

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

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

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи та
технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система маршрутизації БПЛА. Підсистема
складання маршрутів за допомогою
алгоритму табу– пошуку»**

Виконав:

студент IV курсу, групи ІС-92

Чорний Олександр Олександрович _____

Керівник:

доцент, к.т.н., доцент

Жданова Олена Григорівна _____

Рецензент:

доцент каф. ОТ, с.н.с., доцент

Антонюк Андрій Іванович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2023 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Чорному Олександрю Олександровичу

1. Тема проєкту «Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів за допомогою алгоритму табу-пошуку», керівник проєкту Жданова Олена Григорівна, к.т.н., доцент, затверджені наказом по університету від «31» травня 2023 р. №2101-с.
2. Термін подання студентом проєкту: 12.06.2023
3. Вихідні дані до проєкту: мова програмування TypeScript, мова програмування JavaScript, середовище розробки Visual Studio Code, бібліотеки Node.js, Nest.js.
4. Зміст пояснювальної записки: загальні положення, інформаційне забезпечення, математичне забезпечення, програмне і технічне забезпечення, випробування програмного продукту.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):
 - ER-діаграма бази даних;
 - схема структурна компонентів;

- схема структурна послідовності розв’язання задачі;
- схема структурна послідовності проведення експерименту;
- схема структурна алгоритму табу-пошуку.

6. Дата видачі завдання 13.02.2023

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз завдання дипломного проєкту	24.04.2023	
2	Аналіз предметної області, огляд існуючих аналогів, розробка функціональної моделі	26.04.2023	
3	Розробка математичного забезпечення	01.05.2023	
4	Опис інформаційного забезпечення, розробка програмного забезпечення	04.05.2023	
5	Розробка програмного забезпечення	08.05.2023	
6	Опис програмного та технічного забезпечення	10.05.2023	
7	Оформлення пояснювальної записки	17.05.2023	
8	Подання ДП на попередній захист	06.06.2023	
9	Подання ДП на основний захист	12.06.2023	

Студент

Олександр ЧОРНИЙ

Керівник

Олена ЖДАНОВА

АНОТАЦІЯ

Чорний О.О. Система маршрутизації БПЛА. Підсистема складання маршрутів за допомогою алгоритму табу-пошуку. КПІ ім. Ігоря Сікорського, Київ, 2023.

Пояснювальна записка дипломного проекту складається з п'яти розділів, а також 4 рисунки, 15 таблиць, 5 графічних матеріалів, 21 джерел.

Ключові слова: безпілотний літальний апарат, задача маршрутизації транспортних засобів, метод табу-пошуку, складання плану відвідування пунктів, стохастичний метод.

Об'єктом розробки є серверна частина системи маршрутизації БПЛА.

Мета розробки – складання ефективного маршруту, який зменшує час проходження маршруту, враховуючи перезарядку БПЛА.

Дипломний проект присвячений розробці серверної сторони комплексної системи маршрутизації БПЛА, а також проектуванню, розробці та аналізу методу табу-пошуку для розв'язання задачі маршрутизації БПЛА. Для розробленого алгоритму табу-пошуку проведено дослідження по підборі параметрів, що показують прийнятні результати для задачі маршрутизації БПЛА. З підібраними параметрами проведено дослідження роботи алгоритму, що показало задовільні результати.

Отримані результати можна використовувати для досягнення аналогічної мети при автоматизації процесів побудови маршрутів для подібних або схожих об'єктів.

SUMMARY

Chorny O.O. UAV routing system. Subsystem of route planning using the tabu search algorithm. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The explanatory note of the diploma project consists of five chapters, as well as 4 figures, 15 tables, 5 graphic materials, 21 sources.

Keywords: planning of the points visit, stochastic method, tabu search method, unmanned aerial vehicle, vehicle routing problem.

The object of development: a server side of UAV routing system.

The goal of development: to create an efficient route that reduces route travel time, taking into account UAV recharge.

