

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

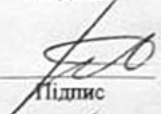
на тему Метод для формування вантажних перевезень з урахуванням  
фрахту за еволюційним алгоритмом

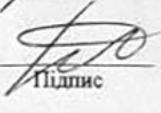
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент групи КН-20-2  Антон НЕБОЖИНСЬКИЙ  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
зав. кафедри КН, д.т.н., професор

20 червня 2024 р.



Олександр БАРМАК  
Ім'я, ПРІЗВИЩЕ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК

«16» лютого 2024 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом»

2. Завдання видано студенту Антону НЕБОЖИНСЬКОМУ

(Ім'я, прізвище)

3. Керівник роботи к.т.н. доцент кафедри КН Руслан БАГРІЙ

(посада, ім'я, прізвище)

4. Затверджено наказом університету від «15» лютого 2024 р. № 8

5. Дата видачі завдання студенту: «16» лютого 2024 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – підвищення ефективності процесу планування маршрутів вантажних перевезень з урахуванням вартості фрахту, спрямованого на зменшення витрат на транспортування вантажів. Для вирішення поставленої задачі необхідно розробити метод формування вантажних перевезень з урахуванням фрахту, в основі якого буде лежати еволюційний алгоритм. Вхідними даними методу є список портів, зв'язків між портами, та вартостей фрахту для проходження цими зв'язками. Вихідними даними методу є знайдений оптимальний маршрут, тобто найдешевший маршрут для доставки вантажу та вартість фрахту проходження ним. Необхідно провести оцінку підвищення ефективності розробленого методу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:


№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2024	виконано
3	Робота над розділом 1– Огляд підходів до систем планування маршрутів вантажних перевезень	лютий 2024	виконано
4	Робота над розділом 2– Метод планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом	березень 2024	виконано
5	Робота над розділом 3– Програмна реалізація методу планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом	квітень 2024	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2024	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2024	виконано
8	Захист кваліфікаційної роботи бакалавра	червень 2024	виконано

Виконавець: студент групи КН-20-2  
Група виконавця

  
Підпис

Антон НЕБОЖИНСЬКИЙ  
Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  
Науковий ступінь, посада

  
Підпис

Руслан БАГРІЙ  
Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-2 Антон НЕБОЖИНСЬКИЙ

Керівник кваліфікаційної роботи бакалавра: к.т.н, доц. каф. КН Руслан БАГРІЙ

Кваліфікаційна робота бакалавра містить:


Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
68	39	4	28	3

Метою кваліфікаційної роботи бакалавра є підвищення ефективності процесу планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів. Для розробки інформаційної системи було використано мову програмування Python та середовище розробки PyCharm.

Реалізована система призначена для логістичних компаній. Де спеціально навчені люди зможуть використовувати її для швидкого автоматизованого прокладання ефективних маршрутів, враховуючи багато факторів та дозволяючи зменшити витрати на їх будівництво.

Напрямами практичного застосування розробленої інформаційної системи визначено автоматизоване планування маршрутів вантажних перевезень з врахуванням фрахту, та графічне відображення необхідного маршруту.

Ключові слова: Вантажоперевезення, фрахт, інформаційна система, еволюційний алгоритм, маршрут.

Виконавець: студент групи КН-20-2  Антон НЕБОЖИНСЬКИЙ  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	4
Вступ.....	5
Розділ 1 Характеристика предметної області формування вантажоперевезень....	6
1.1 Аналіз предметної області формування вантажоперевезень .....	6
1.2 Огляд еволюційних алгоритмів для вирішення задачі планування маршрутів вантажних перевезень .....	9
1.3 Аналіз існуючих програмних засобів для планування маршрутів вантажних перевезень .....	14
1.4 Мета та завдання кваліфікаційної роботи .....	19
Розділ 2 Метод планування маршрутів вантажних перевезень .....	20
2.1 Загальна схема методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом.....	20
2.1.1 Вхідні дані методу планування маршруту .....	22
2.1.2 Ініціалізація параметрів методу планування маршрутів вантажних перевезень .....	23
2.1.3 Проходження мурахами маршрутів.....	25
2.1.4 Оновлення феромонів на знайдених маршрутах.....	28
2.2 Формулювання задачі для методу планування маршрутів вантажних перевезень .....	29
2.3 Проектування бази даних.....	31
2.5 Спосіб оцінювання ефективності методу планування маршрутів вантажних перевезень .....	34
2.6 Висновки до розділу 2 .....	36
Розділ 3 Програмна реалізація методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом.....	37
3.1 Програмні бібліотеки та середовище розробки методу планування маршрутів вантажних перевезень .....	38
3.2 Структура та особливості реалізації методу формування вантажоперевезень з врахуванням фрахту за еволюційним алгоритмом .....	42
3.2.1 Вхідні дані методу планування маршрутів вантажних перевезень.....	43
3.2.2 Робота мурашиного алгоритму пошуку найдешевшого маршруту.....	47

3.2.2 Вивід результатів роботи методу пошуку найдешевшого маршруту ....	50
3.3 Підготовка робочих вхідних даних для методу формування вантажоперевезень .....	52
3.4 Тестування методу планування маршрутів вантажних перевезень з врахуванням фрахту.....	55
3.5. Оцінка ефективності запропонованого методу.....	61
3.6 Висновки до розділу 3 .....	62
Загальні висновки.....	64
Перелік посилань.....	66
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
ІТ	Інформаційні технології
ПЗ	Програмне забезпечення
ШІ	Штучний інтелект
МА	Мурашиний алгоритм
ЛСПМ	Логістичні системи планування маршрутів
БД	База даних
ЕА	Еволюційні алгоритми
ГА	Генетичний Алгоритм
МП	Морські перевезення
TSP	Travelling salesman problem

## Вступ

Кваліфікаційна робота бакалавра присвячена підвищенню ефективності процесу планування маршрутів вантажних перевезень з урахуванням вартості фрахту, спрямованого на зменшення витрат на транспортування вантажів.

**Актуальність.** Розробка методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом є актуальною в контексті сучасних вимог до оптимізації та планування транспортних логістичних процесів. Еволюційні алгоритми, такі як генетичні алгоритми, генетичне програмування, роями чи мурашині алгоритми, стали популярними і ефективними методами для вирішення складних задач оптимізації.

**Об'єкт дослідження** – процес оптимізації планування маршрутів вантажних перевезень за критерієм вартості фрахту.

**Предмет дослідження** – методи оптимізації, еволюційні алгоритми, технології та методи проектування інформаційних систем

**Мета кваліфікаційної роботи бакалавра** – підвищення ефективності процесу планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- провести аналіз еволюційних алгоритмів для вирішення задачі планування маршрутів вантажних перевезень з урахуванням фрахту;
- розробити метод планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів;
- розробити програмну реалізацію методу планування маршрутів вантажних перевезень з урахуванням фрахту;
- провести тестування програмної реалізації методу на різних тестових наборах даних та оцінити підвищення ефективності процесу планування маршрутів.

## **Розділ 1 Характеристика предметної області планування маршрутів вантажних перевезень**

### **1.1 Аналіз предметної області планування маршрутів вантажних перевезень**

Логістика невпинно розвивалась разом з розвитком людства, у людей завжди була необхідність у перевезенні вантажів з однієї точки в іншу, максимально ефективним чином, від цього залежало виживання спільноти, наприклад якщо на групу мисливців з здобиччю нападав хижак через те, що був погано обраний маршрут, або доставка вантажу займала багато часу, то плем'я могло залишитись без продовольства, що зменшувало його шанси на виживання. Прикладом є битва 1862 року під Фредеріксбургом [1], коли північні війська намагались прорвати південний фронт генерала Роберта Лі, проте зазнали поразки, однією з причин поразки вважається надто пізнє прибуття підтримки, яка мала надати допомогу військам що були задіяні в атаці.

З часом модернізувались засоби перевезення вантажів, спочатку здобич та товари носили на собі, потім використовували носилки, потім придумали сані, далі люди одомашнили коней, завдяки покращення навичок роботи з деревом люди винайшли колесо, човни, що створило передумови до появи таких засобів як віз та вітрильні кораблі. Після появи парового двигуна, віз еволюціонував в автомобіль, а вітрильний корабель в пароплав, так з'явилися вантажні автомобілі, поїзди та вантажні судна.

Для планування маршрутів вантажних перевезень, людям, які займалися плануванням маршрутів, потрібно було виконувати планування найкоротшого маршруту згідно принципу найбільшої ефективності, потрібно було спланувати витрати палива, їжі, можливих витрат за проходження певних приватних ділянок, відстані маршруту та інших характеристик, так з'явилося поняття фрахт. Фрахт – це сукупна вартість витрат на проходження маршруту, яка включає в себе багато різних характеристик, основні характеристики – це відстань маршруту, розмір вантажу, тип вантажу, безпека маршруту, оплата мит

та інші. Всі ці характеристики впливають на остаточну вартість фрахту. Вартість фрахту рахують як вартість перевезення одного звичайного контейнеру, який зазвичай розміром 20 футів (рис. 1.1).



Рисунок 1.1 – Стандартний контейнер для вантажоперевезень [3]

Підходи до побудови логістики для планування маршрутів вантажних перевезень з часом також змінювались, люди винаходили ефективні методи побудови систем вантажоперевезень, використовуючи технології які були їм доступні в контексті часу, в якому вони проживали [4]. Так люди здавна розуміли що ефективне перевезення вантажу полягає в побудові найкоротшого маршруту з мінімальними ризиками для вантажу та тих хто його доставляє, розвиток логістики будувався саме на таких принципах. Спочатку люди помітили що створення перевалочних пунктів з додатковими запасами їжі та засобами доставки вантажу збільшило швидкість доставки вантажоперевезення. А створення озброєної охорони вантажу та будування маршруту який уникав потенційні небезпечні місця на користь більш безпечних допомогло зберігати ватаж цілим та неушкодженим.

Так і в сучасному світі потреба в формуванні вантажоперевезень ефективним чином, а саме в побудові вантажоперевезень які будуються на принципах пошуку найкоротшого маршруту, при якому ризики для безпеки товару чи його перевізника мінімальні, залишається незмінною. Сучасні вантажоперевезення будуються переважно на морських вантажоперевезеннях [5].

Великі танкери курсують океанами перевозючи величезні маси вантажів. Сфера вантажоперевезень є надзвичайно важливою для світової економіки [6], а компанії які використовують якісні методи планування маршрутів вантажних перевезень, які допомагають сформувати максимально ефективний маршрут, високо оцінюються. З врахуванням сучасних реалій – збільшення ризиків для вільного судноплавства у міжнародних водах (піратство Хуситів поблизу червоного моря [7], Сомалійського піратства [8] та інших зон напруження), існує гостра потреба в ефективних методах планування маршрутів вантажних перевезень, особливо МП, які забезпечать швидкість доставки вантажу, мінімізуючи ризики для доставки товару.

В 21 столітті з розвитком інформаційних технологій, з'явилося багато можливостей автоматизації різних бізнес процесів і сфера планування маршрутів вантажних перевезень не виняток [9]. В сучасній сфері вантажоперевезення, логістичним компаніям яким необхідно прокласти маршрут, тепер не потрібно користуватись паперовими мапами та документами. Цифрові технології покращили цей процес, зробивши його зручним для спеціалістів які відповідають за планування маршрутів вантажних перевезень, адже комп'ютер на відмінну від людини здатен опрацьовувати мільйони операцій за доли секунди, достатньо використати систему яка підходить для вирішення тої чи іншої задачі, та завантажити в неї необхідні вхідні дані, це все відбулось завдяки розвитку ІТ.

Системи автоматизованого планування маршрутів можуть допомогти вирішити проблему планування маршрутів вантажних перевезень. Вони дозволяють вказати початкову та кінцеву точки доставки, вартість фрахту, після чого система автоматично формує найдешевший маршрут доставки. Основна мета таких систем полягає в знаходженні найдешевших маршрутів та наданні результатів користувачу.

Такі системи значно підвищують ефективність прокладання маршрутів вантажоперевезень, для їх використання необхідно щоб підприємства що займаються логістичними вантажоперевезеннями встановили відповідне ПЗ та

навчили необхідний персонал. В системах автоматизованого планування маршрутів вантажних перевезень з врахуванням фрахту є можливість вказати необхідні дані для формування маршруту, після чого система сформує вантажоперевезення з врахуванням фрахту. Логістичні системи автоматизованого планування маршрутів вантажних перевезень дозволяють зменшити залежність процесу планування маршруту від людського фактора [10].

Отже вирішити проблеми планування маршрутів вантажних перевезень з врахуванням фрахту допоможе метод планування маршрутів вантажних перевезень який дозволить мінімізувати людський фактор в процесі формування маршруту та побудувати найдешевший маршрут вантажоперевезення, шляхом введення необхідних вхідних даних, що дозволить підприємству витратити менше коштів та зменшити витрати на вантажоперевезення.

## **1.2 Огляд еволюційних алгоритмів для вирішення задачі планування маршрутів вантажних перевезень**

Для вирішення завдання планування маршрутів вантажних перевезень, а саме планування найкоротшого маршруту використовується напрямок ШІ – еволюційні алгоритми [11-13], вони використовуються для вирішення задач оптимізації та пошуку рішень в складних системах.

Для реалізації завдання планування маршрутів вантажних перевезень вантажоперевезень з врахуванням фрахту, проведемо аналіз ЕА, які використовуються для вирішення подібних задач. При розгляді еволюційних алгоритмів, що використовуються для вирішення задачі планування маршрутів вантажних перевезень за допомогою ІТ, розглянемо еволюційні алгоритми які використовуються для вирішення подібних завдань. Дані еволюційні алгоритми дозволяють підвищити ефективність логістичних процесів, а саме шляхом знаходження оптимального рішення, наприклад пошуку маршруту проходження якого буде коштувати мінімальну вартість.

Розглянемо наступні еволюційні алгоритми, а саме: звичайний генетичний алгоритм [14], множинний генетичний алгоритм [15], мурашиний алгоритм [16]. Дані алгоритми можуть використовуватись для вирішення задач планування маршрутів вантажних перевезень з врахуванням фрахту, а саме пошуку найдешевшого маршруту.

Звичайні генетичні алгоритми моделюють еволюційні процеси на основі природнього відбору (рис. 1.2) для пошуку оптимального рішення.

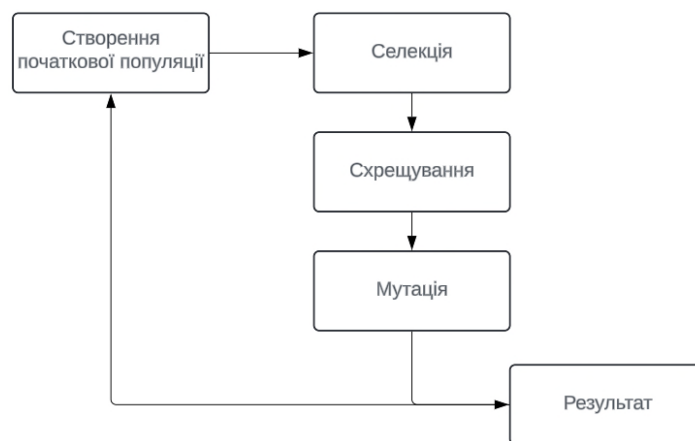


Рисунок 1.2 – Схема роботи звичайного ГА

Алгоритм включає в себе генерацію випадкової популяції рішень (рішення повинні представлятись як множина чисел), це як набір генів (рис. 1.3).

Вибір найкращих рішень (природній відбір), їх селекцію, схрещування та мутацію для генерації нащадків та покращення рішень з кожним поколінням. Перевагами даного алгоритму є здатність працювати з великою кількістю вхідних параметрів та простота в реалізації, даний алгоритм підходить для задач які важко сформулювати математично (наприклад навчити віртуальну істоту ходити). Недоліком звичайних генетичних алгоритмів є те, що дані алгоритми потребують певних обчислювальних потужностей при роботі з великим набором даних, можуть бути недостатньо ефективними для пошуку оптимального рішення в порівнянні з іншими алгоритмами, якщо задачу можна математично сформулювати, а також звичайні генетичні алгоритми не гарантують збереження найкращих рішень.

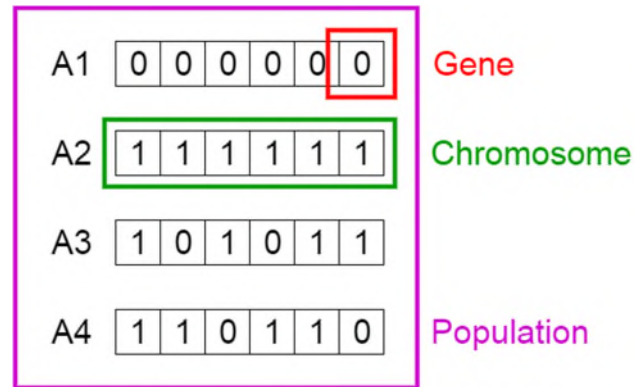


Рисунок 1.3 – Схема представлення вхідних даних в ГА [17]

Множинні генетичні алгоритми використовуються для розв’язання задач оптимізації. Множинні генетичні алгоритми працюють з декількома популяціями, а не з однією, що дозволяє шукати найкращі рішення одночасно, кожна популяція може використовуватись для вирішення певної підзадачі чи варіанту проблем. Перевагою множинних генетичних алгоритмів є можливість формування кількох найкращих рішень одночасно. Недоліком множинних алгоритмів є те, що вони потребують більше обчислювальних ресурсів через роботу з більшою кількістю варіантів рішень, головним недоліком множинних генетичних алгоритмів, є те що вони погано підходять для задач які легко формуються математично.

Мурашині алгоритми використовуються для пошуку оптимальних шляхів. В їх основі лежить приклад використання поведінки реальних мурах, які шукають найкоротший шлях від колонії до джерела їжі. Мурашині алгоритми ефективно вирішують задачу Комівояжера (TSP) [18] та інші оптимізаційні задачі, особливо задачі що пов’язані з роботою з графами та маршрутизацією. Принцип роботи мурашиного алгоритму базується на біологічних процесах, що виникають у мурах. Мурашині колонії ефективно само підтримуються, завдяки тому що мурахи проходять всі можливі маршрути в пошуках їжі, і якщо знаходять їжу, то починають ходити туди регулярно за знайденим маршрутом, залишаючи спеціальний слід – феромон. Мурашиний алгоритм працює схожим чином (рис. 1.4).

Для роботи мурашиного алгоритму, створюються початкові параметри, далі запускаються агенти «мурахи» по всім можливим маршрутам, після чого на кожному маршруті оновлюються феромони – показник що вказує «популярність» маршруту, тобто те як часто ним ходять мурахи, після чого відбувається перевірка виходу з умови (наприклад чи знайдено найкоротший маршрут, або чи закінчилась виділена кількість проходжень) та в залежності від умови відбувається або вихід з програми та вивід результатів, або повернення до етапу «Розміщення мурах».

Множинні генетичні алгоритми є складними у реалізації та вимагають більше обчислювальних ресурсів, тому що вони працюють з кількома популяціями для пошуку кількох можливих рішень. Також недоліком множинних генетичних алгоритмів є те, що якщо потрібно сформувати один найкоротший маршрут, то їх використання буде зайвим та збільшить складність алгоритму.

Звичайні генетичні алгоритми є складними та важко підходять для реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту, тому що дану задачу можна сформулювати математично, в той час як генетичні алгоритми більше підходять для задач які важко сформулювати відповідним чином, також існує проблема формалізації задачі під потребу генетичного алгоритму, а саме налаштувати вхідні дані для відображення у спеціальному форматі.

