logistics переважає над іншими ширшим вибором налаштувань оптимізації перевезень, у ньому доступні такі функції як, маршрути за гео-зонами, багаторейсова маршрутизація, групи сумісності товарів, які не реалізовані іншими сервісами. Однак, даний сервіс має і ряд недоліків, таких як відсутність налаштування для розрахунку за умов внутрішнього впливу та розрахунку за типом палива. Наявність наведених опцій може якісно вплинути на оптимізацію логістичних зв'язків за умов пересування у місті.

1.3 Особливості та перспективи застосування інтелектуальних методів та алгоритмів для оптимізації логістичних операцій

Логістичні операції відіграють вирішальну роль в успіху будь-якого бізнесу. Він передбачає управління потоком товарів, інформації та ресурсів від точки походження до точки споживання. У міру зростання бізнесу їхні логістичні операції стають дедалі складнішими, що призводить до таких проблем, як високі витрати, зниження ефективності та зниження задоволеності клієнтів. Щоб

вирішити ці проблеми, багато компаній звертаються до інтелектуальних методів і алгоритмів для оптимізації своїх логістичних операцій.

Найважливішим аспектом у всій роботі виступає: підвищення ефективності. Використання алгоритмів і машинного навчання може допомогти компаніям оптимізувати свої логістичні операції, що призведе до підвищення ефективності, зниження витрат і підвищення продуктивності. Наприклад, алгоритми можна використовувати для оптимізації маршрутів доставки, скорочення часу доставки та мінімізації витрат на паливо.

Також оптимізація роботи для прийняття рішень у режимі реального часу якісно вплине на ефективність роботи. Алгоритми можуть аналізувати великі обсяги даних у режимі реального часу та надавати можливості прийняття рішень у режимі реального часу, що дозволяє підприємствам швидко реагувати на зміни обставин. Наприклад, алгоритми можна використовувати для коригування графіків доставки в режимі реального часу залежно від умов дорожнього руху, гарантуючи, що відправлення надходять вчасно.

Але для того щоб запобігти певним помилкам необхідно покращити прогнозування. Використання прогнозової аналітики може допомогти підприємствам робити кращі прогнози щодо попиту та пропозиції, дозволяючи їм приймати обґрунтовані рішення щодо виробництва, управління запасами та транспортування. Наприклад, прогнозу аналітику можна використовувати для прогнозування попиту на конкретні продукти, дозволяючи підприємствам оптимізувати рівень своїх запасів і зменшити відходи.

Алгоритми оптимізації можуть допомогти компаніям знайти найефективніші маршрути доставки, скоротивши час доставки та підвищивши задоволеність клієнтів. Аналізуючи схеми трафіку, графіки доставки та інші фактори, алгоритми можуть визначати найкращі маршрути для доставки, скорочуючи час доставки та покращуючи ефективність доставки.

Використання систем відстеження та моніторингу в реальному часі може надати підприємствам покращену видимість їхніх логістичних операцій, дозволяючи їм виявляти вузькі місця та приймати обґрунтовані рішення. Наприклад, системи відстеження в реальному часі можуть надавати компаніям інформацію в режимі реального часу про місцезнаходження та статус їхніх відправлень, дозволяючи їм швидко реагувати на будь-які проблеми, які можуть виникнути.

Алгоритми можуть автоматизувати різні логістичні операції, звільняючи час співробітників, щоб зосередитися на більш стратегічних завданнях. Наприклад, алгоритми можна використовувати для автоматизації планування доставки та самовивозу, зменшуючи потребу в ручному втручанні та звільняючи час співробітників, щоб зосередитися на більш стратегічних завданнях.

Використання інтелектуальних методів і алгоритмів у логістичних операціях має потенціал для трансформації галузі та дозволить підприємствам стати більш ефективними, гнучкими та орієнтованими на клієнта. Крім того, використання інтелектуальних методів і алгоритмів може допомогти підприємствам робити кращі прогнози, зменшуючи відходи та покращуючи управління запасами.

Висновки до розділу 1

У даному розділі розглянуто сучасні задачі транспортної логістики як: транспортування, розподілення, призначення, потік транспортної мережі, задачу комівояжера. Проведено порівняльний аналіз існуючих систем вирішення цих задач, а саме: TMS Planarry, ANT Logistics, OptimoRoute. Досліджено особливості та перспективи застосування інтелектуальних методів та алгоритмів для оптимізації логістичних операцій.

Підсумовуючи, використання інтелектуальних методів і алгоритмів у логістичних операціях пропонує підприємствам потужний інструмент для оптимізації їх операцій, зниження витрат і підвищення рівня задоволеності

клієнтів. Використовуючи потужність алгоритмів і машинного навчання, підприємства можуть оптимізувати свої логістичні операції в режимі реального часу, приймаючи кращі рішення та швидко реагуючи на зміни обставин. Хоча ці методи й алгоритми можуть принести значні переваги, їх впровадження також потребує ретельного планування та виконання, а також інвестицій у необхідні технології та інфраструктуру.

Таким чином для досягнення поставленої мети були визначені такі задачі:

- аналіз особливостей та характеристик сучасних систем оптимізації логістичних операцій підприємств малого бізнесу;
- аналіз та узагальнення існуючих методів оптимізації різнотипних логістичних операцій;
- дослідження різних модифікацій мурашиного алгоритму та особливостей їх застосування для вирішення задачі комівояжера;
- розробка системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt;
- дослідження ефективності системи оптимізації логістичних операцій підприємств малого бізнесу на основі покращеного мурашиного алгоритму 3-opt.

2 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ МУРАШИНИХ АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ МАРШРУТІВ

2.1 Дослідження задачі пошуку оптимального шляху при плануванні транспортних маршрутів

Дослідження пошуку оптимального шляху для планування транспортних маршрутів називається оптимізацією мережі або оптимізацією маршруту. Це передбачає пошук найбільш ефективного та рентабельного маршруту для транспортування товарів або пасажирів від джерела до пункту призначення, беручи до уваги такі фактори, як відстань, час, споживання палива, трафік і вартість. У цій галузі використовуються математичні моделі та алгоритми, такі як лінійне програмування, цілочисельне програмування та теорія графів, щоб знайти оптимальне рішення. Оптимізація мережі має широкий спектр застосувань у логістиці, транспортуванні та управлінні ланцюгами поставок.

2.1.1 Особливості задачі комівояжера

Задача комівояжера має широке прикладне застосування у транспортних системах, автоматизованому проектуванні, тестуванні та виготовленні інтегрованих схем, виробництві друкованих плат, лазерній нарізці пластмас і металів, дослідженні структури білка, технологіях вишивання, зварювання та малювання неперервною лінією та інших галузях.

Вона формулюється таким чином: комівояжер повинен відвідати ряд міст, відстані між якими відомі. Комівояжер вибирає найкоротший замкнений маршрут, що починається й закінчується у місті його проживання, при цьому він повинен відвідати необхідне місто один і лише один раз [10, с. 166].

Задача комівояжера (TSP) – це особливий тип задачі оптимізації мережі, яка передбачає пошук найкоротшого можливого маршруту, який відвідує заданий набір міст і повертається до початкового міста. У контексті транспортного планування TSP можна розглядати як окремий випадок загальнішої задачі оптимізації маршруту, де метою є відвідування набору попередньо визначених місць найбільш ефективним способом. Однак у багатьох реальних сценаріях планування транспортування можуть існувати додаткові обмеження та цілі, які необхідно враховувати, наприклад обмежена пропускна спроможність, часові вікна та численні депо, що може значно ускладнити проблему. Таким чином, незважаючи на те, що TSP є добре відомою та широко дослідженою проблемою оптимізації мережі, вона не обов'язково є репрезентативною для всіх проблем, які виникають під час планування маршруту транспортування.

Вирішення задачі комівояжера є важливим для транспортної логістики, оскільки воно може допомогти оптимізувати маршрути вантажівок, фургонів або інших транспортних засобів, скорочуючи час і витрати, пов'язані з транспортуванням. Знаходячи найкоротший маршрут, який відвідує всі пункти призначення та повертається до початкової точки, TSP може допомогти мінімізувати загальну пройдену відстань, споживання палива та інші витрати, пов'язані з транспортом. Це може суттєво вплинути на загальну ефективність і прибутковість логістики або доставки, особливо для компаній, яким потрібно здійснювати кілька поставок на день або обслуговувати велику кількість клієнтів.

Крім того, у сучасному швидкоплинному та конкурентному бізнес-середовищі надання своєчасних та ефективних послуг доставки стає все більш важливим. Знайшовши оптимальний маршрут, компанії можуть скоротити час доставки та підвищити рівень задоволеності клієнтів, допомагаючи зміцнити лояльність до бренду та залучити новий бізнес.

Підводячи підсумок, можна сказати, що вирішення TSP є важливим для транспортної логістики, оскільки воно може допомогти зменшити витрати,

скоротити терміни доставки та підвищити рівень задоволеності клієнтів, що є критично важливими факторами успіху будь-якої доставки або логістичної операції.

2.1.2 Математична постановка задачі комівояжера

Розглянемо постановку задачі комівояжера на прикладі транспортної системи. Транспортна мережа налічує $(n + 1)$ пункт. Відомі відстані між пунктами c_{ij} , $i, j = 0, n$. Вийжджаючи з початкового пункту (йому приписується номер 0), комівояжер повинен побувати в усіх інших пунктах лише один раз і повернутися в пункт 0. Задача комівояжера знаходить відповідь на питання: в якому порядку потрібно об'їжджати пункти, щоб пройдена сумарна відстань була мінімальною?

Задачу комівояжера можна сформулювати як задачу цілочислового лінійного програмування. Введемо змінні x_{ij} , $i = \overline{0, n}$, $j = \overline{0, n}$; $i \neq j$, що мають такий зміст: $x_{ij} = 1$, якщо комівояжер після пункту i потрапляє в пункт j , або $x_{ij} = 0$, протилежному разі.

Також введемо змінні u_i , u_j ($i, j = \overline{1, n}$), що дозволять сформулювати умову зв'язності маршруту комівояжера: виключити розпадання маршруту на підцикли. Тоді математична модель задачі набуває вигляду:

$$\sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \rightarrow \min_{x_{ij} \in \Omega}; \quad (2.1)$$

$$\Omega: \sum_{j \neq i, j=0}^n x_{ij} \rightarrow 1, i = \overline{0, n}; \quad (2.2)$$

$$\sum_{j \neq i, j=0}^n x_{ij} \rightarrow 1, j = \overline{0, n}; \quad (2.3)$$

$$u_i - u_j + nx_{ij} \leq n - 1, j = \overline{1 \dots n}; \quad (2.4)$$

$$u_i, u_j = \text{int}, j, i = \overline{1 \dots n}; \quad (2.5)$$

$$x_{ij} + x_{ji} \leq 1, i, j = \overline{0 \dots n}, i \neq j. \quad (2.6)$$

Формула (2.1) визначає цільову функцію як сумарну довжину маршруту комівояжера. В умові (2.2) зазначено, що комівояжер в'їжджає у кожний пункт лише один раз, умова (2.3) – що комівояжер виїжджає з кожного пункту лише один раз. Обмеження (2.4) вимагає, щоб будь-який маршрут комівояжера складався з одного циклу. Система рівностей (2.5) обмежує область допустимих значень додаткових змінних цілими числами (додатними чи від'ємними). Останнє обмеження (2.6) виключає повернення комівояжера до пункту, в якому він уже побував [10, с.167].

2.1.3 Методи розв'язку задачі комівояжера

Планування транспортних маршрутів стосується процесу визначення найбільш ефективного та рентабельного способу перевезення вантажів або пасажирів з одного місця в інше. Це передбачає пошук маршруту, який відповідає конкретним цілям, таким як мінімізація часу подорожі, відстані, споживання палива або вартості, враховуючи при цьому різноманітні обмеження, такі як дорожні умови, схеми руху, графіки доставки та місткість транспортних засобів.

Планування транспортних маршрутів є критично важливим аспектом логістики та управління ланцюгом поставок, оскільки воно безпосередньо впливає

на ефективність, надійність і економічну ефективність транспортної операції. Ефективне планування маршруту може допомогти зменшити транспортні витрати, скоротити час доставки та підвищити задоволеність клієнтів, що може призвести до підвищення прибутковості та конкурентоспроможності.

Процес планування транспортних маршрутів зазвичай включає збір даних про місцезнаходження клієнтів або пункти призначення доставки, пропускну спроможність і обмеження транспортних засобів, а також будь-яку відповідну інформацію про дорожні умови, схеми руху та графіки доставки. Потім ця інформація використовується для розробки математичних моделей і алгоритмів, які можуть знайти оптимальний маршрут, який відповідає конкретним цілям і обмеженням транспортної проблеми.

Існує кілька методів планування транспортних маршрутів:

- евристичний;
- генетичні алгоритми;
- програмування обмежень;
- машинне навчання;
- мурашиний алгоритм.

Евристика – це набір правил і стратегій, які можна використовувати для пошуку кращого рішення складної проблеми оптимізації за відносно короткий проміжок часу. Звичайні евристики, що використовуються в оптимізації маршруту, включають алгоритми найближчого сусіда, вставки та кластеру перший-другий маршрут.

Генетичні алгоритми – це методи оптимізації, засновані на принципах еволюції та природного відбору. Їх можна використовувати для пошуку майже оптимальних рішень складних задач оптимізації, таких як TSP.

Програмування обмежень – це метод формулювання та розв’язання задач оптимізації шляхом представлення проблеми як набору обмежень і використання засобу вирішення обмежень для пошуку оптимального рішення.

Машинне навчання. Алгоритми машинного навчання, такі як нейронні мережі та дерева рішень, можна навчити передбачати оптимальний маршрут на основі історичних даних, беручи до уваги такі фактори, як схеми руху, дорожні умови та графіки доставки.

Мурашиний алгоритм (ACO) – це метод метаевристичної оптимізації, натхненний поведінкою колоній мурашок. Його можна використовувати для пошуку майже оптимальних рішень TSP та інших проблем оптимізації.

Це лише кілька прикладів різних методів, які можна використовувати для планування транспортних маршрутів. Вибір методу залежатиме від конкретних вимог транспортної задачі, таких як розмір проблеми, складність обмежень і доступні обчислювальні ресурси.

2.1.4 Розв’язання задачі комівояжера методом гілок і меж

Розглянемо повний симетричний орієнтований граф (X, U) , де $X = \{0, 1, \dots, n\}$ – множина вершин; U – множина дуг. Кожній дузі (i, j) графа приписане число σ_{ij} довжини дуги. Потрібно знайти контур, що проходить через кожную вершину лише один раз (гамільтоновий контур), який має найменшу довжину. Під довжиною контура розуміємо величину, що дорівнює сумі довжин дуг σ_{ij} . Це і буде маршрут комівояжера.

Спочатку для безлічі всіх гамільтонових контурів R визначається деяка оцінка знизу (нижня межа) $\varphi_{(R)}$ довжини контура. Потім безліч усіх гамільтонових контурів розбивається на дві підмножини. Перша підмножина складається з гамільтонових контурів, які включають певну дугу (i, j) . Позначимо цю множину $\{(i, j)\}$. Друга множина складається з гамільтонових контурів, які не включають цю дугу. Позначимо її $\overline{\{(i, j)\}}$. Для кожної з підмножин $\{(i, j)\}$ і $\overline{\{(i, j)\}}$ визначається нижня межа довжини гамільтонових

контурів $\varphi_{(i,j)}$ і $\overline{\varphi(i,j)}$. Кожна нова межа виявляється не меншою від нижньої межі всієї множини гамільтонових контурів $\varphi_{(R)}$.

Серед двох підмножин контурів $\{(i,j) \text{ і } \overline{\{(i,j)\}}\}$ вибирається підмножина з меншою нижньою межею. Ця підмножина знову розбивається на дві підмножини. Для новостворених підмножин знаходиться нижня межа. Процес розбиття підмножин аналогічним чином продовжується до того часу, поки не буде виділено підмножину, що містить єдиний гамільтонів контур. Взаємозв'язок підмножин, одержаних у результаті розбиття, зображується у вигляді дерева, вершинам якого приписуються нижні межі.

