

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра Інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д.т.н., проф.,
_____ Ю.П. Кондратенко
« ____ » _____ 2022 року

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**СИСТЕМА ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ
ТРАНСПОРТНИХ МАРШРУТІВ В УМОВАХ
НЕВИЗНАЧЕНОСТІ**

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810315

Виконав студент 4-го курсу, групи 401
_____ *О. Ю. Кулагін*
« ____ » червня 2022 р.

Керівник: канд. тех. наук, доцент
_____ *О. В. Козлов*
« ____ » червня 2022 р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. П. Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**

Спеціальність **122 «Комп'ютерні науки»**

(шифр і назва)

Галузь знань **12 «Інформаційні технології»**

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю.П. Кондратенко

« ____ » _____ 20__ р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Кулагіну Олексію Юрійовичу.

1. Тема бакалаврської кваліфікаційної роботи «Система планування та оптимізації транспортних маршрутів в умовах невизначеності».

Керівник роботи Козлов Олексій Валерійович канд. тех. наук, доцент.

Затв. наказом Ректора ЧНУ ім.Петра Могили від « 07 » грудня 2021 р. № 318

2. Строк представлення дипломної роботи студентом « ____ » _____ 2022 р.

3. Вхідні (початкові) дані до роботи: забезпечити введення даних згідно з предметною областю та постановленою задачею оптимізації та планування, розробити застосунок, за допомогою якого забезпечити реалізацію системи прийняття рішення в умовах невизначеності за допомогою картографічного сервісу, доступ до інформації організувати за допомогою графічного інтерфейсу.

Очікуваний результат роботи: система оптимізації прогнозування, збору даних та прийняття рішення для планування за визначеними критеріями.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):
- огляд предметної сфери з побудови маршрутів та розрахунку часу проходження;
 - огляд наявних аналогів картографічних сервісів та систем прийняття рішення;
 - експертне оцінювання критеріїв прийняття рішення;
 - вибір технічного та програмного забезпечення для реалізації програмного забезпечення;
 - програмна реалізація системи планування та оптимізації транспортних маршрутів в умовах невизначеності.
5. Перелік графічних матеріалів: презентація .
6. Завдання до спеціальної частини: «Санітарно-гігієнічні вимоги приміщення для роботи з комп'ютерною технікою».
7. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	старший викладач, Макарова О. В.	

Керівник роботи канд. тех. наук, доцент Козлов О. В.

(наук. степінь, вчене звання, прізвище, ініціали)

(підпис)

Завдання прийнято до виконання Кулагін О. Ю.

(прізвище та ініціали)

(підпис)

Дата видачі завдання « 23 » листопада 2021 р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: «Система планування та оптимізації транспортних маршрутів в умовах невизначеності».

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівника БКР	26.10.2021	30.10.2021	виконано
2	Отримання завдання на виконання БКР	25.11.2021	25.11.2021	виконано
3	Складання календарного плану роботи на весь період виконання БКР	26.11.2021	10.12.2021	виконано
4	Дослідження літературних джерел, аналіз предметної області побудови маршрутів та прогнозування часу їх проходження	11.12.2021	20.02.2022	виконано
5	Аналіз існуючих аналогів картографічних веб-сервісів	21.02.2022	08.02.2022	виконано
6	Огляд наявних методів оптимізації прогнозування заторів	08.02.2022	16.02.2022	виконано
7	Розробка методології оптимізації впливу заторів на побудову маршрутів	16.02.2022	28.02.2022	виконано
8	Розробка системи прийняття рішень з планування відправлень за маршрутом	01.03.2022	05.03.2022	виконано
9	Вибір технічного та програмного забезпечення	05.03.2022	16.04.2022	виконано
10	Розробка програмного забезпечення	17.04.2022	20.05.2022	виконано
11	Тестування системи	10.05.2022	22.05.2022	виконано
12	Збір та аналіз статистики з використання програмного забезпечення, оформлення БКР	22.05.2022	16.06.2022	виконано
13	Розробка спеціальної частини з охорони праці	16.05.2022	22.05.2022	виконано
14	Проходження переддипломної практики	26.05.2022	05.06.2022	виконано
15	Попередній захист БКР	30.05.2022	30.05.2022	виконано
16	Подання БКР рецензенту	16.05.2022	18.05.2022	виконано
17	Створення презентації для захисту та написання доповіді	18.06.2022	20.06.2022	виконано
18	Подання БКР, її електронної копії та інших документів до захисту	20.06.2022	22.06.2022	виконано
19	Захист БКР перед ЕК	27.06.2022	29.06.2022	виконано

Розробив студент Кулагін О. Ю.
(прізвище та ініціали)

_____ (підпис)

Керівник роботи канд. тех. наук, доцент Козлов Олексій Валерійович
(ступень, звання, прізвище та ініціали)

_____ (підпис)

« ____ » _____ 202_ р.

АНОТАЦІЯ до бакалаврської роботи

Тема: «Системи планування та оптимізації транспортних маршрутів в умовах невизначеності»

Студент: Кулагін Олексій Юрійович
Керівник: канд. тех. наук, доцент Козлов Олексій Валерійович

Бакалаврська кваліфікаційна робота присвячена розробці, програмній реалізації та впровадженню системи оптимізації та планування транспортних маршрутів в умовах невизначеності.

Об'єкт дослідження – процеси оптимізації маршрутів в умовах невизначеності.

Предмет дослідження – технології та програмні засоби планування та оптимізації транспортних маршрутів в умовах невизначеності.

Мета – підвищення ефективності оцінювання часу проходження та планування маршрутів шляхом розробки програмного застосунку з використанням прикладних програмних інтерфейсів для картографічних сервісів.

Бакалаврська кваліфікаційна робота складається з фахової частини і спеціальної частини з охорони праці. Пояснювальна записка дипломної роботи складається зі вступу, трьох розділів, висновків та додатків.

У першому розділі здійснено аналіз сучасних картографічних сервісів. У другому розділі розглядаються технології та засоби розробки системи оптимізації, планування та прийняття рішень. У третьому розділі описано проектування та програмну реалізацію системи оптимізації та планування транспортних маршрутів в умовах невизначеності.

В результаті виконаної роботи було зроблено висновки щодо можливості та вагомості оптимізації процесу оцінювання часу проходження запланованого маршруту та подальшого процесу прийняття рішень з кращого часу відправлення за визначеним маршрутом.

Сторінок – 105, таблиць – 13, рисунків – 42, посилань – 28, додатків – 3.

Ключові слова: картографічні веб-сервіси, прогнозування заторів, планування маршрутів, система прийняття рішень, прогнозування часу проходження, оптимізація розрахунку часу проходження, Bing Maps API, .Net.

ABSTRACT **for diploma work**

Subject: «System for planning and optimization of transport routes in the uncertain conditions »

Student: Kulahin Oleksii Yuriyovych
Supervisor: Ph.D., Associate Professor Kozlov Oleksii Valeriyovych

The bachelor's thesis is devoted to the design, development, software implementation and implementation of the system of optimization and planning of transport routes in conditions of uncertainty.

The object of research is the processes of route optimization in conditions of uncertainty.

The subject of research - technologies and software for planning and optimization of transport routes in conditions of uncertainty.

The purpose is to increase the efficiency of estimating travel time and route planning by developing a software application using application software interfaces for cartographic services.

The bachelor's qualification work consists of a professional part and a special part on labor protection. The explanatory note of the thesis consists of an introduction, three chapters, conclusions and appendices.

In the first section the analysis of modern cartographic services is carried out. The second section discusses the technologies and tools for developing a system of optimization, planning and decision making. The third section describes the design and software implementation of the system of optimization and planning of transport routes in conditions of uncertainty.

As a result of the work performed, conclusions were made on the possibility and importance of optimizing the process of estimating the time of the planned route and the subsequent decision-making process on the best time of departure for a particular route.

Pages - 105, tables - 13, figures - 42, links - 28, appendices - 3.

Keywords: cartographic web services, road-traffic forecasting, route planning, decision making system, travel time forecasting, travel time optimizing, Bing Maps API, .Net.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ІСНУЮЧИХ КАРТОГРАФІЧНИХ СЕРВІСІВ ПЛАНУВАННЯ МАРШРУТІВ.....	7
1.1 Огляд існуючих картографічних веб-сервісів.....	7
1.2 Використання існуючих веб-сервісів для побудови маршруту з прогнозуванням заторів у майбутньому та їх проблематика	11
1.2.1 Маршрут з прогнозуванням заторів з використанням Google Maps ...	13
1.2.2 Маршрут з прогнозуванням заторів з використанням Bing Maps	15
1.3 Постановка задачі.....	19
Висновки до розділу 1	20
2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА МЕТОДІВ ОПТИМІЗАЦІЇ ПРОГНОЗУВАННЯ.....	21
2.1 Отримання даних про затори та їх вплив на прогнозування.....	21
2.2 Метод дискретизації маршруту з перерахунком часу.....	26
2.3 Система прийняття рішень для планування дати і часу відправлення за нечітких умов.....	29
2.3.1 Матриця рішень.....	30
2.3.2 Оцінювальна функція	31
2.3.3 Поле корисності рішень.....	33
2.4 Використання прикладних програмних інтерфейсів (API)	36
2.5 Мова програмування C#.....	39
2.6 Вибір інтегрованого середовища розробки.....	40
Висновки до розділу 2	42
3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ МАРШРУТІВ	43
3.1 Структура програмного застосунку	43
3.2 Прийняття рішення для кращого варіанту	45
3.3 Побудова та відправлення запиту до сервісу Bing Maps API.....	57
3.4 Побудова та зчитування відповіді від сервісу Bing Maps API	61
3.5 Оптимізація часу проходження маршруту	65
3.6 Обмеження на використання застосунку.....	68
3.7 Збір статистики та аналіз результату роботи	71
3.8 Керівництво користувача	72
Висновки до розділу 3	75
4 СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ.....	78
4.1 Аналіз умов праці.....	79
4.2 Техніка безпеки	85
4.3 Гігієна праці і виробнича санітарія	86
4.4 Пожежна безпека.....	88
Висновки до розділу 4	89
ВИСНОВКИ.....	90

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	92
ДОДАТОК А Лістинг код нормалізації оцінок до різних умов	94
ДОДАТОК Б Лістинг код оптимізації розрахунку часу проходження транспортного маршруту.....	96
ДОДАТОК В Покриття сервісу Bing Maps API.....	98

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ГН – гігієнічні норми

КНО – коефіцієнт природної освітленості

ОС – операційна система

API – Application Programming Interface, прикладний програмний інтерфейс, у нашому випадку для взаємодії з картографічними сервісами

GIS – Geographic information system, геоінформаційна система

GML – Geographic Markup Language

GMT – Greenwich Mean Time, час за Грінвічем

GPS – Global Positioning System, система глобального позиціонування

UTC – Exact Time Now or Coordinated Universal Time, всесвітньо координований час

XML – Extensible Markup Language, розширювана мова розмітки

Пояснювальна записка

до кваліфікаційної роботи

на тему:

СИСТЕМА ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ МАРШРУТІВ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810315

Виконав студент 4-го курсу, групи 401

_____ *О. Ю. Кулагін*

«__» червня 2022 р.

Керівник: канд. тех. наук, доцент

_____ *О. В. Козлов*

«__» червня 2022 р.

Миколаїв – 2022

ВСТУП

Актуальність. Обізнаність суспільства у сучасних умовах завдячується поширеним через мережу Інтернет картографічним веб-сервісам, які дозволяють досліджувати, аналізувати та приймати рішення у численних сценаріях, пов'язаних з картами та транспортними маршрутами. Такі сервіси стали невід'ємною частиною сучасного світу в умовах глобалізації, їх поширення та застосування постійно зростає.

У світі існує багато картографічних веб-сервісів та окремо їх інструментів, які допомагають при вирішенні нетривіальних задач. Серед найбільш популярних та безкоштовних: Google Maps, Bing Maps, MapQuest, OpenStreetMap. Сервіси мають широкий набір інструментів для перегляду знімків карт, побудови маршрутів з різноманітними параметрами, прогнозування заторів тощо. Однак, в більшості сервісів відсутні системи прийняття рішень або планування транспортних маршрутів, також існують проблеми з підрахунками проходження визначених маршрутів. У цих умовах перспективним є створення системи, яка буде зосереджена на оптимізації та плануванні транспортних маршрутів. Така система допоможе проаналізувати варіанти відправлень за маршрутом, знайти найкращі та найгірші, покращити розрахунки часу проходження, що позитивно впливатиме на точність системи прийняття рішення.

Мета – підвищення ефективності оцінювання часу проходження та планування маршрутів шляхом розробки програмного застосунку з використанням прикладних програмних інтерфейсів для картографічних сервісів.

Для поставлення поставленої мети необхідно вирішити наступні **завдання дослідження:**

1. проаналізувати особливості сучасних картографічних веб-сервісів для аналізу трафіку у реальному часі або у майбутньому та побудови і планування маршрутів згідно цьому.

2. дослідити існуючі технології та методи вирішення задач з побудови маршрутів згідно з аналізом трафіку у реальному часі або у майбутньому.

3. реалізувати веб-застосунок для обирання найкращого часу відбуття для окремо взятого маршруту з урахуванням трафіку на дорогах у майбутньому.

Об’єкт дослідження – процеси оптимізації маршрутів в умовах невизначеності.

Предмет дослідження – технології та програмні засоби планування та оптимізації транспортних маршрутів в умовах невизначеності.

Методологічною основою дослідження є загальнонаукові та аналітичні методи, які дозволили вивчити предмет та об’єкт дослідження, дослідити напрями та шляхи оптимізації і планування транспортних маршрутів.

Практичне значення отриманих результатів полягає в тому, що використання розробленої системи дозволить підвищити якість прогнозування часу проходження та планування відправлень за маршрутами.

Структура дипломної роботи. Відповідно до мети, завдань і предмета дослідження, дипломна робота містить основну та спеціальну частини. Основна частина роботи складається із вступу, трьох розділів, висновку, списку використаних джерел та 3 додатків. Загальний обсяг роботи – 105 сторінок, із них основного тексту основної частини – 71 сторінок, спеціальної – 12 сторінок. Кількість використаних джерел – 28.

1 АНАЛІЗ ІСНУЮЧИХ КАРТОГРАФІЧНИХ СЕРВІСІВ ПЛАНУВАННЯ МАРШРУТІВ

1.1 Огляд існуючих картографічних веб-сервісів

Онлайн-картографування або веб-картографування є процесом використання карт, що представляються геоінформаційними системами (GIS) в Інтернеті[1]. Онлайн-карти одночасно подаються і споживаються, тому це вірно назвати веб-картографія та послугою, за допомогою якої споживачі можуть вибрати, що відобразатиметься на карті[5]. У більшості випадків технологія веб-картографії або Web Mapping намагається надати ці дані у форматі GML (Geographic Markup Language) відповідно до специфікацій OpenGIS Consortium. Такі сервіси засновані на логіці хмарних обчислень та може відобразатися у різних векторних форматах, наприклад, SVF (простий векторний формат), DWF (Веб-формат для малювання), VML (мова векторної розмітки), SVG (масштабована векторна графіка), тощо[6].

На онлайн мапах можна отримати маршрути, як їхати використовуючи громадський транспорт, ходити пішки, їздити на велосипеді або навіть на мотоциклі. Часто сервіси пропонують кілька маршрутів для досягнення кінцевого пункту. У такому випадку вираховується та підкреслюється кращий варіант в залежності від часу, який необхідно витратити на проходження маршруту[4]. У якості додаткового функціоналу часто можна побачити наступні особливості[3,2]:

- інциденти на дорогах: аварії, перекриття доріг через ремонт, тощо;
- врахування платних доріг;
- побудова маршрутів через різні типи доріг: магістральні або з їх униканням;
- уникання перетину кордону;
- уникання паромів.

На рис. 1.1 продемонстровано 3 варіанти маршрутів для довільних пунктів, які були знайдені за допомогою сервісу Google Maps. У якості демонстрації роботи особливого параметру було додано виключення магістральних доріг (шосе).

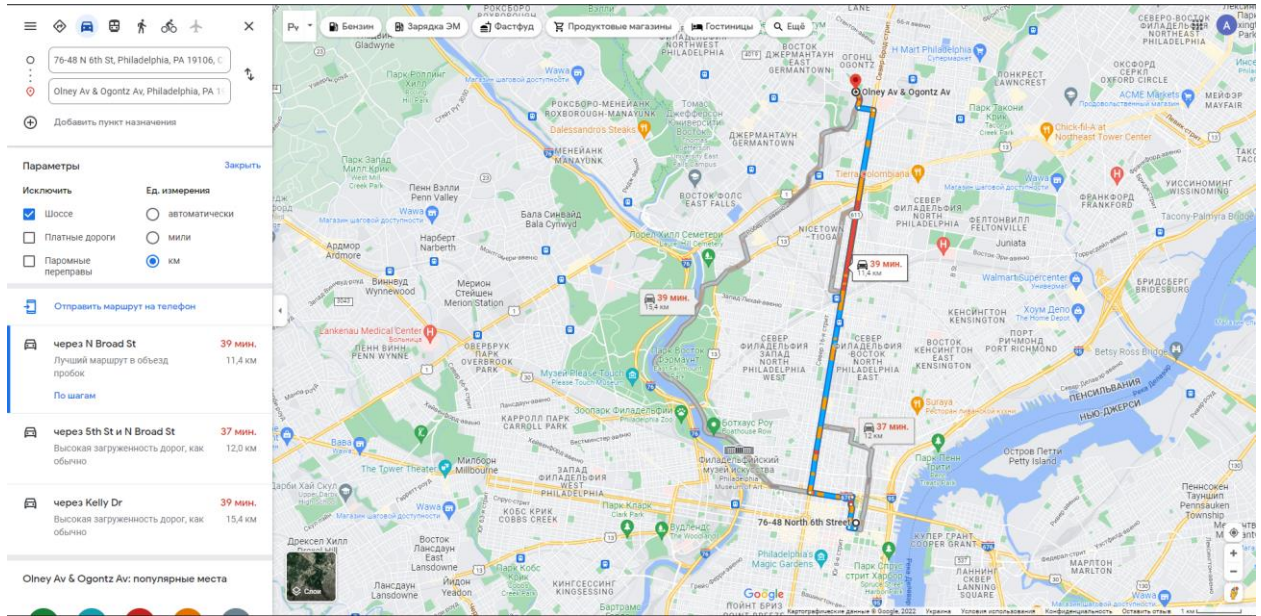


Рисунок 1.1 – Демонстрація знаходження маршрутів за допомогою сервісу Google Map

На разі існує багато веб-сервісів для пошуку кращого маршруту. Пошук маршруту не потребує багато накопиченої інформації, яка постійно оновлюється. За такими можна здійснювати навігацію з приблизним часом проходження. У наведеній табл. 1.2 приведено короткий огляд серед найбільш популярних та безкоштовних.

Таблиця 1.2 – Звітна таблиця з найбільш популярних картографічних веб-сервісів

Картографічний веб-сервіс	Посилання на сервіс	Короткий опис
Google Maps	https://www.google.com.ua/maps/	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів у майбутньому. – Є можливість переглянути стан заторів у майбутньому.[2]
Bing Maps	https://www.bing.com/maps	<ul style="list-style-type: none"> – Є можливість зробити маршрут з врахуванням заторів у майбутньому.[3]
MapQuest	https://www.mapquest.com/routepLANner	<ul style="list-style-type: none"> – Є можливість врахування витрат палива як додаткового критерію; – Немає можливості зробити маршрут з врахуванням заторів у майбутньому; – Немає можливості побудови маршруту за координатами.
HERE WeGo	https://wego.here.com	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів у майбутньому.
TomTom	https://www.tomtom.com/en_us/drive/maps-services/maps/	<ul style="list-style-type: none"> – Немає десктоп застосунку, тільки мобільні рішення.[4]
Maps.me	https://maps.me/	<ul style="list-style-type: none"> – Немає десктоп застосунку, тільки мобільні рішення.
Rand McNally	https://maps.randmcnally.com/	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів у майбутньому.
OpenStreetMap	https://www.openstreetmap.org/	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів у майбутньому.

Закінчення таблиці 1.2

Картографічний веб-сервіс	Посилання на сервіс	Короткий опис
Waze	https://www.waze.com/	<ul style="list-style-type: none"> – Немає можливості побудови маршруту за координатами; – Є можливість зробити спланувати маршрут з врахуванням заторів.
Citymapper	https://citymapper.com/webapp	<ul style="list-style-type: none"> – Немає можливості побудови маршруту за координатами. – Немає можливості зробити маршрут з врахуванням заторів у майбутньому.
Speedyroute	https://www.speedyroute.com/	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів; – Немає можливості побудови маршруту за координатами.
Rac Route Planner	https://www.rac.co.uk/route-planner/	<ul style="list-style-type: none"> – Немає можливості зробити маршрут з врахуванням заторів; – Є можливість врахування витрат палива як додаткового критерію.

Більшість сервісів покриває потреби користувача. Але якщо звернутися саме до функціоналу прогнозування заторів і, тим паче, до впливу заторів на побудову маршруту у майбутньому, то такі особливості є дуже рідкими через високі технічні вимоги до сервісу.

Якщо розглянути саме функціонал обирання найкращого маршруту у майбутньому з врахуванням динамічно змінюваних заторів, то таких функціонал у безкоштовних сервісах відсутній або необхідно власноруч вираховувати найкращий альтернативний варіант.

Серед проаналізованих я зупинився на сервісах Google Maps та Bing Maps як найбільш популярних та перспективних для поглибленого вивчення.

1.2 Використання існуючих веб-сервісів для побудови маршруту з прогнозуванням заторів у майбутньому та їх проблематика

Для того, щоб користувачу отримати дані про запланований маршрут веб-сервісу необхідно мати такі дані та інструменти:

1. Визначені пункти маршруту. Пункти маршруту визначаються користувачем і вводяться у вигляді адреси або координат.

2. Визначені параметри пошуку маршруту (вид транспорту, особливості доріг, тощо.). Параметри пошуку визначаються користувачем за особистим бажанням.

3. Дата і час або день тижня і час відправлення за визначеними пунктами маршруту. Параметри часу визначаються користувачем за особистим бажанням. За замовчуванням встановлено теперішній час, що дозволяє додати до розрахунків маршрутів поточні інциденти на дорогах.

4. Механізми побудови найкоротшого маршруту або маршрутів. Кожна система для пошуку маршрутів використовує методології для класичної TSP (задачі комівояжера) у реальних умовах. Сенс якої полягає у знаходженні найвигіднішого за визначеним критерієм маршруту, що проходить через вказані місця хоча б один раз, у такому випадку розв'язок знаходиться серед гамільтонових циклів. У нашому випадку єдиним критерієм є час.

5. Статистика щодо заторів на побудованому/них маршрутах. Дані про стан заторів в залежності від часу та дня тижня збираються сервісом за допомогою краудсорсингу.

6. Механізми аналізу та розрахунку впливу заторів на час проходження побудованих маршрутів. Зібрана статистика про затори також дає інформацію про середню швидкість проходження окремого відрізка з залежності від часу та дня тижня. Ці дані акумулюються та використовуються для прогнозування часу проходження з огляду на середню швидкість.

7. Система прийняття рішення щодо кращого маршруту за критерієм часу. Система обирає кращий варіант, хоча зазвичай також додає ще кілька альтернативних варіантів з трохи гіршим показником часу. Однак за бажанням користувача вони теж можуть бути обрані як підходящі.

Звертаю вашу увагу, 28 лютого 2022 року веб сервіси тимчасово перестали показувати стан заторів у реальному часі в Україні. У зв'язку з цим, приклади рішення задач та огляду праце спроможності застосунків чи окремих їх функцій будуть відтворюватися на територіях інших країн, де наразі ці сервіси працюють в повному обсязі як того вимагає поставлена вище задача.

Для прикладів будуть використовуватися умови зображені у табл. 1.3.

Таблиця 1.3 – Умови пошуку маршрутів

Змінна	Значення
Вид транспорту	автомобіль
Види доріг	будь-якого рівня
Дата і час відправлення	Приклад 1: Понеділок 11 липня 2022, 7:00 (GMT-4) Приклад 2: Понеділок 11 липня 2022, 8:45 (GMT-4)
Точки маршруту	39.951916, -75.150118 (Independence Hall, 520 Chestnut St, Philadelphia, PA 19106, Соединенные Штаты) 40.875408, -73.990208 40.745702, -73.847184 (Queens Museum, Flushing Meadows Corona Park, Building, Queens, NY 11368, Соединенные Штаты)

Такі дата і час відправлення обрані навмисно з такими цілями:

1. Понеділок обраний через те, що це найбільш навантажений день тижня і буде найкраще побачити зміни між часом пік та звичайним навантаженням.

2. Час 7:00 – це час, коли дороги не сильно навантажені.
3. Час 8:45 – це час, коли дороги сильно навантажені, час пік. Окрім того, це час, коли маршрут з планованим відправленням о 7:00 вже пройшов би більшу частину.

