

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій**

«На правах рукопису»  
УДК 004.4 \_

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Олександр РОЛІК  
«  » \_\_\_\_\_ 2024 р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
зі спеціальності 126 «Інформаційні системи та технології»  
на тему: «Система управління логістикою транспортної компанії з  
використанням гібридного алгоритму багатошляхової  
маршрутизації»**

Виконав:  
студент 2 курсу, групи ІС-32мп  
Заграй Валентин Віталійович \_\_\_\_\_

Керівник:  
Доцент кафедри ІСТ, к.т.н., доцент  
Деведжіогуллари Алла Вікторівна \_\_\_\_\_

Рецензент:  
Доцент каф ОТ, к.т.н., доцент  
Роковий Олександр Петрович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.  
Студент \_\_\_\_\_

Київ – 2024 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Заграю Валентину Віталійовичу**

1. Тема дисертації: «Система управління логістикою транспортної компанії з використанням гібридного алгоритму багатошляхової маршрутизації», науковий керівник дисертації Деведжіогуллари Алла Вікторівна, к.т.н., доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об'єкт дослідження Система управління логістикою транспортної компанії
4. Вихідні дані: мова програмування відсутність передачі персональних даних у інформаційну систему, висока швидкість передачі інформації
5. Перелік завдань, які потрібно розробити: аналіз сучасного стану логістичної сфери, аналіз систем-аналогів та алгоритмів, які раніше використовувались, розробка гібридного алгоритму, розробка програмного продукту, розробка стартап-проекту.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: блок-схема гібридного генетичного алгоритму з покращеним алгоритмом

імітації відпалу, діаграма класів, діаграма прецедентів, діаграма станів, блок-схема алгоритму регенерації маршруту.

7. Орієнтовний перелік публікацій: Тези доповіді Гібридний алгоритм для вирішення задач багатошляхової маршрутизації в логістиці.

8. Дата видачі завдання 02.09.2024 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	до 13.09.24	
2	Визначення методології дослідження	до 20.09.24	
3	Формування цілей та задач розробки	до 27.09.24	
4	Аналіз існуючих рішень	до 04.10.24	
5	Оцінка та порівняльний аналіз алгоритмів	до 11.10.24	
6	Розробка оптимізаційної моделі	до 18.10.24	
7	Розробка програмного продукту	до 15.11.24	
8	Розробка стартап-проекту	до 22.11.24	

Студент

Валентин ЗАГРАЙ

Науковий керівник

Алла ДЕВЕДЖІОГУЛЛАРИ

## РЕФЕРАТ

Система управління логістикою транспортної компанії з використанням гібридного алгоритму багатошляхової маршрутизації: 112 с., 27 табл., 28 рис., 9 дод., 40 джерел.

ГІБРИДНИЙ АЛГОРИТМ, ЗАДАЧА МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ, ГЕНЕТИЧНИЙ АЛГОРИТМ, АЛГОРИТМ ІМІТАЦІЇ ВІДПАЛУ, ЛОГІСТИКА.

Вибрана тема дослідження є актуальною сьогодні через зростання попиту на логістичні послуги та розширення вибраного ринку.

Метою даної роботи є підвищення ефективності та покращення роботи транспортної компанії через розробку системи управління логістикою з використанням гібридного алгоритму багатошляхової маршрутизації з можливістю створення альтернативних маршрутів. Завданнями дослідження є огляд сучасного стану логістики із розглядом проблемних аспектів та наявних систем-аналогів, порівняльний аналіз існуючих методів вирішення задачі маршрутизації, формулювання математичної постановки задачі, розробка нового гібридного генетичного алгоритму з покращеним алгоритмом імітації відпалу, розробка програмного продукту з використанням описаного алгоритму та розробка стартап-проекту.

Об'єктом дослідження даної роботи є система управління логістичними процесами компанії з використанням гібридного алгоритму багатошляхової маршрутизації та всі процеси створення і оптимізації маршрутів доставки замовлень.

Предметом дослідження є алгоритм маршрутизації транспортних засобів у логістичних системах.

За темою дисертації були опубліковані тези доповіді «Гібридний алгоритм для вирішення задач багатошляхової маршрутизації в логістиці»[1].

## ABSTRACT

Logistics management system for a transport company using a hybrid multi-path routing algorithm: 112 p., 27 tab., 28 draw., 9 app., 40 sources.

HYBRID ALGORITHM, VEHICLE ROUTING PROBLEM, GENETIC ALGORITHM, SIMULATED ANNEALING ALGORITHM, LOGISTIC.

The selected research topic is relevant today due to the growing demand for logistics services and the expansion of the selected market.

The purpose of this work is to increase the efficiency of managing routes of the transport company through the development of a logistics management system using a hybrid routing algorithm with the possibility of creating alternative routes. The objectives of the study are to review the current state of logistics with the consideration of problematic aspects and existing analogue systems, comparative analysis of existing methods for solving the vehicle routing problem, mathematical formulation of the problem, development of a new hybrid genetic algorithm with an improved annealing simulation algorithm, development of a software product using the described algorithm and development of a startup project.

The object of the study of this work is the company's logistics process management system using a hybrid multi-layered routing algorithm and all processes for creating and optimizing order delivery routes.

The subject of the study is the algorithm for routing vehicles in logistics systems.

Theses on topic «Hybrid algorithm for solving multi-path routing tasks in logistics» were published based on master's work[1].

## ЗМІСТ

ВСТУП.....	8
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ....	10
1.1 Огляд поточного стану логістичної сфери та ідентифікація проблемних аспектів .....	10
1.1.1 Поточний стан логістичної сфери .....	10
1.1.2 Тенденції розвитку логістичних систем.....	11
1.1.3 Основні виклики і проблеми в логістичних систем та маршрутизації	14
1.2 Аналіз існуючих систем управління логістикою і маршрутизацією .....	16
1.2.1 Аналіз системи Route4Me .....	16
1.2.2 Аналіз системи OptimoRoute .....	18
1.2.3 Аналіз системи MyRouteOnline .....	20
1.2.4 Аналіз системи Paragon.....	21
1.3 Визначення основних метрик при побудові оптимальних маршрутів ....	23
Висновки до розділу 1.....	24
2 ТЕОРЕТИЧНІ ЗАСАДИ ОПТИМІЗАЦІЇ МАРШРУТІВ .....	25
2.1 Теорія графів та маршрутизація .....	25
2.2 Класифікація методів та алгоритмів для задачі створення та оптимізації маршрутів .....	30
2.3 Аналіз гібридних алгоритмів задачі маршрутизації.....	36
2.3.1 Аналіз генетичного алгоритму з локальним пошуком .....	36
2.3.2 Аналіз гібридного мурашиного алгоритму з імітацією відпалу .....	37
2.3.3 Аналіз гібридного алгоритму табу-пошук і методу генерації стовпців	38
2.3.4 Аналіз гібридного алгоритму штучного бджолиного рою з локальним пошуком.....	39
Висновки до розділу 2.....	40
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ТА РОЗРОБЛЕННЯ ГІБРИДНОГО АЛГОРИТМУ БАГАТОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ.....	41
3.1 Математична постановка задачі .....	41

3.2	Опис гібридного генетичного алгоритму з покращеним алгоритмом імітації відпалу для багатошляхової маршрутизації .....	44
3.3	Адаптація алгоритму до перебудови маршрутів .....	53
3.4	Переваги та недоліки розробленого гібридного алгоритму .....	55
	Висновки до розділу 3.....	55
4	РОЗРОБЛЕННЯ ПРОГРАМНОГО ПРОДУКТУ ТА ТЕСТУВАННЯ.....	56
4.1	Архітектура системи .....	56
4.2	Вибір технологій та інструментів.....	58
4.3	Опис основних класів програми .....	59
4.4	Опис функціоналу та інструкція користувача .....	63
4.5	Тестування системи.....	74
	Висновки до розділу 4.....	76
5	РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ .....	77
5.1	Опис ідеї проекту .....	77
5.2	Технологічний аудит ідеї проекту .....	80
5.3	Аналіз ринкових можливостей запуску стартап-проекту .....	83
5.4	Розроблення маркетингової програми стартап-проекту .....	99
	Висновки до розділу 5.....	105
	ВИСНОВКИ .....	107
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	109
	ДОДАТОК А.....	112
	ДОДАТОК Б .....	113
	ДОДАТОК В.....	114
	ДОДАТОК Г .....	115
	ДОДАТОК Д.....	116
	ДОДАТОК Е.....	117
	ДОДАТОК Ж.....	118
	ДОДАТОК И.....	119
	ДОДАТОК К.....	120

## ВСТУП

Багатошляхова маршрутизація є важливим аспектом управління транспортними системами, оскільки вона дозволяє визначати оптимальні шляхи для перевезення вантажів, мінімізуючи витрати та час доставки. У сучасному світі логістики, де швидкість і точність доставки мають вирішальне значення, використання ефективних алгоритмів маршрутизації стає ключовим фактором успіху. Від великих міжнародних компаній до локальних підприємств, всі вони стикаються з викликом забезпечення своєчасної доставки продукції з мінімальними витратами.

Транспортні системи стають все більш складними, оскільки вони повинні враховувати різноманітні фактори, такі як зміни в дорожньому трафіку, погодні умови, обмеження вантажопідйомності, та інші динамічні параметри. В таких умовах прості підходи до маршрутизації, які використовувалися раніше, вже не здатні забезпечити необхідний рівень ефективності та гнучкості. Тому використання сучасних алгоритмів багатошляхової маршрутизації є критично важливим для підтримки конкурентоспроможності на ринку.

Багатошляхова маршрутизація включає в себе різні алгоритми та методи, які дозволяють оптимізувати маршрути не лише для одного транспортного засобу, але і для цілого парку транспорту. Це дозволяє знаходити найкращі комбінації маршрутів, які забезпечують максимальну ефективність використання ресурсів. Зокрема, такі алгоритми дозволяють знизити витрати на паливе, скоротити час доставки, зменшити знос транспортних засобів, а також покращити загальний рівень обслуговування клієнтів.

Метою цієї магістерської роботи є підвищення ефективності та покращення роботи логістичної компанії з доставки замовлень через розробку системи управління логістикою з використанням гібридного алгоритму багатошляхової маршрутизації з можливістю створення альтернативних маршрутів.

Об'єктом дослідження даної роботи є система управління логістичними процесами компанії з використанням гібридного алгоритму багатошляхової

маршрутизації та всі процеси створення і оптимізації маршрутів доставки замовлень.

Завдання, які потрібно вирішити для досягнення поставленої мети проекту:

- дослідити сучасний стан логістики, основні тенденції та вимоги, які поставлені перед логістичними системами;
- проаналізувати існуючі логістичні системи для створення та оптимізації маршрутів та сформувавши на основі цього основні метрики ефективності логістичної системи, які враховуються при оптимізації маршрутів;
- сформувавши вимоги до проекту на основі попереднього аналізу;
- розглянути теоретичні засади різних алгоритмів оптимізації маршрутів, зробити їхній аналіз для створення власного алгоритму маршрутизації;
- розробити гібридний алгоритм багатошляхової маршрутизації з можливістю створення альтернативних маршрутів;
- розробити систему управління логістикою з використанням розробленого алгоритму;
- виконати тестування створеної системи та провести ряд експериментів для перевірки її оптимізаційних та швидкісних якостей.

## 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

### 1.1 Огляд поточного стану логістичної сфери та ідентифікація проблемних аспектів

#### 1.1.1 Поточний стан логістичної сфери

Логістичні системи відіграють ключову роль як у нашому особистому житті так і у глобальних світових процесах. Дані системи дозволяють ефективно управляти маршрутами та доставляти різні товари, інформацію та послуги до кінцевих користувачів. У період глобалізації здатність швидко, надійно та з меншими витратами доставляти продукцію має вирішальне значення для успіху бізнесу та задоволення клієнтів. Логістичні системи охоплюють широкий спектр процесів, від складування та управління запасами до транспортування та розподілу, всі з яких повинні працювати в гармонії, щоб оптимізувати потік ресурсів. Оскільки компанії прагнуть задовольнити зростаючі вимоги споживачів, важливість надійних, адаптованих логістичних систем ніколи не була більшою. Крім того, з розвитком електронної комерції, розвитком технологій та прагненням до більш стійкої практики, логістичний сектор зазнає значних трансформацій. Проведемо огляд поточного стану логістичних систем для розуміння проблем, які існують у даній галузі і сучасних рішень.

На сьогоднішній день логістика є невід'ємною частиною світової економіки, оскільки її застосування в усьому світі дозволяє підприємствам виходити на міжнародний ринок. За допомогою логістики компанії можуть створювати додаткову вартість для своїх продуктів, економити гроші, покращувати роботу з клієнтами і збільшувати популярність свого бренду. Зараз багато компаній у всьому світі шукають стратегічного управління логістикою, щоб знизити свої транспортні витрати.

Станом на 2023 рік ринок логістики оцінюється в майже 9 трильйонів доларів і з кожним роком ця цифра стає все більшою. За прогнозами у 2028 році весь ринок логістики буде оцінюватись в більш ніж 15 трильйонів доларів[3], що показує що дана галузь є широкорозповсюдженою і її вплив на наше життя тільки

збільшується. Більше ніж половину всіх логістичних витрат на даний момент становлять транспортні витрати а саме 58%, що спонукає різні логістичні компанії покращувати свої алгоритми створення логістичних маршрутів для зменшення відстаней і відповідно зменшення транспортних витрат.

### 1.1.2 Тенденції розвитку логістичних систем

Виділяють кілька основних напрямків та тенденцій у яких розвиваються логістичні системи у світі. До таких тенденцій можна віднести:

- впровадження штучного інтелекту;
- автоматизація логістичних систем;
- створення нових алгоритмів маршрутизації;
- «зелена» логістика;
- використання робототехніки;
- багатошляхова маршрутизація.

Розглянемо більш детально всі ці тенденції.

Впровадження штучного інтелекту в логістиці значно змінює способи організації та управління постачанням товарів, оптимізує процеси і дозволяє ефективніше використовувати ресурси.

Штучний інтелект допомагає оптимізувати маршрути доставки, що є особливо актуальним для транспортних компаній, які повинні обслуговувати велику кількість клієнтів. Завдяки використанню алгоритмів машинного навчання створюються нові алгоритми з урахуванням різних факторів, таких як завантаженість доріг, погодні умови та інші[4].

Різні системи штучного інтелекту вже досить тривалий час використовуються для моніторингу стану транспортних засобів і вантажів. Датчики передають інформацію про температуру або цілісність упаковки, і у разі виявлення відхилень системи автоматично сповіщають про це або приймають рішення про зміну умов транспортування.

За прогнозами при сьогоднішніх темпах впровадження логістики до 2035 року штучний інтелект підвищить продуктивність логістики більш ніж на 40%.

Застосування систем штучного інтелекту у логістиці сьогодні це інструмент, який активно розвивається але вже дозволяє досягати значної економії ресурсів, підвищувати точність та швидкість процесів, а також забезпечувати конкурентну перевагу на ринку. Введення таких технологій є обов'язковою для логістичних компаній у найближчому майбутньому.

Системи штучного інтелекту тісно пов'язані з ще одним напрямком розвитку логістики таким як автоматизація систем. Системи управління складом дозволяють автоматизувати процеси зберігання, пошуку та переміщення товарів. Такі системи допомагають відстежувати розташування кожного товару, що дозволяє швидше знаходити й обробляти замовлення.

Крім того, зараз дедалі частіше використовують автоматизовані системи управління замовленнями. Вони дозволяють підприємствам обробляти замовлення в режимі реального часу, синхронізуючи їх з наявністю товарів на складі. Вони можуть автоматично поповнювати запаси, на основі прогнозів, які розраховуються за допомогою машинного навчання та аналізу даних[5].

Ще одним напрямком розвитку логістичних систем є впровадження і використання робототехніки. Сьогодні автономні роботи використовуються майже всюди. Сортувальні роботи автоматично класифікують і розподіляють товари на сортувальних станціях логістичних хабів. Також існують роботи, які можуть автоматично проводити інвентаризацію на складі, перевіряючи рівень запасів, зменшуючи потребу в перевірках напряму людьми[6]. Роботи сканують штрих-коди на товарах і передають дані в систему управління запасами в режимі реального часу.

Велику популярність набирає тенденція на виконання замовлень з використанням дронів. Вони використовуються для швидкої доставки невеликих вантажів на короткі дистанції, наприклад, у важкодоступні або урбанізовані райони.

Останнє десятиліття активно приймаються міри для підтримки екології нашої планети і сфера логістики не є виключенням саме тому з'явилося таке явище як «зелена» логістика – підхід до логістичних процесів спрямований на зменшення негативного впливу на наше довкілля. Цей підхід охоплює використання екологічно чистих технологій, оптимізацію процесів і зменшення викидів парникових газів на всіх етапах ланцюга поставок. Основними аспектами «зеленої» логістики є: зменшення викидів парникових газів, оптимізація споживання електроенергії та зворотна логістика – повернення товару виробнику для утилізації або переробки коробок.

Зелена логістика є важливою, тому що вона не тільки зменшує негативний вплив на довкілля, але також покращує ефективність логістичних операцій, знижує витрати та підвищує конкурентоспроможність компанії.

Іншим напрямком розвитку логістичних систем є створення нових алгоритмів маршрутизації, зокрема для багатошляхової маршрутизації, оскільки це дозволяє оптимізувати використання ресурсів, знижувати витрати та підвищувати ефективність транспортних мереж. Створення нових алгоритмів допомагає брати до уваги різні чинники, які впливають на доставку, такі як дорожні затори, погодні умови, відміна замовлення у певного клієнта та коректування маршруту відповідно до цього. Зокрема також йде створення нових алгоритмів саме для багатошляхової маршрутизації які є стійкими до змін, дозволяють будувати кілька альтернативних шляхів та швидко перебудовувати маршрут в залежності від обставин.

Є кілька напрямків розробки нових алгоритмів для ефективнішої маршрутизації такі як створення гібридних алгоритмів, створення алгоритмів з використанням машинного навчання, створення алгоритмів багатокритеріальної оптимізації та інші напрямки, які детально будуть розглянуті пізніше.

Отже, є кілька основних тенденцій розвитку логістичних систем, які допомагають покращити їхню ефективність та зекономити кошти для логістичних компаній.

### 1.1.3 Основні виклики і проблеми в логістичних систем та маршрутизації

Логістичні системи хоч і активно розвиваються зокрема слідуючи вище зазначеним тенденціям проте є також ряд викликів і проблем з якими компанії стикаються, як при менеджменті товарами так і особливо при маршрутизації товарів. Розглянемо основні виклики і проблеми логістичних систем і компаній.

Однією з основних проблем, з якими стикаються логістичні компанії, як і раніше, є складнощі із створенням маршрутів для доставки товарів. Маршрутизація вимагає врахування багатьох факторів, кількість яких тільки зростає. Серед основних критеріїв: відстань, час доставки, обмеження на транспортні засоби, вимоги клієнтів та інші обмеження. Крім того зростання кількості замовлень в умовах електронної комерції вимагає гнучкого планування маршрутів та застосування нових швидких алгоритмів багатошляхової маршрутизації, які будуть працювати не гірше за звичайні алгоритми.

Іншою проблемою, яка насамперед впливає на терміни доставки є проблеми на дорогах, такі як затори, ремонти та аварії, які призводять до затримок в доставці. У великих містах транспортна система часто перевантажена, що ускладнює доставку товарів у встановлені терміни. Такі проблеми потребують створення гнучких алгоритмів, які можуть переналаштувати маршрути в режимі реального часу та не втрачати багато часу при доставці.

Ще однією проблемою є обмеження транспорту та ресурсів. Логістичні компанії мають обмежену кількість транспортних засобів і ресурсів, які повинні використовуватись максимально ефективно. Транспортні засоби мають обмежену вантажопідйомність і об'єм, що вимагає ретельного планування завантаження. Зокрема дана проблема гостро постає у святкові періоди коли можуть виникати проблеми з доступністю транспортних засобів. Тому ефективне використання транспортних засобів є однією з головних цілей всіх логістичних систем.

Також великою проблемою та викликом сьогодні є доставка останньої милі. «Остання миля» є однією з найбільш дорогих і складних частин логістики, яка включає доставку товару безпосередньо до кінцевого споживача. Через зростання

попиту на швидку доставку очікування клієнтів щодо швидкості доставки постійно зростають, що створює додатковий тиск на логістичні системи. Транспортні засоби часто не використовуються на повну потужність через необхідність здійснювати доставку у різні локації в межах міста або населених пунктів. Крім того, багато клієнтів очікують безкоштовної доставки, що створює додатковий фінансовий тиск на компанії, які повинні забезпечити ефективну та рентабельну логістику[5].

Проте основним викликом для компаній є зменшення вартості логістики. Постійне підвищення вартості пального, оренди складів, зарплат водіїв та інші витрати постійно тиснуть на логістичні компанії. Впровадження автоматизації та цифрових технологій також потребує значних коштів, хоча новітні технології можуть підвищити ефективність і зменшити витрати доставок в довгостроковій перспективі, початкові інвестиції у їх впровадження є значними[8].

Крім того, зважаючи на сьогоденну ситуацію потрібно також зважати на екологію. Зростаючий тиск на компанії з боку екологічних організацій та суспільства щодо скорочення викидів вуглецю та екологічної відповідальності змушує компанії постійно переглядати свою політику та підлаштовуватись під всі вимоги. Так уряди багатьох країн посилюють вимоги до рівнів викидів вуглекислого газу, і вимагають від компаній використовувати більш екологічні транспортні засоби або маршрути. Використання електричного транспорту та гібридних транспортних засобів є дорогими інвестиціями для компаній.

Ще одним викликом для логістичних систем є впровадження цифрових технологій. Логістичні компанії повинні адаптуватися до нових технологій, таких як штучний інтелект, BigData та інтернет речей, для покращення своєї ефективності. Як було раніше сказано, впровадження нових технологій вимагає значних фінансових вливань і не всі компанії мають ресурси для повноцінної цифровізації процесів, що ускладнює управління в умовах високої конкуренції. Крім того із зростанням використання цифрових рішень також зростають ризики кіберзагроз і зломів, що може порушити логістичні процеси, тому потрібно створювати добру безпеку для всіх систем.

Також одним із викликів для логістичних систем є правильне прогнозування попиту. Воно важливе для забезпечення ефективного управління запасами і транспортом. Непередбачувані зміни у попиті можуть призвести до надмірних або недостатніх запасів. Якщо логістична компанія не мати доступу до точних і вчасних даних, прогнозування попиту може бути складним і витрати зростатимуть.

Подолання цих викликів і проблем, або зменшення їх впливу є стратегічним завданням для будь-якої логістичної системи чи компанії. Це потребує інновацій, автоматизації, ефективного використання даних і довгострокового планування, щоб логістична система залишалася конкурентоспроможною і стійкою до змін.

## 1.2 Аналіз існуючих систем управління логістикою і маршрутизацією

### 1.2.1 Аналіз системи Route4Me

Більшість сучасних систем для управління логістикою пропонуються у вигляді складових ERP-системи. ERP-системи (Enterprise Resource Planning) є комплексними програмними рішеннями, що інтегрують різні бізнес-процеси та функції організації в єдину інформаційну платформу. Вони забезпечують централізоване управління і координацію ресурсів, включаючи фінанси, виробництво, закупівлі, продажі, управління персоналом і логістику[10].

Розглянемо популярні системи управління логістикою і маршрутизацією.

Однією з найпопулярніших систем є Route4Me. Route4Me - платформа, яка була заснована в 2009. Платформа дозволяє оптимізувати та інтегрувати важливі робочі процеси доставки останньої милі, включаючи планування маршруту, відправлення товарів, відстеження доставки. Програмне забезпечення компанії базується в хмарі та представлене багатьма різними додатками. Інтерфейс системи зображено на рисунку 1.1.

