

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

**на тему: «Обґрунтування ефективного мурашкового алгоритму
для формування маршрутів заготівлі продовольчої
сировини»**

Виконав: студент 4 курсу групи Кн–41

Спеціальності 122 «Комп'ютерні науки»
(шифр і назва)

Штокалко Роман Богданович
(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Кригуль Р.Є.
(Прізвище та ініціали)

ДУБЛЯНИ–2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А. М. Тригуба

«_____» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Штокалку Роману Богдановичу

1. Тема роботи: «Обґрунтування ефективного мурашкового алгоритму для формування маршрутів заготівлі продовольчої сировини»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 25.02.2025 року № 123/к–с.

2. Строк подання студентом роботи 10.06.2025 р.

3. Вихідні дані до роботи: статистичні дані обсягів заготівлі продовольчої сировини; вимоги до продовольчої сировини.

4. Зміст розрахунково–пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

1. Аналіз стану питання щодо формування маршрутів.

2. Вибір інструментарію для обґрунтування ефективного мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

3. Результати обґрунтування ефективного мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

4. Практичне використання мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

5. Охорона праці.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень): аналіз стану питання щодо формування маршрутів; вибір інструментарію для обґрунтування ефективного мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини; результати обґрунтування ефективного мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини; практичне використання мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	Тригуба А.М., зав. кафедри ІТ		
5	Тимочко В.О., доцент кафедри інженерної механіки		

7. Дата видачі завдання

25 лютого 2025 р.

Календарний план

№ з/П	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	При-мітка
1	Написання першого розділу	25.02–11.03.25	
2	Виконання другого розділу та аркушів ілюстраційного матеріалу до нього	12.03–15.04.25	
3.	Виконання третього розділу та аркушів ілюстраційного матеріалу до нього	16.04–20.05.25	
4.	Написання розділу «Охорона праці»	21.05–21.05.25	
5.	Завершення оформлення розрахунково–пояснювальної записки та аркушів ілюстраційного матеріалу	22–31.05.25	
6.	Завершення роботи в цілому	01–10.06.25	

Студент _____ Штокалко Р.Б.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 004.852:519.876.2:631.563

Обґрунтування ефективного мурашкового алгоритму для формування маршрутів заготівлі продовольчої сировини.

Штокалко Р.Б. Кафедра ІТ – Дубляни, Львівський НУВМ, 2025.

Кваліфікаційна робота: 67 с. текст. част., 16 рис., 5 табл., 14 арк. ілюстраційного матеріалу, 41 джерело.

У роботі проаналізовано особливості формування маршрутів заготівлі продовольчої сировини з використанням інтелектуальних підходів, зокрема алгоритму оптимізації мурашиних колоній (ACO). Проведено огляд сучасних методів логістичної маршрутизації, класифіковано алгоритмічні підходи до побудови транспортних маршрутів та обґрунтовано доцільність застосування мурашкових алгоритмів у аграрній логістиці. З урахуванням виробничих умов, характерних для сільськогосподарського сектору в умовах обмеженого доступу або надзвичайних ситуацій, було розроблено удосконалений варіант ACO-алгоритму.

Запропонована модель включає модифіковані правила поведінки агентів-мурах, механізм урахування стану дорожнього полотна та функцію адаптивного перерахунку маршруту. Здійснено вибір відповідного програмного середовища, реалізовано алгоритмічну модель у середовищі Python, а також виконано її тестування на низці сценаріїв з використанням вхідних даних про стан транспортної мережі. Отримані результати демонструють переваги вдосконаленого підходу у порівнянні з класичними алгоритмами як за критерієм точності, так і за швидкістю знаходження оптимального маршруту.

Розроблено охорони праці під час розробки та використання алгоритму для формування маршрутів заготівлі продовольчої сировини.

Ключові слова: мурашковий алгоритм ACO, маршрутизація, продовольча сировина, оптимізація, логістика.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ ЩОДО ФОРМУВАННЯ МАРШРУТІВ.....	9
1.1. Методи оптимізації маршрутів транспортування сільськогосподарської продукції	9
1.2. Класифікація сучасних алгоритмів вибору маршрутів доставки вантажів та їх особливості	13
1.3. Принципи побудови мурашкових алгоритмів та їх застосування у логістиці	16
1.4. Стан використання інтелектуальних методів формування маршрутів в аграрному секторі.....	19
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ ОБҐРУНТУВАННЯ ЕФЕКТИВНОГО МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ.....	22
2.1. Мета та завдання роботи	22
2.2. Вибір мови програмування та програмного середовища.....	23
2.3. Структура та етапи реалізації алгоритму	26
2.4. Визначення параметрів мурашкового алгоритму (АСО).....	28
2.5. Побудова цільової функції маршрутизації	30
2.6. Засоби візуалізації результатів та аналітичного контролю	33
РОЗДІЛ 3. РЕЗУЛЬТАТИ ОБҐРУНТУВАННЯ ЕФЕКТИВНОГО МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ.....	34
3.1. Особливості визначення виробничих умов заготівлі продовольчої сировини	34
3.2. Результати визначення виробничих умов заготівлі продовольчої сировини	35
3.2. Опис правил поведінки мурах в удосконаленому мурашковому	

алгоритмі	36
3.4. Блок-схема удосконаленого мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини	39
РОЗДІЛ 4. ПРАКТИЧНЕ ВИКОРИСТАННЯ МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ	42
4.1. Програмна реалізація вдосконаленого алгоритму АСО для формування маршрутів заготівлі продовольчої сировини	42
4.2. Результати порівняння вдосконаленого алгоритму АСО із існуючими для задач формування маршрутів заготівлі продовольчої сировини	45
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	48
5.1. Аналіз стану робочого місця та умов праці.....	48
5.2. небезпечні та шкідливі чинники	50
5.3. Інструкція із охорони праці під час розробки та використання алгоритму для формування маршрутів заготівлі продовольчої сировини.....	52
ВИСНОВКИ І ПРОПОЗИЦІЇ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	64

ВСТУП

У сучасному агропромисловому виробництві основною складовою забезпечення ефективності господарської діяльності є оптимізація логістичних процесів. Це пов'язано з транспортуванням сільськогосподарської продукції від місць її первинного виробництва до пунктів зберігання, переробки або реалізації [18]. Особливої ваги ці завдання набувають щодо заготівлі продовольчої сировини, де час, маршрут і витрати є досить важливими параметрами, що впливають не лише на економічні показники підприємства, а й на якість продукції. Своєчасне та раціональне формування маршрутів дає змогу знизити витрати пального, скоротити зношення техніки, уникнути втрат продукції внаслідок затримок, а також забезпечити екологічну безпеку шляхом мінімізації викидів.

Разом із цим, традиційні методи побудови маршрутів, до яких належать жорстко детерміновані або евристичні алгоритми, часто виявляються недостатньо ефективними при великій кількості об'єктів, складних транспортних мережах та змінних умовах виконання завдань [11]. У таких умовах доцільно звернутися до методів обчислювального інтелекту, які здатні адаптуватися до змін середовища, здійснювати пошук оптимальних рішень у багатовимірних просторах та враховувати численні обмеження. Серед таких методів особливе місце займає мурашковий алгоритм – метавевристичний підхід, заснований на принципах поведінки соціальних комах, а саме мурах, під час пошуку ефективних маршрутів до джерел їжі.

Мурашковий алгоритм є одним із найвідоміших прикладів стохастичного оптимізаційного методу, що був запропонований наприкінці 1990-х років як ефективний інструмент розв'язання задач комівояжера, планування логістичних маршрутів, управління потоками транспорту та іншого роду задач комбінаторної оптимізації [15]. Його основною перевагою є здатність одночасного дослідження декількох альтернативних маршрутів з використанням колективного механізму пам'яті (феромонного поля) та

постійного самонавчання. Це дозволяє швидко ітеративно зближуватись до глобального оптимуму навіть у випадках, коли кількість можливих варіантів перевищує мільйони комбінацій.

Актуальність теми кваліфікаційної роботи зумовлена тим, що сільське господарство сьогодні переживає цифрову трансформацію. Збирається велика кількість геопросторових, логістичних та економічних даних, які потребують гнучких алгоритмів для їх ефективного використання. Водночас, у багатьох підприємствах досі використовуються ручні або неадаптивні підходи до побудови маршрутів заготівлі, що призводить до необґрунтованих витрат ресурсів і втрати потенційного прибутку. Тому впровадження адаптивного мурашкового алгоритму, здатного враховувати динамічні фактори та географічні особливості сільськогосподарських регіонів, є своєчасним та важливим кроком до підвищення ефективності всієї системи логістичного управління.

Метою кваліфікаційної роботи є розробка та обґрунтування ефективного варіанту мурашкового алгоритму, що дозволяє формувати оптимальні маршрути заготівлі продовольчої сировини з урахуванням специфіки аграрного виробництва, логістичних обмежень, особливостей дорожньої інфраструктури, обсягів продукції та часових обмежень.

РОЗДІЛ 1.

АНАЛІЗ СТАНУ ПИТАННЯ ЩОДО ФОРМУВАННЯ МАРШРУТІВ

1.1. Методи оптимізації маршрутів транспортування сільськогосподарської продукції

Процес транспортування сільськогосподарської продукції є критичним етапом у ланцюгу поставок, адже він безпосередньо впливає на збереження якості сировини, своєчасність доставки та ефективність використання ресурсів. Особливо це стосується продукції з обмеженим терміном зберігання, яка потребує швидкого збору, перевезення та переробки. У зв'язку з цим завдання побудови раціональних маршрутів доставки сільськогосподарської сировини розглядається як класична задача оптимізації, яка має широкий спектр підходів до вирішення.

Серед найбільш вживаних задач варто виділити задачу комівояжера (Travelling Salesman Problem, TSP) та задачу маршрутизації транспорту (Vehicle Routing Problem, VRP) (рис. 1.1).

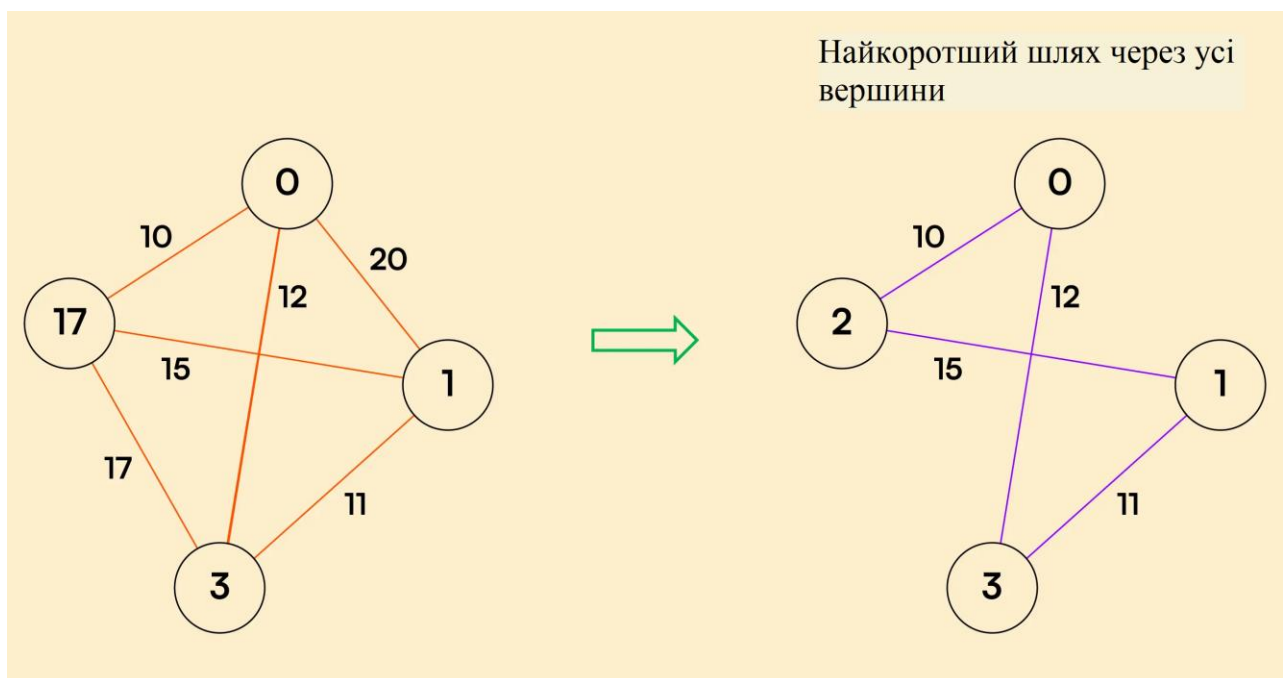


Рисунок 1.1 – Схема представлення задачі комівояжера (Travelling Salesman Problem, TSP)

Ці задачі дозволяють формалізувати питання визначення найкоротшого шляху об'їзду пунктів збору або доставки продукції з урахуванням обмежень – наприклад, вантажопідйомності транспортного засобу, часу роботи водія або обсягів продукції.

Одним із базових методів для розв'язання таких задач є лінійне та цілочисельне програмування, яке дає точні рішення, проте є малопридатним для задач великої розмірності через експоненційне зростання часу обчислень. Із прикладом математичної постановки задачі маршрутизації можна ознайомитися на рис. 1.2.

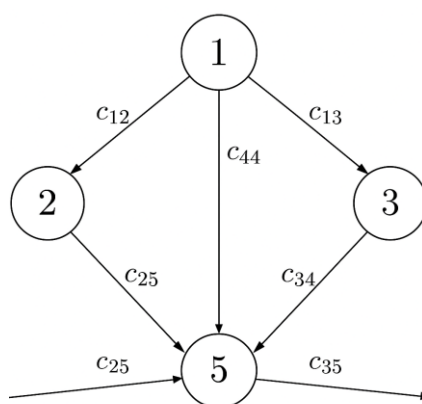


Рисунок 1.2 – Схема маршруту транспортування між пунктами збору

Рисунок 1.2 ілюструє умовну схему транспортної мережі, що використовується для перевезення сільськогосподарської продукції з різних пунктів збору до кінцевого місця призначення або проміжного логістичного центру. На схемі зображено п'ять основних точок, позначених цифрами від 1 до 5, які символізують місця зосередження продовольчої сировини – наприклад, фермерські господарства, польові ділянки або пункти прийому врожаю.

Між цими пунктами наявні стрілки, які відображають можливі маршрути руху транспорту. Кожна стрілка супроводжується позначенням вигляду c_{ij} , де i – початковий пункт маршруту, а j – кінцевий. Таке позначення дозволяє формалізувати вартість або відстань транспортування між відповідними пунктами. Наприклад, c_{12} означає вартість перевезення від пункту 1 до пункту 2, а c_{35} – від пункту 3 до пункту 5.

Варто звернути увагу, що окремі пункти мають кілька вхідних або вихідних маршрутів. Зокрема, пункт 5 отримує продукцію з кількох напрямків, що моделює ситуацію, коли декілька транспортних засобів можуть одночасно доставляти вантаж з різних місць до спільного логістичного центру або складу. Така структура характерна для сільськогосподарських регіонів з розгалуженою мережею постачальників.