Мурашині алгоритми підходять для вирішення завдання планування маршрутів вантажних перевезень з врахуванням фрахту, тому що вони ефективно пристосовані для задач маршрутизації та роботи з графами, а задача планування маршрутів вантажних перевезень підходить для відображення у необхідному для алгоритму форматі, через те що вони призначені для роботи з графами.



Рисунок 1.4 – Схема роботи мурашиного алгоритму

Мурашині алгоритми є ефективними завдяки ефективній пристосованості алгоритмів до роботи з маршрутизацією та графами. Вони дозволяють знаходити множину найкращих рішень завдяки моделюванню біологічних процесів колонії мурах. Дані алгоритми запускають певну кількість агентів по всім можливим маршрутам та знаходять всі можливі маршрути, відбираючи тільки ефективні.

Проаналізувавши еволюційні алгоритми які використовуються для пошуку оптимального маршруту (найдешевшого), а саме генетичних алгоритмів, множинних генетичних алгоритмів та мурашиних алгоритмів, можна зробити висновок, що для вирішення планування маршрутів вантажних перевезень з врахуванням фрахту, а саме формування найкоротшого маршруту враховуючи

необхідні вхідні параметри, мурашині алгоритми є найкращим рішенням для вирішення даної проблеми з високою ефективністю.

### **1.3 Аналіз існуючих програмних засобів для планування маршрутів вантажних перевезень**

Проведемо аналіз існуючих програмних засобів, які вирішують задачу планування маршрутів вантажних перевезень. Для вирішення завдання формування маршрутів вантажоперевезень існує ряд програмних засобів, в основі яких лежать ЛСПМ, вони дозволяють ефективно виконувати задачі перевезення вантажів з врахуванням фрахту. Дані програмні засоби вирішують проблему планування маршрутів вантажних перевезень використовуючи системи з методами пошуку ефективних маршрутів.

Система “3GTMS” [19] (рис. 1.5) відповідає за формування маршрутів вантажоперевезень з початкової точки до кінцевої, щоб сформувати маршрут вантажоперевезення користувачу необхідно вказати усі необхідні дані, після чого система почне формування маршруту для вантажоперевезень, та відобразить необхідну інформацію.

Дана система дозволяє створювати логістичні замовлення, формувати маршрут для вантажоперевезення, заповнювати інформацію про замовлення (рис. 1.6). Однак дана система не дозволяє створити найкоротший маршрут, в ній немає можливості сформувати маршрут морських вантажоперевезень та врахувати фрахт вантажоперевезення.

Система має додаткові функціональні можливості, такі як візуалізація точок на карті, та заповнення інформації про замовлення. Також в системі присутні функції формування маршрутів за вказаними параметрами, визначення кількості та якості логістичних пунктів.

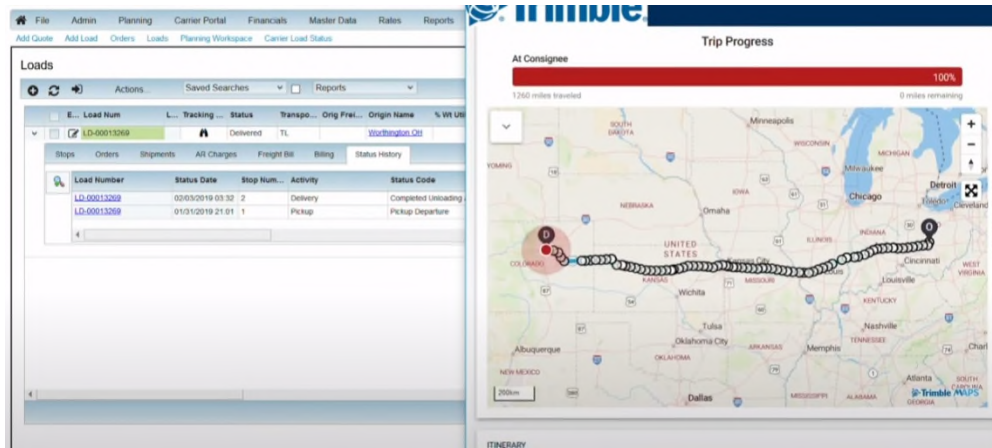


Рисунок 1.5 – Зображення інтерфейсу системи “3GTMS” [20]

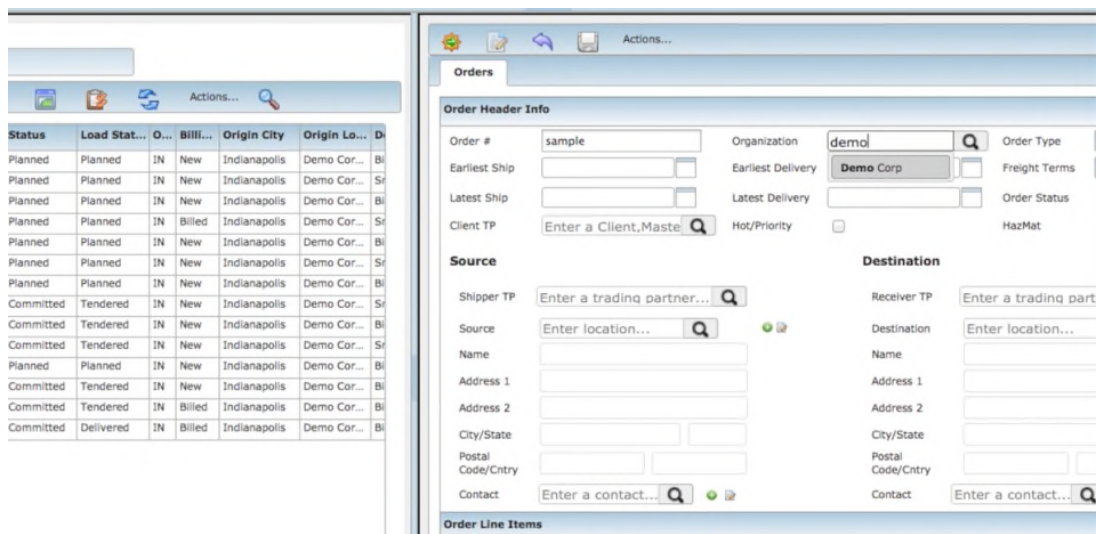


Рисунок 1.6 – Зовнішнє відображення інтерфейсу для вводу інформації в системі “3GTMS” [20]

В даній системі було виявлено ряд недоліків, таких як: відсутнє якісне формування найкоротшого маршруту, прокладання маршруту відбувається тільки по суші, формування маршруту не враховує фрахт. Також виявлено розбіжність у розрахунках часу доставки.

Розглянемо існуючий програмний засіб для планування маршрутів вантажних перевезень під назвою “Ant Logistic” [21] (рис. 1.7). Дана система призначена для автоматизації робочого місця адміністратора з логістики підприємства, що займається вантажоперевезеннями. Дана система здатна будувати найкоротші маршрути в межах міста, працює автоматично, будує найкоротший маршрут.

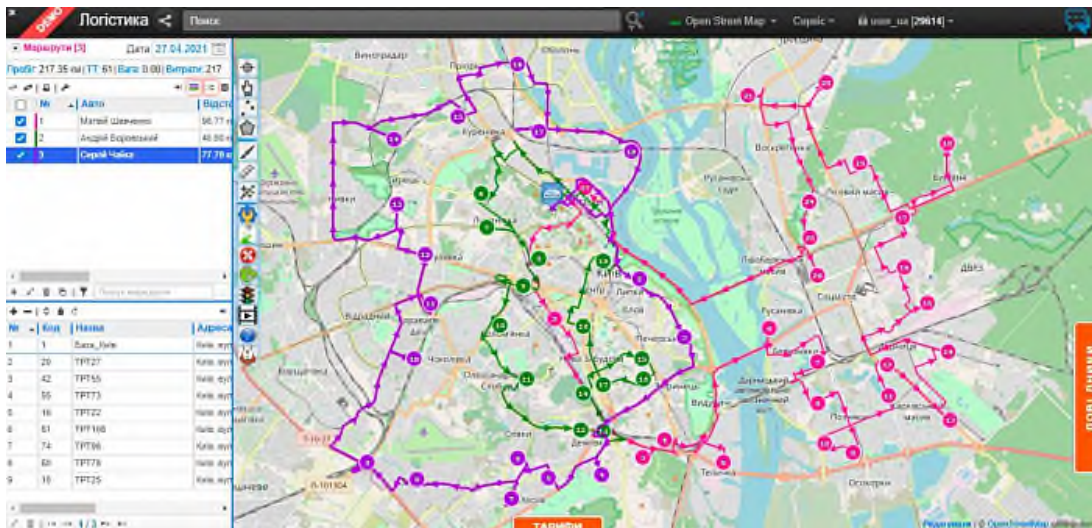


Рисунок 1.7 – Вигляд робочого інтерфейсу системи "Ant Logistics" [22]

У даній системі є можливість формувати найкоротші маршрути, зберігати інформацію та проводити автоматизоване планування маршрутів вантажних перевезень, шляхом цифровізації процесу формування найкоротшого маршруту. Також дана система надає можливість налаштовувати документи які необхідні для функціонування логістичних компаній (рис. 1.8).

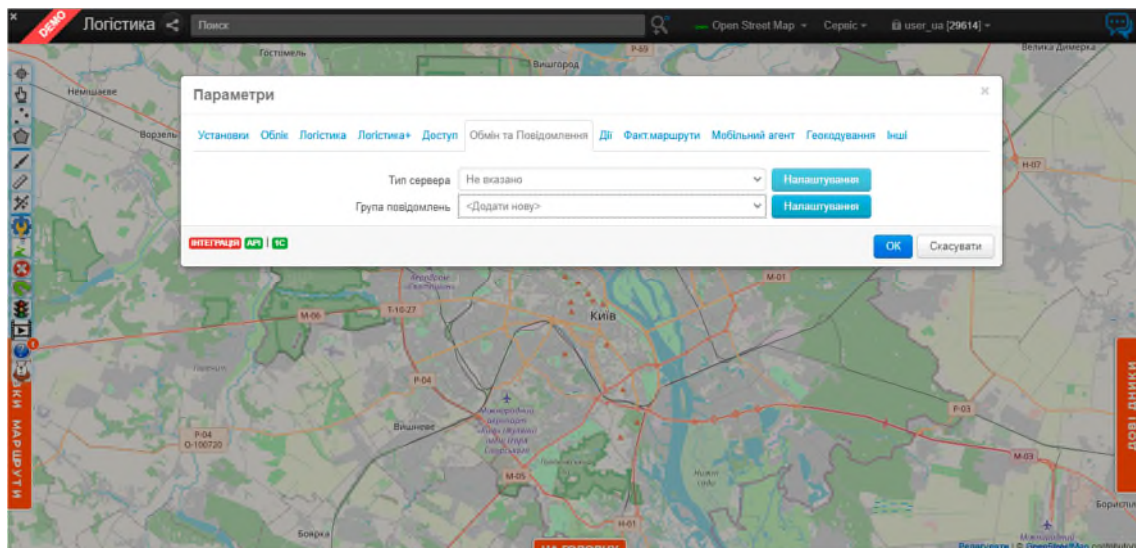


Рисунок 1.8 – Візуальне зображення робочого вікна налаштувань в системі "Ant Logistics" [23]

Розглянемо програмний засіб для планування маршрутів вантажних перевезень під назвою “CargoSmart” [24] (рис. 1.9). Дана система пропонує рішення для маршрутизації МП, оптимізації транспортних ланцюгів та управління ризиками. Дана система надає інноваційні технології для покращення ефективності та надійності перевезень, завдяки цифровізації та автоматизації процесів формування найкоротших маршрутів, ця система збільшила ефективність процесу формування найкоротшого маршруту, та його якість, система охоплює велику кількість логістичних пунктів по всьому світу, але не дозволяє вказувати проміжні логістичні пункти. Дана система здатна будувати морські маршрути, працює автоматично, є можливість заповнювати необхідну інформацію для планування маршрутів вантажних перевезень.

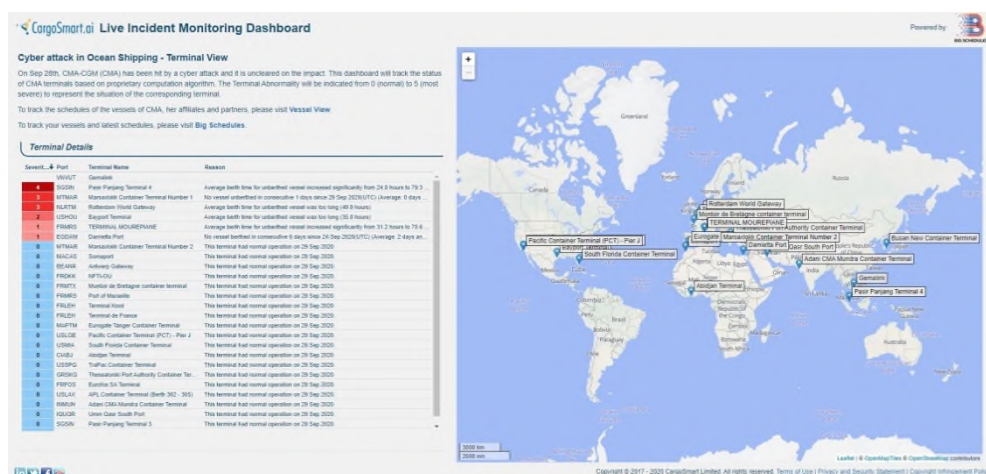


Рисунок 1.9 – Візуальне зображення робочого вікна в системі "CargoSmart" [25]

Дана система має ряд недоліків, маршрути формуються не точно, система не враховує фрахт під час планування маршрутів вантажних перевезень. У системі немає можливості задати проміжні логістичні пункти, система не якісно будує маршрути для планування маршрутів вантажних перевезень, маршрути не є найкоротшими, дана система працює повільно, а також система не враховує фрахт.

Однак дана система не дозволяє створити найкоротший маршрут, в ній немає можливості сформувавши маршрут морських вантажоперевезень та врахувати фрахт вантажоперевезення.

Система “3GTMS” має ряд недоліків, вона створена для формування найкоротших маршрутів у місті, вона не створена для вантажоперевезень морським шляхом, в даній системі маршрути формуються не ефективно та немає врахування фрахту, відсутнє якісне формування найкоротшого маршруту, прокладання маршруту відбувається тільки по суші, формування маршруту не враховує фрахт. Також виявлено розбіжність у розрахунках часу доставки.

Система “Ant Logistic” здатна будувати найкоротші маршрути в межах міста, працює автоматично, будує найкоротший маршрут. Недоліком даної системи є те, що вона не призначена для планування маршрутів морських вантажних перевезень, в системі не враховується вартість фрахту, немає можливості обрати пріоритети доставки, планування маршрутів відбувається повільно, в системі немає можливості обрати пріоритет для планування маршрутів вантажних перевезень.

Система "CargoSmart" система пропонує рішення для маршрутизації морських перевезень, оптимізації транспортних ланцюгів та управління ризиками. Дана система має ряд недоліків, планування маршрутів відбувається не точно, система не враховує фрахт під час планування маршруту вантажних перевезень. У системі немає можливості задати проміжні логістичні пункти, система не якісно будує маршрути для планування маршрутів вантажних перевезень, маршрути не є найкоротшими, дана система працює повільно, а також система не враховує фрахт.

Отже, були розглянуті програмні рішення, які вирішують задачу планування маршрутів вантажних перевезень, системи “CargoSmart”, “Ant Logistics”, “3GMTS” не мають достатнього функціоналу, який би забезпечив якісне планування найдешевших маршрутів вантажних перевезень. Проаналізувавши існуючі програмні рішення, було виявлено що існує потреба в розробці системи автоматизованого планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом.

## 1.4 Мета та завдання кваліфікаційної роботи

Метою кваліфікаційної роботи бакалавра є підвищення ефективності процесу планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- провести аналіз еволюційних алгоритмів для вирішення задачі планування маршрутів вантажних перевезень з урахуванням фрахту;
- розробити метод планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів;
- розробити програмну реалізацію методу планування маршрутів вантажних перевезень з урахуванням фрахту;
- провести тестування програмної реалізації методу на різних тестових наборах даних та оцінити підвищення ефективності процесу планування маршрутів.

## **Розділ 2 Метод планування маршрутів вантажних перевезень**

### **2.1 Загальна схема методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом**

Метод планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом призначений для побудови оптимального (найдешевшого) маршруту розвозу вантажів різних типів по різних шляхам.

Проаналізувавши еволюційні алгоритми які використовуються для пошуку оптимального маршруту (найдешевшого), а саме генетичних алгоритмів, множинних генетичних алгоритмів та мурашиних алгоритмів, а саме їх переваги та недоліки, еволюційним алгоритмом, для методу планування маршрутів вантажних перевезень з врахуванням фрахту було обрано мурашиний алгоритм.

Мурашині алгоритми використовуються для задач оптимізації, тому вони ефективно справляться з задачею пошуку оптимального маршруту. Мурашині алгоритми працюють таким чином, що моделюють біологічні природні процеси мурашиної колонії, таким чином знаходячи оптимальне рішення. Створюються «мурахи», які ідуть певними маршрутами, кожна мураха залишає за собою слід, який називається феромоном, його відчують інші мурахи, і чим сильніше слід, тим більше мурахи будуть іти тим чи іншим маршрутом. Мурашиний алгоритм використовує графи, тому для нього задачі які пов'язані з пошуком маршрутів по заданим параметрам є природними.

Схема методу планування маршрутів вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом графічно відображена на рисунку 2.1. На схемі відображенні стадії роботи даного методу.