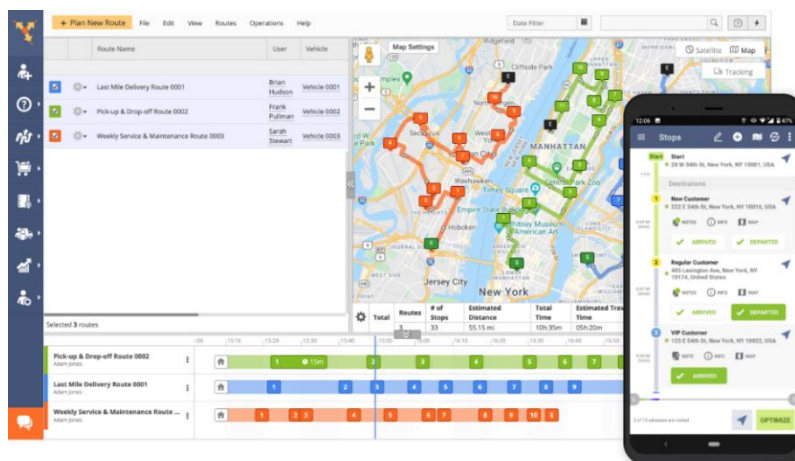


Рисунок 1.1 – Інтерфейс системи Route4Me

Основні функції і можливості Route4Me:

- автоматичне створення та оптимізація наданих користувачем маршрутів;
- розробка маршрутів для різних типів транспортних засобів;
- динамічне планування маршрутів залежно від різних умов планування маршрутів;
- розробка маршрутів з великою кількістю транспортних засобів;
- кросплатформеність(окрім сайту також є мобільні додатки на Android і IOS);
- система підтримує інтеграцію з GPS;
- аналітика та звіти.

Route4Me надає широкий спектр послуг та має багато переваг таких як автоматична оптимізація, гнучкість, надання аналітики, відстеження в реальному часі та інше. Проте Route4Me має також суттєві недоліки. Одним з основних недоліків є вартість тарифів – мінімальний тариф коштує 200 доларів на місяць і дає доступ тільки 5 користувачам, що є великою проблемою для одиночних користувачів, які хочуть оптимізувати свій маршрут. Крім того, дані тарифи можуть бути неприйнятними для малих бізнесів, особливо для нашого регіону, де ціни американського ринку є завеликими.

Іншим недоліком є те, що інтерфейс є складним для розуміння. Крім того, дана система підтримує будівництво лише одного оптимального маршруту без альтернативних рішень, для того щоб їх отримати потрібно змінювати умови.

Отже, система побудови маршрутів Route4Me є багатофункціональною, проте має високу ціну для нашого регіону та дозволяє створювати лише один оптимальний маршрут початково.

## 1.2.2 Аналіз системи OptimoRoute

Розглянемо ще одну популярну логістичну систему для планування маршрутів OptimoRoute. OptimoRoute – система для створення нових і оптимізації створених маршрутів для логістичних компаній. Ця система дозволяє оптимізувати та планувати маршрути та створювати точні розклади для великої кількості замовлень. Крім того, існує мобільний додаток, який використовують переважно водії транспортних засобів доставки для слідування по створеним маршрутам. Дана система була створена у 2012 році в Америці і її дохід складає 6 мільйонів доларів[9].

Інтерфейс даної системи можемо побачити на рисунку 1.2

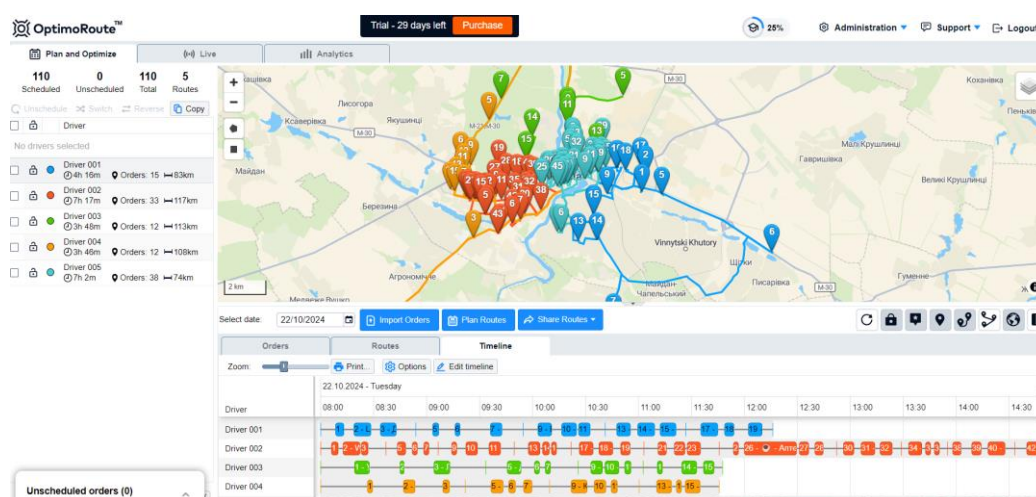


Рисунок 1.2 – Інтерфейс системи OptimoRoute

Розглянемо переваги вибраної системи:

- система автоматично будує маршрути з урахуванням різних факторів, таких як обмеження по часу доставки, обмеження на вантажопідйомність та інші;
- додаток дозволяє планування на багато днів вперед;
- простий інтерфейс, зрозумілий для користувача;
- мобільний додаток для водіїв;
- система надає інтеграційний API, що дозволяє інтегрувати сервіси у свою програму;
- можливість аналізувати свої пройдені маршрути та витрати OptimoRoute;
- система надає пробний безкоштовний 30-денний період.

Хоч система OptimoRoute і є легкою для застосування та має багато переваг проте вона має також істотні недоліки. Серед основних недоліків можна виділити наступні:

- дорога вартість для малого бізнесу, особливо для нашого регіону(стартові плани за місяць починаються з 40 доларів за одного водія);
- аналітика не доступна на стартових тарифних планах і є тільки на дорожчих;
- потрібно зберігати створені маршрути, так як при відсутності інтернету водій не зможе побачити маршрут;
- система створює один маршрут для доставки, не дозволяючи автоматично генерувати кілька альтернативних маршрутів для порівняння;
- немає підтримки системи повернення товару назад.

Хоч OptimoRoute і враховує багато факторів при плануванні маршрутів, проте існує багато умов, які система не враховує або потребує налаштування вручну додатково. Наприклад, система не враховує погодні умови, а пріоритетність клієнтів початково також не враховується, це потребує ручних налаштувань. Також вона не відстежує дорожні затори в реальному часі.

Отже, система OptimoRoute є багатофункціональною та є більш зручною ніж система Route4Me та її легко можна інтегрувати у власні системи. Проте дана система потребує постійного підключення до інтернету та при побудові маршрутів

деякі важливі фактори не враховуються. Крім того, дана система не підтримує багатошляхову маршрутизацію та фокусується на побудові лише одного оптимального маршруту. Цей додаток добре підходить для середніх підприємств, але є не дуже добрим варіантом для малих підприємств.

### 1.2.3 Аналіз системи MyRouteOnline

Розглянемо для порівняння також логістичну систему MyRouteOnline. MyRouteOnline – веб-додаток для планування маршрутів, головною ціллю якого є автоматизація та оптимізація логістичних процесів. MyRouteOnline було створено у 2009 році Баручем Аксельродом з орієнтацією на малий бізнес. Сьогодні компанія надає послуги для різних видів бізнесу[11].

Інтерфейс даної системи зображено на рисунку 1.3.

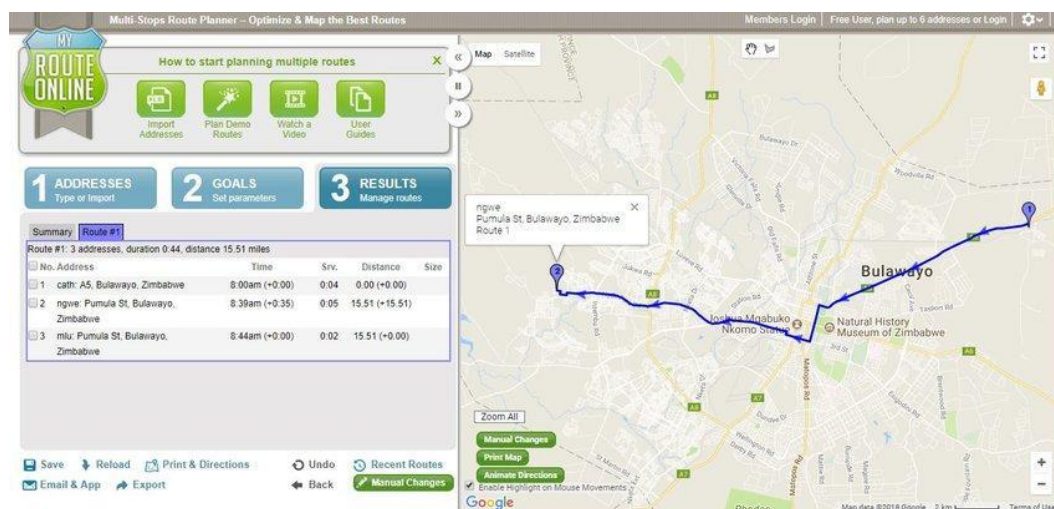


Рисунок 1.3 – Інтерфейс системи MyRouteOnline

Розглянемо основні переваги системи MyRouteOnline:

- простий інтерфейс;
- простий з будь-якого пристрою та з будь-якого місця;
- експорт маршрутів на GPS-пристрої;
- оптимізація маршрутів для кількох транспортних засобів;
- можливість отримати звіти та аналітику;

- гнучкість у налаштуванні маршрутів;
- пробний безкоштовний період.

Система MyRouteOnline має багато переваг, проте у ній є багато мінусів про які варто знати, при роботі з нею:

- висока вартість для починаючих компаній;
- MyRouteOnline не підтримує багатошляхової маршрутизації та буде лише один варіант оптимальних маршрутів;
- недостатня кількість параметрів для оптимізації маршрутів;
- система не має можливості змінити маршрут в реальному часі, наприклад через затори;
- MyRouteOnline надає тільки базову аналітику про пройдений шлях та довжини маршрутів;
- відсутність мобільного додатку та залежність від стабільного інтернет з'єднання;
- складність інтеграції з іншими системами.

Отже, MyRouteOnline - просте та доступне рішення для планування маршрутів, яке підходить для малого та середнього бізнесу, що займається доставкою. Проте, для великих компаній або складних логістичних процесів система не підходить через відсутність складних налаштувань маршрутів, відсутність підтримки в реальному часі та відсутність мобільного додатку для водіїв. MyRouteOnline є дешевшою та простішою альтернативою для Route4Me та OptimoRoute.

#### 1.2.4 Аналіз системи Paragon

Розглянемо також логістичну систему, яка спеціалізується на наданні логістичних послуг великим компаніям – Paragon. Paragon – велика компанія, яка спеціалізується на послугах маршрутизації і оптимізації доставки, яка була створена ще у 1991 році у Великобританії і з того часу стала однією з провідних компаній у галузі логістичного програмного забезпечення. Сьогодні Paragon

використовується багатьма великими міжнародними компаніями в різних галузях: від роздрібної торгівлі до охорони здоров'я та надання кур'єрських послуг[12].

Дана система є ефективною та має складні алгоритми для побудови багатьох маршрутів із врахуванням різних параметрів. Дана система надає найширший спектр різних умов для побудови маршрутів, такі як відстань, час, витрати на паливо, індивідуальні вимоги клієнтів, пріоритетність доставок, часові вікна, обмеження по вантажопідйомності, різні типи транспортних засобів, різна кількість депо, специфічні умови перевезення товару та інші.

Інтерфейс системи Paragon зображено на рисунку 1.4

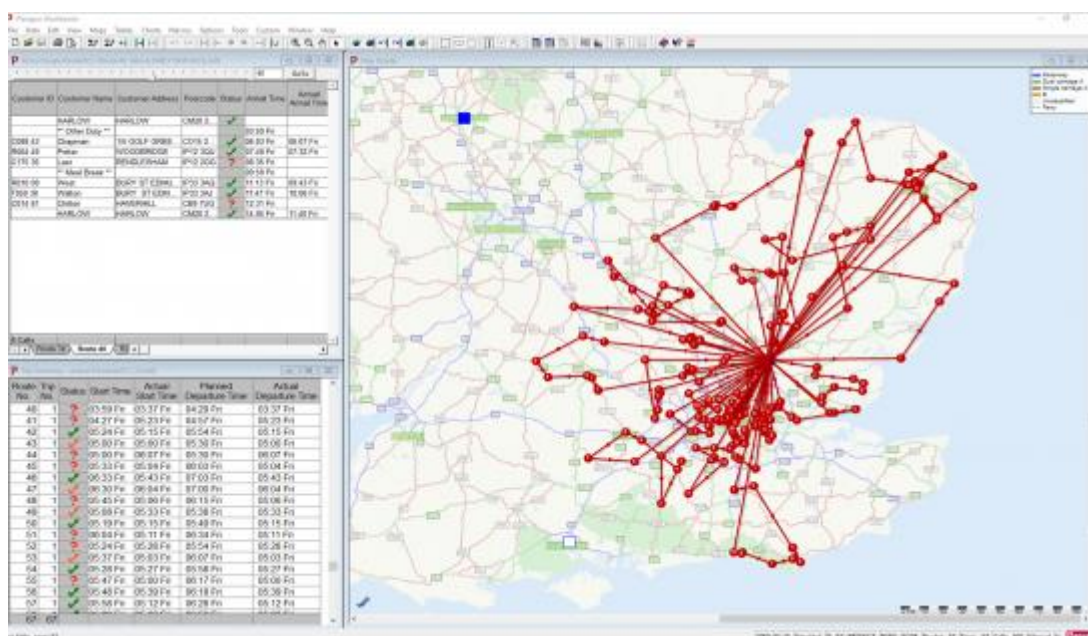


Рисунок 1.4 – Інтерфейс системи Paragon

Система Paragon має обширний функціонал та надає велику кількість послуг, та вона розвивається вже більше 30 років, тому має багато переваг перед іншими системами. Основні переваги Paragon:

- велика кількість параметрів при налаштуванні маршрутів;
- підтримка великої кількості транспортних засобів;
- Paragon надає докладну аналітику про виконання маршрутів, витрати на паливо, продуктивність водіїв і дотримання графіків доставки;

- присутня інтеграція з іншими системами, які використовуються наприклад на складах;

- Paragon підтримує маршрутизацію на міжнародному рівні, враховуючи особливості різних країн, такі як обмеження на транспорт, платні дороги та митні вимоги.

Paragon має багато переваг перед іншими додатками проте також має певні недоліки, що не дозволяють її використовувати всюди:

- висока вартість через орієнтацію на великі компанії;
- складність у використанні;
- неадаптивність для малого бізнесу через складність у масштабуванні проекту;

- відсутність мобільного додатку та залежність від інтернет-з'єднання;
- потреба в регулярному оновленні даних;
- система не створює автоматично альтернативні маршрути і фокусується на створенні одного маршруту, а створення альтернативних маршрутів потрібно виконувати вручну.

Отже, Paragon це потужне рішення для планування та оптимізації маршрутів, яке підходить для середніх і великих компаній зі складними логістичними процесами.

### 1.3 Визначення основних метрик при побудові оптимальних маршрутів

На основі аналізу існуючих логістичних систем у попередньому пункті можна виділити певні метрики, за допомогою яких визначають ефективність логістичних систем сьогодні у побудові маршрутів і які потрібно брати до уваги при розробці. Серед основних таких властивостей, на які потрібно орієнтуватись при розробці можна виділити наступні:

- швидкість побудови маршруту;
- можливість побудови маршрутів для одного та для кількох транспортних засобів;

- можливість проставити часові вікна для замовлень;
- можливість створення альтернативних маршрутів;
- врахування обмеження на вантажопідйомність транспорту;
- врахування пріоритетності маршрутів;
- можливість зміни маршруту при надзвичайних ситуаціях або перебудова маршруту для певного водія;
- врахування дорожніх умов;
- обмеження на максимальну довжину маршруту.

Вище зазначені умови є ключовими, які потрібно враховувати при побудові маршрутів, проте також існують деякі додаткові умови, на які також слід звернути увагу. Це врахування погодних умов, можливість застосування різних транспортних засобів, можливість використання кількох різних відправних точок, врахування можливих повернень замовлень на склад та інші додаткові умови.

Отже, на основі аналізу систем було визначено ряд метрик та умов на які насамперед потрібно звертати увагу при розробці алгоритму оптимізації маршрутів та безпосередньо логістичної системи.

## Висновки до розділу 1

У цьому розділі дисертації було оцінено поточний стан логістики у світі, виокремлено ключові напрямки розвитку логістичних систем сьогодні та наведено проблемні аспекти, з якими стикаються сьогодні та які мають вирішувати логістичні системи.

Також було проаналізовано чотири системи-аналоги, які надають схожі послуги, а саме побудову та оптимізацію маршрутів: MyRouteOnline, Route4Me, Paragon, OptimoRoute. Було визначено переваги та недоліки кожної з систем та на основі цього всього було сформовано основні метрики, які використовуються при побудові маршрутів.

## 2 ТЕОРЕТИЧНІ ЗАСАДИ ОПТИМІЗАЦІЇ МАРШРУТІВ

### 2.1 Теорія графів та маршрутизація

Теорія графів є фундаментальною наукою, що досліджує властивості графів – математичних структур, які моделюють множину об'єктів (вершин) і зв'язків між ними (ребер). У контексті маршрутизації графи використовуються для моделювання транспортних мереж, де вершини представляють пункти призначення (міста, склади тощо), а ребра - шляхи між ними [14,15].

Приклади графів можна знайти в різноманітних сферах науки та практики. Вони природно виникають у таких випадках, як мережі доріг, трубопроводів, електричні кола, структурні формули хімічних сполук та блок-схеми програм, і можуть бути легко побачені.

В математиці існує безліч причин для виникнення графів. Найочевиднішим прикладом є будь-який багатогранник у тривимірному просторі. Вершини та ребра багатогранника можуть бути інтерпретовані як вершини та ребра графа, при цьому ми абстрагуємося від їхнього просторового розташування і зосереджуємося лише на зв'язках між вершинами. На рисунку 2.1 представлені три різні способи візуалізації одного і того ж графа тривимірного куба.

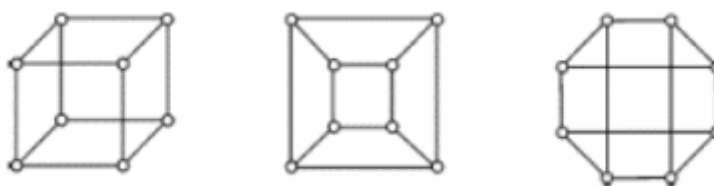


Рисунок 2.1 – Способи зображення графів

Якщо у графі є ребро  $(a, b)$ , то кажуть, що вершини є суміжними в цьому графі, то ребро інцидентне кожній з цих вершин  $a$  і  $b$ , а кожна з вершин інцидентна цьому ребру.

Множина всіх вершин графа  $a$ , що суміжні з даною вершиною, називається околom цієї вершини і позначається через  $V(a)$ .

У багатьох практичних задачах ефективним способом подання графа є використання списків суміжності. Ці списки можуть бути реалізовані різними способами за допомогою конкретних структур даних, але суть полягає в тому, що для кожної вершини перераховуються всі вершини, суміжні з нею, тобто елементи множини  $V(a)$ . Такий метод подання графа дозволяє швидко переглядати окіл вершини.

Часто, особливо при моделюванні реальних систем за допомогою графів, вершинам або ребрам (або і тим, і іншим) присвоюються певні числові значення. Природа цих чисел може бути дуже різноманітною. Наприклад, якщо граф моделює залізничну мережу, то число, приписане ребру, може вказувати на довжину перегону між двома станціями, або на максимальну вагу потягу, допустиму для цієї ділянки шляху, або на середню кількість потягів, що проходять через цю ділянку за добу тощо. Незалежно від того, що саме означають ці числа, існує традиція називати їх вагами, а графи, де задані ваги для вершин і/або ребер – зваженими графами.

Маршрут у графі являє собою спосіб проходження по ребрах від однієї вершини до іншої. Це послідовність  $v_1 e_1 v_2 \dots v_n e_n v_{n+1}$ , де ми починаємо з вершини  $v_1$  і прямуємо до вершини  $v_{n+1}$ . Кожна вершина  $v_i$  з'єднана з наступною вершиною  $v_{i+1}$  за допомогою ребра  $e_i$ , і ребро  $e_i$  можна записати як  $e_i = (v_i, v_{i+1})$ . Для графа, зображеного на рис. 2.2, маршрут  $v_1 e_1 v_2 e_1 v_1$  представляє рух патрульного автомобіля і демонструє, що вершини та ребра можуть повторюватися в маршруті.

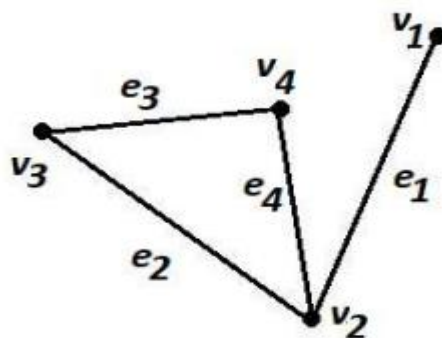


Рисунок 2.2 – Маршрут у простому графі

Якщо маршрут є замкнутим, то перша та остання вершини збігаються:  $v_1 = v_n + 1$ . Довжину маршруту визначають за кількістю ребер, які його формують. Якщо всі ребра в маршруті є унікальними, але сам маршрут замкнутий (тобто вершини можуть повторюватися), його називають циклом. Наприклад, на рис. 2.2  $v_2 e_2 v_3 e_3 v_4 e_4 v_2$  утворює цикл.

Відстань  $d(u, v)$  між будь-якими двома вершинами  $u$  і  $v$  у зв'язному графі визначається як найкоротший маршрут між ними, який дорівнює кількості ребер у цьому маршруті. У простому графі відстань від вершини  $u$  до вершини  $v$  і навпаки є однаковою.

Ексцентриситет  $e(u)$  вершини  $u$  у зв'язному графі – це відстань від даної вершини до найвіддаленішої вершини. Іншими словами, ексцентриситет є максимальним маршрутом для даної вершини у зв'язному графі, або найдовшим серед найкоротших маршрутів від цієї вершини до всіх інших.

Існує безліч методів представлення графів, що залежать від специфіки їх застосування. Наприклад, у сфері інформаційних технологій часто застосовують матричне представлення графів.

Матриця суміжності є одним із найбільш поширених способів подання графів. У цій матриці кожна вершина графа представлена як рядок і стовпець, а значення в матриці вказує на наявність або відсутність ребра між відповідними вершинами. Наприклад, якщо в графі існує ребро між вершинами  $i$  та  $j$ , то елемент матриці  $A[i][j]$  матиме значення 1, інакше 0. Цей спосіб представлення графів особливо ефективний для щільних графів, де більшість можливих ребер присутні, оскільки дозволяє легко перевіряти наявність ребра між будь-якими двома вершинами. Однак для розріджених графів цей метод може бути неефективним через великий обсяг займаної пам'яті.

Графічне представлення графів є візуальним способом подання структур даних, де вершини і ребра графа відображаються як точки і лінії відповідно. Це наочне подання дозволяє легко інтерпретувати структуру графа та взаємозв'язки між його елементами. Існує кілька способів графічного представлення графів, кожен з яких має свої переваги і використовується для різних типів графів.

Існує кілька способів організації вершин на площині. Наприклад, у випадку дерев, вершини часто розташовуються на різних рівнях, з коренем у верхній частині і гілками, що йдуть вниз. Для графів, що представляють мережі або зв'язки, вершини можуть бути розташовані у вигляді кругової діаграми або розподілені випадковим чином, щоб мінімізувати перетин ребер.

У зважених графах ваги ребер можуть бути вказані числовими значеннями поруч із ребрами або різною товщиною або стилем ліній. Це дозволяє відразу побачити, які з'єднання важливіші або мають більшу «вагу».

Кольори можуть бути використані для кращого розрізнення різних частин графа. Наприклад, у орієнтованих графах різні кольори стрілок можуть вказувати на різні типи зв'язків, а вершини можуть бути зафарбовані різними кольорами для позначення різних кластерів або груп.