1.2.1 Маршрут з прогнозуванням заторів з використанням Google Maps

Розглянемо пошук маршрутів з урахуванням заторів у майбутньому від Google Maps за наступним посиланням <https://www.google.com.ua/maps/> (див. рис. 1.4). Цей сервіс не може врахувати вплив заторів у майбутньому на запланований маршрут, але може показати стан заторів. Тому ці дані користувачу доведеться розраховувати власноруч.

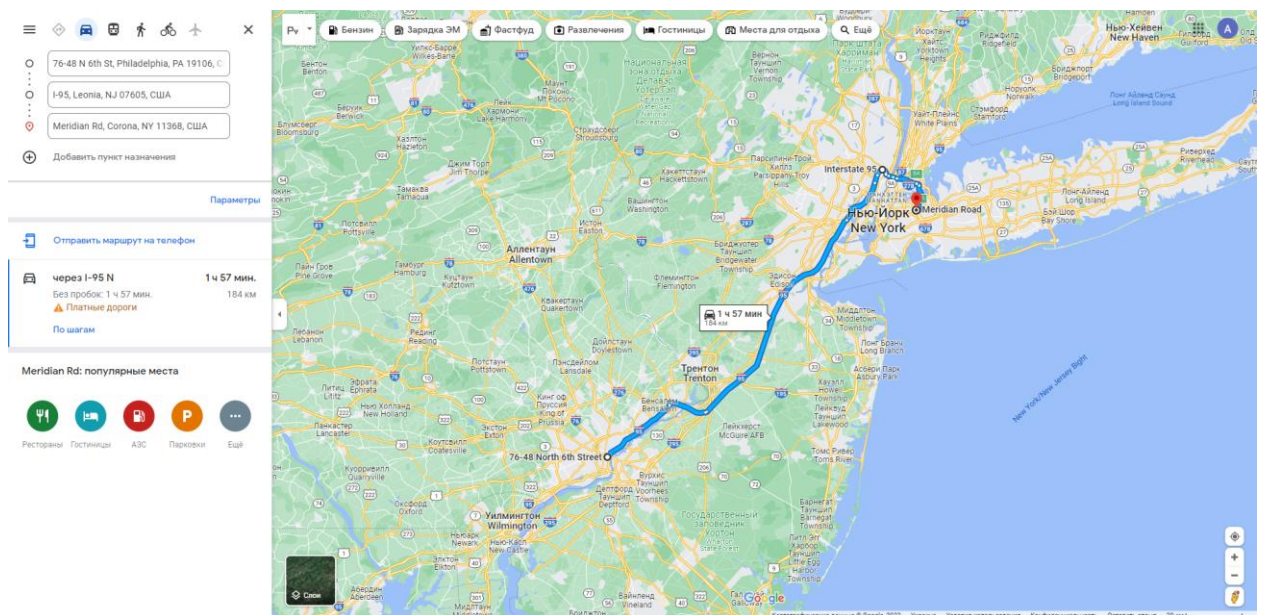


Рисунок 1.4 – Демонстрація пошуку маршруту із заданими умовами з використанням Google Maps

Результатами пошуку є наступні дані:

1. Варіант 1.
 - a. Відстань: 184 км;
 - b. Час: 1:57 годин.

Окремо були отримані дані заторів у майбутньому. Отримати маршрут неможливо. Для цих даних були використані наступні умови:

Приклад 1 (див. рис. 1.5).

1. Прогноз пробок: увімкнено,
2. День тижня: понеділок,
3. Час: 7:00 (GMT-4)

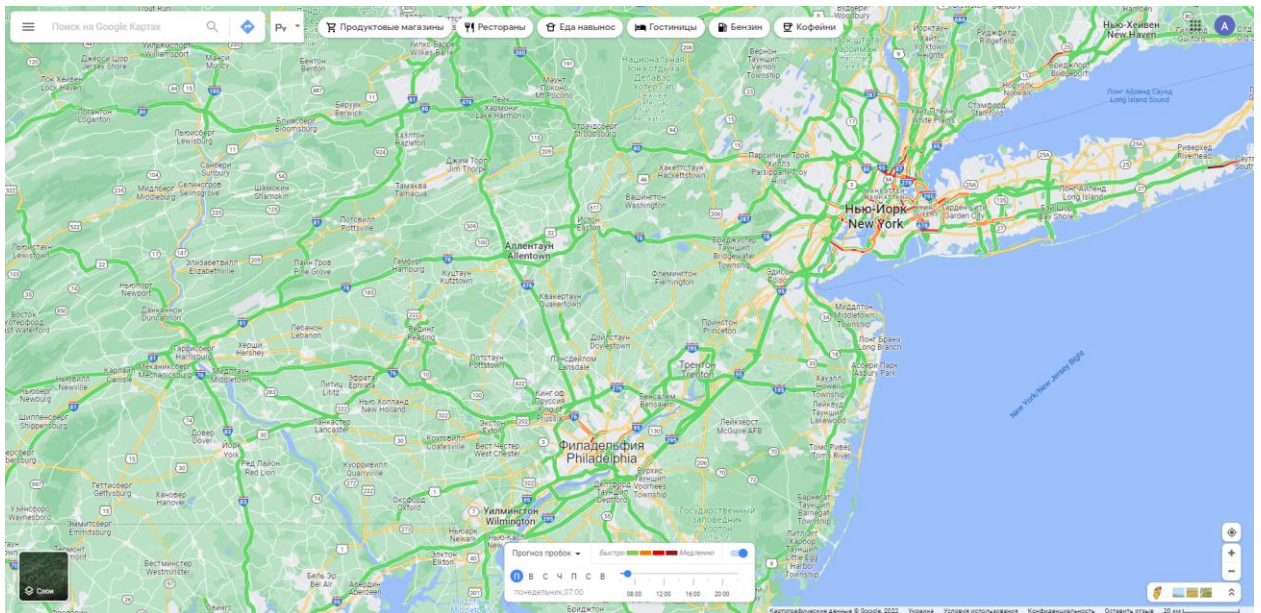


Рисунок 1.5 – Демонстрація стану заторів на визначеній площі за попередніми умовами за допомогою Google Maps

Приклад 2 (див. рис. 1.6).

1. Прогноз пробок: увімкнено,
2. День тижня: понеділок,
3. Час: 9:00 (GMT-4)

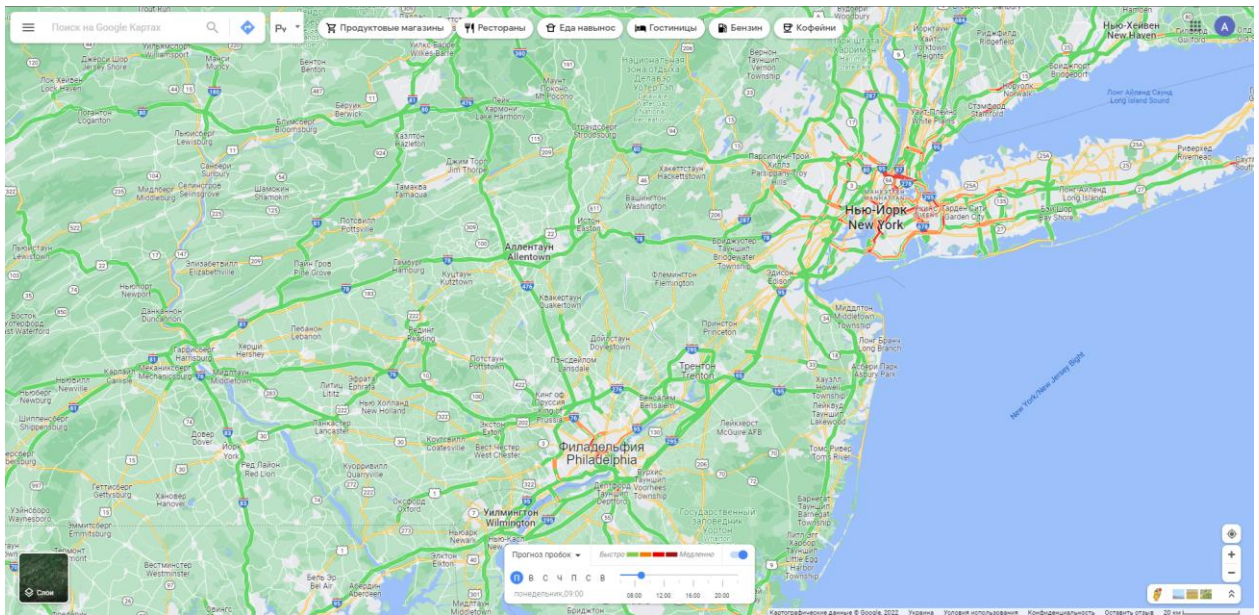


Рисунок 1.6 – Демонстрація стану заторів на визначеній площі за попередніми умовами за допомогою Google Maps

1.2.2 Маршрут з прогнозуванням заторів з використанням Bing Maps

Розглянемо пошук маршрутів з урахуванням заторів у майбутньому від Bing Maps за наступним посиланням <https://www.bing.com/maps?cc=ru>.

Bing Maps має наразі кращу систему побудови маршрутів з урахуванням заторів у майбутньому. Присутні 3 варіанти побудови маршруту:

- Відправитись зараз. Присутнє у всіх інших сервісах, де можна побудувати маршрут;
- Відправитись о конкретну дату і час;
- Прибути о конкретну дату і час.

Відправлення і прибуття у конкретний час є дуже сильною стороною Bing Maps. Мінімальний інтервал часу становить 5 хвилин. Також можна обрати у календарі зручну дату на заміні вибору дня тижня від Google Maps (див. рис. 1.7).

Маршрути ✕

📍 🚗 🚶 Параметры ▾

⊞ A Из

⊞ B До ↕

⊕ Добавить пункт назначения

🕒 ⬆ **Готово**

Иди сейчас **Залиште о** Приїджайте

08 45 ▾

← **липень 2022** →

П	В	С	Ч	П	С	Н
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Рисунок 1.7 – Демонстрація меню налаштування параметрів для побудови маршруту

Розглянемо, як цей сервіс справляється з прогнозуванням. Приклад 1 (див. рис. 1.8).

1. Дата і час: Понеділок 11 липня 2022, 7:00 (GMT-4)

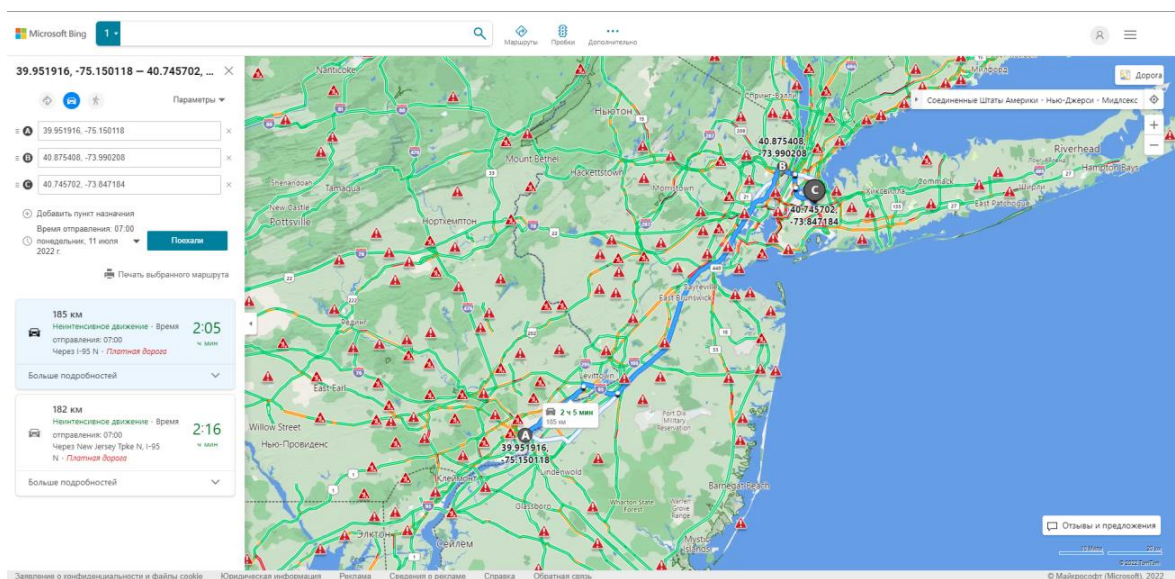


Рисунок 1.8 – Демонстрація пошуку маршруту із заданими умовами з використанням Bing Maps

Результатами пошуку є наступні дані:

1. Варіант 1.
 - с. Відстань: 185 км;
 - д. Час: 2:05 годин.
2. Варіант 2.
 - а. Відстань 182 км;
 - б. Час: 2:16 годин.

Приклад 2 (див. рис. 1.9).

1. Дата і час: Понеділок 11 липня 2022, 8:45 (GMT-4)

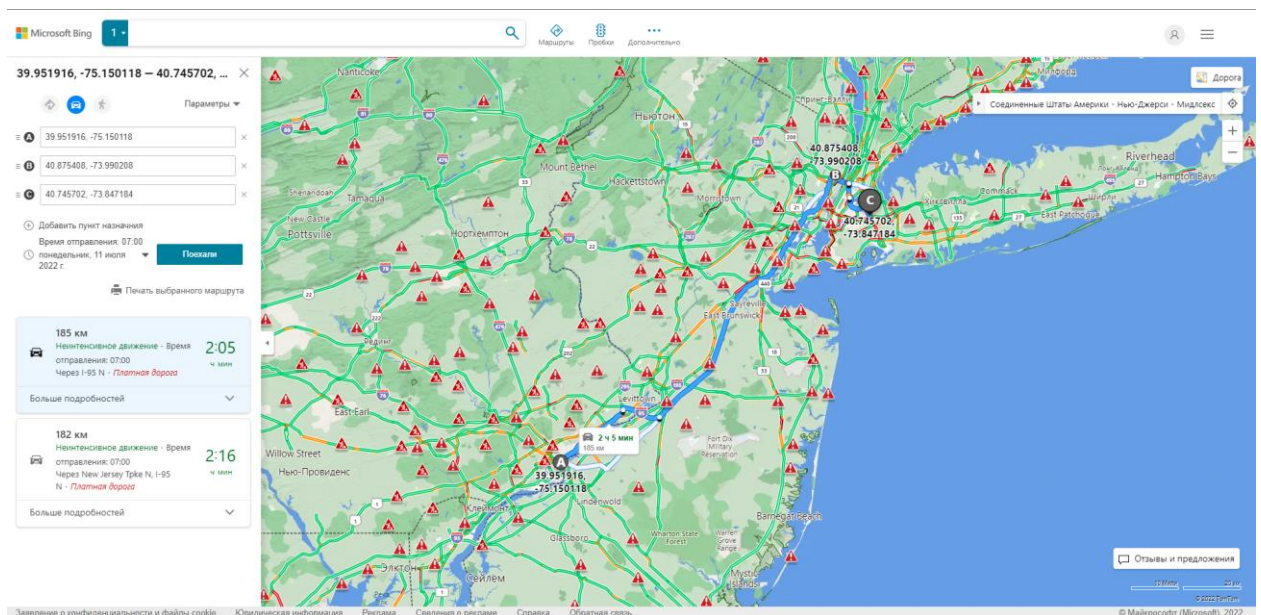


Рисунок 1.9 – Демонстрація пошук маршруту із заданими умовами з використанням Bing Maps

Результатами пошуку є наступні дані:

1. Варіант 1.
 - а. Відстань: 185 км;
 - б. Час: 2:10 годин.

2. Варіант 2.
 - а. Відстань 182 км;
 - б. Час: 2:23 годин.

Дізнавшись різницю у часі ми отримуємо та значення:

1. Варіант 1: 5 хвилин затримки.
2. Варіант 2: 7 хвилин затримки.

Сервіс від Bing Maps значно краще справився із задачею, але є істотні проблеми.

Проблематика визначення маршрутів із прогнозом заторів полягає у наступному: картографічний сервіс не враховує зміни стану заторів та їх впливу на загальний час під час проходження самого маршруту. Якщо ми розглянемо приклад 1 з відправленням о 7:00, то він рахує час згідно заторам на саме цей час. Але вже через 1:45 годин стан заторів зміється настільки сильно, що можна отримати видиму похибку у часі, яку сервіс динамічно не обраховує. Згідно з цим планування відправлення буде мати суттєву похибку у часі. В умовах, коли присутні такі відхилення, прийняти рішення серед багатьох варіантів, коли краще зробити відправлення, є проблемою.

Для того, щоб впевнитись у тому, що дійсно веб-сервіс не робить перерахунків з часом розглянемо приклад з умовами зазначеними вище. На рис. 1.10 з лівої сторони продемонстровано час відправлення о 7:00, з правої – о 8:30. Якщо ви обрали час відправлення о 7:00, то у проміжковій точці В ви будете через 1:30 годин, тобто дані зображені зліва для вас будуть нерелевантні і треба буде використовувати дані з правої частини. Можна побачити, що деякі кроки замість «помірних заторів» почали показувати «серйозні затори». У результаті цього ви ризикуєте втратити 8 хвилин тільки на цьому відрізку або 7% від загального часу. Те саме можна продемонструвати і у зворотному сценарії, коли через деякий час заторів вже не буде і ви навпаки зекономили б час.

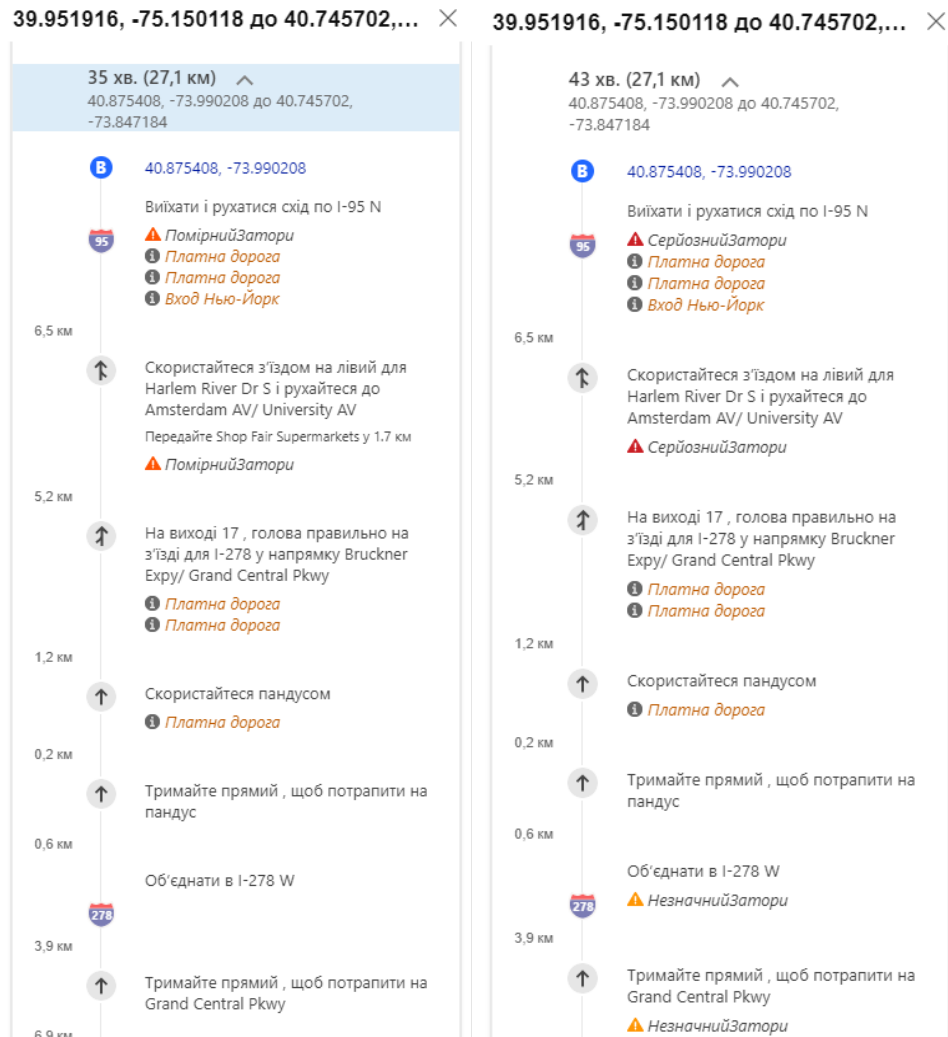


Рисунок 1.10 – Порівняння частини маршруту при різному часі відправлення

1.3 Постановка задачі

Мета: покращення процесу оцінювання часу проходження маршруту та планування часу відправлень за маршрутом шляхом розробки методів оптимізації та розробки і реалізації програмного веб-застосунку з використанням прикладних програмних інтерфейсів для картографічних сервісів для оптимізації обрахунків та системи прийняття рішень.

Для поставлення поставленої мети необхідно вирішити **наступні завдання:**

1. проаналізувати сучасний стан картографічних веб-сервісів для аналізу трафіку у реальному часі або у майбутньому та побудови і планування маршрутів згідно цьому.

2. дослідити існуючі технології та методи вирішення задачі з побудови маршрутів згідно з аналізом трафіку у реальному часі або у майбутньому.

3. реалізувати веб-застосунок для обирання найкращого часу відбуття для окремо взятого маршруту з урахуванням трафіку на дорогах у майбутньому.

Об'єкт дослідження – оптимізація розрахунку часу проходження маршруту та планування часу відправлень за маршрутом з динамічно змінюваним станом заторів.

Предмет дослідження – використання картографічних веб-сервісів та прикладних програмних інтерфейсів для оптимізації розрахунку часу проходження маршруту та планування часу відправлень за маршрутом.

Висновки до розділу 1

Сучасні сервіси мають досить потужні системи та дата центри для обрахування маршрутів. Але при цьому далеко не всі можуть будувати маршрути з прогнозуванням заторів. А навіть ті, що присутні, мають істотну проблему із динамічним обрахуванням часу під час проходження самого маршруту. Згідно з продемонстрованим функціоналом прогнозована похибка може сягати 6-8% для часу проходження маршруту, а у деяких випадках досягати 14-15%, що є досить значним для сервісів світового рівня, якими користуються десятки і сотні мільйонів людей.

Також серед представлених сервісів відсутня можливість у зручному форматі проаналізувати запропонованні альтернативні варіанти та прийняти рішення. Така функція, скоріш за все, наявна тільки у платних застосунках.

2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА МЕТОДІВ ОПТИМІЗАЦІЇ ПРОГНОЗУВАННЯ

2.1 Отримання даних про затори та їх вплив на прогнозування

У минулому єдиний спосіб дізнатися про те, як рухається рух на певній дорозі, — це налаштуватися на щогодинні радіомовлення. Однак завдяки рівню витонченості, якого ми досягли сьогодні, ми можемо перевіряти статус у режимі реального часу за допомогою картографічних сервісів. Багато телефонних програм і програм покладаються на дані Google, щоб надати точну інформацію про ваше географічне розташування.

Розглянемо як саме це відбувається на прикладі сервісів від Google. Усі інші сервіси мають практично такі самі методи отримання даних.

До 2009 року Google збирав дані з датчиків дорожнього руху та камер на дорогах, які в основному встановлювали державні транспортні агентства та деякі приватні компанії, які збирали дані про трафік для різних цілей. Ці датчики дорожнього руху використовували лазерний радар або активну інфрачервону технологію, яка могла визначити, наскільки швидко рухається загальний трафік, спостерігаючи за загальними розмірами та швидкістю автомобілів (рис. 2.1). Потім ця інформація передавалася на сервери, і тоді можна було оголошувати регулярні оновлення трафіку. Google також отримав дані з цих джерел і вставив їх у свій додаток Карти, щоб інформувати користувачів про дорожні умови.



Рисунок 2.1 – Демонстрація система зондування та моніторингу, встановлена на точці виявлення для спостереження за рухом транспортних засобів і збору даних[7]

На жаль, ця техніка мала ряд недоліків. По-перше, ці датчики встановлювали лише на важливих вулицях, тобто на дорогах, які були найбільш схильні до руху транспорту. Тому, якби ви хотіли пройти альтернативним, мало використаним маршрутом до пункту призначення, але на ньому виникла несподівана пробка, ви б не знали про це заздалегідь. Крім того, за допомогою датчиків дорожнього руху ви ніколи не зможете миттєво отримувати оновлення про поточний рух на певній дорозі на своєму смартфоні або іншому портативному пристрої.

У 2009 році Google перейшов на краудсорсинг – складну, швидшу і набагато надійнішу техніку отримання даних про трафік в реальному часі (рис. 2.2). Велика кількість людей анонімно передає дані на сервер, який потім надсилає корисну, корисну інформацію назад своїм учасникам після певного аналізу та аналізу даних.