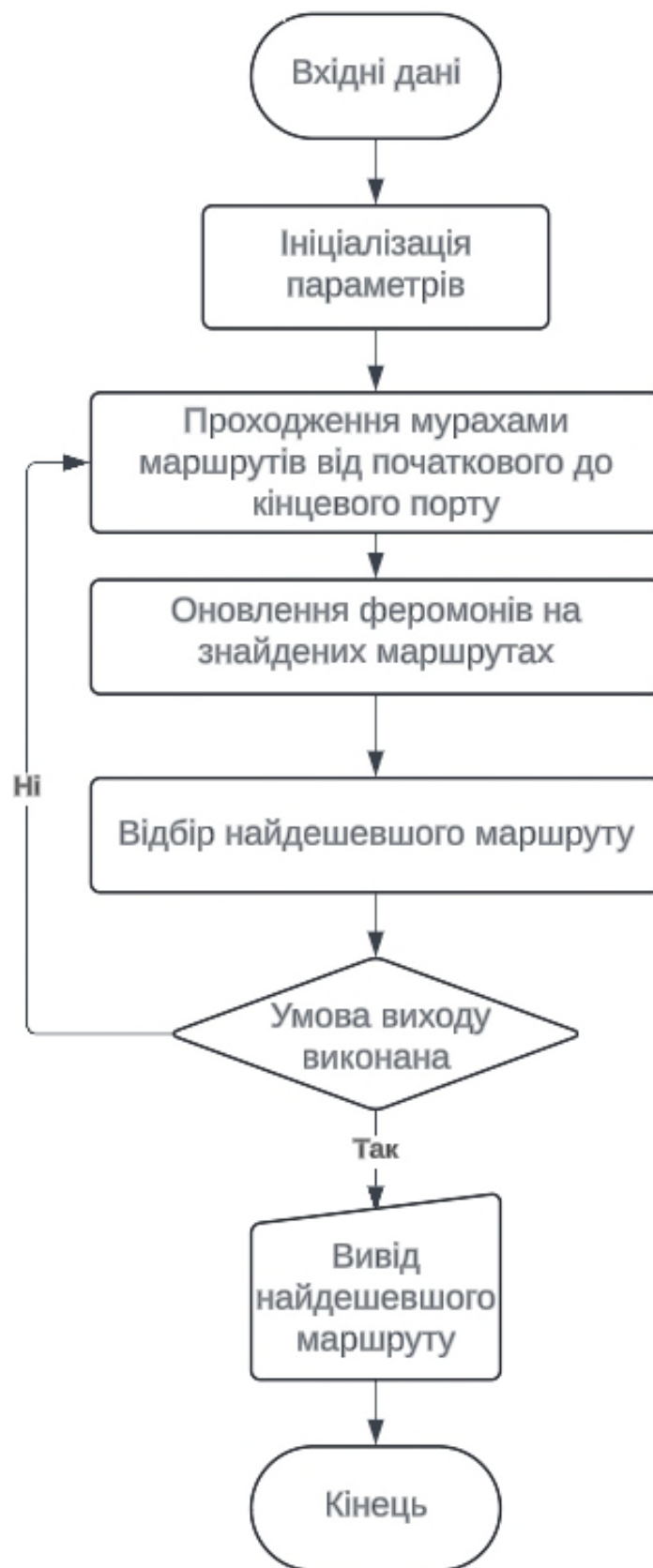


Рисунок 2.1 – Загальна схема роботи методу планування маршрутів вантажних перевезень

### 2.1.1 Вхідні дані методу планування маршруту

Вхідні дані методу планування маршрутів вантажних перевезень з врахуванням фрахту, що побудований на основі мурашиного алгоритму діляться на такі групи:

Користувацькі:

1. Список портів
2. Список маршрутів між портами
3. Список вартості фрахту кожного маршруту
4. Список типів вантажів які необхідно розвести
5. Список портів відправлення для кожного вантажу
6. Список портів призначення для кожного вантажу

Дані налаштування алгоритму:

1. Кількість мурах які будуть проходити маршрути
2. Кількість найкращих мурах
3. Кількість ітерацій
4. Коефіцієнт випаровування феромонів
5. Значення впливу феромонів
6. Значення впливу евристичної інформації

Користувацькі вхідні дані для методу пошуку найдешевшого маршруту, необхідні для вказування необхідних умов задачі, а саме – які існують порти, які існують зв'язки між ними, яка вартість фрахту для кожного маршруту який був вказаний між портами, також які існують вантажі та порти з яких їх потрібно відправити, та куди їх потрібно довести.

Дані налаштування алгоритму методу пошуку найдешевшого маршруту відповідають за налаштування параметрів роботи мурашиного алгоритму, для його налаштування існують наступні вхідні дані – кількість мурах що будуть проходити маршрути, кількість найкращих мурах, кількість ітерацій, коефіцієнт випаровування феромонів, значення впливу феромонів, значення впливу евристичної інформації.

### **2.1.2 Ініціалізація параметрів методу планування маршрутів вантажних перевезень**

На етапі ініціалізації параметрів методу пошуку планування маршрутів вантажних перевезень відбувається процес формування наданих вхідних даних у зрозумілий для алгоритму вигляд, що дозволяє алгоритму почати коректно працювати. На етапі ініціалізації, вхідні дані трансформуються певним чином, щоб алгоритм міг їх розпізнати.

Для роботи мурашиного алгоритму необхідний певний набір вхідних даних, представлений у потрібному для нього вигляді. Так, щоб мурашиний алгоритм міг оперувати мережею міст чи портів та їх маршрутами. Йому необхідно їх представити у вигляді звичайного графа. Тому що мурашиний алгоритм працює саме з графами.

Так, для роботи мурашиного алгоритму, для виконання задачі пошуку оптимального маршруту, а саме самого найдешевшого маршруту, на етапі ініціалізації параметрів такі вхідні дані, як список портів, представляються як граф, де порти – це вершини графа.

Список маршрутів, який передається як вхідні дані для мурашиного алгоритму, представляється як набір ребер між вершинами (портами), а вартість фрахту кожного маршруту – як ваги цих ребер. Список початкових та кінцевих портів представляються як початкові та кінцеві вершини для прокладання оптимального маршруту (рис. 2.2).

Після того, як відбулась ініціалізація користувацьких вхідних даних, відбувається ініціалізація вхідних даних для налаштування мурашиного алгоритму пошуку оптимального маршруту вантажоперевезень. Яка включає в себе задання кількості мурах, які будуть проходити маршрути. Вони представляються як число і необхідні для того щоб мурашиний алгоритм міг створити задану кількість агентів, які будуть проходити можливими маршрутами з початкового до кінцевого порту.

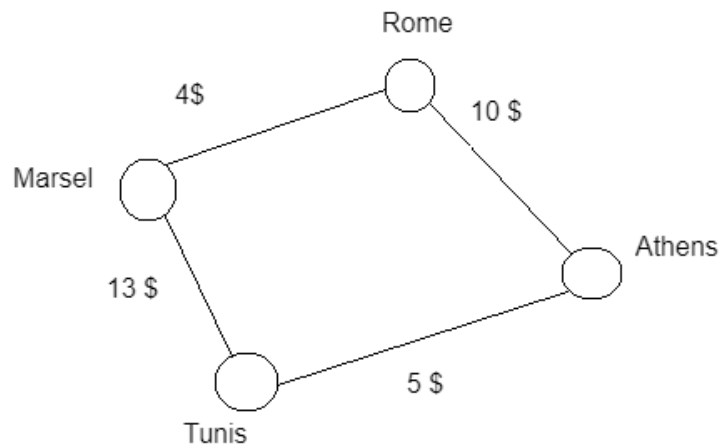


Рисунок 2.2 – Приклад представлення вхідних даних методу пошуку найдешевшого маршруту у вигляді графа, портів – як вершин, маршрутів – як ребер, вартостей фрахту – як ваг ребер

Задання параметрів налаштування мурашиного алгоритму пошуку оптимального маршруту відбувається програмно, для алгоритму налаштовується необхідна кількість мурах, які будуть проходити можливі маршрути.

Також програмно задається кількість найкращих мурах, яка представляється окремим числом, тобто кількість найкращих маршрутів серед всіх знайдених, які будуть далі конкурувати між собою.

Після чого відбувається налаштування такого параметру як кількість ітерацій – числове значення, яке налаштовує кількість проходжень мурашиним алгоритмом процесу пошуку оптимального маршруту.

Після чого налаштовується коефіцієнт випаровування феромонів, який є невід’ємною частиною мурашиного алгоритму, даний коефіцієнт представляється як дробове число від 0 до 1, яке відображає відсоток того, з якою швидкістю кожний окремий маршрут втрачає актуальність.

Далі відбувається налаштування спеціального програмного коефіцієнту впливу феромонів для мурашиного алгоритму пошуку оптимального маршруту, який відіграє ключову роль у процесі побудови необхідного маршруту, феромон є своєрідним маркером, який мурахи залишають на шляху, який інші мурахи використовують для орієнтації при виборі наступного кроку. Значимість

феромону відповідає за те, наскільки сильно феромон впливає на прийняття рішень мурахами при виборі маршруту.

Також налаштовується значення впливу евристичної інформації для мурашиного алгоритму пошуку оптимального маршруту, яке представляється числом та відповідає за визначення мурахами пріоритетності ребер графа (шляху від одного порту до іншого) при побудові маршруту мурахами. Тобто дане значення відповідає за те, наскільки сильно мурахи будуть враховувати інформацію про поточну вартість фрахту маршрутів під час побудови своїх шляхів.

Отже, був розглянутий етап ініціалізації параметрів методу пошуку найдешевшого маршруту, а саме те, які параметри ініціалізуються та яка їх роль у процесі пошуку оптимального маршруту мурашиним алгоритмом.

### 2.1.3 Проходження мурахами маршрутів

Етап проходження мурахами маршрутів включає кілька ключових кроків. Мурахи будують маршрути від початкового до кінцевого порту на основі ймовірнісного вибору, який залежить від феромону та евристичної інформації.

Спершу відбувається ініціалізація мурах, кожна мураха починає свій шлях з порту відправлення, тобто для кожної мурахи  $k$  встановлюється поточний порт  $i$ , та створюється порожній маршрут  $P_k$  до якого буде додаватись послідовність портів від  $i_1$  до  $i_n$  де  $i_1$  – це початковий порт, а  $i_n$  – це кінцевий порт доставки (рис. 2.3).

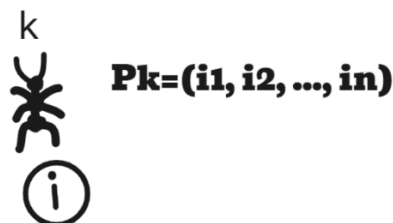


Рисунок 2.3 – Створення мурахи  $k$ , з початковим портом  $i$ , та порожнім маршрутом  $P_k$ , який є впорядкованою послідовністю портів

Після ініціалізації мурах з встановленим початковим портом в мурашиному алгоритмі пошуку оптимального маршруту відбувається процес побудови маршруту кожною мурахою, який виконується за допомогою ймовірнісного вибору кожного наступного порту відносно того в якому мураха знаходиться. Цей процес виконується поки мураха не досягне кінцевого порту, та не сформується маршрут  $P_k$  між початковим та кінцевим портом.

Ймовірність вибору наступного порту для поточного порту в якому знаходиться мураха розраховується за формулою для розрахунку ймовірності вибору наступного порту (2.1).

$$\rho_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in N_i} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}}, \quad (2.1)$$

Де  $P_{ij}$  це ймовірність вибору маршруту з поточного порту  $i$  до сусіднього порту  $j$ . В даній формулі  $\tau_{ij}$  – це показник кількості феромону на маршруті від поточного порту до сусіднього,  $\alpha$  – це параметр що визначає значимість феромону,  $\beta$  – це параметр що визначає значимість евристичної інформації,  $N_i$  – це множина сусідніх портів доступних для вибору з поточного порту  $i$ ,  $\eta_{ij}$  – це показник евристичної інформації для маршруту  $(i, j)$  (2.2).

$$\eta_{ij} = \frac{1}{C_{ij}}, \quad (2.2)$$

Формула розрахунку ймовірності допомагає визначити загальний маршрут мурахи з порту відправлення до порту доставки, що є важливим для роботи мурашиного алгоритму пошуку найдешевшого маршруту.

Побудова маршруту (рис. 2.4) відбувається наступним чином: для кожної мурахи, яка знаходиться в певному порту, обчислюється ймовірність вибору наступного порту  $P_{ij}$ , використовуючи формулу розрахунку ймовірності вибору наступного порту. Далі відбувається вибір наступного для мурахи порту, використовуючи обчислені ймовірності вибору наступного порту, мураха випадковим чином обирає наступний порт  $j$ . Далі обраний порт  $j$  додається до маршруту  $P_k$  мурахи  $k$ . Далі у мурахи відбувається оновлення поточного порту

$i$  на обраний порт  $j$ . Після чого відбувається перевірка чи мураха досягла кінцевого порту, якщо мураха досягла кінцевого порту то побудова маршруту завершується, в іншому випадку процес повторюється.

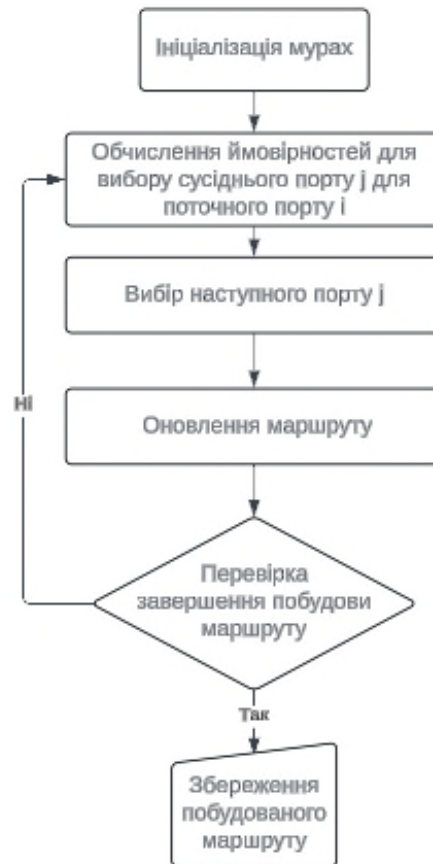


Рисунок 2.4 – Схема роботи процесу побудови маршруту для кожної мурахи

Етап проходження мурахами маршрутів є центральним у мурашиному алгоритмі, де кожна мураха будує свій маршрут на основі ймовірнісного вибору, керованого феромонами та евристичною інформацією. Цей процес дозволяє мурахам досліджувати різні маршрути, що сприяє ефективному пошуку оптимального (або близького до оптимального) рішення. В даному етапі для кожної мурахи формується загальний маршрут проходження від початкового до кінцевого порту доставки.

### 2.1.4 Оновлення феромонів на знайдених маршрутах

Етап оновлення феромонів є одним з ключових етапів мурашиного алгоритму, оскільки він дозволяє алгоритму поступово покращувати рішення шляхом зміни кількості феромонів на маршрутах. Цей процес складається з двох основних під етапів: випаровування феромону та додавання нового феромону. Мураха після проходження маршруту посилює феромоном пройдений маршрут, що дозволяє іншим мурахам частіше ним ходити.

Після проходження маршрутів мурахами, відбувається оновлення феромонів, спершу відбувається випаровування феромону, цей під етап моделює природній процес зменшення кількості феромонів з часом. Це дозволяє алгоритму забути старі неефективні маршрути та зменшити ймовірність їх повторного вибору. Для обрахунку випаровування феромонів існує спеціальна формула (2.3).

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij}, \quad (2.3)$$

В формулі випаровування феромонів описано обрахунок випаровування феромонів, де  $\tau_{ij}$  – це кількість феромону на маршруті (ребрі) між портами  $i$  та  $j$ ,  $\rho$  – це коефіцієнт випаровування феромону ( $0 < \rho < 1$ ). Даний коефіцієнт визначає швидкість випаровування, чим більше його значення тим швидше випаровується феромон з маршруту.

Після випаровування феромону додаються нові феромони на маршрути, які були пройдені мурахами. Це моделює процес зміцнення феромонного сліду на корисних маршрутах, збільшуючи ймовірність їх вибору в наступних ітераціях. Для обрахунку додавання нового феромону існує спеціальна формула (2.4).

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (2.4)$$

В даній формулі  $\tau_{ij}$  – це кількість феромону на маршруті (ребрі) між портами  $i$  та  $j$ ,  $m$  – це кількість мурах, а значення  $\Delta\tau_{ij}^k$  – кількість феромону, який

залишила мураха  $k$  на маршруті (ребрі) між портами  $i$  та  $j$ . Для обрахунку кількості феромону, що залишає кожна мураха  $k$  на маршруті  $(i, j)$  існує спеціальна формула (2.5).

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{C_k} & \text{якщо мураха } k \text{ пройшла через маршрут } (i, j) \\ 0 & \text{інакше} \end{cases} \quad (2.5)$$

В даній формулі обрахунку кількості феромону  $Q$  – це константа, яка визначає загальну кількість феромону, що залишає мураха,  $C_k$  – загальна вартість маршруту, який побудувала мураха  $k$ .

Оновлення феромонів є критичним етапом у мурашиному алгоритмі. Випаровування феромону допомагає уникнути надмірного накопичення феромонів на певних маршрутах і сприяє дослідженню нових рішень. Додавання феромону на основі якості маршрутів забезпечує, що кращі маршрути отримують більше феромону, що підвищує ймовірність їх вибору в майбутніх ітераціях. Цей збалансований підхід дозволяє мурашиному алгоритму ефективно знаходити оптимальні або близькі до оптимальних рішення в задачах комбінаторної оптимізації, а саме в задачі пошуку оптимального маршруту, тобто найдешевшого.

## **2.2 Формулювання задачі для методу планування маршрутів вантажних перевезень**

Задачею для методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом є задача оптимізації, в якій необхідно мінімізувати витрати на перевезення. Для вирішення задачі пошуку найкоротшого маршруту, її можна формалізувати до методів теорії графів та оптимізації. Задачею методу планування маршрутів вантажних перевезень є знаходження оптимального рішення, враховуючи поставлене завдання, а саме пошук найдешевшого маршруту, оптимальним рішенням буде маршрут, вартість

фрахту якого є найдешевшою. Найбільш схоже відображає задачу пошуку найкоротшого маршруту – граф (рис. 2.5). Граф складається з ребер та вершин.

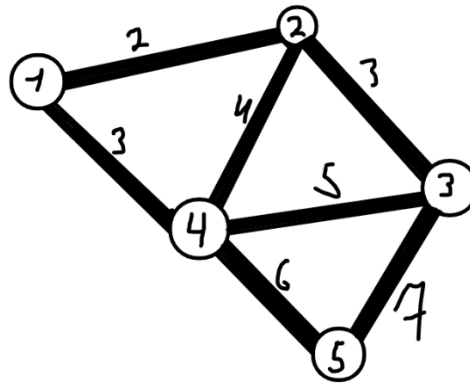


Рисунок 2.5 – Вигляд звичайного графа

Для задачі пошуку найкоротшого маршруту, представимо список портів як набір вершин, а список маршрутів між ними як набір ребер, вартостей фрахту як набір ваг ребер.

Отже задачею методу планування маршрутів вантажних перевезень є пошук найдешевшого маршруту у графі. Вхідні дані методу планування маршрутів вантажних перевезень представляються як графова модель, а саме завдання представлено у вигляді цільової функції:

Графова модель: Представимо мережу портів як граф  $G=(V, E)$ , в якому:

- $V$  – множина вершин (портів)
- $E$  – множина ребер (маршрутів між портами) з вагами, які відповідають вартості фрахту

Цільова функція: цільовою функцією є функція мінімізації загальної вартості перевезення (2.6).

$$\text{Minimize } C = \sum_{(i,j) \in P} C_{ij}, \quad (2.6)$$

Наведено цільову функцію задачі для методу планування маршрутів вантажних перевезень, де  $C$  – вартість фрахту для проходження повним шляхом, а  $C_{ij}$  - це вартість фрахту для проходження маршрутом між сусідніми портами  $i$  та  $j$ , а  $P$  – це вибраний шлях між початковим та кінцевим портом.

Отже було сформульовано задачу для методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом, яка полягає у мінімізації вартості проходження маршруту з початкового до кінцевого порту.

### 2.3 Проектування бази даних

Відповідно до методу планування маршрутів вантажних перевезень з врахуванням фрахту було виділено основні сутності БД, які необхідні для представлення вхідних даних методу планування маршрутів вантажних перевезень (рис. 2.6).

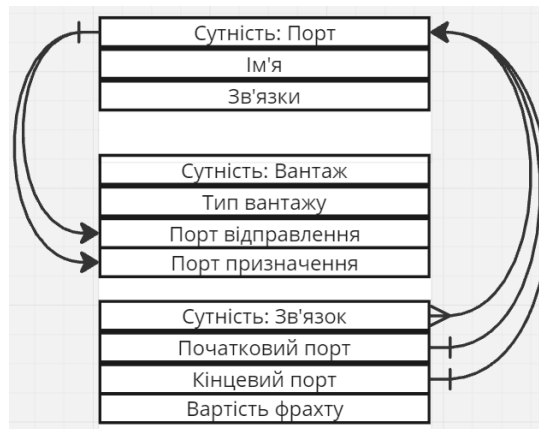


Рисунок 2.6 – Основні сутності методу планування маршрутів вантажних перевезень

Сутність «Порт» містить в собі інформацію про порт, а саме такі атрибути: «Ім'я» та «Зв'язки», атрибут «Ім'я» відповідає за збереження назви порту, атрибут «Зв'язки» описує те, з якими портами зв'язаний конкретний порт, та яка вартість проходження з одного порту до іншого.