Одержавши гамільтонів контур, переглядають обірвані гілки дерева і порівнюють нижні межі множин, що відповідають обірваним гілкам, із довжиною одержаного гамільтонового контура (рекорду). Якщо нижні межі підмножин, що відповідають обірваним гілкам, виявляються меншими від рекорду, то ці гілки розвивають за тим самим правилом. У результаті розвитку гілок можуть бути одержані нові гамільтонові контури. У цьому разі рекорд береться таким, що дорівнює найменшій із довжин гамільтонових контурів.

Розв'язання задачі вважається закінченим, якщо нижні межі названих гілок виявляються не меншими від рекорду. Як оптимальний контур вибирається контур із найменшою довжиною. Розрахунок нижніх меж ґрунтується на такій властивості. Якщо знайти довжину оптимального контура з матрицею відстаней A , а потім з елементів деякого рядка або стовпця матриці A відняти деяке число a і знову розв'язати задачу з новою матрицею, то контур не зміниться, а довжина його зменшиться на це число a . Зміна всіх елементів рядка або стовпця на одне й те саме число не впливає на оптимальне розв'язання задачі. Якщо операцію віднімання проробити й для інших рядків та стовпців, то довжина оптимального контура зі зміненою матрицею буде відрізнятися від довжини оптимального

контура з вихідною матрицею на суму чисел, що віднімаються від елементів рядків і стовпців.

Тому для визначення нижньої межі множини всіх гамільтонових контурів необхідно в кожному рядку матриці A знайти (2.7):

$$a_i = \min_{1 \leq j \leq n} \{a_{ij}\}. \quad (2.7)$$

Потім необхідно відняти це значення від усіх елементів цього рядка (операція зведення матриці відстаней по рядках). У результаті зведення матриці в кожному її рядку буде принаймні по одному нулю (одержана матриця A^*). Потім у матриці, зведеній по рядках, знаходимо найменший елемент (2.8):

$$\beta_i = \min_{1 \leq j \leq n} \{a_{ij}^*\}. \quad (2.8)$$

У кожному стовпці матриці A^* (операція зведення матриці відстаней за стовпцями), a_i, β_i – константи зведення. Повністю зведена матриця містить, принаймні, по одному нулю в кожному рядку і кожному стовпці.

Оскільки довжина оптимального контура L_1 в задачі з повністю зведеною матрицею відрізняється від довжини оптимального контура L в задачі з незведеною матрицею на суму констант зведення (2.9):

$$\gamma = \sum_{i=1}^n a_i + \sum_{j=1}^n \beta_j, \quad (2.9)$$

то $L = L_1 + \gamma$.

У повністю зведений матриці всі елементи невід'ємні, тому $L_1 > 0$, а γ можна вибрати як нижню межу гамільтонового контура, тобто припустити, що $\varphi_{(R)} = \gamma$.

Розглянемо спосіб вибору дуги (i, j) , включення або невключення якої в контур розбиває безліч гамільтонових контурів на підмножини $\{(i, j)$ та $\overline{\{(i, j)\}}$. Виключення дуги (i, j) з гамільтонова контура здійснюється заміною відповідного елемента матриці відстаней на ∞ . У результаті виключення з'являється можливість виконати додаткове зведення в матриці і покращити межу.

Включення дуги (i, j) в гамільтонів контур дозволяє скоротити розмір матриці за рахунок викреслювання i -го рядка та j -го стовпця. Крім того, при включенні дуги (i, j) в гамільтонів контур з'являється можливість утворення негамільтонового контура, тобто контура, що проходить через частину вершин. Тому з метою запобігання утворенню такого контура необхідно виключити з розгляду одну з дуг. У найпростішому випадку при включенні дуги (i, j) у гамільтонів контур необхідно виключити з розгляду дугу (i, j) . Після цієї операції потрібно виконати операцію додаткового зведення матриці і покращити нижню межу.

Найбільш імовірно, що в оптимальний контур увійдуть дуги, яким у наведеній матриці відповідають нульові елементи. Тому вибір необхідно здійснювати так. У зведений матриці елемент $a_{ij} = 0$ умовно замінюють на ∞ . Цим самим дуга (i, j) буде виключатися з гамільтонового контура.

Щоб визначити суму констант зведення одержаної матриці, необхідно скласти найменший елемент α_i i -го рядка з мінімальним елементом β_j j -го стовпця, оскільки решта рядків і стовпців містять принаймні по одному нульовому

елементу. Позначимо суму констант зведення матриці з виключеною дугою (i, j) через (2.10):

$$\gamma_{(i,j)} = \alpha_i + \beta_j. \quad (2.10)$$

Аналогічний розрахунок проводиться для всіх інших нульових елементів наведеної матриці, умовно замінюючи їх на ∞ . Насамперед будемо виключати з контура ту дугу (i, j) , для якої сума констант зведення є найбільшою, оскільки в цьому разі відбудеться найбільш різке змінювання оцінки [10, с. 168-171].

2.1.5 Розв’язування задачі комівояжера методом динамічного програмування

Згідно з методом динамічного програмування розв’язування задачі комівояжера починається з останнього етапу. Зафіксуємо M_n – кінцеве місто, куди повинен потрапити комівояжер. При цьому комівояжер може знаходитися в $M_1, M_2, M_3, \dots, M_{n-1}$ містах. Стан системи будемо подавати у формі $M_i \{0\}$, де M_i – місто, в якому знаходиться комівояжер перед прийняттям рішення; $\{0\}$ – означає, що між кінцевим і цим містом відсутні проміжні міста. Кожному стану системи (місцю перебування комівояжера) відповідає певний локальний прибуток, що обчислюється як відстань $D_{M_i M_n}$ від M_i до M_n міста.

Передостанній етап – комівояжеру потрібно потрапити в кінцеве місто, за умови, коли є одне проміжне місто. Тобто комівояжеру з міста M_i необхідно заїхати в проміжне місто, а звідти – в M_n . І так далі для всіх етапів доти, поки кількість проміжних міст не буде дорівнювати загальній кількості міст [10, с. 175].

2.2 Аналіз математичних методів оптимізації транспортних маршрутів

Існує декілька математичних методів оптимізації транспортних маршрутів, включаючи лінійне програмування, цілочисельне програмування, оптимізацію мережевого потоку та метаевристики (наприклад, генетичні алгоритми, моделювання відпалу, оптимізація колонії мурах). Ці методи спрямовані на мінімізацію витрат на транспортування, таких як відстань, час або споживання палива, в той же час задовольняючи обмеження, такі як пропускна здатність, часові вікна та довжина маршруту. Вибір методу залежить від конкретної проблеми, наприклад розміру та структури мережі, типу вартості та обмежень, а також бажаного рівня точності та обчислювальної ефективності. Для реалізації цих методів часто необхідно змодельовати транспортну задачу як задачу математичної оптимізації та розв'язати її за допомогою відповідних алгоритмів.

У лінійному програмуванні (LP) мета полягає в мінімізації лінійної функції змінних рішень, які підлягають лінійним обмеженням. LP можна використовувати для моделювання проблем маршрутизації з безперервними змінними рішень, такими як безперервне призначення транспортних засобів на маршрути. Однак LP може бути непридатним для проблем із дискретними змінними рішення, такими як вибір конкретних маршрутів, оскільки LP вимагає, щоб усі змінні рішення були безперервними.

Цілочисельне програмування (IP) є більш загальною технікою оптимізації, яка дозволяє приймати дискретні змінні рішення. IP можна використовувати для вирішення проблем маршрутизації за допомогою окремих рішень, таких як вибір маршрутів і розподіл транспортних засобів по маршрутах. IP, як правило, потребує більше обчислень, ніж LP, оскільки вимагає вирішення більш складної проблеми оптимізації.

Оптимізація мережевого потоку (NFO) – це спеціалізована техніка для вирішення проблем маршрутизації в мережах. NFO можна використовувати для

моделювання проблем маршрутизації, в яких транспортні засоби повинні слідувати певним маршрутом через мережу, і в яких потік транспортних засобів залежить від обмежень, таких як пропускна здатність і часові вікна.

Метаевристики – це методи оптимізації, які розроблені для пошуку майже оптимальних рішень складних проблем оптимізації, таких як проблеми маршрутизації, шляхом дослідження простору рішень. Метаевристика часто використовується для вирішення проблем маршрутизації, коли розмір проблеми є занадто великим, щоб точні методи були здійсненими, або коли проблема занадто складна, щоб точні методи могли дати рішення за прийнятний проміжок часу.

Мурашина колонія – це метаевристичний алгоритм для вирішення проблем маршрутизації. Його натхненно поведінкою мурашиних колоній, де мурахи шукають їжу, прокладаючи сліди феромонів. В АСО оптимізаційна задача моделюється як граф, де кожен вузол представляє розташування, а кожне ребро – шлях між двома розташуваннями. Процес оптимізації в АСО включає популяцію віртуальних мурах, кожна з яких генерує рішення проблеми, будуючи маршрут через граф на основі локальної інформації, отриманої зі слідів феромонів.

Варто також зазначити, що деякі проблеми з маршрутизацією можуть мати додаткові обмеження, такі як необхідність відвідування певних місць у певному порядку, вимога для транспортних засобів повертатися до центрального депо після кожного маршруту або необхідність мінімізувати кількість використовуваних транспортних засобів. Ці додаткові обмеження можна змоделювати та включити в задачу оптимізації, але вони також можуть зробити проблему більш складною та важкою для вирішення.

Іншим важливим фактором є дані, які використовуються для моделювання транспортної проблеми. Точні дані про відстані, час у дорозі та пропускну здатність є важливими для отримання хорошого рішення. У багатьох випадках ці дані можна отримати з географічних інформаційних систем (ГІС) або інших

джерел, але також може знадобитися оцінка певних параметрів, таких як час у дорозі, на основі історичних даних або іншої інформації.

Також, важливо оцінити ефективність методу оптимізації, який використовується для вирішення проблеми маршрутизації. Це можна зробити шляхом порівняння отриманого рішення з відомими рішеннями або за допомогою показників продуктивності, таких як вартість транспортування або час проходження набору маршрутів. Результати цієї оцінки потім можна використати для вдосконалення методу оптимізації або для перегляду даних, використаних для моделювання проблеми.

Підсумовуючи, оптимізація транспортних маршрутів є складною проблемою, яка вимагає використання математичних методів і ретельного розгляду конкретної проблеми та даних, які використовуються для її моделювання. Вибір методу та продуктивність отриманого рішення можна покращити шляхом ретельної оцінки та вдосконалення процесу оптимізації. Для конкретної проблеми необхідно підібрати кращий метод оптимізації транспортних маршрутів, який залежить від типу змінних рішення та бажаного рівня точності та ефективності обчислень. Кожен метод має свої переваги та обмеження, і може знадобитися використовувати комбінацію методів, щоб знайти найкраще рішення проблеми маршрутизації.

2.3 Особливості застосування базового мурашиного алгоритму для вирішення задачі пошуку оптимального шляху

В даному розділі будуть розглянуті принципи роботи базового мурашиного алгоритму пошуку шляхів для дилеми комівояжера. Маршрут, або шлях є послідовністю ребер в неорієнтованому графі, в якому кінець кожного ребра збігається з початком наступного ребра. Число ребер маршруту називається його довжиною [11, с. 63].

Для вирішення алгоритмічних задач використовуються різні стратегії та інструменти, які цілодобово розробляються для пошуку розв'язку у високопродуктивних обчисленнях. Насправді, коли алгоритми надихаються законами природи, спостерігаються цікаві результати. До такого класу алгоритмів належать еволюційні алгоритми. Ці алгоритми розроблені таким чином, щоб імітувати певну поведінку, а також еволюційні риси геному людини. Крім того, такий алгоритмічний дизайн не тільки обмежений людьми, але також може бути натхненний природною поведінкою певних тварин. Основна мета створення таких методологій полягає в тому, щоб забезпечити реалістичні, релевантні та водночас деякі недорогі рішення проблем, які досі не вирішувались звичайними засобами.

Таким чином, на основі таких еволюційних алгоритмів розвинулися різні методи оптимізації, що відкрило область метаевристик. Метаевристика походить від двох грецьких слів, а саме “meta”, що означає один рівень вище, і “heuriskein”, що означає знаходити. Метаевристика – це алгоритмічна структура високого рівня, яка не залежить від проблеми, яка надає набір вказівок або стратегій для розробки евристичних алгоритмів оптимізації [12]. Такі алгоритми, як “Метод рою часток” (Particle Swarm Optimization) (PSO) і “Мурашиний алгоритм”, є прикладами ройового інтелекту та метаевристики. Метою ройового інтелекту є розробка інтелектуальних багатоагентних систем, черпаючи натхнення з колективної поведінки соціальних комах, таких як мурахи, терміти, бджоли, оси та інших тваринних суспільств, таких як зграї птахів або зграї риб.

Техніка оптимізації мурашиних колоній повністю натхненна поведінкою колоній мурашок 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. Еусоціальні істоти проживають в кооперативній групі, в якій зазвичай одна самка і кілька самців є репродуктивно активними, а особини, які не розмножуються, піклуються про молодняк або захищають і забезпечують групу [13]. 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.

Мурахи живуть у громадських 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 випробувань 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 є вирішальним фактором, довжину кожного шляху можна легко порахувати.

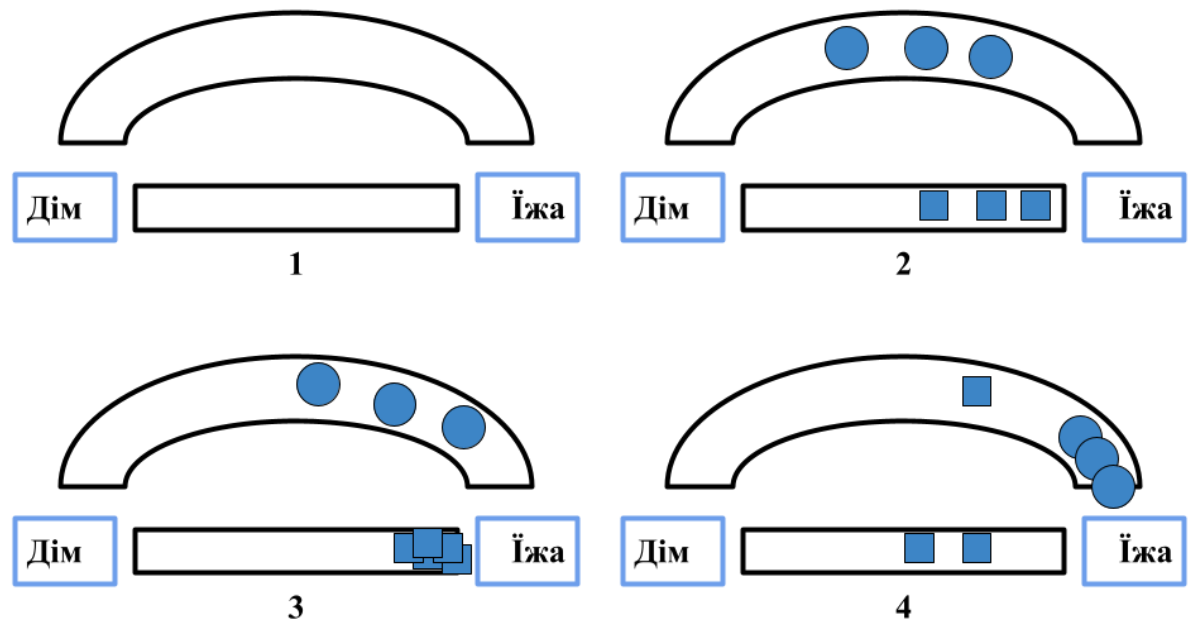


Рисунок 2.1 – Схематичне зображення роботи мурашиного алгоритму

На наведеному вище рис. 2.1 для простоти розглянуто лише два можливі шляхи між джерелом їжі та гніздом мурах. Етапи можна розподілити таким чином:

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

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.

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.

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.

Що стосується описаної вище поведінки мурах, тепер можна розробити алгоритмічний дизайн. Для простоти, єдине джерело їжі та одна колонія мурашок були розглянуті лише з двома можливими шляхами проходження. Весь сценарій можна реалізувати за допомогою зважених графів, де колонія мурашок і джерело їжі діють як вершини (або вузли); шляхи служать ребрами, а значення феромонів є вагами, пов'язаними з ребрами.