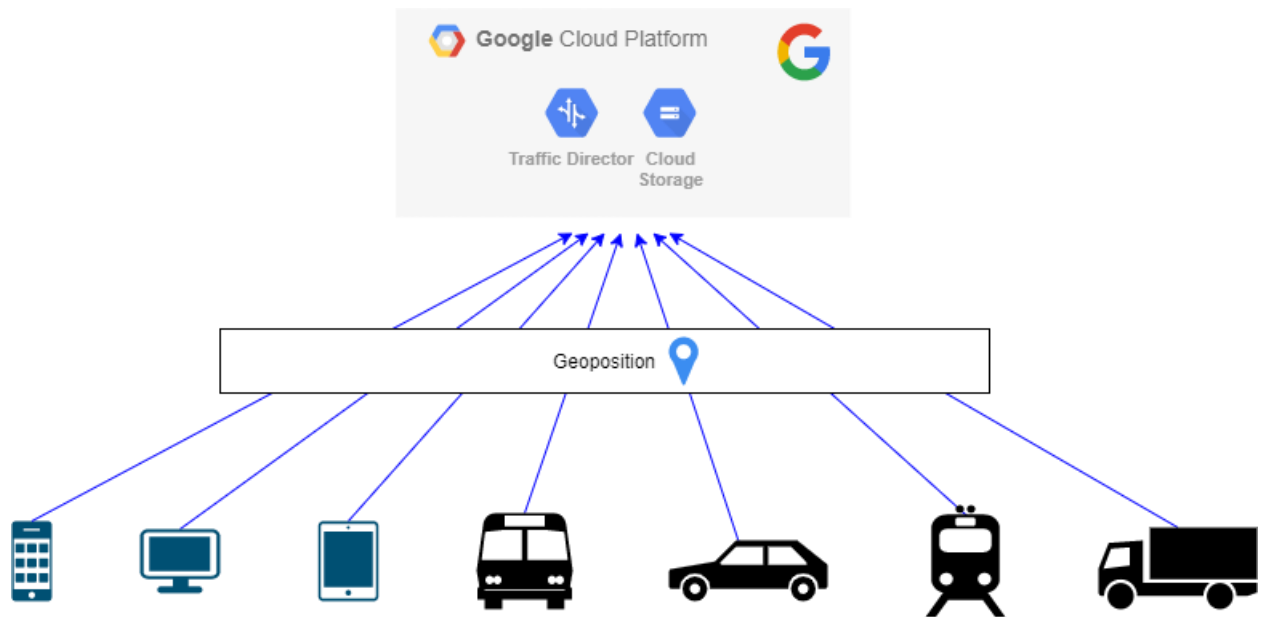


Рисунок 2.2 – Схема відправлення з будь-яких девайсів з геопозицією до серверів Google

Краудсорсинг – це модель пошуку інформації для збору інформації. Припустимо, що ви хочете надавати користувачам оновлення в режимі реального часу про умови дорожнього руху. Спочатку ви розробляєте додаток, за допомогою якого користувачі можуть отримати маршрути до місць, наприклад GPS. Однак, щоб користувач міг отримати інформацію про поточну ситуацію на дорогах, він також повинен (анонімно) поділитися своїм власним географічним розташуванням із програмою. Окрім цього одного користувача, є багато інших користувачів, які хочуть оновлювати трафік на тому самому маршруті. Таким чином, усі вони повинні буде поділитися своїм географічним розташуванням із програмою, щоб отримати доступ до умов дорожнього руху на певній дорозі (рис. 2.3, 2.4).

Таким чином, ви (як розробник програми) отримаєте купу інформації про конкретне географічне розташування, включаючи кількість активних користувачів у цій області, швидкість різних автомобілів, де зараз використовується ваша програма (через супутники GPS), «щільність» автомобілів і ряд інших точок даних. Використовуючи всю цю інформацію, ви б надавали оновлення трафіку в режимі реального часу всім користувачам

вашого додатка, спочатку отримавши від них інформацію. Ось як працює краудсорсинг – ви отримуєте інформацію від користувачів свого додатка, створюєте сукупність даних, робите швидкий аналіз, а потім передаєте значущу інформацію, яку можна зробити, назад своїм користувачам.

Саме так Google розповідає вам про умови дорожнього руху у вашому районі. Як тільки ви увімкнете «Моє місцезнаходження» на Картах Google, ви автоматично почнете надсилати дані про своє місцезнаходження на сервери Google.

Під час подорожі по завантаженій дорозі ви можете перебувати поруч із іншими мандрівниками, які також увімкнули своє місцезнаходження в додатку і, отже, автоматично діляться інформацією про затори з Google. Коли Google бачить, що в певній місцевості є кілька повільно рухаються транспортних засобів (або багато повільно рухаються смартфонів із картами), він позначає це жовтою лінією на Картах. Подібним чином, якщо сталася перешкода або відхилення, що призвело до належної пробки, це позначає цю перешкоду червоною лінією на Картах.

У такому випадку ми спостерігаємо ситуацію, коли навіть частина девайсів відправляє дані про позицію, але цієї інформації вистачає, щоб її проаналізувати та відправляти вже усім користувачам. Якість отримуваної інформації завжди можна збільшити шляхом збільшення відсотку девайсів з увімкненою геопозицією.

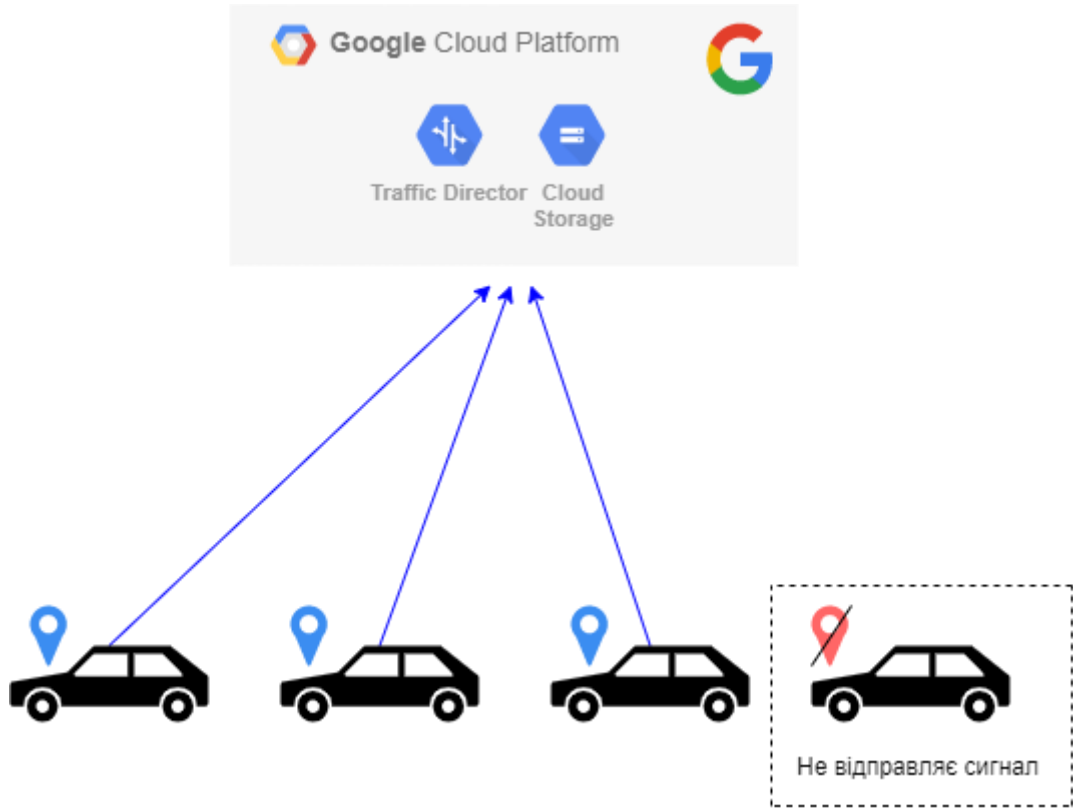


Рисунок 2.3 – Схема принципу отримання даних від девайсів



Рисунок 2.4 – Схема принципу відправлення даних до девайсів

Ви завжди можете відмовитися від системи, не повідомляючи Google про своє місцезнаходження. Ваша конфіденційність може бути захищена.

Важливо зазначити, що Google стверджує, що передача місцезнаходження вашого смартфона повністю анонімна, тому Google не знає точного джерела вхідної інформації. Це надзвичайно важливо, оскільки в результаті створюється надійна, безпечна екосистема, де користувачі можуть ділитися інформацією для спільного блага, не порушуючи власну конфіденційність.

2.2 Метод дискретизації маршруту з перерахунком часу

Для обрахунку більш точного загального часу проходження буде використовуватися метод дискретизації. Метод дискретизації маршруту для знаходження часу проходження з урахуванням змін заторів полягає у наступному:

1. Розбиття маршруту на відрізки по 5-15 км або 5-15 км;
2. Перерахунок часу проходження кожного відрізка з урахуванням вже витраченого часу.

Принциповий алгоритм роботи такого методу продемонстровано на блок-схемі (рис. 2.5).

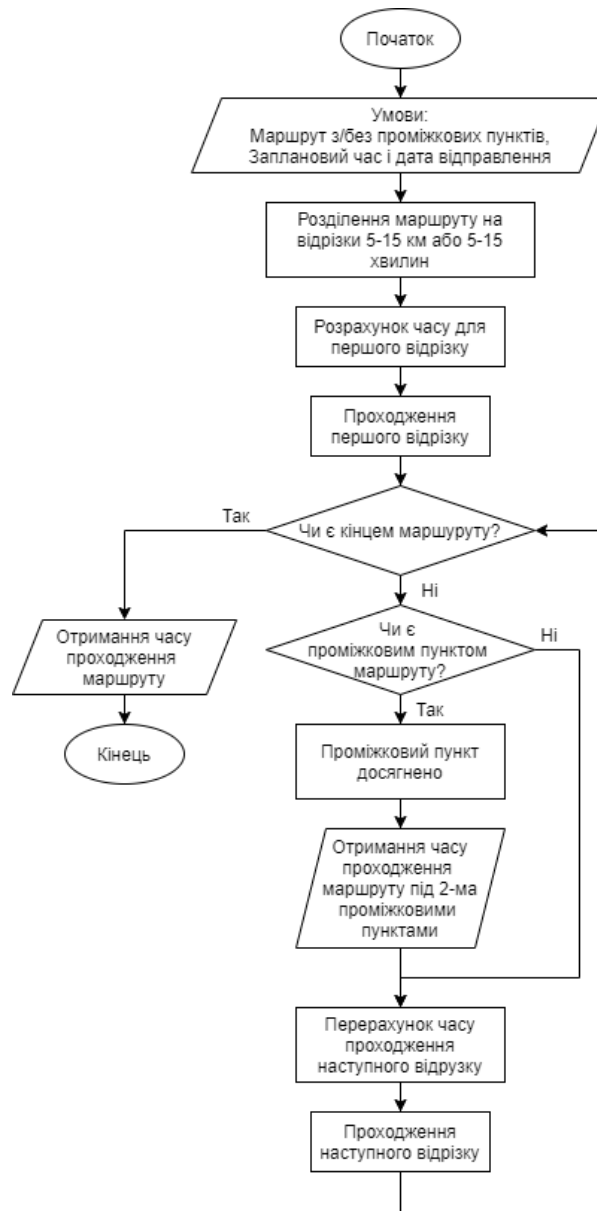


Рисунок 2.5 – Блок-схема методу дискретизації

Розглянемо використання методу дискретизації у грубому вигляді на наступному прикладі (рис. 2.6).

Нехай є існуючий маршрут АВ з однією проміжковою точкою N_1 . $AN_1 = 80$ км з часом проходження 48 хв. $N_1B = 100$ км з часом 75 хв. Усього 180 км з часом 137 хв. Тоді при часу відбуття о 7:00 ми будемо мати час прибуття 9:13. Затори у такому випадку вираховуються станом на 7:00 без подальших змін під час проходження.

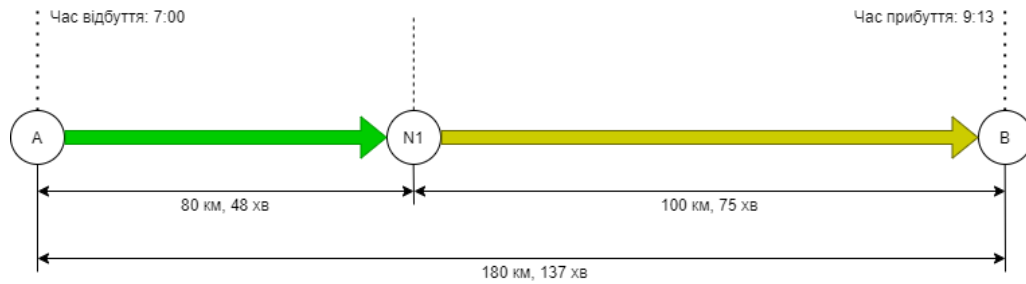


Рисунок 2.6 – Демонстрація отримання часу проходження за замовченням без перерахунку

Розв'яжемо задачу наступним чином (рис. 2.7):

1. Дискретизація маршруту. Нехай ліміт для дискретизації буде більше 80 км, тоді відрізок AN_1 залишається як є, а N_1B ділимо навпіл. Отримуємо 2 відрізки N_1N_2 та N_2B по 50 км.
2. Розраховуємо час для відрізка AN_1 він залишається таким самим, 48 хв.
3. Розраховуємо час для відрізка N_1N_2 з часом відбуття 7:48 за допомогою сервісу і отримуємо 30 хв.
4. Розраховуємо час для відрізка N_2B з часом відбуття 8:18 за допомогою сервісу і отримуємо 55 хв.
5. Сумуємо час за всі відрізки і отримуємо, що маршрут AB буде пройдемо за $AN_1 + N_1N_2 + N_2B = 48 + 30 + 55 = 147$ хвилин.

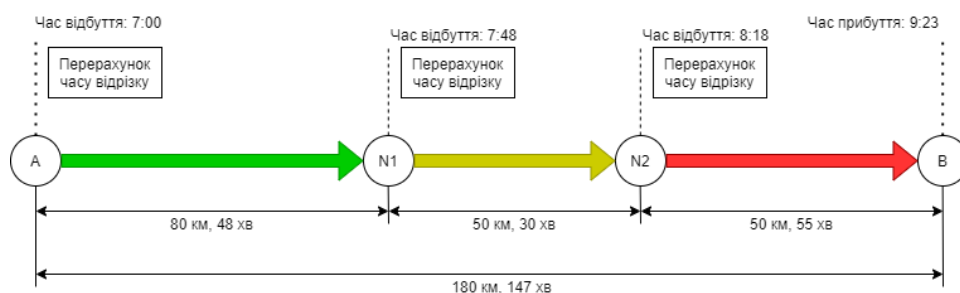


Рисунок 2.7 – Демонстрація отримання часу проходження з дискретизацією з перерахунком

2.3 Система прийняття рішень для планування дати і часу відправлення за нечітких умов

Прийняття рішення кращого часу являє собою процес вибору одного найкращого з точки зору відповідного критерію часу рішення з деякої множини всіх альтернативних варіантів: $E_i \in E$.

У нашому випадку, існує кінцеве число альтернативних варіантів часу відправлення: $E_1, E_2, \dots, E_i, \dots, E_m$. За необхідності можна розширити кількість варіантів шляхом збільшення загального часового проміжку, серед якого необхідно обрати кращий, або зменшення інтервалу, на який буде ділитися загальний часовий проміжок.

Кожному варіанту рішення E_i відповідає деякий результат або оцінка e_i . Цьому результату має відповідати кількісна оцінка.

Оскільки нас цікавить кращий варіант або з найбільшим значенням оцінки, то метою вибору є знаходження найкращого рішення, для якого оцінка є максимальною

$$\max_i e_i$$

Де $e_i (i = 1, \dots, m)$ – оцінки, що характеризують корисність.

Таким чином, вибір оптимального варіанта здійснюється за допомогою критерію у формулі 2.1:

$$E_0 = \{E_{i_0} \mid E_{i_0} \in E \cap e_{i_0} = \max_i e_i\}, \quad (2.1)$$

де множина E_i оптимальних варіантів складається з таких варіантів E_{ii} , які належать множині E всіх альтернативних варіантів і оцінка e_{ii} яких є максимальною серед оцінок e_i (\cap - операція “І”).

Слід додати, що вибір може бути неоднозначний, якщо максимально отримана оцінка буде відповідати одночасно кільком рішенням, але необхідність вибору одного з них зазвичай не викликає проблем.

2.3.1 Матриця рішень

З'ясувавши, що дані, які можна отримати у відомих картографічних веб-сервісах мають суттєву похибку постає питання: коли тоді краще робити відправлення з розрахунку, що під час проходження маршруту стан заторів змінюється?

Необхідно прийняти рішення відносно того, який час відправлення є найкращим для того, щоб витратити найменший час на маршрут та як найближча дата і час відправлення.

Альтернативні варіанти рішень є такими:

E_1 – вибір часу відправлення з боку найшвидшого і як найскорішого;

E_m – вибір часу відправлення з боку найповільнішого і найбільш відкладеного;

E_i – проміжні рішення, ($i = 1, \dots, m$).

Зовнішні умови:

F_1 – умови, що забезпечують найшвидше і як найскоріше відправлення;

F_n – умови, що забезпечують найповільніше і найбільш відкладене відправлення;

F_j – проміжні умови, ($j = 1, \dots, n$).

Результат (оцінка) рішення e_{ij} ($i = 1, \dots, m; j = 1, \dots, n$) – це оцінка, яка відповідає варіанту E_i ($i = 1, \dots, m$) та зовнішній умові F_j , ($j = 1, \dots, n$) і характеризує економічний ефект або корисність.

Множину всіх альтернативних рішень $E = \{E_i, (i = 1, \dots, m)\}$ можна описати відповідною матрицею рішень $\|e_{ij}\|, (i = 1, \dots, m; j = 1, \dots, n)$. Сформуємо відповідну матрицю рішень $\|e_{ij}\|, (i = 1, \dots, m; j = 1, \dots, n)$, яка представлена в таблиці 2.8.

Таблиця 2.8 – Матрицю рішень $\|e_{ij}\|, (i = 1, \dots, m; j = 1, \dots, n)$

E/F	F_1	F_2	F_3	...	F_j	...	F_n
E_1	e_{11}	e_{12}	e_{13}	...	e_{1j}	...	e_{1n}
E_2	e_{21}	e_{22}	e_{23}	...	e_{2j}	...	e_{2n}
...
E_i	e_{i1}	e_{i2}	e_{i3}	...	e_{ij}	...	e_{in}
...
E_m	e_{m1}	e_{m2}	e_{m3}	...	e_{mj}	...	e_{mn}

2.3.2 Оцінювальна функція

Для знаходження однозначного (найбільш вигідного) рішення E_{i0} з множини всіх альтернативних варіантів E доцільно ввести необхідні оцінювальні функції e_{ir} , які б для кожного з альтернативних варіантів рішень E_i дозволяли сформулювати його узагальнену корисність з врахуванням всіх наслідків $e_{ij}, (j = 1, \dots, n)$ для відповідного альтернативного варіанта рішення E_i при всіх можливих зовнішніх станах $F_j, (j = 1, \dots, n)$, тобто з врахуванням кількісних значень всіх елементів e_{ij} , відповідного i -го рядка матриці рішень $\|e_{ij}\|$.

Після обчислення значень оцінювальних функцій $e_{ir}, (i = 1, \dots, m)$ для всіх альтернативних варіантів рішень матриця рішень $\|e_{ij}\|, (i = 1, \dots, m; j = 1, \dots, n)$ зводиться матриці з одним стовпцем $e_{ir}, (i = 1, \dots, m)$, де кожному альтернативному варіанту E_i приписується деякий результат (узагальнена корисність рішення) e_{ir} , що в цілому характеризує всі наслідки цього альтернативного рішення E_i (Таблиця 2.9).

Таблиця 2.9 – Матриця рішень $\|e_{ij}\|, (i = 1, \dots, m)$

Альтернативний варіант рішення	Оцінювальна функція
E_1	e_{1r}
E_2	e_{2r}
E_3	e_{3r}
...	...
E_i	e_{ir}
...	...
E_m	e_{mr}

Проблема формування матриці рішень $\|e_{ij}\|, (i = 1, \dots, m)$ з використанням узагальнених значень корисності рішень полягає у виборі типу оцінювальної функції або в тому, який зміст надати оцінювальній функції e_{ir} при розв'язанні кожної конкретної задачі прийняття рішень.

Розглянемо деякі підходи до формування оцінювальних функцій для наведеного вище прикладу.

Оптимістична позиція – позиція, коли робиться ставка на те, що буде мати місце найкращий випадок

$$e_{ir} = \max_j e_{ij}, (j = 1, \dots, n) \quad (2.2)$$

Песимістична позиція – позиція, коли робиться ставка на найгірший випадок.

$$e_{ir} = \min_j e_{ij} \quad (2.3)$$

Для комбінації найбільшого і найменшого результатів можна прийняти оцінювальну функцію e_{ir} у вигляді

$$e_{ir} = \min_j e_{ij} + \max_j e_{ij} \quad (2.4)$$

Позиція нейтралітету, при якій вважається, що всі відхилення результату рішення від “середнього” випадку є допустимими:

$$e_{ir} = \frac{1}{n} \sum_{j=1}^n e_{ij} \quad (2.5)$$

Позиція відносного песимізму. Для кожного варіанта альтернативних рішень оцінюються втрати в порівнянні з визначеним для кожного зовнішнього стану найкращим результатом (пошук по стовпцям матриці рішень). В подальшому визначають найгірший залишок (пошук по рядкам матриці залишків) для кожного варіанта альтернативних рішень

$$e_{ir} = \max_j (\max_i e_{ij} - e_{ij}) \quad (2.6)$$

2.3.3 Поле корисності рішень

Розглянемо на наступному прикладі: є два зовнішні стани $\{F_1, F_2\}$, $n = 2$ і 24 варіантів рішень $\{E_1, E_2, \dots, E_{24}\}$. F_1 – умова (оцінка) часу відправлення, F_2 – умова (оцінка) часу проходження маршруту (табл. 2.10).

Таблиця 2.10 – Матриця рішень $\|e_{ij}\|$, ($i = 1, \dots, 24; j = 1, 2$) при $n = 2$

	Значення E_m (час)	Приклад 1		Приклад 2		Приклад 3	
		F_1	F_2	F_1	F_2	F_1	F_2
E_1	0:00	0	0	0	0	12	0
E_2	1:00	1	0	1	0	13	0
E_3	2:00	2	0	2	0	14	0
E_4	3:00	3	0	3	0	15	0
E_5	4:00	4	0	4	0	16	0
E_6	5:00	5	0	5	1	17	1
E_7	6:00	6	0	6	3	6	3
E_8	7:00	7	0	7	5	7	5
E_9	8:00	8	0	8	10	8	10
E_{10}	9:00	9	0	9	20	9	20
E_{11}	10:00	10	0	10	19	10	19

Закінчення таблиці 2.10

E_{12}	11:00	11	0	11	17	11	17
E_{13}	12:00	12	0	12	12	12	12
E_{14}	13:00	13	0	13	13	13	13
E_{15}	14:00	14	0	14	11	14	11
E_{16}	15:00	15	0	15	12	15	12
E_{17}	16:00	16	0	16	18	16	18
E_{18}	17:00	17	0	17	16	17	16
E_{19}	18:00	18	0	18	13	18	13
E_{20}	19:00	19	0	19	11	19	11
E_{21}	20:00	20	0	20	10	20	10
E_{22}	21:00	21	0	21	6	21	6
E_{23}	22:00	22	0	22	3	22	3
E_{24}	23:00	23	0	23	1	23	1

Поле корисності рішень для заданої в таблиці 2.10 матриці рішень для прикладу 1 побудовано на рисунках 2.11, 2.12, 2.13.