Наступна сутність «Вантаж» містить в собі інформацію про вантаж який необхідно перевести, а саме наступні атрибути: «Тип вантажу», «Порт відправлення», «Порт призначення». Атрибут «Тип вантажу» відповідає за збереження інформації про те, який вантаж необхідно перевезти. Атрибут «Порт відправлення» зберігає інформацію про те, з якого порту даний вантаж треба

вивести, а атрибут «Порт призначення» зберігає інформацію про те, в який порт необхідно доставити вантаж.

Сутність «Зв'язок» відповідає за збереження інформації про зв'язок між портами, а саме наступні атрибути: «Початковий порт», «Кінцевий порт», «Вартість фрахту». Атрибут «Початковий порт» відповідає за збереження інформації про порт, який має зв'язок з іншим портом, а сам порт є реалізацією сутності «Порт» з усіма її атрибутами. Атрибут «Кінцевий порт», відповідає за збереження інформації про те, з яким портом існує зв'язок у атрибута «Початковий порт», в атрибуті «Кінцевий порт» зберігається інформація про порт з яким існує з'єднання. Атрибут «Вартість фрахту» відповідає за збереження інформації про вартість проходження маршрутом між початковим та кінцевим портом.

Розглянемо відповідні сутності бази даних для методу планування маршрутів вантажних перевезень у вигляді таблиць.

Таблиця «Port» відображає сутність «Порт» яка представляє дані про порти. Атрибутами даної таблиці є: Name, connections. (Таблиця 2.1).

Таблиця 2.1 – Атрибути таблиці «Port»

Назва	Опис
Name	Назва порту
Connections	Множина зв'язків порту з іншими портами

Таблиця «Connection» відображає сутність «Зв'язок», яка представляє собою інформацію про шляхи між портами та містить в собі такі атрибути: first\_port, end\_port, cost. (Таблиця 2.2). Дана таблиця відображає інформацію, яка зберігається сутністю «Зв'язок», яка необхідна для пошуку оптимального маршруту мурашиним алгоритмом.

Таблиця 2.2 – Атрибути таблиці «Connection»

Назва	Опис
first_port	Початковий порт

end_port	Кінцевий порт
cost	Вартість фрахту проходження від початкового порту до кінцевого порту

Таблиця «Cargo» створена для збереження інформації про вантажі які необхідно перевести та містить в собі такі атрибути: cargo\_type, origin\_port, destination\_port (Таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «Cargo»

Назва	Опис
cargo_type	Тип вантажу який необхідно доставити з порту відправлення до порту прибуття
origin_port	Порт відправлення
destination_port	Порт прибуття

Кожний порт містить в собі інформацію про зв'язки між іншими портами, це дозволяє побудувати необхідні структури та правильно сформулювати завдання для ефективного роботи мурашиного алгоритму пошуку оптимального маршруту. А саме побудувати графову модель мережі портів та відповідних зв'язків між ними, налаштувати порт відправки та порт призначення вантажу, налаштувати дані для роботи алгоритму з списком маршрутів між портами, це дозволить мурашиному алгоритму виконати свою роботу, а саме знайти маршрут доставки вантажу.

В кожній таблиці відображено інформацію про сутності які використовуються в методі планування маршрутів вантажних перевезень. Кожна сутність має свої властивості, які є важливими для вирішення поставленого завдання, а саме пошуку оптимального маршруту, тобто маршруту, вартість фрахту якого є найдешевшою.

Отже було розглянуто основні сутності БД для роботи методу планування маршрутів вантажних перевезень, а саме: «Порт», «Вантаж» та «Зв'язок», з їх атрибутами, функціональними призначеннями та взаємозв'язками. Дані сутності дозволять ефективно працювати мурашиному алгоритму, знаходячи оптимальні рішення доставки вантажів.

## **2.5 Спосіб оцінювання ефективності методу планування маршрутів вантажних перевезень**

Для розуміння ефективності роботи методів які вирішують певні задачі, використовують підходи які базують на порівнянні роботи методу за метриками. Для оцінки ефективності певного методу, що вирішує певну задачу, важливо сформувати певні метрики за якими проводитиметься оцінка. Так наприклад для оцінки методу вирішення квадратного рівняння за дискримінантом, метриками можна взяти: час виконання, час збіжності (тобто те, за скільки ітерацій знаходиться необхідне рішення), якість рішення.

Оцінювання ефективності методу пошуку оптимального маршруту за допомогою мурашиного алгоритму включає декілька ключових аспектів і критеріїв. Важливо підготувати набір вхідних даних задачі якою буде відбуватись оцінювання ефективності методу пошуку оптимального маршруту за мурашиним алгоритмом, оптимальне рішення якої буде відомо завчасно. Для оцінювання ефективності методу пошуку оптимального маршруту можна використовувати такі метрики як: час виконання методу, якість рішення, час збіжності, порівняння з іншими способами.

Метрика «Час виконання» для оцінювання ефективності роботи методу пошуку оптимального маршруту відображає показник часу який витрачається на пошук оптимального маршруту методом. Він визначає, скільки часу потрібно алгоритму для знаходження рішення і впливає на практичну застосовність алгоритму для задач різної складності та розміру. Чим менший показник часу виконання методу, тим краще.

Якість рішення є одним із найважливіших аспектів оцінювання ефективності мурашиного алгоритму. Вона визначає, наскільки добре алгоритм знаходить оптимальні або близькі до оптимальних рішення. Для методу пошуку найдешевшого маршруту, оцінити якість рішення можна за допомогою показника знайденої мінімальної вартості маршруту. Основним критерієм якості рішення є вартість знайденого маршруту. Для задачі пошуку найдешевшого маршруту це означає мінімізацію сумарної вартості переходів між портами. Якість рішення також оцінюється за допомогою відхилення від відомого оптимального рішення або від нижньої межі, якщо оптимальне рішення невідоме. Відхилення обчислюється як різниця між вартістю знайденого рішення та вартістю оптимального рішення (2.7).

$$\text{Відхилення} = \frac{C_{\text{знайдене}} - C_{\text{оптимальне}}}{C_{\text{оптимальне}}} \times 100\% , \quad (2.7)$$

Де Відхилення – показник відхилення знайденого рішення від оптимального,  $C_{\text{знайдене}}$  – вартість фрахту знайденого маршруту,  $C_{\text{оптимальне}}$  – вартість оптимального маршруту. Також для кращої оцінки ефективності методу пошуку оптимального маршруту є сенс знаходити середню якість рішення, яка визначається шляхом запуску методу на однакових вхідних даних кілька разів, та знаходження середнього оптимального рішення задачі.

Збіжність у контексті мурашиного алгоритму означає, наскільки швидко і наскільки стабільно алгоритм наближається до оптимального або достатньо хорошого рішення. Важливою характеристикою збіжності є не тільки те, наскільки швидко алгоритм знаходить рішення, але й наскільки якісні ці рішення. Важливо оцінювати те, як швидко алгоритм знаходить оптимальне рішення, та скільки кроків йому знадобилось щоб до нього наблизитись.

Результативним способом оцінити ефективність методу пошуку оптимального маршруту є порівняння його з існуючими методами пошуку оптимального маршруту.

Отже, був розглянутий та обраний спосіб оцінювання ефективності методу пошуку оптимального маршруту, який полягає у порівнянні його з іншим рішенням за допомогою певних метрик.

## **2.6 Висновки до розділу 2**

Отже, в результаті проектування методу планування маршрутів вантажних перевезень, розроблено метод планування маршрутів вантажних перевезень на основі мурашиного алгоритму, розглянуто хід виконання відповідного мурашиного алгоритму пошуку найдешевшого маршруту та його роботу з вхідними даними. Спроектовано структуру даних для роботи з методом планування маршрутів вантажних перевезень з врахуванням фрахту, виділено сутності та їх взаємодії.

Були розглянуті етапи роботи методу пошуку оптимального маршруту, а саме те, як використовується мурашиний алгоритм для пошуку оптимального маршруту доставки вантажів з одного порту до іншого, були розглянуті такі етапи, як: ініціалізація параметрів алгоритму, проходження мурахами маршрутів, оновлення феромону та відбір найдешевшого маршруту. Було описано те, що в основі роботи мурашиного алгоритму лежить моделювання природніх процесів які відбувається у колонії мурах. Мурашиний алгоритм є ключовою частиною методу пошуку оптимального маршруту, тому що він дозволяє знайти оптимальне, або близьке до оптимального рішення, та найкраще підходить для задачі пошуку оптимального маршруту.

В результаті проведеного проектування методу планування маршрутів вантажних перевезень, можна сформулювати наступні завдання на подальшу роботу: розробити програмну реалізацію методу планування маршрутів вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом, за допомогою якої провести дослідження ефективності роботи методу планування маршрутів вантажних перевезень, та для доведення коректності результатів, окремо функціонально дослідити та протестувати метод.

### **Розділ 3 Програмна реалізація методу планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом**

Для програмної реалізації методу пошуку оптимального маршруту, в основі якого лежить МА, варто використати спеціалізовані програмні компоненти. Розробку методу пошуку оптимального маршруту можна розділити на кілька етапів.

1. Розробка системи отримання вхідних даних від користувача – в даному етапі відбувається програмування методу таким чином, щоб користувач міг завантажити в нього необхідні вхідні дані, для розробки системи отримання методом вхідних даних, буде використовуватись мова програмування Python.

2. Розробка методу пошуку оптимального маршруту, в основі якого лежить мурашиний алгоритм: в даному етапі відбувається програмування необхідного методу пошуку оптимального маршруту, а саме програмна реалізація МА, для цього буде використовуватись мова програмування Python. Так як метод вимагає випадкового вибору числових величин, то для реалізації даного етапу буде використовуватись бібліотека Random, мови програмування Python, яка відповідає за роботу з випадковими величинами.

3. Відображення результатів користувачу – для реалізації цього етапу буде використовуватись мова програмування Python, яка дозволить вивести деталі знайденого рішення у консоль. Також для відображення знайденого оптимального маршруту у вигляді графа, буде використовуватись бібліотека Matplotlib та networkx, дані бібліотеки відповідають за графічні побудови різного роду графів та діаграм.

### **3.1 Програмні бібліотеки та середовище розробки методу планування маршрутів вантажних перевезень**

Для розробки методу планування маршрутів вантажних перевезень обрано мову програмування Python [26]. Дана мова програмування має багато важливих переваг, які дозволяють ефективно реалізувати метод пошуку найдешевшого маршруту, використовуючи необхідні інструменти, які полегшують цей процес. Мова програмування Python є високорівневою мовою програмування. Серед переваг мови програмування Python, можна виділити наступні: простота та читабельність, велика кількість стандартних бібліотек, велика екосистема та спільнота, кросплатформеність, інтерпретованість.

Мова програмування Python є простою та читабельною мовою програмування. Синтаксис мови програмування Python простий і зрозумілий, що робить його доступним для початківців. Це дозволяє новим програмістам швидко почати писати код і розуміти чужий код. Також завдяки чистому і лаконічному синтаксису, код на Python легко читати і розуміти, що сприяє більшій продуктивності та зменшує кількість помилок.

Мова програмування Python має велику кількість вбудованих бібліотек, які дозволяють полегшити роботу з різними видами задач. Python має велику кількість стандартних бібліотек, які містять модулі і пакети для різноманітних задач, включаючи роботу з файлами, випадковими числами, мережевими протоколами, веб-сервісами, тестуванням, криптографією і багатьма іншими. Це внутрішнє середовище вбудованих бібліотек зменшує необхідність в сторонніх бібліотеках і дозволяє швидко реалізувати багато задач.

Мова програмування Python має дуже велику спільноту, яка розробляє нові бібліотеки, підтримує існуючі та допомагає вирішувати проблеми іншим розробникам, які починають користуватись даною мовою програмування, що робить мову програмування Python однією з найбільш популярних мов програмування (рис. 3.1).

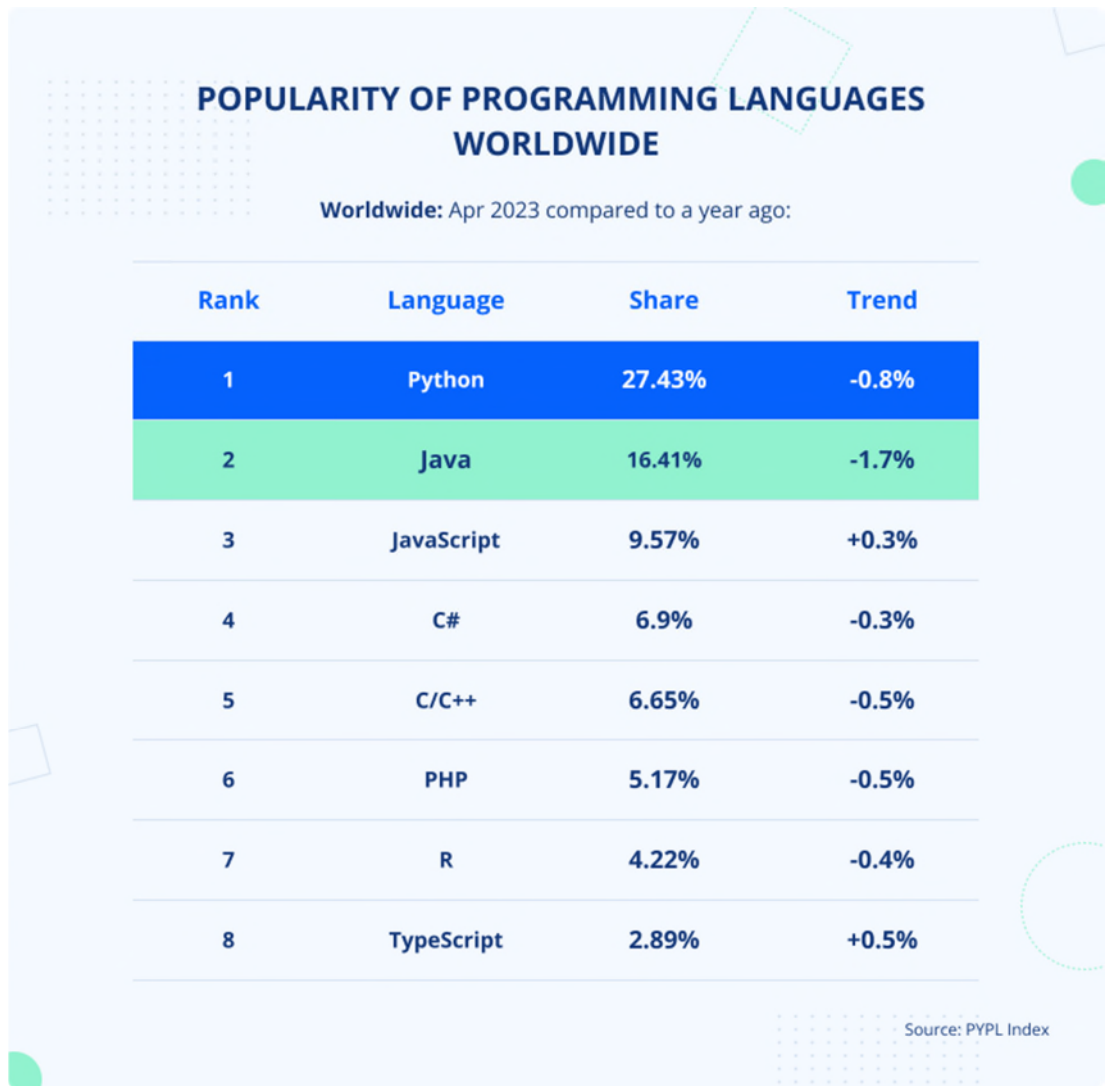


Рисунок 3.1 – Статистика популярності мов програмування у світі [27]

В Python є велика кількість сторонніх бібліотек, які дозволяють виконувати різноманітні задачі, наприклад будувати графи, створювати сайти, працювати з мережами та інших. Python працює на різних операційних системах, включаючи Windows, macOS, Linux, Unix, і навіть на мобільних платформах. Це дозволяє розробникам писати код, який може працювати на будь-якій платформі з мінімальними змінами. Python є інтерпретованою мовою, що означає, що код виконується рядок за рядком. Це полегшує тестування і налагодження, а також забезпечує високу гнучкість під час розробки.

Python є інтерпретованою мовою, що означає, що код виконується рядок за рядком, що дозволяє не витратити час на очікування будівництва програми для

запуску його комп'ютером. Це полегшує тестування і налагодження, а також забезпечує високу гнучкість під час розробки.

Для реалізації відображення знайденого оптимального маршруту у вигляді графу, було обрано бібліотеку `matplotlib`. `Matplotlib` – це потужна бібліотека для візуалізації даних на мові програмування `Python`. Вона надає широкий спектр інструментів для створення різноманітних типів графіків та діаграм. Дана бібліотека дозволить відобразити оптимальний, тобто найдешевший маршрут доставки вантажу у вигляді графу.

Для просунутого налаштування мережі портів у вигляді графів, було обрано бібліотеку для роботи з графами – `networkx`. `NetworkX` – це бібліотека для створення, маніпулювання і вивчення комплексних графів та мереж. Вона особливо корисна для роботи з даними, які можуть бути представлені у вигляді графів, наприклад, соціальних мереж, мереж транспортних систем або комп'ютерних мереж.

Для роботи з випадковими числами, було обрано стандартну бібліотеку `Random`. Дана бібліотека надає набір функцій для генерації випадкових чисел і виконання операцій, які потребують випадковості. Вона корисна для різноманітних задач, таких як симуляції, ігри, статистичний аналіз, і багато інших. Дана бібліотека знадобиться для реалізації мурашиного алгоритму.

Середовищем розробки для програмної реалізації методу пошуку оптимального маршруту, було обрано середовище розробки `PyCharm` компанії `JetBrains` (рис. 3.2). Дане середовище розробки є дуже потужним та зручним для програмування на мові програмування `Python`. В ньому існує велика кількість різноманітних функцій та зручний інтерфейс, що є дуже важливим для ефективної розробки різноманітних систем та програмування програм. Середовище програмування `PyCharm` є одним з найкращих рішень для реалізації програм з багатьма перевагами та стабільною підтримкою.