Ця схема слугує основою для побудови математичної моделі маршрутизації, яка враховує як географічне положення пунктів, так і витрати на транспортування. У подальшому вона використовується для розрахунку оптимальних маршрутів за допомогою алгоритмів, зокрема мурашкових, що імітують пошук найефективнішого шляху в умовах багатьох варіантів та обмежень.

Таким чином, рисунок дозволяє наочно уявити задачу оптимізації транспортної логістики, яка є предметом дослідження в межах даної кваліфікаційної роботи. Він демонструє основні елементи, з якими працює алгоритм: пункти збору, напрямки руху, і вартісні оцінки маршрутів, що загалом відображає реальну структуру перевезень у агропромисловому середовищі.

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \xrightarrow{\min}, \quad (1.1)$$

$$\text{за умови: } \sum_{j=1}^n x_{ij} = 1 \quad \forall i, \quad \sum_{i=1}^n x_{ij} = 1 \quad \forall j, \quad (1.2)$$

$$x_{ij} \in \{0,1\}$$

де c_{ij} – вартість або довжина шляху між пунктами i та j ; x_{ij} – бінарна змінна, що дорівнює 1, якщо маршрут проходить між i та j , і 0 – інакше.

У практиці транспортування сільськогосподарської продукції найчастіше застосовують евристичні та метаетвристичні алгоритми, які дозволяють швидко знаходити рішення, близькі до оптимальних, у межах обмеженого часу. До евристичних належать алгоритми найближчого сусіда, найменшої вартості, вставок та обмінів, які формують маршрути на основі жорстких правил.

Наприклад, алгоритм найближчого сусіда будує маршрут, що послідовно додає до шляху найближчий ще не відвіданий пункт.

У свою чергу, метаввристичні методи є більш гнучкими та базуються на імітації природних чи соціальних процесів. Найпоширенішими серед них є:

- ✓ генетичні алгоритми;
- ✓ алгоритм імітації відпалу;
- ✓ алгоритм рою частинок (Particle Swarm Optimization);
- ✓ мурашкові алгоритми (Ant Colony Optimization).

Порівняльну характеристику основних методів оптимізації наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика методів оптимізації маршрутів

Метод	Тип	Переваги	Недоліки
Лінійне програмування	Точний	Гарантоване оптимальне рішення	Висока складність для великих задач
Евристика «найближчий сусід»	Евристичний	Швидкість, простота реалізації	Низька якість рішень при складних графах
Генетичний алгоритм	Метаввристичний	Ефективність при великій кількості точок	Не завжди стабільний результат
Мурашковий алгоритм	Метаввристичний	Хороше наближення до глобального оптимуму	Вимагає багато ітерацій, чутливість до параметрів
Імітація відпалу	Метаввристичний	Може виходити з локального мінімуму	Потребує точного налаштування температури

З-поміж зазначених методів саме мурашковий алгоритм показує високу ефективність у задачах, де потрібно адаптивно будувати маршрути з урахуванням кількох обмежень і критеріальних функцій. Його здатність до самонавчання через феромонний механізм дозволяє йому уникати локальних мінімумів та з часом формувати стійкі шляхи доставки, які відповідають змінним умовам навколишнього середовища.

Щодо сільського господарства, то важливо, що мурашкові алгоритми дозволяють враховувати такі фактори, як кількість доступного транспорту, обсяг вантажу з кожного поля, погодні умови та час очікування на завантаження. Це робить їх універсальним інструментом при плануванні збору сировини з багатьох розкиданих географічно ділянок у короткі терміни.

Таким чином, огляд методів оптимізації маршрутів транспортування сільськогосподарської продукції свідчить про зростаючу актуальність застосування інтелектуальних алгоритмів, зокрема мурашкових, як перспективного напрямку для підвищення ефективності логістичних систем у галузі агровиробництва.

1.2. Класифікація сучасних алгоритмів вибору маршрутів доставки вантажів та їх особливості

У логістичних системах задача вибору маршруту доставки вантажу є однією з найбільш складних і водночас фундаментальних. Вона охоплює широкий спектр обмежень – від географічної протяжності та стану інфраструктури до обсягів вантажу, графіків постачання, витрат на паливо, часових вікон тощо. У зв'язку з цим протягом останніх десятиліть розроблено велику кількість алгоритмів, які можна умовно класифікувати на кілька основних груп: детерміновані, евристичні та метаетвристичні.

Детерміновані алгоритми, зокрема методи гілок і меж, динамічного програмування та цілочисельного програмування, спрямовані на знаходження

точного розв'язку задачі за наявності повної інформації. Вони працюють ефективно при невеликій кількості вузлів, але мають суттєві обмеження в масштабованості. Наприклад, задача комівояжера з n пунктами має $(n-1)!/2$ можливих маршрутів, що робить її обчислювально складною вже при $n > 20$.

Кількість маршрутів визначається за формулою:

$$N_m = \frac{(n-1)!}{2}, \quad (1.3)$$

де n – кількість пунктів доставки.

На рисунку 1.3 зображено класифікацію алгоритмів вибору маршрутів доставки вантажів, що наочно демонструє зростання обчислювальної складності.

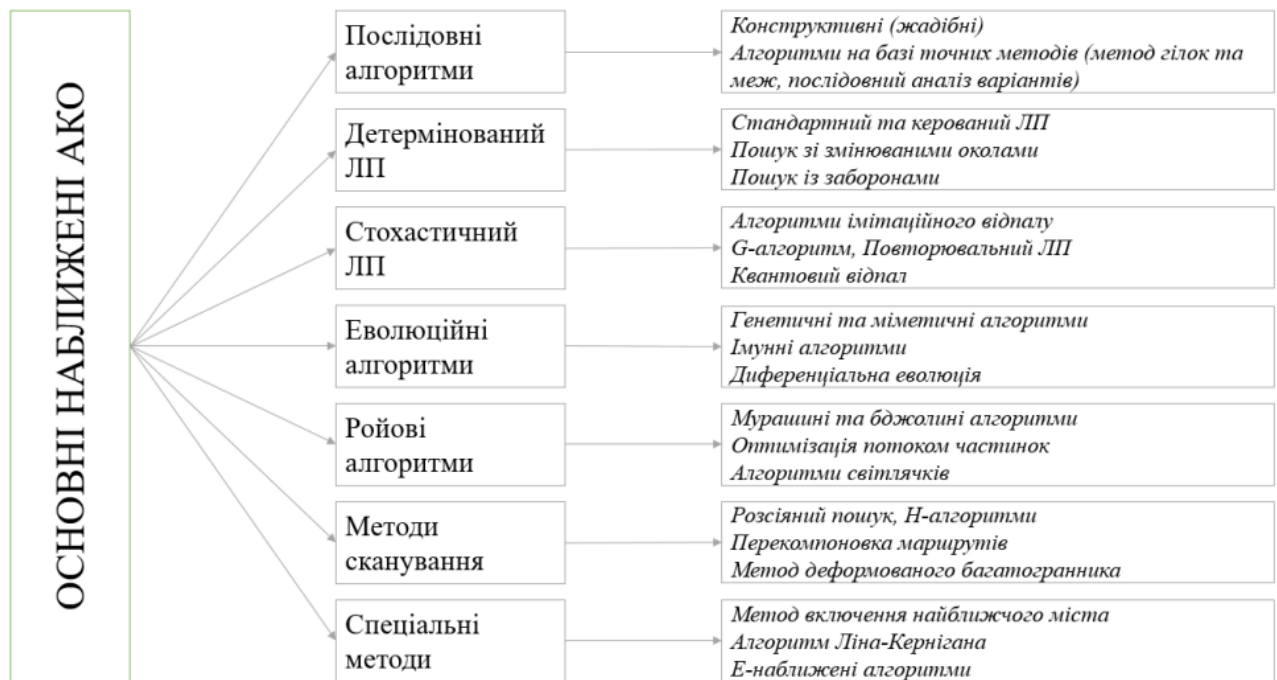


Рисунок 1.3 – Класифікація алгоритмів вибору маршрутів доставки вантажів

Евристичні алгоритми базуються на правилах наближеного пошуку, які не гарантують знаходження оптимального рішення, але дозволяють значно скоротити час обчислень. Серед них найпоширенішими є алгоритм найближчого сусіда, алгоритм кластера та вставкові методи. Вони використовуються в реальному часі для швидкого формування маршрутів,

зокрема в умовах, коли обмежений обсяг даних або рішення необхідно приймати негайно.

Метаевристичні методи, навпаки, базуються на імітації природних чи соціальних процесів. Ці алгоритми мають високу гнучкість, здатність до самонавчання і підлаштування до змін середовища. До них належать генетичні алгоритми, алгоритм імітації відпалу, алгоритм рою частинок, алгоритми колоній мурах (ACO), а також гібридні підходи. Вони дозволяють не лише знаходити високоякісні рішення, але й адаптуватися до реальних обмежень і змін у транспортній системі.

У таблиці 1.2 представлено узагальнену класифікацію алгоритмів вибору маршрутів доставки з урахуванням їх основних характеристик.

Таблиця 1.2 – Класифікація алгоритмів побудови маршрутів доставки вантажів

Тип алгоритму	Приклади	Переваги	Недоліки
Детерміновані	Гілки і межі, Цілочисельне програмування	Точність, формальна оптимальність	Немасштабованість, високі обчислювальні витрати
Евристичні	Найближчий сусід, вставки	Простота, швидкість	Можливе відхилення від оптимуму
Метаевристичні	ACO, GA, PSO, Simulated Annealing	Гнучкість, адаптивність	Потреба в налаштуванні, нестабільність результату
Гібридні	ACO + GA, GA + LP	Сильне поєднання точності й адаптації	Складність реалізації

Залежно від типу вантажу, географії перевезень, вимог до часу доставки та доступного обчислювального ресурсу вибір алгоритму може суттєво відрізнятись. Так, для великих господарств, які мають кілька віддалених точок

збору продукції, метаевристичні підходи демонструють кращу продуктивність. Натомість, у невеликих фермерських господарствах, де логістика проста, достатньо використати евристику з базовими обчисленнями вартості.

Особливістю мурашкових алгоритмів, які розглядатимуться в наступних розділах, є використання колективної пам'яті у вигляді феромонного сліду, що забезпечує ефективну маршрутизацію навіть у складних та динамічних системах. У порівнянні з іншими метаевристиками, АСО краще справляється із задачами комбінаторного характеру, де необхідно балансувати між відстанню, вартістю та іншими факторами.

Таким чином, сучасна класифікація алгоритмів маршрутизації дозволяє обрати метод, що найкраще відповідає конкретним умовам доставки сільськогосподарської продукції, враховуючи як технічні можливості, так і потреби бізнесу.

1.3. Принципи побудови мурашкових алгоритмів та їх застосування у логістиці

Мурашкові алгоритми (Ant Colony Optimization, ACO) є окремим класом метаевристичних методів, які були натхненні поведінкою реальних мурашок під час пошуку їжі. Вперше вони були запропоновані Марко Доріго у 1992 році для розв'язання задачі комівояжера, а згодом набули широкого поширення у логістиці, мережевому плануванні, управлінні транспортом і в багатьох інших прикладних галузях.

Основна ідея алгоритму полягає у симуляції поведінки колонії мурах, які, переміщаючись між вузлами графа, залишають на шляху умовну «феромонну мітку». Інші мурахи, орієнтуючись на концентрацію феромону, з більшою ймовірністю обирають маршрути, якими вже пройшли попередні особини. З часом ці феромонні сліди підсилюються для ефективних маршрутів і зникають

для неефективних, що дозволяє алгоритму поступово сходиться до оптимального або близького до оптимального розв'язку.

У загальному вигляді ймовірність того, що мураха k переміститься з пункту i до пункту j , визначається за формулою:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, & \text{якщо } j \in N_i^k, \\ 0, & \text{інакше} \end{cases}, \quad (1.4)$$

де $\tau_{ij}(t)$ – інтенсивність феромону на ребрі $i \rightarrow j$ у момент часу t ; $\eta_{ij} = 1/d_{ij}$ – евристична бажаність переходу (обернена відстань); α, β – параметри, що регулюють вплив феромону та евристики; N_i^k – множина доступних вузлів для k -ої мурахи.

З плином часу феромон оновлюється за формулою:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (1.5)$$

де $\rho \in (0,1)$ – коефіцієнт випаровування феромону; $\Delta\tau_{ij}(t)$ – загальна кількість феромону, яку мурахи залишили на ребрі ij за поточну ітерацію.

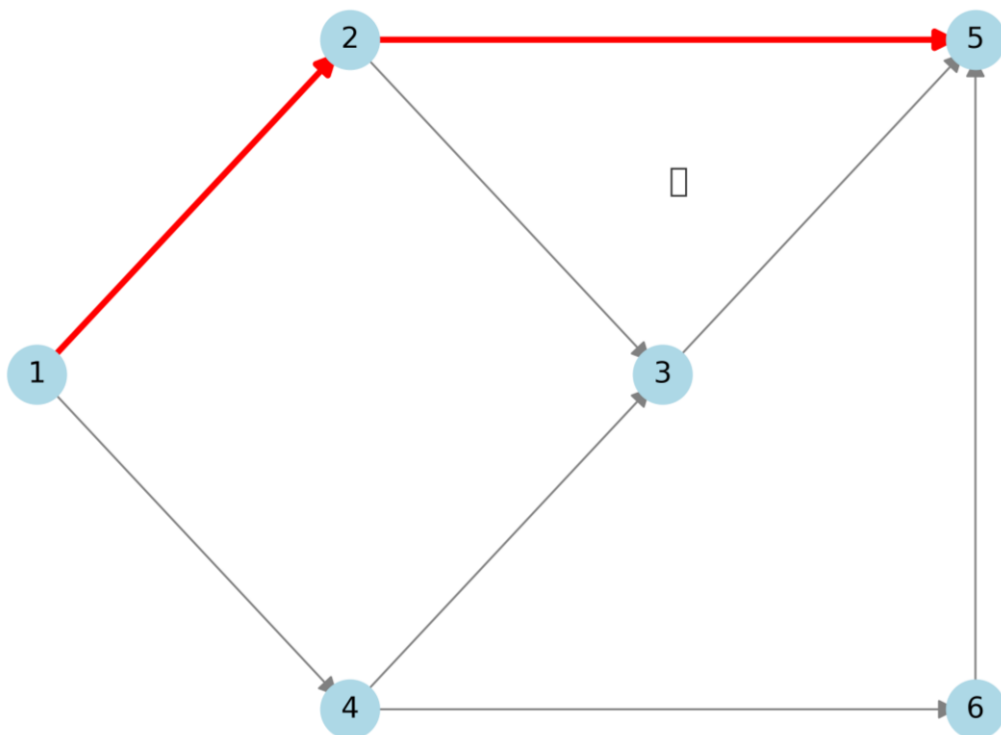


Рисунок 1.4 – Імітація пошуку оптимального маршруту мурахою у графі

На рисунку 1.4 зображено принцип побудови маршруту однією мурахою між вузлами системи. Умовні феромонні шляхи позначені стрілками різної товщини, які відповідають інтенсивності феромонного сліду.