Нехай граф $G = (V, E)$, де V, E – ребра та вершини графа. Вершини відповідно до нашого розгляду є V_s (вершина джерела – колонія мурашок) і V_d (вершина призначення – джерело їжі), два ребра є E_1 і E_2 з довжинами L_1 і L_2 , призначеними кожному. Тепер пов'язані значення феромонів (що вказують на їх силу) можна вважати R_1 і R_2 для вершин E_1 і E_2 відповідно. Таким чином, для кожної мурахи початкова ймовірність вибору шляху (між E_1 і E_2) може бути виражена таким чином утворюється рівняння (2.11):

$$P_i = \frac{R_i}{R_1 + R_2}; \quad i = 1, 2. \quad (2.11)$$

Очевидно, що якщо $R_1 > R_2$, то ймовірність вибору E_1 вища і навпаки. Тепер під час повернення цим найкоротшим шляхом, скажімо E_1 , значення феромону оновлюється для відповідного шляху. Оновлення виконується на основі довжини шляхів, а також швидкості випаровування феромону. Таким чином, оновлення можна поетапно реалізувати наступною формулою (2.12):

1. Відповідно до довжини шляху:

$$R_i \leftarrow R_i + \frac{K}{L_i}. \quad (2.12)$$

У наведеному вище оновленні $i = 1, 2$ і K служить параметром моделі. Крім того, оновлення залежить від довжини шляху. Чим коротший шлях, тим більше феромонів.

2. Відповідно до швидкості випаровування феромону за формулою (2.13):

$$R_i \leftarrow (1 - v) * R_i. \quad (2.13)$$

Параметр v належить до інтервалу $(0, 1]$, який регулює випаровування феромону. Далі $i = 1, 2$.

На кожній ітерації всі мурахи розміщуються у вихідній вершині V_s (колонії мурах). Згодом мурахи переходять від V_s до V_d (джерело їжі) після кроку 1. Далі всі мурахи здійснюють свою зворотну подорож і зміцнюють свій обраний шлях на основі кроку 2.

Базовою математичним алгоритмом вважається коли у його основі лежить поведінка мурашиної колонії – маркування вдалих доріг великою кількістю феромону. Робота починається з розміщення мурашок у вершинах графу (містах), потім починається рух мурашок – напрям визначається імовірнісним методом, на підставі формули (2.14):

$$P_i = \frac{l_i^q * f_i^p}{\sum_{k=0}^N l_k^q * f_k^p}; \quad i = 1, 2, \quad (2.14)$$

де P_i – ймовірність вибору шляху i ;

l_i – величина, обернена до довжини (ваги) i – ого переходу;

f_i – кількість феромонів на i – ому переход;

f_i – величина, яка визначає «жадібність» алгоритму;

f_i – величина, яка визначає «стадність» алгоритму i ;

$$q + p = 1.$$

Результат не є точним і навіть може бути одним з гірших, проте, в силу ймовірності рішення, повторення алгоритму може видавати (досить) точний результат [14].

Слід зазначити, що використовуваний мурахами феромон через деякий час повільно випаровується. Дійсно, довжина шляхів, які не були обрані протягом деякого часу, досить велика, і ці шляхи майже не будуть містити слідів феромонів через їх випаровування через певний час, і, як наслідок, ймовірність вибору мурахами коротких шляхів зростає.

Більш детальна схема мурашиного алгоритму представлена на рис. 2.2.

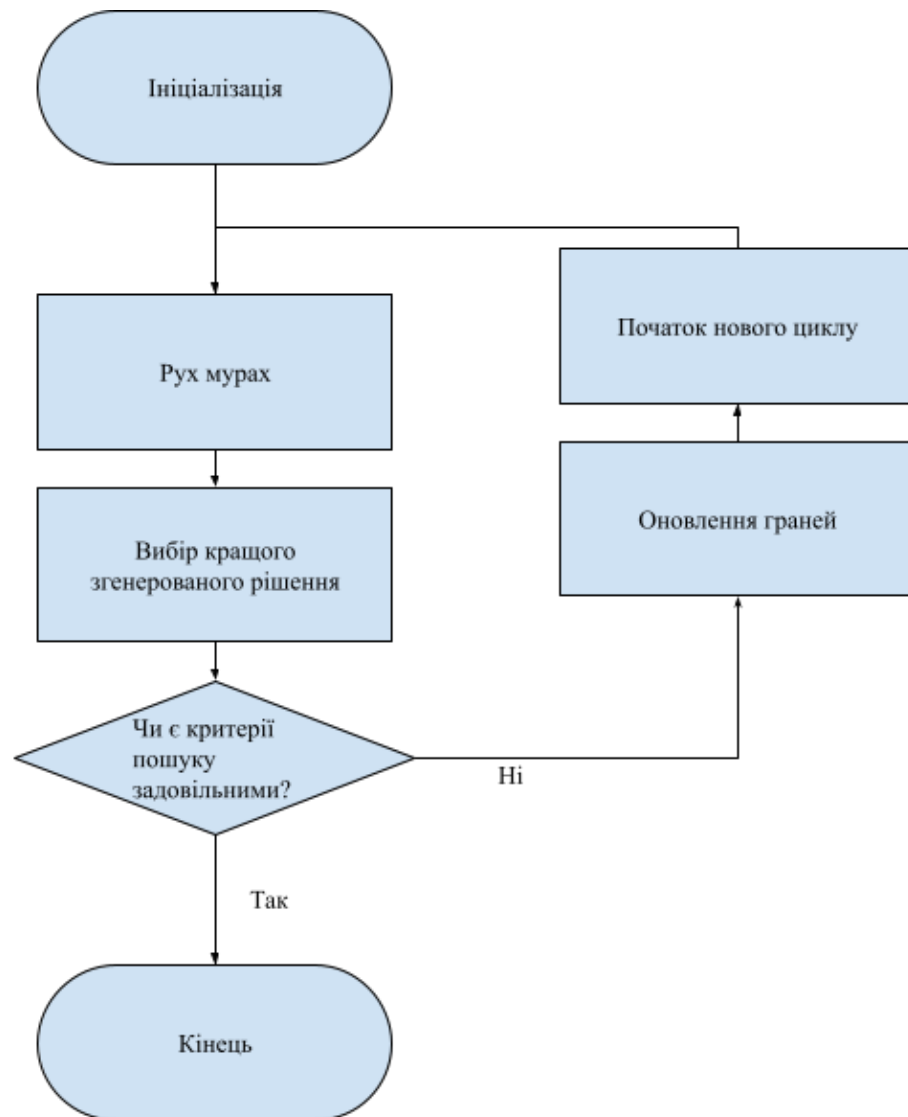


Рисунок 2.2 – Схема роботи мурашиного алгоритму

Однією з перших задач, для вирішення якої був застосований метод мурашиних колоній (алгоритм мурахи), були задачі комівояжера. Основною причиною вибору цього завдання є те, що в цьому типі завдань необхідно знайти найкоротший шлях між точками, тому аналогія методу мурашиних колоній легко адаптується для вирішення цієї проблеми. Для вирішення цієї проблеми було розроблено кілька різних методів, які на основі оптимізації з використанням мурашиних колоній.

Першим методом був метод мурашиних систем (Ant System) (AS). У майбутньому такі метод послужив основою для багатьох інших методів, що працюють за принципом мурашиної колонії.

У методі мурашиних систем агент формує своє рішення в процесі руху від одного вузла до іншого на графі рішень. Метод працює до виконання t_{max} ітерації. На кожній ітерації агенти формують свої рішення за n кроків, на кожному з яких використовується правило вибору наступного вузла - правило вибору агентом, який розташований на якомусь вузлі наступного вузла для переміщення.

Три методи мурашиних систем були спочатку запропоновані Марком Доріго, іншим способом оновлення шляхів - країв. Ці методи мали наступні назви: щільнісний (ant - density), кількісний (ant - quantity) і циклічний (ant - cycle) методи мурашиних систем. У щільності та кількісному методі агенти залишали феромони в процесі прийняття рішень, тоді як у циклічному методі агенти залишив феромони після закінчення руху, тобто після вибору рішення.

Експерименти, проведені для вирішення тестових завдань, показали, що циклічний метод дав значно кращі результати порівняно з двома іншими. У зв'язку з цим були відкинуті два менш ефективні методи. Тому в подальшому метод мурашиних колоній відноситься до циклічного методу мурашиних систем.

2.4 Модифікації мурашиного алгоритму

Поведінка мурах може мати різний математичний опис, тому були створені розширення мурашиного алгоритму (також називається методом мурашиних систем). Розрізняють такі методи: метод на основі ранжування (ASrank); метод мурашиних систем, який використовує елітну стратегію; метод системи мурашиної колонії; максимальний метод мурашиних систем (MAX - MIN AS - MMAS).

Найбільш раннім розширенням методу мурашиних систем була елітна стратегія. Цей підхід заснований на більш високій концентрації феромонів, що

дозволяє знайти найкращий глобальний шлях у деякий момент часу t . Для таких специфічних процесів складання феромон для дуг, які відповідають поточному часовому шляху, проводиться знову. Концентрація феромону за таких умов розраховується відповідно до тривалості оптимального шляху.

Пізніше з'явився метод мурашиних систем, в основі якого лежить ранжування (ASrank). Цей метод є вдосконаленням елітної стратегії і виглядає наступним чином: агенти впорядковані відповідно до довжини утворених ними шляхів, в результаті чого на глобально кращий шлях кількість феромонів збільшиться на деяку вагу w , також збільшення феромонів виробляється лише для дуг, які були обрані найкращими агентами ($w - 1$). При цьому найкращий агент додасть феромон з вагою (wk) згідно (2.15):

$$\phi_{ij}(t + 1) = c * \phi_{ij}(t) + w * \Delta\phi_{ij,gl}(t) + \sum_{k=1}^{w-1} (k - w) * \Delta\phi_{ij,k}(t), \quad (2.15)$$

де $\Delta\phi_{ij,gl}(t) = \frac{1}{L_g}(t)$, L_g – довжина найкращого глобального шляху.

Метод Ant Colony System дає змогу вдосконалити алгоритм мурах, використовуючи інформацію, отриману мураками, які вже завершили обхід для дослідження простору пошуку. Це досягається двома механізмами. По-перше, при оновленні феромонів на краях використовується сувора елітна стратегія. По-друге, агенти обирають наступне місце для переїзду, користуючись так званім псевдовипадковим пропорційним правилом: з імовірністю q_0 агент переміщується в точку u , для якої добуток кількості феромонів i евристичної інформації є максимальним (2.16) :

$$u = \underset{u \in j^i}{\operatorname{argmax}} \{ \phi_{ij} * \varepsilon_{ij}(t)^B \} ds. \quad (2.16)$$

Тоді як з імовірністю $1 - q_0$ базовий підхід буде застосовано для визначення наступної точки для переходу. Значення q_0 є константою. При цьому, якщо q_0 прагне до 1, то використовується тільки псевдовипадкове пропорційне правило, коли $q_0 = 0$, то метод мурашиних колоній працює за принципом методу мурашиних систем. У процесі оновлення шляхів, як було сказано раніше, використовується сувора елітна стратегія, згідно з якою єдиний агент, який зробив оптимальне рішення, виробляє феромон на шляху свого руху. Кількість феромонів на обличчях змінюється за формулою (2.17):

$$\phi_{ij}(t + 1) = c * \phi_{ij}(t) + (1 - c) * \Delta\phi_{ij, best}(t). \quad (2.17)$$

Найкращим агентом може бути агент, який отримав найкраще рішення під час виконання поточної ітерації, або глобально найкращий агент, який отримав найкраще рішення у всіх ітераціях від початку методу.

Ще одна перевага методу мурашиних колоній полягає в тому, що агенти оновлюють кількість феромонів у процесі прийняття рішення (подібно до методів щільності та кількості мурашиних систем). Такий підхід призводить до зниження ймовірності вибору однакових шляхів усіма агентами. Завдяки цьому зменшується ймовірність зациклення в локальному оптимумі.

Максимальний метод мурашиних систем (MAX - MIN AS - MMAS) дозволяє ввести нижню і верхню межі для можливих значень феромонів на межі, також цей метод відрізняється підходом до опису їх значення при виконанні початкової ініціалізації. Для MMAS використовується список значень феромонів, обмежений ϕ_{min} і ϕ_{max} , $\forall \phi_{ij}$, тобто $\phi_{min} \leq \phi_{ij} \leq \phi_{max}$.

У загальному вигляді відмінності між різновидами методу мурашиних колоній можна відобразити в табл. 2.1.

Таблиця 2.1 – Відмінності між різновидами методів мурашиних колоній

Критерій	AS	ASrank	ACS	MMAS
Завдання, які вирішуються	Задача комівояжера, транспортна задача, квадратична задача призначення	Задача комівояжера	Задача комівояжера, календарне планування	Задача комівояжера, квадратична задача призначення
Як сильно впливає кількість мурах на пошук результату	Сильно	Середньо	Слабко	Слабко
Коли додається феромон	Після пошуку рішення	Після пошуку рішення	Під час пошуку рішення	Під час пошуку рішення
Правило обрання наступного пункту	Базове правило	Псевдовипадкове пропорційне правило	Базове правило	Базове правило

Висновки до розділу 2

У даному розділі досліджено задачі пошуку оптимального шляху при плануванні транспортних маршрутів TSP, її математичне формулювання і способи розв'язання в частості методом гілок і меж, проаналізовано математичні методи оптимізації транспортних маршрутів, досліджено мурашиний алгоритм та його базові модифікації.

Мурашиний алгоритм являє собою збільшення напрямних, які отримали на залежному від мурашиних феромонів шляху. Це досягається шляхом заміни дискретного розподілу ймовірностей неперервною функцією, яка називається функцією щільності ймовірності, також відомою як розподіл Гауса. Таким чином ACSO можна використовувати для вирішення не лише TSP, а й інших графових задач.

3 РОЗРОБКА СИСТЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ МАРШРУТІВ НА ОСНОВІ АЛГОРИТМУ ПОКРАЩЕНОЇ МУРАШИНОЇ КОЛОНІЇ

3.1 Особливості алгоритму графа згорткової мережі покращеної мурашиної колонії

Метою створення алгоритму на основі згорткового мережевого графіка мурашиної колонії стало моделювання та вивчення поведінки та процесу прийняття рішень мурахами в колонії. Згортковий мережевий графік – це тип нейронної мережі, яка зазвичай використовується в обробці зображень і відео, і характеризується використанням згорткових шарів, які допомагають ідентифікувати закономірності в даних.

Створивши алгоритм, заснований на графі згорткової мережі колонії мурах, дослідники могли б краще зрозуміти, як мурахи в колонії взаємодіють один з одним і колективно приймають рішення. Алгоритм може допомогти ідентифікувати та зрозуміти моделі поведінки, які відбуваються в колонії, такі як спілкування між мурахами, розподіл праці та процеси прийняття рішень.

Крім того, алгоритм може допомогти дослідникам змоделювати поведінку колонії мурашок у різних сценаріях, що може бути корисним для розуміння того, як колонія може реагувати на зміни в навколишньому середовищі, такі як наявність їжі або присутність хижаків.

Загалом, створення алгоритму, заснованого на графі згорткової мережі колонії мурашок, надає цінну інформацію про поведінку та процеси прийняття рішень мурахами в колонії, а також допомагає інформувати майбутні дослідження в галузі колективного інтелекту та інтелекту роїв.