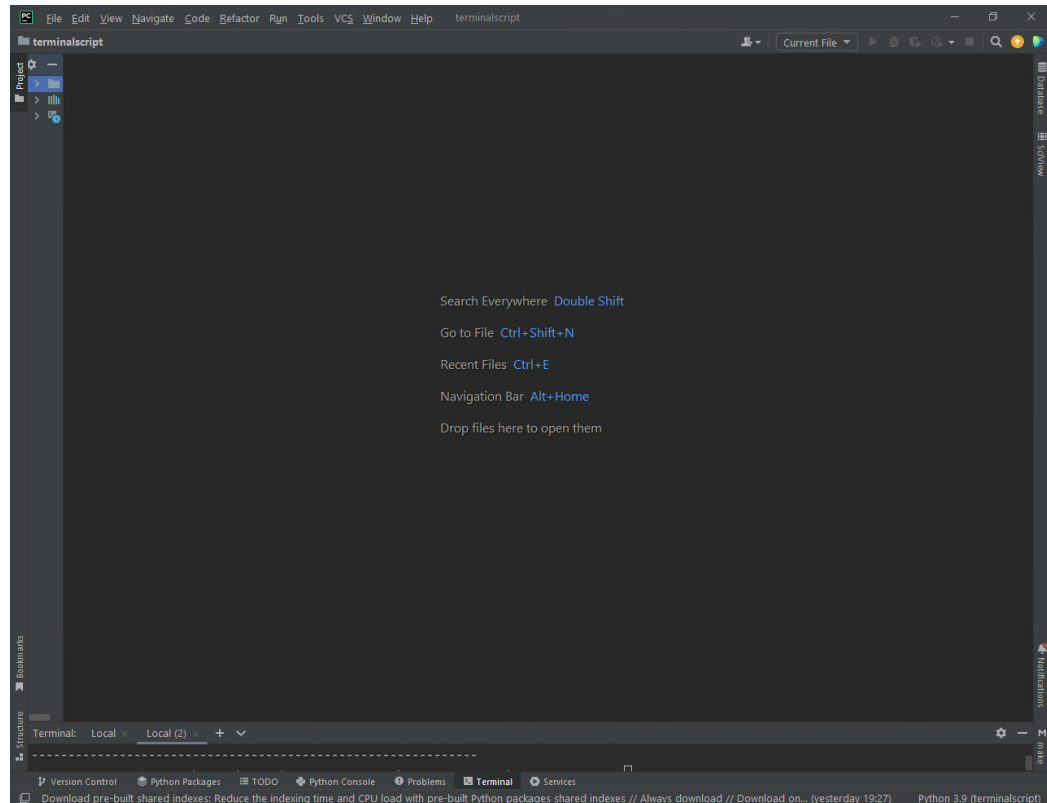


Рисунок 3.2 – Зовнішній вигляд середовища розробки PyCharm

Середовище розробки є ключовим інструментом для програмістів у процесі розробки програмного забезпечення. Воно надає різноманітні засоби для створення, тестування, налагодження та вдосконалення коду. Середовище розробки зазвичай поєднує в собі текстовий редактор, консоль для виконання коду, систему керування версіями, автоматичне доповнення коду та інші корисні інструменти, що сприяють зручності та ефективності роботи. Середовище розробки допомагає організувати проекти, структурувати файли та ресурси, що робить робочий процес більш систематичним та зручним.

Отже, в даному розділі було розглянуто та обрано мову програмування, програмні бібліотеки та середовище розробки, такі як мова програмування Python, яка є зручною та потужною мовою програмування, бібліотеки Matplotlib та NetworkX для побудови графів, стандартну бібліотеку Random для роботи з випадковими числами та середовище розробки PyCharm компанії JetBrains для розробки програм мовою програмування Python. Дані інструменти дозволять ефективно реалізувати метод планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом.

### 3.2 Структура та особливості реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом

Метод планування маршрутів вантажних перевезень був реалізований за допомогою мови програмування Python та необхідних бібліотек, зокрема Matplotlib та Networkx.

Даний метод складається з кількох окремих частин: отримання вхідних даних, робота мурашиного алгоритму, вивід результату, а саме вартості найдешевшого маршруту та відповідного маршруту у вигляді множини портів та відповідного графічного відображення маршруту у вигляді графу (рис. 3.3).

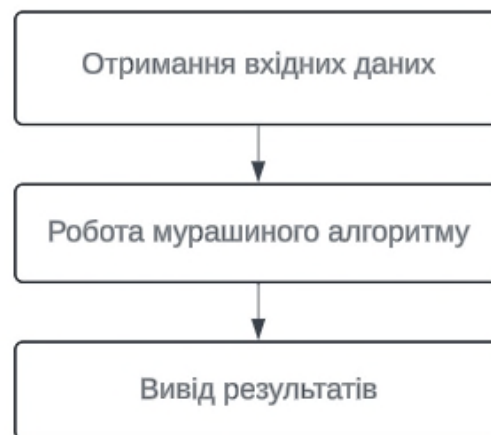


Рисунок 3.3 – Схема реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом

В наведеній схемі показані особливості реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту. Метод складається з певних частин, спочатку методом відбувається отримання вхідних даних, які необхідні для розуміння задачі, після чого запускається мурашиний алгоритм, завданням якого є пошук найдешевшого маршруту, далі метод виводить результати роботи мурашиного алгоритму.

### 3.2.1 Вхідні дані методу планування маршрутів вантажних перевезень

Отримання вхідних даних в методі пошуку найдешевшого маршруту відбувається за допомогою зчитування текстового файлу з необхідними вхідними даними, та створення необхідних об'єктів в самій програмі. Дані в текстовому файлі записуються певним чином (рис. 3.4).

```
Ports:  
Port1,Port2,Port3  
Costs:  
Port1:Port2=3;Port3=1  
Port2:Port3=2;  
Cargos:  
Zerno:Port1:Port3
```

Рисунок 3.4 – Приклад заповнення файлу вхідними даними

В наведеному прикладі заповнення файлу вхідними даними для методу планування маршрутів вантажних перевезень, показано яким чином необхідно передати необхідні методу вхідні дані, а саме: список портів, список шляхів між портами та вартостей фрахту для проходження кожного з них. Для передачі множини портів, необхідно спочатку ввести ключове слово «Ports:», після чого на наступному рядку, необхідно перелічити назви портів через кому, таким чином відбувається передача інформації про існуючі порти для задачі пошуку найдешевшого маршруту, дані порти в подальшому будуть використовуватись для формування графу, та для пошуку необхідних маршрутів.

Після чого в даному прикладі заповнення файлу з вхідними даними, відбувається вказання списку зв'язків між вказаними портами та вартостей фрахту кожного зв'язку. Для введення необхідного списку шляхів, використовується ключове слово «Costs:», яке вказує на те, що далі будуть вказані самі шляхи між портами. Вказання кожного шляху відбувається з нового рядка, спочатку вказується назва порту, між яким існують шляхи, далі за допомогою двокрапки починають перелічуватись зв'язки даного порту з іншими

портами, в яких описується вартість фрахту для кожного з них. Вказання зв'язку відбувається наступним чином: спочатку вказується порт з яким відбувається з'єднання початкового порту, далі ставить знак «=», після якого пишеться числове значення вартості фрахту проходження з'єднанням від початкового до вказаного порту, після чого вказується завершення вказання з'єднання символом «;», що дозволяє вказати кілька з'єднань поточного порту з іншими. Ті з'єднання які були вказані раніше, не потрібно описувати ще раз у поточному порту (рис. 3.5).

```
Ports:
Port1,Port2
Costs:
Port1:Port2=5;
```

Рисунок 3.5 – Приклад вказання зв'язків між двома портами

В даному прикладі, вказаний зв'язок між двома портами, а саме «Port1» та «Port2», при чому так як зв'язок вказаний для «Port1», то немає необхідності вводити його для «Port2».

Наступним вказується саме завдання – список вантажів які потрібно перевести, та порт відправки для кожного вантажу, та відповідно порт призначення для кожного вантажу. Для цього спочатку необхідно вказати ключове слово «Cargos:», воно позначає що на наступних рядках будуть описані завдання для перевезення кожного вантажу. Саме завдання вказується наступним чином: спочатку пишеться тип вантажу який необхідно перевезти, далі за допомогою символу «:» описується порт відправки, після чого за допомогою цього ж символу описується порт призначення (рис. 3.6).

```
Cargos:
Zerno:Port1:Port2
Laptops:Port2:Port1
```

Рисунок 3.6 – Приклад вказання вхідних даних про перевезення вантажу

В даному прикладі відображено, як для методу планування маршрутів вантажних перевезень відбувається вказання завдань для перевезення вантажу різних типів між різними портами. В кожному рядочку окремо вказується тип вантажу, порт відправлення та порт призначення. Це дозволить методу планування маршрутів вантажних перевезень почати пошук найдешевшого маршруту.

Після заповнення вхідними даними текстового файлу, програмно відбувається зчитування вхідного файлу, та створення необхідних програмних об'єктів, для портів та вантажів програмно створюються відповідні об'єкти класів «Port» та «Cargo». Діаграму класів «Port» та «Cargo» відображено на рисунку 3.7.

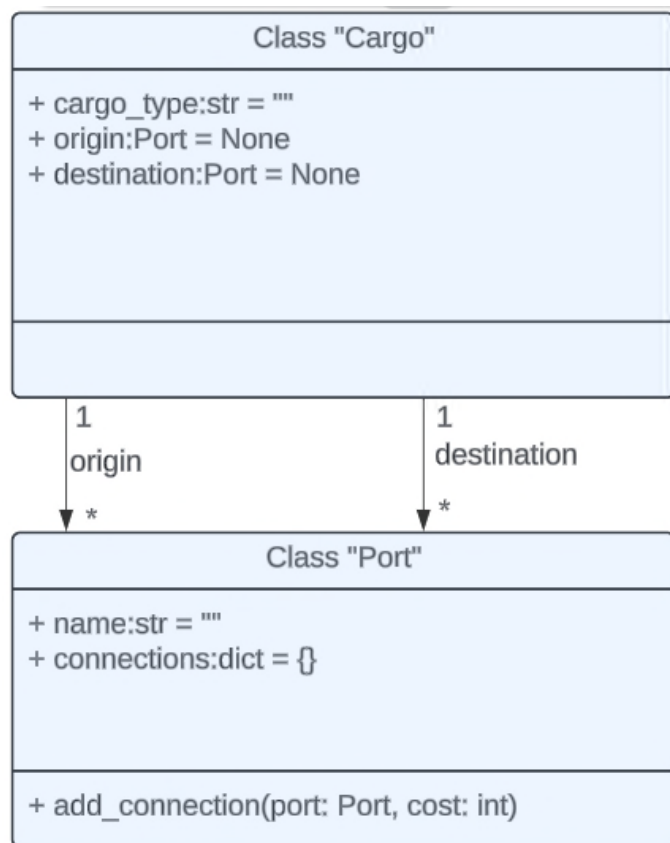


Рисунок 3.7 – Діаграма класів «Cargo» та «Port»

На даній діаграмі класів відображається структура двох класів методу планування маршрутів вантажних перевезень, класу «Cargo» та класу «Port».

Клас «Cargo» відповідає за зберігання інформації про вантаж, в ньому описані властивості, які буде мати кожен вантаж, а саме: cargo\_type, origin,

destination. Властивість `cargo_type` відповідає за збереження інформації про тип вантажу який необхідно перевезти. Властивість `origin` відповідає за порт відправки з якого необхідно перевести вантаж. Властивість `destination` відповідає за порт призначення, куди необхідно перевести вантаж.

Клас «Port» відповідає за зберігання інформації про порт, в даному класі описані властивості, які буде мати кожен порт, а саме: `name`, `connections`. Властивість `name` відповідає за зберігання інформації про ім'я порту. Властивість `connections` відповідає за зберігання інформації про множину з'єднань з іншими портами та вартостями фрахту цих з'єднань. Також даний клас має метод `add_connection`, який відповідає за додавання нового з'єднання з відповідною вартістю до множини з'єднань у конкретного порту.

Отже, в даному розділі було розглянуто отримання вхідних даних методом планування маршрутів вантажних перевезень з врахуванням фрахту. Вхідні дані заповнюються в відповідний текстовий файл певним чином. Спочатку перелічують імена портів, які необхідні для побудови найдешевшого маршруту, та загального графу мережі портів. Далі вказуються відповідні зв'язки між вказаними портами, та вартості фрахту кожного з них, тобто те, за яку вартість можливе проходження відповідними зв'язками, що є важливим для побудови оптимального маршруту. Після чого відбувається вказання відповідного списку завдань, про перевезення вантажів, а саме вказується тип вантажу, порт відправки, тобто звідки вантаж буде відправлятися, та порт призначення вантажу. Після чого метод зчитував відповідний вхідний файл, та створював необхідні програмні об'єкти класів «Port» та «Cargo», які полегшують роботу методу пошуку оптимального маршруту. Далі метод пошуку оптимального маршруту буде використовувати вказані вхідні дані в мурашиному алгоритмі, який буде використовуватись для пошуку найдешевшого маршруту.

### 3.2.2 Робота мурашиного алгоритму пошуку найдешевшого маршруту

Етап роботи мурашиного алгоритму, полягає у пошуку найдешевшого маршруту, на основі отриманих вхідних даних. Мурашиний алгоритм був реалізований у вигляді програмного класу «AntColony». Властивостями даного класу є: `distances`, `ports`, `pheromone`, `all_inds`, `n_ants`, `n_best`, `n_itarations`, `decay`, `alpha`, `beta`. Методами даного класу є: `run`, `spread_pheromone`, `gen_path_dist`, `gen_all_path`, `gen_path`, `pick_move`, діаграма класу «AntColony» зібрана на рисунку 3.8.

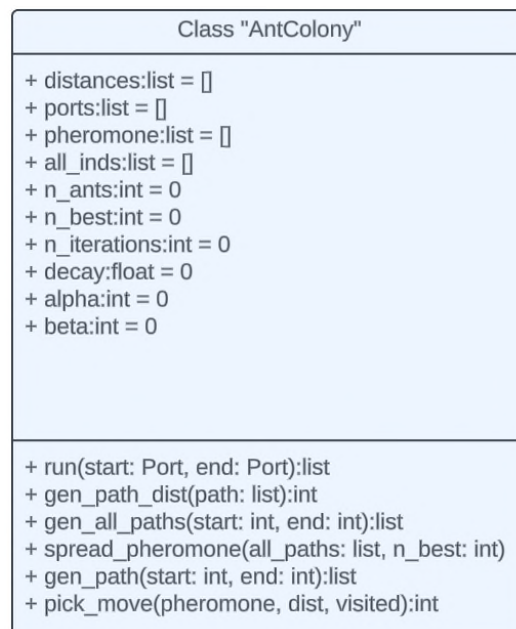


Рисунок 3.8 – Діаграма класу мурашиного алгоритму «AntColony»

Властивість `distances` зберігає інформацію про матрицю вартостей, яка зберігає інформацію про вартість фрахту між портами. Властивість `ports` зберігає список портів, які були отримані як вхідні дані. Властивість `n_ants` зберігає інформацію про параметр налаштування алгоритму, який відповідає за кількість агентів. Властивість `n_best` відповідає за збереження кількості найкращих мурах, на знайдених якими маршрутах буде виділяться більше феромону. Властивість «`all_inds`» відповідає за збереження інформації про номери усіх портів. Властивість «`pheromone`» відповідає за збереження інформації про рівень

феромону на кожному маршруті, початковий рівень феромонів однаковий на кожному маршруті (рис. 3.9).

```
[0.33333333 0.33333333 0.33333333]
[0.33333333 0.33333333 0.33333333]
[0.33333333 0.33333333 0.33333333]
```

Рисунок. 3.9 – Приклад матриці рівня феромонів для маршрутів які з'єднують три порти

Властивість «`n_iterations`» зберігає інформацію про те, яку кількість ітерацій алгоритм має пройти, перед тим як повернути знайдене рішення. Властивість «`decay`» відповідає за збереження інформації про коефіцієнт випаровування феромонів. Властивість «`alpha`» відповідає за збереження інформації про показник впливу феромонів. Властивість «`beta`» відповідає за збереження інформації про показник впливу евристичної інформації.

Метод `run` є ключовим методом мурашиного алгоритму, він відповідає за початок роботи алгоритмом, коли даний метод запускається, мурашиний алгоритм починає процес пошуку оптимального маршруту (рис. 3.10).

Даний метод має два параметри, `start` та `end`, вони вказують з якого порту та до якого порту необхідно перевезти вантаж, та є об'єктами класу `Port`. В даному методі відбувається знаходження індексів даних портів в матриці відстаней `distances` (рис. 3.10), після чого створюється змінна для збереження найкоротшого шляху – `shortest_path`, яка є спочатку пустою та створюється змінна загальної вартості найдешевшого маршруту – `all_time_shortest_path`, в якій зберігається спочатку надвелике значення. Після чого запускається цикл, який повторюється задану кількість ітерацій `n_iterations`, на кожній ітерації якого відбувається пошук мурахами можливих маршрутів від початкового до кінцевого порту, далі на кожному маршруті оновлюються феромони, після чого відбирається найдешевший маршрут, та перевіряється чи він дешевший за той, який вже відомий, якщо це так, то відомий найдешевший маршрут замінюється на знайдений, після чого все повторюється спочатку, поки не вичерпається

задана кількість ітерацій, після чого метод поверне знайдений загальний найдешевший маршрут.

Метод `spread_pheromone` відповідає за оновлення феромону на заданих маршрутах графу, кожний маршрут має певну значимість, яку встановлює даний метод. Метод `gen_path_dist` відповідає за обрахунок загальної вартості фрахту для заданого маршруту. Метод `gen_all_paths` відповідає за генерацію мурахами можливих маршрутів від початкового до кінцевого порту доставки. Метод `gen_path` відповідає за пошук мурахою маршруту від початкового до кінцевого порту доставки. Метод `pick_move` відповідає за ймовірнісний вибір мурахою сусіднього порту.

Ports	Port1	Port2	Port3	Port4	Port5
Port1	$\infty$	100\$	150\$	10\$	$\infty$
Port2	100\$	$\infty$	$\infty$	50\$	$\infty$
Port3	150\$	$\infty$	$\infty$	10\$	$\infty$
Port4	$\infty$	50\$	10\$	$\infty$	20\$
Port5	$\infty$	$\infty$	$\infty$	20\$	$\infty$

Рисунок 3.10 – Приклад матриці відстаней `distances`

Отже в даному розділі була розглянута робота мурашиного алгоритму для виконання задачі пошуку оптимального маршруту доставки вантажу з початкового порту до кінцевого. Були розглянуті програмні особливості реалізації мурашиного алгоритму, зокрема властивості та методи які реалізують мурашиний алгоритм.

### 3.2.2 Вивід результатів роботи методу пошуку найдешевшого маршруту

Вивід результатів роботи методу пошуку оптимального маршруту є як текстове відображення результатів у консолі, так і графічне представлення маршруту у вигляді графу. Перш за все, у консолі виводиться оптимальний маршрут у вигляді послідовності портів, через які проходить маршрут. Це дозволяє користувачу швидко ознайомитися з ключовими етапами маршруту. Разом із переліком портів відображається загальна вартість проходження цим маршрутом, що є критичним показником для оцінки ефективності запропонованого рішення.

Наприклад, після виконання алгоритму в консолі може з'явитися такий результат: "Найдешевший маршрут з Antwerp до Токуо: Antwerp -> Hamburg -> Shanghai -> Токуо; Загальна вартість знайденого маршруту: 23.0". Цей формат є інтуїтивно зрозумілим і дозволяє користувачеві легко оцінити, які порти включені в маршрут і скільки коштує його проходження.

Окрім текстового виводу, метод також генерує графічне представлення маршруту у вигляді графу. Графічна візуалізація відображає порти як вузли, а шляхи між ними як ребра, що дозволяє наочно побачити структуру маршруту. Це особливо корисно для складних мереж з багатьма можливими маршрутами, оскільки візуалізація допомагає краще зрозуміти топологію мережі і взаємозв'язки між портами. Наприклад, вузли можуть бути позначені іменами портів, а ребра можуть бути підписані вартістю переходу між портами. Графічне представлення може бути згенероване з використанням таких бібліотек як Matplotlib або NetworkX (рис. 3.11).