Практичне застосування мурашкових алгоритмів у логістиці особливо ефективно у задачах, де класичні алгоритми не справляються через складність, змінні обмеження або велику кількість точок маршрутизації. В аграрному секторі такі алгоритми застосовуються для планування маршрутів заготівлі врожаю, визначення послідовності збору продукції з полів, оптимального розміщення складів і вибору транспортних шляхів.

У таблиці 1.3 наведено порівняння класичних і мурашкових підходів у контексті логістичних завдань.

Таблиця 1.3 – Порівняння класичних і мурашкових алгоритмів у логістиці

Критерій	Класичні методи	Мурашковий алгоритм
Точність	Висока при малих задачах	Висока для великих задач
Масштабованість	Низька	Висока
Адаптація до змінних умов	Обмежена	Висока
Час виконання	Високий у великих задачах	Помірний (залежить від налаштувань)
Простота реалізації	Стандартна	Середня
Необхідність тюнінгу параметрів	Низька	Висока

Таким чином, мурашкові алгоритми демонструють високу ефективність для задач, де необхідно обрати найкращу комбінацію маршрутів серед великої кількості можливих варіантів. Їх здатність адаптуватися до складних і динамічних систем, таких як аграрна логістика, робить цей підхід надзвичайно актуальним у сучасних умовах оптимізації ресурсів і часу.

1.4. Стан використання інтелектуальних методів формування маршрутів в аграрному секторі

В умовах цифровізації сільського господарства дедалі актуальнішим стає впровадження інтелектуальних методів для оптимізації логістичних процесів, зокрема планування маршрутів транспортування продовольчої продукції, врожаю, добрив, пального та інших ресурсів. У більшості аграрних підприємств України, як і в країнах Східної Європи, логістичні рішення досі здебільшого приймаються вручну або з використанням спрощених схем у таблицях Excel, що призводить до неефективного використання пального, часу й трудових ресурсів.

Інтелектуальні методи – це сукупність алгоритмічних рішень, заснованих на обчислювальному інтелекті, які включають: нейронні мережі, мурашкові алгоритми, генетичні алгоритми, метод рою частинок, логічне виведення, fuzzy-логіку та гібридні підходи. Вони дозволяють адаптивно формувати маршрути на основі великої кількості вхідних параметрів – геопросторових даних, метеоумов, стану доріг, типу техніки, обсягів вантажу, наявності пунктів приймання тощо.

На рисунку 1.5 наведено загальну структуру застосування інтелектуального алгоритму у системі маршрутизації транспорту на агропідприємстві. Вона охоплює блоки введення даних, модуль обчислень, візуалізацію маршрутів і блок контролю виконання.

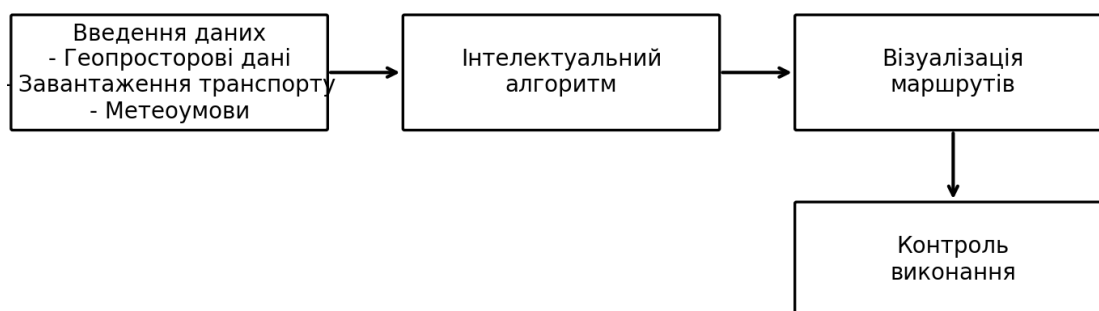


Рисунок 1.5 – Загальна структура застосування інтелектуального алгоритму для планування маршрутів в аграрній логістиці

У таблиці 1.4 представлено короткий аналіз використання різних інтелектуальних підходів у сільському господарстві, з прикладами застосування та очікуваними результатами.

Таблиця 1.4 – Аналіз застосування інтелектуальних методів для маршрутизації аграрної продукції

Метод	Сфера застосування	Приклад результату	Коментар
Мурашковий алгоритм	Збір врожаю	Зменшення витрат на паливе на 17%	Найбільш адаптивний до розкиданих полів
Генетичний алгоритм	Планування транспорту добрив	Скорочення часу маршруту на 20%	Потребує тонкого налаштування параметрів
Нейронні мережі	Прогнозування часу прибуття	Похибка менш ніж 8%	Працює ефективно з історичними даними
Fuzzy-логіка	Управління навантаженням техніки	Оптимальне розподілення ресурсів	Враховує неточні вхідні оцінки
Гібридні системи (ACO+NN)	Загальне планування збору	Баланс швидкості та точності	Високий потенціал, але складна реалізація

В аграрному секторі ЄС вже існують приклади впровадження інтелектуальних маршрутних систем у рамках концепції Smart Farming. Зокрема, у Нідерландах та Німеччині аграрні компанії активно використовують GPS-навігацію в комбінації з нейромережевим прогнозуванням потреби в транспортуванні. В Україні ці процеси перебувають на початковому етапі, проте спостерігається активний інтерес з боку агрохолдингів до таких інструментів у рамках проєктів AgTech.

Формально, задачу планування маршрутів можна записати як задачу мінімізації витрат за умов наявності ресурсних обмежень:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad \text{за умов:} \quad \sum_j x_{ij} = 1, \quad \sum_i x_{ij} = 1, \quad x_{ij} \in \{0,1\}, \quad (1.6)$$

де c_{ij} – вартість перевезення між пунктами i та j ; x_{ij} – рішення щодо наявності маршруту.

Таким чином, сучасний стан застосування інтелектуальних методів у маршрутизації в аграрному секторі можна охарактеризувати як перспективний, із високим потенціалом для покращення ефективності логістики, особливо в умовах динамічного середовища та великої територіальної розпорошеності виробничих точок.

РОЗДІЛ 2.

ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ ОБҐРУНТУВАННЯ ЕФЕКТИВНОГО МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ

2.1. Мета та завдання роботи

Планування логістичних маршрутів у сільському господарстві вимагає особливого підходу через просторову розосередженість джерел продовольчої сировини, змінність погодних умов, різну прохідність транспортних шляхів та сезонність заготівельних робіт. У цій ситуації традиційні методи маршрутизації часто виявляються неефективними або занадто ресурсоємними. Тому все більшої актуальності набуває використання інтелектуальних алгоритмів, які здатні адаптуватися до складного середовища та забезпечувати прийняття гнучких і збалансованих рішень.

Метою даної роботи є розробка та обґрунтування ефективного мурашкового алгоритму для формування маршрутів заготівлі продовольчої сировини. Передбачається, що цей алгоритм буде здатен урахувувати географічне розташування пунктів збору, кількість та обсяг вантажів, обмеження часу, стан транспортної інфраструктури, а також інші логістичні фактори. Реалізація такого підходу дозволить зменшити витрати на перевезення, покращити використання наявних ресурсів і сприяти оптимізації процесів заготівлі.

Досягнення поставленої мети передбачає розв'язання кількох важливих завдань. Насамперед необхідно дослідити сучасні методи маршрутизації, що використовуються в аграрному секторі, зокрема зосередитись на їхніх обмеженнях у контексті реальних умов сільського господарства. Далі слід розглянути принципи функціонування мурашкових алгоритмів як одного з найперспективніших напрямів обчислювального інтелекту, здатного ефективно працювати в умовах великої кількості змінних і неповноти даних.

Наступним кроком стане вибір відповідного інструментарію для реалізації алгоритму. Це стосується як мови програмування, так і бібліотек для моделювання, оптимізації та візуалізації маршрутів. Важливо не тільки правильно реалізувати сам алгоритм, а й забезпечити зручний спосіб введення початкових даних та інтерпретації отриманих результатів.

У межах цієї роботи також заплановано створити експериментальну модель, яка дозволить перевірити ефективність алгоритму на прикладі кількох типових сценаріїв: від маршруту до одного складу до багатоточкового збору з розгалуженою системою постачання. Після реалізації моделі буде проведено її тестування та порівняння результатів із результатами, отриманими за допомогою класичних методів маршрутизації, щоб виявити переваги та потенційні обмеження розробленого рішення.

Загалом, робота має на меті поєднати теоретичне обґрунтування з практичною реалізацією, сформувати базу для впровадження адаптивних логістичних стратегій у сільськогосподарських підприємствах і продемонструвати, яким чином мурашковий алгоритм може підвищити ефективність прийняття рішень у сфері управління маршрутами заготівлі.

2.2. Вибір мови програмування та програмного середовища

При реалізації оптимізаційних алгоритмів, зокрема мурашкових, особливе значення має вибір інструментів, які не лише дозволяють ефективно реалізувати логіку алгоритму, але й забезпечують зручну роботу з даними, візуалізацію результатів, інтеграцію з зовнішніми джерелами інформації та, за потреби, подальше масштабування. У рамках цієї роботи для реалізації алгоритму було обрано мову програмування Python, оскільки вона поєднує в собі потужний обчислювальний потенціал і простоту синтаксису, що особливо важливо при роботі з математичними моделями та візуалізацією.

Python має низку переваг у порівнянні з іншими мовами, що активно використовуються в задачах оптимізації, такими як C++, Java або MATLAB. Вона підтримує велике число спеціалізованих бібліотек, які значно полегшують реалізацію складних алгоритмів. Зокрема, для мурашкового алгоритму доцільним є використання бібліотек NumPy, Matplotlib, NetworkX та Tkinter (для графічного інтерфейсу).

На рисунку нижче представлено загальну логіку реалізації програмного модуля мурашкового алгоритму, з урахуванням вхідних, обчислювальних і візуалізаційних компонентів.



Рисунок 2.1 – Структура програмної реалізації мурашкового алгоритму

Програмна реалізація передбачає, що на кожному етапі ітерації виконується оновлення феромонної карти згідно з формулою:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^{(k)}(t), \quad (2.1)$$

де ρ – коефіцієнт випаровування феромону; $\Delta\tau_{ij}^{(k)}(t)$ – кількість феромону, що залишила k -та мураха на ребрі $i \rightarrow j$ під час поточної ітерації.

У процесі реалізації використовуються інтерфейси введення початкових даних (наприклад, координати пунктів збору, матриця відстаней), розрахунковий блок з ітерацією мурах, функція оцінки вартості маршруту, а також графічний блок для побудови карти маршрутів. Усі етапи реалізовано в середовищі Jupyter Notebook, що дає змогу швидко відлагоджувати код та інтерактивно перевіряти результати.

У таблиці 2.1 наведено порівняння популярних мов програмування для реалізації оптимізаційних задач для логістики, що дозволило аргументовано обрати Python.

Таблиця 2.1 – Порівняння мов програмування для задач оптимізації

Критерій	Python	C++	Java	MATLAB
Простота синтаксису	Висока	Низька	Середня	Висока
Швидкість виконання	Середня	Висока	Висока	Середня
Візуалізація	Зручна (Matplotlib, Plotly)	Вимагає окремих бібліотек	Обмежена	Вбудована
Інтерфейси користувача	Tkinter, PyQt	Qt	JavaFX	AppDesigner
Бібліотеки для оптимізації	scipy, DEAP, ACOLib	Boost, OR-tools	OptaPlanner	GlobalOptTool
Поріг входження для розробника	Низький	Високий	Середній	Високий

Як бачимо, Python є оптимальним вибором у випадках, коли акцент робиться не на високочастотному обчисленні, а на швидкому розгортанні, гнучкості та можливості візуального аналізу результатів. Саме ці переваги стали вирішальними при виборі інструменту для реалізації мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

2.3. Структура та етапи реалізації алгоритму

Розробка мурашкового алгоритму для формування маршрутів заготівлі продовольчої сировини передбачає чітке структурування процесу, який складається з окремих логічних етапів. Кожен із цих етапів виконує визначену функцію в рамках загального обчислювального циклу та дозволяє поступово зближуватися до оптимального рішення. Алгоритм є ітеративним, тобто з кожним проходом поліпшується точність і стабільність маршруту.

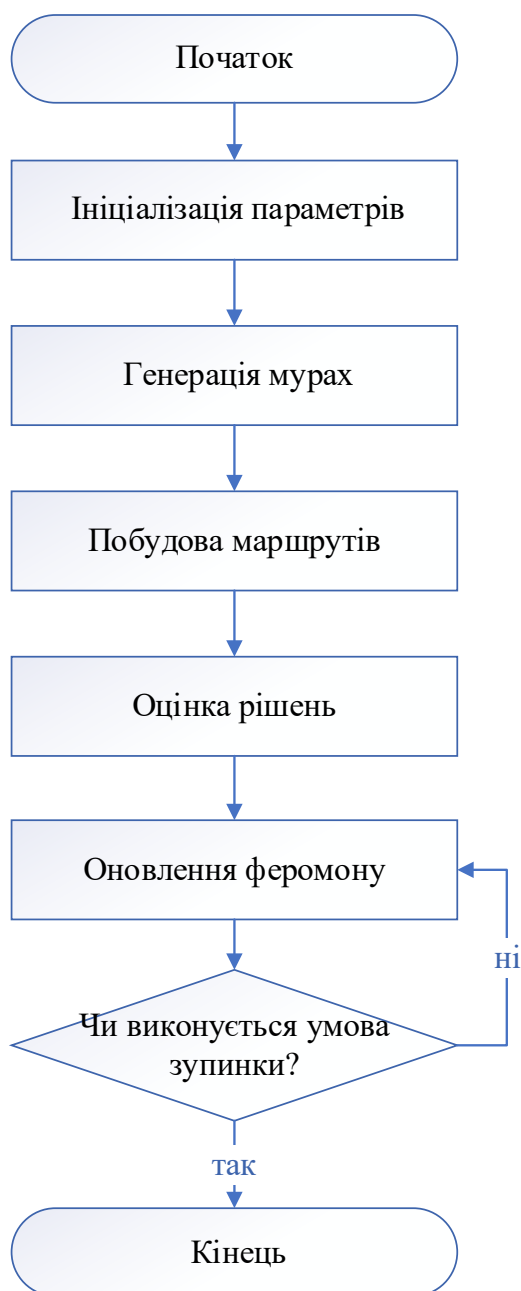


Рисунок 2.2 – Блок-схема реалізації мурашкового алгоритму маршрутизації

На рисунку 2.2 наведено блок-схему основних етапів реалізації алгоритму. Вона включає ініціалізацію параметрів, генерацію початкових рішень, вибір маршрутів, оновлення феромонної карти та перевірку критерію зупинки.

Формально, побудова маршруту базується на імовірнісному виборі наступної точки:

$$P_{ij}^{(k)}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, \quad (2.2)$$

де $\tau_{ij}(t)$ – значення феромону між пунктами i та j на ітерації; $\eta_{ij} = 1/d_{ij}$ – евристична привабливість шляху (обернено пропорційна до відстані); α, β коефіцієнти впливу феромону та евристики.