The project is devoted to the development of the server side of an integrated UAV routing system, as well as the design, development, and analysis of a tabu search method for solving the UAV routing problem. For the developed tabu search algorithm, a study was conducted to select parameters that show good results for the UAV routing problem. With the selected parameters, the algorithm was studied, which showed effective results.

The obtained results can be used to achieve a similar goal when automating the processes of building routes on similar or similar objects.

Номер рядка	Формат	Позначення	Найменування	Кількість аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC92.220БАК.004 ПЗ	Пояснювальна записка	84		
6	A3	IC92.220БАК.004 Д1	ER-діаграма бази даних	1		
7	A3	IC92.220БАК.004 Д2	Схема структурна	1		
8			компонентів			
9	A3	IC92.220БАК.004 Д3	Схема структурна	1		
10			послідовності розв'язання			
11			задачі			
12	A3	IC92.220БАК.004 Д4	Схема структурна	1		
13			послідовності проведення			
14			експерименту			
15	A3	IC92.220БАК.004 Д5	Схема структурна алгоритму	1		
16			табу-пошуку			
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						

					IC92.220БАК.004 ТП			
Зм.	Аркуш	№ докум.	Підп.	Дата				
Розроб.		Чорний О.О.			Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів за допомогою методу табу-пошуку. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Жданова О.Г.				т	1	1
						КПІ ім. Ігоря Сікорського Група IC-92		
Затв.								

Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система маршрутизації
БПЛА. Підсистема складання маршрутів за
допомогою алгоритму табу-пошуку»

Київ – 2023 року

ЗМІСТ

ВСТУП.....	4
Н У 1.1 Опис предметного середовища.....	6
Р 1.1.1 Опис функціональної моделі	8
Е 1.2 Огляд наявних аналогів	9
Л 1.3 Постановка задачі.....	11
І 1.3.1 Призначення розробки	11
К 1.3.2 Цілі та задачі розробки	11
\ Висновок до розділу	11
Н 2.1 Вхідні дані.....	13
У 2.2 Вихідні дані.....	13
Р 2.3 Опис структури файлів.....	13
Б 2.4 Опис структури бази даних.....	15
В Висновок до розділу	18
М МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	19
У 3.1 Змістовна постановка задачі	19
4 3.2 Математична постановка задачі	20
5 3.3 Обґрунтування методу розв'язання	23
Т 3.3.1 Класифікація задач транспортної маршрутизації.....	23
Ф 3.3.2 Аналіз методів розв'язання задач оптимізації	26
с ЗАГАЛЬНЕ ПОЛОЖЕННЯ для розв'язання VRP	34
1 3.4 Опис методу розв'язання.....	38
3 3.4.1 Підбір параметрів для розробленого алгоритму	40
1	
4	
6	

З					<i>IS92.220BAK.004 ПЗ</i>		
"							
Зм.	Лист	№ докум.	Підпис				
Розробив	Чорний О.О.			<i>Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів за допомогою алгоритму</i>	Літ.	Арк.	Аркушів
Перевірив	Жданова О.Г.				Т	2	84
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ <i>табу-лашуку</i>					13		
Пояснювальна записка					КІП ім. Ігоря Сікорського		
Затв.							

3.4.2	Результати роботи розробленого алгоритму.....	45
	Висновок до розділу	47
4	ПРОГРАМНЕ І ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	49
4.1	Засоби розробки	49
4.1.1	Програмне середовище розробки системи.....	49
4.1.2	Мова розробки системи.....	50
4.1.3	Фреймворки використані для розробки системи.....	52
4.1.4	Бібліотеки використані для розробки системи	55
4.2	Вимоги до технічного забезпечення	56
4.3	Архітектура програмного забезпечення	57
4.3.1	Діаграма компонентів.....	57
4.3.2	Діаграма послідовності	58
4.3.4	Опис класів, функцій та методів програмного забезпечення.....	59
	Висновок до розділу	66
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	67
5.1	Випробування програмного продукту	67
5.1.1	Мета випробувань	67
5.1.2	Результати випробувань	67
	Висновок до розділу	78
	ВИСНОВКИ.....	80
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82

ВСТУП

В XXI столітті така технологія як безпілотні літальні апарати знайшла своє місце і застосування у дуже широкому спектрі задач, який постійно тільки збільшується.