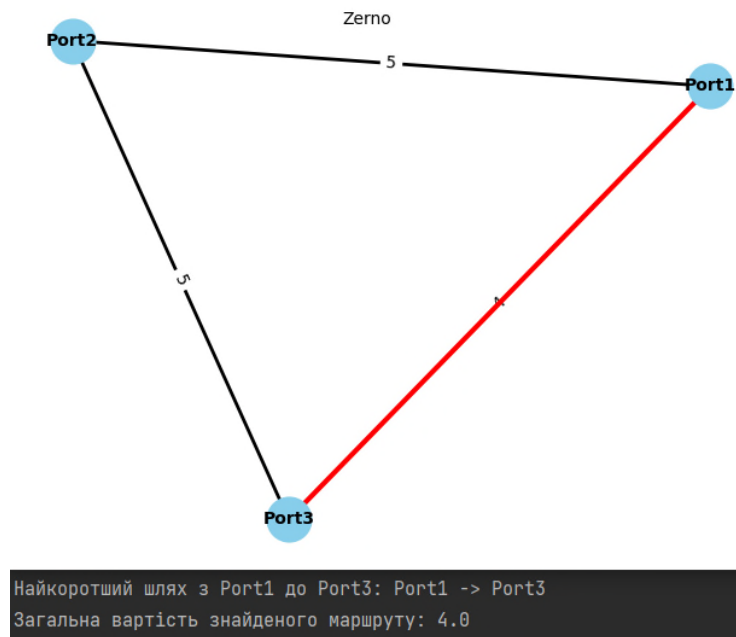


Рисунок 3.11 – Приклад виводу результатів роботи методу планування маршрутів вантажних перевезень з врахуванням фрахту

Загальна діаграма класів методу планування маршрутів вантажних перевезень, відображена на рисунку 3.12.

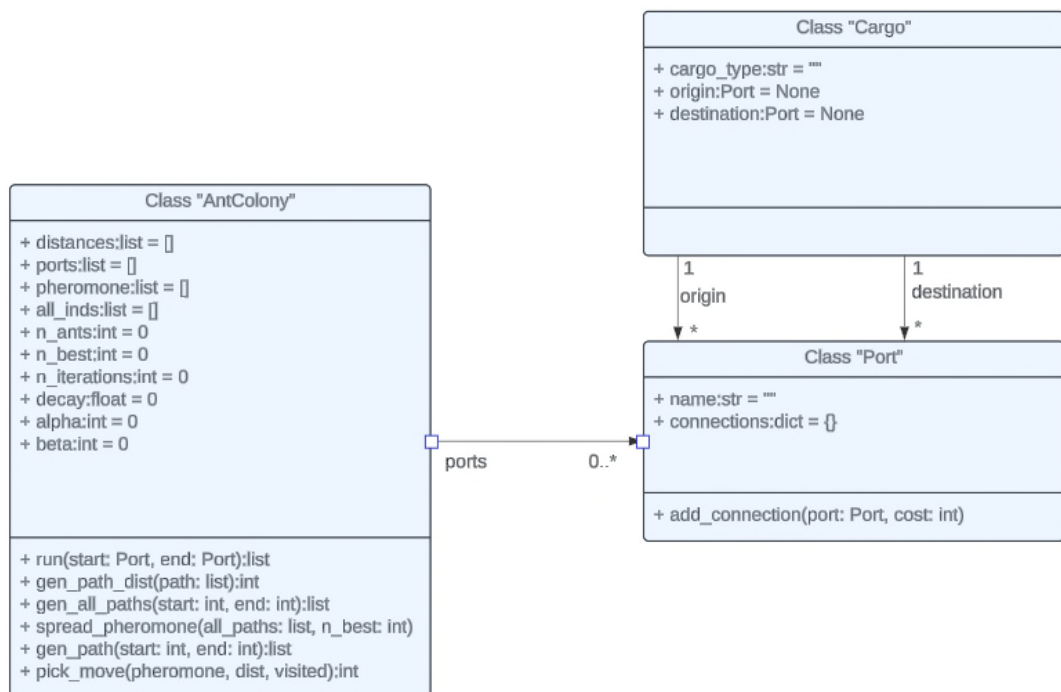


Рисунок 3.12 – Загальна діаграма класів методу формування вантажоперевезень з врахуванням фрахту

В наведеній загальній діаграмі класів методу планування маршрутів вантажних перевезень з врахуванням фрахту, описано класи які необхідні для роботи методу планування маршрутів вантажних перевезень, а саме клас «Cargo», «Port» та «AntColony». Дані властивості та методи необхідні для повноцінної роботи методу. В їх основі лежить мурашиний алгоритм, який дозволить знайти оптимальне рішення, а саме найдешевший маршрут розвозу вантажів.

Цей двоаспектний підхід до відображення результатів дозволяє користувачеві не лише отримати чітке і зручне текстове резюме маршруту і його вартості, але й візуально оцінити маршрут у контексті всієї мережі портів. Такий підхід забезпечує більш глибоке розуміння отриманих результатів і сприяє прийняттю обґрунтованих рішень на основі візуалізованих даних.

### **3.3 Підготовка робочих вхідних даних для методу планування маршрутів вантажних перевезень**

Для нормальної роботи та якісної оцінки ефективності методу планування маршрутів вантажних перевезень важливо сформувати тестовий набір вхідних даних. Для методу планування маршрутів вантажних перевезень необхідні такі дані:

- список портів;
- список можливих маршрутів між портами вартостей фрахту для кожного з них;
- список типів вантажів та початкового та кінцевого порту доставки для кожного з них;

Щоб сформувати список назв портів, було використано сервіс ChatGPT[28]. Для тестування методу, створено запит на формування списку з тридцяти випадкових назв портів у необхідному для роботи методу вигляді(Рисунок 3.13).

Звичайно! Ось згенеровані вхідні дані для 30 портів.

```

CSS
Copy code

Ports:
Hamburg, Rotterdam, Antwerp, Shanghai, Singapore, Busan, Qingdao, Ningbo, Guangzhou, Los Angeles, Ne

```

Рисунок 3.13 – Згенерований сервісом ChatGPT список з тридцяти випадкових назв портів

Сервіс ChatGPT згенерував необхідний список назв портів для методу планування маршрутів вантажних перевезень, що дозволить якісно протестувати метод, ти дослідити його ефективність.

Для формування списку можливих маршрутів між створеними портами, та вартостей фрахту для кожного з них, був створений відповідний запит до сервісу ChatGPT (рис. 3.14).

```

Costs:
Hamburg: Rotterdam=8; Antwerp=7; Shanghai=12
Rotterdam: Antwerp=2; Singapore=15; New York=14
Antwerp: Shanghai=14; Singapore=11; Busan=18
Shanghai: Singapore=7; Busan=5; Tokyo=4
Singapore: Busan=4; Qingdao=8; Ningbo=6
Busan: Qingdao=3; Ningbo=2; Osaka=9
Qingdao: Ningbo=1; Guangzhou=8; Manila=7
Ningbo: Guangzhou=6; Los Angeles=11; Melbourne=12
Guangzhou: Los Angeles=10; San Francisco=9; Jakarta=13
Los Angeles: New York=15; San Francisco=3; Sydney=18
New York: San Francisco=7; Tokyo=16; Miami=10
San Francisco: Tokyo=12; Osaka=5; Bangkok=14
Tokyo: Osaka=4; Manila=8; Durban=20
Osaka: Manila=6; Sydney=10; Cape Town=22
Manila: Sydney=9; Melbourne=7; Rio de Janeiro=19
Sydney: Melbourne=6; Jakarta=8; Buenos Aires=25
Melbourne: Jakarta=6; Bangkok=14; Lima=21
Jakarta: Bangkok=8; Hong Kong=11; Valparaiso=26
Bangkok: Hong Kong=7; Durban=16; Panama=28
Hong Kong: Durban=15; Cape Town=17; Miami=12
Durban: Cape Town=5; Rio de Janeiro=23; Vancouver=24
Cape Town: Rio de Janeiro=18; Buenos Aires=21; Montreal=27
Rio de Janeiro: Buenos Aires=9; Lima=14; Hamburg=26
Buenos Aires: Lima=7; Valparaiso=19; Rotterdam=28
Lima: Valparaiso=6; Panama=15; Antwerp=22
Valparaiso: Panama=9; Miami=17; Shanghai=24
Panama: Miami=7; Vancouver=26; Singapore=21
Miami: Vancouver=18; Montreal=14; Busan=22
Vancouver: Montreal=9; Qingdao=16; Ningbo=23
Montreal: Hamburg=12; Tokyo=15; Guangzhou=28

```

Рисунок 3.14 – Згенеровані відповідним чином вхідні дані сервісом ChatGPT

Також був заданий у відповідному форматі вантаж, який необхідно перевезти з одного порту до іншого, для цього спочатку був вказаний відповідний тип вантажу, а далі початковий порт та кінцевий порт доставки вантажу (рис. 3.15). Типом вантажу вказане «Grain», тобто зерно, портом відправлення вказаний порт «Hamburg», а портом призначення порт «Ningbo».

```
Cargos:
Grain:Hamburg:Ningbo
```

Рисунок 3.15 – Вказаний вантаж для перевезення з одного порту в інший

В результаті був сформований файл з вхідними даними, який складається з списку портів, списку маршрутів між вказаними портами з відповідними вартостями фрахту для кожного з маршруту. Зрозумілим для методу форматом. (рис. 3.16)

```
Ports:
Hamburg, Rotterdam, Antwerp, Shanghai, Singapore, Busan, Qingdao,
Costs:
Hamburg:Rotterdam=8;Antwerp=7;Shanghai=12
Rotterdam:Antwerp=2;Singapore=15;New York=14
Antwerp:Shanghai=14;Singapore=11;Busan=18
Shanghai:Singapore=7;Busan=5;Tokyo=4
Singapore:Busan=4;Qingdao=8;Ningbo=6
Busan:Qingdao=3;Ningbo=2;Osaka=9
Qingdao:Ningbo=2;Guangzhou=8;Manila=7
Ningbo:Guangzhou=6;Los Angeles=11;Melbourne=12
Guangzhou:Los Angeles=10;San Francisco=9;Jakarta=13
Los Angeles:New York=15;San Francisco=3;Sydney=18
New York:San Francisco=7;Tokyo=16;Miami=10
San Francisco:Tokyo=12;Osaka=5;Bangkok=14
Tokyo:Osaka=4;Manila=8;Durban=20
Osaka:Manila=6;Sydney=10;Cape Town=22
Manila:Sydney=9;Melbourne=7;Rio de Janeiro=19
Sydney:Melbourne=5;Jakarta=8;Buenos Aires=25
Melbourne:Jakarta=6;Bangkok=14;Lima=21
Jakarta:Bangkok=8;Hong Kong=11;Valparaiso=26
Bangkok:Hong Kong=7;Durban=16;Panama=20
Hong Kong:Durban=15;Cape Town=17;Miami=12
Durban:Cape Town=5;Rio de Janeiro=23;Vancouver=24
Cape Town:Rio de Janeiro=18;Buenos Aires=21;Montreal=27
Rio de Janeiro:Buenos Aires=8;Lima=14;Hamburg=26
Buenos Aires:Lima=7;Valparaiso=19;Rotterdam=20
Lima:Valparaiso=6;Panama=15;Antwerp=22
Valparaiso:Panama=9;Miami=17;Shanghai=24
Panama:Miami=7;Vancouver=20;Singapore=21
Miami:Vancouver=18;Montreal=14;Busan=22
Vancouver:Montreal=9;Qingdao=16;Ningbo=23
Montreal:Hamburg=12;Tokyo=15;Guangzhou=28
Cargos:
Grain:Hamburg:Ningbo
```

Рисунок 3.16 – Сформований набір вхідних даних для методу планування маршрутів вантажних перевезень з врахуванням фрахту

Сформований набір вхідних даних, містить в собі список з тридцяти назв портів, також список шляхів між портами та вартостями фрахту для проходження кожного з них. Даний набір вхідних даних зберігається у текстовому файлі, який в подальшому буде використовувати метод планування маршрутів вантажних перевезень

Отже, були підготовані робочі вхідні дані для методу планування маршрутів вантажних перевезень з врахуванням фрахту, які мають в своїй основі логічне обґрунтування та результат роботи методу з використанням даних вхідних даних можна рахувати правдивими.

### **3.4 Тестування методу планування маршрутів вантажних перевезень з врахуванням фрахту**

Для проведення тестування розробленого методу планування вантажних перевезень, були обрані наступні критерії – кількість ітерацій які алгоритму необхідні для пошуку оптимального рішення, час виконання алгоритму. А також порівняння роботи методу з іншим способом знайдення рішення.

Порівняння мурашиного алгоритму методу планування маршрутів вантажних перевезень з врахуванням фрахту, буде відбуватись з алгоритмом Дейкстри. Для тестування методу, були взяті вхідні дані, описані в розділі 3.3. Однакові вхідні дані є важливою умовою для проведення якісного тестування методу, тобто такого тестування, при якому ефективність методу не підлягає сумнівам.

Розпочнемо з поставлення задачі методу планування маршрутів вантажних перевезень за мурашиним алгоритмом та алгоритмом Дейкстри. Алгоритмам необхідно знайти оптимальний, тобто найдешевший маршрут з порту «Hamburg» до порту «Ningbo». Відомим оптимальним рішенням, відносно якого буде відбуватись обрахунок відхилення є вартість маршруту – 19 одиниць.

Після знаходження рішення кожним алгоритмом, для кожного знайденого рішення відбудеться обрахунок кількості ітерацій які знадобились кожному алгоритму для пошуку оптимального рішення. Для мурашиного алгоритму відбудеться порівняння знайденого рішення з оптимальним, тобто відомої вартості найдешевшого маршруту, яка дорівнює 19 одиниць. Це дозволить провести дослідження ефективності методу планування маршрутів вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом.

Перед початком тестування методів на основних вхідних даних, вважається важливим спочатку протестувати їх на більш простих вхідних даних. Це має на меті дозволити нам отримати загальне уявлення про ефективність та працездатність методів у більш контрольованому середовищі перед їх застосуванням до складніших даних.

Розглянемо результати роботи мурашиного алгоритму та алгоритму перебору на вхідних даних з низькою складністю, які складаються з 4 портів, задачею є пошук оптимального маршруту з «Port1» до «Port3».

Результати роботи мурашиного алгоритму відображено у вигляді графу (рис. 3.17) та в консолі (рис. 3.18).

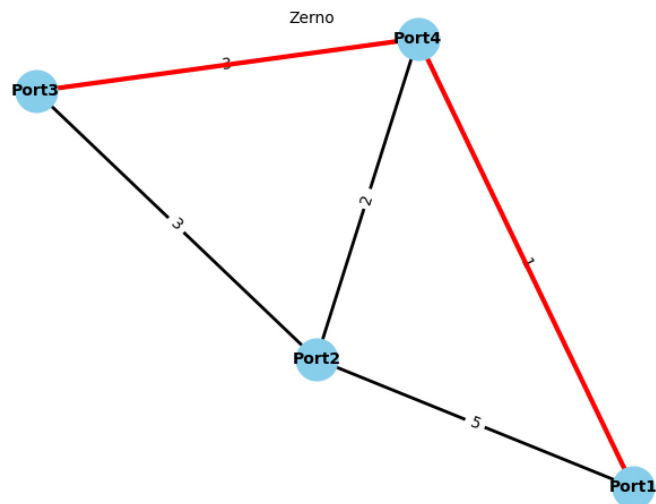


Рисунок 3.17 – Відображення знайденого оптимального маршруту мурашиним алгоритмом у вигляді графу

```

Кількість ітерацій для мурашиного алгоритму: 2
Кількість часу мурашиного алгоритму: 0.0069963932037353516
Найкоротший шлях з Port1 до Port3: Port1 -> Port4 -> Port3
Загальна вартість знайденого маршруту: 4.0

```

Рисунок 3.18 – Відображення в консолі знайденого оптимального маршруту мурашиним алгоритмом

Результатом роботи мурашиного алгоритму є знайдений оптимальний, тобто найдешевший маршрут, вартість якого дорівнює 4 одиниці, алгоритму знадобилось 2 ітерації та 0.007 секунди.

Розглянемо роботу алгоритму Дейкстри для пошуку оптимального рішення в поставленій задачі. Результати алгоритму Дейкстри були виведені в консоль (рис. 3.19).

```

Найкоротший шлях від порту Port1 до порту Port3 має довжину 4
Шлях: Port1 -> Port4 -> Port3
Часу витрачено алгоритмом Дейкстри: 0.0010018348693847656 секунд
Кількість ітерацій алгоритму Дейкстри: 4

```

Рисунок 3.19 – Відображення в консолі знайденого оптимального маршруту алгоритмом Дейкстри

Результатом роботи алгоритму Дейкстри є знайдений оптимальний маршрут, вартість якого дорівнює 4 одиниці, алгоритму знадобилось 4 ітерації на пошук оптимального маршруту, також алгоритм витратив 0.01 секунду часу.

З результатів роботи мурашиного алгоритму та алгоритму Дейкстри, можна зробити висновок що на вхідних даних низької складності, обидва алгоритми працюють коректно та знаходять оптимальний маршрут. З кожною ітерацією знайдене рішення наближається до оптимального (рис. 3.20).

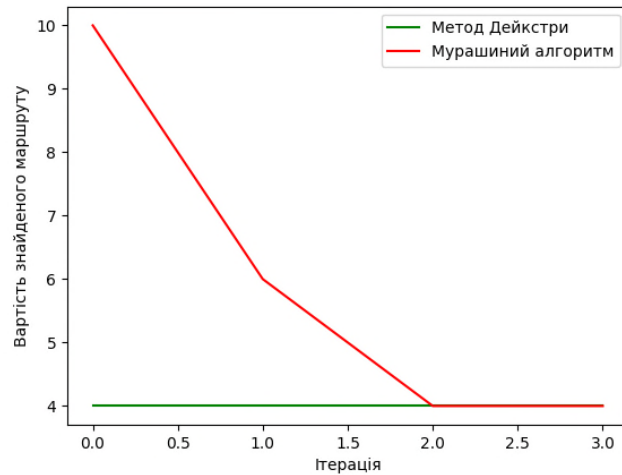


Рисунок 3.20 – Відображення знайдених рішень алгоритмами на кожній ітерації

Розглянемо результати роботи мурашиного алгоритму на основних вхідних даних, які відображені у вигляді графу на рисунку 3.21 та в консолі на рисунку 3.22.