Маршрутизація графів - це процес знаходження оптимального шляху або маршруту між двома або більше вузлами (вершинами) у графі. Графи складаються з вузлів, з'єднаних ребрами (зв'язками), і маршрутизація визначає, як пересуватися по цих зв'язках, щоб досягти кінцевого пункту з мінімальними витратами, наприклад, за найкоротший час, найменшу відстань, або з урахуванням інших критеріїв, таких як надійність або пропускна здатність. Цей процес часто використовується в мережах передачі даних, транспортних системах та інших сферах, де важливо визначити найкращий шлях через складні мережі[13,15].

Існує кілька методів представлення графів, які обираються в залежності від сфери їхнього застосування. У галузі інформаційних технологій часто використовується матричне представлення графів.

Граф визначається як дві множини  $G = \langle V, E \rangle$ , де  $V$  – множина вершин  $V = \{a, b, c, d\}$ , а  $E$  – множина ребер  $E = \{(a, b), (a, c), (b, d), (b, c)\}$ .

Представлення графа у вигляді зображення є популярним через свою зручність і наочність. Для графа  $G$ , описаного в першому способі, можливі зображення, як показано на рисунку 2.3: ліворуч (1) або праворуч (2). Обидва ці зображення графа  $G$  є коректними, і, по суті, існує нескінченна кількість таких варіантів представлення одного й того ж графа.

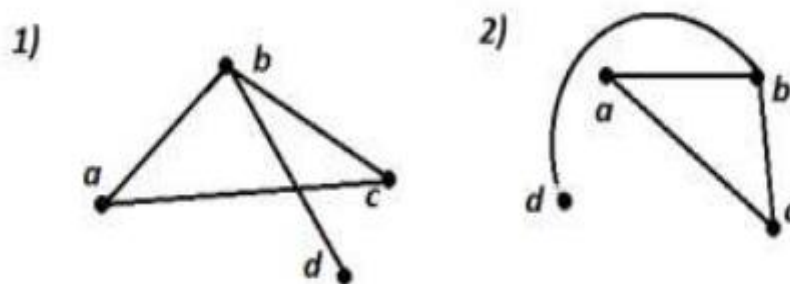


Рисунок 2.3 – Два зображення графа G

У задачі маршрутизації теорія графів дозволяє представити транспортну мережу як граф, де вершини це точки доставки (клієнти, депо), а ребра - дороги або маршрути між ними. Оптимізація маршрутів включає знаходження найкоротших або найвигідніших шляхів у графі. Теорія графів враховує ваги ребер (відстань, вартість) та обмеження, наприклад, місткість транспортних засобів, що моделюється через розширення графа. Всі математичні моделі маршрутизації, зокрема для задачі маршрутизації транспортних засобів з обмеженням на вагу чи багатозадачних задачі маршрутизації транспортних засобів, формуються на основі графів, де рішення оптимізуються через розв'язання підграфів або мінімізацію вартості циклів.

Отже, теорія графів є фундаментальною основою для моделювання, аналізу та оптимізації транспортних задач. За допомогою теорії графів представляються усі задачі маршрутизації та візуалізуються усі рішення.

## 2.2 Класифікація методів для задачі створення та оптимізації маршрутів

Задача маршрутизації транспортних засобів, також відома як Vehicle Routing Problem (VRP) існує вже багато десятиліть і впродовж цього часу було створено багато різноманітних методів та алгоритмів, які повинні були вирішити дану проблему. Основною метою VRP є мінімізація витрат на доставку товарів чи надання послуг декільком клієнтам, які розташовані в різних точках, з використанням певної кількості транспортних засобів з обмеженою

вантажопідйомністю. Всі алгоритми та методи можна поділити на кілька основних груп: точні методи, евристичні алгоритми, метаевристичні алгоритми та гібридні алгоритми. Крім того, останнім часом набирають популярність алгоритми на основі роботи нейронних мереж. Розглянемо детальніше всі ці методи.

Задачі маршрутизації транспортних засобів почали з'являтися ще в середині минулого століття і тоді вони не мали таку велику розмірність як сьогодні і для їх вирішення розроблялись різні точні методи, які могли давати найоптимальніший маршрут для невеликої задачі.

До основних точних методів для розв'язання VRP належать:

- метод гілок та меж (Branch and Bound);
- динамічне програмування;
- метод генерації стовпців (Column Generation);
- метод гілок, цін і відсікань (Branch-Price-and-Cut).

Розглянемо поверхнево суть цих методів.

Метод гілок та меж будує дерево можливих розв'язків, де кожна вершина представляє часткове рішення. Гілки дерева відображають вибір відвідування клієнтів транспортними засобами. На кожному кроці розраховуються нижні межі для гілок дерева, і якщо вони перевищують відомий мінімум, ця гілка відсікається. Цей метод є одним із перших точних методів – він був запропонований ще у 1960-му році[16].

Використання методу динамічного програмування для VRP обмежене задачами невеликого розміру, але для простих варіантів VRP воно було одним із основних підходів у 1970-х роках і є одним з кращих методів для малих розмірностей. Цей метод розподіляє задачу на менші підзадачі, розв'язуючи кожную з них і комбінуючи результати для знаходження оптимального маршруту[18].

Метод генерації стовпців розробили у 1990-х роках. Спочатку застосовувався для задач, пов'язаних з розподілом ресурсів, він став одним з ефективних підходів до VRP у 2000-х роках. Метод дозволяє працювати з великими наборами даних, і в поєднанні з іншими методами він і досі широко використовується. Метод починається з невеликої підмножини змінних і поступово додає нові змінні, які

потенційно можуть покращити розв'язок. Це дозволяє розв'язувати великі задачі VRP, обмежуючи розмір моделі на кожному кроці[19].

Метод цін, гілок та відсікань для вирішення задачі маршрутизації транспортних засобів є розширенням класичних методів гілок та відсікань (Branch and Cut) та методу генерації стовпців (Column Generation). Цей підхід комбінує переваги обох методів, що дозволяє ефективно обробляти складні та великі задачі VRP. Вибраний метод є одним із найсучасніших і найефективніших методів для точного розв'язання VRP з різними обмеженнями. Метод починається з використання методу генерації стовпців для побудови базових розв'язків, а потім йде відкидання зайвих рішень та покращення розв'язку за допомогою методу гілок та відсікань[20].

Існують ще інші точні методи для розв'язання задачі VRP однак їх використовують не так часто. Точні методи забезпечують оптимальні розв'язки VRP, однак при збільшенні розміру задачі вони стають менш ефективними через експоненційне зростання обчислювальної складності. Для великих реальних задач в основному сьогодні використовуються наближені методи.

Велику частку серед наближених методів займають евристичні методи. Евристичні алгоритми - це алгоритми, які побудовані на основі специфічних правил або інтуїтивних принципів, адаптованих до конкретної задачі. Вони забезпечують швидкий результат, але не завжди гарантують якість або оптимальність розв'язку. Евристики часто є «жадібними» і можуть мати обмеження щодо масштабування на більші задачі. Серед основних евристичних алгоритмів можна виділити наступні:

- метод найближчого сусіда (Nearest Neighbour);
- алгоритм Кларка-Райта (Clarke-Wright Savings);
- алгоритм вставки по найбільшій економії (Largest Savings Heuristic).

Розглянемо коротко суть цих методів.

Один з перших методів для задачі комівояжера, який з'явився у 1950-х роках був метод найближчого сусіда. Він надає швидке рішення, але може призвести до дуже неоптимальних маршрутів у складних випадках. Алгоритм починає з

початкової точки (депо або випадково обраного клієнта) і на кожному кроці додає до маршруту найближчого клієнта, поки всі точки не будуть відвідані. Якщо вантажопідйомність автомобіля стає переповненою то алгоритм розпочинає побудову маршруту для другого автомобіля. Даний алгоритм є мало ефективним сьогодні і використовується в цілях навчання[21].

Розроблений Джеффри Кларком і Джоном Райтом у 1964 році, алгоритм Кларка-Райта став класичним і використовувався як базовий підхід для вирішення VRP. Основна ідея цього алгоритму полягає в тому що спочатку кожен клієнт має окремий маршрут. Алгоритм обчислює економію від об'єднання двох маршрутів і поступово з'єднує ті маршрути, що забезпечують найбільшу економію[22].

Багато початкових евристичних методів з часом вдосконалювались. Так алгоритм вставки по найбільшій економії було вдосконалено на основі методу Кларка-Райта. Даний алгоритм має схожу структуру як і алгоритм Кларка-Райта. На кожному кроці алгоритм обирає дві точки для об'єднання, що дають найбільшу економію. Це подібно до алгоритму Кларка-Райта, але з більш жорстким підходом до оптимізації економії на кожному кроці[24].

Евристичні методи перераховані вище були створені спеціально для задачі транспортної маршрутизації та використовуються для її вирішення проте є більш загальні підходи, які називають метаевристичними алгоритмами. Метаевристичні методи - більш загальні підходи, що не прив'язані до конкретної задачі. Вони мають механізми для уникнення локальних мінімумів і більш ефективно досліджують простір рішень, що дозволяє їм знаходити кращі наближені рішення. Вони зазвичай працюють трохи довше ніж евристичні методи проте забезпечують кращі розв'язки[2].

Виділимо основні метаевристичні алгоритми, які використовуються для вирішення задач маршрутизації транспортних засобів.

Одним з найперших таких алгоритмів був генетичний алгоритм. Натхненний теорією еволюції, Джон Холанд розробив генетичний алгоритм у 1970-х роках. Пізніше даний алгоритм став популярним як універсальний метаевристичний підхід до різних задач оптимізації, включаючи задачу маршрутизації транспортних

засобів. Генетичний алгоритм використовує популяцію рішень, які схрещуються та зазнають мутацій з метою пошуку кращих розв'язків. Після кожної ітерації відбувається відбір найкращих особин популяції, де найкращі розв'язки залишаються для наступного покоління[26].

Пізніше було представлено метод симуляції відпалу. Даний метод був запропонований у 1983 році Скоттом Кіркпатріком і його колегами та отримав назву через явище у фізиці – відпал металу, суть якого і була покладена в основу даного алгоритму. Він став популярним для задач оптимізації завдяки здатності ефективно уникати локальних мінімумів. Алгоритм імітує процес відпалу металу, де спочатку дозволяються менш оптимальні зміни, щоб уникнути локальних мінімумів, а потім поступово знижуються температура(один з основних параметрів алгоритму), змінюється розв'язок і поступово разом із температурою зменшується ймовірність прийняття зміненого розв'язку[27].

Ще одним важливим метаевристичним алгоритмом є алгоритм оптимізації мурашиними колоніями. Цей алгоритм був розроблений Марко Доріго у 1992 році. Суть цього алгоритму полягає у імітації поведінки мурах у пошуку найкоротшого шляху до їжі. Мурахи залишають феромони на своєму шляху до їжі, а цей алгоритм використовує імовірність вибору шляху на основі накопичених феромонів та відстані. З часом алгоритм посилює найкоротші маршрути, так як там збільшується кількість феромонів[28].

Також популярним підходом є розроблений Фредом Гловером у 1986 році, табу-пошук. Він став популярним методом для комбінаторних задач, зокрема для VRP через його просту природу та ефективність. Ідеєю цього алгоритму є те що після кожної ітерації алгоритм додає рішення на поточній ітерації до списку заборонених (ставить табу), щоб уникнути повернення до нього і уникнути локальних мінімумів і продовжує пошук серед ще недосліджених маршрутів. Якщо кращі не знаходяться то вибирає найкращий який був досліджений[29].

Іншим більш сучасним підходом є алгоритм рою частинок. Як і більшість метаевристичних алгоритмів цей метод був розроблений Расселом Еберхартом та Джеймсом Кеннеді у 1995 році, натхненний природою, а саме поведінкою птахів і

риб. PSO використовує групу агентів(частинок), які рухаються в просторі рішень, змінюючи своє положення на основі власного досвіду та інформації від інших частинок, щоб поступово наближатися до оптимального рішення[30].

Хоч евристичні, метаевристичні та точні методи покращуються із часом та розвиток технологій також сприяє появі нового типу алгоритмів. Так останнім часом для вирішення задачі транспортної маршрутизації все частіше застосовуються алгоритми штучного інтелекту та нейронних мереж. Вони дозволяють розв'язувати складні варіанти VRP, враховуючи численні обмеження та велику кількість клієнтів, і з часом показали суттєве зростання в ефективності та адаптивності. Завдяки потужності сучасних обчислювальних ресурсів і доступності великих обсягів даних, алгоритми штучного інтелекту демонструють високий рівень точності та оптимізації. Нейронні мережі здатні самонавчатися на основі історичних даних і ефективно знаходити закономірності, що дозволяє враховувати всі помилки попередніх алгоритмів та уникати їх при побудові маршрутів. Ця здатність робить подальше використання даних алгоритмів перспективним[31]. На сьогоднішній день можна виділити такі приклади алгоритмів на основі штучного інтелекту та нейронних мереж:

- генеративно-змагальні мережі (Generative Adversarial Networks, GANs);
- графові нейронні мережі (Graph Neural Networks, GNNs);
- рекурентні нейронні мережі з підкріпленням (Reinforcement Learning with Recurrent Neural Networks, RNN-RL).

Генеративно-змагальні мережі були запроваджені Яном Гудфелоу в 2014 році. GANs складаються з двох нейронних мереж - генератора та дискримінатора. Генератор створює можливі маршрути, а дискримінатор оцінює їхню якість. Поступово генератор навчається створювати маршрути, що мають високу якість, наближаючись до оптимальних рішень. Використання GANs для VRP дозволяє отримувати різні варіанти маршрутів на основі певних обмежень. GANs допомагають у випадках, коли потрібно швидко створити оптимізовані маршрути.

Графові нейронні мережі працюють з даними, представленими у вигляді графів, де кожен клієнт та маршрут представляються як вершини і ребра відповідно. Вони дозволяють обробляти інформацію про відносини між точками та розробляти ефективні маршрути на основі географічної та логістичної структури. Ці мережі навчаються на історичних даних маршрутизації і прогнозують найкращі шляхи, враховуючи динамічні обмеження.

Рекурентні нейронні мережі з підкріпленням комбінують методи навчання з підкріпленням із рекурентними нейронними мережами, щоб покращити якість маршрутів через ітеративне навчання. Агент навчання з підкріплення вивчає найкращі дії, отримуючи винагороди за покращення маршрутів, а рекурентна нейронна мережа дозволяє обробляти послідовні дані та передбачати оптимальні маршрути. Метод RNN-RL підходить для VRP із змінними умовами, де необхідно адаптувати маршрути в реальному часі. Цей метод використовується для динамічної маршрутизації таксі та доставки в умовах міського трафіку.

Хоч багато з вказаних методів та алгоритмів показують гарні результати, такі як евристичний алгоритм з використанням евристики на основі розподілу і злиття маршрутів та метаевристичний метод бджолиного рою, та мають гарну перспективу у майбутньому як методи штучного інтелекту з використанням нейронних мереж, проте найкраще себе показують гібридні алгоритми. Гібридні алгоритми поєднують у собі відразу кілька підходів, деколи навіть із різних класів алгоритмів(можуть поєднувати точні і метаевристичні підходи, евристичні і метаевристичні тощо). Гібридні алгоритми сьогодні складають великий відсоток всіх алгоритмів для розв'язання задачі транспортної маршрутизації. Крім того, саме вони показують найкращі результати по співвідношенню часу і точності розв'язку. Гібридні алгоритми своїм поєднанням кількох методик дозволяють поєднати сильні сторони різних методів компенсуючи їхні недоліки. Також вони добре поєднують глобальний та локальний пошуки, що покращує розв'язки на різних рівнях. І найголовнішою перевагою гібридних алгоритмів є те що вони адаптивні до різних умов та змін в середовищі. У наступному розділі проаналізуємо

основні гібридні алгоритми для визначення всіх їх плюсів та мінусів для створення власного гібридного алгоритму маршрутизації.

Отже, для вирішення задачі транспортної маршрутизації використовуються різні методи та підходи. Всі методи та алгоритми поділяють на точні, евристичні, метаевристичні, методи з використанням штучного інтелекту та гібридні. Кожен клас методів включає методи, які знайшли своє практичне застосування і використовуються для різних задач маршрутизації, проте поєднання різних методів у нові гібридні алгоритми є найефективнішим сьогодні та дозволяє знаходити розв'язки швидко та якісно.

## 2.3 Аналіз гібридних алгоритмів задачі маршрутизації транспортних засобів

### 2.3.1 Аналіз генетичного алгоритму з локальним пошуком

Розглянемо основні гібридні алгоритми для вирішення задачі транспортної маршрутизації та проведемо їх порівняльний аналіз для визначення найкращого співвідношення для власного гібридного алгоритму.

Одним з перших гібридних алгоритмів, який почав використовуватись для вирішення задачі маршрутизації був генетичний алгоритм із локальним пошуком.

У 1990-х роках почалося активне впровадження локальних пошукових методів, таких як 2-opt, 3-opt та інші евристики, у структуру генетичних алгоритмів. Виник гібридний підхід, який отримав назву міметичний алгоритм або генетичний алгоритм з локальним пошуком. Його успіх пов'язаний з ефективним використанням генетичних алгоритмів у глобальному пошуку і локального пошуку для покращення розв'язків[32].

Генетичний алгоритм використовується для глобального пошуку, генеруючи початкову популяцію можливих маршрутів, які піддаються операціям схрещення і мутації. Потім до кожного нового рішення застосовується локальний пошук, щоб покращити якість рішень.

Генетичний алгоритм із локальним пошуком має чимало переваг, серед яких ключовими є те, що алгоритм добре працює для великих задач, надаючи близькі до

точних рішення та поєднання методик ефективного глобального пошуку генетичного алгоритму з покращенням рішень на локальному рівні

Проте він має свої недоліки, що не дозволяє його використовувати сьогодні. Зокрема, генетичний алгоритм потребує значних обчислювальних ресурсів та даний алгоритм може затримуватись на локальних мінімумах, так як локальний пошук не завжди достатньо ефективний.

Отже, даний алгоритм був проривним в кінці минулого століття і активно використовувався, проте сьогодні високі обчислювальні витрати на генетичний алгоритм не компенсуються локальним пошуком, який не завжди є ефективним при вирішенні задач.

### 2.3.2 Аналіз гібридного мурашиного алгоритму з імітацією відпалу

Іншим популярним гібридним алгоритмом, який поєднував дві відомих стратегії був мурашиний алгоритм з імітацією відпалу (Ant-Colony with Simulated Annealing). Він був створений на початку цього століття і швидко став використовуватись для вирішення задач транспортної маршрутизації. Поєднання мурашиного алгоритму та імітаційного відпалу було покликано покращити точність та завадити виникненню локальних мінімумів.

Суть цього алгоритму полягала в тому, що мурашиний алгоритм спочатку будує маршрути, покращуючи їх на основі феромонів, а потім алгоритм імітації відпалу оптимізує ці маршрути, дозволяючи приймати менш оптимальні рішення для уникнення локальних мінімумів.

Перевагами даного алгоритму є те, що алгоритм імітації відпалу є чудовим інструментом для уникнення локальних мінімумів. Даний алгоритм розроблений спеціально для великорозмірних задач. Крім того, мурашиний алгоритм ефективний у знаходженні глобальних оптимумів. Крім того, алгоритм може видавати кілька альтернативних розв'язків.

Проте цей алгоритм, як і багато варіацій мурашиного алгоритму потребує точного налаштування параметрів для ефективного роботи, а зважаючи на те що

алгоритм імітації відпалу є також нелегким для налаштування – це робить налаштування гібридного алгоритму надзвичайно важким і підбір правильних значень параметрів займає багато часу.

Отже, гібрид мурашиного алгоритму та алгоритму імітації відпалу є ефективним алгоритмом для вирішення задач VRP, проте велика кількість налаштовуваних параметрів робить важким його застосування на практиці[33].

### 2.3.3 Аналіз гібридного алгоритму табу-пошуку і методу генерації стовпців

Ще одним гібридним алгоритмом, який поєднує два сучасних методи оптимізації для VRP, є гібридний алгоритм табу-пошуку з методом генерації стовпців (Tabu Search and Column Generation Algorithm). Цей алгоритм був запропонований у 2011. Цей підхід дозволяє зменшити кількість варіантів, з якими працює алгоритм, зберігаючи можливість знайти оптимальні маршрути.

Головною ідеєю алгоритму є те що, табу-пошук забезпечує ефективний локальний пошук, уникаючи повернення до нещодавно відвіданих рішень, тоді як метод генерації стовпців дозволяє динамічно збільшувати кількість маршрутів, оптимізуючи простір рішень[34].

Переваги цього алгоритму полягають у тому, що табу-пошук дозволяє уникати локальних мінімумів, а метод генерації стовпців ефективно додає нові маршрути, збільшуючи гнучкість рішення. Крім того, даний алгоритм добре підходить для складних задач з великою кількістю клієнтів та великою кількістю обмежень, дозволяючи знаходити гарні рішення навіть за багатьох різних умов, які накладаються на задачу.

Проте також є і недоліки. Вони включають можливі проблеми з пам'яттю, так як при великих обсягах інформації обсяг табу-таблиць може бути великим. Також алгоритм вимагає ретельного налаштування частоти оновлення таблиці табу та кількості колонок, що можуть бути додані на кожному етапі.

Отже, гібрид табу-пошуку та колонного породження є ефективним методом для задач VRP, особливо коли потрібно уникати локальних мінімумів.

### 2.3.4 Аналіз гібридного алгоритму штучного бджолиного рою з локальним пошуком

Гібридний алгоритм на основі штучного бджолиного рою та локального пошуку (ABC-LS) поєднує переваги глобального пошуку, властивого алгоритму штучного бджолиного рою, з ефективністю локальної оптимізації, яку забезпечують різні методи локального пошуку. Алгоритм ABC-LS став популярним у вирішенні задач транспортної маршрутизації, оскільки бджолиний рій ефективно досліджує простір рішень, а локальний пошук дозволяє вдосконалювати кожне отримане рішення, що допомагає знаходити ефективні розв'язки задач різної складності[35].

Суть цього алгоритму полягає в тому, що алгоритм бджолиного рою виконує глобальний пошук, знаходячи області з потенційно хорошими рішеннями, після чого локальний пошук покращує кожне знайдене рішення, щоб забезпечити високу точність та стабільність кінцевого результату.

Перевагами алгоритму ABC-LS є те що він може досліджувати великий простір рішень, знаходячи перспективні області. Це підвищує ймовірність знайти нові рішення. Виконання локального пошуку після глобального пошуку робочих бджіл дозволяє алгоритму уникати локальних мінімумів і покращувати якість знайдених рішень. Також оскільки робота бджіл відбувається незалежно, ABC-LS легко піддається паралельному виконанню, що може суттєво зменшити час роботи алгоритму.

Недоліком алгоритму ABC-LS є сильна залежність від налаштування початкових параметрів, таких як кількість бджіл, частота оновлення маршрутів і критерії вибору рішень. Неправильне налаштування може призвести до погіршення якості рішень

Отже, гібридний алгоритм ABC-LS є ефективним підходом для вирішення задач VRP, оскільки забезпечує широкий глобальний пошук і детальну локальну оптимізацію. Однак його практичне застосування вимагає ретельного

налаштування параметрів та значних обчислювальних ресурсів для досягнення високої ефективності.

## Висновки до розділу 2

У цьому розділі було розглянуто теоретичні аспекти задачі маршрутизації транспортних засобів, зокрема зв'язок теорії графів та маршрутизації транспортних засобів.

Також було розглянуто основні методи та підходи із прикладами для вирішення задачі маршрутизації транспортних засобів. Зокрема, було виділено чотири основні групи методів: точні, евристичні, метаевристичні та гібридні методи. З аналізу різних методів та алгоритмів було з'ясовано, що гібридні алгоритми є найкращими для розв'язання задач маршрутизації. Також було наведено порівняльний аналіз кількох гібридних алгоритмів для визначення оптимальної стратегії для створення власного гібридного алгоритму на основі поєднання різних метаевристичних підходів.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ТА РОЗРОБЛЕННЯ ГІБРИДНОГО АЛГОРИТМУ БАГАТОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ

#### 3.1 Математична постановка задачі

Алгоритм оптимізації маршрутів у системі повинен вирішувати задачу мінімізації витрат на шляхи при обмеженнях на вантажопідйомність транспорту, відомої як Capacitated Vehicle Routing Problem (CVRP), де необхідно враховувати побудову кількох можливих маршрутів.