Сьогодні БПЛА є глибоко інтегрованими у такі галузі, як військова справа, агрономія, проведення наукових досліджень, рекреація, проведення пошукових операцій, створення фото і відеоматеріалів, спорт тощо. З'явилися навіть принципово нові види заходів пов'язані безпосередньо із безпілотними літальними апаратами. Для прикладу – так звані “Шоу дронів”, під час яких велика кількість квадрокоптерів з підсвіткою в темний час доби синхронно рухаються, створюючи в небі різноманітні узори, написи тощо.

Крім того існує безліч експериментальних проєктів і заходів, покликаних дослідити ефективність використання безпілотних літальних апаратів у тих галузях, у яких їх використання ще не здобуло такого широкого розповсюдження. Серед них можна виділити комерційні доставки, патрулювання вулиць та переслідування вуличних злочинців силовими структурами, а також так звані “аеротаксі”, які здатні, оминаючи затори на дорогах, швидко доставляти людей у спеціально підготовлені для їх посадки місця, такі як аеропорти.

Переваги безпілотників, такі як здатність діяти в небезпечних умовах без ризику для людей, високий рівень автономності і мобільності роблять їх незамінною технологією, яка приносить користь і покращує багатьом людям життя.

Але, як і будь-яку технологію, безпілотні літальні апарати мають свій простір для покращень. Інколи їх можна покращити змінюючи дизайн, конструкцію або її складники. Інколи можна покращити програмне забезпечення, яке керує БПЛА. А іноді – можна покращити сам спосіб використання даної технології.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

Побудова ефективних маршрутів для безпілотних літальних апаратів – це один зі способів збільшити ефективність і користь даної технології. І якщо у деяких сферах застосування дронів ефект недостатньо ефективного маршруту обмежується збільшенням фінансових витрат, то в деяких, таких як військова справа або проведення пошуково-рятувальних операцій, від цього можуть залежати життя людей.

Для того щоб БПЛА ефективно виконували свої завдання, потрібно розглянути задачі оптимізації маршрутів БПЛА.

Об'єктом розробки є задачі маршрутизації для безпілотних літальних апаратів.

Предметом розробки є інформаційна система маршрутизації для БПЛА, а саме її серверна частина. Ця система включає в себе алгоритми, які допомагають планувати маршрути для БПЛА з метою досягнення ефективності та безпеки місій.

Метою дипломного проектування є розробка серверної частини інформаційної системи, що складає ефективний маршрут для БПЛА, який зменшує на час проходження маршруту, враховуючи перезарядку БПЛА. Для досягнення мети необхідно розробити серверну частину наступних складових системи:

а) складова дослідження алгоритмів розв'язання задачі:

- 1) генерація вхідних даних;
- 2) проведення експериментів;
- 3) збереження результатів дослідження у файл;
- 4) статистичний аналіз результатів експериментів.

б) складова побудови маршрутів БПЛА методом табу-пошуку;

- 1) ведення вхідних даних;
- 2) розв'язання задачі методом табу-пошуку;
- 3) збереження результатів розв'язання задачі у файл.

									Арк.
									5
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220BAK.004 ПЗ</i>				

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Для дослідження та розв'язання було обрано задачу маршрутизації безпілотних літальних апаратів (БПЛА). Сформульована задача значно відрізняється від класичної за рахунок особливостей роботи БПЛА, і відповідно обмежень що накладаються на задачу. Було обрано задачу маршрутизації транспортних засобів з кільком депо.