Після завершення побудови маршрутів кожною мурахою здійснюється оновлення феромонної карти за формулою (2.1).

На основі вищеописаних етапів, можна виділити такі основні блоки структури алгоритму.

Таблиця 2.2 – Етапи реалізації мурашкового алгоритму

Етап	Опис функціоналу
Ініціалізація параметрів	Задання кількості мурах, вагових коефіцієнтів, феромонної матриці
Побудова маршрутів	Ітеративний вибір вузлів на основі ймовірнісного правила
Оцінка рішень	Обчислення довжини/вартості маршруту для кожної мурахи
Оновлення феромону	Підсилення ефективних маршрутів, випаровування феромону
Умова зупинки	Кількість ітерацій або стабільність результату

На практиці алгоритм реалізовано так, щоб на кожній ітерації користувач міг бачити побудовану карту маршрутів, а також отримати числові значення

для метрики ефективності: середню довжину шляху, кращий знайдений маршрут, час виконання тощо. Це забезпечує не лише автоматичний пошук оптимального рішення, але й прозорість у процесі прийняття логістичних рішень.

2.4. Визначення параметрів мурашкового алгоритму (ACO)

Успішне функціонування мурашкового алгоритму значною мірою залежить від правильно обраних параметрів. Ці параметри визначають, як саме агенти (мурахи) досліджують простір рішень, наскільки вони покладаються на попередній досвід і як швидко адаптуються до нових умов. Налаштування значень таких параметрів, як кількість агентів, коефіцієнти впливу феромону та евристики, а також критерії зупинки, є важливим етапом, що суттєво впливає на якість остаточного рішення.

Одним із ключових параметрів є кількість агентів, тобто мурах, які одночасно шукають розв'язки в кожній ітерації. Збільшення кількості агентів дозволяє охопити ширший простір можливих маршрутів, але водночас зростає обчислювальна складність. Практично ефективним вважається співвідношення кількості агентів до кількості вузлів (пунктів збору) у межах від 0.5 до 2.

Ще два дуже важливі параметри – це коефіцієнти α та β , які визначають ступінь впливу феромонної інформації та евристики відповідно. Значення α впливає на «довіру» до сліду попередніх мурах, тоді як β посилює роль близькості або зручності переходу до певної вершини. Зазвичай рекомендується використовувати $\alpha \in [0.5; 1.5]$, а $\beta \in [2; 5]$ – така конфігурація дозволяє балансувати між досвідом та локальною логікою побудови маршруту.

Також надзвичайно важливу роль відіграє параметр випаровування феромону ρ , який контролює, як швидко алгоритм забуває про попередні рішення. Чим менше значення ρ , тим довше зберігається інформація, але й тим

більше алгоритм ризикує застрягнути в локальному оптимумі. Типовими є значення $\rho \in [0.1; 0.5]$.

Процес оновлення феромонів описується формулою (2.1).

У таблиці нижче представлено оптимальні діапазони для основних параметрів алгоритму.

Таблиця 2.3 – Основні параметри АСО та рекомендовані значення

Параметр	Позначення	Рекомендований діапазон	Коментар
Кількість мурах	m	$n - 2n$	n – кількість вузлів (пунктів)
Вплив феромону	α	0.5 – 1.5	Збільшує роль досвіду
Вплив евристики	β	2 – 5	Підсилює локальну оцінку
Випаровування	ρ	0.1 – 0.5	Визначає «пам'ять» системи
Кількість ітерацій	T	100 – 1000	Залежить від задачі та кількості вузлів

Типова реалізація в Python передбачає оголошення параметрів на початку програми у вигляді (рис. 2.3).

```
# Параметри алгоритму
num_ants = 20
num_iterations = 200
alpha = 1.0
beta = 3.0
rho = 0.2
Q = 100 # Константа для оновлення феромону
```

Рисунок 2.3 – Фрагмент коду оголошення параметрів на початку програми для створення АСО

На рисунку 2.4 представлено візуалізовану карту феромонних маршрутів після кількох ітерацій алгоритму. Товщина лінії відповідає концентрації феромону – чим товща, тим більше мурах обрали цей шлях.

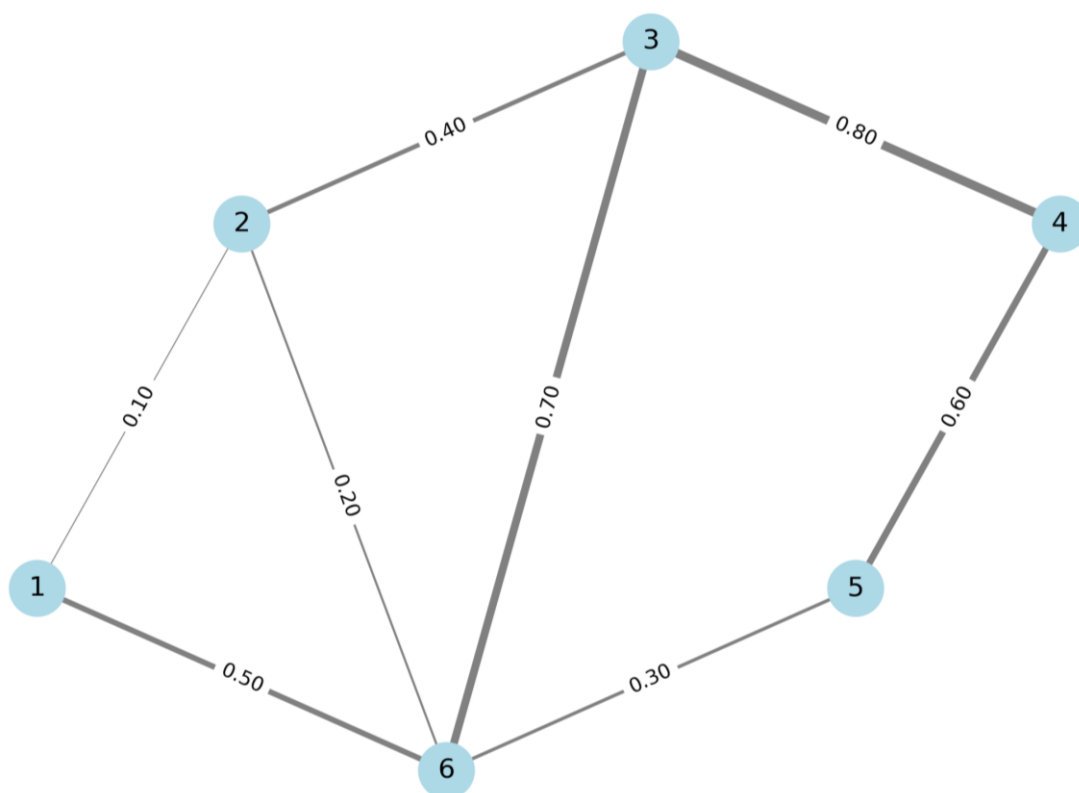


Рисунок 2.4 – Приклад розподілу феромонів між маршрутами після 50 ітерацій

Таким чином, правильне налаштування параметрів мурашкового алгоритму є не менш важливим, ніж його математична структура. Від них залежить не тільки результат, але й стабільність роботи моделі, особливо при масштабуванні до великої кількості точок збору. Саме тому добір значень параметрів у рамках цього дослідження виконувався на основі тестування та аналізу поведінки моделі в умовах, наближених до реального середовища.

2.5. Побудова цільової функції маршрутизації

Будь-який алгоритм оптимізації функціонує на основі певної цільової функції – математичного виразу, що відображає бажаний результат і формалізує критерії ефективності. У випадку маршрутизації заготівлі продовольчої сировини, такою функцією може бути мінімізація загальної

довжини маршруту, зменшення витрат пального, зниження часу перевезення чи балансування завантаження транспорту.

У цій роботі основним критерієм обрано мінімізацію сумарної довжини маршруту, що обумовлено пріоритетом скорочення логістичних витрат і часу пересування між пунктами збору. Формально цільову функцію можна подати у вигляді:

$$\min \left(L = \sum_{k=1}^m \sum_{i=1}^{n-1} d_{i_k, i_{k+1}} \cdot x_{i_k, i_{k+1}} \right), \quad (2.3)$$

де L – загальна довжина маршруту; m – кількість мурах (агентів, які будують маршрути); n – кількість пунктів збору; $d_{i,j}$ – відстань між пунктами i та j ; $x_{i,j} \in \{0,1\}$ – бінарна змінна, яка дорівнює 1, якщо маршрут включає перехід $i \rightarrow j$.

Цей підхід дозволяє оцінити ефективність кожного маршруту, сформованого мурахою, на основі відстані між пунктами. Якщо завдання включає додаткові обмеження – наприклад, часові вікна або обмеження по вантажу, – їх можна врахувати у вигляді штрафних коефіцієнтів у розширеній цільовій функції:

$$L' = L + \lambda_1 \cdot P_{time} + \lambda_2 \cdot P_{capacity}, \quad (2.4)$$

де P_{time} , $P_{capacity}$ – функції штрафу за порушення часових чи вантажних обмежень; λ_1 , λ_2 – коефіцієнти ваги обмежень.

Таблиця 2.5 – Варіанти цільових функцій для задач формування маршрутів

Тип цільової функції	Формула	Коментар
Мінімізація відстані	$\sum d_{ij} \cdot x_{ij}$	Найпоширеніший базовий підхід
Мінімізація часу	$\sum t_{ij} \cdot x_{ij}$	Потрібні дані про середню швидкість
Балансування навантаження	$\max(L_k) - \min(L_k)$	Застосовується для багатьох транспортів
Зважена функція	$Tw_1 \cdot L + w_2 \cdot Tw1$	Комбінація часу та відстані

У таблиці 2.4 наведено можливі варіанти критеріїв маршрутизації в аграрній логістиці.

Для практичної реалізації в Python, після побудови маршруту можна обчислити його довжину наступним чином, як представлено у фрагменті коду на рис. 2.5.

```
def route_length(route, distance_matrix):
    length = 0
    for i in range(len(route) - 1):
        length += distance_matrix[route[i]][route[i+1]]
    return length
```

Рисунок 2.5 – Фрагмент коду для визначення довжини маршруту

Результати обчислення довжини використовуються в АСО для визначення, наскільки ефективним був маршрут кожної мурахи в поточній ітерації. Від цього значення також залежить обсяг феромону, який буде нанесений на відповідні ребра графа.

На рисунку 2.6 зображено приклад сформованого маршруту для одного агента (мурахи), де підписані відстані між пунктами.

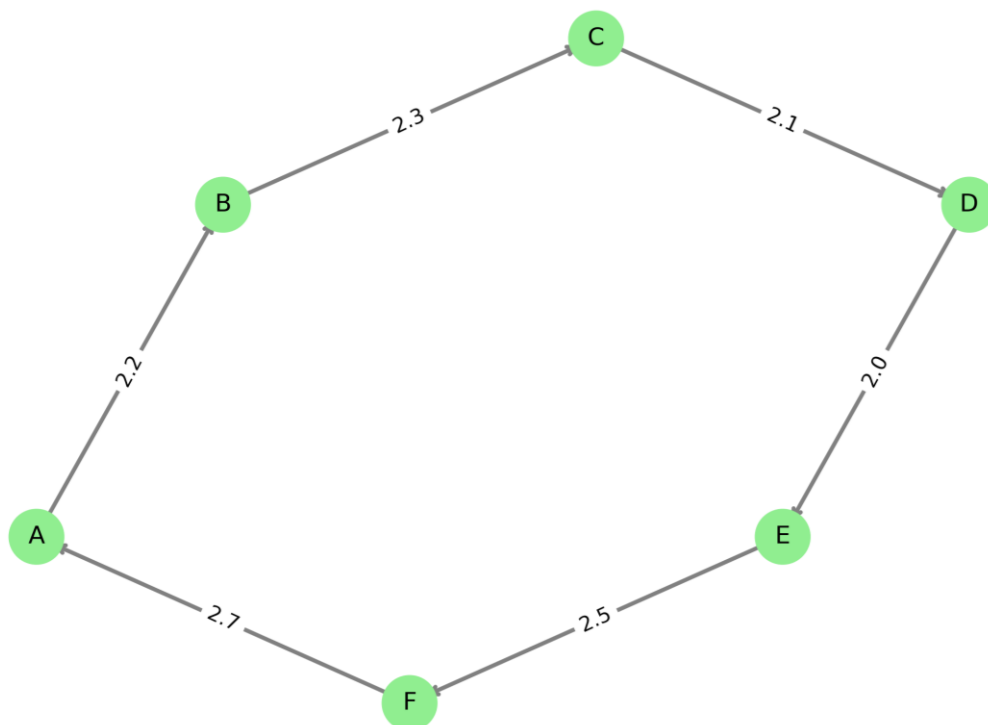


Рисунок 2.6 – Побудований маршрут агента з розрахованою довжиною

Таким чином, правильно побудована цільова функція дозволяє не лише формалізувати мету моделі, але й виступає основою для навчання, адаптації та

порівняння ефективності маршрутів. У разі розширення задачі – наприклад, при додаванні складів, часових обмежень чи змінних витрат – ця функція може бути модифікована без необхідності переписування основного алгоритму.

2.6. Засоби візуалізації результатів та аналітичного контролю

Для оцінки ефективності роботи мурашкового алгоритму важливо не лише виконати математичне обчислення цільової функції, а й надати зручний візуальний та аналітичний інтерфейс, який дозволить дослідити хід моделювання, зіставити варіанти маршрутів та прийняти обґрунтоване рішення на основі результатів. У цьому розділі розглядаються засоби візуалізації, які були використані для контролю процесу побудови маршрутів, оцінки ефективності алгоритму та аналізу зміни показників у часі.

У рамках реалізації алгоритму в середовищі Python як основні інструменти візуалізації було обрано бібліотеки Matplotlib та NetworkX. Перша дозволяє будувати графіки, гістограми, тренди та зображення феромонної динаміки, друга – працювати з графовими структурами, що ідеально відповідає природі маршрутизації.

Важливим аналітичним інструментом також є таблиці результатів, у яких порівнюється ефективність маршрутів за різними конфігураціями параметрів. Окрім графіків та таблиць, до аналітичного контролю також відноситься зберігання даних про кожну ітерацію (довжина маршруту, феромони, послідовність переходів) у CSV-файл або базу даних. Це дозволяє провести додатковий аналіз у зовнішніх інструментах, наприклад, Excel або Power BI.

Отже, ефективне візуальне подання результатів не лише покращує сприйняття моделі, а й суттєво підвищує якість прийняття управлінських рішень. Саме тому у цій роботі було закладено гнучку систему виводу проміжних і фінальних результатів з можливістю адаптації під різні сценарії логістичного моделювання.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ ОБҐРУНТУВАННЯ ЕФЕКТИВНОГО МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ

3.1. Особливості визначення виробничих умов заготівлі продовольчої сировини

Для формування вхідних даних у моделі маршрутизації між господарствами – виробниками продовольчої сировини – було використано дані з відкритих картографічних сервісів, таких як OpenStreetMap [30], Google Maps [27] та Bing Maps [1]. На основі цієї інформації створюється матриця відстаней між відповідними пунктами, а також матриця середніх швидкостей руху транспорту на ділянках транспортної мережі певної сільської громади (позначеної як t).