3.1.1 Новітні дослідження вирішені у задачі комівояжера

Такого роду задача оптимізації, яка знаходить оптимальне рішення в скінченній множині можливих рішень, називається задачею комбінаторної оптимізації. З широким застосуванням у різних галузях промисловості та бурхливим розвитком обчислювальної техніки задача комбінаторної оптимізації виросла в самостійну галузь операційних досліджень. Питання дослідження стосуються фінансових інвестицій, екологічного середовища, медичної біології, управління логістикою, транспорту, промислового будівництва, медичного іміджу та багато інших областей. Задача про комівояжера є однією з найпопулярніших задач комбінаторної оптимізації останніх років і широко використовується як еталон для різних методів оптимізації та метаевристичних пошуків. Задача комівояжера була запропонована в 1930 році, її мета полягає в тому, щоб знайти найкоротший шлях, щоб відвідати кожне місто рівно один раз і повернутися до міста початкової точки на основі заданого списку міст і відстані між містами. Оскільки можливим розв'язком цієї проблеми є повна перестановка всіх вершин, із збільшенням кількості вершин відбуватиметься комбінаторний вибух, тому це також NP-складна задача. Точні алгоритми, такі як січна площина, розгалуження та межа, найкоротше остовне дерево, можуть бути використані для вирішення точного рішення TSP. Однак, коли проблема комівояжера є надто великою, точність вибірки даних недостатня, мета завдання конфліктує тощо, точний алгоритм працює довго, і отримати задовільні результати важко. Тому вчені використовують метаевристичні та евристичні алгоритми для ефективного вирішення таких проблем. У цей час алгоритм не наполягає на точному оптимальному розв'язку, а знаходить прийнятне оптимальне рішення за розумний час.

У мета-евристичному алгоритмі принцип і механізм АСО роблять його природним алгоритмом вирішення задач комівояжера. Коли мурахи шукають їжу,

вони передають інформацію між особинами, виділяючи феромони на своєму шляху. У той же час мурахи будуть вибирати шляхи з більшою концентрацією феромону для пересування. Під час безперервного пошуку колонії мурашок, чим коротший шлях, тим більше мурах пройде, і тим більша концентрація феромонів залишиться на шляху, що спонукає більше мурах вибирати цей шлях для руху, таким чином створюючи цикл для формування механізму позитивного зворотного зв'язку. Це дозволяє колонії мурашок швидко знаходити найкоротший шлях до їжі. У 1991 році італійський вчений Доріго з іншими просто імітували таку поведінку мурашиних колоній у біологічному світі в пошуках їжі та запропонували інтелектуальний алгоритм біонічної оптимізації рою: мурашиної колонії. Але й сама АСО має певні дефекти та недоліки. Завдяки постійному поглибленому розвитку досліджень вчені запропонували багато вдосконалень і схем злиття алгоритмів. Було досягнуто хороших результатів і вирішено багато практичних проблем у різних ситуаціях. З розвитком машинного навчання, особливо з розвитком графових нейронних мереж, підхід, заснований на навчанні, також ефективний у розв'язанні TSP. Але АСО має дефекти та недоліки, позбувшись яких вдосконаленням алгоритму можна досягти кращих результатів. У той же час графова згортка також має можливість розв'язувати TSP. Поєднанням АСО з графовою згорткою стала нова вдосконалена оптимізація мурашиної колонії: Graph Convolutional Network Improved Ant Colony Optimization (GCNIACO). З метою усунення недоліків, пов'язаних із відсутністю феромону та повільною конвергенцією на початковій стадії АСО, для створення кращого рішення введено мережу згортки графів, і це краще рішення перетворюється на початковий феромон колонії мурашок за допомогою стратегії перетворення феромонів. Таким чином, початкова швидкість вирішення алгоритму була покращена. Потім, беручи до уваги недоліки оптимізації мурашиної колонії, а саме стагнацію та впадання в локальний оптимум на пізнішій ітераційній стадії, було покращено здатність

алгоритму вистрибувати з локального оптимуму шляхом динамічного регулювання фактор мінливості феромонів і введення алгоритму 3-opt.

Щоб отримати задовільний розв'язок NP-проблеми комівояжера, вчені запропонували багато метаевристик для вирішення TSP. Гюндуз та інші запропонували новий метод дискретної оптимізації під назвою DJAYA для задач оптимізації, кодованих перестановкою, і дослідили продуктивність запропонованого алгоритму на 14 різних симетричних наборах даних TSP. Експериментальні результати показали, що алгоритм є опціональним і конкурентним для пооптимізації. Панвар та інші на основі метаевристичного алгоритму для розв'язання задач безперервної оптимізації: оптимізатор сірого вовка (GWO) у поєднанні з 2-оптимальним алгоритмом створили новий дискретний алгоритм GWO для розв'язання симетричної задачі комівояжера. А ефективність запропонованого методу перевірено шляхом експериментального моделювання. Канна та інші запропонували новий гібридний алгоритм шляхом інтеграції двох мета-евристичних алгоритмів із хорошою продуктивністю: алгоритм оптимізації полювання на оленів (DHOA) на основі дощових черв'яків. Запропонований алгоритм оптимізації показав кращу продуктивність конвергенції та меншу обчислювальну складність при розв'язанні TSP. На основі алгоритму оптимізації наїзника (ROA) і алгоритму оптимізації плямистої гієни (SHO), Крішна та інші побудували новий гібридний алгоритм: оптимізацію вершника на основі плямистої гієни (S-ROA). Розв'язавши задачу комівояжера, було підтверджено конкурентоспроможність побудованого методу. Саджі та інші запропонували новий дискретний bat-алгоритм для вирішення TSP. Польоти Леві та оператор нейтрального кросовера були введені, щоб підвищити здатність алгоритму виходити за локальний оптимум. Таким чином, загальна продуктивність алгоритму була покращена. Існують також метаевристичні алгоритми, такі як дискретні алгоритми, натхненні вороною (DC), дискретний алгоритм стрибка жаби (DSFLA), диференціальна еволюція (DE), дискретна

оптимізація мавпи-павука (DSMO) тощо також були запропоновані та застосовані для вирішення TSP, і всі вони отримали хороші результати. Завдяки поглибленим дослідженням і розробкам машинного навчання та графових нейронних мереж розв'язання TSP на основі навчання також привернуло увагу вчених. Деудон та інші запропонували структуру нейронної комбінаторної оптимізації для вирішення TSP. У структурі координати міста використовувалися як вхідні дані, а нейронна мережа була навчена за допомогою навчання з підкріпленням для прогнозування розподілу розташування міста. Потім результати запропонованого каркасу були перевірені шляхом експериментального моделювання. Пратес та інші успішно застосували графові нейронні мережі для вирішення TSP, довівши, що графові нейронні мережі можуть вивчати змінні рішення для вирішення проблеми комівояжера з невеликим наглядом. Кул та інші запропонували модель, засновану на рівні уваги, і навчили модель за допомогою REINFORCE з простою базовою лінією на основі детермінованого жадібного розгортання. Нарешті, задовільні результати були отримані при розв'язанні таких завдань, як задача орієнтування (OP) і збір призивів TSP (PCTSP). Ху та інші запропонували двонаправлену графову нейронну мережу для вирішення проблеми комівояжера на будь-якому симетричному графі. Мережа використовувала імітаційне навчання, щоб послідовно генерувати наступне місто для відвідування. У той же час, розробивши двонаправлений рівень передачі повідомлень, граф було закодовано на основі крайових і часткових розв'язків, щоб побудувати майже оптимальне розв'язання задачі комівояжера на будь-якому симетричному графі.

Кожне з цих вдосконалень принесло свої результати але все ж залишило місце для експериментів, розвитку і покращень.

3.1.2 Новітні дослідження у покращенні мурашиного алгоритму

Рагмані та інші запропонували новий гібридний алгоритм на основі нечіткої логіки та АСО для покращення вирівнювання навантаження в середовищах хмарних обчислень. Алгоритм використовував експериментальний дизайн Тагучі для отримання оптимального значення параметра оптимізації мурашиної колонії та визначив нечіткий модуль для оцінки значення феромону, таким чином прискоривши обчислення алгоритму. Націлюючись на недоліки повільної швидкості конвергенції оптимізації мурашиної колонії та схильності потрапляти до локального оптимуму, Ебадінежад та інші запропонували динамічну оптимізацію колонії мурашок (DEACO), яка динамічно регулювала параметри АСО. Він був використаний для вирішення прикладу TSP, який підтвердив хорошу продуктивність запропонованого алгоритму з точки зору швидкості конвергенції та точності пошуку. Націлюючись на недоліки низької точності конвергенції та низької швидкості конвергенції в оптимізації мурашиної колонії, Лі та інші запропонували псеводинамічний АСО пошуку з покращеним механізмом негативного зворотного зв'язку. Завдяки введенню правила обчислення кута та кута в правилі передачі феромонів це вплинуло на ймовірність вибору міста та покращило здатність алгоритму виходити за межі локального оптимуму. Потім алгоритм оновив концентрацію феромонів на оптимальному та найгіршому шляху одночасно, а також підвищив вагу концентрації феромонів на оптимальному шляху, тим самим підвищивши швидкість збіжності алгоритму. Туані та інші запропонували гетерогенну адаптивну оптимізацію колонії мурашок (HAACO), яка модифікувала формулу ймовірності переходу в АСО, щоб ввести неоднорідність АСО. Потім шляхом введення набору правил адаптації параметрів була досягнута адаптивність і оптимізація параметрів колонії мурашок. У той же час 3-опційний алгоритм локального пошуку був інтегрований у запропонований алгоритм для подальшого покращення пошукових можливостей алгоритму.

Націлюючись на проблему, що АСО має сильну залежність від феромону і схильний потрапляти в локальний оптимум, Чжен та інші запропонували вдосконалену оптимізацію колонії гібридних генетичних мурах. Оптимальне рішення, створене генетичним алгоритмом, було використано як початкова інформація феромону в оптимізації колонії мурашок для покращення здатності алгоритму глобального пошуку та швидкої конвергенції. Лі та інші запропонували вдосконалену оптимізацію мурашиної колонії для вирішення проблеми планування шляху безпілотних кранів під час підйому, а також покращили продуктивність алгоритму шляхом вдосконалення евристичної функції, механізму оновлення феромонів і стратегії вибору шляху. Націлюючись на недоліки оптимізації мурашиних колоній для вирішення проблеми планування шляху, такі як потрапляння в локальний оптимум і повільну швидкість конвергенції, Танг та інші запропонували удосконалений алгоритм колонії мурашок. Вони використовували стратегію диференційованого розподілу, щоб покращити початкову концентрацію феромону на шляху. Потім АСО було вдосконалено шляхом прийняття стратегії оптимізації локального блоку шляху, введення коефіцієнта посилення саморегулювання феромонів, введення параметрів випадкового переходу між станами тощо, щоб покращити конвергенцію та стабільність алгоритму. Лю та інші запропонували покращену динамічну адаптивну оптимізацію колонії мурах (IDAACO) для вирішення проблеми проектування маршруту труб. IDAACO розробив чотири вдосконалені нові механізми: евристична стратегія з направленою інформацією, адаптивна стратегія псевдовипадкової передачі, покращений локальний механізм оновлення феромонів і вдосконалений механізм глобального оновлення феромонів. Потім за допомогою експериментального моделювання було перевірено чудову продуктивність покращеного алгоритму з точки зору практичності та високої ефективності. Ян та інші запропонували багатofакторну покращену оптимізацію мурашиної колонії для вирішення проблеми планування шляху мобільних роботів.

Продуктивність алгоритму було покращено шляхом побудови багатofакторних евристичних функцій, поетапного призначення початкових феромонів, оновлення класифікації феромонів, прийняття максимальної та мінімальної стратегії мурашок та адаптивного коригування фактора мінливості феромонів. Хі та інші запропонували покращену оптимізацію колонії мурашок (ІАСО) для вирішення проблеми маршрутизації транспортного засобу з м'якими часовими вікнами. Базуючись на базовій оптимізації мурашиної колонії, вони покращили формулу ймовірності переходу, розробили адаптивно налаштований коефіцієнт мінливості феромонів і запровадили локальний пошук змінної околиці, вбудований оператором вставки та оператором обміну. Вони також встановлюють умови для запуску та виходу з локального пошуку та оновлюють поточне локальне оптимальне рішення. Нарешті, ефективність вдосконаленого алгоритму була перевірена шляхом експериментального моделювання. Ванг та інші запропонували удосконалений алгоритм мурашиної колонії для вирішення проблеми періодичного зеленого маршруту автомобіля з часовими вікнами. Розробляючи механізм оновлення феромонів на основі інформаційної ентропії в алгоритмі та вводячи змінний пошук сусідства, алгоритм глобального та покращено можливості локального пошуку.

АСО має високу надійність, а механізм позитивного зворотного зв'язку феромонів дає можливість оптимізації мурашиної колонії знайти оптимальне рішення. Але в той же час алгоритм має й такі недоліки:

- на початковій стадії розв'язку бракує феромону, швидкість розв'язку повільна;
- на пізній стадії розчину під впливом феромону легко потрапити в локальний оптимум.

Тому, з огляду на недоліки АСО, запропоновано граф згорткової мережі покращеної мурашиної колонії. По-перше, графова згортка вводиться для створення кращого рішення, а краще рішення перетворюється на початкове

значення феромону за допомогою стратегії перетворення феромону, щоб покращити початкову швидкість вирішення алгоритму, а здатність алгоритму виходити за межі локального оптимуму потім посилюється шляхом динамічного регулювання коефіцієнта мінливості феромонів і поєднання з алгоритмом 3-opt.

3.1.3 Аналіз графу згорткової мережі

У початковий момент оптимізації мурашиної колонії всі шляхи мають однакову концентрацію феромонів, і привабливість для мурашок також однакова. Як наслідок, феромон на початковій стадії погано керує мурахами вибором шляхів, і швидкість збіжності оптимального рішення є нижчою. Крім того, він може шукати нерелевантні шляхи, які утворюють велику кількість неоптимальних компонентів шляху. Посилення феромону на цих шляхах заважає оновленню загального феромону в алгоритмі, що не сприяє дослідженню кращого шляху. Тому з метою підвищення продуктивності пошуку алгоритму на початковому етапі для генерації кращого рішення вводиться графова згортка, а кращий розчин перетворюється на початкове значення феромону за допомогою стратегії перетворення феромону. Тоді різниця в концентрації феромонів між кращим і гіршим шляхом на початковому етапі збільшується. Згорткова нейронна мережа (CNN) вимагає звичайних доменів даних, таких як 2- або 3-вимірні зображення евклідової сітки в комп'ютерному зорі (CV), 1-вимірний послідовний текст у обробці природної мови тощо. Однак багато даних зазвичай знаходяться не в області звичайних даних, а в області різномірних графіків. Це вимагає використання іншої загальної структури даних, тобто графа, що складається з вершин і ребер. Але також важко безпосередньо визначити такі операції, як згортка та об'єднання в графі, що перешкоджає розвитку CNN, тому ці приводи утворюють мережу згортки графів (GCN). У 2019 році Чайтанья та інші представили новий метод, заснований на навчанні, для розв'язання задачі комівояжера, використавши згорткову мережу глибокого графа для побудови

ефективного представлення графа задачі комівояжера, а потім вивели оптимальний шлях у неавторегресійний спосіб через високо паралелізований пошук променя. Основна практика:

- 1) візьміть граф як вхідні дані, витягніть синтетичні характеристики з вузлів і ребер графа шляхом укладання кількох згорткових шарів графа та навчіть модель згортки графа безпосередньо виводити матрицю суміжності, що відповідає шляху;
- 2) потім використовуйте технологію post-hoc пошуку променя, щоб перетворити матрицю суміжності, отриману з моделі, у дійсний шлях.

Серед них матриця суміжності позначає ймовірність появи ребра на шляху; Параметри моделі згортки графа навчаються наскрізно шляхом мінімізації втрат перехресної ентропії через градієнтний спад. У побудові згорткового шару графа нехай x_i^1 і e_{ij}^l відповідно позначають вектор ознаки вузла та вектор ознаки краю на шарі згортки l графа, пов'язаному з вузлом i та ребром i, j , тоді ознака вузла та ознака краю наступного шару визначаються як рівняння (3.1) і (3.2):

$$x_i^{l+1} = x_i^l + \text{ReLU}(\text{BN}(W_1^l x_i^l + \sum_{j \sim i} \eta_{ij}^l \odot w_2^l x_j^l)) \text{ with } \eta_{ij}^l = \frac{\sigma(e_{ij}^l)}{\sum_{j \sim i} \sigma(e_{ij}^l) + \epsilon} \quad ; \quad (3.1)$$

$$e_i^{l+1} = e_i^l + \text{ReLU}(\text{BN}(W_3^l e_{ij}^l + W_4^l x_i^l + W_5^l x_j^l)), \quad (3.2)$$

де $W \in R^{h \times h}$ – параметр, який необхідний для навчання моделі згортки графа;
 h – прихована розмірність кожного шару згортки графа;
 σ – сигмоподібна функція;

ε – це невелике значення, яке може гарантувати, що знаменник η_{ij}^l не дорівнює 0;

ReLU – випрямлена лінійна одиниця, яка використовується як функція активації в графовій згортковій мережі;

BN – означає нормалізацію партії.