БПЛА, на сьогоднішній день є актуальною та поширеною технологією. Вони використовуються в різних сферах, наприклад таких як:

- військові операції – БПЛА використовуються для збору інформації про території та ворожі позиції, а також для виконання наступу або оборони;
- зйомка з висоти – БПЛА можуть здійснювати зйомку з висоти, що дає можливість отримати детальну інформацію про ландшафти, збільшити область покриття зйомки та забезпечити високу якість зображення;
- пошук і рятування – БПЛА використовуються для швидкого та ефективного пошуку втрачених або постраждалих людей та надання їм допомоги;
- дослідження довкілля – БПЛА можуть бути використані для збору даних про стан довкілля, таких як вимірювання рівня забруднення повітря та води, виявлення лісових пожеж та інше;
- землеробство – БПЛА можуть бути використані для моніторингу стану ґрунту та рослин, що дозволяє фермерам виявляти проблеми та забезпечувати ефективне вирощування рослин;
- доставка – БПЛА можуть бути використані для доставки товарів або медичних препаратів в важкодоступні або небезпечні місця;
- рекламні акції та події – БПЛА можуть бути використані для проведення рекламних акцій, де вони можуть надавати панорамні зображення з висоти.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

Однією з головних переваг використання БПЛА є те, що вони можуть легко дістатися до місць, недоступних для людей або традиційних літальних апаратів. Вони також можуть бути використані для виконання завдань, які були б надзвичайно небезпечними для пілотів або інших працівників.

Ще однією перевагою є те, що БПЛА можуть працювати в будь-який час доби, що дозволяє здійснювати моніторинг або розвідку протягом 24 годин на день, що робить їх особливо корисними для військових операцій та інших критичних ситуацій.

Маршрутизація є важливою задачею при використанні БПЛА. Оскільки вони можуть полетіти на дуже великі відстані, важливо забезпечити, щоб вони дісталися до місця призначення в найбільш ефективний спосіб. Це може охоплювати вибір оптимального маршруту, уникнення небезпечних зон, організацію зв'язку зі землею та інші аспекти, які допоможуть БПЛА виконувати свої завдання ефективно та безпечно.

Розглянемо задачу даного дипломного проекту на прикладі предметної області служби порятунку.

Нехай є потреба у проведенні розвідки ряду географічних точок у складнодоступній місцевості для пошуку людини або групи людей, які заблукали. Був підготовлений перелік найбільш ймовірних точок перебування групи людей і необхідно в максимально короткі терміни обстежити усі ці точки з повітря. Оператор БПЛА вводить в систему швидкість руху даної моделі БПЛА, максимальний час польоту БПЛА без необхідності обслуговування (наприклад, заміни батареї), час обслуговування, а також координати точок, які необхідно облетіти та координати двох баз. Базою ми вважаємо точку, у якій можливо провести обслуговування БПЛА між вильотами.

Розбиття всього маршруту БПЛА на окремі вильоти необхідне через те, що час маршруту з урахуванням швидкості БПЛА може перевищувати максимальний час польоту.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

Таким чином БПЛА має робити ряд окремих вильотів, між якими він буде проходити обслуговування на одній із двох баз.

Старт маршруту має відбутися в так званій стартовій базі – в тій, в якій БПЛА знаходиться на момент початку пошукової операції. Кінець маршруту БПЛА може бути у будь-якій із двох баз.

Всі точки мають бути розташовані таким чином, щоб БПЛА міг долетіти до кожної точки із однієї з двох баз і повернутися до однієї з двох баз не перевищуючи свій максимальний польотний час. Базис, в свою чергу, мають бути розташовані таким чином, щоб БПЛА міг долетіти із однієї бази в іншу не перевищуючи максимальний польотний час.

Необхідно побудувати маршрут, який складається з ряду підмаршрутів, кожен з яких починається і закінчується в одній з двох баз (окрім першого підмаршруту, який має починатися виключно в стартовій базі). Причому необхідно мінімізувати сумарний час маршруту. Сумарний час маршруту є сумою часу кожного з підмаршрутів і часу кожної сесії обслуговування на базі між підмаршрутами.