Сформуємо змістовну постановку задачі маршрутизації транспортних засобів з обмеженою вантажопідйомністю. Вхідними даними задачі є: множина клієнтів  $V = \{1, 2, \dots, N\}$ , кожен з цих клієнтів має певну потребу  $q_i$ , яку потрібно задовільнити. Також є матриця відстаней  $c_{ij}$ , яка визначає відстань або вартість переміщення від клієнта  $i$  до клієнта  $j$  для всіх  $i, j \in \{0, 1, \dots, N\}$ . На вході також вказується точка  $0$  – депо звідки вирушають всі транспортні засоби[36].

На виході задачі необхідно отримати набір маршрутів для транспортних засобів, які відповідають таким умовам:

- всі маршрути починаються та закінчуються в депо;
- кожен клієнт  $i$  повинен бути відвіданий рівно один раз;
- сумарний попит клієнтів на кожному маршруті не має бути більшим вантажопідйомності конкретного транспортного засобу, яка дорівнює  $Q$ .

Основна ціль задачі: мінімізація загальної відстані або вартості всіх маршрутів.

Це включає в себе оптимальне поєднання між довжиною кожного маршруту та кількістю необхідних транспортних засобів. Вартість транспортування визначається як витрати, пов'язані з переміщенням вантажу від однієї точки до іншої. Тепер розглянемо математичну модель цієї задачі:

$$\sum_{i \in V} \sum_{j \in V, j \neq i} \sum_{k \in K} c_{ij} x_{ijk} \rightarrow \min, \quad (3.1)$$

$$\sum_{j \in V, j \neq i} \sum_{k \in K} x_{ijk} = 1, \text{ для } \forall i \in V, \quad (3.2)$$

$$\sum_{j \in V} x_{0jk} \leq 1, \text{ для } \forall k \in K, \quad (3.3)$$

$$\sum_{j \in V, j \neq 0} x_{jok} = \sum_{j \in V, j \neq 0} x_{0jk} \leq 1, \text{ для } \forall k \in K, \quad (3.4)$$

$$\sum_{i \in V, i \neq 0} q_i \sum_{j \in V, j \neq i} x_{ijk} \leq Q, \text{ для } \forall k \in K, \quad (3.5)$$

$$\sum_{i \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \text{ для } \forall i \in V, \forall k \in K, \quad (3.6)$$

$$x_{ijk} \in \{0,1\}, \forall (i,j) \in E, \forall k \in K, \quad (3.7),$$

де змінною є

$$x_{ijk} = \begin{cases} 1, & \text{якщо } k \text{ – ий автомобіль обслуговував маршрут } (i,j) \\ 0, & \text{якщо не обслуговував} \end{cases}$$

Відомими даними у задачі є:

$c_{ij}$  – довжина маршруту від  $i$ -го клієнта до  $j$ -го;

$q_i$  – кількість товару, який потрібно доставити до пункту  $i$ ,  $i \in N$ ;

$K$  – сумарна кількість транспортних засобів у депо;

$Q$  – вантажопідйомність кожного транспортного засобу;

$N$ ,  $n+1$  – кількість пунктів доставки, включаючи депо, з якого виїжджають транспортні засоби.

Тепер розпишемо кожен із зазначених формул 3.1-3.7.

(3.1) - цільова функція задачі, для визначення вартості всіх маршрутів, спрямована на мінімізацію витрат;

(3.2) - обмеження, яке показує, що певного  $j$ -го клієнта можна відвідати лише один раз;

(3.3) - обмеження, яке відповідає за те, що вибраний транспорт може лише один раз виїхати з депо;

(3.4) - обмеження, яке відповідає за те, що якщо автомобіль виїхав з депо то він має туди повернутись;

(3.5) - обмеження на вантажопідйомність кожного транспортного засобу; вага кожного транспорту не має перевищувати значення  $Q$ ;

(3.6) - обмеження, яке вказує на те що транспортний засіб  $k$  може виїхати від клієнта  $j$  тільки якщо він заїхав до нього;

(3.7) - обмеження, яке показує можливий діапазон значень наших змінних у задачі.

Отже, було сформульовану математичну постановку задачі, для вирішення якої буде розроблено гібридний алгоритм багатошляхової маршрутизації з можливістю створення альтернативних маршрутів. Дана задача є основною задачею у системі створення маршрутів тому побудова її математичної моделі є важливим завданням.

### 3.2 Опис гібридного генетичного алгоритму з покращеним алгоритмом імітації відпалу для багатошляхової маршрутизації

Розглянемо спочатку основні положення розробленого алгоритму. Розроблений алгоритм є гібридним алгоритмом генетичного алгоритму та покращеного алгоритму імітації відпалу з використанням оператора кросоверу. Даний алгоритм має розв'язувати задачу транспортної маршрутизації з обмеженням по вантажопідйомності кожного транспортного засобу (Capacitated Vehicle Routing Problem), математичну постановку якої ми розглянули у попередньому розділі.

Цей алгоритм використовує два метаевристичних підходи до вирішення задачі транспортної маршрутизації, а саме генетичний алгоритм та покращений алгоритм імітації відпалу. Ці два підходи самі по собі є потужними засобами для вирішення задачі CVRP, проте даний підхід покликаний об'єднати два підходи для зменшення впливу їх недоліків та примноження ефекту їхніх переваг. Як було вказано раніше у розділі 2.2 генетичний алгоритм є еволюційним методом оптимізації, який працює з популяцією потенційних розв'язків. Він використовує оператори селекції, кросоверу та мутації для еволюції популяції та пошуку оптимальних або близьких до оптимальних розв'язків. Він добре підходить для глобального пошуку в просторі рішень, але може застрягати в локальних екстремумах. Також було відзначено, що алгоритм імітації відпалу моделює процес охолодження металів для пошуку глобального мінімуму функції. Він ефективний для локального пошуку та може уникати локальних мінімумів завдяки контрольованому зниженню «температури» в процесі пошуку.

Гібридний підхід покликаний об'єднати ці два алгоритми, використовуючи генетичний алгоритм для генерування початкової популяції та глобального пошуку, а алгоритм імітації відпалу для локального вдосконалення розв'язків.

У 2010-х роках вже було розроблено гібридний алгоритм генетичного алгоритму та алгоритму імітації відпалу, який показав добрі результати для вирішення задачі у порівнянні з простим генетичним алгоритмом чи алгоритмом імітації відпалу, зокрема він набагато швидше знаходить розв'язки та вони є точнішими(рис.3.1[37]).

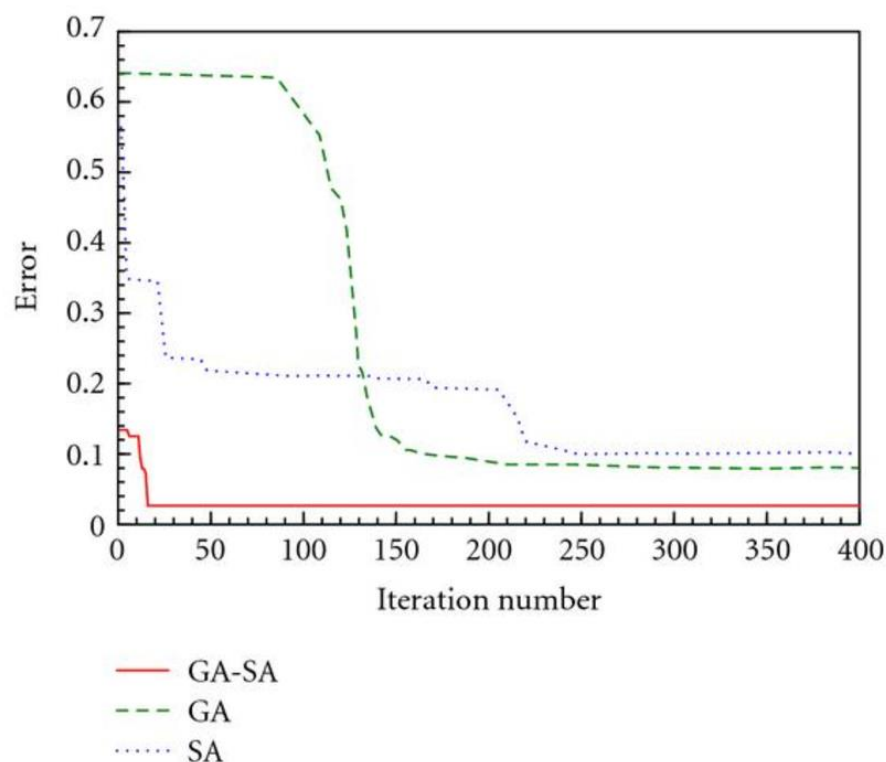


Рисунок 3.1 – Порівняння генетичного алгоритму та імітації відпалу та їх гібридного алгоритму

Проте на сьогоднішній день базовий алгоритм імітації відпалу є малоефективним, тому будемо розробляти новий алгоритм, який використовує покращену версію алгоритму імітації відпалу, яка використовує оператор схрещення для покращення розв'язків. Оператор схрещення в алгоритмі допомагає обмінюватися інформацією між розв'язками, що підвищує різноманітність та

покращує здатність уникати локальних мінімумів. Завдяки вдосконаленому алгоритму імітації відпалу, гібридний алгоритм працювати швидше і якісніше за попередній.

Таким чином, основна ідея полягає в тому, щоб скористатися глобальними пошуковими можливостями генетичного алгоритму та локальною оптимізацією знайдених розв'язків за допомогою покращеного алгоритму імітації відпалу з кросовером. Це дозволить більш ефективно та точно вирішувати CVRP, знаходячи оптимальні або близькі до оптимальних маршрути з урахуванням обмежень по вантажопідйомності транспортних засобів.

Розглянемо більш детально основні етапи роботи алгоритму, всі налаштовувані параметри, які будуть використовуватись та всі операції.

Гібридний алгоритм складається з таких етапів:

Крок 1. Ініціалізація початкової популяції.

На початковому етапі треба створити початкову популяцію – кілька розв'язків задачі з яких будуть розпочинатись основні етапи алгоритму. Створення початкової популяції для генетичного алгоритму є важливим етапом, тому що наближеність даного розв'язку до оптимального прискорює роботу алгоритму та збіжність даного алгоритму. Проте, створення початкової популяції повинно відбуватись швидко, бо це не основний етап задачі і на нього потрібно витратити мінімальну кількість часу. Зважаючи на все вищезазначене, для створення початкової популяції потрібно обрати швидкий евристичний алгоритм, який дає прийнятний результат по точності.

Згідно з дослідженнями проведеними кілька років тому однією з найкращих евристик є евристика, яка використовується у алгоритмі Кларка-Райта. Як було зазначено у розділі 2.2 це швидкий евристичний алгоритм, основна ідея якого полягає в тому щоб вираховувати найбільш економні варіанти об'єднання клієнтів у маршрути. Цей жадібний алгоритм має кілька простих та швидких етапів: створення маршрутів де є тільки клієнт і депо, розрахунок значень евристики - «збереження» від об'єднання(формула 3.8), сортування розрахованих евристик і об'єднання маршрутів за відсортованими значеннями.

$$S(i,j)=\text{dist}(0,i)+\text{dist}(0,j)-\text{dist}(i,j), \quad (3.8)$$

де  $i, j$  – клієнти, а точка  $0$  – це депо.

Початкова популяція з рішень алгоритму Кларка-Райта буде швидше сходиться до оптимуму, оскільки кожен маршрут вже є досить оптимізованим, тому кінцевий алгоритм витратить меншу кількість ітерацій для досягнення оптимального результату.

Крок 2. Відбір кращих нащадків із популяції.

На другому етапі потрібно вибрати кілька нащадків із заданої популяції, які будуть використовуватись безпосередньо у генетичному алгоритмі. Для цього існує багато різних методик, проте у цьому алгоритмі буде використовуватись універсальна методика, яка називається турнірний відбір. Її головна ідея полягає в тому що обирається 2-3 випадкових особини(розв'язки) з популяції і та, яка має найкраще значення функції пристосованості(найменша відстань всіх маршрутів) переходить до нового покоління розв'язків. Такі дії повторюються кілька разів доки не буде сформована необхідна кількість особин для подальших операцій алгоритму.

Крок 3. Схрещення

На цьому етапі відбувається схрещення вибраних «батьків» на попередньому кроці для утворення нових особин. Схрещення для задачі транспортної маршрутизації полягає в тому, що окремі клієнти обмінюються між маршрутами, або закінчення маршрутів змінюються або окремі частини маршрутів змінюються.

Для вибраного алгоритму буде використовуватись метод, який переносить частину одного маршруту до іншого. Називається цей метод Partial Mapped Crossover(PMX) або кросинговер з частковим заміщенням. Основна ідея PMX полягає в тому, щоб зберегти послідовність і позиції певних елементів від обох батьківських рішень, гарантуючи що кожен елемент (клієнт) буде представлений лише один раз у нащадку тому після перенесення частини маршруту вносяться відповідні поправки до маршрутів, щоб елементи не повторювались. Краще розуміння роботи даного методу може дати зображення на рисунку 3.2.



Рисунок 3.2 – Принцип роботи оператора схрещування, який використовується в гібридному алгоритмі

В результаті на виході після третього кроку ми отримуємо популяцію з нащадками, які були отримані під час схрещування. Вже з цією популяцією будуть проводитись подальші дії.

#### Крок 4. Застосування мутації

Наступним кроком є застосування принципу мутації. Мутація – це процес випадкових змін у розв’язках вирішуваної задачі, тобто зміні маршрутів. Мутація потрібна для уникання локальних мінімумів, забезпеченню різноманітності популяції та створенню маршрутів, які не можна отримати за допомогою схрещення. Часто мутація призводить до погіршення результатів на певній ітерації, проте може допомогти знайти кращий глобальний розв’язок у перспективі.

Мутація для задачі транспортної маршрутизації може відбуватись кількома методами:

- мутація перестановки (зміна двох клієнтів місцями);
- мутація інверсії (вибирається підмножина клієнтів із певного маршруту і їхній порядок у маршруті інвертується);
- мутація вставки(один випадковий клієнт забирається з одного місця маршруту і переставляється в друге);
- мутація між маршрутами(підмножина клієнтів забирається з одного маршруту та переноситься до іншого).

У гібридному алгоритмі буде використовуватись три типи мутацій, мутація інверсії, мутація вставки та мутація між маршрутами. Мутація інверсії може допомогти зменшити загальну відстань маршруту конкретного транспортного засобу. Мутація вставки дозволяє створювати нові комбінації клієнтів у маршрутах і може сильно змінити структуру маршруту. Мутація між маршрутами необхідна для збалансування розподілу клієнтів між транспортними засобами, особливо якщо деякі маршрути перевантажені, проте дана мутація також включає перевірку на можливу перевантаженість транспортного засобу. Під час ітерації алгоритму вибирається одна з трьох мутацій з рівномірним шансом 25 %. Також існує варіант не застосування мутації на вибраній ітерації. Ймовірність такого варіанту також складає 25 %.

Крок 5. Оцінка отриманих рішень та застосування алгоритму імітації відпалу

На п'ятому етапі алгоритму спочатку оцінюються за допомогою цільової функції всі значення з останньої популяції і вибирається кілька найкращих рішень(розв'язків з найменшою довжиною маршрутів). Вибрані розв'язки будуть покращуватись за допомогою алгоритму імітації відпалу.

Перед початком роботи алгоритму імітації відпалу встановлюються початкові значення параметрів для алгоритму. Основним параметром для алгоритму імітації відпалу є «температура». Значення цього параметру визначає ймовірність прийняття гіршого розв'язку. Також встановлюються початкові значення таких параметрів як коефіцієнт охолодження та кількість ітерацій.

Далі відбуваються ітерації алгоритму імітації відпалу. В кожній ітерації спочатку застосовуються оператори локального пошуку для покращення поточних рішень. Конкретно для вибраного алгоритму буде використовуватись зміна клієнтів у маршруті(перестановка двох клієнтів місцями). Також буде використовуватись механізм вставки клієнта у певне місце в маршруті, який допомагає знайти більш ефективні розташування клієнтів у маршруті. Крім того для створення нових варіантів маршрутів буде використовуватись механізм перемішування, суть якого в тому що береться певний набір клієнтів з маршруту і вони випадковим чином перемішуються.

Так як у гібридному алгоритмі використовується покращена версія алгоритму імітації відпалу з оператором схрещення то після створення кількох пар рішень за допомогою локальних змін відбувається процес схрещення цих рішень. Для схрещення рішень у алгоритмі імітації відпалу буде використовуватись порядковий кросинговер(Order crossover). Суть цього оператора полягає в тому що як і в кросинговері часткового заміщення, який був описаний раніше у розділі, обирається частина маршруту з одного рішення і така ж сама частина з іншого рішення. Частина маршруту з першого «батька» встає на те саме місце при формуванні першого нащадка, і аналогічно частина маршруту з другого «батька» формує основу другого нащадка. Після цього всі клієнти, для першого нащадка, які ще не були відвідані у маршруті заповнюються беручи порядок відвідин цих самих клієнтів у маршруті другого «батька». Аналогічно, довершується заповнення маршруту другого нащадка потрібними клієнтами беручи порядок цих клієнтів із першого «батька». Детальніше принцип роботи порядкового кросинговера можна побачити на рисунку 3.3.

Після застосування порядкового кросинговера йде оцінка рішень, які були утворені. При цій оцінці, якщо утворені рішення мають краще значення функції пристосованості, то вони приймаються відразу, проте якщо утворені рішення є гіршими, то вони можуть бути прийняті з певною ймовірністю, яка обраховується по формулі 3.9, де основним параметром виступає температура  $T$ .

$$P = e^{-\frac{\Delta E}{T}}, \quad (3.9)$$

де  $P$  – ймовірність прийняття гіршого розв'язку від 0 до 1,

$T$  – параметр температури,

$\Delta E$  – зміна вартості цільової функції,

$e$  – математична стала.

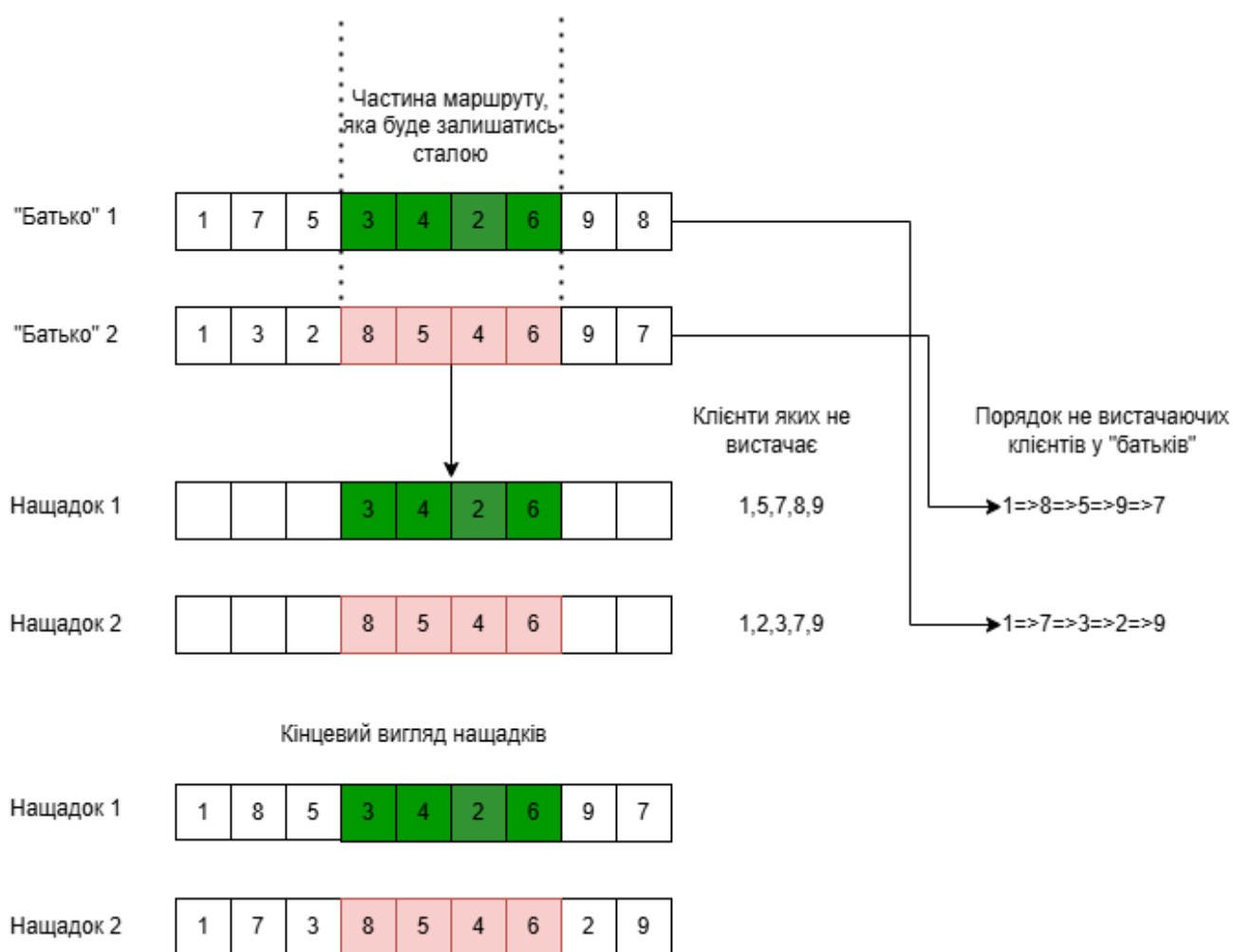


Рисунок 3.3 – Принцип роботи порядкового кросинговеру

Після оцінки всіх нових отриманих рішень потрібно оновити головний параметр алгоритму – температуру. Оновлення температури описується за допомогою формули 3.10.

$$T_{i+1} = T_i * a, \quad (3.10)$$

де  $T_{i+1}$  – значення температури оновлене,

$T_i$  – значення температури на даній ітерації,

$\alpha$  – параметр охолодження, який встановлюють в межах від 0 до 1.

Після оновлення температури окрема ітерація алгоритму імітації відпалу завершується і виконуються три перевірки на завершення алгоритму імітації відпалу і переходу до наступного кроку гібридного алгоритму. Частина алгоритму імітації відпалу завершується, якщо досягнуто мінімальної температури або пройдено максимальну кількість ітерацій або протягом певної кількості ітерацій немає покращення.

Якщо критерії зупинки не виконуються то алгоритм імітації відпалу продовжує свою роботу.

Цей етап гібридного алгоритму є досить громіздким, тому що включає в себе використання інструментів алгоритму імітації відпалу для локального покращення раніше знайдених розв'язків. Після завершення кроку 5 ми отримуємо покращені локальним пошуком розв'язки.

Крок 6. Оцінка розв'язків отриманих алгоритмом імітації відпалу та відбір нової популяції

Після отримання рішень, які були покращені за допомогою алгоритму імітації відпалу, нові рішення порівнюються з існуючими в популяції за значеннями функції пристосованості. Якщо нові алгоритми мають краще значення цієї функції то вони замінюють собою найгірші особини з популяції.

Крок 7. Перевірка умов завершення алгоритму.

Після кожної ітерації алгоритму виконується кілька перевірок на зупинку алгоритму. У цьому гібридному алгоритмі є дві умови які перевіряються: досягнуто максимально можливої відведеної для алгоритму кількості ітерацій та другою умовою є відсутність покращення розв'язку(тобто відсутність покращення значення цільової функції) протягом певної кількості ітерацій.

Крок 8. Ітераційне повторення кроків 2-6 до виконання умов зупинки алгоритму у кроці 7.

Крок 9. Вибір кращого рішення та кількох альтернативних рішень.