Коли $l = 0$, $x_i^{l=0}$ і $e_{ij}^{l=0}$ відповідно представляють функцію вузла та характеристику краю вхідного шару, а їх визначення є такими, як показано в рівняннях (3.3) та (3.4) відповідно:

$$x_i^{l=0} = A_1 x_i + b_l; \quad (3.3)$$

$$e_{ij}^{l=0} = A_2 d_{ij} + b_2 || A_3 \delta_{ij}^{k-NN}; \quad (3.4)$$

$$\delta_{ij}^{k-NN} = \begin{cases} 1, \text{ якщо вузол } i \text{ та вузол } j \text{ є найближчими сусідами } k \\ 2, \text{ якщо вузол } i \text{ та вузол } j \text{ є самоз'єднаними} \\ 0, \text{ за інших умов} \end{cases}, \quad (3.5)$$

де x_i – двовимірний координатний вектор вхідного графіка, $x_i \in [0, 1]^2$;

$A_1 \in R^{h \times 2}$, $A_2 \in R^{\frac{h}{2} \times 1}$, $A_3 \in R^{\frac{h}{2} \times 3}$, h – прихований розмір шару згортки графа,
 A_1, A_2, A_3, b_1, b_2 – початкові значення параметрів моделі згортки графа;

$||$ – оператор конкатенації.

Рівняння (3.3) вбудовує двовимірну координату як h -вимірний вектор ознаки;

Рівняння (3.4) включає евклідову відстань d_{ij} краю як $\frac{h}{2}$ -вимірний вектор ознак;

δ_{ij}^{k-NN} у рівнянні (3.5) визначається як індикаторна функція ребра в TSP, і процес навчання моделі прискорюється шляхом введення k -графа найближчих сусідів, зазвичай $k = 20$.

Характеристика ребра e_{ij}^l останнього шару моделі згортки графа використовується для обчислення ймовірності p_{ij}^{TSP} тому, що край з'єднаний на шляху TSP. p_{ij}^{TSP} можна розглядати як імовірнісну теплову карту граничних з'єднань, обчислену на матриці суміжності, кожна $p_{ij}^{TSP} \in [0, 1]^2$ і обчислюється за допомогою багат шарового перцептрона (MLP):

$$p_{ij}^{TSP} = MLP(e_{ij}^l). \quad (3.6)$$

Після того, як модель згортки графа видає матрицю суміжності, що передбачає ймовірність появи країв, отримується імовірнісна теплова карта крайових з'єднань. Використовуючи техніку пошуку променя, починаючи з першого вузла, імовірнісна теплова карта досліджується шляхом розширення b найбільш вірогідних краєвих з'єднань між сусідніми вузлами. Перші b локальні шляхи кожного етапу потім ітеративно розширюються, доки не будуть відвідані всі вузли на графі. У той же час під час процесу пошуку вузли, відвідані раніше, екрануються, щоб забезпечити дійсність шляху. Остаточним прогнозованим оптимальним шляхом є шлях із найменшою довжиною серед b повних шляхів після закінчення пошуку променя. Під час удосконалення початкового феромону АСО, якщо оптимальне рішення, передбачене та згенероване графовою

згортковою мережею, безпосередньо перетворюється на початковий феромон, інформація графового згорткового мережевого рішення може бути використана не повністю, оскільки кількість вибірок надто мала. Тому, щоб повноцінно та ефективно використовувати інформацію графових згорткових мережевих рішень, розроблено наступну стратегію перетворення феромонів. Основна практика така:

– встановіть початковий феромон оптимізації колонії мурашок як $\tau = \tau_C + \tau_{GCN}$;

– серед них τ_C є початковою константою феромону, і значення можна спеціально встановити на основі фактичної ситуації, необхідна велика різниця в концентрації феромону між кращим і гіршим шляхом після використання, і ця стаття встановлює його як $\tau_C = 0.1$.

τ_{GCN} генерується перетворенням оптимального розв'язку графа згорткової мережі, який можна розрахувати за такими формулами (3.7):

$$\tau_{ij}^{GCN} = \frac{k_{ij}}{q}; \quad (3.7)$$

$$\tau_{GCN} = \begin{vmatrix} \tau_{11}^{GCN} & \dots & \tau_{1n}^{GCN} \\ \tau_{n1}^{GCN} & \dots & \tau_{nn}^{GCN} \end{vmatrix}, \quad (3.8)$$

де τ_{ij}^{GCN} – концентрація феромону на шляху між вузлами i та j ;

q – перші q коротких шляхів, взятих із пошуку променя графової згорткової мережі, відповідно до літератури, значення q становить $q = 30$;

k_{ij} – кількість разів, коли кожні два вузли міста (i, j) з'являються на q шляхах;

n – кількість міських вузлів.

Щоб зменшити вплив феромонного посилення на погані шляхи, у цьому дослідженні встановлено поріг для k_{ij} . Коли кількість входжень двох вузлів міста (i, j) у q шляхів менше або дорівнює $10\%q$, $k_{ij} = 0$, інакше значення k_{ij} залишається незмінним. Конкретна формула показана в рівнянні (3.9):

$$\begin{aligned} k_{ij} &= k_{ij}, \text{ якщо } k_{ij} > 10\%q; \\ k_{ij} &= 0, \text{ якщо } k_{ij} \leq 10\%q. \end{aligned} \quad (3.9)$$

3.1.4 Аналіз динамічного коефіцієнту летючості феромонів ρ

Коефіцієнт мінливості феромону представляє ступінь мінливості феромону в кожній ітерації, що впливає на здатність глобального пошуку та швидкість конвергенції оптимізації колонії мурах. Коли значення занадто велике, феромон на кожному шляху випаровується швидше, і колонія мурах може шукати шлях неодноразово, що призводить до зниження швидкості конвергенції. Коли значення занадто мале, швидкість мінливості феромону на кожному шляху є повільною, а накопичення концентрації феромону на шляху є занадто високим, що може вплинути на випадковість і глобальну пошукову здатність пошуку колонії мурах, в результаті чого алгоритм потрапляє в локальний оптимум. Тому у цьому дослідженні було використано вдосконалений метод для динамічного коефіцієнта летючості феромонів, як показано в рівнянні (3.10):

$$\begin{aligned} \rho &= \log\left(\frac{2 \times NC_{max}}{NC_{max} + NC}\right), \text{ якщо } \log\left(\frac{2 \times NC_{max}}{NC_{max} + NC}\right) > \rho_{min}; \\ \rho &= \rho_{min}, \text{ за інших умов,} \end{aligned} \quad (3.10)$$

де NC_{max} – загальна кількість ітерацій алгоритму;

NC – номер поточної ітерації алгоритму;

ρ_{min} – мінімальне значення ρ , яке встановлюється, щоб запобігти тому, щоб воно було занадто малим і зменшувало швидкість збіжності алгоритму.

Згідно з літературою [15], найкращий діапазон емпіричних результатів ρ становить $0.1 \leq \rho \leq 0.99$, у дослідженні встановлюється значення ρ_{min} до $\rho_{min} = 0.1$.

Удосконалена крива зміни коефіцієнта летючості феромонів показана на рис. 3.1.

Основна ідея динамічного коефіцієнта летючості феромонів, що було досліджена в роботі, полягає в тому, існує ймовірність, що значення коефіцієнта летючості феромонів змінюватиметься з кількістю ітерацій. На початковому етапі ітерації алгоритму значення коефіцієнта мінливості феромонів встановлюється більшим, накопичення концентрації феромонів на шляху є меншим, і мурахи можуть шукати більше шляхів, так що алгоритм має сильну здатність глобального пошуку. Після того, як алгоритм повторює певну кількість разів, числове налаштування коефіцієнта летючості феромону поступово зменшується, так що накопичення концентрації феромону на шляху поступово збільшується, що посилює тягнучу та направляючу дію феромону на колонію мурашок. Мурахи поступово сходяться до шляху з високою концентрацією феромонів, що прискорює швидкість збіжності алгоритму.

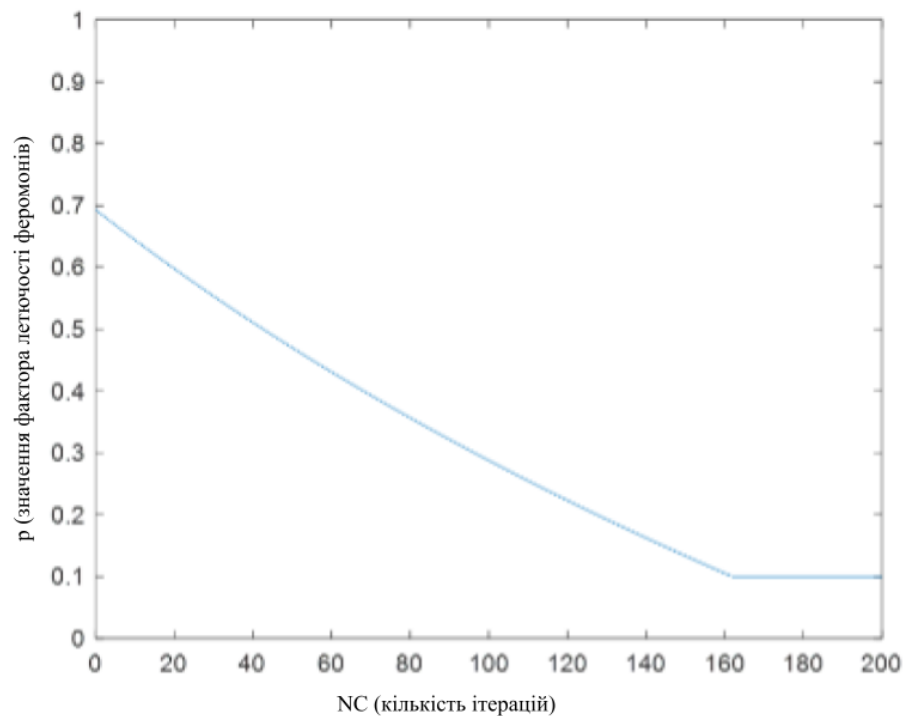


Рисунок 3.1 – Динамічна крива зміни фактора летючості феромонів

3.1.5 Особливості алгоритму 3-opt

Коли оптимізація мурашиної колонії шукає можливе рішення шляху, вона може генерувати неоптимальний шлях із явищем перехрещення, подібним до рис. 3.2(a). Чим більше шляхів перетинаються, тим більше довжина шляху і тим гірша якість отриманого можливого рішення. Таким чином, у цьому дослідженні використовується алгоритм 3-opt [16] для оптимізації мурашиної колонії для подальшого скорочення довжини шляху. Тобто операція деперетину виконується на частині шляху, яка має подібне явище перетину на малюнку 2(a), і вона оптимізується в оптимальний шлях на малюнку 2(b). Конкретні кроки операції:

– починаючи з першого вузла $t = 1, 5$ точок $c(t), c(t + 1), c(t + 2), c(t + 3), c(t + 4)$ є щоразу постійно вибирається;

– по черзі поміняйте місцями три середні точки, щоб отримати шість видів послідовного розміщення, і обчислені довжини такі

$$L_1 = d[c(t), c(t + 1)] + d[c(t + 1), c(t + 2)] + d[c(t + 2), c(t + 3)] + d[c(t + 3), c(t + 4)];$$

$$L_2 = d[c(t), c(t + 1)] + d[c(t + 1), c(t + 3)] + d[c(t + 3), c(t + 2)] + d[c(t + 2), c(t + 4)];$$

$$L_3 = d[c(t), c(t + 2)] + d[c(t + 2), c(t + 1)] + d[c(t + 1), c(t + 3)] + d[c(t + 3), c(t + 4)];$$

$$L_4 = d[c(t), c(t + 2)] + d[c(t + 2), c(t + 3)] + d[c(t + 3), c(t + 1)] + d[c(t + 1), c(t + 4)];$$

$$L_5 = d[c(t), c(t + 3)] + d[c(t + 3), c(t + 2)] + d[c(t + 2), c(t + 1)] + d[c(t + 1), c(t + 4)];$$

$$L_6 = d[c(t), c(t + 3)] + d[c(t + 3), c(t + 1)] + d[c(t + 1), c(t + 2)] + d[c(t + 2), c(t + 4)].$$

– потім порівнюють розмір довжини $L_1, L_2, L_3, L_4, L_5, L_6$ і замінити початкове розташування від $c(t)$ до $c(t + 4)$ на шляху розташуванням, що відповідає мінімальній довжині;

– повторювати $t = t + 1$ по черзі, поки $t = n$.

У TSP важливо відзначити, що:
 $c(n + 1) = c(1), c(n + 2) = c(2), c(n + 3) = c(3), c(n + 4) = c(4)$.

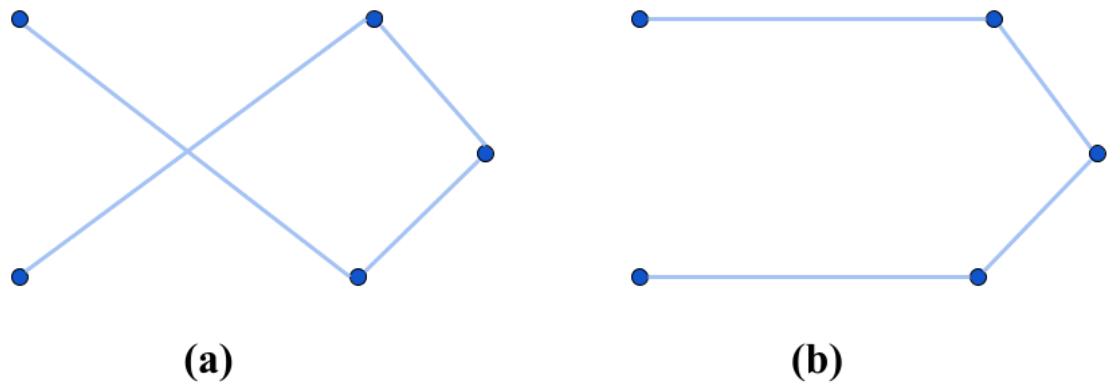


Рисунок 3.2 – Приклад декросингу алгоритму з 3 варіантами

3.1.6 Аналіз потоку GCNIACO

Основна ідея покращеної оптимізації мурашиної колонії графової згорткової мережі:

- кращий розчин, створений графовою згортковою мережею GCN, перетворюється на початковий феромон ACO за допомогою стратегії перетворення феромонів, а різниця початкової концентрації феромону на найкращому і на побічному шляхах покращується;
- тоді коефіцієнт мінливості феромонів покращується, щоб бути динамічно адаптивним, і алгоритм 3-opt використовується для операції де-перетину, що додатково оптимізує довжину шляху та покращує здатність алгоритму виходити за межі локального оптимуму.

Блок-схема покращеної оптимізації мурашиної колонії графової згорткової мережі показана на рис. 3.3.