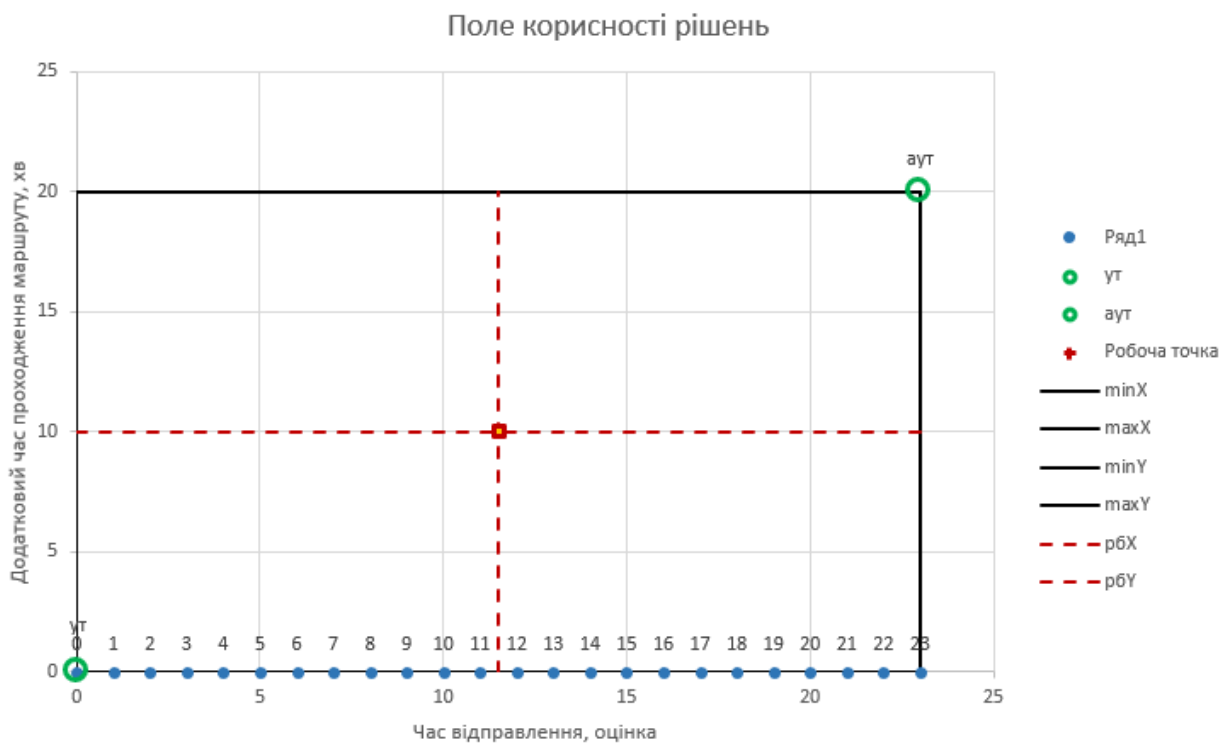


Рисунок 2.11 – Демонстрація поля корисності рішень для прикладу 1

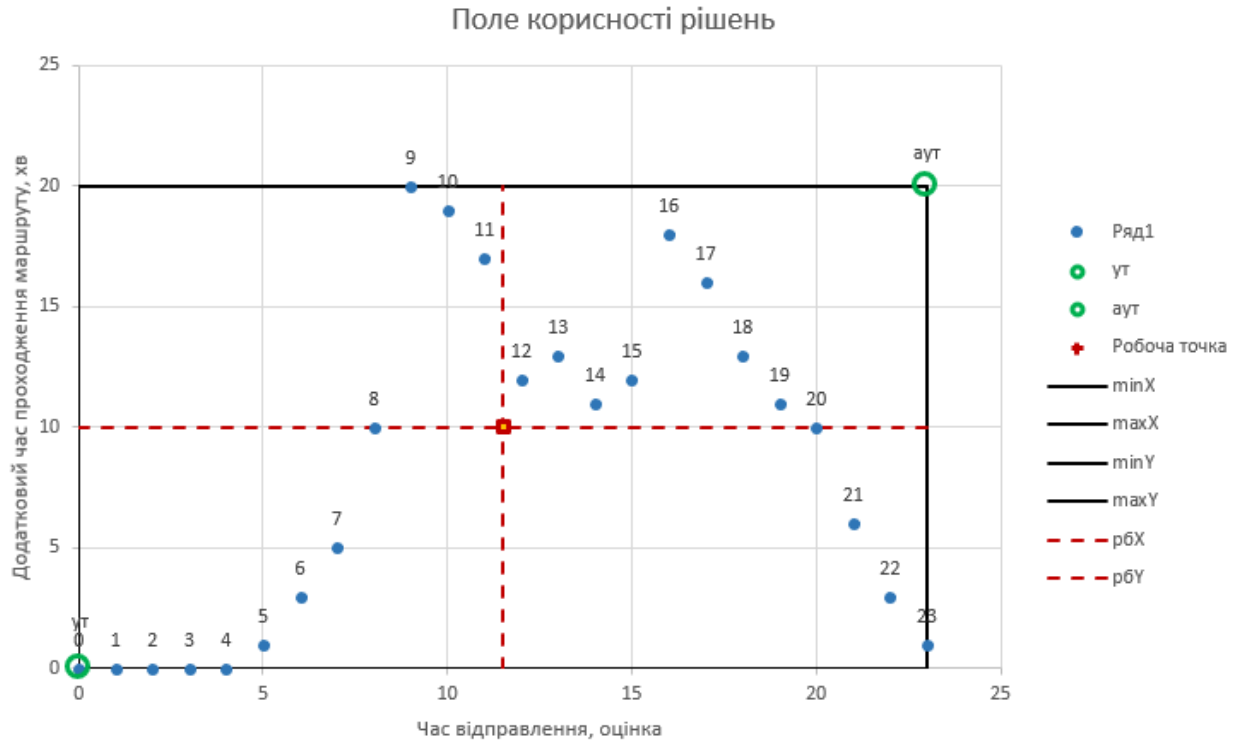


Рисунок 2.12 – Демонстрація поля корисності рішень для прикладу 2

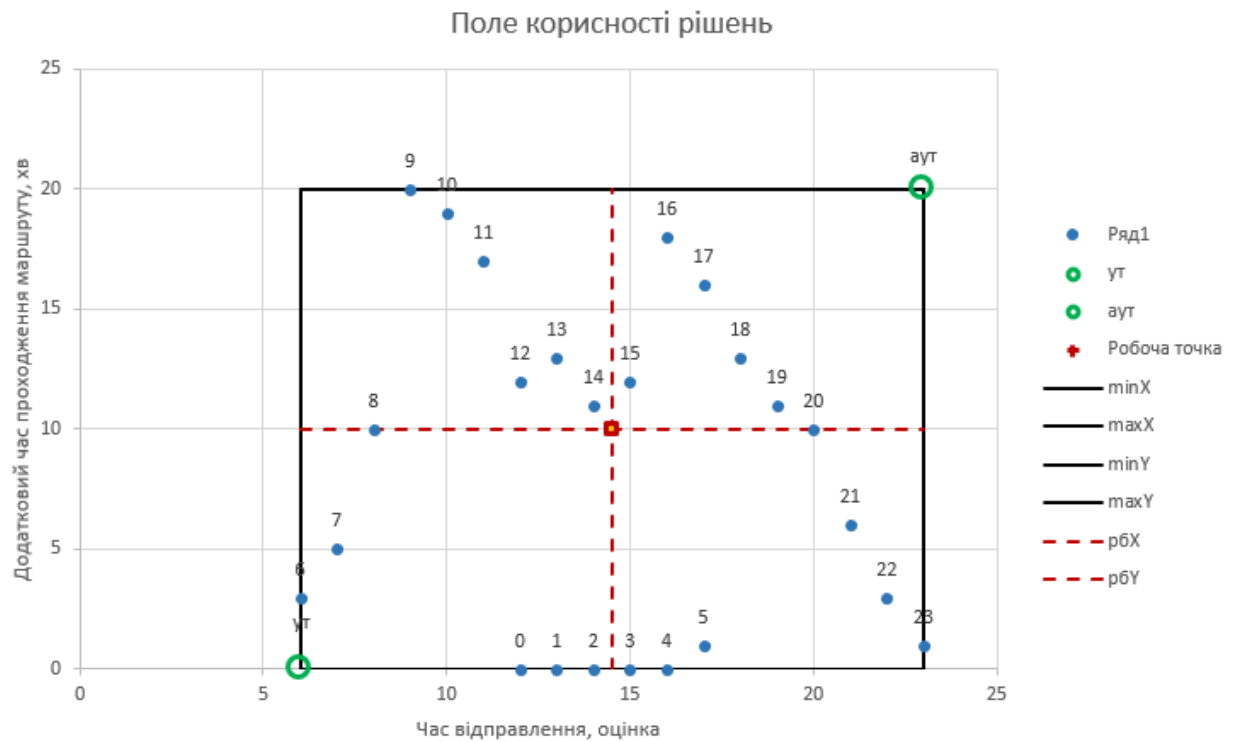


Рисунок 2.13 – Демонстрація поля корисності рішень для прикладу 3

2.4 Використання прикладних програмних інтерфейсів (API)

Для того, щоб обрахувати дані та прийняти рішення необхідно мати інструменти для отримання цих даних. Саме для цього використовуються картографічні прикладні програмні інтерфейси (API) або веб-сервіси, які дають дані за відповідним запитом. Серед існуючих сервісів слід обрати саме той, який буде задовольняти усі вимоги до нього від програмного застосунку.

Вимоги до сервісів:

- можливість побудови маршруту між двома або більше пунктами;
- можливість розрахунку часу проходження маршруту з врахуванням трафіку;
- можливість врахування важливих аспектів: перетин кордону, платні дороги, паромні переправи, пересування автомагістралями, тощо.
- можливість відправлення за маршрутом у визначений день та час;
- прогнозування заторів у майбутньому;
- можливість отримати графічне відображення побудованого маршруту на статичній або інтерактивній карті;
- відсутність оплати за користування або обмежене користування, яке буде покривати вимоги.
- наявність повної документації та зручність користування;
- якнайбільше покриття країн та регіонів світу;
- наявність існуючого прикладу реалізації роботи сервісу;

Серед усього списку вимог були відокремлені основні та за ними був проведений аналіз існуючих технологій. Сервіси були порівняні та обраний кращий у таблиці 2.14.

Таблиця 2.14 – Порівняльна таблиця прикладних програмних інтерфейсів

Назва	Врахування трафіку для часу проходження	Врахування побічних аспектів	Відправлення у визначений день і час	Прогнозування заторів	Безоплатне
Google Maps API	Так	Паромні переправи, інциденти, аварії	Так, дні тижня	Так, але без врахування часу проходження	Так, але є ліміт запитів
Bing Maps API	Так	Паромні переправи, платні дорого, автомагістралі, перетин кордону, інциденти	Так	Так	Так, але є ліміт запитів
OpenLayers Direction Map API	Ні	Ні	Ні	Ні	Так
TomTom Direction Map API	Так	Інциденти, погода, стихійні лиха, аварії	Так	Так	Так, але є ліміт запитів
Mapbox Direction Map API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
HERE Direction Map API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
Mapfit Direction Map API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
Open Source Routing Machine	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
YOURS navigation API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
Open Route Service Library	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
Graphhopper Direction Map API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів

Закінчення таблиці 2.14

ESRI Direction Map API	Ні	Ні	Ні	Ні	Так, але є ліміт запитів
MapQuest Direction Map API	Так	Врахування палива, автомагістралі	Ні	Ні	Так, але є ліміт запитів
Open Street Maps: Traffic	Так	автомагістралі та інші типи доріг	Ні	Ні	Так, але є ліміт запитів
Quartz Maps	Ні	Ні	Ні	Ні	Так, але є ліміт запитів

Кожен проаналізований сервіс має свої сильні сторони, деякі зосереджені на відображенні карт, деякі на оптимізації маршрутів, на варіативності транспортних засобів, тощо. Але серед написаних вище умов лише частина зможе стати повноцінним інструментом. Це пов'язано з тим, що лише найбільші компанії можуть собі дозволити застосовувати технологію краудсорсингу для збирання та аналізу даних для заторів, які є ключовими для реалізації програмного застосунку. У цілому усі програмні інтерфейси мають базовий функціонал для знаходження маршрутів різними методами, відображення маршруту на карті, високе покриття країн та регіонів світу. Документація та обізнаність у мережі Інтернет у багатьох не є сильною стороною.

Серед розглянутих найкращими є 3 варіанти: Google Maps API, Bing Maps API та TomTom Direction Map API.

Google Maps API – чудовий варіант, який знають усі через однойменний картографічний сервіс. Серед мінусів слід зазначити відсутність впливу заторів на проходження маршруту у майбутньому. Також виникли труднощі з отриманням доступу, а саме ключа для сервісу, через що вимушено є відкладений.

TomTom Direction Map API – варіант, який один з небагатьох вмів аналізувати затори, але не має реалізації у вигляді картографічного сервісу, що

є суттєвим для ознайомлення. У цілому є альтернативою до Bing Maps API. Більше того, сам Bing Maps API частково заснований на цьому прикладному програмному інтерфейсі.

Bing Maps API – кращий варіант, який має все необхідне для програмної реалізації, присутня повна документація, це як і завжди для Microsoft. Присутній однойменний картографічний сервіс, який є досить популярним і є одним із конкурентів Google Maps.

Також було досліджено, що ці кандидати на мають динамічного перераховування часу проходження маршруту з врахуванням зміни стану заторів, тому усіх можна оптимізувати.

2.5 Мова програмування C#

Для програмної реалізації була обрана мова програмування C#. C# має велике розмаїття прикладних програмних інтерфейсів та синтаксичних конструкцій.

Велика надійність цієї мови програмування була досягнута через роботу CLR машини, адже на відміну від інших компілятор CLR запускає розроблене додаток на віртуальному процесорі. Тому у випадках виникнення будь-яких помилок, це ніяк не вплине на роботу інших програм в системі, але це також означає, що для запуску програми необхідно додатковий час. Відповідно програми написані на мові програмування C# надійніші, але менш швидкі (ніж ті ж програми написані на C++). Така надійність допомагає компенсувати не ідеальність програмної реалізації[4].

Однією з головних причин вибору мови C# є прикладний програмний інтерфейс Windows Forms (Рис.2.1). Цей інтерфейс програмування відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework[23]. Як і Abstract Window Toolkit (AWT) (схожий API для мови Java), бібліотека Windows Forms була розроблена як частина .NET Framework для спрощення розробки компонентів графічного інтерфейсу користувача.

Windows Forms побудована на основі застарілого Windows API. Windows Forms надає доступ до власних елементів керування користувальницьким інтерфейсом Windows, обертаючи існуючий API Windows у керований код. За допомогою Windows Forms .NET Framework забезпечує більш повну абстракцію над API Win32, ніж Visual Basic або MFC.

Переваги C #:

- У загальномовній середовищі виконання (CLR) збирач сміття (GC) виконує функції автоматичного диспетчера пам'яті. Збирач сміття керує розподілом та вивільненням пам'яті для програми, що дозволяє компенсувати неідеальність програмної реалізації та не перейматися виділенням пам'яті[21].
- C # - це об'єктно-орієнтована, простий і в той же час потужна мова програмування, який дозволяє розробникам створювати багатофункціональні програми.
- C # ставитися до мов компільованого типу, тому він має всі переваги таких мов.
- C # об'єднує кращі ідеї сучасних мов програмування Java, C++, Visual Basic, тощо.
- Через великого розмаїття синтаксичних конструкцій і можливості працювати з платформою .Net, C# дозволяє швидше, ніж будь-який інший мову, розробляти програмні рішення.

2.6 Вибір інтегрованого середовища розробки

Оскільки за мову програмування був обраний C#, доцільніше за все використовувати зручне інтегроване середовище розробки додатків Microsoft Visual Studio .NET[3].

Переваги використання Visual Studio:

- Менше писати, але отримувати більше результату. Працюючи з її засобами для розробки VS додає багато речей завдяки інтелектуальній системі.

- Швидке читання коду. Інтелектуальна система середовища розробки вміє підлаштовувати код під необхідний формат і синтаксис. Таким чином, він стає більш зручним для читання і доступним для редагування.
- CodeLens. Утиліта, яка дозволяє знаходити помилки в коді. По суті, це коректор, який вбудований прямо в Visual Studio.
- Хвиляста лінія. Функція для визначення помилок. При наведенні і натисканні можна побачити подробиці помилки і способи її усунення (рис.2.15).
- Дебагер або відладчик. Він допомагає детально вивчити код і знайти в ньому помилки. Без подібного кошти писати великі мобільні або веб-додатки неможливо. За його допомогою з легкістю можна дізнатися про використання ресурсів системи (рис. 2.16).

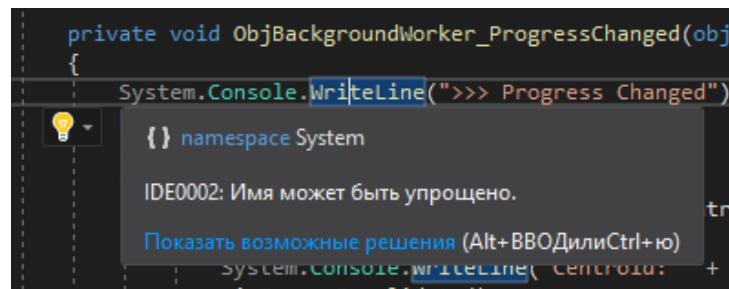


Рис.2.15. – Демонстрація пропозиції Visual Studio змінити код

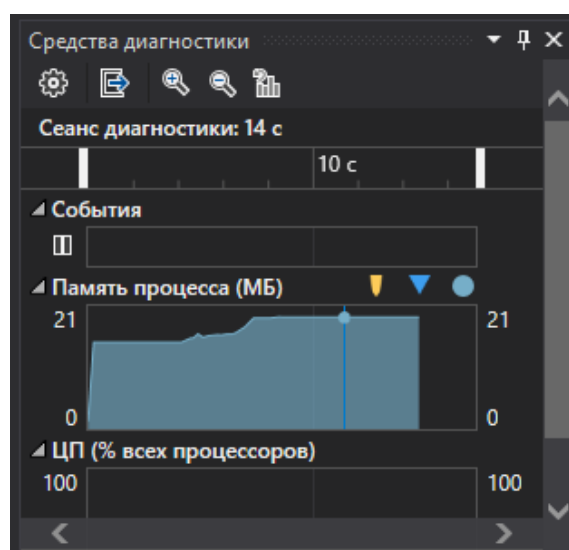


Рис.2.16. – Демонстрація діагностики роботи застосунку

Висновки до розділу 2

Існуючі картографічні веб-сервіси використовують сучасні технології, а саме краудсорсингу, для отримання та обробки великих обсягів даних користувачів з усієї планети для прогнозування заторів. Такі технології вимагають потужних серверів, тому лише одиниці з всесвітньо відомих сервісів мають їх мають. Також сервіси мають просту систему прийняття рішення, коли присутнє тільки одне краще рішення, та максимум одне альтернативне і така система обирає тільки між маршрутами, інших сутностей немає.

Розроблювана система буде мати на меті оптимізацію прогнозування часу проходження маршруту за допомогою простого методу дискретизації, який дозволить покращити значно покращити процес обрахування. Також буде розроблена система прийняття рішень за трьома критеріями.

Кожен проаналізований сервіс має свої сильні сторони, деякі зосереджені на відображенні карт, деякі на оптимізації маршрутів, на варіативності транспортних засобів, тощо. Для програмного застосунку був обраний прикладний програмний інтерфейс Bing Maps API, який має значні переваги у порівнянні з іншими популярними інтерфейсами у контексті поставлених до проекту завдань. Також була обрана мова C# з відповідним фреймворком .NET. Обрано середовище програмування Visual Studio, яке є найбільш зручним для досягнення поставлених цілей

3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ МАРШРУТІВ

3.1 Структура програмного застосунку

Проект програмного застосунку не вимагає використання бази даних. Увесь функціонал програми уміщується у відносно невеликому масштабі, де найголовнішу роль грають модулі, які працюють для обробки відповіді від АРІ, подальшого зчитування та прийняття рішення. Структура зображена на рис. 3.1. Як будь-який застосунок на базі Windows Forms присутні і інші допоміжні файли та бібліотеки від .Net за замовченням.

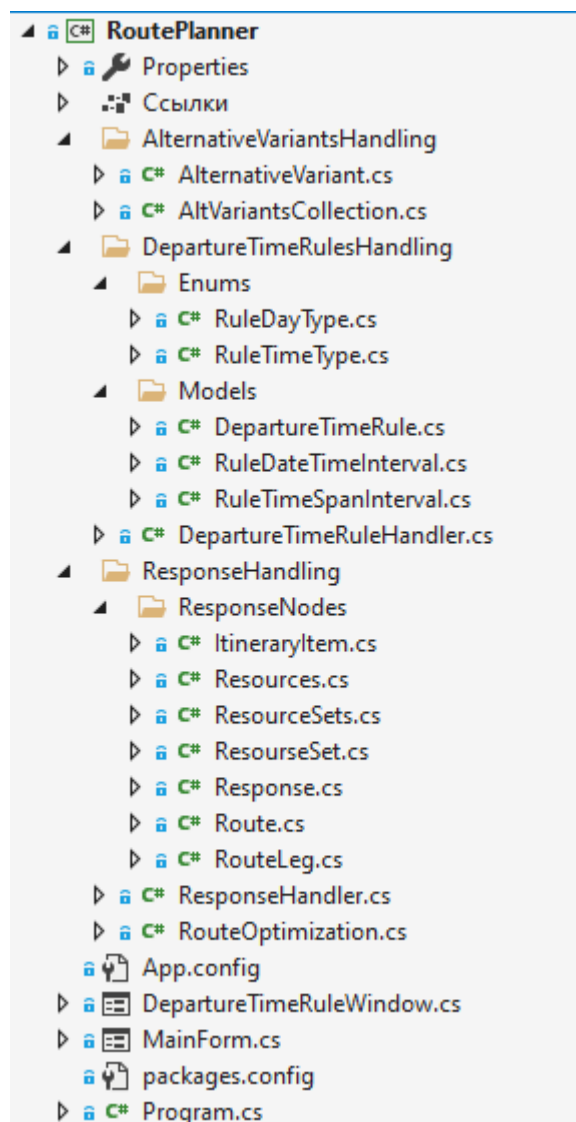


Рисунок 3.1 – Демонстрація структури проекту

До проекту додано систему керування версіями Git. Весь вихідний код та програму можна знайти у Github за посиланням: <https://github.com/Apache360/RoutePlanner>.

Також до проекту доданий Bing Maps REST Tool Kit. Головна мета цієї бібліотеки – надання набору інструментів для легко взаємодії з Bing Maps REST serviced у ПЗ, яке засноване на технології .Net. Нажаль, повноцінно використовувати функціонал цієї бібліотеки не вдалось через деякі її обмеження, але використовуються деякі класи, наприклад клас Coordinates.cs.

Загальний вигляд програми після її завантаження зображено на рисунку 3.2. Для вхідних даних відокремлена ліва панель з трьома сторінками та основною кнопкою запуску обрахування. Для виведення результату – центральна панель з трьома сторінками та текстове поле унизу для кращого варіанту. Для логуювання – права панель з текстовим вікном, часом витрачених на роботу програми та часом, скільки ще залишилось.

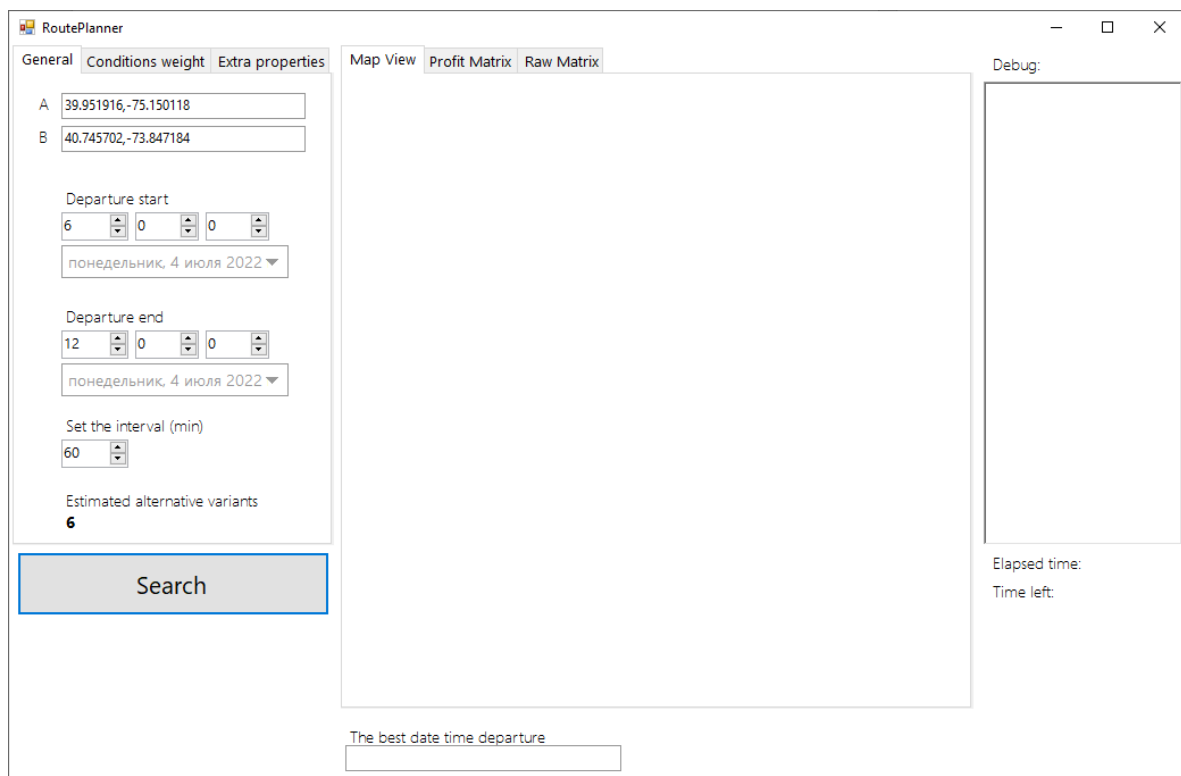


Рисунок 3.2 – Демонстрація загального вигляду програми

3.2 Прийняття рішення для кращого варіанту

Перед тим як прийняти рішення необхідно мати список альтернатив, які мають дату відправлення, відповідні оцінки до кожної з умов та загальну оцінку. Повну структуру для роботи з ними зображено на рис.3.3.