Після зупинки роботи алгоритму вибирається одне основне рішення, яке буде використовуватись у програмі для побудови маршруту та обираються 2-3 (залежно від налаштувань) альтернативних рішень, які при потребі користувача будуть миттєво надані. Блок-схема цього алгоритму зображена в додатку Б.

Після виконання усіх вище вказаних кроків ми отримаємо розв'язок задачі маршрутизації транспортних засобів, а саме множину автомобілів із вказаною послідовністю відвідин клієнтів для кожного автомобіля, довжину кожного маршруту та кількість вантажу, який кожний автомобіль буде перевозити згідно з потребами клієнтів.

Отже, розроблений гібридний алгоритм поєднує у собі генетичний алгоритм для глобального пошуку рішень та покращений варіант алгоритму імітації відпалу для локального покращення знайдених рішень. Загалом алгоритм складається з 9 етапів, та використовує такі параметри як температура та коефіцієнт охолодження. У алгоритмі використовуються два оператори схрещення часткове заміщення та порядковий кросинговер.

### 3.3 Адаптація алгоритму до перебудови маршрутів

Дедалі частіше сьогодні стаються ситуації коли згенерований маршрут не задовольняє користувача і потрібно перебудувати маршрут або іноді виникає потреба перебудувати маршрут для конкретного транспортного засобу, виключивши маршрут, який не задовольняє вимогам наприклад, через зміну умов дорожнього руху або нові замовлення. Зважаючи на все це, програмний продукт потрібно адаптувати до можливої перебудови маршрутів. Зокрема потрібно розглядати 2 випадки, у яких може відбуватись перебудова маршруту:

- перебудова цілого рішення;
- перебудова маршруту для конкретного транспортного засобу.

При першому варіанті потрібно перебудувати весь розв'язок, тобто надати альтернативне рішення, яке буде не набагато гірше за наданий основний розв'язок.

При запиту клієнта на альтернативне планування всіх маршрутів передбачено 2-3 (залежно від налаштувань) альтернативних рішень, які були отримані в результаті гібридного алгоритму, так як генетичний алгоритм використовує цілі популяції, то це легко зробити, взявши замість одного найкращого розв'язку кілька особин з останньої популяції. Таким чином програмний продукт потребує лише часу для графічного представлення нових маршрутів і не потрібно заново запускати алгоритм для визначення нового шляху.

Перебудова маршруту для конкретного засобу є потрібним інструментом, так як клієнт може відмінити замовлення, або дорожні умови не дозволяють добратись до певного клієнта по маршруту. Тому може знадобитись перебудова маршруту. Перебудова маршруту для конкретного транспортного засобу повинна відбуватись швидко та не сильно поступатись за точністю попередньому значенню довжини цього маршруту. Тому для вирішення цієї задачі буде використовуватись подвійний підхід, залежно від кількості клієнтів, які повинен транспортний засіб обслужити на даному маршруті. При малій кількості клієнтів(до 10) буде здійснюватися побудова нового маршруту за допомогою найшвидшого евристичного методу – методу найближчого сусіда. Він є досить ефективним на задачах малої розмірності та швидким, тому перебудова маршруту буде доволі точною та проведена за кілька секунд. Проте якщо кількість клієнтів у маршруті перевищує 10 клієнтів, то евристичного методу буде недостатньо. Тому спочатку буде проведений евристичний пошук за допомогою методу найближчого сусіда а потім це рішення буде вдосконалене на невеликій кількості ітерацій алгоритму імітації відпалу. Сам алгоритм набуде певних невеликих змін у порівнянні з гібридним і буде містити такі ключові операції:

- формування нових умов маршруту з виключенням можливості обрання забракованого маршруту;
- формування початкового рішення за допомогою методу найближчого сусіда та ініціалізація параметрів(початкова температура встановлюється досить низькою, для пришвидшення збіжності алгоритму);
- ітераційний процес алгоритму;

- зупинка алгоритму і надання нового рішення.

Поєднання методу найближчого сусіда та пришвидшеного алгоритму імітації відпалу дозволить отримувати нові маршрути для конкретного транспортного засобу із малими втратами по точності та незначними витратами часу. Блок-схема цього алгоритму зображена у додатку В.

Отже, у програмному продукті передбачено два можливих варіанти перебудови маршрутів. При перебудові всіх маршрутів використовуються альтернативні рішення з гібридного алгоритму, які були в одній популяції з основним розв'язком. При необхідності перебудови маршруту для певного клієнта використовується евристичний метод найближчого сусіда, який формує новий маршрут, згідно з новими умовами, а при великій кількості клієнтів, отриманий маршрут вдосконалюється за допомогою пришвидшеного алгоритму імітації відпалу.

### 3.4 Переваги та недоліки розробленого гібридного алгоритму

Розглянути у попередньому розділі гібридний алгоритм має багато переваг, які роблять його гарним вибором для вирішення задачі CVRP. Далі вкажемо переваги розробленого алгоритму

Першою перевагою є баланс між локальним і глобальним пошуком. Гібридний алгоритм використовує два різних підходи, які дозволяють виконувати як глобальний так і локальний пошук. Генетичний алгоритм добре виконує глобальний пошук у просторі рішень, а покращений алгоритм імітації відпалу доповнює його роботу виконуючи локальний пошук.

Крім того, у порівнянні із звичайним GA-SA алгоритмом покращений алгоритм імітації відпалу працює краще ніж звичайний алгоритм імітації відпалу завдяки використанню додаткового оператора схрещування. Крім того, завдяки механізму прийняття гірших рішень з певною ймовірністю, алгоритм імітації відпалу допомагає уникнути локальних мінімумів, до яких може привести генетичний алгоритм.

Ще однією перевагою є безумовно надання альтернативних рішень. Генетичний алгоритм працює з популяцією рішень, що дозволяє зберігати кілька альтернативних маршрутів з популяції. Наявність кількох рішень дає можливість швидко адаптуватися до змін, використовуючи альтернативні маршрути без необхідності запуску алгоритму заново і даючи користувачу кілька варіантів побудови маршрутів.

Крім того, даний алгоритм показує гарну ефективність на задачах великої розмірності. Через велику кількість операцій гібридний алгоритм є не дуже доречним для задач малої розмірності проте він краще справляється з задачами, що мають складні обмеження, та велику кількість транспортних засобів.

Ще однією перевагою є можливість перебудови маршрутів. Завдяки використанню алгоритму імітації відпалу, який досить швидкий, існує можливість перебудови маршрутів для конкретного транспортного засобу, не втрачаючи у точності.

Розроблений алгоритм є перспективним та ефективним для задачі CVRP проте має свої недоліки:

- складність реалізації;
- високі обчислювальні витрати(поєднання двох алгоритмів збільшує час роботи програми, порівняно з простими евристичними алгоритмами та даний алгоритм є затратним по пам'яті);
- складність налаштування та залежність від параметрів.

### Висновки до розділу 3

У цьому розділі було сформовано математичну постановку задачі, яка розв'язується. Також було надано детальний опис розробленого гібридного генетичного алгоритму з покращеним алгоритмом імітації відпалу з усіма етапами алгоритму. Крім того, було вказано як даний алгоритм буде змінюватись та враховувати перебудову маршрутів та генерацію альтернативних шляхів. Було вказано всі переваги та потенційні недоліки розробленого алгоритму.

## 4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ПРОДУКТУ ТА ТЕСТУВАННЯ

### 4.1 Архітектура системи

Розроблена система управління логістикою транспортної компанії має клієнт-серверну архітектуру.

Клієнтська частина реалізована за допомогою React і складається з кількох компонентів, таких як карта, форма для даних, динамічний список клієнтів, пошуку локацій через сервіси гугл-карт та компонент результату, куди виводяться дані про маршрути. Повну структуру цих всіх компонентів, та їх взаємодію між собою та з серверною частиною можна побачити у додатку Г. Структура фронтенд-частини зображена на рисунку 4.1.

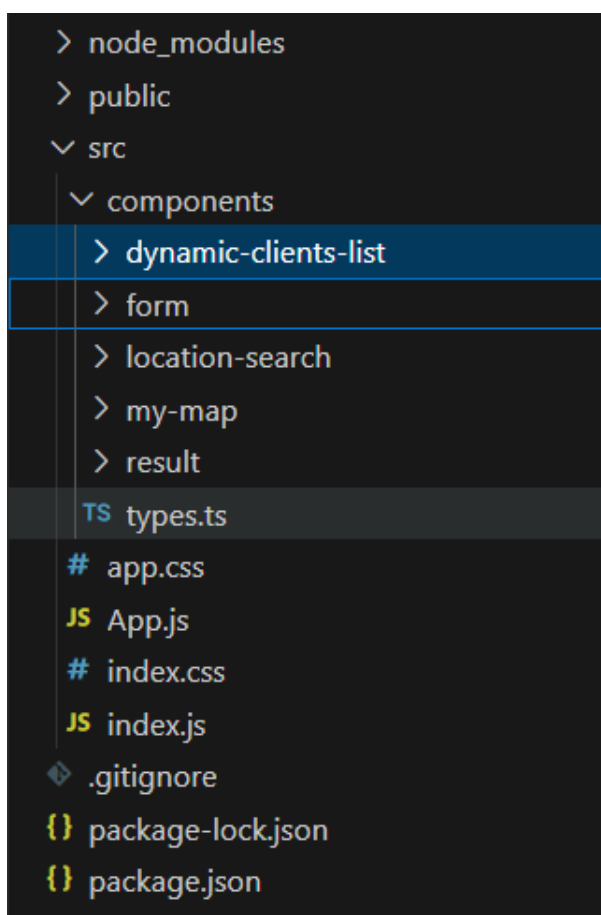


Рисунок 4.1 – Структура фронтенд частини

Серверна частина розроблена на платформі .NET за допомогою мови програмування C#. Вона включає в себе різні класи для генерації маршрутів за

допомогою алгоритмів, API-контролери та різні допоміжні класи для вирахування довжин маршрутів, зчитування даних з файлу. Повністю структура зображена на рисунку 4.2.

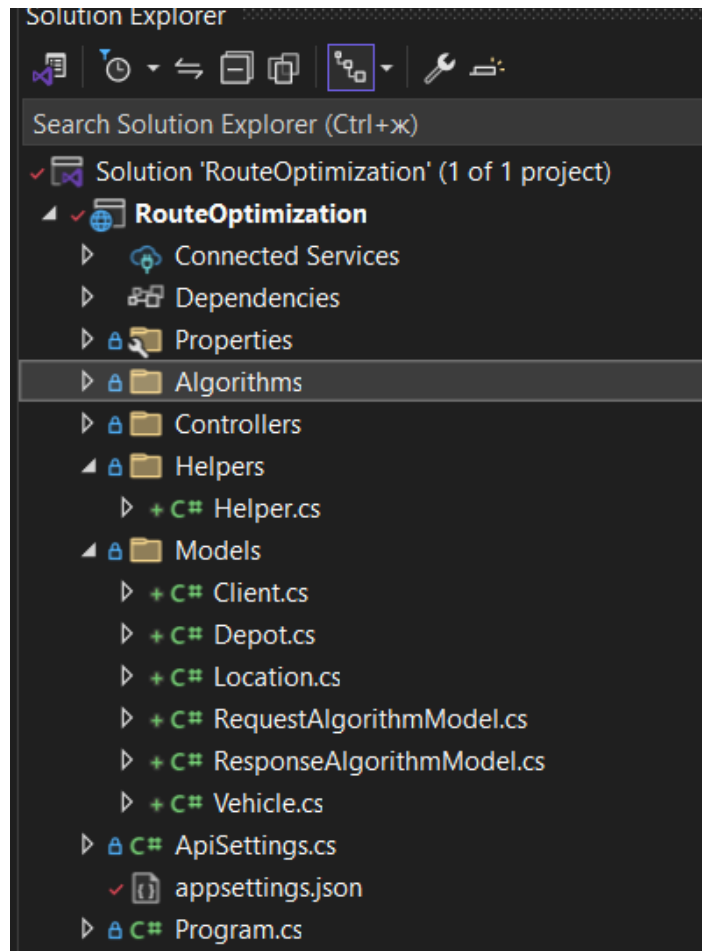


Рисунок 4.2 – Структура серверної частини

У додатку Д зображена ER-діаграма системи, яка показує всі сутності, які є у проекті та як вони взаємодіють між собою. Основними сутностями є RequestModel та ResponseModel, які є основними сутностями для обміну даними та є також сутності клієнта, депо та автомобіля, класи яких будуть описані пізніше.

Взаємодія клієнтської частини з серверною відбувається за допомогою HTTP-запитів.

Система має також інтеграцію з сервісами гугл карт та використовує Google Maps API для отримання даних локацій та дозволяє використовувати карти для побудови маршрутів.

## 4.2 Вибір технологій та інструментів

Для побудованої системи необхідно було обрати технології та інструменти, які дозволять зробити систему швидкою масштабованою та з можливістю використання незалежних компонентів у фронтенд частині для генерації різних результатів без перезавантаження сторінки та переходу на нові сторінки.

При розробці бекенд частини використовувалась мова програмування C#, так як вона використовує вбудовані механізми безпеки та управління пам'яттю та є однією з найбільш безпечних швидких та широко розповсюджених мов програмування та має необхідні вбудовані структури для реалізації алгоритмів, таких як хеш-таблиці, списки, словники та інші.

Платформа ASP.NET Web Api використовувалась для створення серверної частини, так як вона дозволяє легко створювати веб-сервіси, що відповідають REST принципам на C# та підтримує автоматичну серіалізацію та десеріалізацію даних з JSON[38].

Для розробки клієнтської частини було використано бібліотеку JavaScript React js[39], яка підтримує розробку незалежних компонентів сторінки та їх незалежне оновлення один від одного, що дозволяє змінювати тільки ту частину сторінки, яка потрібно так як React має механізм відстежування стану компоненту і оновлює компонент тільки при оновленні стану вибраного компоненту. Крім того дана бібліотека має зрозумілу структуру написання коду, що робить зручним її використання.

Для відображення карт та автоматичного пошуку місць по вводу використовуються сторонні сервіси Google Maps API. Вони мають широкий функціонал, зокрема підтримують карти, маркери, полігони, автодоповнення місць. Крім того дані сервіси надають високу точність та постійно оновлюються та мають детальну інструкцію інтеграції у додаток[40].

Для виконання запитів від клієнта до сервера використовується клієнт Http – Axios. Він є зручним у використанні та полегшує асинхронне програмування та обробку запитів.

### 4.3 Опис основних класів програми

Розглянемо основні класи програмного продукту із методами, які вони мають та полями, для того щоб краще зрозуміти які функції відповідають за який розроблений функціонал. Повністю діаграма класів зображена у додатку Е.

Спочатку розглянемо клас покращеного Алгоритму імітації відпалу або SAAlgorithm class, який є відповідником алгоритму та використовується пізніше у гібридному генетичному алгоритмі. Представимо цей клас у вигляді таблиці(таблиця 4.1).

Таблиця 4.1 – Опис класу SAAlgorithm

Тип	Назва	Зміст
Поле	random	Рандомайзер випадкових чисел
Поле	iterations	Кількість ітерацій
Поле	Temperature	Параметр алгоритму - температура
Поле	Alpha	Параметр алгоритму – коефіцієнт охолодження
Метод	OptimizeRoutes ()	Основний метод, який виконує маршрутизацію
Метод	CalculateTotal Distance()	Метод обрахування відстані для всіх автомобілів
Метод	GenerateNeighbour	Метод генерації наступного розв'язку задачі
Метод	PerformCrossover()	Метод, який підготовлює дані для кросинговеру та запускає його
Метод	Crossover()	Метод кросинговеру
Метод	GetAllClientsFrom Parents()	Метод отримання віх клієнтів з батьківських елементів
Метод	DeepCopySolution()	Метод глибокого копіювання розв'язків

Як бачимо цей клас надає весь необхідний спектр методів для алгоритму імітації відпалу.

Далі опишемо основний клас програми GeneticAlgorithm який відповідає за гібридний генетичний алгоритм(таблиця 4.2).

Таблиця 4.2 – Опис класу GeneticAlgorithm

Тип	Назва	Зміст
Поле	Random	Рандомайзер випадкових чисел
Поле	showAlternativeRoutes	Булева змінна, яка вказує чи генерувати кілька оптимальних маршрутів
Поле	Generations	Максимальна кількість генерацій
Поле	mutationProbability	Параметр алгоритму – коефіцієнт мутації
Поле	populationSize	Параметр алгоритму – розмір популяції
Метод	OptimizeRoutes ()	Основний метод, який підготовляє дані та запускає маршрутизацію
Метод	ApplyAlgorithm()	Метод, який виконує маршрутизацію
Метод	GenerateInitial Population()	Метод генерації початкової популяції
Метод	CalculateTotal Distance()	Метод, який обраховує загальну відстань всіх маршрутів
Метод	SelectBestIndividuals()	Метод, який вибирає кращі особини з популяції
Метод	Crossover()	Метод кросинговеру
Метод	Reproduce()	Метод отримання віх клієнтів та підготовки кросинговеру
Метод	Mutate()	Метод виконання мутації

Також необхідно описати класи моделей такі, як клас Client, Vehicle та допоміжний клас Helper, який використовується у всіх алгоритмах та класах моделей та надає ключові можливості для роботи. Класи, які менш важливі, або ті які не мають широкого функціоналу представляти не будемо. Сюди входять так класи як NearestNeighbourAlgorithm, Depot, Request- і ResponseModel.

Спочатку розглянемо клас, який відповідає за клієнта та реалізує всі його основні властивості – клас Client.(Табл. 4.3)

Таблиця 4.3 – Опис класу Client

Тип	Назва	Зміст
Поле	Id	Поле ідентифікатор
Поле	Quantity	Розмір потреби клієнта
Поле	IsVisible	Булева змінна, яка показує чи вже використаний цей клієнт
Поле	Latitude	Параметр широти
Поле	Longitude	Параметр довготи
Метод	DeepCopy ()	Метод для глибокого копіювання всіх даних клієнта
Метод	SetVisible()	Метод, який робить клієнта видимим

Далі розберемо будову класа, який відповідає за транспортний засіб – Vehicle(таблиця 4.4)

Таблиця 4.4 – Опис класу Vehicle

Тип	Назва	Зміст
Поле	Id	Ідентифікатор машини
Поле	Capacity	Місткість транспорту
Поле	Clients	Список клієнтів, яких обслуговує даний транспортний засіб
Поле	TotalLoad	Загальна завантаженість ТЗ

Тип	Назва	Зміст
Поле	TotalDistance	Загальна пройдена відстань
Метод	DeepCopy ()	Метод для глибокого копіювання
Метод	ClearClients()	Метод, який видаляє всіх клієнтів
Метод	CanAccomodate ()	Метод перевірки чи вміститься у транспорт вказаний вантаж
Метод	AddClient()	Метод, який додає клієнта
Метод	RemoveClient()	Метод, який видаляє клієнта
Метод	CalculateRouteMetric()	Метод який обраховує загальну довжину маршруту транспортного засобу

Також опишемо важливий клас Helper, який містить допоміжні методи.(табл.4.5)

Таблиця 4.5 – Опис класу Helper

Тип	Назва	Зміст
Метод	GetDataFromFile ()	Метод для парсингу даних з файлу отриманого користувачем
Метод	GetDistance()	Метод, який обраховує відстань між двома точками за заданими довготою і широтою використовуючи математичні перетворення

Опишемо також клас NearestNeighbourAlgorithm, який використовується для перебудови конкретного маршруту(табл.4.6).

Таблиця 4.6 – Опис класу NearestNeighbourAlgorithm

Тип	Назва	Зміст
Метод	OptimizeRoutes ()	Основний метод, який виконує налаштування параметрів та запускає маршрутизацію
Метод	ApplyAlgorithm()	Метод знайдення оптимального розв'язку
Метод	CalculateAlternative Route()	Метод для пошуку альтернативних розв'язків наближених до основного розв'язку

Отже, ми описали основні класи програмного продукту, повна діаграма класів знаходиться у додатку Е.

Тепер розглянемо функціонал, який надає описаний програмний продукт та інструкцію користувача. Посилання на повний код програмного продукту знаходиться у додатку А.

#### 4.4 Опис функціоналу та інструкція користувача

Розпочнемо опис з функціоналу, який надає серверна частина, тобто API нашої програми. Серверна частина має широкий функціонал, повна діаграма класів якої зображена у додатку Е. Серверна частина отримує запити клієнтської частини та може надавати відповіді по таким запитам як отримання оптимальних маршрутів через файлові дані(`api/v1/optimize/from-file`), отримання оптимальних маршрутів через дані заповнені формою(`api/v1/optimize/with-algorithm`), регенерація маршруту для конкретного транспортного засобу(`api/v1/optimize/specific-route`). На рисунку 4.3 показано вказані можливі запити до серверної частини за допомогою допоміжної бібліотеки Swagger.

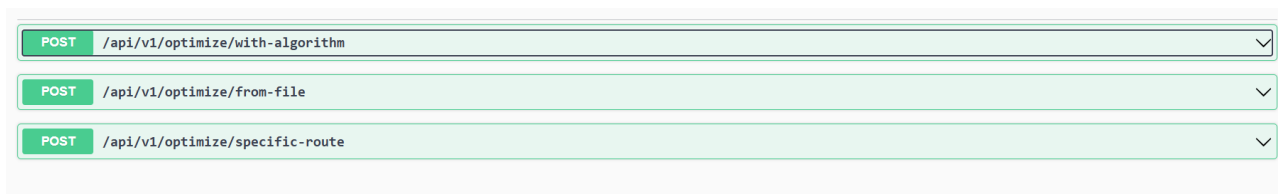


Рисунок 4.3 – Можливі запити до API

Для побудови маршрутів як з даних по файлу так і з даних отриманих за допомогою форми використовується гібридний алгоритм із генетичного алгоритму і покращеного алгоритму імітації відпалу, схема роботи якого була описана у розділі 3.

Для регенерації окремого маршруту використовується алгоритм найближчого сусіда та алгоритм симуляції відпалу. Детальна блок-схема цього алгоритму зображена у додатку В.

У серверній частині є класи які відповідають моделям депо, клієнта автомобіля, request та response моделі. Також є класи, які відповідають вище згаданим алгоритмам. Кожен клас алгоритму походить від інтерфейсу IAlgorithm та має два основних методи OptimizeRoutes та ApplyAlgorithm. Також у кожного алгоритму є свої методи які необхідні відповідно до вимог алгоритму.

Отже, ми розглянули функціонал, який надає серверна частина, тепер розглянемо інструкцію користувача для додатку та опишемо функціонал на стороні клієнта.

Головною метою системи є побудова та оптимізація маршрутів з можливістю регенерації повністю всіх маршрутів та окремо вибраного маршруту для певного транспортного засобу.

Клієнтський додаток являє собою веб-додаток, головний екран, якого зображено на рисунку 4.4.

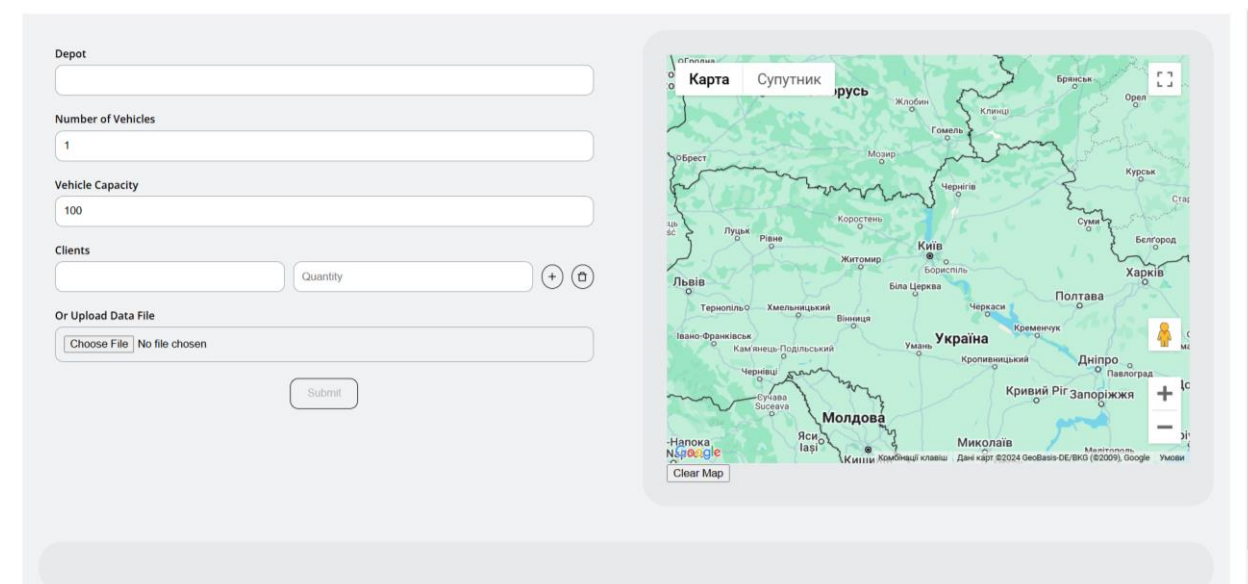


Рисунок 4.4 – Головний екран веб-додатку

З рисунку 4.4 можна бачити, що основне вікно системи розділено на три основні частини: форма для заповнення даних користувачем, карта та результат (поки немає побудованих маршрутів, ця частина є пустою).