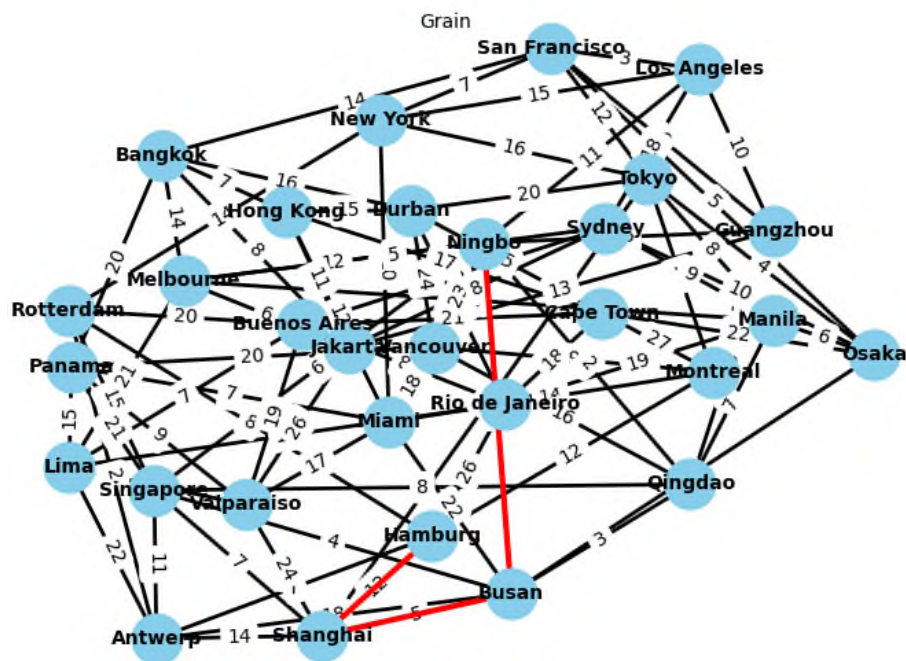


Рисунок 3.21 – Відображення знайденого найдешевшого маршруту мурашиним алгоритмом у вигляді графу

```

Кількість ітерацій для мурашиного алгоритму: 12
Кількість часу мурашиного алгоритму: 0.08894729614257812
Найкоротший шлях з Hamburg до Ningbo: Hamburg -> Shanghai -> Busan -> Ningbo
Загальна вартість знайденого маршруту: 19.0

```

Рисунок 3.22 – Відображення в консолі знайденого найдешевшого маршруту мурашиним алгоритмом

Результатом роботи мурашиного алгоритму є знайдений оптимальний, тобто найдешевший маршрут доставки вантажу. Алгоритму знадобилось 12 ітерацій на пошук рішення та 0.0889 секунди часу, загальна вартість фрахту знайденого маршруту дорівнює 19 одиниць. Метод відображає не тільки інформацію в консоль, але також дозволяє графічно побачити знайдений найдешевший маршрут у вигляді графу. Знайдене рішення мурашиним алгоритмом відповідає відомому оптимальному рішення, тобто маршруту з вартістю фрахту 19 одиниць.

Розглянемо результати роботи алгоритму Дейкстри на основних вхідних даних, відображені на рисунку 3.23.

```

Найкоротший шлях від порту Hamburg до порту Ningbo має довжину 19
Шлях: Hamburg -> Shanghai -> Busan -> Ningbo
Часу витрачено алгоритмом Дейкстри: 0.10099825859069825 секунд
Кількість ітерацій алгоритму Дейкстри: 30

```

Рисунок 3.23 – Результати роботи алгоритму Дейкстри на основних вхідних даних

Результатом роботи алгоритму Дейкстри на основних вхідних даних, є знайдений оптимальний маршрут, вартість фрахту якого дорівнює 19 одиниць. Алгоритму Дейкстри знадобилось 0.1 секунди часу для пошуку оптимального маршруту та 30 ітерацій. Алгоритм Дейкстри знайшов оптимальне рішення.

Результатом тестування мурашиного алгоритму методу планування маршрутів вантажних перевезень з врахуванням фрахту та алгоритму Дейкстри, в задачі пошуку оптимального, тобто найдешевшого маршруту на заданих

вхідних даних є вихідні дані, які включають в себе час виконання та кількість ітерацій. Кількість ітерацій мурашиного алгоритму дорівнює 12, час витрачений мурашиним алгоритмом дорівнює 0.0889 секунд. Алгоритм Дейкстри витратив 0.1 секунду часу та 30 ітерацій на пошук оптимального рішення (таблиця 3.1).

Таблиця 3.1 – Результати тестування методу планування маршрутів вантажних перевезень за мурашиним алгоритмом в порівнянні з алгоритмом перебору.

Критерій тестування	Мурашиний алгоритм	Алгоритм Дейкстри
Кількість ітерацій	12	30
Час виконання	0.0889	0.1

З результатів тестування видно що мурашиний алгоритм знайшов рішення на 0.011 секунди швидше за алгоритм Дейкстри, тобто на 11.1%, кількість ітерацій для пошуку оптимального рішення у алгоритму Дейкстри та мурашиного алгоритму різні. Мурашиному алгоритму знадобилось менше ітерацій (рис. 3.24).

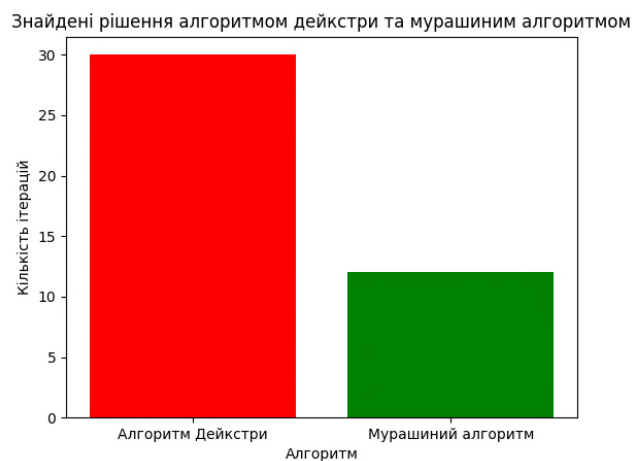


Рисунок 3.24 – Кількість ітерацій, що знадобилась алгоритмам для пошуку оптимального маршруту

Мурашиному алгоритму знадобилось 12 ітерацій для пошуку оптимального рішення (рис. 3.25).

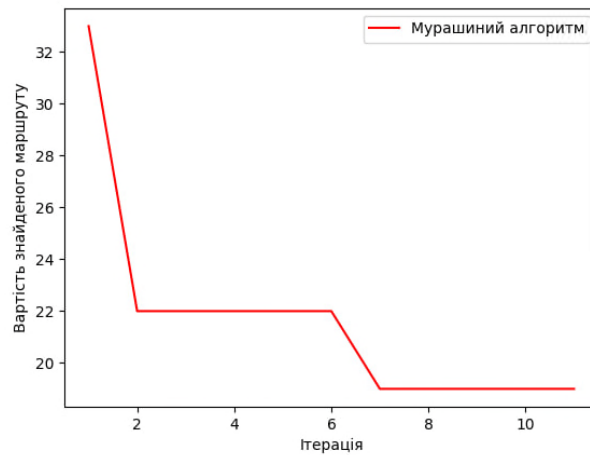


Рисунок 3.25 – Графік знайденого оптимального рішення на кожній ітерації мурашиним алгоритмом.

Отже, проведено тестування методу планування вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом, а саме порівняння його результатів з результатами роботи алгоритму Дейкстри та оптимальним рішенням для поставленої задачі за наступними критеріями: кількість ітерацій, час виконання. З результатів тестування видно що метод підвищив ефективність пошуку оптимального рішення, тобто найдешевшого маршруту на 11.1%.

### 3.5. Оцінка ефективності запропонованого методу

Щоб оцінити ефективність методу планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом, було проведено тестування методу на підготовлених вхідних даних. Відоме оптимальне рішення на вхідних даних, на яких відбувалось тестування методу, дорівнює – 19 одиниць.

Методу знадобилось 12 ітерацій для пошуку оптимального рішення на підготовлених вхідних даних, а алгоритму Дейкстри 30 ітерацій, тобто методу планування маршрутів вантажних перевезень необхідна менша кількість ітерацій для пошуку оптимального рішення ніж алгоритму перебору.

Методу знадобилось 0.0889 секунди часу для пошуку оптимального рішення на підготовлених вхідних даних, в той час як алгоритму Дейкстри для цього знадобилось 0.1 секунди, метод планування маршрутів вантажних перевезень з врахуванням фрахту знайшов рішення швидше.

Оптимальне рішення, яке знайшов метод пройшовши необхідну кількість ітерацій дорівнює 19 одиниць. З результату обчислення за формулою обчислення відхилення, видно що відхилення знайденого методом рішення від оптимального на підготовлених вхідних даних дорівнює 0, що свідчить про коректність роботи методу. Враховуючи те, що методу знадобилось менше часу на пошук оптимального рішення, а саме на 0.0111 секунди, підвищення ефективності планування маршрутів вантажних перевезень відбулося на 11.1%.

Отже, в результаті оцінки ефективності запропонованого методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом, а саме обчислення метрики відхилення від оптимального рішення, часу виконання методу для пошуку оптимального рішення та порівняння результатів роботи методу з алгоритмом Дейкстри, метод підвищив ефективність планування маршрутів вантажних перевезень на 11.1%.

### **3.6 Висновки до розділу 3**

Отже, в результаті програмної реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту, було розглянуто та обрано мову програмування Python, програмні бібліотеки Matplotlib та NetworkX для побудови графів стандартну бібліотеку Random для роботи з випадковими числами та середовище розробки PyCharm.

Після чого було розглянуто структуру та особливості реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту, метод складається з кількох окремих частин: отримання вхідних даних, робота

мурашиного алгоритму, вивід результату, а саме вартості найдешевшого маршруту.

Далі було підготовлено робочі вхідні дані для методу планування маршрутів вантажних перевезень, які є необхідними для ефективного тестування та оцінки ефективності методу. Робочі вхідні дані були отримані за допомогою сервісу ChatGPT.

Після чого було виконано тестування методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом, були сформовані критерії тестування, а саме: відхилення, кількість ітерацій та час виконання. Також було проведено порівняння роботи мурашиного алгоритму методу планування маршрутів вантажних перевезень з алгоритмом Дейкстри.

В результаті проведено тестування методу планування маршрутів вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом, а саме порівняння його результатів з результатами роботи алгоритму Дейкстри та оптимальним рішенням для поставленої задачі за наступним критеріями: кількість ітерацій, час виконання. З результатів тестування видно що метод підвищив ефективність планування маршрутів на 11.1%.

## Загальні висновки

Метою кваліфікаційної роботи бакалавра є підвищення ефективності процесу планування маршрутів вантажних перевезень з урахуванням вартості фрахту, спрямованого на зменшення витрат на транспортування вантажів.

Для досягнення поставленої мети реалізовано виконання наступних задач:

- проведено аналіз еволюційних алгоритмів для вирішення задачі формування перевезень з урахуванням фрахту;
- розроблено метод планування маршрутів вантажних перевезень за критерієм зменшення витрат на транспортування вантажів;
- розроблено програмну реалізацію методу планування маршрутів вантажних перевезень з урахуванням фрахту;
- проведено тестування програмної реалізації методу на різних тестових наборах даних та оцінено підвищення ефективності процесу планування маршрутів.

Для виконання поставленої задачі, був проведений аналіз предметної області, розглянуті основні поняття та проблеми, які характерні для вантажоперевезень з врахуванням фрахту. Було проаналізовано еволюційні алгоритми до вирішення задачі планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом, а саме: звичайні генетичні алгоритми, мурашині алгоритми, множинні генетичні алгоритми. Проаналізувавши еволюційні алгоритми які використовуються для пошуку оптимального маршруту (найдешевшого), а саме генетичних алгоритмів, множинних генетичних алгоритмів та мурашиних алгоритмів, було зроблено висновок що для вирішення задачі планування маршрутів вантажних перевезень з врахуванням фрахту, мурашині алгоритми є найкращим рішенням.

Проаналізувавши існуючі програмні рішення, було виявлено що існує потреба в розробці системи автоматизованого планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом. В результаті було

сформовано мету, задачі та вимоги для реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту.

Далі відбулось проектування методу планування маршрутів вантажних перевезень за еволюційним алгоритмом. Розроблено метод планування маршрутів вантажних перевезень на основі мурашиного алгоритму, розглянуто хід виконання відповідного мурашиного алгоритму пошуку найдешевшого маршруту та його роботу з вхідними даними. Спроектовано базу даних для роботи з методом планування маршрутів вантажних перевезень з врахуванням фрахту, виділено сутності та їх взаємодії. Був розглянутий та обраний спосіб оцінювання ефективності методу пошуку оптимального маршруту, який полягає у порівнянні його з іншим рішенням за допомогою певних метрик.

Наступним, було програмно реалізовано методи планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом. Обрано необхідні програмні бібліотеки, мову програмування та середовище розробки. Розроблено структуру та особливості реалізації методу планування маршрутів вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом, було розглянуто як відбувається отримання вхідних даних, робота з ними та вивід результатів.

Далі було підготовлено робочі вхідні дані методу планування маршрутів вантажних перевезень з врахуванням фрахту за допомогою сервісу ChatGPT, було сформовано основні вхідні дані.

Далі було проведено тестування та проведено оцінку ефективності методу планування маршрутів вантажних перевезень з врахуванням фрахту за мурашиним алгоритмом, а саме порівняння його результатів з результатами роботи алгоритму Дейкстри та оптимальним рішенням для поставленої задачі за наступним критеріями: кількість ітерацій, час виконання, відхилення від оптимального рішення. З результатів тестування видно що метод підвищив ефективність процесу планування маршрутів на 11.1%.

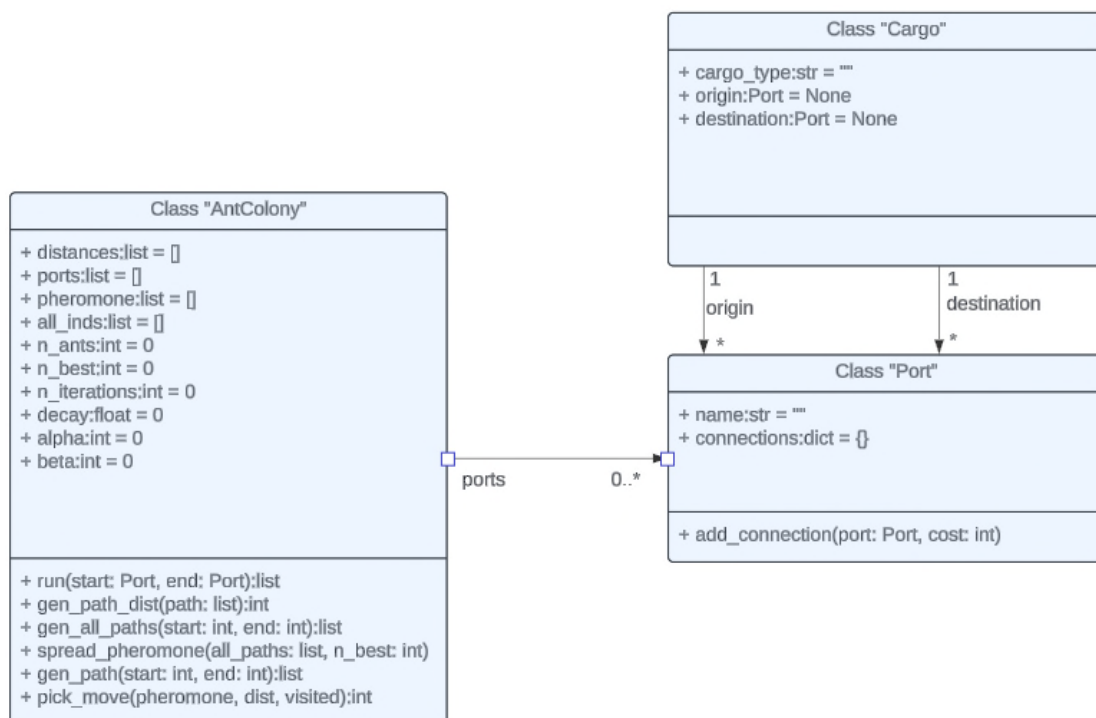
## Перелік посилань

1. History-Maps. URL: <https://history-maps.com/uk/story/American-Civil-War/event/Battle-of-Fredericksburg>.
2. Wikipedia. Freight Rate. URL: [https://en.wikipedia.org/wiki/Freight\\_rate](https://en.wikipedia.org/wiki/Freight_rate).
3. Dolphincargo URL: <https://dolphincargo.com.ua/ua/shho-take-fraxt-i-yak-vin-rozraxovuyetsya/>.
4. Wikipedia. Logistics. URL: <https://en.wikipedia.org/wiki/Logistics>.
5. Cristina Capineri, Thomas R. Leinbach. Freight Transport, Seamlessness, and Competitive Advantage in the Global Economy. URL: <https://journals.open.tudelft.nl/ejtir/article/view/4321>.
6. Michel Savy, June Burnham. Freight Transport and the Modern Economy. 2013. №1. С. 1-400. URL: <https://www.taylorfrancis.com/books/mono/10.4324/9780203070338/freight-transport-modern-economy-michel-savy-june-burnham>.
7. Wikipedia. Houthi movement. URL: [https://en.wikipedia.org/wiki/Houthi\\_movement](https://en.wikipedia.org/wiki/Houthi_movement).
8. Wikipedia. Piracy off the coast of Somalia. URL: [https://en.wikipedia.org/wiki/Piracy\\_off\\_the\\_coast\\_of\\_Somalia](https://en.wikipedia.org/wiki/Piracy_off_the_coast_of_Somalia).
9. Wikipedia. Automation. URL: <https://en.wikipedia.org/wiki/Automation>.
10. Wikipedia. Human Error. URL: [https://en.wikipedia.org/wiki/Human\\_error](https://en.wikipedia.org/wiki/Human_error).
11. A.E. Eiben, J.E. Smith. Introducing to Evolutionary Computing. 2015. № 2. С. 25-48.
12. IEEE. Evolutionary algorithms: A critical review and its future prospects. URL: <https://ieeexplore.ieee.org/abstract/document/7955308>
13. MIT Press. An Overview of Evolutionary Algorithms for Parameter Optimization. URL: <https://ieeexplore.ieee.org/abstract/document/6791438>
14. Wikipedia. Genetic algorithm. URL: [https://en.wikipedia.org/wiki/Genetic\\_algorithm](https://en.wikipedia.org/wiki/Genetic_algorithm).