1.1.1 Опис функціональної моделі

У дипломному проєкті реалізована серверна частина. Система повинна виконувати такі функції як:

- зчитування вхідних даних для розв’язання задачі з файлу;
- зчитування введених користувачем вручну даних;
- складання маршруту польоту БПЛА;
- збереження результатів складання маршруту польоту БПЛА у файл;
- збереження результатів експерименту у файл;
- проведення експерименту за заданими параметрами, на автоматично згенерованих даних;

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

– вивід на екран результатів складання маршруту польоту БПЛА для користувача;

– вивід на екран результатів експерименту.

У графічних матеріалах наведена структурна схема варіантів використання системи складання польотів БПЛА.

1.2 Огляд наявних аналогів

Приладом актуального аналогу є DroneDeploy – це платформа для планування та керування польотами безпілотних літальних апаратів (БПЛА), яка дозволяє користувачам легко створювати, запланувати та виконувати місії збору даних.

DroneDeploy – це американська компанія, яка була заснована в 2013 році Майклом Вінтером, Майклом Вайзом та Ніком Паркером. Компанія розробляє платформу для створення карт та отримання геопросторових даних за допомогою безпілотних літальних апаратів [16].

У 2014 році DroneDeploy отримала перший раунд фінансування на суму 2 мільйони доларів від партнерів зі стартапами та інвесторів з сфери технологій. У 2015 році компанія випустила свою першу мобільну програму, яка дозволяла керувати БПЛА та отримувати дані з апарату [16].

У 2016 році компанія представила нову версію своєї платформи, яка дозволяє планувати маршрути для БПЛА з використанням зручного веб-інтерфейсу та отримувати високоякісні зображення з повітря. Також у 2016 році DroneDeploy отримала ще 20 мільйонів доларів інвестицій [16].

У наступні роки DroneDeploy продовжувала розвивати свою платформу та працювати над новими функціями. У 2019 році компанія запустила нову функцію зі створення 3D-моделей зображень з БПЛА. На сьогоднішній день DroneDeploy є однією з провідних компаній у сфері розробки програмного забезпечення для БПЛА та отримання геопросторових даних [16].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		9

Платформа має зручний веб-інтерфейс, що дозволяє користувачам легко створювати маршрути для БПЛА, задавати точки збору даних та налаштовувати параметри польоту для досягнення оптимальної маршрутизації.

За допомогою DroneDeploy можна планувати різноманітні місії, включаючи зйомку з повітря високої роздільної здатності, картографію, 3D-моделювання та інше. Платформа також дозволяє користувачам збирати дані з різних датчиків, таких як RGB-камери, термальні камери та LiDAR, для збору даних про навколишнє середовище [16].

Крім того, DroneDeploy дозволяє користувачам використовувати попередньо побудовані карти та зображення для аналізу даних та отримання цінної інформації. Користувачі можуть експортувати дані з платформи для подальшої обробки та аналізу в інших програмах. Загалом, DroneDeploy є потужним інструментом для збору даних з повітря, який дозволяє користувачам легко планувати та виконувати місії з збору даних, що дозволяє підвищити ефективність та точність збору та зменшити витрати на їх збір.

Основною перевагою системи, що створюються у дипломному проєкті є те, що маршрути створюються автоматично. Система самостійно буде оптимальний маршрут і відповідно впорядковує пункти обльоту. У системі DroneDeploy користувач самостійно планує порядок відвідування пунктів обльоту, що, по-перше, може бути дуже трудозатратним при великій кількості пунктів, а по-друге, не дозволяє ефективно спланувати відвідування цілей.

Також, перевагою проєктованої системи є відкритий доступ до використання, в той час як DroneDeploy це платна система, що не є сприятливою умовою використання для невеликих підприємств та приватних осіб.