У разі виявлення пошкоджених відрізків дороги, які унеможливають проїзд, відповідні ділянки виключаються з транспортного графа. Такий підхід дозволяє адекватно реагувати на реальний стан дорожньої інфраструктури та уникнути побудови маршрутів через непридатні для руху шляхи.

Якщо ж на певних сегментах (позначених як l) присутні лише часткові ушкодження дорожнього полотна або спостерігаються затори, тоді для кожного з типів транспортних засобів передбачається зменшення середньої швидкості руху. Зміна швидкості відображається як функція стану ділянки та типу транспорту, що дозволяє моделі залишатись динамічною й адаптивною до умов.

Таким чином, середня швидкість руху між j -ми населеними пунктами розглядається як змінна величина, що залежить від характеристик конкретної ділянки дороги. Це забезпечує врахування виробничо-логістичних обмежень, притаманних кожній окремій громаді.

3.2. Результати визначення виробничих умов заготівлі продовольчої сировини

У якості ілюстрації розглянуто фрагмент транспортної мережі Заболотцівської територіальної громади, що знаходиться у Золочівському районі Львівської області. Зокрема, ділянку дороги між селами Заболотці та Велин, де часткові ушкодження покриття призводять до суттєвого зниження швидкості руху. Така ситуація чітко демонструє, як стан дорожньої інфраструктури впливає на побудову маршрутів (рис. 3.1).

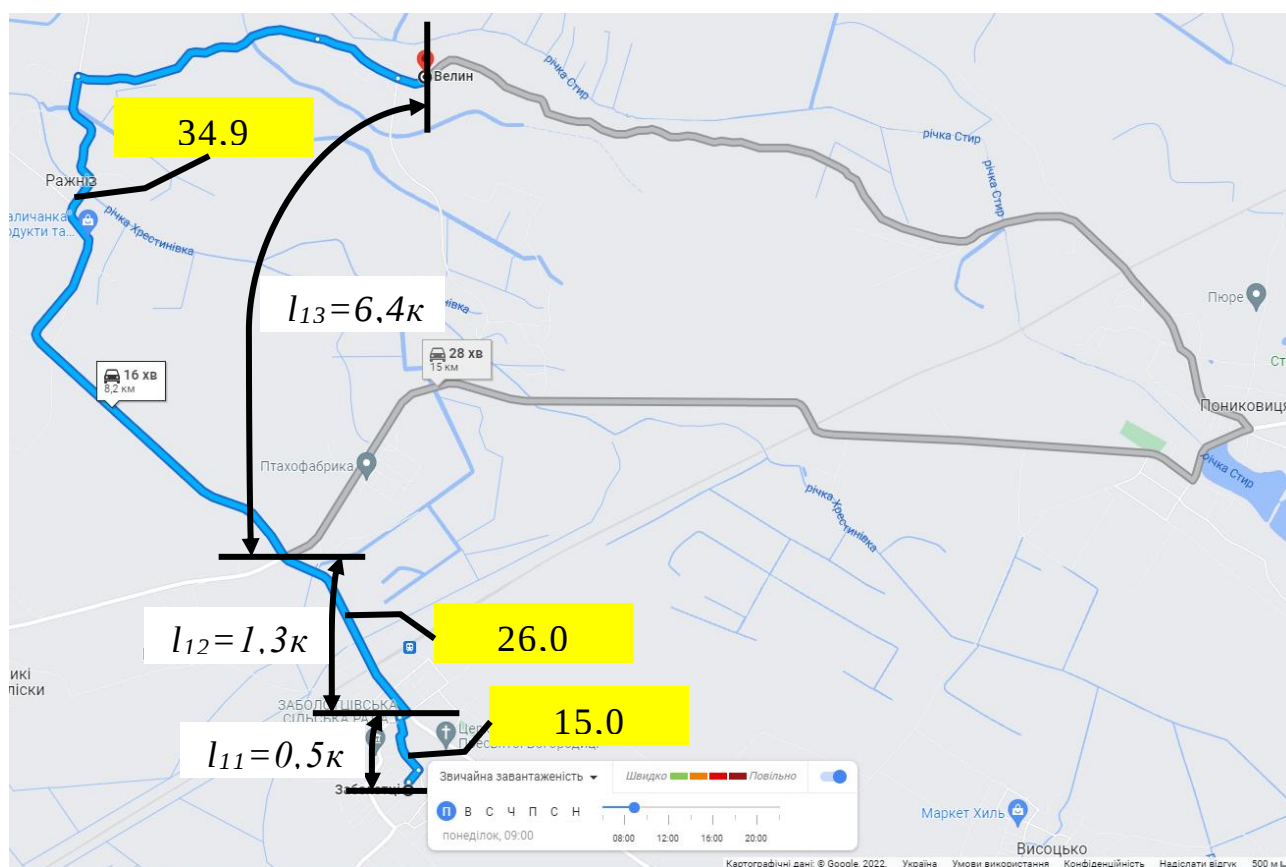


Рисунок 3.1 – Оцінювання середньої швидкості руху транспортного засобу на окремому відрізку транспортної інфраструктури Заболотцівської територіальної громади між селами Заболотці та Велин (Золочівський район, Львівська область).

У цьому прикладі маємо наступне:

- відстань між окремими населеними пунктами:

$$l_{ij} = l_{11} + l_{12} + l_{13} = 0,5 + 1,3 + 6,4 = 8,2 \text{ км};$$

➤ середню тривалість руху автомобіля між окремими населеними пунктами:

$$t_{ij} = t_{11} + t_{12} + t_{13} = 2 + 3 + 11 = 16 \text{ хв} \approx 0,26 \text{ год};$$

➤ середню швидкість руху автомобіля між окремими населеними пунктами:

$$V_{ij}^c = \frac{l_{ij}}{t_{ij}} = \frac{8,2}{0,26} = 30,8 \text{ км / год}.$$

3.2. Опис правил поведінки мурах в удосконаленому мурашковому алгоритмі

У межах цієї роботи запропоновано вдосконалення класичного алгоритму оптимізації мурашиних колоній (АСО) для врахування реальних виробничих умов, зокрема в умовах надзвичайних ситуацій. Основні зміни стосуються поведінкових механізмів агентів (мурах), які імітують рух транспортних засобів на пошкодженій або обмежено доступній транспортній мережі сільської громади.

Передусім, кожна мураха веде індивідуальний журнал відвіданих пунктів збору (вузлів графа), причому дозволяється відвідати кожен вузол лише один раз. Таким чином, формується список заборонених для повторного візиту пунктів. Крім того, кожен транспортний засіб (мураха) стартує з певного початкового вузла (господарства) і має доступ до підмножини сумісних із ним пунктів збору.

Рух мурах визначається не лише феромонною привабливістю, а й видимістю вузлів. Видимість визначається як обернена величина до відстані між вузлами:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad , \quad (3.1)$$

де d_{ij} – відстань між пунктами i та j .

Мурахи обирають наступний вузол із урахуванням рівня феромонів на ребрі та видимості, а також додатково – стану дорожнього покриття. Класичне правило переходу:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad , \quad (3.2)$$

змінюється на

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta \cdot \delta_{ij}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta \cdot \delta_{il}} \quad , \quad (3.3)$$

де δ_{ij} – коефіцієнт, що відображає стан дорожнього покриття між вузлами.

Значення δ_{ij} приймає:

$\delta_{ij} = 0$ – рух неможливий через пошкодження дороги;

$\delta_{ij} = 1$ – нормальний стан дороги;

$\delta_{ij} \in (0;1)$ – часткове ушкодження або затори.

У випадку часткового обмеження руху показник δ_{ij} визначається за формулою:

$$\delta_{ij} = \frac{v_{ij}^{сер}}{v_{ij}^{факт}} \quad , \quad (3.4)$$

де $v_{ij}^{сер}$ – середня швидкість на ділянці, км/год; $v_{ij}^{факт}$ – фактична прогнозована швидкість з урахуванням перешкод, км/год.

Феромонна карта оновлюється за допомогою механізму випаровування та підсилення найкращих маршрутів. Формула оновлення феромону:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad , \quad (3.5)$$

де ρ – інтенсивність випаровування феромону; $\Delta\tau_{ij}(t)$ – обсяг феромону, залишеного мурахами на ребрі.

Феромон, що залишає одна мураха:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L_k} \quad , \quad (3.6)$$

Загальна кількість феромону:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad , \quad (3.7)$$

де m – кількість мурах у колонії.

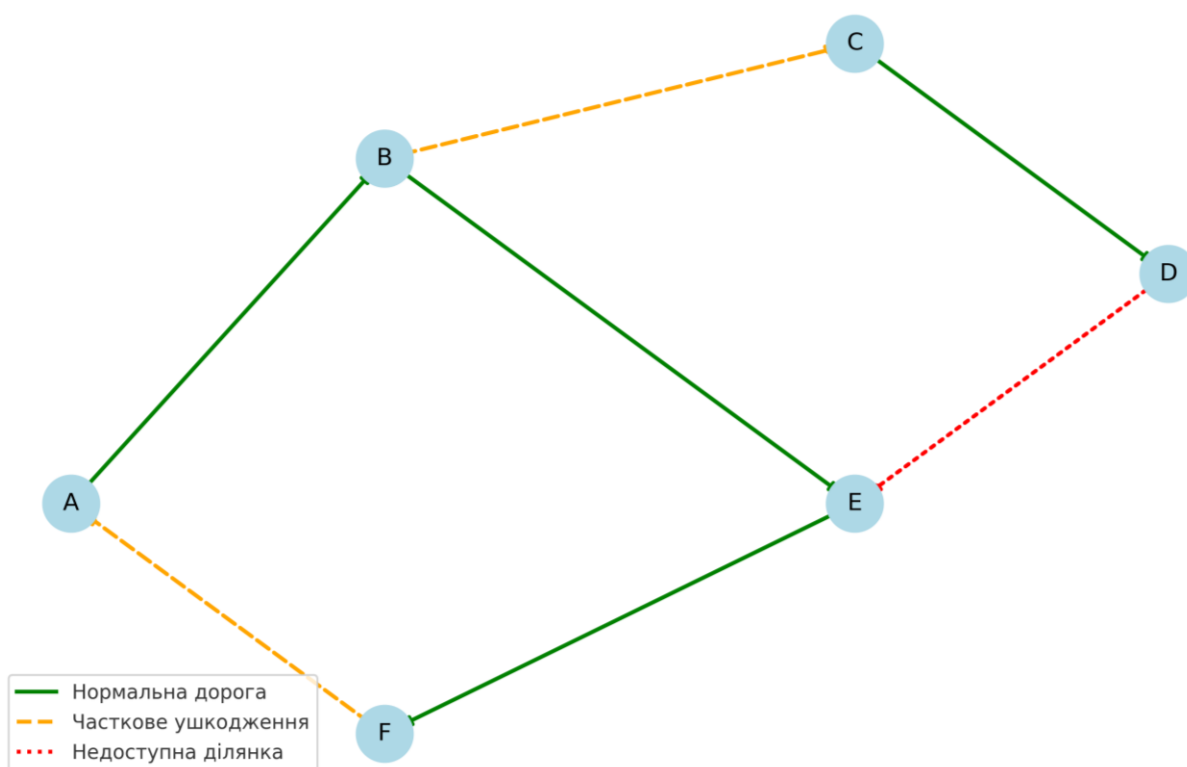


Рисунок 3.2 – Схема переходів мурах із урахуванням стану дорожнього покриття

Таким чином, адаптований АСО-алгоритм враховує стан дорожнього покриття та динамічні обмеження, що виникають у надзвичайних ситуаціях. Він дозволяє здійснювати маршрутну оптимізацію із урахуванням швидкості, перешкод і допустимих ділянок мережі. У випадках, коли деякі маршрути стають недоступними, виконується автоматичне перепланування без потреби повного перерахунку мережі.

Додатково, за умови перевищення вантажопідйомності транспортного засобу над обсягом сировини, що підлягає збору, маршрути автоматично розділяються та закріплюються за різними транспортними засобами, що забезпечує гнучкість та відповідність реальним логістичним обмеженням.

3.4. Блок-схема удосконаленого мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини

На основі раніше викладених положень було побудовано блок-схему вдосконаленого алгоритму формування маршрутів заготівлі продовольчої сировини в межах сільської громади, з урахуванням виробничих обмежень, які виникають у надзвичайних ситуаціях. Схему представлено на рисунку 3.10. Вона наочно ілюструє основні етапи роботи алгоритму, починаючи від введення вихідних даних і завершуючи визначенням оптимального маршруту з відповідними показниками ефективності.

У процесі використання вдосконаленого мурашкового алгоритму приймаються такі початкові припущення:

- кількість вершин графа (господарств-виробників сировини) дорівнює кількості мурах, які починають рух із різних точок мережі;
- феромонна інтенсивність на всіх ребрах графа до початку симуляції є однаковою;
- для першої ітерації кожна мураха вибирає найближчий доступний вузол, після чого подальші переходи визначаються за удосконаленим правилом (3.3).