15. Jordan Storms. Multi-Objective Genetic Algorithms: Combining CS and Evolution. URL: <https://medium.com/@jordanstorms/multi-objective-genetic-algorithms-combining-cs-and-evolution-4ac111ef98a4>.
16. Wikipedia. Ant colony optimization algorithms. URL: [https://en.wikipedia.org/wiki/Ant\\_colony\\_optimization\\_algorithms](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms).
17. Vijini Mallawaarachchi. Introducing to Genetic Algorithms. URL: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>.
18. Wikipedia. Travelling salesman problem. URL: [https://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem).
19. 3GTMS. URL: <https://go3g.com/>.
20. FreightWaves. Transparency19 Demo: 3Gtms. URL: [https://www.youtube.com/watch?app=desktop&v=ZG3fedP-uS4&ab\\_channel=FreightWaves](https://www.youtube.com/watch?app=desktop&v=ZG3fedP-uS4&ab_channel=FreightWaves).
21. Ant Logistics. URL: <https://ant-logistics.com/uk/main.html>.
22. Softkey. URL: <https://www.softkey.ua/ua/catalog/cloud/ant-logistics/>.
23. Ant Logistics. URL: <https://ant-logistics.com/intro/blk/images/features-gifs/integrations-api/uk/2-logistyka-crm.gif>
24. Cargo Smart. URL: <https://www.cargosmart.com/>
25. Cargo Smart Photo. [https://pbs.twimg.com/media/EjKz\\_YJXsAE9R28.jpg](https://pbs.twimg.com/media/EjKz_YJXsAE9R28.jpg).
26. Wikipedia. Python (programming language). URL: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
27. Jaydevs. URL: <https://jaydevs.com/python-vs-java/>
28. ChatGPT. URL: <https://chatgpt.com/>

# ДОДАТКИ

## Додаток А

**Розгорнута структура класів програмно реалізованого методу планування маршрутів вантажних перевезень з врахуванням фрахту**

## Додаток Б

### Програмні коди

Лістинг gen.py:

```
import time

import numpy as np

from algorithm import AntColony
from data_worker import (
    load_ports,
    gen_distances_matrice,
    print_best_route
)

ports, cargos = load_ports("input.txt")
distances = gen_distances_matrice(ports)

# Відстані між вершинами графа
distances = np.array(distances)

# Параметри мурашиного алгоритму
n_ants = 10 # Кількість мурах
n_best = 2 # Кількість найкращих мурах
n_iterations = 12 # Кількість ітерацій
decay = 0.95 # Коефіцієнт випаровування феромонів
alpha = 1 # Вплив феромонів
beta = 2 # Вплив евристичної інформації
def run_ant(distances, ports, start, end):
    distances = np.array(distances)
    start_time = time.time() # Початок вимірювання часу
    ant_colony = AntColony(
        distances,
        ports,
        n_ants,
        n_best,
        n_iterations,
        decay,
        alpha,
        beta
    )

    shortest_path = ant_colony.run(start, end)
    end_time = time.time() # Кінець вимірювання часу

    print(f"Кількість ітерацій для мурашиного алгоритму: {n_iterations}")
    print(f"Кількість часу мурашиного алгоритму: {end_time-start_time}")

    print_best_route(ports, shortest_path, start, end, cargo_task.cargo_type)

for cargo_task in cargos: # запуск пошуку оптимального маршруту для всіх вантажів
    start = cargo_task.origin
    end = cargo_task.destination
    run_ant(distances, ports, start, end)
```

Лістинг algorithm.py:

```
import numpy as np

class AntColony:
    def __init__(self, distances, ports, n_ants, n_best, n_iterations, decay, alpha=1,
beta=1):
        self.distances = distances
        self.ports = ports
        self.pheromone = np.ones(self.distances.shape) / len(distances)
        self.all_inds = range(len(distances))
        self.n_ants = n_ants
        self.n_best = n_best
        self.n_iterations = n_iterations
        self.decay = decay
        self.alpha = alpha
        self.beta = beta

    def run(self, start, end):
        start_index = 0
        end_index = 0

        for port in self.ports:
            if port.name == start:
                start_index = self.ports.index(port)

            if port.name == end:
                end_index = self.ports.index(port)

        all_time_shortest_path = ("placeholder", np.inf)
        for i in range(self.n_iterations):
            all_paths = self.gen_all_paths(start_index, end_index)
            self.spread_pheromone(all_paths, self.n_best)
            shortest_path = min(all_paths, key=lambda x: x[1])

            if shortest_path[1] < all_time_shortest_path[1]:
                all_time_shortest_path = shortest_path
                self.pheromone *= self.decay

        return all_time_shortest_path

    def spread_pheromone(self, all_paths, n_best):
        sorted_paths = sorted(all_paths, key=lambda x: x[1])
        for path, dist in sorted_paths[:n_best]:
            for move in path:
                self.pheromone[move] += 1.0 / self.distances[move]

    def gen_path_dist(self, path):
        total_dist = 0
        for ele in path:
            total_dist += self.distances[ele]
        return total_dist

    def gen_all_paths(self, start, end):
        all_paths = []
        for i in range(self.n_ants):
            path = self.gen_path(start, end)
            all_paths.append((path, self.gen_path_dist(path)))
        return all_paths

    def gen_path(self, start, end):
        path = []
```

```

visited = set()
visited.add(start)
prev = start
while prev != end:
    move = self.pick_move(self.pheromone[prev], self.distances[prev], visited)
    path.append((prev, move))
    prev = move
    visited.add(move)
return path

def pick_move(self, pheromone, dist, visited):
    pheromone = np.copy(pheromone)
    pheromone[list(visited)] = 0

    pheromone = np.maximum(pheromone, 1e-10)

    row = pheromone ** self.alpha * ((1.0 / dist) ** self.beta)

    norm_row = row / row.sum()

    move = np_choice(self.all_inds, 1, p=norm_row)[0]
    return move
def np_choice(a, size, replace=True, p=None):
    return np.random.choice(a, size, replace, p)

```

Лістинг data\_worker.py:

```

import networkx as nx
import numpy as np
from matplotlib import pyplot as plt

from entities import Port, Cargo

def gen_distances_matrice(ports):
    distances = []

    for row in range(len(ports)):
        distance = []

        for column in range(len(ports)):
            if ports[column] in ports[row].connections:
                distance.append(ports[row].connections[ports[column]])
            else:
                distance.append(np.inf)

        distances.append(distance)

    return distances
def load_ports(filename):
    ports = []
    cargo = []

    with open(filename, 'r') as file:
        section = ""
        for line in file:
            line = line.strip()
            if not line:
                continue

            elif line == "Cargos:":
                section = "cargos"
                continue

            elif line == "Ports:":
                section = "ports"
                continue

            elif line == "Costs:":
                section = "costs"
                continue

```

```

if section == "ports":
    port_names = line.split(',')

    for port_name in port_names:
        port = Port(port_name)
        ports.append(port)

elif section == "costs":
    port_name, connections = line.split(":")
    connections = connections.split(";")
    for origin_port in ports:
        if origin_port.name == port_name:
            port = origin_port
            break

    for connection in connections:
        connected_port_name, cost = connection.split('=')

        for origin_conn_port in ports:
            if origin_conn_port.name == connected_port_name:
                connected_port = origin_conn_port
                break

        port.add_connection(connected_port, int(cost))

elif section == "cargos":
    cargo_type, origin, destination = line.split(':')
    cargo.append(Cargo(cargo_type, origin, destination))

return ports, cargo

def print_best_route(ports, shortest_path, start, end, cargo):
    shortest_route_text = ""
    shortest = []

    for route in shortest_path[0]:
        port_1 = ports[route[0]].name
        port_2 = ports[route[1]].name

        if not port_1 in shortest:
            shortest.append(port_1)

        if not port_2 in shortest:
            shortest.append(port_2)

    for port in shortest:
        shortest_route_text += f"{port} -> "

    shortest_route_text = shortest_route_text[:-4]
    print(f"Найкоротший шлях з {start} до {end}:", shortest_route_text)
    print(f"Загальна вартість знайденого маршруту: {shortest_path[1]}")
    print("-----")

    show_graph(ports, shortest, cargo)

def show_graph(ports, shortest, cargo):
    G = nx.Graph()
    for port in ports:
        for connected_port, cost in port.connections.items():
            G.add_edge(port.name, connected_port.name, weight=cost)

    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, node_size=700, node_color='skyblue', font_size=10, font_weight='bold', width=2)
    edge_labels = nx.get_edge_attributes(G, 'weight')
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)

    # Highlight the best route
    edges = [(shortest[i], shortest[i + 1]) for i in range(len(shortest) - 1)]
    nx.draw_networkx_edges(G, pos, edgelist=edges, edge_color='r', width=3)

    plt.title("Ports Connections with Best Route Highlighted")
    plt.text(0.5, 0.95, cargo, horizontalalignment='right',
            verticalalignment='center', transform=plt.gca().transAxes)

    plt.show()

```

## Додаток В

### Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# СИСТЕМА ФОРМУВАННЯ ВАНТАЖОПЕРЕВЕЗЕНЬ З ВРАХУВАННЯМ ФРАХТУ ЗА ЕВОЛЮЦІЙНИМ АЛГОРИТМОМ



**Виконав:**  
*студент 4 курсу, групи КН-20-2*  
**Небожинський Антон Андрійович**



**Керівник:**  
*доцент кафедри КН*  
**Багрій Руслан Олександрович**

2

## Актуальність

В 21 столітті з розвитком інформаційних технологій, з'явилося багато можливостей автоматизації різних бізнес процесів і сфера формування вантажоперевезень не виняток [9]. В сучасній сфері вантажоперевезення, логістичним компаніям яким необхідно прокласти маршрут, тепер не потрібно користуватись паперовими мапами та документами. Цифрові технології покращили цей процес, зробивши його зручним для спеціалістів які відповідають за формування вантажоперевезення, адже комп'ютер на відмінну від людини здатен опрацьовувати мільйони операцій за доли секунди, достатньо використати систему яка підходить для вирішення тої чи іншої задачі, та завантажити в неї необхідні вхідні дані.

Системи автоматизованого планування маршрутів можуть допомогти вирішити проблему формування вантажоперевезень. Вони дозволяють вказати початкову та кінцеву точки доставки, вартість фрахту, після чого система автоматично формує найдешевший маршрут доставки. Основна мета таких систем полягає в знаходженні найдешевших маршрутів та наданні результатів користувачу.

Мурашині алгоритми використовуються для пошуку оптимальних шляхів. В їх основі лежить приклад використання поведінки реальних мурах, які шукають найкоротший шлях від колонії до джерела їжі.

## Мета і задачі роботи

**Метою кваліфікаційної роботи бакалавра** є підвищення ефективності процесу планування маршрутів вантажних перевезень з урахуванням вартості фрахту, спрямованого на зменшення витрат на транспортування вантажів.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- розробити математичну модель вантажних перевезень: маршрут, вартість фрахту;
- визначити еволюційний алгоритм для рішення задачі оптимізації планування маршрутів вантажних перевезень;
- розробити метод для формування вантажних перевезень з врахуванням фрахту за еволюційним алгоритмом;
- розробити програмну реалізацію з використанням запропонованого методу для формування вантажних перевезень;
- провести тестування програмної реалізації на різних тестових наборах даних.

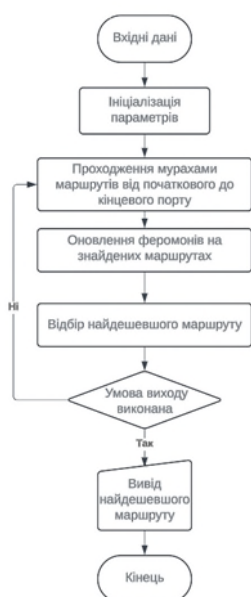
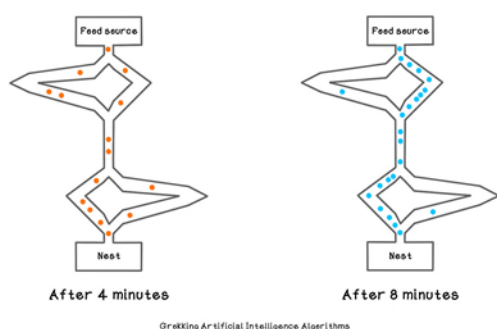


Схема роботи методу формування вантажоперевезень з врахуванням фрахту за еволюційним алгоритмом



## Схема роботи мурашиного алгоритму

## Математична модель методу формування вантажоперевезень

Цільовою функцією математичної моделі вантажоперевезень є функція мінімізації загальної вартості фрахту:

$$\text{Minimize } C = \sum_{(i,j) \in P} C_{ij}$$

де,  $C$  – вартість фрахту для проходження повним шляхом, а  $C_{ij}$  - це вартість фрахту для проходження маршрутом між сусідніми портами  $i$  та  $j$ , а  $P$  – це вибраний шлях між початковим та кінцевим портом.

## Приклад матриці відстаней між портами

Ports	Port1	Port2	Port3	Port4	Port5
Port1	$\infty$	100\$	150\$	10\$	$\infty$
Port2	100\$	$\infty$	$\infty$	50\$	$\infty$
Port3	150\$	$\infty$	$\infty$	10\$	$\infty$
Port4	$\infty$	50\$	10\$	$\infty$	20\$
Port5	$\infty$	$\infty$	$\infty$	20\$	$\infty$

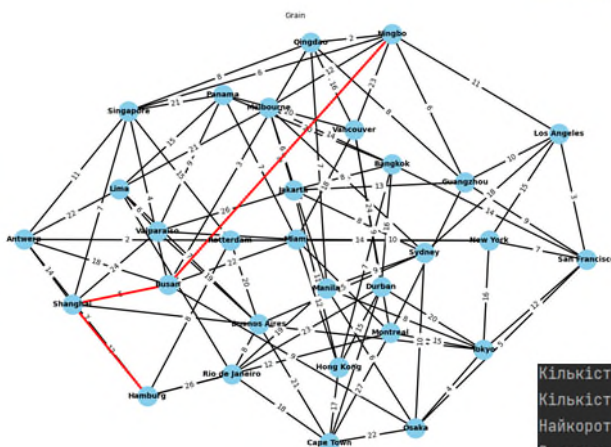
## Інформаційна структура методу формування вантажоперевезень

Сутність: Порт
Ім'я
Зв'язки

Сутність: Вантаж
Тип вантажу
Порт відправлення
Порт призначення

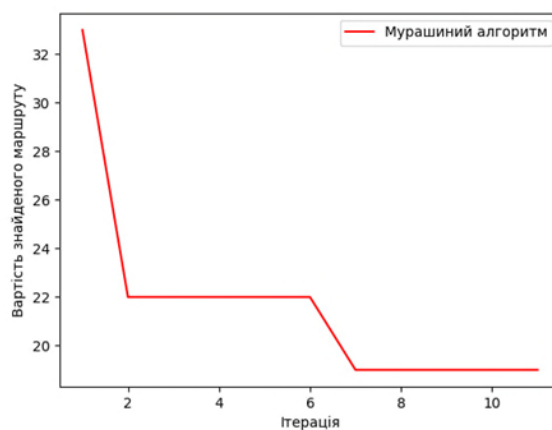
Сутність: Зв'язок
Початковий порт
Кінцевий порт
Вартість фрахту

## Результат програмної реалізації методу формування вантажоперевезень з врахуванням фрахту за еволюційним алгоритмом



Кількість ітерацій для мурашиного алгоритму: 12  
 Кількість часу мурашиного алгоритму: 0.09594368934631348  
 Найкоротший шлях з Hamburg до Ningbo: Hamburg -> Shanghai -> Busan -> Ningbo  
 Загальна вартість знайденого маршруту: 19.0

## Графік зміни знайденого оптимального рішення на кожній ітерації



## Висновки

У рамках виконання кваліфікаційної роботи бакалавра було успішно розроблено метод формування вантажоперевезень з врахуванням фрахту за еволюційним алгоритмом. Було проведено аналіз предметної області, досліджено рішення до формування вантажоперевезень, розглянуто існуючі програмні рішення формування вантажоперевезень.

Розроблена програмна реалізація методу формування вантажоперевезень з врахуванням фрахту за еволюційним алгоритмом за допомогою мурашиного алгоритму на мові програмування Python виконує наступні основні функції:

- Отримання та організація вхідних даних.
- Пошук оптимального рішення.
- Вивід результатів.

У якості засобів розробки було обрано середовище розробки PyCharm, мову програмування Python.

# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 2.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 7%

ID: 131392 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом Додано в БД: 2024-06-18 Автора: Антон НЕБОЖИНСЬКИЙ Керівники: Руслан БАГРІЙ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	80042	1188	2592 (3%)	36 (3%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра КН

ID перевірки:  
1016372414

Дата перевірки:  
18.06.2024 16:25:59 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
18.06.2024 17:30:12 EEST

ID користувача:  
100005671

Назва документа: КН-20-2 Небожинський\_ЗАПИСКА

Кількість сторінок: 70 Кількість слів: 11773 Кількість символів: 96182 Розмір файлу: 2.51 MB ID файлу: 1016179731

## 6.02% Схожість

Найбільша схожість: 3.04% з джерелом з Бібліотеки (ID файлу: 1016162677)

3.55% Джерела з Інтернету

441

Сторінка 72

4.71% Джерела з Бібліотеки

137

Сторінка 75

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

7

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом

Автор: студент гр. КН-20-2 Антон Небожинський

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Руслан Багрій

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- 1) за системою Anti-Plagiarism виявлені 2% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.
- 2) За системою UNICHECK виявлені 6,02%, що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 2% і 6.02% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН



Руслан БАГРІЙ

Олександр МАЗУРЕЦЬ

Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА**  
**на кваліфікаційну роботу бакалавра**

студента гр. КН-20-2 Небожвинського Антона Андрійовича

за темою Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом

**1. Актуальність теми**

Актуальність теми достатньо обґрунтована, оскільки сучасні логістичні системи потребують оптимізації процесів перевезення вантажів для зниження витрат і підвищення ефективності. Еволюційні алгоритми, які дозволяють оптимізувати процес планування маршрутів, є одним з перспективних інструментів для досягнення цих цілей. Врахування вартості фрахту у цих алгоритмах дає можливість не тільки скоротити витрати на транспортування, але й підвищити надійність та швидкість доставки.

**2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки**

Теми кваліфікаційної роботи "Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра, оскільки результатом роботи є метод для формування вантажних перевезень, що використовує мурашиний алгоритм, який є одним з різновидів еволюційних алгоритмів. Цей метод дозволяє ефективно вирішувати складні оптимізаційні задачі, характерні для планування логістичних маршрутів, шляхом моделювання поведінки мурах у природі для знаходження найкоротших шляхів. При вирішенні поставленої задачі використано методи оптимізації, еволюційні алгоритми, технології та методи проектування інформаційних систем.

**3. Професійні та особистісні якості бакалавра**

Небожвинський О.О. під час роботи над кваліфікаційною роботою бакалавра продемонстрував високий рівень знань у сфері комп'ютерних наук, особливо у застосуванні еволюційних алгоритмів для вирішення практичних завдань в логістиці. У своїй роботі він проаналізував існуючі методи, провів експерименти та оцінив ефективність запропонованого рішення, що свідчить про його здатність застосовувати теоретичні знання на практиці та вирішувати складні технічні завдання.

#### **4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

#### **5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи показала високий рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

#### **6. Повнота та якість розкриття теми роботи**

Тема роботи повністю розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, виконані усі поставлені задачі та розроблено програмку реалізація для підтвердження запропонованого методу.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

**8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений метод для формування вантажних перевезень з урахуванням фрахту, що базується на мурашиному алгоритмі, може бути впроваджений у логістичні компанії для оптимізації маршрутів і зменшення витрат на транспортування.

**9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Керівник \_\_\_\_\_



к.т.н., доц. Руслан БАГРІЙ



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента гр. КН-20-2 Небожсинського Антона Андрійовича

за темою: Метод для формування вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом

#### 1. Актуальність обраної теми

В сучасних умовах глобалізації та інтенсивного інформаційного обміну, логістичні компанії стикаються з великою кількістю даних, що впливають на планування маршрутів. Метод планування маршрутів вантажних перевезень з урахуванням фрахту дозволяє ефективно керувати цими даними.

#### 2. Повнота розкриття мети та завдань роботи

Під час виконання кваліфікаційної роботи бакалавра було реалізовано метод планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом для підвищення ефективності процесу планування маршрутів, що відповідає меті та завданням кваліфікаційної роботи і розкриває їх повною мірою.

#### 3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи складається з трьох розділів. У першому розділі проводиться аналіз предметної області, а саме: огляд еволюційних алгоритмів, аналіз існуючих програмних рішень та визначення мети і завдань. Другий розділ присвячений розробці методу планування маршрутів вантажних перевезень з урахуванням фрахту за еволюційним алгоритмом. В третьому розділі проводиться програмна реалізація методу планування маршрутів вантажних перевезень з урахуванням фрахту.

#### 4. Оцінка розробленого методу, його практична цінність

Розроблений метод планування маршрутів вантажних перевезень, підвищує ефективність планування маршрутів, що допоможе логістичним компаніям виконувати логістичні завдання, а саме планування маршрутів вантажних перевезень з врахуванням фрахту.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Записка оформлена якісно, відповідно до поставлених вимог. Зрозуміло написана, логістично послідовна та з виразною структурою.

#### 6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано розглянути можливість використання методу, інтегрованого у відповідну логістичну систему та забезпечення взаємодії з компонентами цієї системи.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслугоує кваліфікаційна робота.

Враховуючи якість виконання та відповідність усім необхідним вимогам, робота може бути допущена до захисту. Рекомендована оцінка дуже.

Рецензент

д.т.н., проф. Лисенко С.М.