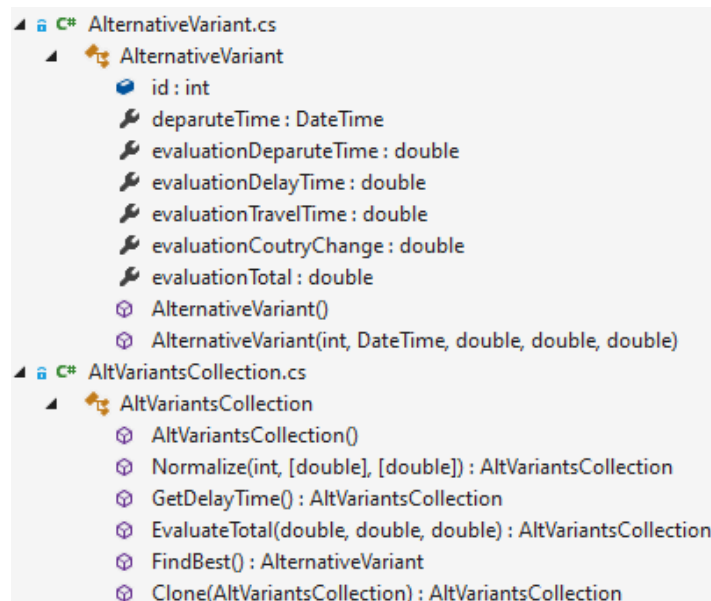


Рисунок 3.3 – Демонстрація структури класів для роботи з альтернативними варіантами

Для окремо взятої альтернативи створено клас `AlternativeVariant.cs`. Кожен альтернативний варіант зараз має 3 умови, за якими буде проводитись оцінка:

- `EvaluationDepartureTime` – оцінка часу відбуття.
- `EvaluationDelayTime` – оцінка затримки проходження маршруту.
- `EvaluationCountryChange` – оцінка перетину кордонів країн.

Потенційно можливо додати ще інші умови, наприклад, погодні умови, інциденти, платні дороги, паромні переправи, тощо, які теж будуть впливати на прийняття рішення.

Альтернативні варіанти записуються у відповідний список, для чого був створений клас `AltVariantsCollection.cs`. Також цей клас дозволяє маніпулювати даними у списку. Можна виділити такі головні методи:

1. `Normalize`: цей метод робить нормалізацію оцінок перед прийняттям рішення. Цей метод зрівнює умови один між одним у інтервалі від 0 до 100. Код методу зображено у додатку А.

2. `GetDelayTime`: знаходить різницю між поточним варіантом та кращим часом проходження серед існуючих, зазвичай це проходження вночі або найменш завантажених.

3. `EvaluateTotal`: підраховує загальну оцінку для кожного з варіантів з поточних оцінок кожної з умов та вагових коефіцієнтів. Принципово метод знаходить середнє арифметичне, яке є найбільш універсальним методом, але також доцільно використовувати інші критерії оцінювання, наприклад критерій добутків.

4. `FindBest`: знаходить альтернативний варіант з кращою загальною оцінкою. Є реалізацією оптимістичної позиції. Окрім того, ще можна додати знаходження найгіршого варіанту, знаходження груп кращих, нейтральних або толерантних та гірший варіантів.

Розглянемо реалізований інтерфейс для базового функціоналу (рис. 3.4). У лівій панелі на сторінці «General» присутні поля для вхідних даних для окремо взятого сценарію. Поле «А» для точки відправлення. Поле «В» для точки прибуття. Поля визначені після «Departure start» необхідні для визначення першого прийняттого часу і дня відправлення. Відповідно після «Departure end» – для останнього прийняттого часу і дня відправлення. Обов'язково ці значення мають бути у часі після початкового. Для визначення інтервалу між альтернативними варіантами є поле «Set the interval (min)». Додатково виведено кількість альтернативних варіантів для обрахунку. Рекомендована кількість варіантів – менше 100, через дуже довгу обробку, яка

може бути нестабільна. Також існує вимога, щоб був хоча б 1 альтернативний варіант.

General Conditions weight Extra properties

A 39.951916,-75.150118

B 40.745702,-73.847184

Departure start

6 0 0

понедельник, 4 июля 2022

Departure end

12 0 0

понедельник, 4 июля 2022

Set the interval (min)

60

Estimated alternative variants

6

Search

Рисунок 3.4 – Демонстрація сторінки з базовими вхідними даними

Після заповнення базових необхідних вхідних даних можна натиснути кнопку «Search» та зачекати. Результат роботи зображено на рисунках 3.5 та 3.6, де у центральній панелі на сторінці «Profit Matrix» зображено у вигляді таблиці матрицю прибутку, яка побудована згідно списку альтернативних варіантів.

Кафедра інтелектуальних інформаційних систем
Система планування та оптимізації транспортних маршрутів в умовах невизначеності

RoutePlanner

General Conditions weight Extra properties

A 39.951916,-75.150118
B 40.745702,-73.847184

Departure start
6 0 0
понедельник, 4 июля 2022

Departure end
18 0 0
понедельник, 4 июля 2022

Set the interval (min)
60

Estimated alternative variants
12

Search

Map View Profit Matrix Raw Matrix

Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
0	2022/07/04 6:00:00	20	38,19	0	19,4
1	2022/07/04 7:00:00	18,182	25,52	0	14,57
2	2022/07/04 8:00:00	16,364	70,05	0	28,8
3	2022/07/04 9:00:00	14,546	95,4	0	36,65
4	2022/07/04 10:00:00	12,728	100	0	37,58
5	2022/07/04 11:00:00	10,91	99,21	0	36,71
6	2022/07/04 12:00:00	9,09	95,32	0	34,8
7	2022/07/04 13:00:00	7,272	76,47	0	27,91
8	2022/07/04 14:00:00	5,454	58,64	0	21,36
9	2022/07/04 15:00:00	3,636	20,76	0	8,13
10	2022/07/04 16:00:00	1,818	0	0	0,61
11	2022/07/04 17:00:00	0	22,58	0	7,53

Debug:

02h:01m:16s
#8 Travel duration (optimized): 04/07/2022 14:00:00: 02h:04m:36s
#9 Travel duration (raw): 04/07/2022 15:00:00: 02h:06m:10s
#9 Travel duration (optimized): 04/07/2022 15:00:00: 02h:12m:34s
#10 Travel duration (raw): 04/07/2022 16:00:00: 02h:14m:58s
#10 Travel duration (optimized): 04/07/2022 16:00:00: 02h:16m:56s
#11 Travel duration (raw): 04/07/2022 17:00:00: 02h:20m:26s
#11 Travel duration (optimized): 04/07/2022 17:00:00: 02h:12m:11s

Elapsed time: 00:01:38
Time left: 00:00:00

The best date time departure
#4: Monday, July 4, 2022 10:00:00 AM

Рисунок 3.5 – Демонстрація результату роботи програми за базовим сценарієм, сценарій без перетинів кордону

RoutePlanner

General Conditions weight Extra properties

A 3835 Luna Pier Rd, Erie, Мічиган 48133,
B 19 18th St, Буффало, Нью-Йорк 14213,

Departure start
0 0 0
понедельник, 4 июля 2022

Departure end
6 0 0
понедельник, 4 июля 2022

Set the interval (min)
30

Estimated alternative variants
12

Search

Map View Profit Matrix Raw Matrix

Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
0	2022/07/04 0:00:00	80	80	0	53,33
1	2022/07/04 0:30:00	72,728	76,784	0	49,84
2	2022/07/04 1:00:00	65,456	75,576	0	47,01
3	2022/07/04 1:30:00	58,184	71,256	0	43,15
4	2022/07/04 2:00:00	50,912	69,752	0	40,22
5	2022/07/04 2:30:00	43,64	68,544	0	37,39
6	2022/07/04 3:00:00	36,36	29,848	100	55,4
7	2022/07/04 3:30:00	29,088	30,952	100	53,35
8	2022/07/04 4:00:00	21,816	19,896	100	47,24
9	2022/07/04 4:30:00	14,544	8,04	100	40,86
10	2022/07/04 5:00:00	7,272	0	100	35,76
11	2022/07/04 5:30:00	0	1,104	100	33,7

Debug:

04h:38m:54s
#8 Travel duration (optimized): 04/07/2022 4:00:00: 04h:38m:43s
#9 Travel duration (raw): 04/07/2022 4:30:00: 04h:39m:25s
#9 Travel duration (optimized): 04/07/2022 4:30:00: 04h:40m:41s
#10 Travel duration (raw): 04/07/2022 5:00:00: 04h:39m:29s
#10 Travel duration (optimized): 04/07/2022 5:00:00: 04h:42m:01s
#11 Travel duration (raw): 04/07/2022 5:30:00: 04h:39m:07s
#11 Travel duration (optimized): 04/07/2022 5:30:00: 04h:41m:50s

Elapsed time: 00:01:02
Time left: 00:00:00

The best date time departure
#6: Monday, July 4, 2022 3:00:00 AM

Рисунок 3.6 – Демонстрація результату роботи програми за базовим сценарієм, сценарій з перетином кордону

Для того, щоб користувач краще розумів, який кращий результат обрано додано відображення карти з маршрутом за кращий варіантом у центральній частині екрана.

панелі на сторінці «Map View» (рис. 3.7). Саме ця сторінка встановлена за замовчуванням під час відкриття програми та після обрахунків.

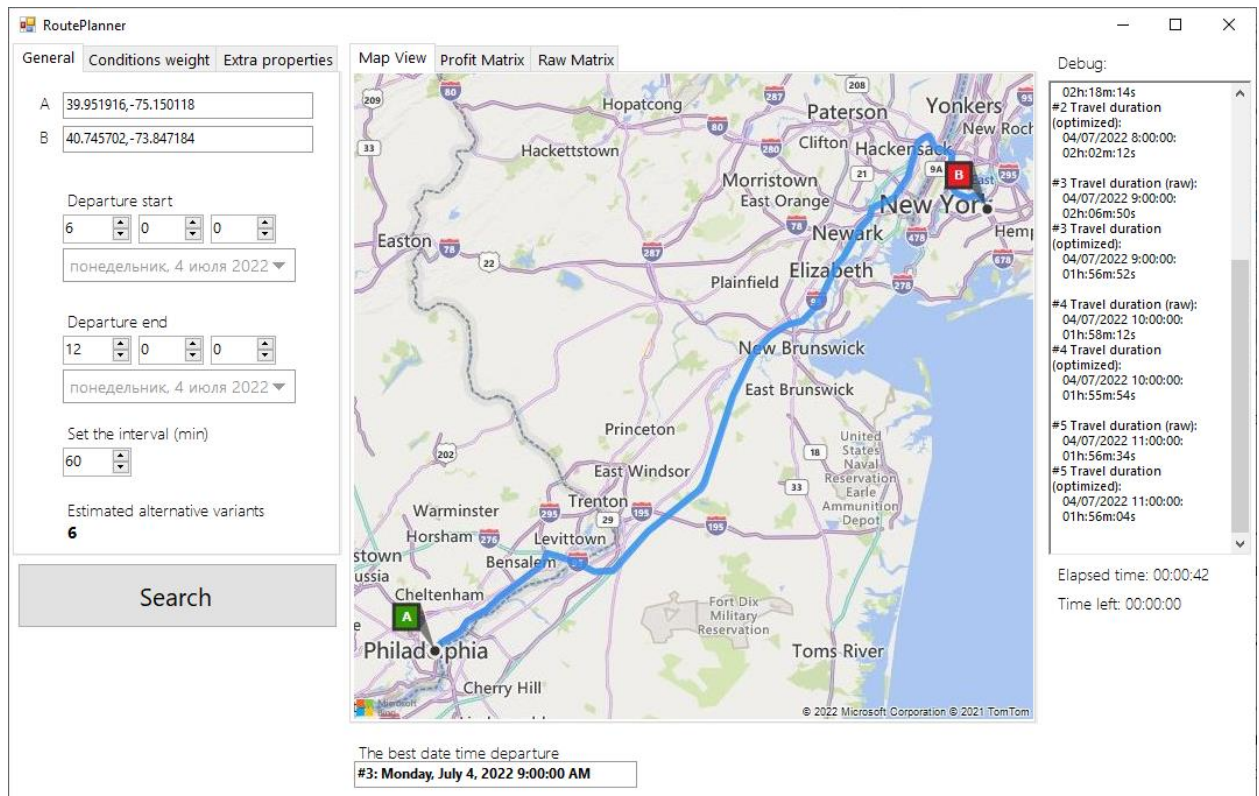


Рисунок 3.7 – Демонстрація результату роботи програми після базового сценарію, відображення карти маршруту кращого варіанту

Для кожної з умов можна вказати бажаний коефіцієнт у інтервалі від 0 до 100 з кроком 1, що означає вагу або важливість умови у відсотках. Вагу кожної з умов вирішує користувач, оператор-спеціаліст самотужки та може підібрати найближчий до своїх умов. У такому випадку значення 0 буде відповідати найменш важливій умові, 100 – найбільш важливій умові. Функціонал знаходиться у лівій панелі на сторінці «Conditions weight», інтерфейс зображено на рисунку 3.8.

The image shows a software interface with three tabs: 'General', 'Conditions weight', and 'Extra properties'. The 'Conditions weight' tab is active. It contains three sliders, each with a corresponding numerical input field and a percentage sign. The sliders are for 'Coefficient for Departure', 'Coefficient for Delay', and 'Coefficient for Border Crossing'. All three sliders are positioned at the 100% mark. Below the sliders is a button labeled 'Update' with a refresh icon. At the bottom of the interface is a large button labeled 'Search'.

Рисунок 3.8 – Демонстрація сторінки з редагуванням вхідних даних для коефіцієнтів для умов

Обирати можна двома способами: зміщувати повзунок або вписати значення у відповідне поле.

Застосувати коефіцієнти можна двома способами:

- задати умови для маршруту;
- поновити вже існуючий за допомогою відповідної кнопки.

Реалізація коефіцієнтів зроблена у класі `AltVariantsCollection.cs` у методі `EvaluateTotal` (рис. 3.9).


```

public AltVariantsCollection EvaluateTotal(double coeff1, double coeff2, double coeff3)
{
    coeff1 /= 100;
    coeff2 /= 100;
    coeff3 /= 100;
    foreach (AlternativeVariant altVar in this)
    {
        altVar.EvaluationDeparuteTime *= coeff1;
        altVar.EvaluationDelayTime *= coeff2;
        altVar.EvaluationCountryChange *= coeff3;
        altVar.EvaluationTotal = (altVar.EvaluationDeparuteTime + altVar.EvaluationDelayTime + altVar.EvaluationCountryChange)/3;
        altVar.EvaluationTotal = Math.Round(altVar.EvaluationTotal, 2);
    }
    return this;
}

```

Рисунок 3.9 – Демонстрація обрахування загальної оцінки з використанням коефіцієнтів

Результат роботи до використання коефіцієнтів продемонстровано на рисунку 3.10, після використання – на рисунку 3.11. Можна побачити, що кращий варіант був змінений з 3 на 4 після зміни коефіцієнту для відправлення 100% до 20%.

General			Map View					
Conditions weight			Profit Matrix					
Extra properties			Raw Matrix					
Coefficient for Departure	100 %		Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
Coefficient for Delay	100 %		0	2022/07/04 6:00:00	100	17,02	0	39,01
Coefficient for Border Crossing	100 %		1	2022/07/04 7:00:00	80	0	0	26,67
Update			2	2022/07/04 8:00:00	60	59,79	0	39,93
			3	2022/07/04 9:00:00	40	93,83	0	44,61
			4	2022/07/04 10:00:00	20	100	0	40
			5	2022/07/04 11:00:00	0	98,94	0	32,98
			*					

Рисунок 3.10 – Демонстрація матриці прибутку без використання коефіцієнтів

The screenshot shows a software interface with two main panels. The left panel, titled 'General', contains three sliders for coefficients: 'Coefficient for Departure' (set to 20%), 'Coefficient for Delay' (set to 100%), and 'Coefficient for Border Crossing' (set to 100%). An 'Update' button is located at the bottom of this panel. The right panel, titled 'Profit Matrix', displays a table with the following data:

	Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
	0	2022/07/04 6:00:00	20	17,02	0	12,34
	1	2022/07/04 7:00:00	16	0	0	5,33
	2	2022/07/04 8:00:00	12	59,79	0	23,93
	3	2022/07/04 9:00:00	8	93,83	0	33,94
	4	2022/07/04 10:00:00	4	100	0	34,67
	5	2022/07/04 11:00:00	0	98,94	0	32,98
	*					

Рисунок 3.11 – Демонстрація матриці прибутку з використанням коефіцієнтів

Окрім вагових коефіцієнтів за окремими умовами також можна додати коефіцієнти до загальної оцінки за окремими альтернативними варіантами за критерієм часу відправлення, що називається *Departure Time Rule*. За допомогою таких коефіцієнтів або правил можна виокремити варіанти за деякий період часу. Таке правило формується з трьох складових:

1. Особливий день або дні;
2. Особливий час;
3. Коефіцієнт для цього проміжку або проміжків часу.

День або дні можна задати двома способами:

– вказати, у який день або дні тижня буде особливе правило, у такому випадку правило застосовується періодично;

– вказати інтервал особливих днів у календарі.

Час можна вказати двома способами, можна обрати:

– цілий день;

– інтервал часу протягом дня.

Застосувати коефіцієнти для особливого дня та часу можна двома способами: задати умови для маршруту або поновити вже існуючий за допомогою відповідної кнопки.

Такий інструмент допомагає відкинути неприйнятні варіанти, наприклад, нічні відправлення, відправлення по вихідним, по святам або просто зробити менш пріоритетними. Це є необхідністю, оскільки у багатьох випадках програма буде обирати нічні варіанти, які зазвичай є найближчими і найшвидшими.

Структура модуля з програмною реалізацією зображена на рисунку 3.12.

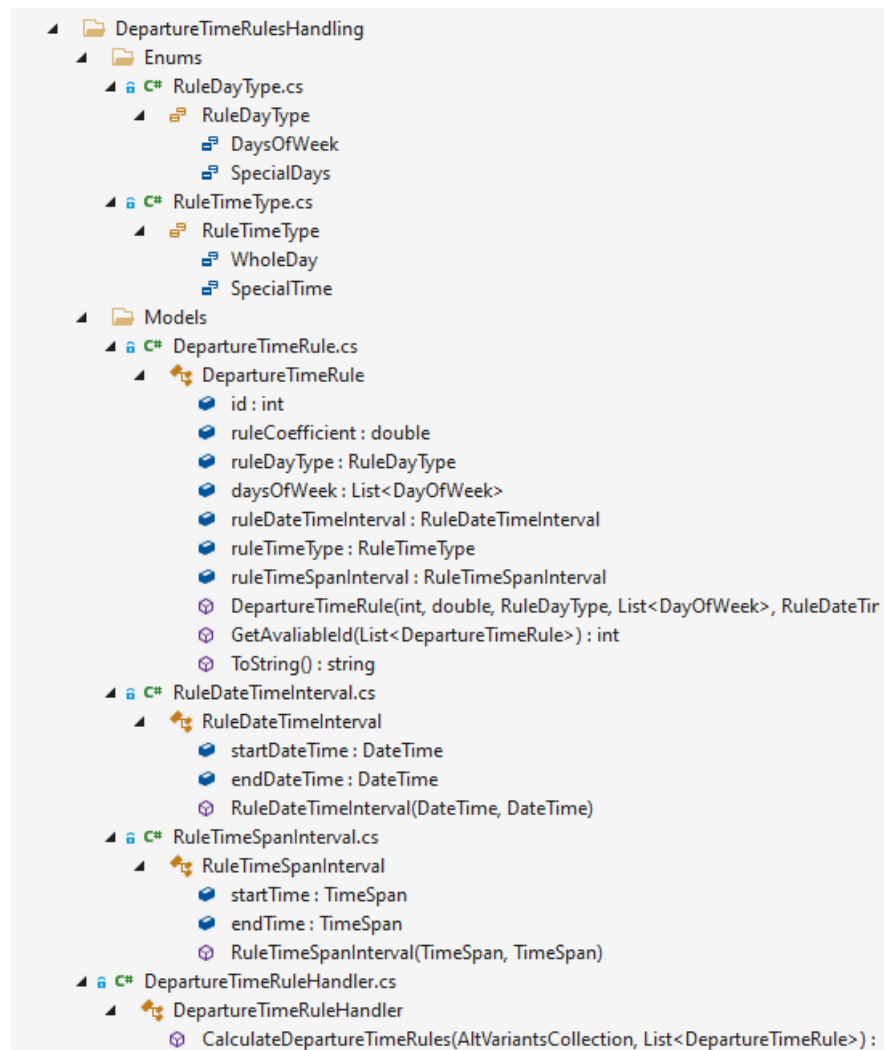


Рисунок 3.12 – Демонстрація структури модуля для роботи з альтернативними варіантами з особливим днем і часом відправлення

Функціонал для особливий правил до часу відправлення знаходиться у лівій панелі на сторінці «Extra properties» (рис. 3.13). На сторінці присутня таблиця з поточними правилами, можливість додати нові, видалити існуючі.

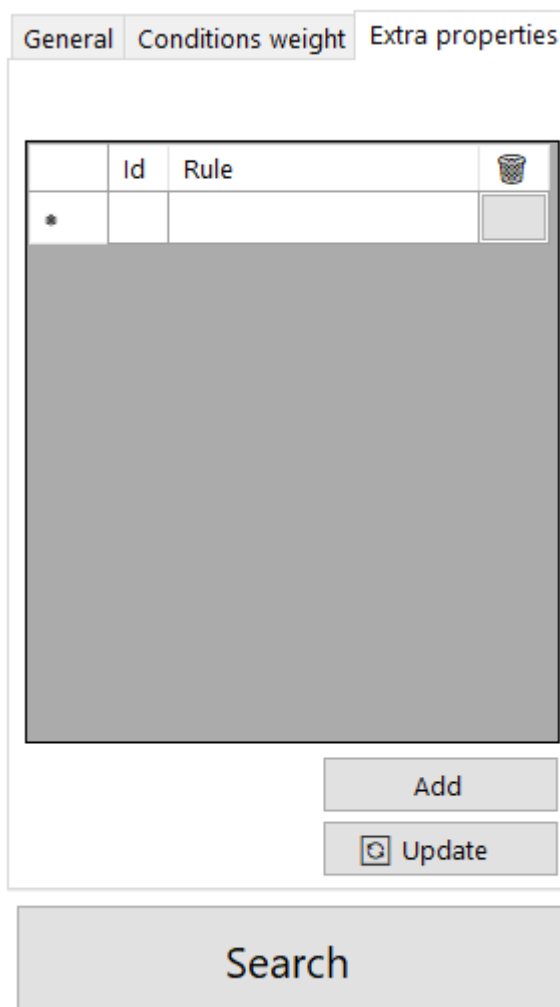


Рисунок 3.13 – Демонстрація сторінки для особливих правил для відправлень

Після натискання на кнопку «Add» відкривається вікно для заповнення вхідних даних для особливого правила (рис. 3.14). У вікні присутні усі необхідні поля відповідно до формування зазначеного вище.

Рисунок 3.14 – Демонстрація вікна з вхідними даними для особливого часу відправлення

Результат роботи до використання коефіцієнтів для особливого дня та часу продемонстровано на рисунку 3.15, після використання – на рисунку 3.16. Можна побачити, що кращий варіант був змінений з 0 на 9 після зміни коефіцієнту для додавання особливий часових правил. Було додане правило для понеділків з часовим проміжком від 0:00:00 до 6:00:00 та коефіцієнтом 10%.