Вдосконалений алгоритм передбачає такі етапи роботи:

1. Ініціалізація – введення вихідних даних у вигляді матриці показників, що включає відстані, час у дорозі, вартість перевезення та інші релевантні логістичні характеристики;

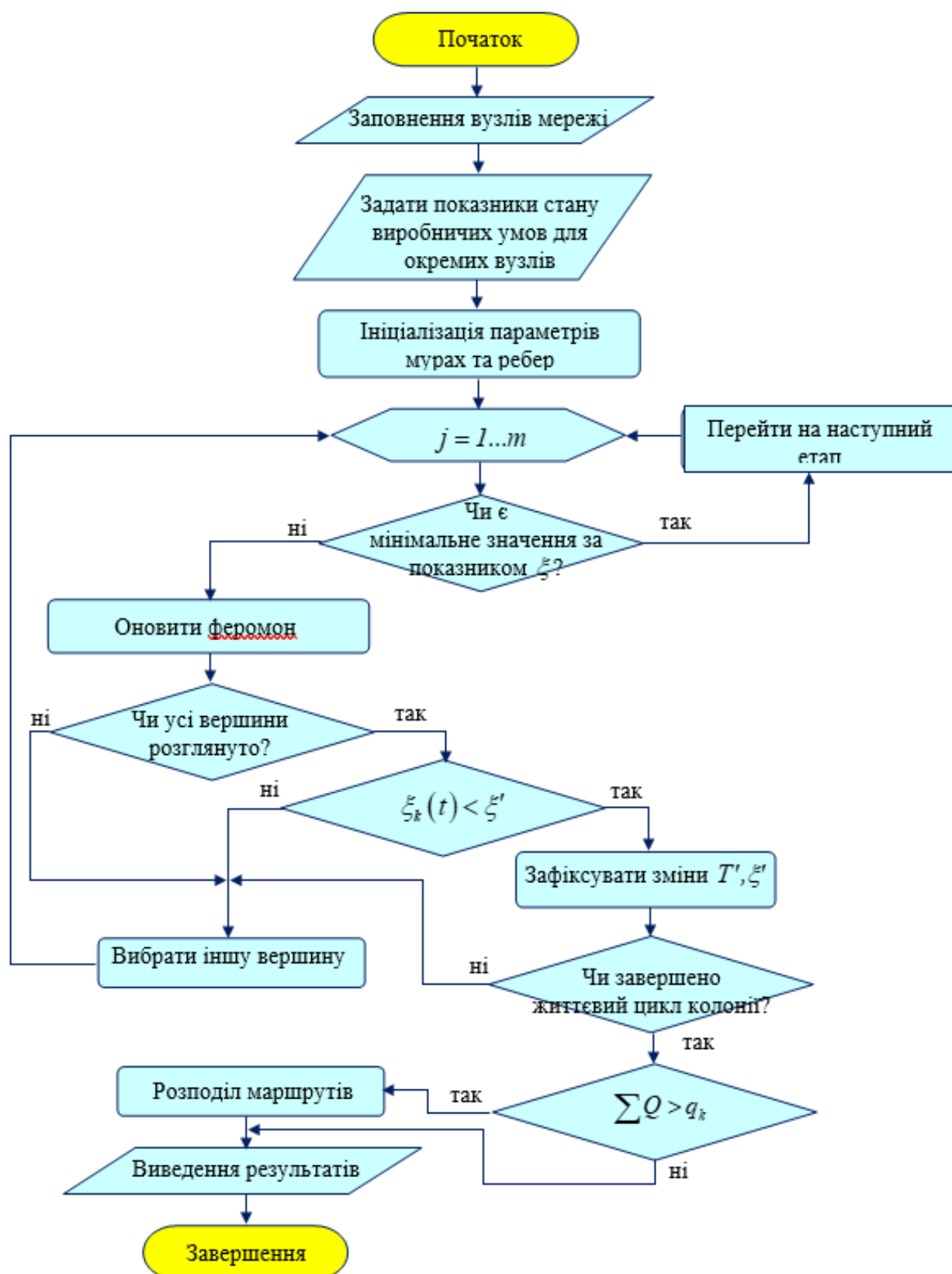


Рисунок 3.3 – Блок-схема удосконаленого мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини: ξ – показник вибору раціональних варіантів (відстань, час, вартість тощо); T', ξ' – поточний маршрут та його показник; m – кількість мурах (агентів) у колонії

2. Аналіз стану дорожньої мережі – внесення до моделі інформації про стан виробничих умов (пошкодження, затори, обмеження швидкості) для кожного окремого вузла транспортної мережі;
3. Параметризація моделі – встановлення початкових параметрів алгоритму: кількість агентів, значення α, β, ρ , інтенсивність феромону на ребрах;
4. Розміщення мурах – випадковий розподіл агентів по доступних вершинах без дублювань;
5. Цикл ітерацій – запуск ітераційного процесу пошуку оптимального маршруту, де для кожної мурахи враховується стан дорожнього полотна. У разі виявлення змін (наприклад, ускладнення руху), перебудовується лише та частина маршруту, яка ще не була пройдена;
6. Фіксація результату – збереження поточного маршруту та обчислення його ключових показників (відстань, час, вартість);
7. Перевірка умов перевантаження – якщо загальний обсяг сировини у господарствах перевищує вантажопідйомність транспортного засобу, здійснюється автоматичний розподіл маршрутів між декількома транспортними засобами;
8. Формування результатів – виведення оптимізованих маршрутів і супутніх логістичних показників для подальшого аналізу або практичного використання.

РОЗДІЛ 4.

ПРАКТИЧНЕ ВИКОРИСТАННЯ МУРАШКОВОГО АЛГОРИТМУ ФОРМУВАННЯ МАРШРУТІВ ЗАГОТІВЛІ ПРОДОВОЛЬЧОЇ СИРОВИНИ

4.1. Програмна реалізація вдосконаленого алгоритму АСО для формування маршрутів заготівлі продовольчої сировини

Для практичної перевірки ефективності вдосконаленого алгоритму оптимізації мурашиних колоній було створено програмну модель, реалізовану мовою програмування Python (додаток А). Програмна реалізація враховує як класичні принципи АСО, так і доповнення, пов'язані з відображенням реального стану транспортної мережі, обмеженнями проїзду через ушкоджені ділянки доріг, а також необхідністю перерозподілу маршрутів у разі виникнення надзвичайних ситуацій.

Структура програмного коду охоплює кілька взаємопов'язаних модулів, що відповідають за ініціалізацію параметрів, побудову вхідних матриць, вибір маршрутів мурахами, обчислення довжини маршрутів, оновлення феромонної карти та збереження найкращого знайденого маршруту.

На початковому етапі задаються основні параметри, зокрема кількість вузлів (господарств), кількість мурах, кількість ітерацій, а також коефіцієнти, які визначають вплив феромонної інформації та евристичної привабливості. Окремо створюється матриця відстаней між пунктами збору сировини та матриця стану дорожнього полотна. У матриці відстаней значення «inf» означають, що між двома пунктами відсутнє пряме з'єднання або дорога непридатна для руху.

```

n_nodes = 6
alpha = 1.0
beta = 3.0
rho = 0.2
Q = 100
n_ants = n_nodes
n_iterations = 50

```

Рисунок 4.1 – Фрагмент коду із початковими даними для використання алгоритму АСО для формування маршрутів заготівлі продовольчої сировини

Після задання параметрів у пам'яті формується матриця відстаней та матриця, яка відображає стан кожного дорожнього відрізка. Наприклад, значення 1 означає нормальний стан дороги, 0.5 – частково пошкоджене покриття, 0 – повна непрохідність. Це дозволяє алгоритму автоматично враховувати ці обмеження при побудові маршрутів.

Далі виконується головний ітераційний цикл. Для кожної мурахи ініціалізується список відвіданих пунктів, і вона починає рух із випадково обраного стартового вузла. Під час побудови маршруту перевіряється можливість переходу до кожного з вузлів – якщо вузол ще не відвіданий і дорога до нього доступна, обчислюється ймовірність переходу. Обчислені значення нормалізуються, після чого агент із певною ймовірністю обирає наступний пункт маршруту (рис. 4.2).

```

while len(visited) < n_nodes:
    probabilities = []
    for j in range(n_nodes):
        if j not in visited and condition_matrix[current][j] > 0 and distance_matrix[current][j] != np.inf:
            eta = 1 / distance_matrix[current][j]
            tau = pheromone[current][j]
            delta = condition_matrix[current][j]
            p = (tau ** alpha) * (eta ** beta) * delta
            probabilities.append((j, p))

```

Рисунок 4.2 – Фрагмент коду із алгоритмом вибору наступного вузла мурахи з урахуванням феромонної привабливості, евристичної видимості та стану дорожнього покриття

Після побудови маршруту мураха накопичує його довжину, а найкоротший знайдений маршрут зберігається як найкращий. Далі оновлюється феромонна карта. На всіх ребрах значення феромону частково випаровується, а на тих, якими пройшли мурахи, додається нове значення феромону пропорційно до якості маршруту (рис. 4.3).

```
pheromone = (1 - rho) * pheromone
for i in range(len(path) - 1):
    a = path[i]
    b = path[i + 1]
    pheromone[a][b] += Q / length
    pheromone[b][a] += Q / length # симетрія
```

Рисунок 4.3 – Фрагмент коду із оновленням феромонної карти з урахуванням випаровування та підсилення пройдених маршрутів

Таким чином, алгоритм імітує природний процес самонавчання колонії, де ефективні маршрути підсилюються, а неефективні з часом втрачають привабливість. Підсумковий маршрут, що має найменшу сумарну довжину, виводиться як результат роботи моделі.

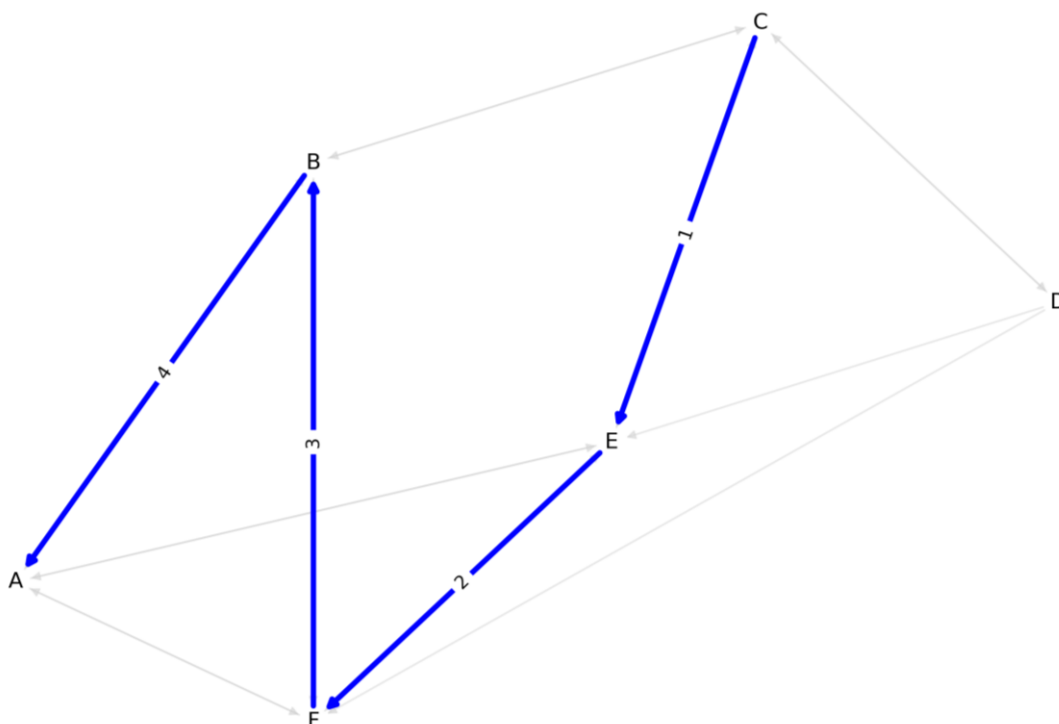


Рисунок 4.4 – Візуалізація найкращого маршруту визначеного за удосконаленням алгоритмом АСО після 50 ітерацій

У процесі експериментів підтверджено, що навіть за наявності обмежень у дорожній мережі алгоритм здатен адаптивно будувати маршрути, що враховують поточні умови (рис. 4.4).

У таблиці 4.1 подано приклад знайденого маршруту після 50 ітерацій та відповідну довжину шляху.

Таблиця 4.1 – Результати визначення найкращого маршруту за удосконаленням алгоритмом АСО

Найкращий маршрут	Сумарна довжина (км)
[2, 4, 5, 1, 0]	12.0

Отже, створена програмна модель дозволяє ефективно застосовувати вдосконалений алгоритм мурашиної оптимізації для побудови маршрутів заготівлі продовольчої сировини в умовах, наближених до реальних. Врахування динамічних виробничих обмежень та пошкоджень транспортної мережі робить цей підхід перспективним для використання в логістиці аграрного сектору в післявоєнний період.

4.2. Результати порівняння вдосконаленого алгоритму АСО із існуючими для задач формування маршрутів заготівлі продовольчої сировини

Для об'єктивної оцінки ефективності розробленого вдосконаленого алгоритму формування маршрутів було проведено порівняльне дослідження з класичним алгоритмом оптимізації мурашиних колоній (АСО). Аналіз виконували на основі низки тестових задач, що моделюють процеси збору продовольчої сировини на території громад, у тому числі з урахуванням обмежень, викликаних надзвичайними ситуаціями.

Порівняння здійснювалося за кількома ключовими критеріями: довжина знайденого маршруту, час обчислення та відхилення від найкращого відомого результату. Зведені результати наведено у таблиці 4.2.

Таблиця 4.2 – Результати порівняння вдосконаленого алгоритму АСО із існуючими для задач формування маршрутів заготівлі продовольчої сировини

Задача	К-сть вершин	Найкраще відоме (км)	Класичний АСО (км, сек)	Удосконалений АСО (км, сек)	Відхилення по довжині (%)	Економія часу (%)
Els 19	19	532	532 / 0.37	532 / 0.37	0.00	0
R208.25	25	332	332 / 0.38	332 / 0.38	0.00	0
Bur 26a	26	280	280 / 0.41	280 / 0.40	0.00	2
Oliver 30	30	423	423 / 0.42	422 / 0.41	0.24	2
Eilon 50	50	428	428 / 1.52	427 / 1.43	0.23	6
Eil 51	51	429	429 / 1.72	427 / 1.52	0.47	12
Berlin 52	52	755	755 / 2.13	755 / 1.85	0.00	13
St 70	70	681	681 / 3.95	680 / 3.38	0.15	14
Eilon 75	75	548	548 / 4.54	543 / 3.89	0.91	14
Eil 76	76	551	551 / 5.15	546 / 4.38	0.91	15
KroA 100	100	21452	21452 / 10.65	21392 / 9.08	0.28	15

Аналіз результатів свідчить про те, що запропонований алгоритм демонструє стабільну якість побудови маршрутів навіть за умов ускладненої дорожньої мережі. Відхилення у довжині маршрутів у порівнянні з класичним підходом не перевищує 1% у всіх тестах, а в деяких задачах (наприклад, Els 19, R208.25) взагалі не фіксується.

Особливо помітним є скорочення часу обчислення при зростанні кількості вершин. Якщо для задач з кількістю вузлів до 50 одиниць економія становить у середньому до 6%, то при переході до більш складних задач із 51 до 100 вершин – виграш за часом зростає до 12–15%.

На рисунку 4.5 представлено графік залежності часу визначення маршруту від кількості вершин для обох алгоритмів, що наочно демонструє ефективність удосконаленої моделі.