Крім того, DroneDeploy має велику кількість функцій і налаштувань, що може бути перешкодою для користувачів з меншим досвідом або новачків. Оскільки програма має великий обсяг можливостей, потрібен час для повного ознайомлення з усіма її можливостями.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розробки є створення плану відвідування пунктів об'льоту БПЛА, за заданими координатами баз та пунктів об'льоту так, щоб для кожного підмаршруту БПЛА вистачило заряду для повернення на одну з баз та перезарядки. При цьому, обліт пунктів має потребувати мінімальних ресурсів, заряду та часу БПЛА.

1.3.2 Цілі та задачі розробки

Метою дипломного проектування є розробка серверної частини інформаційної системи, що складає ефективний маршрут для БПЛА, який зменшує на час проходження маршруту, враховуючи перезарядку БПЛА. Для досягнення мети необхідно розробити серверну частину наступних складових системи:

- а) складова дослідження алгоритмів розв'язання задачі:
 - 1) генерація вхідних даних;
 - 2) проведення експериментів;
 - 3) збереження результатів дослідження у файл;
 - 4) статистичний аналіз результатів експериментів.
- б) складова побудови маршрутів БПЛА методом табу-пошуку;
 - 1) ведення вхідних даних;
 - 2) розв'язання задачі методом табу-пошуку;
 - 3) збереження результатів розв'язання задачі у файл.

Висновок до розділу

У даному розділі було описане предметне середовище, а саме маршрутизацію транспортних засобів та застосування БПЛА. Були описані

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

процеси взаємодії та співвідношення ролей користувачів з проєктованою системою, побудовані діаграми процесів діяльності. Також було описано функції проєктованої системи, схематично проілюстровано варіанти взаємодії цих функцій між собою та з користувачем діаграмою варіантів використання. Проаналізовано систему для підтримки маршрутизації БПЛА DroneDeploy, визначено її недоліки порівняно з проєктованою системою. Також описано призначення, цілі та задачі розробки.

					<i>IS92.220БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані для системи є інформацією, яка надходить у систему зовнішніми джерелами і використовується для обробки, аналізу, зберігання або виведення результатів. Вони можуть бути надані користувачами або зчитаними з файлів.

Дані про БПЛА:

- середня швидкість польоту;
- максимальний час автономної роботи;
- час, що потрібен для зарядки.

Дані про пункти обльоту:

- географічні координати (довгота та широта).

Дані про бази:

- географічні координати (довгота та широта).

2.2 Вихідні дані

Вихідні результати цієї системи включають графічну план-схему, що відображає розподілення маршрутів між базами та пунктами обльоту у зручній для користувача графічній формі, а також звіт, що містить загальний час обльоту для всіх маршрутів. Також, результати роботи системи можна за потребою користувача зберегти у файл.

2.3 Опис структури файлів

JSON (JavaScript Object Notation) – це простий формат обміну даними, що широко використовується для структурованого представлення інформації. Він використовується для передачі даних між різними компонентами програмного забезпечення.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

Файли JSON мають розширення .json і містять інформацію у вигляді тексту, організовану у вигляді пар "ключ-значення". У JSON можуть зустрічатись строкові значення, числа, об'єкти, масиви, булеві значення або значення null.

Ми можемо зчитувати та зберігати JSON файли.

Вхідний JSON для розв'язання задачі із заданими вхідними даними користувача має наступні поля:

- points – масив координат із метаданими пунктів обльоту;
- startBase – координати бази з якої ми починаємо маршрут;
- anotherBase – координати бази, в якій ми закінчуємо маршрут або перезаряджаємось;
- speed – середня швидкість польоту БПЛА;
- maxFlightTime – максимальний час автономної роботи;
- chargeTime – час, що потрібен для зарядки.

Вихідний JSON має наступні поля:

- route – маршрут, який складається з координат пунктів обльотів та трьох видів бази, а саме: початкової, бази для перезарядки та кінцевої;
- fitness – значення цільової функції.

Вихідний JSON для проведення експерименту над алгоритмами має наступні поля:

- numberOfPoints – розмірність задачі (кількість випадково згенерованих точок);
- numberOfRuns – кількість запусків алгоритму;
- algorithm – назва алгоритму, над яким проводився експеримент;
- mean – середнє значення сумарного часу обльоту пунктів із урахування часу зарядки;
- standardDeviation – стандартне відхилення сумарного часу обльоту пунктів із урахування часу зарядки.