General			Conditions weight	Extra properties	Map View	Profit Matrix	Raw Matrix
	Id	Rule					
*							
			Add				
			Update				

	Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
▶	0	2022/07/04 0:00:00	100	100	0	66,67
	1	2022/07/04 1:00:00	90,91	100	0	63,64
	2	2022/07/04 2:00:00	81,82	100	0	60,61
	3	2022/07/04 3:00:00	72,73	98,37	0	57,03
	4	2022/07/04 4:00:00	63,64	90,48	0	51,37
	5	2022/07/04 5:00:00	54,55	63,74	0	39,43
	6	2022/07/04 6:00:00	45,45	10,02	0	18,49
	7	2022/07/04 7:00:00	36,36	0	0	12,12
	8	2022/07/04 8:00:00	27,27	35,19	0	20,82
	9	2022/07/04 9:00:00	18,18	55,23	0	24,47
	10	2022/07/04 10:00:00	9,09	58,86	0	22,65
	11	2022/07/04 11:00:00	0	58,23	0	19,41
*						

Рисунок 3.15 – Демонстрація матриці прибутку без використання коефіцієнтів для особливого часу відправлення

General			Conditions weight	Extra properties	Map View	Profit Matrix	Raw Matrix
	Id	Rule					
▶	0	Rule coefficient: 0,1, Day type: DaysOfWeek, Days of week: Mon , Date interval: 20.06.2022 21.06.2022, Time type: SpecialTime, Time interval: 00:00:00 06:00:00					
*							
			Add				
			Update				

	Id	Departure Time	F1 Evaluation (Departure Time)	F2 Evaluation (Delay Time)	F3 Evaluation (Border Crossing)	Total Evaluation
	0	2022/07/04 0:00:00	100	100	0	6,667
	1	2022/07/04 1:00:00	90,91	100	0	6,364
	2	2022/07/04 2:00:00	81,82	100	0	6,061
	3	2022/07/04 3:00:00	72,73	98,37	0	5,703
	4	2022/07/04 4:00:00	63,64	90,48	0	5,137
	5	2022/07/04 5:00:00	54,55	63,74	0	3,943
	6	2022/07/04 6:00:00	45,45	10,02	0	1,849
	7	2022/07/04 7:00:00	36,36	0	0	12,12
	8	2022/07/04 8:00:00	27,27	35,19	0	20,82
▶	9	2022/07/04 9:00:00	18,18	55,23	0	24,47
	10	2022/07/04 10:00:00	9,09	58,86	0	22,65
	11	2022/07/04 11:00:00	0	58,23	0	19,41
*						

Рисунок 3.16 – Демонстрація матриці прибутку з використанням коефіцієнтів для особливого часу відправлення

Як було зазначено вище, метод знаходження середнього арифметичного є універсальним для більшості сценаріїв, але окрім цього, також можна розглянути кілька потенційних критеріїв оцінювання для сценаріїв такої програми.

Критерій мінімаксий найчастіше використовується для прийняття рішень на різних етапах проектування, коли відсутність ризику може призвести до втрат під час реалізації рішення. У такому випадку буде прийнято рішення з мінімально можливими втратами.

Критерій Гурвіца доцільно використовувати при невеликій кількості рішень, допускається певний ризик при реалізації рішення та коли ймовірність виникнення зовнішніх умов невідома.

Критерій добутків подібний до мінімаксного, але більш оптимістичний. Наближений до середнього арифметичного, але у разі нормалізації від 0 буде мати багато варіантів, але повністю відпадають або отримують мінімально можливу загальну оцінку. У цьому також є сенс, коли слід відкинути усі сумнівні варіанти.

3.3 Побудова та відправлення запиту до сервісу Bing Maps API

Для того, щоб отримати відповідь від сервісу попередньо слід сформулювати запит до нього за допомогою URL-адреси. Сформована адреса вже може бути використана або перевірена багатьма способами:

- вставити у рядок адреси у будь-якому браузері;
- вставлена у допоміжні програмні застосунки, наприклад Swagger або Postman;
- бути сформована та оброблена сторонніми бібліотеками, наприклад Bing Maps REST Tool Kit від Microsoft;
- бути оброблена безпосередньо у розроблюваному застосунку.

У загальному вигляді повний запит для двох пунктів з часом відправлення при використанні автомобіля виглядає наступним чином:

<https://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0={wayPoint1}&wp.1={wayPoint2}&optmz={optimize}&timeType={timeType}&dateTime={dateTime}&output={output}&key={BingMapsKey}>

Розглянемо таблицю 3.17, де описано кожен з необхідних параметрів для реалізації програми.

Таблиця 3.17 – Параметри для запиту до сервісу

Назва параметру	Опис	Значення
wayPoint1	Точка для початку маршруту.	Може приймати значення у вигляді координат та у вигляді повної адреси багатьма державними мовами. Наприклад, “39.951916,-75.150118” або “Миколаїв”.
wayPoint2	Точка для кінця маршруту. Може бути додано до 20 проміжкових пунктів.	
optimize	Додатковий атрибут. Визначає метод оптимізації при побудові маршруту.	Використовується значення “timeWithTraffic”. Маршрут розраховується для мінімізації часу та використовує поточну або прогнозовану інформацію про затори.
timeType	Вказує, як інтерпретувати значення дати та часу перевезення, визначені dateTime параметром.	Використовується “Departure”. Тоді dateTime містить бажаний день і час відправлення.

Закінчення таблиці 3.17

dateTime	Вказує час, який необхідний для виконання побудови маршруту. Те як буде використано вказано у параметрі timeType.	Наприклад, 07/04/2022 08:45:00
output	Формат відповіді.	Використовується "xml".
BingMapsKey	Ключ авторизації для доступу до API.	Ключ отримується під час реєстрації свого проекту у сервісі Bing Map API.

Прикладами для запиту зі вказаними значеннями параметрів можуть бути:

Приклад 1:

1. Пункт відправлення: 39.951916,-75.150118.
2. Пункт прибуття: 40.745702,-73.847184.
3. Дата і час відбуття: 2022/07/04 6:00:00.

Запит:

<https://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0=39.951916,-75.150118&wp.1=40.745702,-73.847184&optmz=timeWithTraffic&timeType=Departure&dateTime=2022/07/04%206:00:00&output=xml&key=ApNf4cdMo33Rss3h5mOCtQYIYgEsonbD4PatMfaq8-9RPSQ-orq8vnk3lMuEcMx9>

Приклад 2:

1. Пункт відправлення: Миколаїв.
2. Пункт прибуття: Одеса.
3. Дата і час відбуття: 2022/06/29 8:00:00.

Запит:

<https://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0=Миколаїв&wp.1=Одеса&optmz=timeWithTraffic&timeType=Departure&dateTime=2022/06/29%208:00:00&output=xml&key=ApNf4cdMo33Rss3h5mOCtQYIYgEsonbD4PatMfaq8-9RPSQ-orq8vnk3lMuEcMx9>

Приклад 3:

1. Пункт відправлення: вулиця Лазурна, 52А, Миколаїв, Україна.
2. Пункт прибуття: вулиця Десантників, 69, Миколаїв, Україна.
3. Дата і час відбуття: 2022/07/04 11:55:00.

Запит:

[https://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0=вулиця Лазурна, 52А, Миколаїв, Україна&wp.1=вулиця Десантників, 69, Миколаїв, Україна&optmz=timeWithTraffic&timeType=Departure&dateTime=2022/07/04%2011:55:00&output=xml&key=ApNf4cdMo33Rss3h5mOCtQYIYgEsonbD4PatMfaq8-9RPSQ-orq8vnk3lMuEcMx9](https://dev.virtualearth.net/REST/V1/Routes/Driving?wp.0=вулиця%20Лазурна,%2052А,%20Миколаїв,%20Україна&wp.1=вулиця%20Десантників,%2069,%20Миколаїв,%20Україна&optmz=timeWithTraffic&timeType=Departure&dateTime=2022/07/04%2011:55:00&output=xml&key=ApNf4cdMo33Rss3h5mOCtQYIYgEsonbD4PatMfaq8-9RPSQ-orq8vnk3lMuEcMx9)

Перевірити будь-який з цих запитів можна у браузері (рис 3.18). Стрічка з параметром <StatusCode> зі значенням 200 означає, що запит пройшов вдало.

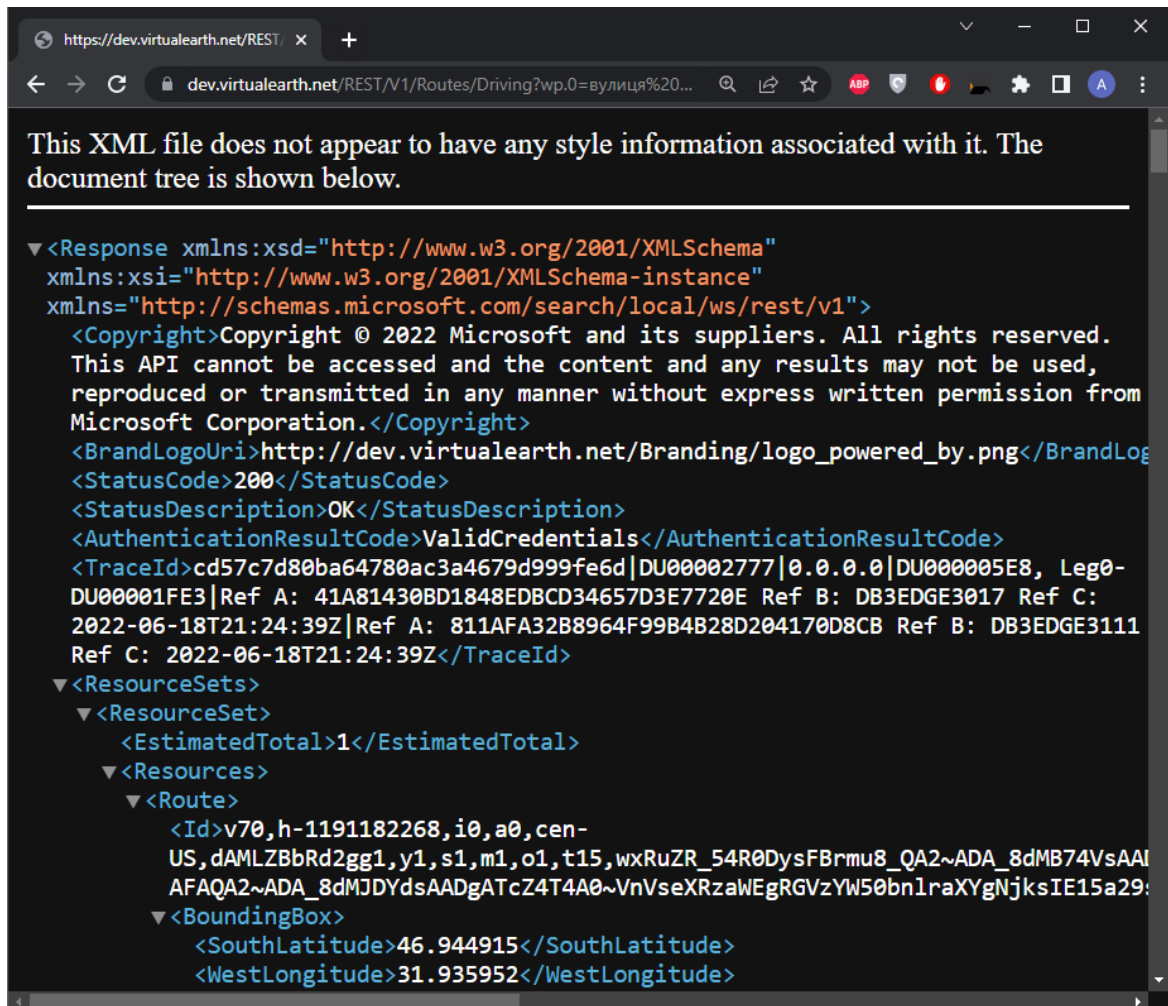


Рисунок 3.18 – Демонстрація запити до сервісу

3.4 Побудова та зчитування відповіді від сервісу Bing Maps API

Обрахунки програми повністю базуються на відповіді, яка отримується від сервісу Bing Maps API за URL-адресою. Ця відповідь містить усі необхідні дані про маршрут, інформація про час у дорозі, відстань та інші деталі про кожний відрізок маршруту. Відрізок маршруту – це ділянка маршруту, яка визначена двома точками. Наразі реалізовано сценарії з маршрутами тільки з двома точками, але додавання інших точок повністю можливе. Відрізок маршруту складається з кроків, які містять інструкції для проходження та додаткові дані, наприклад попередження. У моєму випадку використовуються точки поворотів з інструкції для подальшої оптимізації у вигляді широти та довготи та попередження про перетин кордону. Оскільки запит робиться для XML файлу, розглянемо побудову такого файлу. Будуть розглянуті тільки головні ноди файлу.

- `<Route>` – один або більше, якщо необхідно дізнатися кілька повністю різних шляхів. У моєму випадку це тільки один маршрут визначений двома точками.
 - `<DistanceUnit>` – одиниця вимірювання довжини, використовуються кілометри.
 - `<DurationUnit>` – одиниця вимірювання часу, використовуються секунди.
 - `<TravelDistance>` – відстань за кращим варіантом, якщо присутні альтернативи, у моєму випадку альтернатива тільки одна.
 - `<TravelDuration>` – час проходження за кращим варіантом, якщо присутні альтернативи, у моєму випадку альтернатива тільки одна.
 - `<TravelDurationTraffic>` – час проходження за кращим варіантом з врахуванням заторів, якщо присутні альтернативи, у моєму випадку альтернатива тільки одна.

- <RouteLeg> – один або більше, якщо необхідно дізнатися кілька альтернативних варіантів проходження між кількома точками. У моєму випадку це тільки один маршрут, який є кращим у конкретний час.
 - <TravelDistance> – відстань одного з альтернативних шляхів.
 - <TravelDuration> – час проходження одного з альтернативних шляхів.
 - <TravelMode> – транспортний засіб або спосіб пересування, використовується «Driving», тобто автомобілем.
 - <ActualStart> – точка початку маршруту.
 - <Latitude> – широта.
 - <Longitude> – довгота.
 - <ActualEnd> – точка кінця маршруту.
 - <Latitude> – широта.
 - <Longitude> – довгота.
 - <ItineraryItem> – крок проходження маршруту. В залежності від довжини та кількості поворотів за маршрутом може бути від двох (кінець і початок) до кількох сотень, обмеження немає. Зазвичай це 10-40 кроків.
 - <TravelMode> – транспортний засіб або спосіб пересування за окремим кроком, використовується «Driving».
 - <TravelDistance> – відстань проходження кроку, може бути десятків метрів у випадку з щільними кварталами та вулицями, до десятків кілометрів у випадку з довгим прямим відрізом магістралі.

- <TravelDuration> – час проходження кроку (без врахування заторів).
- <ManeuverPoint> – точка маневру або повороту для проходження до наступного кроку.
 - <Latitude> – широта.
 - <Longitude> – довгота.
- <Warning> – повідомлення про особливості кроку з потенційною загрозою. Використовуються попередження про перетин кордону.
- <StartTime> – дата і час відбуття з початкової точки маршруту.
- <EndTime> – дата і час прибуття до кінцевої точки маршруту.

Окрім зазначених нодів, також присутні ще батьківські до <Route>. У порядку спадання: <Response>, <ResourceSets>, <ResourceSet>, <Resources>. Ці ноди зберігають метадані відповіді від API. Наприклад, дані про авторське право, статус відповіді, посилання на логотип, статус авторизації, тощо.

Для того, щоб використовувати дані з відповіді спочатку треба розробити серіалізацію XML файлу з відповідними нодами. При цьому, як зазначено вище, усі ноди не будуть використовуватися. Згідно будови відповіді розроблено таку структуру (рис. 3.19).

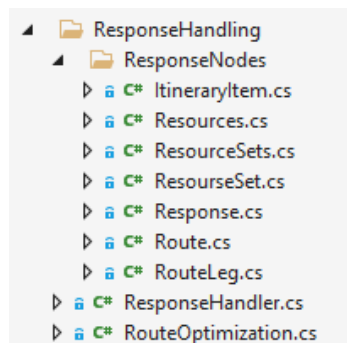


Рис. 3.19 – Демонстрація структури папки з класами для зчитування відповіді API.

Папка ResponseNodes – містить усі необхідні класи відповідно до кожної ноди для аналізу.

Клас ResponseHandler оперує з нодами на записує їх у списки з альтернативними варіантами для подальшої роботи з ними.

Клас RouteOptimization містить методи для оптимізації розрахунку часу проходження за кожним з альтернативних варіантів.

Для того, щоб переглянути дані, які були зчитані з відповіді API та ще не були оброблені для системи прийняття рішення був зроблений відповідний інтерфейс. У центральній панелі на сторінці «Raw Matrix» (рис. 3.20) присутня таблиця з необробленими, ненормованими даними на проміжковому етапі, коли можна подивитися не якісні дані, а кількісні. Окрім часу відправлення, проходження, затримки та перетину кордонів, також присутні інші дані після оптимізації, але вони будуть розглянуті пізніше.

Map View Profit Matrix Raw Matrix									
	Id	Departure Time	Travel Time (Raw)	Travel Time (Optimized)	Delay Time (Raw)	Delay Time (Optimized)	Travel Time Difference	Travel Time Difference %	Border Crossing
▶	0	2022/07/04 6:00:00	01:53:04	02:08:54	00:00:00	00:13:00	00:15:50	14	0
	1	2022/07/04 7:00:00	02:03:39	02:11:34	00:10:35	00:15:40	00:07:55	6,4	0
	2	2022/07/04 8:00:00	02:18:14	02:02:12	00:25:10	00:06:18	00:16:02	-11,6	0
	3	2022/07/04 9:00:00	02:06:50	01:56:52	00:13:46	00:00:58	00:09:58	-7,9	0
	4	2022/07/04 10:00:00	01:58:12	01:55:54	00:05:08	00:00:00	00:02:18	-1,9	0
	5	2022/07/04 11:00:00	01:56:34	01:56:04	00:03:30	00:00:10	00:00:30	-0,4	0
*									

Рисунок 3.20 – Демонстрація сторінки з необробленими даними, проміжковий етап з кількісними значеннями

3.5 Оптимізація часу проходження маршруту

Як було зазначено раніше, було вирішено реалізувати оптимізацію розрахунку часу проходження маршрутів з врахуванням стану заторів. Використовуючи метод дискретизації необхідно розбити маршрут на проміжкові пункти, які будуть опорними для перерахунку. У якості таких пунктів або точок були використані кроки «Itinerary Item» проходження маршруту. Оскільки кожен крок вже містить інструкції по проходженню поворотів та розвилок у ноді «Maneuver Point», були використані саме вони. Точка «Maneuver Point» задовольняє вимоги до дискретизації. Затяжні відрізки через магістралі як раз не вимагають частого перерахунку, а короткі відрізки у щільних районах як раз мають більше інструкцій до проходження.

Потенційно можна отримати ще додаткові проміжні точки, розділивши довгі магістральні відрізки, наприклад, довжиною 60 км, на дрібні по 10 км. Та навпаки, точки, що знаходяться дуже близько одна до одної, наприклад, до 200 метрів, можна об'єднувати. Таким чином можна збільшити точність та продуктивність.

Також можливий повністю новий метод знаходження маршруту. Кожного разу після перерахунку одного з проміжних пунктів проходження робити пошук найшвидшого маршруту та таким чином корегувати загальний шлях. У такому випадку буде отримано взагалі новий шлях, у якому враховуються затори, що з'являються та зникають під час проходження маршруту, та перебудовується найшвидший шлях згідно з поточними даними про затори або інші обставини.

Схема методу оптимізації зображена на рисунку 3.21, код у додатку Б. Робота методу підвищує точність обрахунку часу проходження, але вимагає значного більше часу для роботи, аніж просто аналіз відповіді «як є», тобто просто взяття даних з відповіді до усього маршруту, а не окремих його відрізків. Загалом, цей метод вирішує проблему, яку можна дізнатися зі структури відповіді. Проблема у тому, що для кожного ItineraryItem відсутній

параметр часу проходження з врахуванням заторів. Такий параметр `TravelDurationTraffic` присутній лише на усьому маршруті у ноді `Route`, а не в окремих його кроках.

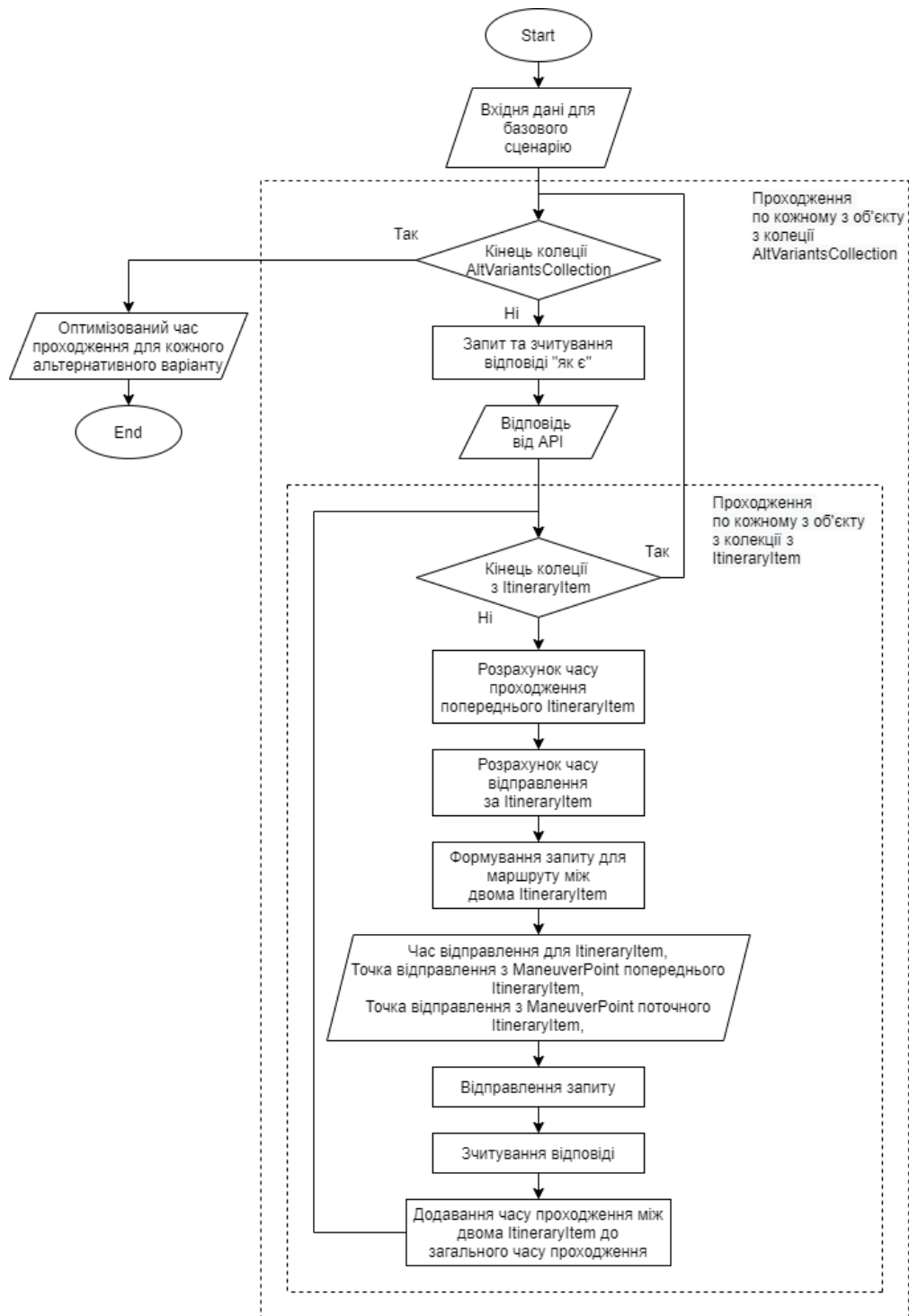


Рисунок 3.21 – Блок-схема алгоритму роботи методу Optimize

Розглянемо роботу методу Optimize у застосунку. Спеціально для порівняння часу до та після оптимізації існує таблиця на сторінці «Raw Matrix» у центральній панелі (рис. 3.22). У таблиці присутні такі стовпці, які цікаві у контексті оптимізації:

1. Travel Time (Raw) – загальний час проходження маршруту отриманий з відповіді «як є».
2. Travel Time (Optimized) – загальний час проходження маршруту отриманий після оптимізації.
3. Delay Time (Raw) – різниця між поточним та найкращим часом проходження серед поточних альтернативних варіантів.
4. Delay Time (Optimized) – різниця між поточним та найкращим часом проходження серед поточних альтернативних варіантів після оптимізації.
5. Travel Time Difference – кількісна різниця у часі між необробленим та оптимізованим часом проходження.
6. Travel Time Difference,% – якісна різниця у часі між необробленим та оптимізованим часом проходження.

Розглянемо детальніше таблицю зі сторінки «Raw Matrix» (рис. 3.21). Таблиця була відсортована за стовпцем «Travel Time Difference, %», тому можна побачити, які варіанти зазнали найбільших змін. Від’ємні відсотки означають, що оптимізований час виявився менше за необроблений, додатні відсотки – навпаки.