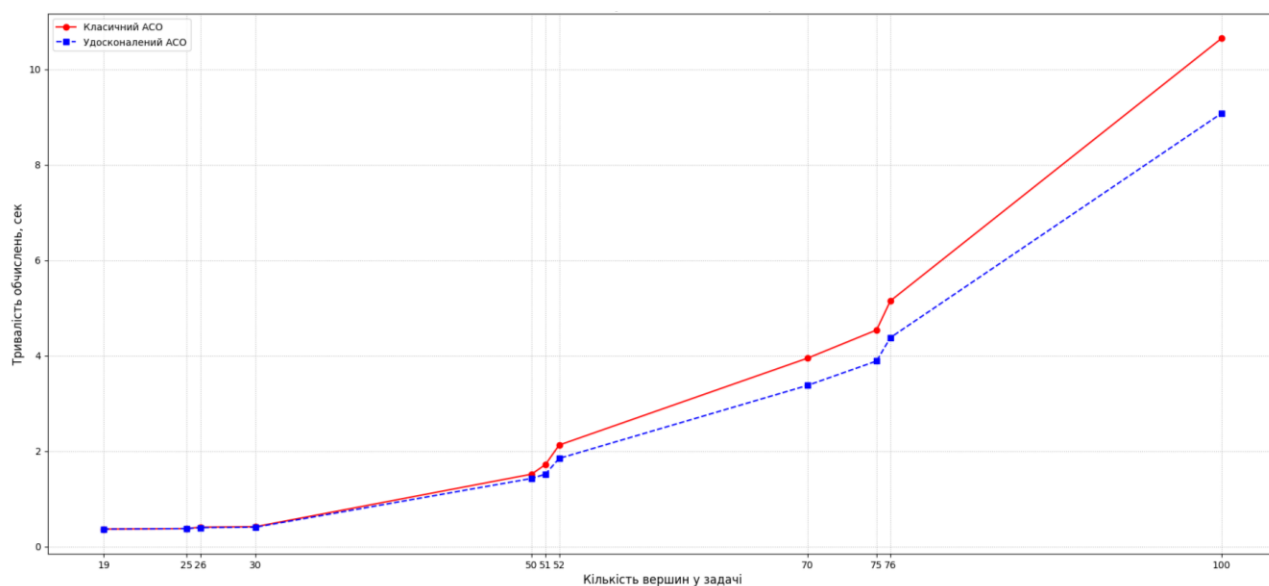


Рисунок 4.5 – Залежність тривалості обчислень від кількості вершин для класичного та удосконаленого алгоритмів АСО

Таким чином, результати експериментального порівняння підтверджують доцільність використання вдосконаленого мурашкового алгоритму для вирішення задач оптимального формування маршрутів заготівлі продовольчої сировини. Удосконалений підхід забезпечує не лише збереження точності, але й пришвидшує процес прийняття управлінських рішень, що є критично важливим у контексті надзвичайних ситуацій.

Запропоновану модель доцільно впроваджувати для планування оперативних маршрутів у громадах, де обмеження руху й стан дорожнього полотна суттєво впливають на логістику, зокрема під час воєнних дій, стихійних лих або кризових періодів в аграрному секторі.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

5.1. Аналіз стану робочого місця та умов праці

У рамках виконання кваліфікаційної роботи з теми впровадження удосконаленого мурашкового алгоритму для формування маршрутів заготівлі продовольчої сировини було передбачено роботу за комп'ютером у середовищі програмування Python. З огляду на це, особливу увагу слід приділити аналізу умов праці оператора ПК, який здійснює як розробку алгоритмів, так і тестування їх у симуляційних середовищах.

Робоче місце було організовано в окремому приміщенні з природним освітленням та можливістю регулювання штучного освітлення відповідно до санітарно-гігієнічних норм. Робоча зона відповідала стандартним вимогам до ергономіки: монітор розміщено на відстані 60–70 см від очей користувача, центр екрана знаходився на рівні очей, а робоче крісло було оснащено регулюванням по висоті та підтримкою поперекового відділу хребта.

З точки зору мікроклімату, у приміщенні підтримувався стабільний температурний режим у межах 21–24°C, відносна вологість повітря становила 45–55%, а рівень шуму не перевищував 45 дБ, що відповідає нормативним вимогам до умов офісної праці. Було забезпечено регулярне провітрювання та використання зволожувачів повітря в опалювальний період.

Для оцінки ризиків, пов'язаних з експлуатацією персонального комп'ютера, у таблиці 5.1 узагальнено основні шкідливі й небезпечні чинники, які виникають на робочому місці оператора ПК.

Таблиця 5.1 – Основні шкідливі й небезпечні чинники, які виникають на
робочому місці оператора ПК

№	Потенційно шкідливий фактор	Характер впливу	Заходи профілактики
1	Напруження зору	Зорове перевантаження	Регулярні перерви, гімнастика для очей
2	Статичне навантаження на хребет	Порушення постави, біль у спині	Ергономічне крісло, зміна положення тіла
3	Недостатнє освітлення або блиск	Напруження зору, головний біль	Регулювання яскравості, освітлення
4	Електромагнітне випромінювання	Тривалий вплив (умовний ризик)	Вибір якісної техніки, екранні фільтри
5	Тривала робота без перерв	М'язове напруження, зниження уваги	Організація регламентованих перерв

Загалом, умови праці в процесі виконання дослідження відповідають сучасним вимогам безпеки. Оскільки робота пов'язана переважно з інтелектуальною діяльністю, найбільше навантаження припадає на зоровий апарат та опорно-рухову систему. Для мінімізації професійних ризиків були дотримані рекомендації щодо тривалості безперервної роботи за комп'ютером, а також реалізовано заходи профілактики (чергування видів діяльності, фізичні вправи, контроль освітлення).

За результатами аналізу можна зробити висновок, що робоче місце відповідає вимогам нормативних документів з охорони праці (ДСанПіН 3.3.2.007-98, ДСТУ EN ISO 9241) і не створює критичних ризиків для здоров'я користувача при дотриманні регламентів безпеки праці.

5.2. Небезпечні та шкідливі чинники

Під час розробки алгоритмічного забезпечення та тестуванням моделей у цифровому середовищі, необхідно враховувати дію низки потенційно шкідливих та небезпечних виробничих чинників, що характерні для роботи за комп'ютером упродовж тривалого часу.

До основних шкідливих чинників, що мають вплив на користувача комп'ютерної техніки, належать: недостатнє або нерівномірне освітлення, тривале статичне положення тіла, вплив електромагнітного поля, підвищене зорове навантаження, а також психоемоційна втома, що виникає внаслідок інтенсивної розумової діяльності.

На рисунку 5.1 представлено узагальнену класифікацію небезпечних і шкідливих чинників, що притаманні роботі в інформаційно-аналітичному середовищі.



Рисунок 5.1 – Узагальнена класифікація небезпечних і шкідливих чинників, що притаманні роботі в інформаційно-аналітичному середовищі

До фізичних чинників відносяться: шум від комп'ютерного обладнання, коливання температури повітря, нераціональне розташування техніки, недостатнє провітрювання, наявність електростатичних розрядів. Хімічні фактори пов'язані з можливим випаровуванням шкідливих речовин з пластмасових поверхонь устаткування у разі перегріву або недотримання технічного регламенту.

Біологічні чинники, як правило, не виявляються при офісній роботі, однак в умовах пандемій або сезонного підйому захворюваності можуть мати вплив через використання спільного обладнання (клавіатури, миші, поверхні столу).

Психофізіологічні чинники виявляються у вигляді втоми, головного болю, погіршення концентрації, дратівливості. Такі прояви часто зумовлені не лише самою тривалістю роботи, а й надмірною складністю завдань, високим темпом діяльності, емоційним тиском під час прийняття рішень.

У таблиці 5.2 наведено узагальнені характеристики чинників ризику та відповідні заходи для зниження їхнього впливу на працівника.

Таблиця 5.2 – Характеристики чинників ризику та відповідні заходи для зниження їхнього впливу на працівника

Категорія чинників	Приклад впливу	Заходи безпеки
Фізичні	ЕМ-випромінювання, шум	Заземлення, фільтри, шумопоглинання
Хімічні	Випаровування з пластику	Вибір сертифікованої техніки, провітрювання
Психофізіологічні	Втома, стрес	Перерви, чергування розумової і фізичної роботи
Освітлювальні	Блиск на екрані, недостатнє світло	Регулювання яскравості, додаткове освітлення
Ергономічні	Неправильна поза, перенапруга	Ергономічні меблі, змінне положення тіла

Таким чином, аналіз небезпечних та шкідливих чинників у роботі з комп'ютером дозволяє своєчасно впровадити профілактичні заходи, що знижують ризики професійних захворювань і сприяють підтримці високої працездатності працівника під час виконання інтелектуальних завдань.

5.3. Інструкція із охорони праці під час розробки та використання алгоритму для формування маршрутів заготівлі продовольчої сировини

Під час розробки, тестування та практичного використання програмного забезпечення, що реалізує вдосконалений алгоритм АСО для формування маршрутів заготівлі продовольчої сировини, необхідно дотримуватись вимог охорони праці, спрямованих на збереження здоров'я користувача та запобігання виробничим ризикам.

Інструкція з охорони праці для розробника або користувача ПЗ передбачає дотримання наступних положень:

До початку роботи:

- перевірити справність персонального комп'ютера, наявність заземлення та відсутність пошкоджень кабелів;
- переконатися у правильному розташуванні обладнання: монітор має бути на відстані не менше 60 см від очей, а клавіатура розміщена так, щоб не спричиняти напруження рук;
- налаштувати робоче місце згідно з ергономічними вимогами, забезпечивши достатнє освітлення, температуру (21–24 °С) і вологість (40–60%).

Під час роботи:

- здійснювати перерви кожні 60 хвилин на 10–15 хвилин для зняття зорового та фізичного напруження;

- виконувати вправи для очей (рух поглядом по діагоналі, фокусування на віддалених предметах);
- змінювати положення тіла, вставати для розминки або короткої прогулянки;
- уникати роботи за комп'ютером у затемненому приміщенні або за наявності відблисків на моніторі;
- не допускати роботи з ПЗ у стані втоми або нервового напруження, що може вплинути на точність прийняття рішень.

Після завершення роботи:

- вимкнути комп'ютер та інші пристрої, від'єднати від джерела живлення при потребі;
- виконати вправи для релаксації м'язів рук та плечового поясу;
- зафіксувати в журналі спостережень можливі технічні збої або зауваження щодо роботи системи.

Додаткові положення:

- у разі виявлення будь-яких збоїв в роботі програмного забезпечення, які можуть створити ризик для збереження даних або перервати процес моделювання, необхідно припинити роботу та звернутись до технічного спеціаліста;
- усі дії, пов'язані з модифікацією програмного коду або роботою з відкритими бібліотеками, слід виконувати з попереднім збереженням проєкту та резервним копіюванням;
- не дозволяється виконання роботи без ознайомлення з положеннями цієї інструкції.

Дотримання вищезазначених правил дозволяє забезпечити безпечні умови праці при розробці інтелектуальних алгоритмів та сприяє зниженню рівня професійної втоми і ризику виникнення захворювань, пов'язаних із комп'ютерною діяльністю.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Актуальність теми кваліфікаційної роботи зумовлена тим, що сільське господарство сьогодні переживає цифрову трансформацію. У багатьох підприємствах досі використовуються ручні або неадаптивні підходи до побудови маршрутів, що призводить до необґрунтованих витрат ресурсів і втрати потенційного прибутку. Тому впровадження адаптивного мурашкового алгоритму, здатного враховувати динамічні фактори та географічні особливості сільськогосподарських регіонів, є своєчасним та важливим кроком до підвищення ефективності всієї системи логістичного управління.

Огляд методів оптимізації маршрутів транспортування сільськогосподарської продукції свідчить про зростаючу актуальність застосування інтелектуальних алгоритмів, зокрема мурашкових, як перспективного напрямку для підвищення ефективності логістичних систем у галузі агровиробництва.

Впродовж останніх десятиліть розроблено велику кількість алгоритмів, які можна умовно класифікувати на кілька основних груп: детерміновані, евристичні та метаевристичні. Подана класифікація алгоритмів маршрутизації дозволяє обрати метод, що найкраще відповідає конкретним умовам доставки сільськогосподарської продукції, враховуючи як технічні можливості, так і потреби підприємств.

Основна ідея мурашкового алгоритму (Ant Colony Optimization, ACO) полягає у симуляції поведінки колонії мурах, які, переміщаючись між вузлами графа, залишають на шляху умовну «феромонну мітку». Інші мурахи, орієнтуючись на концентрацію феромону, з більшою ймовірністю обирають маршрути, якими вже пройшли попередні особини. З часом ці феромонні сліди підсилюються для ефективних маршрутів і зникають для неефективних, що дозволяє алгоритму поступово сходитися до оптимального або близького до оптимального розв'язку.

Нами виконано порівняння класичних і мурашкових алгоритмів у логістиці. Встановлено, що мурашкові алгоритми демонструють високу ефективність для задач, де необхідно обрати найкращу комбінацію маршрутів серед великої кількості можливих варіантів. Їх здатність адаптуватися до складних і динамічних систем, таких як аграрна логістика, робить цей підхід надзвичайно актуальним у сучасних умовах оптимізації ресурсів і часу.

Нами виконано аналіз стану використання інтелектуальних методів формування маршрутів в аграрному секторі. У Нідерландах та Німеччині аграрні компанії активно використовують GPS-навігацію в комбінації з нейромережевим прогнозуванням потреби в транспортуванні. В Україні ці процеси перебувають на початковому етапі, проте спостерігається активний інтерес з боку агрохолдингів до таких інструментів у рамках проєктів AgTech. Водночас, є високий потенціалом для покращення ефективності логістики, особливо в умовах динамічного середовища та великої територіальної розпорошеності виробничих точок.

У роботі для реалізації алгоритму було обрано мову програмування Python, оскільки вона поєднує в собі потужний обчислювальний потенціал і простоту синтаксису, що особливо важливо при роботі з математичними моделями та візуалізацією. Саме ці переваги стали вирішальними при виборі інструменту для реалізації мурашкового алгоритму формування маршрутів заготівлі продовольчої сировини.

Розроблена блок-схема реалізації мурашкового алгоритму маршрутизації передбачає, що на кожній ітерації користувач може бачити побудовану карту маршрутів, а також отримати числові значення для оцінення ефективності: середню довжину шляху, кращий знайдений маршрут, час виконання тощо. Це забезпечує не лише автоматичний пошук оптимального рішення, але й прозорість у процесі прийняття логістичних рішень.

Правильне налаштування параметрів мурашкового алгоритму є не менш важливим, ніж його математична структура. Від них залежить не тільки результат, але й стабільність роботи моделі, особливо при масштабуванні до

великої кількості точок збору. Саме тому добір значень параметрів у рамках цього дослідження виконувався на основі тестування та аналізу поведінки моделі в умовах, наближених до реальних виробничих умов.

Нами обґрунтована цільова функція, яка дозволяє не лише формалізувати мету моделі, але й виступає основою для навчання, адаптації та порівняння ефективності маршрутів. У разі розширення задачі – наприклад, при додаванні складів, часових обмежень чи змінних витрат – ця функція може бути модифікована без необхідності переписування основного алгоритму.

Нами подано результати визначення виробничих умов заготівлі продовольчої сировини. У якості ілюстрації розглянуто фрагмент транспортної мережі Заболотцівської територіальної громади, що знаходиться у Золочівському районі Львівської області. Зокрема, ділянку дороги між селами Заболотці та Велин, де часткові ушкодження дорожнього покриття призводять до суттєвого зниження швидкості руху. Така ситуація чітко демонструє, як стан дорожньої інфраструктури впливає на побудову маршрутів (рис. 3.1).

У нашій роботі запропоновано вдосконалення класичного алгоритму оптимізації мурашиних колоній (АСО) для врахування реальних виробничих умов, зокрема в умовах надзвичайних ситуацій. Основні зміни стосуються поведінкових механізмів агентів (мурах), які імітують рух транспортних засобів на пошкодженій або обмежено доступній транспортній мережі сільської громади.