Розглянемо основні дії клієнта у програмі. Це відправка форми з клієнтами для оптимізації маршрутів, відправка готового файлу з клієнтами, введення нових клієнтів, генерація альтернативних маршрутів, вибір маршруту для перебудови та оцінювання отриманих результатів на екрані у текстовому та графічному вигляді на гугл-карті. Всі можливі дії клієнта зображені на діаграмі прецедентів у додатку Ж, розглянемо їх по чергово.

При заході у додаток, клієнт може вибрати два шляхи: вибір файлу з раніше заготовленими клієнтами або заповнення форми з вводом адрес клієнтів та всіх інших потрібних даних вручну. Для цього використовується частина додатку, яка на рисунку 4.4 зображена зліва. Розберемо її далі детальніше з описом всіх можливих сценаріїв користувача.

Розглянемо першу частину програми, а саме форму для введення даних користувачем про маршрути, кількість автомобілів, їх місткість та точку відправки (Рис. 4.5).

The form contains the following fields and controls:

- Depot:** A large empty text input field.
- Number of Vehicles:** A text input field containing the number '1'.
- Vehicle Capacity:** A text input field containing the number '100'.
- Clients:** A section with two text input fields. The first is empty, and the second is labeled 'Quantity'. To the right of the 'Quantity' field are two circular icons: a plus sign (+) and a trash can icon.
- Or Upload Data File:** A section with a file selection button labeled 'Вибрати файл' and a status indicator 'Файл не вибрано'.
- Submit:** A rounded rectangular button at the bottom center.

Рисунок 4.5 – Форма для введення користувачем

З рисунку 4.5 можемо бачити, що перший рядок форми це поле для вводу адреси депо – точки звідки відправляються і куди повертаються всі транспортні засоби. Дане поле має функцію автоматичного доповнення текстів з реальними адресами(рис.4.6).

The image shows the 'Depot' field with the following details:

- Text in the field:** 'Вулиця академіка янгеля 58, Вінниця, Вінницька область, Україн'.
- Dropdown suggestions:**
  - Вулиця академіка янгеля 58 Вінниця, Вінницька область, Україна
  - провулок Академіка Янгеля, 58 Вінниця, Вінницька область, Україна
- Footer:** 'powered by Google' logo.

Рисунок 4.6 – Функція автодоповнення адрес

Після поля депо йде поле для вводу місткості для автомобілів, яке обмежене тільки цифровими та додатніми значеннями.

Наступним йде динамічний список клієнтів де перше поле для вводу відповідає адресі клієнта, а друге кількості вантажу, яку йому необхідно привезти.

Також у рядку списку клієнтів є дві кнопки, додати нового клієнта та видалити клієнта. Кнопка додати клієнта дозволяє додавати багато клієнтів, що створює динамічний список клієнтів(рис.4.5)

Clients		
вулиця Соборна, 15, Вінниця, Вінни.	50	+ ☒
Житомир, Житомирська область, Ук	Quantity	+ ☒
	Quantity	+ ☒
	Quantity	+ ☒
	Quantity	+ ☒
	Quantity	+ ☒

Рисунок 4.7 – Динамічний список клієнтів

Також бачимо з рисунку 4.5, що кнопка форми є недоступною поки не будуть заповнені усі необхідні дані.

Всі поля форми мають валідацію та перевіряються. Так наприклад, поля адрес повинні бути всі заповнені, всі поля Quantity повинні бути заповнені та можуть бути лише додатніми числами.

Відправлення даних на сервер відбувається за допомогою кнопки Submit, яка стає активною тільки тоді коли всі необхідні поля заповнені, або вибраний альтернативний шлях завантаження даних, про який розповімо нижче.

Крім заповнення даних форми вручну є також опція внизу у формі завантажити файл з даними, який був заготовлений раніше працівником логістичної компанії, для того щоб не вводити всі адреси вручну, та використовувати дані файлу для моделювання майбутніх маршрутів. Для обробки даних з файлу використовується спеціальний метод. Для вибору файлу потрібно натиснути кнопку Choose File та відкриється вікно для вибору файлу з комп'ютера користувача. Після вибору файлу, який підходить по формату, всі поля форми стають не активними та не дозволяють нічого ввести поки не буде відправлений файл на обробку(рис.4.8).

Number of Vehicles

1

Vehicle Capacity

100

Clients

Quantity

+

-

Or Upload Data File

Choose File Test.txt

Submit

Рисунок 4.8 – Блокування полів для вводу при виборі файлу

Структура файлу, який підходить для обробки алгоритмом зображена на рисунку 4.9.

File	Edit	View
6	100	
0	50,4501	30,5234
1	49,1234	24,5678 10
2	48,8765	30,1234 12
3	50,2345	32,8765 8
4	46,3456	35,2345 15
5	49,6789	28,4567 7
6	47,8901	33,5678 9
7	50,7890	29,6789 11
8	48,0123	31,8901 13
9	49,3456	27,0123 6
10	47,4567	34,1234 14
11	50,5678	30,2345 5
12	48,6789	29,3456 10
13	49,7890	26,4567 12
14	47,0123	32,5678 9
15	50,8901	28,6789 8
16	48,2345	31,7890 7
17	49,9012	25,8901 13
18	47,3456	34,9012 11
19	50,0123	29,0123 6

Рисунок 4.9 – Структура файлу з даними

У першому рядку файлу вказується загальна інформація – кількість автомобілів та обмеження по вантажопідйомності кожного автомобіля. У другому рядку вказуються інформація про депо – Id і значення довготи і широти у градусах. Починаючи з третього рядка йде інформація про клієнтів. Інформація про клієнтів

надається у також вигляді як і депо, тільки додається ще одне число це потреба кожного конкретного користувача. Якщо записи у файлі не відповідатимуть структурі, то при обробці даних програмою буде отримана помилка і це буде виведено на екран. Після вибору файлу і натискання кнопки Submit йде також запуск роботи алгоритмів.

Наступною важливою частиною додатку є карта, яка розташована справа на головному екрані. На ній відображаються маршрути вираховані з даних відправлених формою. На рисунку 4.10 показано вигляд карти до відправлення форми.

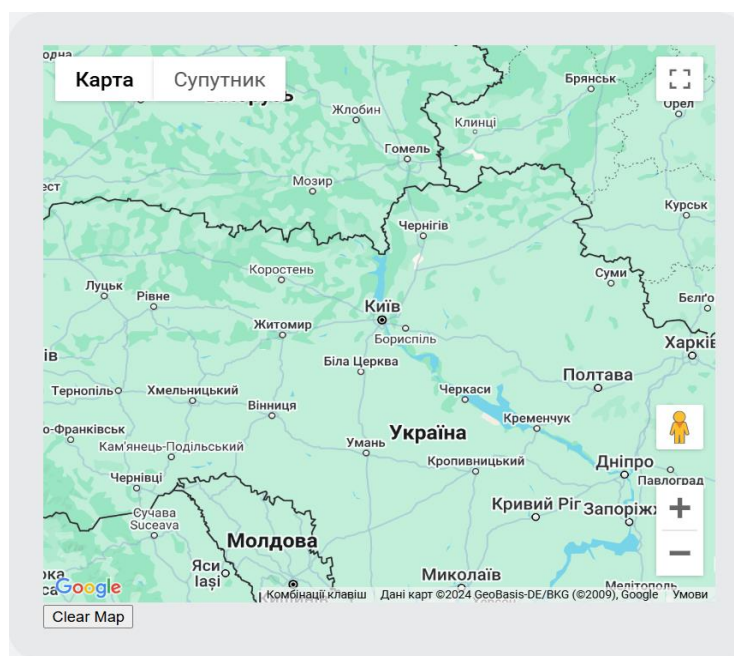


Рисунок 4.10 – Карта без маршрутів

Карта створена за допомогою використання сервісів Google Maps API. Без вказання точки депо карта по налаштуванням показується з центром на Києві.

Як бачимо на рисунку 4.10 можна збільшити або зменшити масштаб карти, відкрити карту на повний екран. Також є можливість перейти у режим вулиць або режим супутника.

Після відправлення форми з даними або відправлення файлу і відправки запиту на серверну частину приходять результати з маршрутами. На рисунку 4.11 показано як виглядає карта із побудованими оптимізованими маршрутами, на

тестовому прикладі де є 6 транспортних засобів та 50 клієнтів, яких потрібно обслужити. Місткість кожного транспортного засобу становить 100 одиниць.

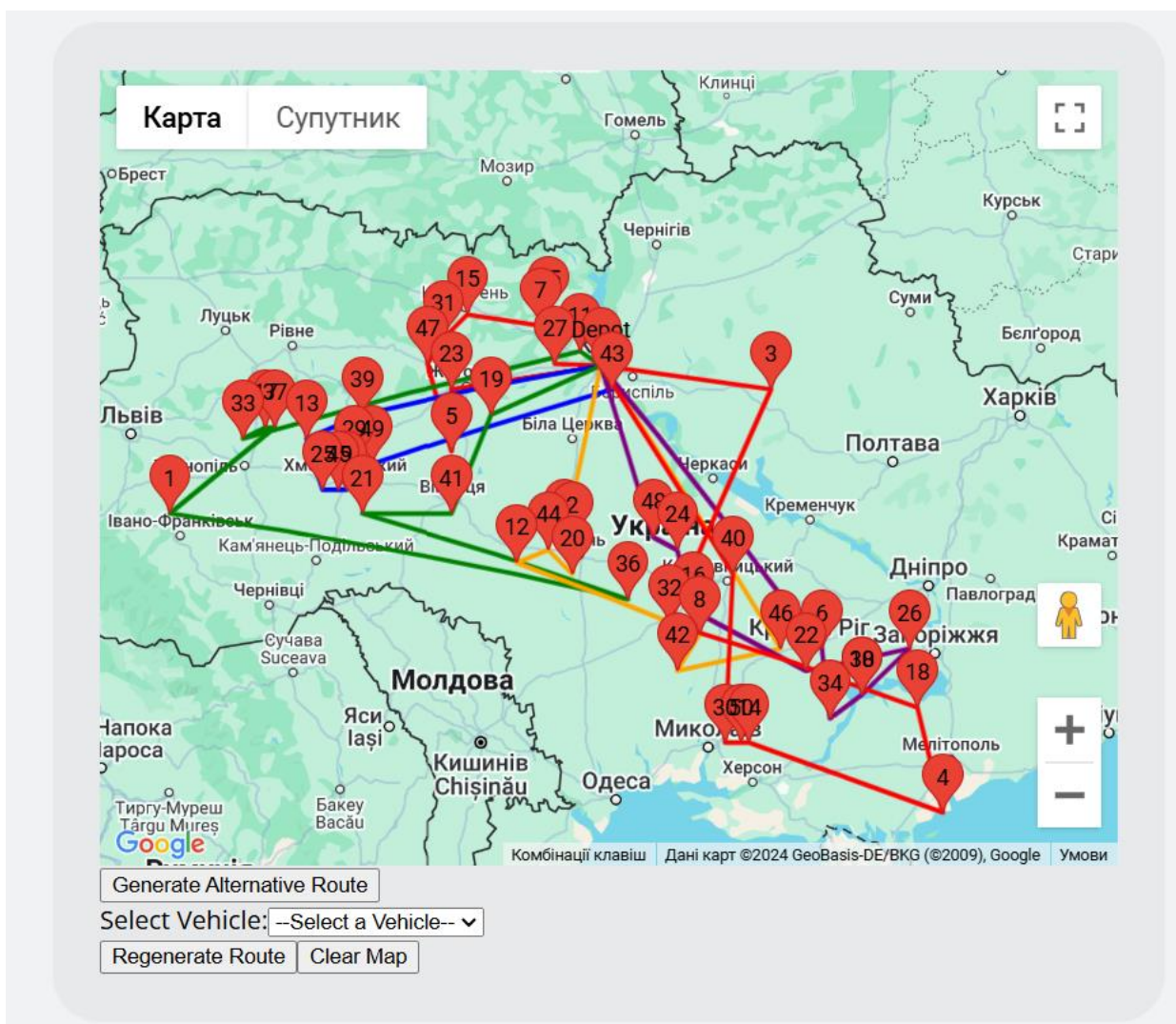


Рисунок 4.11 – Карта із оптимізованими маршрутами для 50 клієнтів

На рисунку 4.11 також видно що кожен маркер позначений числом, яке відповідає Id клієнта та окремо позначено депо. Кожен маршрут позначений іншим кольором, що дозволяє легко їх розрізняти.

Крім того, нижче карти є кілька кнопок, які дають додатковий функціонал. Кнопка *Generate Alternative Route* надає альтернативні маршрути для всіх транспортних засобів, які не сильно відрізняються від початкового у плані витрат. Результат перебудови всіх маршрутів можна побачити на рисунку 4.12.

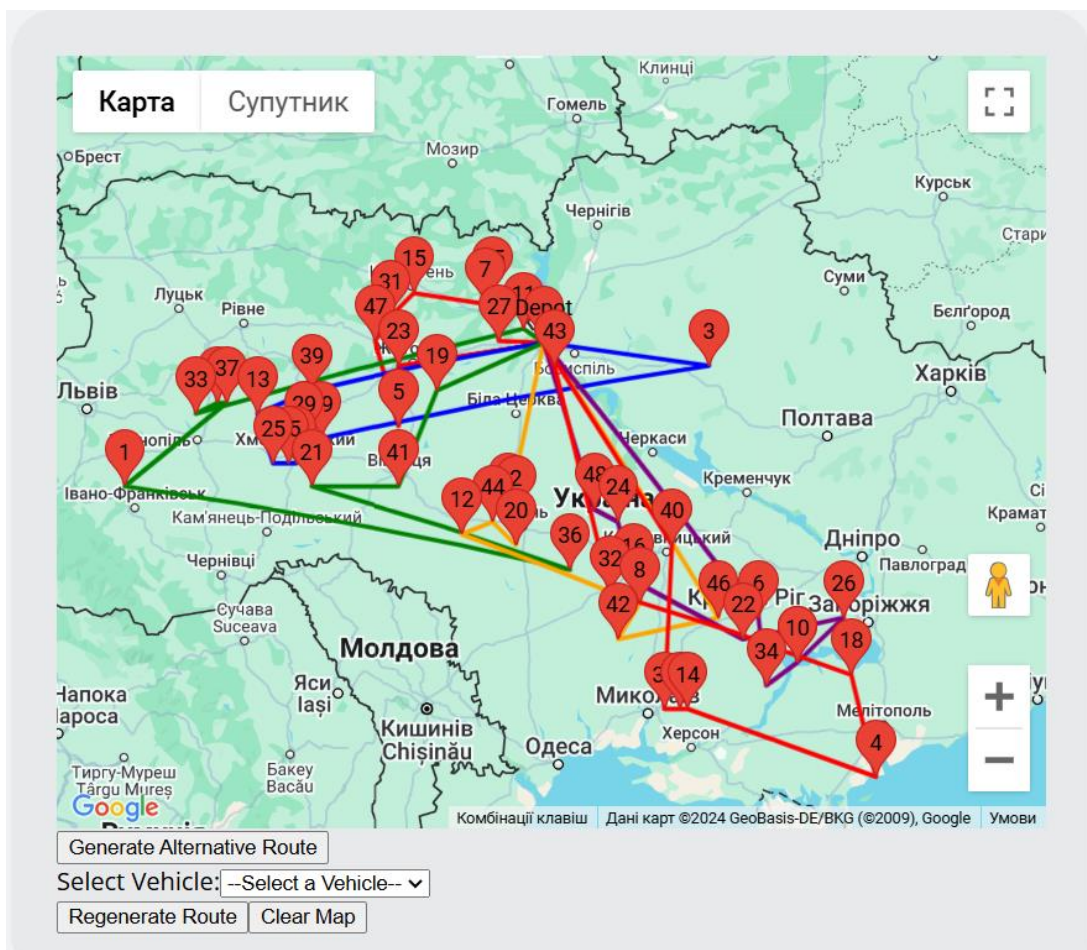


Рисунок 4.12 – Альтернативні маршрути

Внизу головного екрана є третій основний блок програми, у якому відображається вся інформація про маршрути. Для цього прикладу, що був показаний вище результат показаний на рисунку 4.13.

Optimized Routes	
Total Distance: 4286.42 km	
<b>Vehicle 1</b>	<b>Vehicle 2</b>
Total Load: 74	Total Load: 100
Total Distance: 430.87 km	Total Distance: 512.82 km
<b>Route steps:</b>	<b>Route steps:</b>
1. Depot Start	1. Depot Start
2. Client: 27 - Quantity: 13	2. Client: 3 - Quantity: 8
3. Client: 35 - Quantity: 8	3. Client: 49 - Quantity: 6
4. Client: 7 - Quantity: 11	4. Client: 29 - Quantity: 36
5. Client: 15 - Quantity: 8	5. Client: 9 - Quantity: 6
6. Client: 31 - Quantity: 9	6. Client: 45 - Quantity: 8
7. Client: 47 - Quantity: 13	7. Client: 25 - Quantity: 18
8. Client: 5 - Quantity: 7	8. Client: 13 - Quantity: 12
9. Client: 23 - Quantity: 5	9. Client: 39 - Quantity: 6
10. Depot End	10. Depot End
<b>Vehicle 3</b>	<b>Vehicle 4</b>
Total Load: 97	Total Load: 99
Total Distance: 732.05 km	Total Distance: 760.29 km

Рисунок 4.13 – Результати для 50 клієнтів

Якщо були введені адреси через форму напряму то у результатах показується адреса клієнта повністю(рисунок 4.14).

<b>Total Distance:</b> 1647.97 km	
<b>Vehicle 1</b>	<b>Vehicle 2</b>
<b>Total Load:</b> 100	<b>Total Load:</b> 100
<b>Total Distance:</b> 370.29 km	<b>Total Distance:</b> 462.93 km
<b>Route steps:</b>	<b>Route steps:</b>
1. Depot Вінниця, Вінницька область, Україна, 21000	1. Depot Вінниця, Вінницька область, Україна, 21000
2. Client: вулиця Київська, 10, Житомир, Житомирська область, Україна, 10001 - Quantity: 50	2. Client: вулиця Окружна, 5, Кривий Ріг, Дніпропетровська область, Україна, 50000 - Quantity: 50
3. Client: вулиця Хмельницького, 10, Винники, Львівська область, Україна, 79495 - Quantity: 50	3. Client: вулиця Небесної Сотні, 5, Умань, Черкаська область, Україна, 20300 - Quantity: 50
4. Depot Вінниця, Вінницька область, Україна, 21000	4. Depot Вінниця, Вінницька область, Україна, 21000
<b>Vehicle 3</b>	
<b>Total Load:</b> 100	
<b>Total Distance:</b> 814.75 km	
<b>Route steps:</b>	
1. Depot Вінниця, Вінницька область, Україна, 21000	
2. Client: вулиця Соборна, 12, Хмельницький, Хмельницька область, Україна, 29000 - Quantity: 50	
3. Client: вулиця Борщагівська, 53, Київ, Україна, 02000 - Quantity: 50	

Рисунок 4.14 – Блок результатів маршрутів із повними адресами

Основною функцією користувача також є можливість створення регенерації маршруту для конкретного вибраного маршруту. Необхідні кнопки для цього знаходять під картою після побудови основних маршрутів. Вибравши транспортний засіб, маршрут якого потрібно перебудувати у відповідному полі та натиснувши кнопку Regenerate route можна перебудувати маршрут для вибраного транспортного засобу із незначними втратами. На цій кнопці також є валідація, і при не вводі транспортного засобу, маршрут, кого хочемо змінити воно видає попередження.

Перебудуємо маршрут для транспортного засобу номер 3 на рисунку 4.12, якому відповідає зелений колір.

Новий отриманий маршрут можна побачити зображенням на рисунку 4.15(зелений колір).

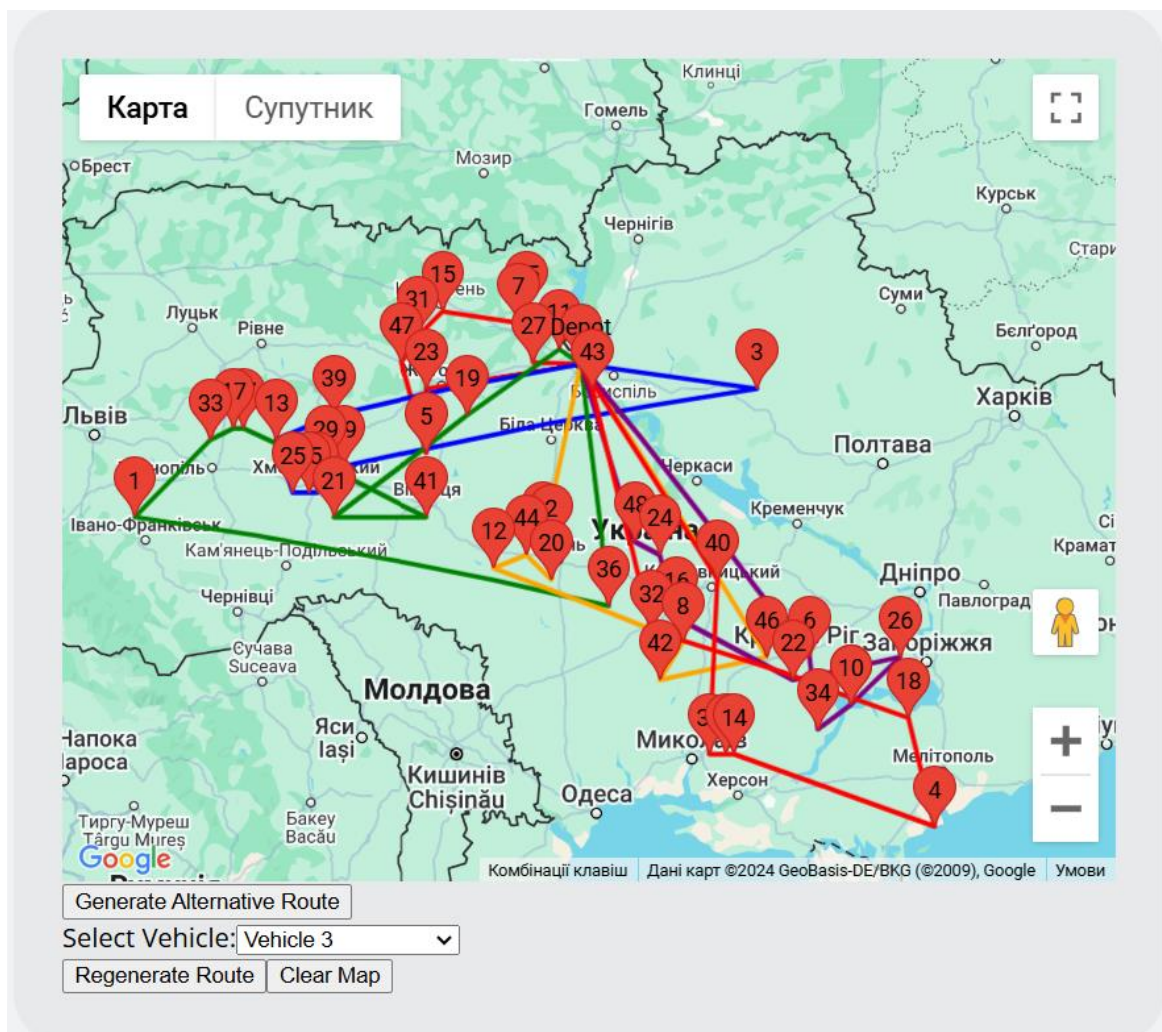


Рисунок 4.15 – Результати з перебудованим маршрутом для транспортного засобу номер 3

З рисунка 4.15 видно, що маршрут транспортного засобу під номером 3 набув певних змін проте його сумарна довжина не сильно змінилась. Також відповідно до зміни маршруту на карті, змінюються також результати у блоці результатів нижче.