Якщо розглянути варіант 7 з відправленням о 8:20:00, то час зменшиться на 12%. Це обумовлено тим, через деякий час проходження маршруту стан заторів станом на 8:20:00 є дуже нерелевантним, оскільки «час пік» вже буде у минулому.

Відповідно у альтернативному варіанті 1 з відправленням о 6:00 маємо обернену ситуацію – збільшення на 14%. Це обумовлено тим, що стан заторів о 6:00 є невірним під кінець і через деякий час проходження якраз почнеться «час пік», що збільшить загальний час.

Id	Departure Time	Travel Time (Raw)	Travel Time (Optimized)	Delay Time (Raw)	Delay Time (Optimized)	Travel Time Difference	Travel Time Difference, %
7	2022/07/04 8:20:00	02:16:11	01:59:50	00:23:07	00:04:00	00:16:21	-12
6	2022/07/04 8:00:00	02:18:14	02:02:12	00:25:10	00:06:22	00:16:02	-11,6
8	2022/07/04 8:40:00	02:11:48	01:58:00	00:18:44	00:02:10	00:13:48	-10,5
9	2022/07/04 9:00:00	02:06:50	01:56:52	00:13:46	00:01:02	00:09:58	-7,9
5	2022/07/04 7:40:00	02:14:24	02:04:00	00:21:20	00:08:10	00:10:24	-7,7
10	2022/07/04 9:20:00	02:02:48	01:56:14	00:09:44	00:00:24	00:06:34	-5,3
11	2022/07/04 9:40:00	02:00:03	01:55:57	00:06:59	00:00:07	00:04:06	-3,4
12	2022/07/04 10:00:00	01:58:12	01:55:54	00:05:08	00:00:04	00:02:18	-1,9
4	2022/07/04 7:20:00	02:09:35	02:07:32	00:16:31	00:11:42	00:02:03	-1,6
13	2022/07/04 10:20:00	01:57:08	01:55:50	00:04:04	00:00:00	00:01:18	-1,1
14	2022/07/04 10:40:00	01:56:42	01:55:54	00:03:38	00:00:04	00:00:48	-0,7
15	2022/07/04 11:00:00	01:56:34	01:56:04	00:03:30	00:00:14	00:00:30	-0,4
16	2022/07/04 11:20:00	01:56:37	01:56:11	00:03:33	00:00:21	00:00:26	-0,4
17	2022/07/04 11:40:00	01:56:43	01:56:28	00:03:39	00:00:38	00:00:15	-0,2
3	2022/07/04 7:00:00	02:03:39	02:11:34	00:10:35	00:15:44	00:07:55	6,4
2	2022/07/04 6:40:00	01:58:39	02:13:37	00:05:35	00:17:47	00:14:58	12,6
0	2022/07/04 6:00:00	01:53:04	02:08:54	00:00:00	00:13:04	00:15:50	14
1	2022/07/04 6:20:00	01:55:07	02:12:44	00:02:03	00:16:54	00:17:37	15,3

The best date time departure
#9: Monday, July 4, 2022 9:00:00 AM

Рисунок 3.22 – Демонстрація сторінки «Raw Matrix» з відсортованою таблицею за стовпцем «Travel Time Difference, %»

3.6 Обмеження на використання застосунку

Для того, щоб використовувати інструменти, які дає Bing Maps API необхідна ліцензія. Транзакції зараховуються для кожного запиту REST Services Maps Bing, надісланого з дійсним ключем Bing Maps.

Ліцензії бувають різні в залежності від цілей продукту, від малих бізнесів до великих Enterprise-рішень. У таблиці 3.23 можна ознайомитись з існуючими варіантами, які будуть відповідати вимогам веб-картографування продукту і не будуть мати вимоги до сплати.

Таблиця 3.23 – Варіанти ліцензування Bing Maps API.

Тип програми	Очікуване використання програми
Веб-сайт. Відкритий для широкого загалу або через приватну мережу.	Менше 125 000 транзакцій, що підлягають оплаті за календарний рік.
Програма для розробки Windows.	Менше 50 000 сукупних транзакцій, які підлягають оплаті за будь-який 24-годинний період.
Мобільний додаток.	Менше 125 000 транзакцій, що підлягають оплаті за календарний рік.
Відстеження управління мобільними активами.	Тільки платне.
Веб-сайт для освітніх або некомерційних організацій, додаток Windows або мобільний додаток.	Менше 50 000 сукупних транзакцій, які підлягають оплаті за будь-який 24-годинний період.

Для цілей створюваної програми використовується ліцензія за останнім типом. Для того, щоб зрозуміти мало чи багато 50000 транзакцій або запитів до сервісу за 24-годинний період можна привести такий приклад.

Приклад (рис. 3.24):

- Маршрут від вулиця Десантників, 69, Миколаїв, Україна до Києва;
- Відправлення від 0:00 до 24:00 з інтервалом 1 година: 24 альтернативні варіанти;
- Час проходження: більше 6-ти годин;
- Проміжкові пункти для оптимізації часу проходження: приблизно 64.

Усього у такому прикладу буде зроблено 1500-1600 запитів. Таким чином, цей приклад можна обраховувати 31-33 разів на день, при рахунку того, що це є дуже довгий маршрут.

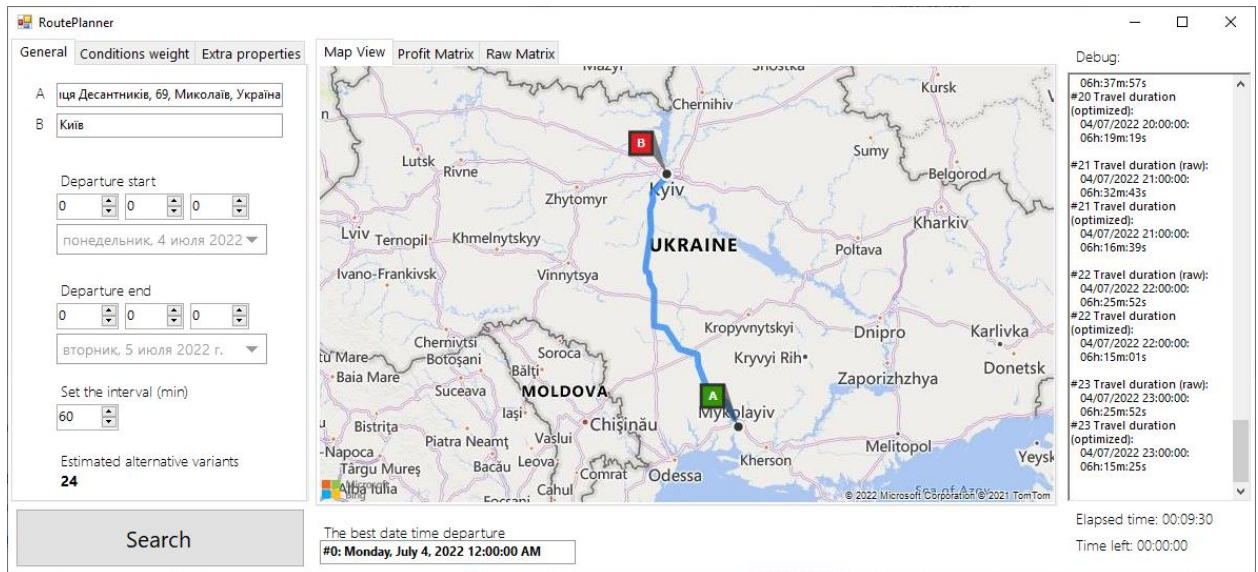


Рисунок 3.24 – Демонстрація побудованого маршруту від ЧНУ ім. Петра Могили до Києва

Якщо використовувати програму для більш простіших сценаріїв з часом проходження близько 2-х годин, то кількість запитів зменшиться до 400-500. Тобто буде обмежене кількістю у 100-125 обрахунків на день, що є повністю задовільним.

Карти та інструменти веб-картографування Bing мають різні рівні покриття для кожної країни або регіону світу. У додатку В приведено відомості про покриття доступних країн.

Також при великій кількості запитів іноді сервіс не може обробити одне або більше запитів та можна отримати повідомлення про помилку (рис. 3.25).

```
"RoutePlanner.exe" (CLR v4.0.30319: RoutePlanner.exe). Загружено "C:\Windows
\Microsoft.Net\assembly\GAC_MSIL\System.resources\v4.0.4.0.0_ru_b77a5c561934e089
\System.resources.dll". Сборка модуля выполнена без символов.
Вызвано исключение: "System.Net.WebException" в System.dll
Удаленный сервер возвратил ошибку: (404) Не найден.
Error. Getting response again... Try 1/10
```

Рисунок 3.25 – Демонстрація повідомлення про помилку отримання відповіді

Такі випадки бувають 1 до 200, але програма вміє опрацьовувати такі виключні випадки та посилає запит ще раз. Зазвичай така затримка не перебільшує кількох секунд. Якщо не обробляти такі ситуації обрахунок може зовсім зупинитися та довелось би починати спочатку.

3.7 Збір статистики та аналіз результату роботи

Найкраще програма себе проявляє на маршрутах більше 10-ти кілометрів, оскільки на невеликих відстанях затримка настільки мала, що потрапляє у похибку обрахування оптимізації. Похибка під час оптимізації присутня і дорівнює 0-2%, але може бути покращена.

Час обрахування є важливим фактором для продуктивності програми. Час обрахування практично повністю залежить від часу отримання відповідей від API, а отже, від якості з'єднання та швидкості передачі даних у мережі Інтернет. Час обрахунку для маршрутів довжиною 100-200 км, часом проходження до 2х годин та середньою кількістю проміжкових точок для оптимізації (Maneuver points) 20-30 за одним часом відправлення та швидкістю Інтернету у 1 Мбіт/сек становить 3-5 секунд. Таким чином 24 альтернативні варіанти можна обрахувати приблизно за 2 хвилини. При швидкому Інтернету час скорочується у багато разів.

Для більш детального аналізу розглянемо приклад:

1. Точка відправлення: 39.951916,-75.150118.
2. Точка прибуття: 40.745702,-73.847184
3. Кількість проміжкових пунктів: 20
4. Кількість альтернативних варіантів: 24
5. Всього запитів: 480 по 7-9 Кб та 24 по 40 Кб, сума: 4,2-5,3 Мб.
6. Швидкість Інтернету: 0,5 Мбіт/сек.

Сценарій був обрахований за 3:11 хв або 191 секунду. Тобто приблизно 40-50% часу очікується завантаження відповідей, увесь інший час займає надсилання запитів, оновлення графічного інтерфейсу та обробка відповідей.

3.8 Керівництво користувача

Для початку користування програмою виконайте кроки:

1. Перейдіть за посиланням

<https://github.com/Apache360/RoutePlanner>

2. Скачайте програму/проект у будь-яке зручне для вас місце.
3. Запустіть файл формату .exe. Програма не вимагає додаткового програмного забезпечення, окрім підключення до мережі Інтернет.

Для того, щоб зробити базовий запит виконайте наступні кроки:

1. Заповніть поле пункту відправлення «А». Можна вписати координати, можна адресу будь-якою мовою та будь-якої країни або регіону, де підтримуються сервіси Bing Maps.

2. Заповніть поле пункту призначення «В». Маршрут має бути сухопутний. Можна перетинати кордони країн. Не рекомендовано брати маршрути більше 1000 км через їх можливу недоцільність та відносно великий час для обрахунку.

3. Оберіть дату та час для першого прийняттого для вас відправлення за маршрутом у полях після «Departure start».

4. Оберіть дату та час для останнього прийняттого для вас відправлення за маршрутом у полях після «Departure end». Ці дані повинні бути пізніше у часі.

5. Оберіть інтервал часу у полі «Set the interval», за яким буде визначена періодичність відправлень. Значення у хвилинах.

6. Натисніть «Search» та зачекайте.

У полі «Estimated alternative variants» зображена кількість альтернативних варіантів для різних часів відправлення. Не бажано обраховувати більше 1000 варіантів через занадто довгий час обрахування. Час обрахування практично повністю залежить від якості вашого Інтернет-з'єднання та швидкості передачі даних з мережі Інтернет.

Час, який пройшов для обрахування зображень у полі «Elapsed time» у правій панелі.

Приблизний час, який залишився для обрахування зазначений у полі «Time left» у правій панелі.

У текстовому полі «Debug» можна спостерігати процес оброблення даних.

Результат роботи програми:

– Найкращий варіант буде виведено у центральній панелі у поле «The best date time departure» з номером альтернативного варіанту, часом і датою відправлення.

– На сторінці «Map View» у центральній панелі буде виведено зображення маршруту за кращим варіантом. Зазвичай за одним сценарієм розрахунку присутні кілька маршрутів, які змінюються в залежності від часу відправлення.

– На сторінці «Profit Matrix» у центральній панелі буде виведено усі альтернативні варіанти з такими параметрами: номер, дата і час відправлення, оцінка дати і часу відправлення, оцінка затримки проходження маршруту, оцінка перетину кордонів, загальна оцінка. Саме за загальною оцінкою робиться прийняття рішення з кращого варіанту, але користувач може подивитися інші варіанти та їх значення для порівняння. Перевірити найгірші варіанти, нейтральні та інші кращі.

– На сторінці «Raw Matrix» у центральній панелі буде виведено усі альтернативні варіанти у неоціненому форматі: номер, дата і час відправлення, час проходження необроблених, час проходження оптимізованих, час затримки необроблених, час затримки оптимізованих, різниця між часом проходження необробленим та оптимізованим, відносна різниця між часом проходження необробленим та оптимізованим, кількість перетинів кордону.

У якості додаткових параметрів існують ваги коефіцієнтів «Conditions weight» та «Extra properties». Обидва допомагають змінити пріоритети при прийнятті рішення.

Для того, щоб змінити пріоритети у вигляді зміни коефіцієнтів при прийнятті рішення зробіть наступні кроки:

1. Заповніть дані для базового запиту або вже майте обрахований сценарій
2. Перейдіть до сторінки «Conditions weight».
3. Визначте необхідні для вас відсотки для кожного з присутніх коефіцієнтів.
4. Якщо маєте обрахований сценарій натисніть «Update» та зачекайте. Якщо не маєте – «Search» та зачекайте.
5. При змінених коефіцієнтах кращий результат може змінитися.

Для того, щоб змінити пріоритети у вигляді зміни важливості окремого часу відправлення при прийнятті рішення зробіть наступні кроки:

1. Заповніть дані для базового запиту або вже майте обрахований сценарій
2. Перейдіть до сторінки «Extra properties».
3. Натисніть «Add».
4. У запущеному вікні оберіть дні тижня, які будуть обраховуватися по-особливому або оберіть інтервал з двох дат.
5. Оберіть час для особливого обрахунку: цілий день або інтервал часу дня.
6. Оберіть коефіцієнт для обраних днів і часу. При обиранні коефіцієнту 0, такі варіанти будуть мати мінімальну можливу оцінку і будуть найбільш неприйнятними.

7. Натисніть «ОК».
8. Нове правило з'явиться у таблиці. Якщо бажаєте видалити його, натисніть кнопку справа від нього.
9. Якщо маєте обрахований сценарій натисніть «Update» та зачекайте. Якщо не маєте – «Search» та зачекайте.
10. При змінених коефіцієнтах кращий результат може змінитися.

Висновки до розділу 3

В даному розділі спроектовано та реалізовано програмне забезпечення, яке повністю задовольняє поставленим до нього вимогам. Успішно реалізовані функції відправлення, отримання та обробки запитів до прикладного програмного інтерфейсу Bing Maps API. Згідно із зібраною статистикою та її аналізом, програма показує себе дуже ефективно, для більшості звичайних запитів затримок чи проблем з обрахуванням не виникало. Реалізована система прийняття рішення для кращого альтернативного рішення для часу відправлення за окремим маршрутом. Користувач має змогу оцінити критерії за власним розсудом та додати особливі для конкретного часового інтервалу. Також має можливість візуально подивитись на маршрут за допомогою карти з маршрутом. ПЗ потенційно може бути розширене та ускладнене додатковими критеріями для більш гнучкого налаштування користувацьких сценаріїв пошуку.

4 СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

У наші дні все більше підприємств почитають використовувати комп'ютери для роботи. Використання комп'ютерної техніки полегшує різні процеси на підприємствах. Але при всіх перевагах комп'ютерів, він може нести негативні наслідки для працівників, які працюють за такою технікою.

Люди які постійно працюють за комп'ютером зазвичай мають проблеми із зором, із спиною, а також страждають від електромагнітного випромінювання.

Але іншою важливою проблемою є дотримання санітарно – гігієнічних вимог приміщення для роботи з комп'ютерною технікою. Для нормальної роботи слід дотримуватися умов мікроклімату, освітлення. Також треба захистити користувача від впливу іонізуючих та неіонізуючих електромагнітних полів та впливу від випромінювання моніторів.

Метою спеціального розділу з охорони праці є створення безпечних і здорових умов праці на робочому місці. Відповідно до поставленої мети було сформульовано **завдання дослідження**:

1. Аналіз основних вимог до мікроклімату.
2. Розглянуті вимоги до освітлення приміщень та робочих місць та розрахунок коефіцієнту природного освітлення у приміщенні.
3. Захист користувачів від впливу іонізуючих та неіонізуючих електромагнітних полів та випромінювання моніторів.
4. Розглянути вимоги пожежної безпеки

4.1 Аналіз умов праці

Освітленість є складовою умов праці, яку необхідно відслідковувати та нормувати. Правильно спроектоване виробниче освітлення покращує умови зорової роботи, знижує стомлюваність, сприяє підвищенню продуктивності праці, позитивно впливає на виробниче середовище, надаючи позитивну психологічну дію на працюючу, підвищує безпеку праці і знижує травматизм. Нормованим параметром природного освітлення являється коефіцієнт природного освітлення (КПО), який визначається за формулою:

$$\text{КПО} = \left(\frac{E_{\text{внутр}}}{E_{\text{зовн}}} \right) * 100\%, \quad (4.1)$$

де:

$E_{\text{внутр}}$ — освітленість, що створюється в деякій точці приміщення світлом неба;

$E_{\text{зовн}}$ — зовнішня горизонтальна освітленість, створювана повністю відкритим склепінням.

В залежності від розряду виконуваних зорових робіт КПО має різні норми. Робота оператора ПК відноситься до робіт середньої точності (ІУ розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5 - 1,0мм), для яких при використанні бокового освітлення КПО=1,5%.

Для штучного освітлення нормованим параметром виступає $E_{\text{мін}}$ - мінімальний рівень освітленості, та $K_{\text{п}}$ - коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. У таблиці 4.1 наведені Норми штучного та природного освітлення виробничих приміщень.

Таблиця 4.1 – Норми штучного та природного освітлення виробничих приміщень

Характеристика зорових робіт	Найменший розмір об'єкта розпізнавання, мм	Розряд зорової роботи	Підрозряд зорової роботи	Штучне освітлення	Природне освітлення
				Освітленість, лк	КПО, %
				загальне освітлення	бокове освітлення
Середньої точності	0,5-1	IV	a	300	1,5
			б	200	
			в	200	
			г	200	
Малої точності	1-5	V	a	300	1,0
			б	200	
			в	200	
			г	200	
Груба	Більше 5	VI	–	200	1,0

Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк.

На практиці використовують метод відносної площі світлових прорізів для розрахунку природного освітлення. Відносна площа світлових прорізів – це відношення площі вікон до площі підлоги приміщення, що освітлюється:

$$\alpha = \frac{S_{\text{вікн}}}{S_{\text{підл}}} * 100\%, \quad (4.2)$$

де α - відносна площа світлових прорізів;

$S_{\text{вікн}}$ - сумарна площа вікон у приміщенні, м²;

$S_{\text{підл}}$ - площа підлоги у цьому ж приміщенні, м².

Норми штучного та природного освітлення виробничих приміщень за методом відносної площі світлових прорізів наведені у таблиці 4.2.

Таблиця 4.2 – Рекомендовані значення відносної площі світлових прорізів α для виробничих приміщень

Розряд зорової роботи	Вид робіт за ступенем точності	$\alpha, \%$
II	Дуже високої точності	16-20
III	Високої точності	14-16
IV	Середньої точності	12-14
V	Малої точності	10-12
VI	Грубі	8-10

Розрахуємо, якого розряду зорову роботу можна виконувати у приміщенні користувача ПК з однобічним природним освітленням. площа приміщення – 60 м²; 3 вікна розміром 2×1,5 м кожне.

Спочатку визначаємо сумарну площу вікон (світлових прорізів):

$$S_{\text{вік}} = 3 \cdot 2 \cdot 1,5 = 9 \text{ м}^2.$$

Визначаємо відносну площу світлових прорізів за формулою 4.2:

$$\alpha = \frac{S_{\text{вікн}}}{S_{\text{підл}}} * 100\% = \frac{9}{60} * 100\% = 15\%$$

За даними табл. 4.2 визначаємо, що у приміщенні можна запланувати виконання зорової роботи високої точності, що відповідає III розряду зорової роботи, що є задовільними умовами для роботи за ПК.

Оцінка умов праці за мікрокліматом виконується відносно параметрів температура, відносна вологість повітря, рухливість повітря, тощо.

В залежності від періоду року та категорії важкості виконуваних робіт параметри нормування змінюються. Робочі місця операторів ПК є постійними робочими місцями і для них встановлені оптимальні параметри мікроклімату, а при неможливості їх дотримання використовують допустимі параметри. В таблиці 4.3. наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу.

Таблиця 4.3 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 - 24°C
	Відносна вологість	40 - 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 - 25°C
	Відносна вологість	40 - 60%
	Швидкість руху повітря	0,1 - 0,2 м/с

За допомогою приладів було виміряно температуру та вологість у приміщенні і вони відповідають вказаним у таблиці для теплого періоду року.

Для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції. Об'єм кімнати складає 15м³, отже витрата свіжого повітря, що подається в приміщення, на одну людину в годину має становити не менше 30м³.

ПК, який розташований у приміщенні, являється джерелом тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Гранично допустима густина потоку енергії

$I_{г.д}$, Вт/м² є нормованим показником ІЧВ і встановлюється в залежності від площі опромінюваної поверхні тіла людини ($S_{опр}$).

Нормовані рівні складають:

$$I_{г.д} = 35 \text{ Вт/м}^2 \text{ при } S_{опр} > 50\%;$$

$$I_{г.д} = 70 \text{ Вт/м}^2 \text{ при } S_{опр} \sim 25-50\%;$$

$$I_{г.д} = 100 \text{ Вт/м}^2 \text{ при } S_{опр} < 25\%$$

Наступним важливим параметром умов праці є шум. В приміщенні знаходиться одне робоче місце з монітором, вінчестером в системному блоці, трьома вентиляторами системи охолодження ПК та клавіатурою. Таким чином у приміщенні мають місце шуми механічного і аеродинамічного походження.

Орієнтовні еквівалентні рівні звукового тиску джерел шуму, що діють на оператора на його робочому місці, представлені в таблиці 4.4.

Таблиця 4.4 – Рівні звукового тиску від різних джерел

Джерело шуму	Рівень шуму, дБА
Жорсткий диск	45
Вентилятор	45

Рівень шуму, що виникає від декількох некогерентних джерел, що працюють одночасно, підраховується на підставі принципу енергетичного підсумовування рівня інтенсивності окремих джерел:

$$L = 10 \lg \sum_{i=1}^n 10^{0,1 \cdot L_i}, \quad (4.3)$$

де L_i - рівень звукового тиску i -го джерела шуму;

n - кількість джерел шуму.

Підставивши значення рівня звукового тиску для кожного виду устаткування у формулу 4.3, отримаємо:

$$L = 10 \lg (10^{4,5i} + 10^{4,5}) = 48,01 \text{ дБ}$$

Допустимий еквівалентний рівень шуму для робочого місця оператора складає 65 дБА. Розраховане значення середнього рівня шуму не перевищує гранично допустимого рівня шуму для робочого місця оператора, тобто спеціальні заходи по зниженню рівня шуму не потребуються.

Останнім критерієм умов праці, який розглядається, є випромінювання монітору. В таблиці 4.5 представлені значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера, які є допустимими.

Таблиця 4.5 – Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50см від поверхні відеомонітора	10 В/м
Напруженість магнітної складової електромагнітного поля на відстані 50см від поверхні відеомонітора	0,3 А/м
Напруженість електростатичного поля не повинна перевищувати: для дорослих користувачів	20кВ/м
для дітей дошкільних установ і що вчаться середніх спеціальних і вищих учбових закладів	15кВ/м

Потужність експозиційної дози – це нормований параметр невикористаного рентгенівського випромінювання виступає. На відстані 5 см від поверхні екрану монітору її рівень не повинен перевищувати 100 мкР/год. Максимальний рівень рентгенівського випромінювання на робочому місці оператора комп'ютера не перевищує 20 мкР/год.