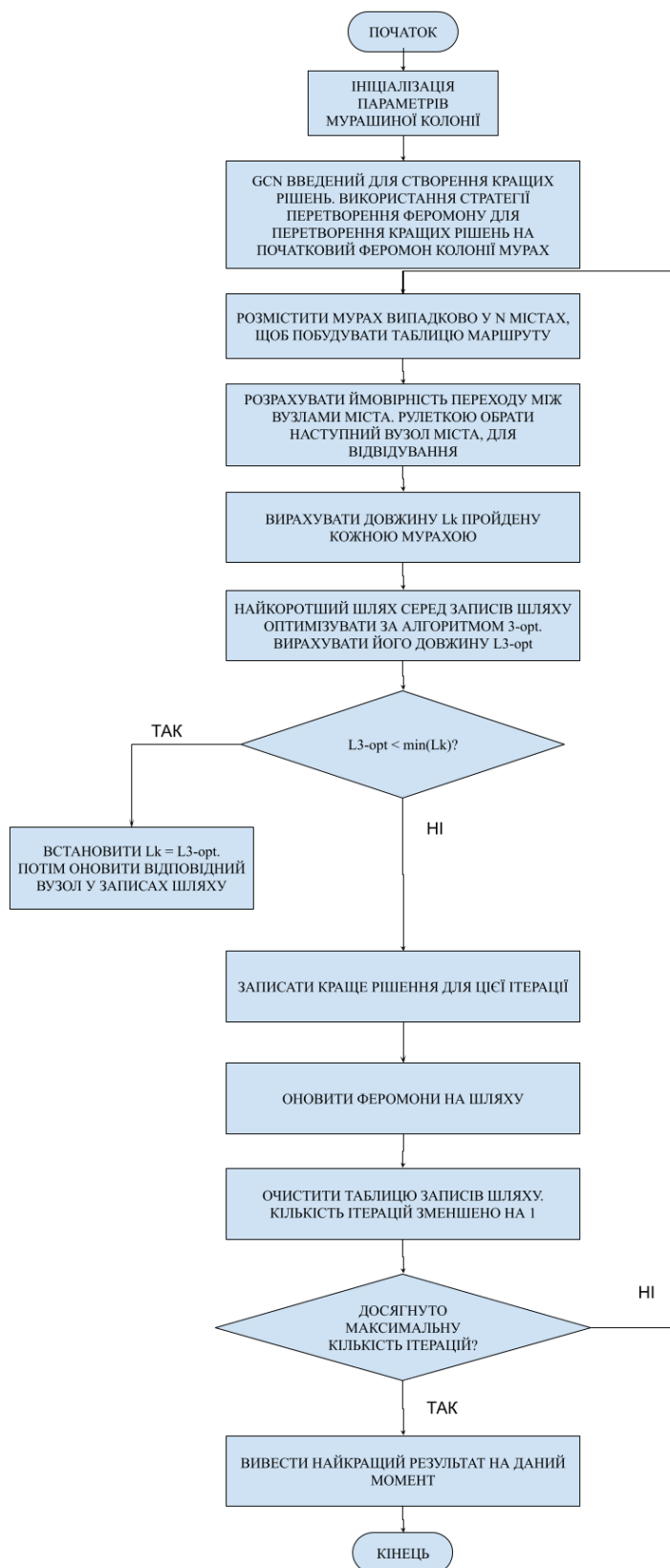


Рисунок 3.3 – блок-схема GCNIACO

Етапи реалізації алгоритму GCNIACO наступні:

Крок 1: ініціалізація коефіцієнтів, пов'язаних з оптимізацією мурашиної колонії, та обчислення матриці відстані між вузлами;

Крок 2: ініціалізація графової згорткової мережі та створення кращого рішення; потім перетворити краще рішення у початковий феромон колонії мурах за допомогою стратегії перетворення феромонів;

Крок 3: навмання виберіть перший міський вузол, щоб почати рух для мурах; обчислити ймовірність переходу для кожної мурахи між вузлами міста відповідно до рівняння (3.1); вибрати наступний вузол міста для відвідування рулеткою та записати його в таблицю записів шляху; повторюйте цей процес, доки всі мурахи не відвідають усі вузли міста;

Крок 4: обчислити довжину шляху L_k , який проходить кожна мураха, у таблиці записів шляху;

Крок 5: введено алгоритм 3-opt для виконання операції де-перетинання найкращого шляху в таблиці запису шляху; якщо оптимізована довжина шляху L_{3-opt} менша за найкращий шлях $\min(L_k)$, використовуйте L_{3-opt} для оновлення $\min(L_k)$ і в той же час оновіть відповідну послідовність вузлів шляху в таблиці запису шляху; якщо L_{3-opt} більше $\min(L_k)$, перейти безпосередньо до кроку 6;

Крок 6: записати найкраще рішення в поточному номері ітерації;

Крок 7: згідно з рівняннями (3.2) – (3.4), оновіть феромон на шляху, де коефіцієнт випаровування феромону обчислюється відповідно до рівняння (3.10); потім очистіть таблицю запису шляху та перейдіть до кроку 3, щоб продовжити ітерацію;

Крок 8: коли алгоритм досягне заданої максимальної кількості ітерацій NC_{max} , зупиніть ітерацію та виведіть найкраще рішення.

3.2 Порівняння результатів роботи базового і покращеного мурашиних алгоритмів

Використовуючи GCNIACO та ACO для вирішення TSP на однакової кількості міст, вузлів, аналізується ефект покращення вдосконаленого алгоритму. Кількість вузлів міста встановлюється на 20, 50, 100 і 200 відповідно, а конкретні координати вузла випадково та рівномірно відбираються у двовимірному одиничному квадраті. Налаштування параметрів GCNIACO наведено у табл. 3.1. Налаштування параметрів в ACO в основному такі ж, як і в GCNIACO, а параметр за замовчуванням становить $\rho = 0.5$. Результати 30 моделювань наведено в табл. 3.2.

Таблиця 3.1 – Налаштування параметрів GCNIACO

Параметр	Значення
Кількість мурах m	$m = n(n - \text{кількість вузлів міста})$
Фактор феромонів a	$a = 1$
Фактор евристичної функції β	$\beta = 5$
Постійність феромонів Q	$Q = 100$
Максимальна кількість ітерацій NC_{max}	$NC_{max} = 200$

Таблиця 3.2 – Результати 30 моделювань GCNIACO і ACO

		TSP 20	TSP 50	TSP 100	TSP 200
TSP оптимальне рішення		4.0024	6.0438	7.6685	10.9946
Симуляція: кращий результат	ACO	4.0024	6.1705	8.1672	12.0719
	GCNIACO	4.0024	6.0835	7.8627	11.7334
Симуляція: гірший результат	ACO	4.1389	6.3367	8.4472	12.388
	GCNIACO	4.0024	6.2340	8.0831	12.1438
Середнє значення	ACO	4.0570	6.2497	8.2631	12.2386
	GCNIACO	4.0024	6.1602	7.9750	11.9566
Час	ACO	2.3213	29.8602	261.19449	2004.3610
	GCNIACO	10.439	43.1829	275.8855	2267.17775
Коефіцієнт відхилення %	ACO	0	2.0964	6.5032	9.7984
	GCNIACO	0	0.6569	2.324	6.7197

Де оптимальне рішення TSP – це оптимальна схема шляху, отримана за допомогою розв’язувача Concorde;

Час – середній час виконання 30 симуляцій;

Коефіцієнт відхилення вказує на ступінь відхилення між найкращим значенням, отриманим моделюванням алгоритму, та оптимальним рішенням TSP. Конкретне рівняння рівня відхилення показано у рівнянні (3.11):

$$\text{коефіцієнт відхилення} = \frac{\text{симуляція: кращий результат} - \text{TSP оптимальне рішення}}{\text{TSP оптимальне рішення}} \cdot 100\% \quad (3.11)$$

З табл. 3.2 можна дізнатися, що коли загальні значення налаштувань параметрів однакові, найкраще рішення моделювання, найгірше рішення та середнє значення, отримані GCNIACO при розв'язанні проблеми комівояжера, є набагато кращими, ніж АСО. Проте запропонована графова згортка мережі покращує оптимізацію мурашиної колонії також демонструє теорему про відсутність безкоштовного обіду [17], тобто, коли алгоритм покращує продуктивність розв'язання одного аспекту проблеми, продуктивність розв'язання з іншого боку обов'язково зменшується [18]. GCNIACO покращує здатність знаходити найкоротший шлях, подовжуючи час виконання програми.

Для задачі комівояжера масштабу 20 міст як GCNIACO, так і АСО можуть знайти таке ж оптимальне рішення, як розв'язувач Concorde. Але найгірше рішення та середнє значення, отримане за допомогою моделювання GCNIACO, на 0.1365 та 0.0546 менше, ніж АСО відповідно. Для проблеми комівояжера з 50, 100 і 200 містами, хоча ані GCNIACO, ані АСО не знаходять того самого або меншого шляху, ніж оптимальне рішення TSP, GCNIACO все ще демонструє кращу продуктивність, ніж АСО. Найкраще рішення та середнє значення, отримані GCNIACO для моделювання TSP 50, відповідно на 0.087 та 0.0895 менші, ніж АСО, а коефіцієнт відхилення також на 1.4395% менший. Найкраще рішення та середнє значення, отримані GCNIACO для моделювання TSP 200, відповідно на 0.3385 і 0.282 менші, ніж АСО, а коефіцієнт відхилення також на 3.0787% менший. Для задачі комівояжера в 100 містах найгірше рішення, отримане моделюванням GCNIACO, навіть на 0.0841 менше, ніж найкраще рішення моделювання АСО, а коефіцієнт відхилення на 3.9708% менший.

Таким чином, GCNIACO має кращу продуктивність рішення, ніж АСО. Щоб більш інтуїтивно відобразити переваги рішення GCNIACO, у статті представлено шляхи оптимізації для вирішення TSP 50 і криві оптимізації двох алгоритмів для вирішення TSP 20, TSP 50, TSP 100 і TSP 200, як показано на малюнках 3.4–3.8 відповідно.

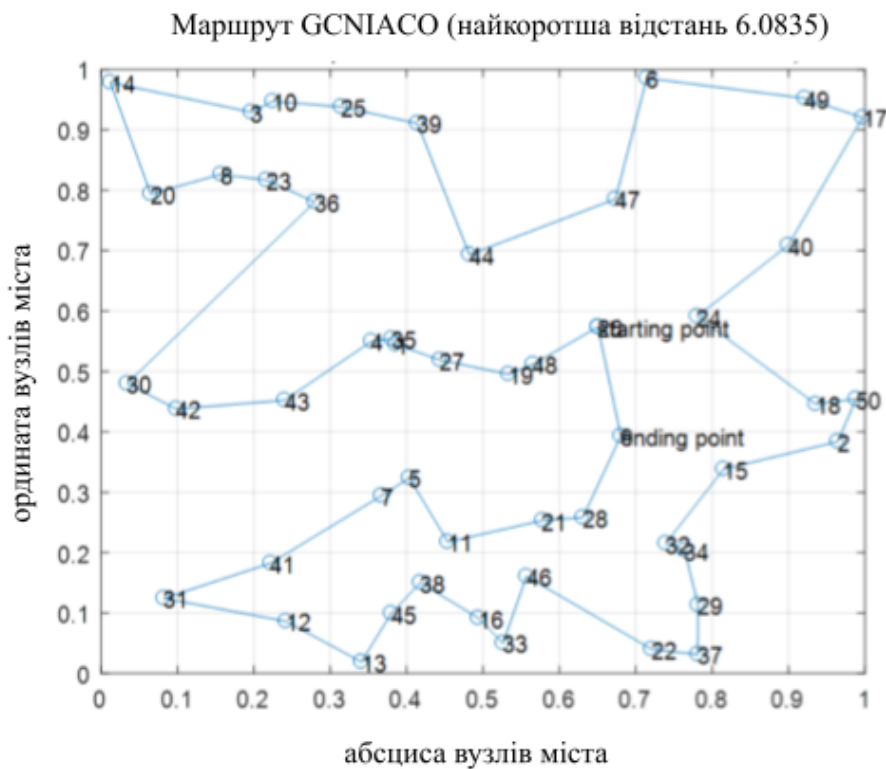
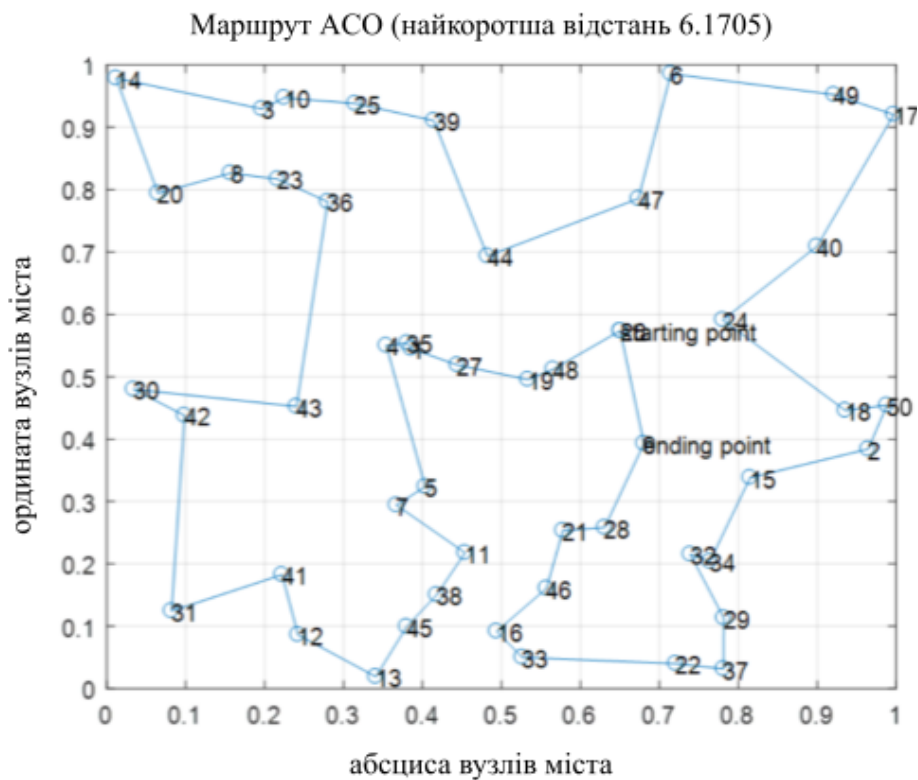