Також є кнопка Clear Map яка очищує карту від усіх маршрутів.

Це весь функціонал, який надає клієнтська частина. Розглянемо весь цикл роботи програми. Його можна побачити на діаграмі станів програми у додатку И та діаграмі послідовностей у додатку К.

Згідно діаграми послідовностей спочатку користувач вводить дані, потім веб-додаток перевіряє ці дані на валідність на стороні клієнта та відправляє запит на сервер для отримання маршрутів.

Сервер в свою чергу також проводить обробку даних, потім будує маршрути за допомогою алгоритмів та відправляє дані назад на клієнтську частину для побудови маршрутів та показу результатів.

Після отримання даних від сервера, клієнтська частина спочатку виконує побудову маршрутів на карті, потім відображає дані у вигляді маршрутів з покроковим відвідуванням клієнтів.

Після виводу результатів клієнт може задовольнитись отриманими результатами а може захотіти їх змінити. Для цього він обирає перебудову конкретного маршруту. Веб-додаток обробляє ці дані та відправляє їх на сервер. Сервер виконує перебудову вибраного маршруту та відправляє дані клієнту, після чого відбувається перебудова маршрутів на карті та зміна результатів у блоці результатів. На цього життєвий цикл завершується та можна приступати до нової задачі або повторювати вище описані дії.

Розглянемо також варіанти коли система отримує помилку. Ці всі стани показані у діаграмі станів у додатку І.

Програма виводить дані про помилку після неправильних даних у формі або у файлі, після чого можна доповнити форму або змінити файл і відправити заново.

Програма може припинити свою роботу у разі успішного завершення алгоритмів, виводу результатів та виходу з програми, або при негативних сценаріях таких як критична помилка при обробці даних сервером, помилка в роботі алгоритму, або помилка з сервісом карт і неможливістю побудувати оптимізовані маршрути. Проте всі ці сценарії малоімовірні так як система добре протестована, що ми розглянемо у наступному розділі дисертації.

#### 4.5 Тестування системи

Було проведено тестування всіх компонентів системи, зокрема було перевірено валідацію записів у формі і вона не дозволяє залишати пусті поля при відправці форми(рис.4.16).

Рисунок 4.16 – Тестування валідності форми

Було перевірено роботу функції регенерації маршруту і при спробі це зробити без вибраного транспортного засобу система попереджає.(рис.4.17)

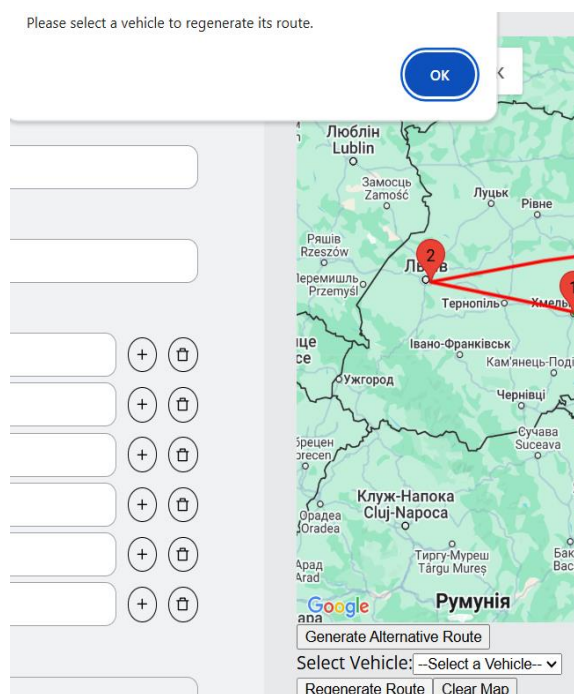


Рисунок 4.17 – Перевірка коректності роботи функції регенерації маршруту.

Окрім цього було перевірено всі інші функції роботи системи і вони працювали без збоїв з виводом помилок, там де це необхідно та обумовлено системою.

Також було перевірено ефективність роботи нового гібридного алгоритму у порівнянні з іншими алгоритмами. Зокрема його було порівняно на 50-ти клієнтах із простим генетичним алгоритмом та алгоритмом імітації відпалу. Результати цього порівняння можна побачити на рисунку 4.18.

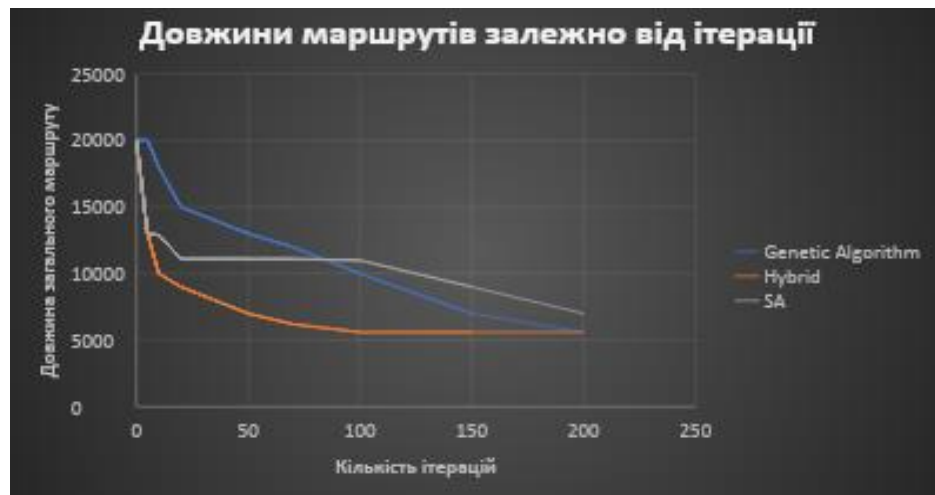


Рисунок 4.18 – Порівняння роботи гібридного алгоритму у порівнянні з іншими алгоритмами

Як бачимо з рисунку 4.18 гібридний алгоритм працює набагато швидше та вже на ітерації 50 дає значення, яке близьке до найкращого а на ітерації 70 знаходить оптимальне рішення, в той час як інші алгоритми потребують набагато більше ітерацій, що показує що розроблений алгоритм є ефективним.

#### Висновки до розділу 4

У цьому розділі було описано архітектуру розробленої системи та вибрані інструменти розробки

Крім того, було проведено опис всіх елементів системи та їх функцій, таких як введення даних з файлу і форми, генерація альтернативних маршрутів для всіх транспортних засобів та для окремо вибраного транспортного засобу, показ результатів та побудову маршрутів на карті. Також було проведено тестування системи та перевірка ефективності роботи гібридного алгоритму.

## 5 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

### 5.1 Опис ідеї проекту

Розглянемо основні ідеї стартап проекту у таблиці 5.1.

Таблиця 5.1 – Опис ідей стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Управління логістикою компанії з доставки	Управління логістикою для компаній доставок посилок	Оптимізація процесу побудови маршрутів,
	Управління логістикою для великих транспортних компаній	можливість вибрати серед кількох альтернативних
	Управління логістикою для компаній з доставки їжі	маршрутів, можливість перебудови
	Управління логістикою для поштових служб	конкретного маршруту, підвищення продуктивності
Планування персональних маршрутів	Особисте планування маршрутів	Оптимізація власних маршрутів та зменшення часу на поїздки, заощадження коштів
Формування туристичних маршрутів	Планування маршрутів для туристичних компаній, туризм	Планування оптимальних маршрутів для туристичних груп

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	Розробка індивідуальних туристичних маршрутів	Побудова персональних маршрутів для туристів, оптимізація їх часу та витрат
Маршрутизація для служб таксі	Оптимізація маршрутів для таксопарків	Зменшення часу очікування для клієнтів, оптимізація роботи, зменшення витрат на паливо
Управління логістикою в будівництві	Доставка матеріалів на будівельні майданчики	Оптимізація маршрутів для великогабаритного транспорту, зменшення простоїв

Даний стартап є корисним насамперед для компаній з доставки, таких як компаній з доставок посилок, для великих транспортних компаній, для компаній з доставки їжі та для поштових сервісів таких як Укрпошта чи Нова пошта. Цей стартап допоможе їх оптимізувати маршрути та підвищити ефективність роботи, надаючи також можливість зміни маршрутів.

Крім того система може використовуватись для формування персональних маршрутів, що допомагає зменшити цим самим витрати по грошах та часу поїздки.

Також система може використовуватися туристичними компаніями для планування оптимальних маршрутів екскурсій, враховуючи час, відстань та цікавість місць.

Даний стартап також може використовуватись сервісами таксі. Вони можуть покращити ефективність перевезень, оптимізуючи маршрути водіїв та зменшуючи час очікування для пасажирів.

Також будівельні компанії можуть оптимізувати доставку матеріалів та обладнання на різні майданчики, зменшуючи витрати та простої використовуючи дану систему

Після опису ідей проекту необхідно порівняти його з системами, які надають схожий функціонал. Тому далі будемо розглядати сильні, слабкі та нейтральні сторони проекту відносно таких систем як MyRouteOnline, Route4Me, та Paragon. Порівняльний аналіз із визначенням сильних сторін буде зазначено нижче у таблиці 5.2

Таблиця 5.2 – Визначення сильних слабких та нейтральних сторін проекту

№ п/п	Техніко-економічні характеристики ідеї	Системи планування маршрутів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	MyRoute Online	Route4Me	Paragon			
1	Безкоштовний сервіс	Так	Пробний період	Пробний період	Пробний період	Ні	Ні	Так
2	Генерація альтернатив до основного маршруту	Так	Ні	Після додаткового запиту	Після додаткового запиту	Ні	Ні	Так
3	Можливість перебудови конкретного маршруту	Так	Ні	Так	Так	Ні	Так	Так
4	Простий інтерфейс користувача	Так	Так	Так	Ні	Ні	Так	Так
5	Підтримка побудови маршрутів для багатьох автомобілів	Так	Ні	Так	Так	Ні	Так	Ні
6	Можливість інтеграції з іншими системами	В майбутньому планується	Так	Так	Так	Ні	Так	Ні

№ п/п	Техніко-економічні характеристики ідеї	Системи планування маршрутів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	MyRoute Online	Route4Me	Paragon			
7	Використання потужного гібридного алгоритму оптимізації побудови маршрутів	Так	Ні	Так	Так	Ні	Так	Ні
8	Наявність мобільного додатку	Ні	Так	Так	Ні	Так	Ні	Ні
9	Звіти та аналітика	Планується в майбутньому	Базовий функціонал	Розширена звітність	Розширена звітність	Так	Так	Ні

Таким чином розроблений проект має свої переваги, особливо в питаннях вартості, використанні нових алгоритмів та можливостей оптимізації. Він дозволяє також змінювати свої маршрути, що доступно частково тільки у деяких аналогах. Крім того система має простий функціонал.

Проте, розширення можливостей інтеграції з іншими системами, подумати про створення мобільного додатку та додати звітність та аналітику, для того щоб підвищити конкурентоспроможність розробленої системи на ринку логістики сьогодні.

## 5.2 Технологічний аудит ідеї проекту

Після опису ідей проекту необхідно провести аудит технологій, які будуть використовуватись у проекті, та визначимо технологічну здійсненність вибраних ідей проекту.

Всі дані стосовно технологій та технологічної здійсненності показані нижче у таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Система управління логістикою з використанням гібридного генетичного алгоритму та алгоритму імітації відпалу для побудови маршрутів	Гібридний генетичний алгоритм, покращений алгоритм імітації відпалу; мова програмування C#, платформа .NET, карти	Гібридний алгоритм було розроблено, інші технології наявні	Всі технології доступні
2	Реалізація системи побудови альтернативних маршрутів	Генетичний алгоритм, мова програмування C#	Технології наявні	Технології доступні
3	Модуль перебудови маршрутів для конкретного транспортного засобу	Евристичний алгоритм, алгоритм імітації відпалу	Технології наявні	Технології доступні
4	Розробка інтерфейсу користувача для управління системою	Front-end фреймворк Razor	Технології наявні	Технології доступні
6	Інтеграція з веб картами	API Google maps	Технології наявні, API	Технології доступні

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
			карт можна отримати	

Як бачимо із таблиці технології необхідні для розроблення системи, такі як платформа .NET, генетичний алгоритм та покращений алгоритм імітації відпалу, мова програмування C#, API Google maps, база даних PostgreSQL, фреймворк Razor існують та є доступними. Майже всі технології є безкоштовними, що вказує на високу здійсненність проекту. Однак також є технологія, а саме гібридний генетичний алгоритм з покращеним алгоритмом імітації відпалу, яка не була наявною та доступною на ринку, та була спеціально розроблена для даної системи, щоб відповідати всім поставленим вимогам.

### 5.3 Аналіз ринкових можливостей запуску стартап-проекту

Тепер розглянемо ринкові можливості, які можна використати під час ринкового впровадження проекту. Дані про ринкові можливості наведено у таблиці 5.4

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	Кілька великих компаній, які надають подібні послуги (5-6), та менші локальні компанії, конкуренція середня-висока
2	Загальний обсяг продаж, грн/ум.од	Обсяг продажів оцінюється в кілька мільярдів доларів на рік

№ п/п	Показники стану ринку (найменування)	Характеристика
3	Динаміка ринку	Ринок логістики стабільно розвивається і його вартість зростає з кожним роком
4	Наявність обмежень для входу	Обмеження для входу помірні: великі витрати на розробку бізнес-продукту, необхідність спеціалізованих знань, конкуренція з боку встановлених компаній, вимоги щодо безпеки даних користувачів
5	Специфічні вимоги до стандартизації та сертифікації	Строге дотримання установлених норм безпеки клієнтів
6	Середня норма рентабельності в галузі (або по ринку), %	Середня норма рентабельності становить від 5 до 20 % у різних регіонах, залежно від потреби, у нашому регіоні близько 15%

Проаналізувавши характеристики потенційного ринку можна сказати що розроблений стартап має великі можливості успішно увійти на зростаючий ринок з високою потенційною рентабельністю. Незважаючи на наявність деяких обмежень для входу, таких як конкуренція та необхідність спеціалізованих знань, інноваційність розробленої системи завдяки використанню нового гібридного алгоритму та надання рідкісних послуг як можливість використання альтернативних маршрутів може стати конкурентною перевагою при вході на ринок.

Далі визначимо потенційні групи клієнтів, які будуть використовувати дану систему та їх характеристики такі як відмінності у поведінці вибраної цільової групи та вимоги споживачів до товару, та надамо перелік вимог до системи для кожного клієнта.

Дані про це запишемо у таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних цільових груп клієнтів	Вимоги споживачів до товару
1	Оптимізація логістичних маршрутів для зниження витрат та підвищення ефективності	Транспортні компанії з доставки вантажів та посилок	Різні розміри компаній, різний рівень технічної підготовки персоналу, відмінності в бюджетах, різна кількість маршрутів	До продукції: висока точність оптимізації маршрутів, швидка генерація маршрутів. До постачальника: надійність та технічна підтримка, безпека даних
2	Швидка доставка та оптимізація маршрутів для покращення сервісу клієнтів	Компанії з доставки їжі та кур'єрські служби	Висока частота замовлень, необхідність частої перебудови маршрутів	До продукції: Швидка перебудова маршрутів, простий інтерфейс До постачальника:

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних цільових груп клієнтів	Вимоги споживачів до товару
				технічна підтримка
3	Планування оптимальних туристичних маршрутів для підвищення задоволеності клієнтів	Туристичні компанії	Персоналізація під уподобання клієнтів, попит залежить від сезону	Можливість персоналізації маршрутів, технічна підтримка, простий інтерфейс
4	Оптимізація персональних маршрутів для економії часу та ресурсів	Індивідуальні користувачі	Різний рівень технічних знань, чутливість до вартості послуг	Простий інтерфейс, мінімальна ціна або безкоштовний сервіс
5	Ефективне планування доставки будматеріалів та обладнання	Будівельні компанії	Різна вага вантажів, різна кількість техніки	Врахування великогабаритни х вантажів, простий інтерфейс, техпідтримка

За результатами дослідження можна сказати, що розроблена система може задовольнити потреби багатьох клієнтів у різних галузях. Кожна цільова аудиторія має свої вимоги та очікування від запропонованої системи.

Далі розглянемо фактори, які можуть перешкоджати впровадженню проекту на розглянутий ринок. Аналіз факторів ризиків та можливої реакції компанії на них наведено у таблиці 5.6.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція з боку відомих компаній	Ймовірність не використання продукту через серйозних конкурентів	Впровадження новітнього підходу, зокрема нового алгоритму та можливості альтернативних маршрутів
2	Постійне оновлення підходів у логістиці	Швидкий розвиток технологій у управлінні логістикою може зробити систему повільною та не відповідною новим вимогам	Моніторинг нових технологій у сфері, гнучка архітектура системи для швидкого впровадження нововведень
3	Масштабування	При збільшенні кількості користувачів можуть виникнути проблеми з продуктивністю	Розробка архітектури, яку легко можна масштабувати, використання хмар для зберігання даних
4	Технічні поломки	Можливі поломки системи, збої у роботі	Регулярні оновлення та підтримка розробленого продукту
5	Вразливості безпеки	Можливі атаки хакерів або витік даних	Надійний захист даних у системі та оновлення протоколів безпеки регулярно

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
6	Викрадення технологій	Можливе порушення авторських прав та привласнення технологій системи	Реєстрація власного продукту

Є чимало загроз на ринку логістики, зокрема конкуренція відомих брендів, постійний розвиток технологій, масштабування системи та можливі вразливості безпеки системи, проте при дотриманні розглянутих вище у таблиці порад можна звести до мінімуму вплив всіх цих загроз.

Незважаючи на певні загрози впровадження продукту на ринок логістики є також ряд можливостей, які сприяють та допомагають для виводу продукту на ринок. Детальніше ці можливості розглянуто у таблиці 5.7.

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання логістичного ринку	Ринок логістики стабільно розвивається та збільшується попит відповідно	Розширення функціоналу для підтримки різних типів клієнтів, швидкий вихід на ринок який розвивається
2	Використання нових технологій як переваги	Використання інноваційного гібридного алгоритму та генерації альтернативних маршрутів як переваги	Підкреслення унікальності технології під час реклами продукту

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3	Мала кількість рішень для невеликих бізнесів	Малі підприємства шукають доступні рішення для оптимізації логістики, але часто стикаються з високими цінами	Пропозиція гнучких тарифних планів, можливе впровадження безкоштовного сервісу на певний час(на 1 рік)
4	Можливість отримання інвестицій	Отримання додаткових інвестицій від інвесторів	Залучення інвесторів за допомогою якісного бізнес-плану
5	Вихід на міжнародний ринок	Можливість використання системи для планування міжнародних маршрутів	Локалізація інтерфейсів під різні мови, адаптація системи до умов різних країн

Отже, ринкове середовище логістики хоч і має чимало загроз проте надає також такі можливості, як стабільне зростання ринку, недостатня кількість рішень для малого бізнесу та просування продукту як системи з новим алгоритмом, що дозволяє позитивно дивитись на вихід розробленої системи на ринок.

Для повного дослідження ринку проведемо також ступеневий аналіз конкуренції. Дані аналізу у таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного Середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Тип конкуренції: Монополія	На ринку присутня певна кількість компаній, що	Впровадження новітніх технологій та алгоритмів,

Особливості конкурентного Середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
	пропонують програмні рішення для управління логістикою	реклама, безкоштовний сервіс протягом першого року запуску системи
Рівень конкурентної Боротьби: локальний, міжнародний	Конкуренція з міжнародними компаніями для великого бізнесу та конкуренція з локальними системами для малого бізнесу та персональних потреб	Адаптація до умов ринку, створення партнерств з місцевими компаніями
Галузева ознака конкуренції: Внутрішньогалузева конкуренція	Конкуренція між компаніями, що пропонують схожі програмні продукти	Виділення серед конкурентів через унікальні пропозиції та нові технології
Конкуренція за видом товару: товарно-видова	Порівняння продуктів за функціональністю, ціною та якістю обслуговування	Фокус на унікальності продукту
Характер конкурентних переваг: Нецінова конкуренція	Конкуренція базується на якості продукту, інноваціях, функціональності	Запровадження нового функціоналу, висока якість
Інтенсивність конкуренції: Марочна конкуренція	Впізнаваність бренду	Рекламна кампанія

Отже, було проведено ступеневий аналіз конкуренції де було визначено основні особливості конкурентного середовища проте необхідно провести більш

детальний аналіз конкуренції за моделлю 5 сил М. Портера. Деталі аналізу у таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М.Портером

	Прямі конкуренти в галузі	Потенційні конкуренти (бар'єри входження на ринок)	Постачальники (фактори сили постачальників)	Клієнти (фактори сили клієнтів)	Товари замітники
Складові аналізу	Route4Me, MyRouteOnline, Paragon	Бар'єрами для входу на ринок є: необхідність спеціальних знань, конкуренція відомих брендів	Обмежена кількість якісних постачальників, залежність від API карт	Вибір альтернатив, чутливість до цін,	Ручне планування для малого бізнесу чи особистого використання
Висновки	На ринку багато компаній що пропонують схожі продукти, потрібно виділятися через унікальні функції та нові технології	Є можливості для входу нових гравців на ринок, проте це потребує чималої кількості знань та часу	Постачальники частково диктують умови на ринку, зокрема встановлюючи свою цінову політику для їхніх послуг	Вибір альтернатив дає клієнтам силу, тому потрібні орієнтуватись на їхні вимоги	Товари-замітники впливають в основному на ринок малого бізнесу тому необхідно показувати переваги свого продукту

Отже, ми розглянули аналіз конкуренції за моделлю 5 сил М. Портера і тепер на основі всіх раніше проведених аналізів визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Деталі у таблиці 5.10

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Унікальність продукту	Використання нового гібридного генетичного алгоритму та алгоритму

№	Фактор конкурентоспроможності	Обґрунтування
		імітації відпалу надає перевагу у порівнянні із стандартними методами конкурентів
2	Безкоштовний сервіс на початку і низькі ціни пізніше	Безкоштовне або недороге рішення робить систему привабливою для малих підприємств, які обмежені в бюджеті. Це розширює потенційну клієнтську базу та підвищує конкурентоспроможність на ринку.
3	Генерація альтернативних маршрутів	Система може генерувати кілька альтернативних рішень, надаючи клієнтам можливість вибору підходящого маршруту.
4	Швидка перебудова маршрутів з малими втратами по точності	Використання алгоритму імітації відпалу дозволяє швидко перебудовувати маршрути для конкретних транспортних засобів.
5	Орієнтація на різні типи клієнтів	Система може застосовуватися в різних галузях бізнесу, таких як служби доставки, поштові служби, туристичні фірми та для особистого використання для економії коштів та часу
6	Якість і надійність	Гарна якість і надійність системи може стати вирішальним фактором
7	Адаптивність	Систему можна адаптувати під різні потреби

Далі на основі визначених факторів проведемо аналіз сильних та слабких сторін стартап-проекту. Деталі аналізу будуть у таблиці 5.11.