На основі раніше викладених положень було побудовано блок-схему вдосконаленого алгоритму формування маршрутів заготівлі продовольчої сировини в межах сільської громади, з урахуванням виробничих обмежень, які виникають у надзвичайних ситуаціях. Схему представлено на рисунку 3.10. Вона наочно ілюструє основні етапи роботи алгоритму, починаючи від введення вихідних даних і завершуючи визначенням оптимального маршруту з відповідними показниками ефективності.

Для практичної перевірки ефективності вдосконаленого алгоритму оптимізації мурашиних колоній було створено програмну модель, реалізовану

мовою програмування Python. Програмна реалізація враховує як класичні принципи АСО, так і доповнення, пов'язані з відображенням реального стану транспортної мережі, обмеженнями проїзду через ушкоджені ділянки доріг, а також необхідністю перерозподілу маршрутів у разі виникнення надзвичайних ситуацій.

Результати експериментального порівняння підтверджують доцільність використання вдосконаленого мурашкового алгоритму для вирішення задач оптимального формування маршрутів заготівлі продовольчої сировини. Удосконалений алгоритм забезпечує скорочення часу обчислення при зростанні кількості вершин. Якщо для задач з кількістю вузлів до 50 одиниць економія становить у середньому до 6%, то при переході до більш складних задач із 51 до 100 вершин – виграш за часом зростає до 12–15%.

Нами розроблено заходи щодо забезпечення безпечних умов праці. Проведено детальний аналіз потенційно небезпечних і шкідливих факторів, характерних для тривалої роботи з комп'ютером, а також розроблено інструкцію з охорони праці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барбашов І.А., Огнєвий О.В. Застосування алгоритму мурашиних колоній для вирішення задач маршрутизації транспортних засобів. Вимірювальна та обчислювальна техніка в технологічних процесах, 2014. №4. С.178-180.
2. Береза А. М. Основи створення інформаційних систем. Навч. посібник. 2–е вид., перероб. і доп. К.: КНЕУ, 2001. 214 с.
3. Вахнюк С.В. Технологія створення програмних та інтелектуальних систем: навчальний посібник / С. В. Вахнюк. Суми: ДВНЗ “УАБС НБУ”, 2011. 254 с.
4. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред. Гандзюка М. П. К.: Каравела 2003. 405 с.
5. Жидецький В.Ц. Основи охорони праці: Підручник. К.: Основа, 2002. 320с.
6. Коваль Н., Кисіль С., Тригуба А. Розробка бази даних інформаційної системи планування заготівлі молока. Інформаційна безпека та інформаційні технології: збірник тез доповідей V Всеукраїнської науково-практичної конференції молодих учених, студентів і курсантів, м. Львів, 26 листопада 2021 року. Львів, ЛДУ БЖД, 2021, С.137-139.
7. Коваль Н.Я., Кондисюк І.В., Тригуба А.М. Алгоритм навчання нейронної мережі для планування часу виконання робіт у гібридних проектах. Молодь у світі сучасних технологій за тематикою: Сучасні інформаційні технології: стан та перспективи розвитку : матеріали міжнар. наук.-практ. конф. (4 червня 2021 р., м. Херсон) за заг. ред. Г.О. Райко. Херсон: Видавництво ФОП Вишемирський В. С., 2021. С. 153-156.
8. Коротенко В. В. Рішення задачі комівояжера алгоритмом мурашиної колонії та порівняння з методом простого перебору: бакалаврська робота / Сумський державний університет. – Суми, 2021. – 41 с.

9. Кулик Я. А., Книш Б. П., Барабан М. В. Моделювання переміщення вантажів на основі алгоритму оптимізації колонії мурах // Вісник Вінницького політехнічного інституту. – 2022. – № 5. – С. 74–79.

10. Москальова В. М. Основи охорони праці. Підручник. Київ: ВД Професіонал, 2005. 666 с.

11. Мурашиний алгоритм для розв'язання задач маршрутизації: кваліфікаційна робота / Харківський національний університет радіоелектроніки. – Харків, 2022. – 65 с.

12. Павлюк Т., Волонтир Л. Використання сучасних інформаційних технологій в сільському господарстві. Формування ринкової економіки в Україні. 2017. 38. С. 122-127. URL: <http://repository.vsau.org/getfile.php/16997.PDF>

13. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. К. : Видавничо–поліграфічний центр «Київський університет», 2014. 640 с.

14. Сидорчук О. В., Тригуба А. М., Панюра Я. Й., Шолудько П. В. Особливості ситуаційного управління змістом та часом виконання робіт у інтегрованих проектах аграрного виробництва. Східно–європейський журнал передових технологій. 2010. № 1/2 (43). С. 46–48. URL: <http://journals.uran.ua/eejet/article/view/2507/2295>

15. Тригуба А. М., Луб П. М., Шарибура А. О., Грабовець В. В. Вплив територіального розташування пунктів заготівлі сільськогосподарської продукції на ефективність функціонування транспортної системи. Міжвузівський збірник наукових праць Луцького національного технічного університету. 2014. Вип. 45. С. 564–567.

16. Тригуба А. Системна модель цифрової трансформації сільських територіальних громад на основі обчислювального інтелекту. Вісник Львівського національного екологічного університету. Агроінженерні дослідження, 2022, № 26, р. 177–184.

17. Тригуба А., Коваль Н., Тригуба І., Падюка Р., Боярчук О. Системна модель цифрової трансформації сільських територіальних громад на основі обчислювального інтелекту. Вісник Львівського національного аграрного університету: агроінженерні дослідження. №26. Львів: Львів НАУ, 2022. С.177-184.

18. Тригуба А., Тригуба І., Чубик Р., Кондисюк І., Коваль Н., Панюра Я. Прогнозування обсягів заготівлі сировини на території громад із використанням штучних нейронних мереж. Вісник Львівського національного аграрного університету: агроінженерні дослідження. №24. Львів: Львів НАУ, 2020. С.143-151.

19. Тригуба А.М., Коваль Н.Я. Алгоритм прогнозування добових обсягів молока на території громад. Вчені Львівського національного аграрного університету виробництву: каталог інноваційних розробок; за заг. ред. В. В. Снітинського, І. Б. Яціва. Вип. 21. Львів: Львів. нац. аграр. ун-т, 2021. С. 51.

20. Тригуба А.М., Коваль Н.Я. Модель SARIMA для оперативного прогнозування обсягів заготівлі молока на території громад. Вчені Львівського національного аграрного університету виробництву: каталог інноваційних розробок; за заг. ред. В. В. Снітинського, І. Б. Яціва. Вип. 22. Львів: Львів. нац. аграр. ун-т, 2022. С. 27.

21. Тригуба А.М., Коваль Н.Я., Тригуба І.Л., Кисіль С.Р. Архітектура інформаційної системи оперативного планування заготівлі молока на території громад. Інформаційні технології в енергетиці та агропромисловому комплексі: матеріали Х-ї міжнародної наукової конференції, присвяченої 165-річчю університету. Львів-Дубляни, 2021, С. 111–113.

22. Тригуба А.М., Кондисюк І., Коваль Н. Алгоритм прийняття управлінських рішень в умовах невизначеності із використанням машинного навчання. Вчені Львівського національного аграрного університету виробництву: каталог інноваційних розробок; за заг. ред. В. В. Снітинського, І. Б. Яціва. Вип. 20. Львів: Львів. нац. аграр. ун-т, 2020. С. 39.

23. Худов Г. В., Хижняк І. А., Марченко В. П., Горошко О. О. Метод визначення маршруту руху транспортних засобів з використанням модифікованого алгоритму мурашиної колонії // Системи обробки інформації. – 2022. – № 3 (170). – С. 58–66. DOI: 10.30748/soi.2022.170.08.

24. Bing Maps URL: <https://www.bing.com/maps> (дата звернення 16.04.2025).

25. Chen J., Gui P., Ding T., Na S., Zhou Y. Optimization of Transportation Routing Problem for Fresh Food by Improved Ant Colony Algorithm Based on Tabu Search // Sustainability. – 2019. – Vol. 11, No. 23. – Article 6584. Tawfeek M.A., Alrashdi I., Alruwaili M., Jamel L., Elhady G.F., Elwahsh H. Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization // EURASIP Journal on Wireless Communications and Networking. – 2025. – Article 22. DOI: 10.1186/s13638-025-02449-w.

26. Chen J., Gui P., Ding T., Na S., Zhou Y. Optimization of Transportation Routing Problem for Fresh Food by Improved Ant Colony Algorithm Based on Tabu Search // Sustainability. – 2019. – Vol. 11, No. 23. – Article 6584. DOI: 10.3390/su11236584.

27. Google Maps. URL: www.google.com.ua/maps (дата звернення 11.04.2025).

28. Li H., Chen J., Wang F., Bai M. Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: a review. European Journal of Operational Research, 2021. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2021.02.022>

29. Li H., Wang H., Chen J., Bai M. Two-echelon vehicle routing problem with satellite bisynchronization. European Journal of Operational Research, 288(3): 775–793, 2021. doi: <https://doi.org/10.1016/j.ejor.2020.06.019>

30. OpenStreetMap. URL: <https://www.openstreetmap.org/#map=14/50.0456/24.9593> (дата звернення 16.04.2025).

31. Pandas [Електронний ресурс]. URL: <http://pandas.pydata.org/pandas-docs/stable/overview.html#license>.
32. Python (programming language). URL: <https://bitly.su/paHJe>.
33. Renjbar Sh. Othman, Ibrahim Mahmood Ibrahim. A review of exploring recent advances in ant colony optimization: applications and improvements // International Journal of Scientific World. – 2025. – Vol. 11, No. 1. – P. 114–122.
Yang Y., Feng T., Luo C., Li C., Wang J., Xiong Y. A hybrid ant colony optimization algorithm for drone-based cooperative EV routing problem with simultaneously delivery and pickup // Annals of Operations Research. – 2025. DOI: 10.1007/s10479-025-06581-0.
34. Tofighi S., Torabi S.A., Mansouri S.A. Humanitarian logistics network design under mixed uncertainty. Eur. J. Oper. Res. 2016. 250 (1). pp. 239-250.
35. Toregas C., Swain R., ReVelle C., Bergman L. The location of emergency service facilities. Oper. Res, 1971. 19 (6). pp. 1363-1373.
36. Tryhuba A., Ivanyshyn V., Chaban V., Mushenyk I., Zharikova O. Computer model of resource demand planning for dairy farms. Independent Journal of Management & Production (Special Edition ISE, S&P). 2021. 12(3), pp. 138-149. URL: <http://www.ijmp.jor.br/index.php/ijmp/article/view/1531/1971>
37. Tryhuba A. M., Koval N. Ya., Ratushnyi A. R., Tryhuba I. L., Shevchuk V. V. Algorithm for the routes formation of food raw materials procurement on the community territory taking into account the production conditions during emergency situations. Applied Aspects of Information Technology 2023; Vol.6 No.1. pp. 60–73.
38. Tryhuba A., Bashynsky O. Coordination of dairy workshops projects on the community territory and their project environment. 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 2019. 3. P. 51-54. URL: <https://ieeexplore.ieee.org/document/8929816> (Last accessed: 27.04.2021).
39. Tryhuba A., Boyarchuk V., Tryhuba I., Ftoma O., Padyuka R., Rudynets M., Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning, Proceedings of the 2nd International Workshop on Modern

Machine Learning Technologies and Data Science (MoMLLeT+DS 2020). Volume I: Main Conference. Lviv–Shatsk, Ukraine, June 2–3, 2020. pp.327–340.

40. Tryhuba A., Boyarchuk V., Tryhuba I., Ftoma O., Padyuka R., Rudynets. M. Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning. Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLLeT+DS 2020). 2020. I. P. 327-340.

41. Tryhuba A., Kondysiuk I., Tryhuba I., Koval N., Boiarchuk O., Tatomyr A. Intellectual information system for formation of portfolio projects of motor transport enterprises, in: I Workshop Information Technologies in Energy and Agro-industrial Complex, ITEA-WS 2021, CEUR Workshop Proceedings 3109, Dubliany, Lviv region, 2021, pp. 44–52.

Додатки

Додаток А

Фрагмент програмного коду удосконаленого АСО-алгоритму з урахуванням пошкоджень дорожнього покриття та умов маршрутизації

```
import numpy as np
import random

# ==== Параметри ====
n_nodes = 6 # кількість пунктів збору (вузлів)
alpha = 1.0 # вплив феромону
beta = 3.0 # вплив евристики (видимість)
rho = 0.2 # швидкість випаровування феромону
Q = 100 # константа для оновлення феромону
n_ants = n_nodes
n_iterations = 50

# ==== Матриця відстаней між вузлами (km) ====
distance_matrix = np.array([
    [0, 2, 9, np.inf, 6, 4],
    [2, 0, 7, 3, np.inf, 5],
    [9, 7, 0, 4, 2, np.inf],
    [np.inf, 3, 4, 0, 8, 6],
    [6, np.inf, 2, 8, 0, 3],
    [4, 5, np.inf, 6, 3, 0]
])

# ==== Матриця стану дороги (1 - нормальна, 0.5 - частково пошкоджена, 0 - непридатна) ====
condition_matrix = np.array([
    [0, 1, 1, 0, 1, 0.5],
    [1, 0, 0.5, 1, 0, 1],
    [1, 0.5, 0, 1, 1, 0],
    [0, 1, 1, 0, 0.5, 1],
    [1, 0, 1, 0.5, 0, 1],
    [0.5, 1, 0, 1, 1, 0]
])

# ==== Ініціалізація феромонної матриці ====
pheromone = np.ones_like(distance_matrix)

# ==== Основний цикл АСО ====
best_length = float('inf')
best_path = []

for iteration in range(n_iterations):
    all_paths = []
    all_lengths = []

    for ant in range(n_ants):
        visited = [ant]
        current = ant
        path_length = 0

        while len(visited) < n_nodes:
            probabilities = []
            for j in range(n_nodes):
                if j not in visited and condition_matrix[current][j] > 0 and
distance_matrix[current][j] != np.inf:
                    eta = 1 / distance_matrix[current][j]
                    tau = pheromone[current][j]
```

```

        delta = condition_matrix[current][j]
        p = (tau ** alpha) * (eta ** beta) * delta
        probabilities.append((j, p))
    if not probabilities:
        break
    nodes, probs = zip(*probabilities)
    probs_sum = sum(probs)
    probs = [p / probs_sum for p in probs]
    next_node = random.choices(nodes, weights=probs, k=1)[0]
    visited.append(next_node)
    path_length += distance_matrix[current][next_node]
    current = next_node

all_paths.append(visited)
all_lengths.append(path_length)

if path_length < best_length:
    best_length = path_length
    best_path = visited

# ==== Оновлення феромонів ====
pheromone = (1 - rho) * pheromone
for idx, path in enumerate(all_paths):
    length = all_lengths[idx]
    for i in range(len(path) - 1):
        a = path[i]
        b = path[i + 1]
        pheromone[a][b] += Q / length
        pheromone[b][a] += Q / length # для симетрії

# ==== Результат ====
best_path, round(best_length, 2)

```