Рисунок 3.4 – АСО і GCNIACO кращі шляхи вирішення TSP 50

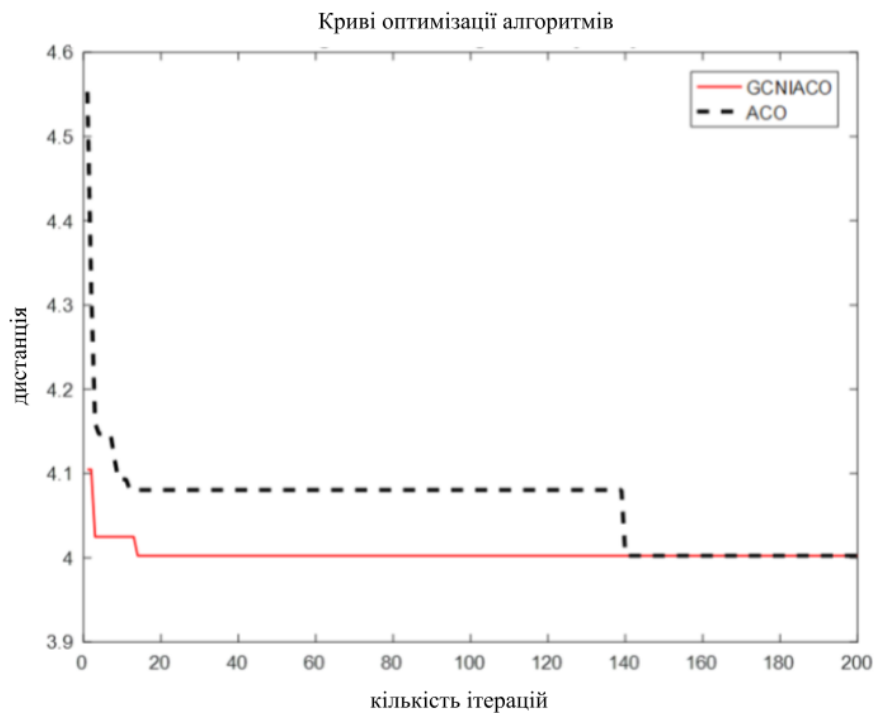


Рисунок 3.5 – Криві оптимізації ACO і GCNIACO TSP 20

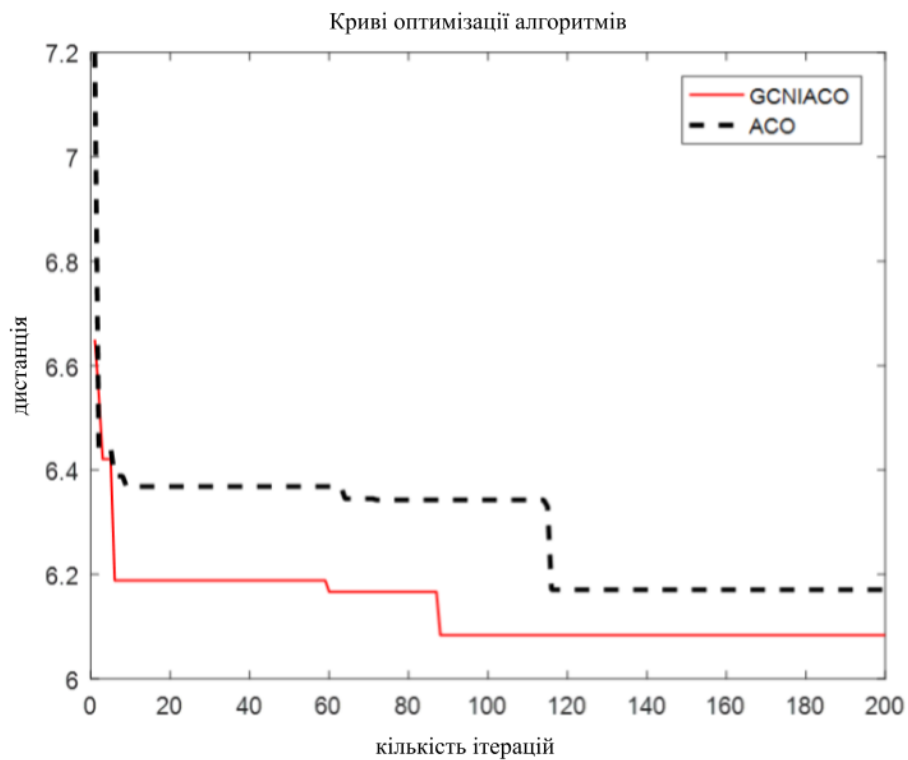


Рисунок 3.6 – Криві оптимізації ACO і GCNIACO TSP 50

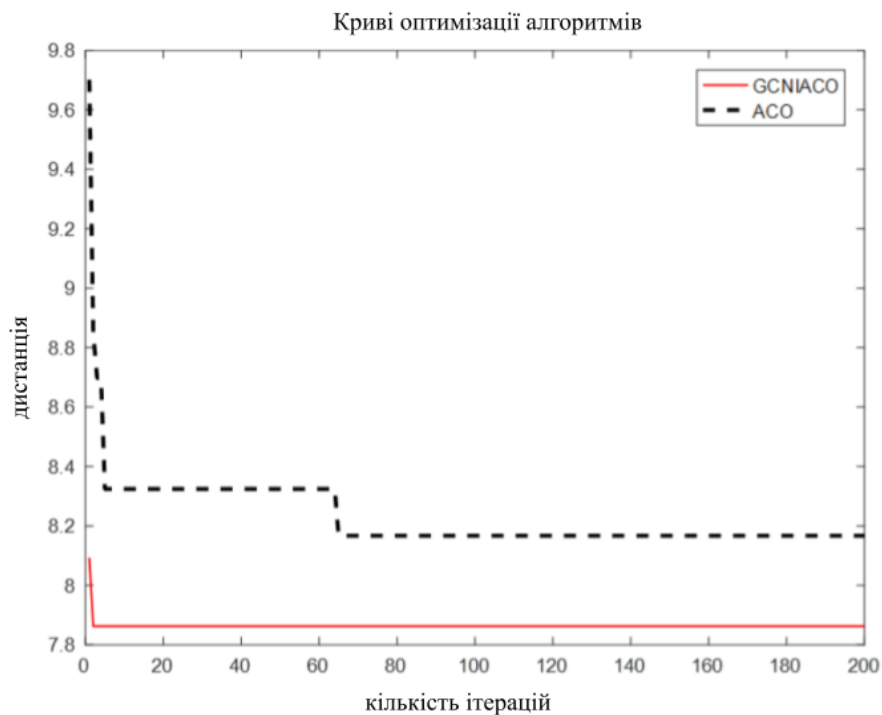


Рисунок 3.7 – Криві оптимізації ACO і GCNIACO TSP 100

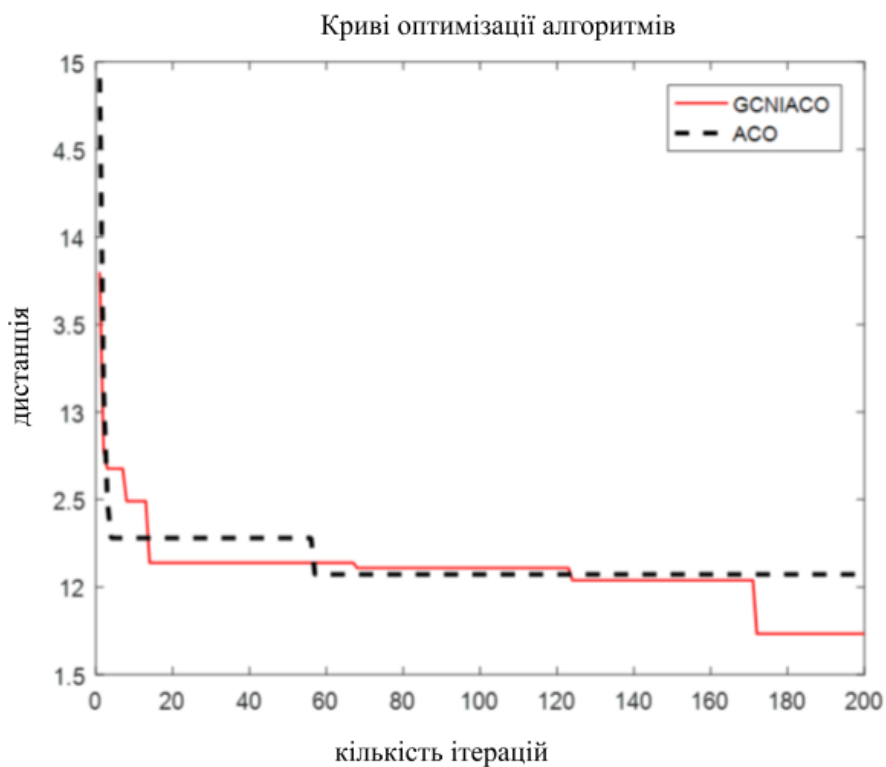


Рисунок 3.8 – Криві оптимізації ACO і GCNIACO TSP 200

3.3 Проектування системи та її інтерфейсу користувача

Використовуючи можливості Інтернету, веб-додаток TSP може полегшити користувачам введення даних, візуалізацію проблеми та отримання рішень. Крім того, веб-додаток може надати платформу для співпраці та обміну рішеннями, дозволяючи користувачам працювати разом, щоб знайти найкраще рішення для своїх проблем TSP. Крім того, веб-програма може бути розроблена так, щоб вона була масштабованою, що дозволяє їй вирішувати великі проблеми TSP і надавати рішення для складних сценаріїв реального світу.

Крім того, використовуючи розширені функції, такі як візуалізація в реальному часі, інтерактивні карти та аналітика даних, веб-додаток TSP може стати потужним інструментом для підприємств і організацій, які мають справу зі складними проблемами транспортування та логістики.

Проблема TSP – це лише одна з багатьох проблем оптимізації, які можна вирішити за допомогою веб-додатку. Поєднуючи потужність алгоритмів і гнучкість веб-технологій, розробники можуть створювати інноваційні рішення для широкого кола проблем оптимізації, які можуть мати значний вплив на різні галузі та сфери.

Для побудови застосунку на базі покращеного мурашиного алгоритму для вирішення задачі комівояжера необхідно звернути увагу на наступні кроки:

Крок 1: Розуміння проблеми TSP. Першим кроком у програмуванні веб-додатку для вирішення проблеми TSP є розуміння самої проблеми. Задачу TSP можна представити у вигляді графа, де кожне місто представлено вузлом, а відстані між містами представлені ребрами. Мета полягає в тому, щоб знайти найкоротший можливий маршрут, який відвідує кожне місто рівно один раз і повертається до початкового міста.

Крок 2: Вибір алгоритму. Існує кілька алгоритмів, які можна використовувати для вирішення проблеми TSP, у тому числі точні алгоритми, такі

як розгалуження та зв'язки та динамічне програмування, а також евристичні алгоритми, такі як найближчий сусід і імітований відпал. Вибір алгоритму залежатиме від конкретних вимог веб-додатку та розміру проблеми TSP. У розробці цього застосунку було використано розглянутий вище алгоритм GCNIACO.

Крок 3: Розробка інтерфейсу користувача. Інтерфейс користувача повинен дозволяти користувачам вводити дані для проблеми TSP, такі як міста, початкова точка та відстані між кожним містом. Він також повинен забезпечувати візуальне представлення проблеми TSP, наприклад карту або графік, і відображати рішення після його розрахунку.

Крок 4: Створення Back-end. Верхня частина веб-додатку відповідатиме за вирішення проблеми TSP за допомогою вибраного алгоритму та збереження даних про проблему TSP кожного користувача в базі даних. Сервер має бути побудований з використанням серверної мови, такої як PHP, Python або Ruby, і повинен мати можливість обробляти кілька запитів від користувачів одночасно.

Крок 5: Зберігання даних у базі даних. База даних буде використовуватися для зберігання даних для кожної проблеми TSP, таких як міста, відстані та рішення. Ці дані можна використовувати для відстеження прогресу користувача та надання уявлення про те, як можна оптимізувати проблему TSP. База даних має бути розроблена таким чином, щоб бути масштабованою та здатною обробляти великі обсяги даних.

Крок 6: Розгортання веб-програми. Веб-додаток розгортається на хмарній платформі, такій як Amazon Web Services або Google Cloud, для легкого доступу користувачів. Процес розгортання включатиме налаштування сервера, конфігурацію бази даних і розгортання інтерфейсу та серверної частини програми.

Крок 7: Тестування та налагодження. Веб-програму слід ретельно протестувати перед розгортанням, щоб переконатися, що вона надає точні рішення для проблеми TSP і проста у використанні. Буде проведено тестування

прийнятності для користувачів, щоб переконатися, що додаток відповідає очікуванням цільової аудиторії. Налаштування слід виконувати за необхідності, щоб виправити будь-які помилки або проблеми, виявлені під час тестування.

Крок 8: Технічне обслуговування та підтримка. Після розгортання веб-додатку для забезпечення його безперервної роботи буде необхідне регулярне обслуговування та підтримка. Це включатиме моніторинг продуктивності сервера та бази даних, виправлення помилок і оновлення алгоритму для покращення його продуктивності. Крім того, буде надано підтримку користувачів для вирішення будь-яких питань або проблем, які можуть виникнути з програмою.

Крок 9: Аналітика. Веб-програма повинна мати можливість збирати та аналізувати дані про використання та продуктивність алгоритму TSP. Ця інформація може бути використана для визначення областей для вдосконалення та оптимізації алгоритму для кращої продуктивності.

Крок 10: Безпека. Безпека має бути головним пріоритетом для веб-програми, оскільки вона оброблятиме конфіденційні дані, такі як інформація користувача та рішення TSP. Необхідно вжити відповідних заходів, наприклад шифрування, щоб захистити дані від несанкціонованого доступу.

Для візуалізації мапи використано сервіс OpenStreetMap – це ініціатива зі створення та надання всім безкоштовних географічних даних, наприклад карт вулиць. OpenStreetMap Foundation – це міжнародна некомерційна організація, яка підтримує, але не контролює проект OpenStreetMap. Він спрямований на заохочення зростання, розвитку та розповсюдження безкоштовних геопросторових даних, а також надання геопросторових даних для використання та обміну будь-ким. OpenStreetMap (OSM) – це спільний краудсорсинговий проект із відкритим вихідним кодом, спрямований на створення безкоштовної карти світу, яку можна редагувати. Він був заснований у 2004 році і з тих пір перетворився на одне з найбільших джерел географічних даних із мільйонами учасників по всьому світу. Дані карти в OpenStreetMap постійно оновлюються волонтерами та

організаціями, що робить їх дуже точним і актуальним джерелом інформації. Дані доступні для використання будь-ким без обмежень і можуть бути інтегровані в широкий спектр програм, включаючи веб-карти та карти для мобільних пристроїв, маршрутизацію та системи географічної інформації (ГІС). OpenStreetMap став цінним ресурсом для урядів, підприємств і окремих осіб, що забезпечує безкоштовний доступ до якісних картографічних даних.

Для імплементації OpenStreetMap було використано Geofabrik – сервер містить витяги даних із проекту OpenStreetMap, які зазвичай оновлюються щодня. Можно обрати свій континент, а потім необхідну країну зі списку. Цю послугу завантаження відкритих даних пропонує Geofabrik GmbH безкоштовно.

Далі за допомогою довідки і рішень що так само надає API OpenStreetMap було стягнуто дані про дороги і траси, на основі яких алгоритм будував маршрут.

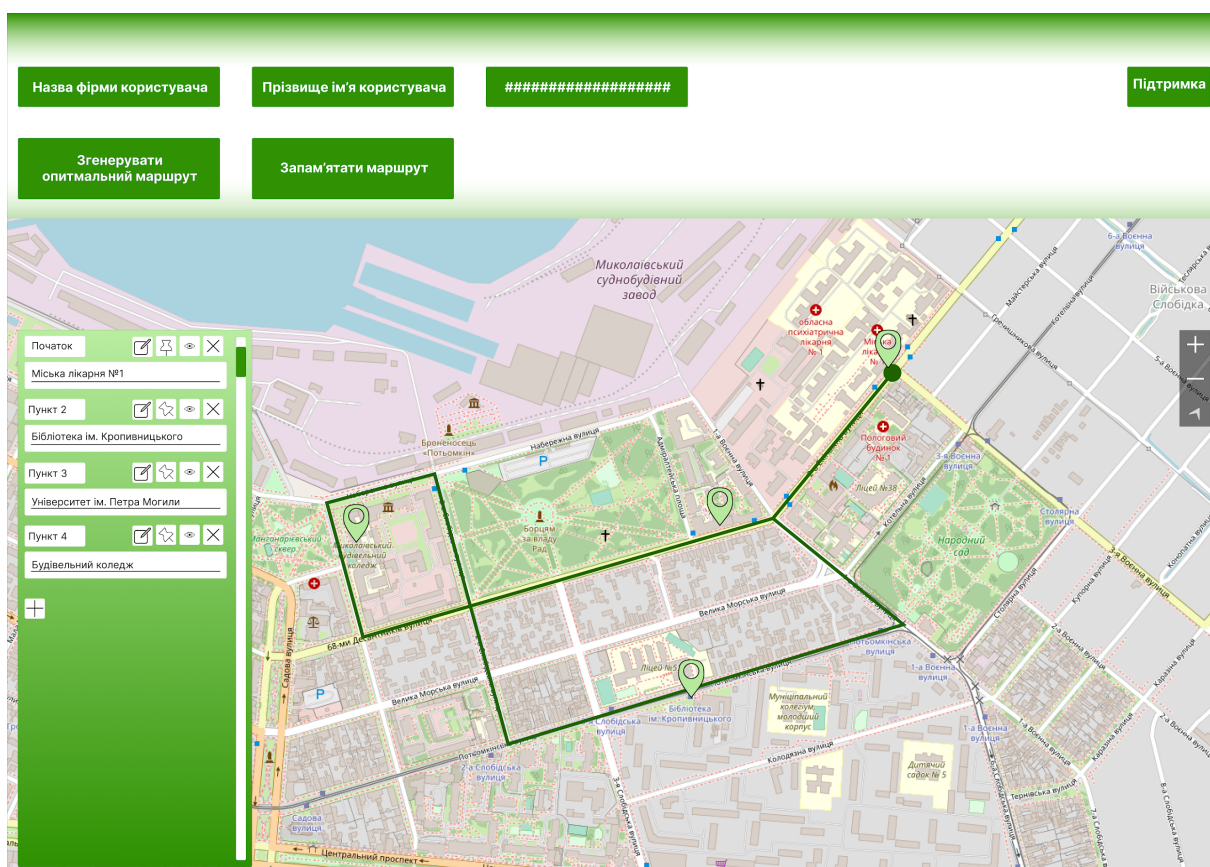


Рисунок 3.9 – Інтерфейс застосунку

На рис. 3.9 зображено інтерфейс застосунку для пошуку оптимального шляху на приклад маршруту доставки медикаментів у Миколаєві з Міської лікарні №1. Введено 4 пункти маршруту, 1 пункт закріплено – він став початком або “базою”. Далі кнопкою “Згенерувати маршрут” будується найкоротший шлях який обходить всі 4 пункти і вертається на “базу”. Є можливість запам’ятати маршрут, перемістити черговість уже відсортованих пунктів на власний розсуд. Недолік даної реалізації є накладання маршруту на уже існуючі напрямні, що ускладнює навігацію.