Таблиця 5.11 – Аналіз сильних та слабких сторін системи управління логістикою з використанням гібридного алгоритму багатошляхової маршрутизації

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розробленою системою						
			-3	-2	-1	0	+1	+2	+3
1	Унікальність продукту	17	0	1	0	0	0	0	0
2	Безкоштовний сервіс	16	0	1	0	0	0	0	0
3	Генерація альтернативних маршрутів	15	0	0	1	0	0	0	0
4	Швидка перебудова конкретних маршрутів	13	0	0	0	1	0	0	0
5	Орієнтація на різні галузі ринку	14	0	0	1	0	0	0	0
6	Якість і надійність	10	0	0	0	0	1	0	0
7	Адаптивність	12	0	0	0	0	1	0	0

Отже, ми розглянули порівняльний аналіз сильних та слабких сторін проекту і тепер необхідно скласти матрицю SWOT-аналізу(таблиця 5.12)

Таблиця 5.12 – SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> <li>Унікальний гібридний алгоритм оптимізації</li> <li>Безкоштовний або доступний сервіс(пізніше)</li> <li>Генерація альтернативних маршрутів</li> </ol>	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> <li>Менші можливості інтеграції з іншими системами</li> <li>Необхідність покращення масштабованості та продуктивності:</li> <li>Низька впізнаваність бренду:</li> </ol>
--	---

<ul style="list-style-type: none"> <li>4. Швидка перебудова маршрутів</li> <li>5. Орієнтація на різні галузі ринку</li> <li>6. Якість і надійність</li> <li>7. Адаптивність</li> </ul>	<ul style="list-style-type: none"> <li>4. Обмежена технічна підтримка:</li> <li>5. Відсутність деяких функцій</li> </ul>
<p>Можливості:</p> <ul style="list-style-type: none"> <li>1. Зростання логістичного ринку</li> <li>2. Використання нових технологій як переваги</li> <li>3. Мала кількість рішень для невеликих бізнесів</li> <li>4. Можливість отримання інвестицій</li> <li>5. Вихід на міжнародний ринок</li> </ul>	<p>Загрози:</p> <ul style="list-style-type: none"> <li>1. Конкуренція з боку відомих компаній</li> <li>2. Постійне оновлення підходів у логістиці</li> <li>3. Необхідність масштабування</li> <li>4. Технічні поломки</li> <li>5. Вразливості безпеки</li> <li>6. Викрадення технологій</li> </ul>

SWOT-аналіз показує, що стартап-проект має значні сильні сторони, такі як інноваційний алгоритм та нові функції та має ряд можливостей, які можна використати для успішного виходу на ринок. Проте слід також звертати увагу на слабкі сторони проекту та потенційні загрози, на які слід реагувати.

На основі SWOT-аналізу визначимо альтернативи ринкового впровадження стартап-проекту(див. таблицю 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розширення функціональності та додавання інтеграцій з різними сервісами	Висока	5 місяців
2	Розробка мобільного додатку	Висока	7 місяців
3	Підвищення якості та безпеки	Низька	4 місяці

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
4	Вихід на міжнародний ринок	Середня	9 місяців
5	Пошук партнерів серед компаній	Середня	6 місяців
6	Залучення додаткових інвестицій	Низька	12 місяців

На основі даного аналізу можна сказати, що насамперед варто зосередитись на розширенні функціональності системи та впровадженню її інтеграцій з різними сервісами. Також чудовою альтернативою є розробка мобільного додатку, яка дозволить залучити нових користувачів, але це буде потребувати більшої кількості часу.

#### 5.4 Розроблення ринкової стратегії продукту

Розглянемо цільові групи потенційних споживачів та їх готовність прийняти продукт. Деталі аналізу показані у таблиці 5.14

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі та середні транспортні компанії з доставки вантажів та посилок	Висока	Великий	Середня	Простий вхід

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
2	Великі транспортні компанії з доставки вантажів	Середня	Великий	Середня	Середня простота входу
3	Компанії з доставки їжі та кур'єрські служби	Висока	Великий	Середня	Простий вхід
4	Індивідуальні користувачі	Висока	Середній	Середня	Простий вхід
5	Будівельні компанії	Середня	Середня	Середня	Середня простота входу

З даних аналізу можна сказати, що попит на продукт в цілому достатньо високий та є багато цільових груп, які можуть прийняти продукт. Пріоритетними цільовими групами для проекту є насамперед малі та середні транспортні компанії, компанії з доставки їжі та кур'єрські служби. Також варто звернути увагу на ринок індивідуальних користувачів. Для успішного заходу в кожен галузь необхідно адаптувати продукт під специфічні потреби клієнтів.

Далі сформуємо базову стратегію розвитку. Аналіз базової стратегії ринку показано у таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Швидкий вихід на ринок з унікальною пропозицією	Фокус на малий та середній бізнес	Унікальний гібридний алгоритм оптимізації, безкоштовний сервіс, генерація альтернативних маршрутів	Стратегія спеціалізації
2	Розширення функціоналу та інтеграція з іншими системами	Фокус на великі компанії, та зосередженість на різних галузях ринку	Персоналізація під потреби клієнта, широкий функціонал, унікальність продукту	Стратегія диференціації
3	Вихід на міжнародний ринок	Фокус на міжнародні компанії	Адаптація продукту до місцевих ринків, розширення клієнтської бази	Стратегія диференціації
4	Розробка мобільного додатку	Фокус на індивідуальних користувачів, малий бізнес та на зручність використання	Підвищення зручності використання продукту	Стратегія диференціації
5	Партнерство з відомими українськими компаніями	Фокус на великі транспортні компанії	Підвищення впізнаваності бренду	Стратегія спеціалізації

Як видно з аналізу необхідно почати зі стратегії диференціації, використовуючи унікальні переваги продукту для залучення клієнтів у пріоритетних сегментах та поступово розширювати функціональність та інтеграції, переходячи до стратегії спеціалізації.

Далі визначимо базову стратегію поведінки на ринку(таблиця 5.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів ?	Чи буде компанія копіювати основні характеристик и товару конкурента, і які?	Стратегія конкурентно ї поведінки
1 (Швидкий вихід проекту на ринок)	Ні	Шукати нових споживачів та забирати існуючих	Компанія буде повторювати основні функції конкурента, проте буде надавати власні унікальні можливості	Стратегія наслідування лідеру
2 (Розширення функціоналу та інтеграція з іншими системами)	Ні	Шукати нових споживачів та забирати існуючих	Компанія буде пропонувати нові унікальні алгоритми та функції	Стратегія заняття конкурентно ї ніші

На основі вимог споживачів з обраних галузей стартап-проекту та до продукту та в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування (таблиця 5.17), яка полягає у формуванні ринкової позиції щодо проекту.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Висока точність оптимізації маршрутів, генерація альтернативних маршрутів, інтеграція з існуючими системами	Стратегія диференціації, стратегія спеціалізації	Унікальний гібридний алгоритм оптимізації генерація альтернативних маршрутів	Ефективна оптимізація маршрутів, інноваційність продукту, надійність
2	Швидка перебудова маршрутів в реальному часі, простий інтерфейс	Стратегія диференціації	Швидка перебудова маршрутів, якість	Простий інтерфейс, швидка перебудова маршруту, Доступність
3	Простий інтерфейс, мінімальна ціна або безкоштовний	Стратегія диференціації	Цінова доступність, простота	Доступність, простота, клієнтоорієнтованість

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
	сервіс, персоналізація сервісу			

З аналізу видно що стратегія позиціонування для стартап-проекту базується на диференціації та підкресленні унікальних переваг, які відповідають вимогам цільових аудиторій. Три ключові асоціації, що формують комплексну позицію проекту, включають ефективність та інноваційність рішень, доступність, а також клієнтоорієнтованість та високу якість запропонованої системи.

#### 5.4 Розроблення маркетингової програми стартап-проекту

Для розроблення маркетингової стратегії необхідно спочатку визначити ключові переваги концепції потенційного товару(таблиця 5.18).

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Ефективна оптимізація маршрутів з високою точністю	Підвищення ефективності логістичних операцій, можливість використання альтернативних маршрутів	Унікальний гібридний алгоритм оптимізації та можливість вибирати альтернативні маршрути
2	Доступне або безкоштовне рішення для	Конкурентоспроможність	Доступний сервіс, орієнтованість на клієнтів

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
	оптимізації маршрутів		
3	Швидка перебудова маршрутів в реальному часі	Оперативна реакція на зміни	Використання швидкого алгоритму для перебудови маршруту
4	Мобільний додаток для побудови маршрутів	Зручність використання	Надання зручного мобільного додатку (планується)

З аналізу бачимо, що основними перевагами стартап-проекту є унікальні технологічні рішення, доступність, гнучкість та орієнтацію на клієнта.

Далі розробимо трирівневу маркетингову модель товару: уточнимо ідею продукту, його фізичні складові, особливості процесу його надання(таблиця 5.19)

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Базова потреба споживача, яку задовольняє товар - ефективна оптимізація маршрутів з можливістю перебудови шляхів Основні функціональні вигоди: нова технологія, генерація альтернативних маршрутів		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Гл/Е/Ор
	1. Унікальний гібридний алгоритм оптимізації 2.Генерація альтернативних маршрутів	Немонотонний	Технологія

Рівні товару	Сутність та складові		
	3. Швидка перебудова маршрутів		
	4. Простий інтерфейс		
	5. Безкоштовний сервіс		
	Якість: відповідність стандартам безпеки інформації, використання сучасних засобів		
	Пакування: цифрове пакування		
	Марка: VajenStudio + EffectRoute		
III. Товар із підкріпленням	До продажу: Тестування продукту тестовими користувачами, консультації щодо продукту		
	Після продажу: технічна підтримка через електронну пошту, виправлення несправностей		
Товар буде захищений від копіювання спочатку ліцензією коду, а потім реєстрацією торгової марки			

Отже, ми описали трирівневу модель товару, що допомогло визначити, яку цінність продукт надає клієнтам, які його основні характеристики та як він буде підтримуватися та захищатися.

Далі визначимо цінові межі проекту(таблиця 5.20).

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на послугу
1	Безкоштовні або низькі(Гугл-карти, Excel)	Рівень цін становить від 20 до 500+ доларів	Середній рівень доходів(малі та середні компанії доставки)	Нижня межа: безкоштовно з обмеженням по

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на послугу
				кількості маршрутів Верхня межа: 100 доларів на місяць
2	Великі витрати (власні розробки або ручне планування)	Рівень цін становить від 50 до 500+ доларів	Високий рівень доходів(великі компанії доставки)	Нижня межа: безкоштовний пробний період на місяць-два і 20 доларів потім Верхня межа: 200 доларів на місяць
3	Безкоштовні або низькі(ручне планування, гугл-карти)	Рівень цін на аналоги становить від безкоштовного до 50 доларів	Низький рівень доходів (особисте планування маршрутів)	Нижня межа: безкоштовно Верхня межа: 10 доларів на місяць
4	Безкоштовні мобільні додатки для навігації	Від 20 до 50 доларів на місяць	Середній рівень доходів(служби доставки їжі)	Нижня межа: безкоштовний пробний період на місяць-два і 10 доларів потім Верхня межа: 25 доларів на місяць

З таблиці 5.20 видно що потрібно надавати безкоштовні або дешеві послуги для особистого користування і для малих компаній і брати більше коштів з великих компаній з складнішими маршрутами та більшими потребами.

Далі визначимо оптимальну систему збуту(таблиця 5.21).

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Малі та середні транспортні компанії шукають доступні рішення, проводять короткі переговори або купляють онлайн	Створення реклами на різних сайтах, демонстрація роботи продукту, онлайн-продажі	Прямий канал	Продаж через офіційний веб-сайт, використання інтернет реклами
2	Великі транспортні компанії вимагають якісних рішень та інтеграції з різними системами	Надання пропозицій із презентацією, переговори, технічна підтримка	Прямий канал	Проведення особистих перемовин з компаніями
3	Індивідуальні клієнти вимагають простих дешевих рішень	Створення реклами на різних сайтах, технічна підтримка	Прямий канал	Розміщення реклами, онлайн-продажі
4	Компанії з доставки їжі вимагають надійних, доступних швидких рішень	Створення реклами на різних сайтах, техпідтримка, проведення консультацій	Прямий канал	Розміщення реклами, онлайн-продажі

Тепер розробимо концепцію маркетингових комунікацій для різних груп клієнтів(таблиця 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п / п	Специфіка поведінки / цільових клієнтів	Канали комунікацій, якими користуються клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Малі компанії шукають доступні рішення	Інтернет-реклама, соцмережі, професійні форуми	Доступність продукту, ефективність, висока якість	Повідомити про доступність і простоту продукту	«Оптимізуйте маршрути легко та доступно»
2	Великі транспортні компанії вибирають технологічні та інноваційні рішення	Персональні канали комунікації, професійні заходи, конференції	Надійність продукту, унікальність, інноваційність	Повідомити про новий унікальний алгоритм маршрутизації та нові функції генерації альтернативних маршрутів	«Управляйте логістикою з новим інноваційним підходом»
3	Компанії з доставки їжі вимагають надійних, доступних	Соцмережі, реклама в додатках	Швидкість та оперативність	Показати функції швидкої перебудови маршрутів і оптимальне	«Будуй та змінюй маршрути швидко»

№ п / п	Специфіка поведінки / цільових клієнтів	Канали комунікацій, якими користуються клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	швидких рішень			планування маршрутів	
4	Персональні користувачі шукають прості та доступні рішення	Соцмережі, реклама в додатках	Доступність та простота продукту	Повідомити про легкий метод зменшення часу на свої подорожі	«Оптимізація твоїх маршрутів за кілька хвилин»

З проведеного аналізу можна зробити висновки, що до кожної цільової групи користувачів потрібно обирати окрему стратегію, показуючи ті переваги стартап проекту, які є найбільш необхідними для обраної групи користувачів, зокрема для індивідуальних користувачів та малих компаній потрібно робити акцент на доступності розробленого проекту та швидкості роботи, для великих транспортних компаній необхідно робити акцент на новітньому підході до побудови та планування маршрутів, а для компаній з доставки їжі робити акцент на швидкій можливій перебудові маршрутів. Застосовуючи індивідуальні стратегії для кожної групи можна досягти високих показників зацікавленості та використання розробленого проекту серед різних груп користувачів.

#### Висновки до розділу 5

На основі проведеного аналізу можна зробити наступні висновки по проекту. Насамперед варто вказати, що продукт має всі шанси бути успішно виведеним на ринок, адже ринок логістичних послуг має великий попит та постійно зростає. Крім

того, продукт є унікальним, та використовує інноваційний підхід до побудови логістичних маршрутів. Рентабельність проекту також повинна бути гарною, зважаючи на середню норму рентабельності по ринку. Також низькі витрати на розробку забезпечують потенціал для отримання прибутку.

Даний проект має непогані перспективи впровадження, так як є кілька потенційних груп клієнтів, які можуть використовувати розроблену систему. Конкуренція у галузі існує, проте за рахунок унікальності продукту можна захопити свою частку на ринку. Бар'єри входження на ринок можна подолати.

Найбільш доцільно обрати альтернативу швидкого виходу на ринок з унікальною пропозицією, з подальшим розвитком функціоналу проекту та розширенням ринків збуту.

Реалізація проекту є доцільною, оскільки він має всі передумови для успішного виходу на ринок та розвитку, такі як унікальний алгоритм маршрутизації, доступність, можливість генерації альтернативних маршрутів, швидка регенерація маршрутів та інше.

Загалом проект має значний потенціал для впровадження на ринок завдяки наявному попиту, унікальним перевагам та чіткій стратегії розвитку.

## ВИСНОВКИ

У даній дисертації було проведено велике дослідження у сфері логістики та описано розробку системи управління логістикою транспортної компанії з використанням гібридного алгоритму маршрутизації транспортних засобів.

Було розглянуто актуальність вибраного дослідження, його мету та сформульовано основні завдання роботи.

У першому розділі роботи було проаналізовано поточний стан логістичної сфери, зокрема розглянуто основні тенденції розвитку сучасних логістичних систем, які включають впровадження штучного інтелекту, автоматизацію логістичних систем, розроблення нових алгоритмів маршрутизації, використання робототехніки. Крім того, було виокремлено основні виклики і проблеми в логістичних системах сьогодні, зробивши акцент на «доставці останньої милі». Також було проаналізовано такі системи як Route4Me, OptimoRoute MyRouteOnline та Paragon з визначенням всіх їхніх переваг та недоліків. На основі цього аналізу було визначено основні метрики при побудові оптимальних маршрутів.

У другому розділі роботи було показано зв'язок теорії графів та маршрутизації та вказано як теорія графів допомагає вирішувати задачі маршрутизації. Також було наведено класифікацію методів та алгоритмів для задачі створення та оптимізації маршрутів виокремивши точні, евристичні, метаевристичні та гібридні класи методів та проведено порівняльний аналіз кількох гібридних алгоритмів для визначення їхніх переваг та недоліків та врахування цього при побудові власного гібридного алгоритму.

У третьому розділі було наведено математичну постановку задачі маршрутизації транспортних засобів, яка вирішується в рамках даної роботи. Також було описано гібридний генетичний алгоритм з покращеним алгоритмом імітації відпалу з усіма етапами та показано адаптацію даного алгоритму до генерації альтернативних розв'язків та регенерації конкретного маршруту. Крім того, сформульовано всі переваги даного алгоритму та потенційні недоліки.

У наступному розділі представлено архітектуру розробленої системи та наведено засоби та інструменти для розробки системи, а саме платформа .NET для серверної частини та бібліотека React.js для клієнтської частини та вказано причини такого вибору. Також показано функціонал розробленого додатку з скріншотами та детальним поясненням всіх функцій, зокрема вказано що основними функціями є введення даних через форму і файл, генерація маршрутів, побудова маршрутів на карті, відображення результатів на екрані та можливості вибору альтернативного маршруту та перебудови маршруту для вибраного транспортного засобу. Крім того, було протестовано всю систему на наявність помилок та коректної роботи.

В останньому розділі роботи були описані основні ідеї проекту та проведений технологічний аудит ідей проекту. Додатково було проаналізовано ринкові можливості запуску проекту та визначено що запуск проект має всі шанси на успіх так як ринок логістики сприяє цьому завдяки попиту та постійному розвитку ринку. Крім того було розроблено маркетингову програму стартап-проекту відповідно від різних цільових груп користувачів.

Отже, можна зробити висновки, що розроблена всіх цілей було досягнуто та розроблену ефективну систему управління логістикою з використанням нового гібридного алгоритму маршрутизації. Розроблена система є простою у використанні, швидкою та ефективною при плануванні маршрутів, що дозволяє їй гідно конкурувати з аналогами на ринку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В.В. Заграй, А. В. Коган. Гібридний алгоритм для вирішення задач багатошляхової маршрутизації в логістиці. Збірник тез доповідей II Міжнародної науково-практичної інтернет-конференції «Інновації та перспективні шляхи розвитку інформаційних технологій». – 2023. – pp. 20-22.
2. Л. Ф. Гуляницький, О. Ю. Мулеса. Прикладні методи комбінаторної оптимізації. – 2016 – pp. 25-28.
3. 27+ Logistics Statistics & Facts (Must Know in 2024) // Contimod. – 2024. – URL: <https://www.altexsoft.com/blog/transportation-management-system/>
4. AI in logistics: Transforming Operational Efficiency in Transportation Businesses // SPD Technology – 2024. – URL: <https://spd.tech/artificial-intelligence/ai-in-logistics-transforming-operational-efficiency-in-transportation-businesses/>
5. 2024 Challenges in the Logistics Industry and the Solutions // Alpega – 2024. – URL: <https://www.alpegagroup.com/en-en/community/blog/2024-challenges-in-the-logistics-industry-and-the-solutions/>
6. Robotics in logistics: how much are companies investing in robots? // Synex Logistics – 2024. – URL: <https://synexlogistics.com/articles/robotics-in-logistics-how-much-are-companies-investing-in-robots/>
7. Emerging trends in international logistics: Challenges and opportunities // Antonio Marco – 2024. – URL: <https://antoniomarco.com/news/en/emerging-trends-in-international-logistics-challenges-and-opportunities/>
8. Key Challenges in Logistics Management: Strategies to Overcome Them // Inbound Logistics – 2024. – URL: <https://www.inboundlogistics.com/articles/logistics-management-challenges/>
9. OptimoRoute // OptimoRoute – 2024. – URL: <https://optimoroute.com/>
10. Simplify Last Mile Transportation //Route4Me – 2024. – URL: <https://www.route4me.com/>
11. Optimize Your Routes Online // MyRouteOnline – 2024. – URL: <https://www.myrouteonline.com/>

12. Advanced routing and scheduling software // Paragon – 2024. – URL: <https://www.paragonrouting.com/en-us/>
13. L. Volner. Using graph theory in logistics from a 21st century perspective. – 2007.
14. N. Aulia, R.M. Prihandini, A.B. Agatha. Graph Theory: Applications of Graph Theory in Delivery Systems – 2024.
15. Теорія графів. Навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Комп’ютерний моніторинг та геометричне моделювання процесів і систем» спеціальності 122 «Комп’ютерні науки»/ І.М. Кузьменко; КПІ ім. Ігоря Сікорського. — Київ: КПІ ім. Ігоря Сікорського, 2020. — 71 с.
16. M. Fischetti, P. Toth, D. Vigo. A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs. – 1994.. – p. 854
17. A. N. Letchford, J. Lysgaard. A Branch-and-Cut Algorithm for the Capacitated Open Vehicle Routing Problem. – 2007. – pp.1646-1647.
18. A. L. Kok, H. Kopfer. Dynamic programming algorithm for the vehicle routing problem with time windows and EC social legislation – 2009 – pp. 10-20.
19. M. Povlovitsch, A. Mendes. Column Generation for a Multitrip Vehicle Routing Problem with Time Windows, Driver Work Hours, and Heterogeneous Fleet – 2023.
20. J. Desrosiers, M. Lubbecke. Branch-Price-and-Cut Algorithms – 2011.
21. C. L. Reeves. Heuristic Search Methods: A Review. – 1996.
22. T. Pichpibul, R. Kawtummachai. An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem – 2012.
23. S. Komert. A cluster first-route second approach for a capacitated vehicle routing problem: A case study – 2018.
24. L. Martins, C. Bayliss. A Savings-Based Heuristic for Solving the Omnichannel Vehicle Routing Problem with Pick-up and Delivery – 2020.
25. D. Pisinger, S. Ropke. A general heuristic for vehicle routing problems – 2005.
26. A. Salah. Genetic algorithms – 2020.

27. F. Busetti. Simulated Annealing overview. – 2001. – pp. 2-6.
28. M. Dorigo, M. Birattari, T. Stutzle. Ant Colony Optimization. – 2006. – pp. 28-29.
29. F.Glover, M. Laguna, R. Marti. Principles of Tabu Search – 2008.
30. Understanding Particle Swarm Optimization (PSO): From Basics to Brilliance // – 2024. – URL: <https://thisisrishi.medium.com/understanding-particle-swarm-optimization-pso-from-basics-to-brilliance-d0373ad059b6>
31. R. Shahbazian. Integrating Machine Learning Into Vehicle Routing Problem: Methods and Applications – 2024.
32. C. Martinez, M.Lozano. Local Search Based on Genetic Algorithms – 2007.
33. Система управління логістикою транспортної компанії з доставки замовлень. pp. 45-46 // В. Заграй – 2023. – URL: <https://ela.kpi.ua/items/08d6cc1a-6374-4bef-b98a-312a440eda8a>
34. E. Figielska. Column generation technique combined with a tabu search algorithm for scheduling in a hybrid flowshop. – 2011.
35. An artificial bee colony algorithm with local search for vehicle routing problem with backhauls and time windows // N. Tuntitippawan – 2016. URL: <https://www.researchgate.net/publication/310239236>
36. A. A. Ibrahim, R. O. Abdulaziz, J. A. Ishaya. Capacitated Vehicle Routing Problem. – 2019.
37. A.Chen, T.Jiang. Comparison of GA-SA, GA, and SA in terms of convergence. p.6 – 2012.
38. A tour of the C# language // Microsoft – 2024. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
39. The library for web and native user interfaces. // React – 2024 – URL: <https://react.dev/>
40. Google Maps Platform. Google for Developers // Google Maps Platform – 2024. - URL: <https://developers.google.com/maps?hl=en>