4.2 Техніка безпеки

До самостійної роботи з комп'ютером, ноутбуком, принтером, ксероксом, сканером, плазмовою панеллю, LCD-дисплеєм та іншою оргтехнікою допускаються особи, які досягли 18 річного віку, пройшли медичний огляд, ознайомлені з інструкцією з охорони праці при роботі з оргтехнікою, не мають протипоказань за станом здоров'я.

Під час роботи з ПК, необхідно дотримуватись ряду вимог:

1. Вмикайте і вимикайте комп'ютер, ноутбук та іншу оргтехніку тільки вимикачами, забороняється проводити вимкнення витягуванням вилки з розетки;
2. Забороняється знімати захисні пристрої з обладнання і працювати без них;
3. Не допускати до комп'ютера та оргтехніки сторонніх осіб, які не беруть участі в роботі;
4. Забороняється переміщати та переносити системний блок, монітор, принтер, будь-яке обладнання, яке знаходиться під напругою;
5. Забороняється під час роботи пити будь-які напої, приймати їжу;
6. Забороняється будь-яке фізичне втручання у пристрій комп'ютера, принтера, сканера, ксерокса під час їх роботи;
7. Забороняється залишати включене обладнання без нагляду;
8. Забороняється класти предмети на комп'ютерне обладнання, монітори, екрани та оргтехніку;
9. Суворо виконувати загальні вимоги з електробезпеки та пожежної безпеки;
10. Під час усунення застрягання паперу на ксероксі чи принтері, задля уникнення ураження електрострумом, необхідно відключити обладнання від електромережі. Необхідно також вимкати обладнання від мережі при тривалому простої;

11. Самостійно розбирати та проводити ремонт електронної та електронно-механічної частини комп'ютера, периферійних пристроїв, оргтехніки категорично забороняється. Ці роботи може виконувати тільки спеціаліст або інженер з технічного обслуговування комп'ютерної техніки;

12. Сумарний час безпосередньої роботи з персональним комп'ютером та іншою оргтехнікою протягом робочого дня має бути не більше 6 годин, для педагогів — не більше 4 годин у день;

13. Тривалість безперервної роботи з персональним комп'ютером та іншою оргтехнікою без регламентованої перерви не повинна перевищувати 2-х годин. Через кожну годину роботи слід робити перерву тривалістю 15 хв;

14. Під час регламентованих перерв, з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, слід виконувати комплекси вправ для очей або організовувати фізкультурні паузи.

4.3 Гігієна праці і виробнича санітарія

Виходячи з принципів гігієнічної класифікації, умови праці діляться на 4 класи – оптимальні, допустимі, шкідливі та небезпечні (екстремальні).

1 клас — оптимальні умови праці – такі умови, за яких зберігається не лише здоров'я працюючих, а й створюються передумови для підтримання високого рівня працездатності. Оптимальні гігієнічні нормативи виробничих факторів встановлені для мікроклімату і факторів трудового процесу. Для інших факторів за оптимальні умовно приймаються такі умови праці, за яких несприятливі фактори виробничого середовища не перевищують рівнів, прийнятих за безпечні для населення.

2 клас — допустимі умови праці – характеризуються такими рівнями факторів виробничого середовища і трудового процесу, які не перевищують встановлених гігієнічних нормативів, а можливі зміни функціонального стану організму відновлюються за час регламентованого відпочинку або до початку

наступної зміни та не чинять несприятливого впливу на стан здоров'я працюючих та їх потомство в найближчому і віддаленому періодах.

3 клас — шкідливі умови праці – характеризуються такими рівнями шкідливих виробничих факторів, які перевищують гігієнічні нормативи і здатні чинити несприятливий вплив на організм працюючого та/або його потомство.

Площа на одне робоче місце з комп'ютером для дорослих користувачів повинна складати не менше 6 м² для ЕОМ з електронно-променевою трубкою і 4,5 м² для рідкокристалічних моніторів, а обсяг не менше -20 м³. У приміщенні з ПК повинно бути природне і штучне освітлення. Неприпустимо розміщення робочих місць за моніторами для дорослих користувачів в підвальних приміщеннях. Приміщення з комп'ютерами повинні бути обладнані системами опалення, кондиціонування повітря або припливно-витяжної вентиляцією. Поверхня підлоги в приміщеннях експлуатації комп'ютерів повинна бути рівною, без вибоїн, неслизькою, зручною для очищення та вологого прибирання, мати антистатичні властивості. У приміщенні повинні бути аптечка першої медичної допомоги, вуглекислотний вогнегасник для гасіння пожежі. Робочі місця з персональними комп'ютерами по відношенню до світлових прорізів повинні розташовуватися так, щоб природне світло падало збоку, бажано зліва.

Робочий стіл може бути будь-якої конструкції, що відповідає сучасним вимогам ергономіки і дозволяє зручно розмістити на робочій поверхні обладнання з урахуванням його кількості, розмірів і характеру виконуваної роботи. Доцільно застосування столів, що мають окрему від основної стільниці спеціальну робочу поверхню для розміщення клавіатури. Використовуються робочі столи з регульованим і нерегульованим висотою робочої поверхні.

4.4 Пожежна безпека

У сучасних комп'ютерах дуже висока щільність розміщення елементів електронних систем, в безпосередній близькості один від одного розташовуються сполучні дроти, комунікаційні кабелі, що може посприяти небезпеці різного роду загорянь. При цьому можливі оплавлення ізоляції сполучних проводів, їх оголення і, як наслідок, коротке замикання, яке супроводжується іскрінням, яке веде до неприпустимих перевантажень елементів електронних схем. При перенагріванні, вони згорають з розбризкуванням іскор.

Системи вентиляції служать для відводу надлишкового тепла від ЕОМ. Однак ці системи також представляють додаткову пожежну небезпеку для машинного залу та інших приміщень, так як, з одного боку, вентиляція забезпечує подачу кисню, що є окиснювачем, в усі приміщення, а з іншого – при виникненні пожежі швидко поширюють вогонь і продукти горіння по всіх приміщеннях і пристроїв, з якими вони пов'язані. Струм до електроустановок подається по кабельних лініях, які становлять особливу пожежну небезпеку. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання в вигляді електричних іскор і дуг, розгалуженість і важко доступність роблять кабельні лінії місцями найбільш ймовірного виникнення і розвитку пожежі.

У разі пожежі спрацьовує знаходиться в приміщеннях автоматична установка пожежогасіння (АУП). Найчастіше застосовуються газові АУП вони забезпечені світловою та звуковою сигналізацією. Для запобігання поширенню вогню під час пожежі з однієї частини будівлі на іншу влаштовують протипожежні перепони у вигляді протипожежних стін, перегородок, перекриттів, зон, тамбурів-шлюзів, дверей, вікон, люків, клапанів. У будівлі на випадок виникнення пожежі передбачається не менше двох евакуаційних виходів; але через машинний зал, який має теж менше двох

виходів, не повинні проходити шляхи евакуації співробітників, працюючих в інших підрозділах.

Проходи, коридори і робочі місця не слід захищувати архівними матеріалами, папером. На евакуаційних шляхах встановлюють як природне, так і штучне аварійне освітлення. В інших виробничих приміщеннях допускається проектувати один вихід, якщо відстань від найбільш віддаленого місця до виходу не перевищує 25 м, а кількість працюючих в зміні не більше 25 осіб. Для зберігання носіїв інформації використовуються вогнетривкі металеві шафи, двері в сховище також повинні бути вогнетривкими.

Висновки до розділу 4

Під час виконання спеціальної частини з охорони праці було досліджено санітарно – гігієнічні умови при роботі з комп'ютерною технікою, а саме:

1. Основні вимоги до мікроклімату.
2. Вимоги до освітлення приміщень та робочих місць.
3. Захист користувачів від впливу іонізуючих та неіонізуючих електромагнітних полів та випромінювання моніторів.
4. Вимоги до пожежної безпеки.

Також був проведений розрахунок коефіцієнту природного освітлення у приміщенні. У результаті розрахунків у приміщенні можна запланувати виконання зорової роботи високої точності, що відповідає III розряду зорової роботи, що є задовільними умовами для роботи за комп'ютерною технікою.

Вимоги, які прописані у нормативних документах є актуальними на сьогоднішній день, бо збільшується кількість працівників які працюють за комп'ютерами.

Вимоги та нормативи, щодо охорони праці мають дотримуватися на підприємствах під час трудової діяльності людини. Дотримання поставлених вимог до працівників та власників підприємств дозволить мінімізувати шкідливі наслідки, які мають вплив на здоров'я людини.

ВИСНОВКИ

В умовах поширення через мережу Інтернет картографічних веб-сервісах, які дозволяють досліджувати, аналізувати та приймати рішення у численних сценаріях, пов'язаних з картами та транспортними маршрутами існує необхідність час від часу оптимізувати закладені у них процеси. Проведене дослідження дозволяє зробити наступні висновки.

Здійснений аналіз показав, що серед картографічних веб-сервісів та окремо їх інструментів, існує великий обсяг можливостей при вирішенні нетривіальних задач. Виявлено, що сервіси мають широкий набір інструментів для перегляду знімків карт, побудови маршрутів з різноманітними параметрами, прогнозування заторів тощо. У більшості серверів відсутні такі розвинуті системи прийняття рішень або планування транспортних маршрутів або існують проблеми з підрахунками. У таких умовах з'являється можливість до оптимізації у численних сферах, зокрема прогнозування часу проходження та прийняття рішення в умовах невизначеності.

Серед інструментальних засобів розробки інформаційної системи для розробки системи було обрано мову програмування C#, платформу .NET, яка має можливість використання хмарних технологій, функціонал для кросплатформеності і модульності. Для виконання запитів та отримання відповідей було обрано картографічний прикладний програмний інтерфейс Bing Maps API. У якості інтегрованого середовища розробки було обрано Visual Studio.

У результаті проведеного дослідження було спроектовано та реалізовано програмне забезпечення, яке повністю задовольняє поставлені до нього завдання. Успішно реалізовані функції відправлення, отримання та обробки запитів до прикладного програмного інтерфейсу Bing Maps API. Згідно із зібраною статистикою та її аналізом, програма показує себе дуже добре для більшості звичайних запитів. Реалізована система прийняття рішення для

кращого альтернативного рішення для часу відправлення за окремим маршрутом. Користувач має змогу оцінити критерії за власним розсудом та додати особливі для конкретного часового інтервалу. Також має можливість візуально подивитись на маршрут за допомогою карти з маршрутом.

У спеціальному розділі було проаналізовано санітарно-гігієнічні вимоги приміщення для роботи з комп'ютерною технікою.

Поставлені завдання виконані повністю. Однак, існують побічні проблеми пов'язані з подальшим зменшенням похибки обрахування під час оптимізації у нетривіальних випадках користування та покращення обробки запитів для збільшення швидкодії розробленої програми під час трудомістких запитів. ПЗ потенційно може бути успішно розширене додатковими критеріями до альтернативних варіантів для більш гнучкого налаштування користувацьких сценаріїв пошуку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кондратенко Ю. П. Оптимізація процесів прийняття рішень в умовах невизначеності – Миколаїв: МДГУ, 2006.
2. Decision theory article. *Вікіпедія* : вебсайт. URL: https://en.wikipedia.org/wiki/Decision_theory (дата звернення: 04.11.2021).
3. Minimax criterion. *Ceopedia* : вебсайт. URL: https://ceopedia.org/index.php/Minimax_criterion (дата звернення: 04.24.2021).
4. UNIT – V : DECISION THEORY. *Gurunanakcollege* : вебсайт. URL: <https://gurunanakcollege.edu.in/files/commerce-management/STATISTICS-UNIT-5.pdf> (дата звернення: 04.25.2021).
5. Кондратенко Ю. П. «Теорія прийняття рішень». Методичні вказівки до лабораторних робіт.
6. Бланк И.А. «Принятие решений в условиях риска и неопределенности» – «Элитариум», 2016.
7. How Does Google Maps Know About Traffic Conditions. *ScienceABC*: вебсайт. URL: <https://www.scienceabc.com/innovation/how-does-google-maps-know-about-traffic-conditions.html> (дата звернення: 04.26.2021).
8. 6 методов принятия решений в условиях неопределенности. *Книги Коротко* : вебсайт. URL: <https://www.knigikratko.ru/articles/6-metodov-prinyatiya-reshenij-v-usloviyah-neopredelennosti> (дата звернення: 04.22.2021).
9. Катренко А.В., Пасічник В.В. «Теорія прийняття рішень» – Київ: БХВ, 2009.
10. Зайченко Ю.П. Теорія прийняття рішень – Дніпропетровськ: НГУ, 2015.
11. Ситник В.Ф. Системи підтримки прийняття рішень – Київ: КНЕУ, 2009.
12. Борисов А.Н. Модели принятия решений на основе лингвистической переменной – Рига: Зинатне, 1982.
13. Зайченко Ю.П. Дослідження операцій – Київ, 2000.
14. Ліцензування для сервісу Bing Maps API. *Microsoft* : вебсайт. URL: <https://www.microsoft.com/en-us/maps/licensing/licensing-options> (дата звернення: 04.27.2021).
15. Bing Maps Geographic Coverage. *Microsoft* : вебсайт. URL: <https://docs.microsoft.com/en-us/bingmaps/coverage/geographic-coverage> (дата звернення: 04.23.2021).
16. Інтегроване середовище розробки – Visual Studio. *WebForMySelf* : вебсайт. URL: <https://webformyself.com/sreda-dlya-veb-razrabotki-visual-studio/> (дата звернення: 04.10.2021).
17. Преимущества языка C#. ITVDN FORUM : вебсайт. URL: <https://forum.itvdn.com/t/urok-1-preimushhestva-yazyka-c/2941> (дата звернення: 04.10.2021).

18. Законодавство України про охорону праці // Збірник нормативних документів у 4 т. К. : Держнагляд охорони праці: Основа, 2006.
19. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: <http://zakon3.rada.gov.ua/laws/show/z0508-18> (дата звернення: 11.05.2022).
20. ДСТУ 2293-99 Охорона праці. Терміни та визначення основних понять. Київ, 1999.
21. Москальова В. М. Основи охорони праці. Підручник. - Київ: ВД Професіонал, 2005. 666 с.
22. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред. Гандзюка М. П. - К.: Каравела 2003. 405 с.
23. Ткачук К. Н., Халімовський М. О., Зацарний В.В., та інші. Основи охорони праці: Підручник. -К.: Основа, 2006. 444 с.
24. Жидецький В.Ц. Основи охорои праці: Підручник. - К.: Основа, 2002. 320 с.
25. Державні санітарні правила і норми роботи з ВДТ ЕОМ ДСанПІН 3.3.2.007-98. URL: <http://mozdocs.kiev.ua/view.php?id=2445> (дата звернення: 13.04.2022).
26. СНиП II-4-79. Природне і штучне освітлення. URL: https://dnaop.com/html/45036/doc-СНиП_II-4-79 (дата звернення: 13.05.2022)..
27. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. URL: https://dnaop.com/html/34094/doc-ДСН_3.3.6.042-99 (дата звернення: 23.05.2022).
28. СН 3044-84. Санітарні норми вібрації робочих місць. URL: https://dnaop.com/html/43248/doc_ДНАОП_3044-84 (дата звернення: 10.05.2022).

ДОДАТОК А

Лістинг код нормалізації оцінок до різних умов

```

public AltVariantsCollection Normalize(int intervalMin, double min = 0, double max = 100)
{
    double evaluationDepartureTimeMin= double.MaxValue;
    double evaluationTravelTimeMin= double.MaxValue;
    double evaluationCountryChangeMin = double.MaxValue;

    double evaluationDepartureTimeMax = double.MinValue;
    double evaluationTravelTimeMax = double.MinValue;
    double evaluationCountryChangeMax = double.MinValue;

    foreach (AlternativeVariant altVar in this)
    {
        if (altVar.EvaluationDepartureTime < evaluationDepartureTimeMin)
            evaluationDepartureTimeMin = altVar.EvaluationDepartureTime;
        if (altVar.EvaluationTravelTime < evaluationTravelTimeMin)
            evaluationTravelTimeMin = altVar.EvaluationTravelTime;
        if (altVar.EvaluationCountryChange < evaluationCountryChangeMin)
            evaluationCountryChangeMin = altVar.EvaluationCountryChange;

        if (altVar.EvaluationDepartureTime > evaluationDepartureTimeMax)
            evaluationDepartureTimeMax = altVar.EvaluationDepartureTime;
        if (altVar.EvaluationTravelTime > evaluationTravelTimeMax)
            evaluationTravelTimeMax = altVar.EvaluationTravelTime;
        if (altVar.EvaluationCountryChange > evaluationCountryChangeMax)
            evaluationCountryChangeMax = altVar.EvaluationCountryChange;
    }

    foreach (AlternativeVariant altVar in this)
    {
        if (evaluationTravelTimeMax != 0) altVar.EvaluationDelayTime =
            altVar.EvaluationTravelTime - evaluationTravelTimeMin;
    }
    double evaluationDelayTimeMax = double.MinValue;

    foreach (AlternativeVariant altVar in this)
    {
        if (altVar.EvaluationDelayTime > evaluationDelayTimeMax)
            evaluationDelayTimeMax = altVar.EvaluationDelayTime;
    }

    foreach (AlternativeVariant altVar in this)
    {
        if (evaluationDepartureTimeMax!=0)
        {
            altVar.EvaluationDepartureTime *= (max / evaluationDepartureTimeMax);
        }
    }
}

```

```
        altVar.EvaluationDeparuteTime = Math.Round( max -
altVar.EvaluationDeparuteTime,2);
    }
    if (evaluationDelayTimeMax != 0)
    {
        altVar.EvaluationDelayTime *= (max / evaluationDelayTimeMax);
        altVar.EvaluationDelayTime = Math.Round(max - altVar.EvaluationDelayTime,2);
    }
    if (evaluationCoutryChangeMax!=0)
    {
        altVar.EvaluationCoutryChange *= (max / evaluationCoutryChangeMax);
        altVar.EvaluationCoutryChange = Math.Round(max -
altVar.EvaluationCoutryChange,2);
    }
    }
    return this;
    }
```

ДОДАТОК Б

Лістинг код оптимізації розрахунку часу проходження транспортного маршруту

```
class RouteOptimization
{
    public Response Optimize(Response response, MainForm mainForm)
    {
        Route route = response.ResourceSets.ResourceSet.Resources.Route;
        Response responseRaw;
        mainForm.UpdateElapsedTime();

        string wp0;
        string wp1;
        string key = "ApNf4cdMo33Rss3h5mOQtQYIYGesonbD4PatMfaq8-9RPSQ-
orq8vnk31MuEcMx9";
        route.travelDurationTraffic = new TimeSpan();
        for (int i = 0; i < route.routeLeg.itineraryItems.Count-1; i++)
        {
            wp0 = $"{route.routeLeg.itineraryItems[i].maneuverPoint.Latitude},
{route.routeLeg.itineraryItems[i].maneuverPoint.Longitude}";
            wp1 = $"{route.routeLeg.itineraryItems[i + 1].maneuverPoint.Latitude},
{route.routeLeg.itineraryItems[i+1].maneuverPoint.Longitude}";
            DateTime dateTimeDepartureTemp;
            if (i == 0)
            {
                dateTimeDepartureTemp = route.departureTime;
            }
            else
            {
                dateTimeDepartureTemp = route.routeLeg.itineraryItems[i - 1].localDepartureTime
                + TimeSpan.FromSeconds(route.routeLeg.itineraryItems[i - 1].travelDuration);
            }
            route.routeLeg.itineraryItems[i].localDepartureTime = dateTimeDepartureTemp;

            string dateTimeTempStr =
            route.routeLeg.itineraryItems[i].localDepartureTime.ToString("yyyy/MM/dd'%20H':%mm':%ss")
            ;

            var url = $"https://dev.virtualearth.net/REST/V1/Routes/Driving" +
                $"?wp.0={wp0}" +
                $"&wp.1={wp1}" +
                $"&optmz=timeWithTraffic" +
                $"&timeType=Departure" +
                $"&dateTime={dateTimeTempStr}" +
                $"&output=xml" +
                $"&key={key}";
        }
    }
}
```

```

Console.WriteLine($"#{i} Getting response...");
XmlElement xRoot = ResponseHandler.GetResponse(url);
responseRaw = ResponseHandler.ReadResponse(xRoot, dateTimeDepartureTemp);

responseRaw.ResourceSets.ResourceSet.Resources.Route.routeLeg.itineraryItems[0].localDepartureTime = dateTimeDepartureTemp;
    route.travelDurationTraffic +=
responseRaw.ResourceSets.ResourceSet.Resources.Route.travelDurationTraffic;

    string travelDurationTrafficStr = string.Format("{0:D2}h:{1:D2}m:{2:D2}s",
        route.travelDurationTraffic.Hours,
        route.travelDurationTraffic.Minutes,
        route.travelDurationTraffic.Seconds);
    route.travelDurationTrafficStr = travelDurationTrafficStr;

    route.routeLeg.itineraryItems[i].localDepartureTime = dateTimeDepartureTemp;
    route.routeLeg.itineraryItems[i].travelDurationTraffic =
Convert.ToInt32(responseRaw.ResourceSets.ResourceSet.Resources.Route.travelDurationTraffic.TotalSeconds);
    if (i == route.routeLeg.itineraryItems.Count - 2)
        mainForm.UpdateElapsedTime();
    }
return response;
}
}

```

ДОДАТОК В

Покриття сервісу Bing Maps API

Країна/Регіон	Дорожні дані / Маршрутизація	Трафік	Маршрутизація вантажівок	Геокодування
Андорра	добре	x	x	Дах
Австралія	добре	x	x	Дах
Австрія	добре	x	x	Дах
Аргентина	добре	x	x	Дах
Бахрейн	добре	x		Дах
Бельгія	добре	x	x	Дах
Болгарія	добре	x	x	Дах
Бразилія	добре	x	x	Дах
Бруней	добре	x		Дах
Ватикан	добре	x	x	Назва вулиці
В'єтнам	добре	x		Дах
Гернсі	добре	x		Адреса
Гібралтар	добре	x		Адреса
Греція	добре	x	x	Дах
Данія	добре	x	x	Дах
Джерсі	добре	x		Адреса
Естонія	добре	x	x	Дах
Єгипет	добре	x		Дах
Ізраїль	добре	x		Дах
Індія	добре	x		Дах
Індонезія	добре	x		Дах
Ірландія	добре	x	x	Дах
Ісландія	добре	x		Дах
Іспанія	добре	x	x	Дах
Італія	добре	x	x	Дах
Канада	добре	x	x	Дах
Катар	добре	x		Дах

Кафедра інтелектуальних інформаційних систем
Система планування та оптимізації транспортних маршрутів в умовах невизначеності

Кенія	добре	x		Назва вулиці
Китай	добре	x		Адреса
Колумбія	добре	x	x	Дах
Корея	добре	x		Адреса
Кувейт	добре	x		Дах
Латвія	добре	x	x	Адреса
Лесото	добре	x		Назва вулиці
Литва	добре	x	x	Дах
Ліхтенштейн	добре	x	x	Дах
Люксембург	добре	x	x	Дах
Малайзія	добре	x		Дах
Мальта	добре	x	x	Адреса
Марокко	добре	x		Дах
Мексика	добре	x	x	Дах
Мозамбік	добре	x		Назва вулиці
Монако	добре	x	x	Адреса
Нігерія	добре	x		Адреса
Нідерланди	добре	x	x	Дах
Німеччина	добре	x	x	Дах
Нова Зеландія	добре	x	x	Дах
Норвегія	добре	x	x	Дах
Об'єднане Королівство	добре	x	x	Дах
Об'єднані Арабські Емірати	добре	x		Дах
Оман	добре	x		Назва вулиці
Острів Мен	добре	x		Адреса
ПАР Гонконг	добре	x		Дах
Перу	добре	x		Адреса
Південна Африка	добре	x		Дах
Польща	добре	x	x	Дах
Португалія	добре	x	x	Дах

Кафедра інтелектуальних інформаційних систем
Система планування та оптимізації транспортних маршрутів в умовах невизначеності

Пуерто-Рико	добре	x		Дах
Росія	добре	x	x	Дах
Румунія	добре	x	x	Адреса
Сан-Марино	добре	x	x	Дах
Саудівська Аравія	добре	x		Адреса
Сінгапур	добре	x		Дах
Словаччина	добре	x	x	Дах
Словенія	добре	x	x	Дах
Сполучені Штати	добре	x	x	Дах
Таїланд	добре	x		Дах
Тайвань	добре	x		Дах
Туреччина	добре	x	x	Дах
Угорщина	добре	x	x	Дах
Україна	добре	x		Дах
Уругвай	добре	x	x	Дах
Філіппіни	добре	x		Дах
Фінляндія	добре	x	x	Дах
Франція	добре	x	x	Дах
Хорватія	добре	x	x	Дах
Чеська Республіка	добре	x	x	Дах
Чилі	добре	x	x	Дах
Швейцарія	добре	x	x	Дах
Швеція	добре	x	x	Адреса
Японія	добре	x		Дах