Висновки до розділу 3

У даному розділі досліджено особливості алгоритму графа згорткової мережі покращеної мурашиної колонії, наведені приклади уже існуючих модифікацій, порівняно результати роботи базового і покращеного мурашиних алгоритмів за сталих налаштувань, спроектована застосунок та інтерфейс веб-програми для використання наведених алгоритмів.

Результати симуляційного випробування покращеної оптимізації графової згорткової мережі на наборах даних TSP підтверджують продуктивність покращеного алгоритму в пошуку оптимального рішення. Він може бути застосований до задач оптимізації в різних сферах.

Крім того, GCNI-ACO, як вдосконалений алгоритм базової оптимізації мурашиної колонії, може отримати відносно кращі результати при вирішенні задач оптимізації.

Програмування веб-додатку для вирішення проблеми комівояжера вимагає поєднання технічних навичок веб-розробки, алгоритмів і керування базами даних. Однак за умови ретельного планування та проектування можна створити веб-програму, яка надасть зручне та точне рішення для користувачів.

ВИСНОВКИ

У першому розділі розглянуто сучасні задачі транспортної логістики як: транспортування, розподілення, призначення, потік транспортної мережі, задачу комівояжера. Проведено порівняльний аналіз існуючих систем вирішення цих задач, а саме: TMS Planarry, ANT Logistics, OptimoRoute. Досліджено особливості та перспективи застосування інтелектуальних методів та алгоритмів для оптимізації логістичних операцій.

У другому розділі досліджено задачі пошуку оптимального шляху при плануванні транспортних маршрутів TSP, її математичне формулювання і способи розв'язання в частості методом гілок і меж, проаналізовано математичні методи оптимізації транспортних маршрутів, досліджено мурашиний алгоритм та його базові модифікації.

У третьому розділі досліджено особливості алгоритму графа згорткової мережі покращеної мурашиної колонії, наведені приклади уже існуючих модифікацій, порівняно результати роботи базового і покращеного мурашиних алгоритмів за сталих налаштувань, спроектована застосунок та інтерфейс веб-програми для використання наведених алгоритмів.

Підсумовуючи, використання інтелектуальних методів і алгоритмів у логістичних операціях пропонує підприємствам потужний інструмент для оптимізації їх операцій, зниження витрат і підвищення рівня задоволеності клієнтів. Використовуючи потужність алгоритмів і машинного навчання, підприємства можуть оптимізувати свої логістичні операції в режимі реального часу, приймаючи кращі рішення та швидко реагуючи на зміни обставин. Хоча ці методи й алгоритми можуть принести значні переваги, їх впровадження також потребує ретельного планування та виконання, а також інвестицій у необхідні технології та інфраструктуру.

Мурашиний алгоритм являє собою збільшення напрямних, які отримали на залежному від мурашиних феромонів шляху. Це досягається шляхом заміни дискретного розподілу ймовірностей неперервною функцією, яка називається функцією щільності ймовірності, також відомою як розподіл Гауса. Таким чином АСО можна використовувати для вирішення не лише TSP, а й інших графових задач.

Результати симуляційного випробування покращеної оптимізації графової згорткової мережі на наборах даних TSP підтверджують продуктивність покращеного алгоритму в пошуку оптимального рішення. Він може бути застосований до задач оптимізації в різних сферах.

Крім того, GCNI-ACO, як вдосконалений алгоритм базової оптимізації мурашиної колонії, може отримати відносно кращі результати при вирішенні задач оптимізації.

Програмування веб-додатку для вирішення проблеми комівояжера вимагає поєднання технічних навичок веб-розробки, алгоритмів і керування базами даних. Однак за умови ретельного планування та проектування можна створити веб-програму, яка надасть зручне та точне рішення для користувачів.

У цьому дослідженні запропоновано покращену оптимізацію мурашиної колонії згорткової мережі графів. Шляхом впровадження графової згорткової мережі, динамічно настроюваного фактора нестійкості феромонів та впровадженням алгоритму з 3 опціями компенсуються недоліки АСО у розв'язанні TSP. У дослідженні поєднується графова згортка для покращення оптимізації мурашиних колоній, що дає ідею поєднання машинного навчання та інтелекту роя. При цьому вибирається модель узагальненої графової згорткової мережі, і відповідна графова згорткова мережа не потребує навчання для кожного конкретного масштабу вузла міста, що знижує експлуатаційні витрати та певною мірою економить обчислювальні ресурси. Результати симуляційного випробування покращеної оптимізації графової згорткової мережі на наборах даних TSP

підтверджують продуктивність покращеного алгоритму в пошуку оптимального рішення. GCNIACO – це метаевристичний алгоритм, заснований на інтелекті роїв. Він може бути застосований до задач оптимізації в різних сферах, таких як вирівнювання біологічної послідовності, усунення дефектів забруднення цифрового мікрофлюїдного біочіпа, вибір довжини хвилі молекулярного спектру, молекулярний докінг, виявлення епістазу повсюдногеномної асоціації тощо.

У дослідженні адаптивно покращено лише фактор мінливості феромонів в оптимізації колонії мурах. На додаток до поєднання з графовою згортковою мережею, поєднання оптимізації мурашиних колоній з іншими нещодавно запропонованими алгоритмами також є одним із майбутніх напрямків дослідження. Застосування вдосконаленого алгоритму для вирішення практичних завдань у реальному житті, таких як проблема розташування логістичного розподільчого центру, задача планування шляху робота тощо, також є напрямком, який варто розглянути для майбутніх досліджень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Навчальний посібник «Лінійне програмування» для студентів напрямів підготовки 122 Комп'ютерні науки та 121 Інженерія програмного забезпечення / О. О. Ємець, О. С. Пічугіна, О. Б. Маций, К. П. Коробчинський. – Х. : ХНАДУ, 2019. 102 с. URL: <https://dSPACE.khadi.kharkov.ua/dSPACE/bitstream/123456789/3448/1/%D0%9B%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F.pdf> (дата звернення: 26.12.2022).
2. Конспект лекції № 6 Тема № 6. Транспортна задача: навчальний матеріал. URL: <https://financial.lnu.edu.ua/wp-content/uploads/2019/09/ME-lektsiia-6.pdf> (дата звернення: 26.12.2022).
3. Тема 6. Розподільча логістика: навчальний матеріал. URL: <https://mk.nmu.org.ua/en/source/Logistic16.pdf> (дата звернення: 26.12.2022).
4. Задача про призначення - реферати та учбові матеріали на um.co.ua: вебсайт. URL: <http://um.co.ua/3/3-14/3-14636.html> (дата звернення: 26.12.2022).
5. Ключка Т. А., Шаповалов С. П., Довбиш А. С. Комп'ютерний порівняльний аналіз алгоритмів Дініца та Форда-Фалкерсона: випускна робота. Суми, 2020. 44 с. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/80062/1/Kluthka_bac_rob.pdf (дата звернення: 26.12.2022).
6. Задача визначення найкоротших відстаней між елементами транспортної мережі (Алгоритм Флойда). URL: https://vuzlit.com/984280/zadacha_viznachennya_naykorotshih_vidstaney_elementami_transportnoyi_merezhi_algorithm_floyda (дата звернення: 26.12.2022).

7. ▶【PLANARRY】- керування транспортними витратами | UIT. URL: <https://uit.kiev.ua/product/plannary/> (дата звернення: 02.12.2022).
8. ANT-Logistics. URL: <https://ant-logistics.com/> (дата звернення: 02.12.2022).
9. OptimoRoute. URL: <https://optimoroute.com/> (дата звернення: 02.12.2022).
10. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрік та ін. – Суми : Сумський державний університет, 2017. – 212 с.
11. Левитин А. Алгоритмы. Введение в разработку и анализ, 2016. – 453 с.
12. Metaheuristics – Scholarpedia: вебсайт. URL: <http://www.scholarpedia.org/article/Metaheuristics> (дата звернення: 21.12.2022).
13. Eusocial Definition & Meaning - Merriam-Webster: вебсайт. URL: <https://www.merriam-webster.com/dictionary/eusocial> (дата звернення: 21.12.2022).
14. Мурашиний алгоритм – Вікіпедія: вебсайт. URL: https://uk.wikipedia.org/wiki/%D0%9C%D1%83%D1%80%D0%B0%D1%88%D0%B8%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC (дата звернення: 21.12.2022).
15. Duan H. B. The Principle and Application of Ant Colony Algorithm: Science Press. Beijing, 2005. 16 с.
16. Zhong X. H. On the approximation ratio of the 3-Opt algorithm for the (1,2)-TSP, Oper. Res. Lett., 2021. 521 с. URL: <https://doi.org/10.1016/j.orl.2021.05.012> (дата звернення: 08.02.2023)
17. Wolpert D. H., Macready W. G. No free lunch theorems for optimization, IEEE Trans. Evol. Comput., 1997. 82 с. URL: <https://doi.org/10.1109/4235.585893> (дата звернення: 08.02.2023)
18. Bharati S., Podder P., Mondal M. R. H., Gandhi N. Optimized NASNet for Diagnosis of COVID-19 from Lung CT Images, Intell. Syst. Design Appl., 2021. 656 с.

URL: https://link.springer.com/chapter/10.1007/978-3-030-71187-0_59 (дата звернення: 08.02.2023)

19. ТРАНСПОРТНА ЛОГІСТИКА, Сутність і завдання транспортної логістики, Види транспорту - Логістика - Навчальні матеріали онлайн: вебсайт. URL: https://pidru4niki.com/18800413/ekonomika/transportna_logistika (дата звернення: 26.12.2022).

20. Завдання транспортної логістики від NEOLIT™: вебсайт. URL: <https://neolit.ua/ua/articles/82> (дата звернення: 26.12.2022).

21. Козар Л.М., Романович Є.В., Афанасов Г.М. Методи транспортної логістики: навч. посіб. Харків, 2015. 174 с. URL: <http://lib.kart.edu.ua/bitstream/123456789/2677/1/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA.pdf> (дата звернення: 26.12.2022).

22. Соловйова О. О., Висоцька І. І., Герасименко І. М. Загальний курс транспорту: навч. посіб. Київ, 2019. 244 с. URL: <https://er.nau.edu.ua/bitstream/NAU/43642/1/%D0%A1%D0%BE%D0%BB%D0%BE%D0%B2%D0%B9%D0%BE%D0%B2%D0%B0%20%D0%92%D0%B8%D1%81%D0%BE%D1%86%D1%8C%D0%BA%D0%B0.pdf> (дата звернення: 26.12.2022).

23. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ, 2016. 26 с. URL: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF (дата звернення: 21.12.2022).

24. ДСТУ 2873-94. Системи оброблення інформації. Програмування. Терміни та визначення.

25. Кондратенко Ю. П. Оптимізація процесів прийняття рішень в умовах невизначеності. Методичні вказівки до курсового проектування з дисципліни «Теорія прийняття рішень». Миколаїв : ЧДУ ім. Петра Могили, 2005. 46 с.

26. Вимоги до оформлення дисертації : затв. наказом МОН України від 12.01.2017 No 40. URL: <https://zakon.rada.gov.ua/laws/show/z0155-17> (дата звернення: 21.12.2022).
27. Кондратенко Ю.П., Сіденко Є. В., Кулаковська І. В. Методичні рекомендації до виконання магістерської кваліфікаційної роботи. Миколаїв, 2020. 58 с.
28. Левитин А. Алгоритмы. Введение в разработку и анализ, 2016. – 453 с.
29. Introduction to Ant Colony Optimization - GeeksforGeeks: вебсайт. URL: <https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/> (дата звернення: 21.12.2022).
30. Metaheuristics – Scholarpedia: вебсайт. URL: <http://www.scholarpedia.org/article/Metaheuristics> (дата звернення: 21.12.2022).
31. Eusocial Definition & Meaning - Merriam-Webster: вебсайт. URL: <https://www.merriam-webster.com/dictionary/eusocial> (дата звернення: 21.12.2022).
32. Мурашиний алгоритм – Вікіпедія: вебсайт. URL: https://uk.wikipedia.org/wiki/%D0%9C%D1%83%D1%80%D0%B0%D1%88%D0%B8%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC (дата звернення: 21.12.2022).
33. Dorigo M., Maniezzo V., Colorni A. Ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics - Part B, No.1, 1996. 26 с.
34. Субботин С.А., Олейник Ан.А., Олейник Ал.А. Интеллектуальные мультиагентные методы (swarm intelligence) –http://csit.narod.ru/subject/МА/МА_lect.pdf (дата звернення: 22.12.2022).
35. Кирсанов М. Н. Графы в Maple. М.: Физматлит, 2017. 168 с. <http://vuz.exponenta.ru/PDF/book/GrMaple.pdf> (дата звернення: 22.12.2022).
36. Ali I. M., Essam D., Kasmarik K. A novel design of differential evolution for solving discrete traveling salesman problems, Swarm Evol. Comput, 2020. 52 с.

37. Akhand M. A. H., Ayon S. I., Shahriyar S. A., Siddique N., Adeli H. Discrete spider monkey optimization for travelling salesman problem, *Appl. Soft Comput.*, 2020. 87 с. URL: <https://doi.org/10.1016/j.asoc.2019.105887> (дата звернення: 20.01.2023).
38. Deudon M., Cournut P., Lacoste A., Adulyasak Y., Rousseau L. M. Learning heuristics for the TSP by policy gradient, in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer, Cham, 10848, 2018. 181 с.
39. Prates M. O. R., Avelar P. H. C., Lemos H., Lamb L., Vardi M. Learning to solve np-complete problems: A graph neural network for decision tsp, in *Proceedings of the AAAI Conference on Artificial Intelligence 2019*. 4738 с.
40. Hu Y. J., Zhang Z., Yao Y., Huyan X. P., Zhou X. S., Lee W. S. A bidirectional graph neural network for traveling salesman problems on arbitrary symmetric graphs, *Eng. Appl. Artif. Intell.*, 2021. 97 с.
41. Ragmani A., Elomri A., Abghour N., Moussaid K., Rida M. An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment, *Procedia Comput. Sci.* 2019. 526 с.
42. Ebadinezhad S. DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem, *Eng. Appl. Artif. Intell.* 2020. 92 с.
43. Li J., Xia Y., Li B., Zeng Z. G. A pseudo-dynamic search ant colony optimization algorithm with improved negative feedback mechanism, *Cognit. Syst. Res.*, 2020. 9 с.
44. Fei T., Zhang L. Y., Sun Y. S. Solution of vehicle routing optimization problem based on DNA-ant colony algorithm, *Comput. Eng.* 2014. 213 с.
45. Wang Z. F., Du H. L., An S. F., Zhang C. J. An improved ant colony algorithm based on vehicle routing problem, *J. Huaqiao Univ. (Nat. Sci.)*. 2013. 39 с.

46. Охорона праці в офісі. Вимоги до робочого місця офісного працівника.

URL:<https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/> (дата звернення: 18.12.2022).

47. Практикум із охорони праці. Навчальний посібник / В.Ц. Жидецький, В.С. Джигирей, В.М. Сторожук та ін. за ред. В.Ц. Жидецького. Львів: Афіша, 2000. 352 с.

48. Копань В.С. Охорона праці в гуманітарній сфері: Навчальний посібник. К.: Університетське вид-во ПУЛЬСАРИ, 2007. 200 с.

49. Фабрика Меблів. Проектування приміщення. URL: <http://8marta.planningwiz.com> (дата звернення: 18.12.2022).

50. Желібо Є. П., Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. К., 2002. 162 с.

51. Пособие к СНиП II-4-79 «Пособие по расчету и проектированию естественного, искусственного и совмещенного освещения». URL: http://tehlit.ru/1lib_norma_doc/6/6141/index.htm (дата звернення: 18.12.2022).