									Арк.
									14
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220BAK.004 ПЗ</i>				

fitness	DOUBLE PRECISION	Значення цільової функції
created_at	TIMESTAMP	Час створення розв'язку у системі

Таблиця 2.3 – Структура таблиці points (бази та пункти об'їзду)

Назва поля	Тип даних	Опис поля
id	SERIAL	Унікальний ідентифікатор бази або пункту об'їзду
domain_id	INT	Унікальний в межах задачі ідентифікатор бази або пункту об'їзду
solutions_id	INT	Унікальний ідентифікатор розв'язку, до якого належить база або пункт об'їзду
lat	DOUBLE	Широта координати знаходження бази або пункт об'їзду
lng	DOUBLE	Довгота координати знаходження бази або пункт об'їзду
isBase	BOOLEAN	Булевий ідентифікатор того, чи є точка базою
isStartBase	BOOLEAN	Булевий ідентифікатор того, чи є точка першою базою (тобто такою, з якої розпочинається маршрут)
label	text	Текстова назва на точці, для відображення навігації на користувачському інтерфейсі

sequence_number	INT	Порядковий номер бази або пункту обльоту у розв'язку
created_at	TIMESTAMP	Час створення точки у системі

Таблиця 2.4 – Структура таблиці experiments(експерименти)

Назва поля	Тип даних	Опис поля
id	SERIAL	Унікальний ідентифікатор розв'язку задачі
user_id	INT	Ідентифікатор користувача
numberOfPoints	INT	Кількість пунктів обльотів, які повинен облетіти БПЛА
numberOfRuns	INT	Кількість прогонів задачі
algorithm	VARCHAR	Алгоритм, яким ми розв'язуємо задачу
params	JSONB	Параметри алгоритму
mean	DOUBLE	Середнє значення цільової функції
standardDeviation	DOUBLE	Середнє квадратичне відхилення
created_at	TIMESTAMP	Час створення експерименту у системі

У графічних матеріалах до дипломного проєкту представлена діаграма бази даних сутність-зв'язок, що графічно ілюструє її структуру. А саме, таблиці бази даних (тобто сутності), їх поля (тобто атрибути), а також взаємозв'язки між таблицями, ступінь яких вказується типом кінцівки зв'язку.

Висновок до розділу

У даному розділі було описане інформаційне забезпечення системи, а саме набір, структуру та представлення вхідних та вихідних даних для роботи з системою. Описані структури файлів формату JSON: файл з вхідними даними для розв'язання задачі, файл з вихідними даними розв'язання задачі та файл з вихідними даними експерименту. Також розроблена ER– діаграма бази даних.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Задача, що досліджуватиметься, має певні особливості постановки.

По-перше, класифікуємо її як задачу транспортної маршрутизації з кількома базами, оскільки маємо певну кількість баз, а саме 2. Шлях БПЛА розпочинається у конкретній базі та завершується у будь-якій базі. Для підзарядки між маршрутами також можна використовувати будь-яку базу. Бази облітаються у довільній послідовності без пріоритету та кожен пункт обльоту повинен бути відвіданим БПЛА один та тільки один раз.

По-друге, на відміну від класичної постановки задачі, що розглядалася у попередніх розділах, цілі просто облітаються, без доставки вантажу. Тобто, відсутні значення обсягу товару, який потрібно доставити до кожного з пунктів обльоту. Відповідно, немає обмеження на вантажопідйомність БПЛА.

Але, незважаючи на спрощення постановки зі скасуванням обмежень та обсяги товару, що перевозиться, є строге обмеження на довжину кожного з маршрутів, вона не повинна перевищувати ту, на яку у БПЛА вистачає ресурсу, оскільки транспортний засіб повинен повертатися на одну з баз для заміни батареї. Метою буде мінімізація часу, витраченого на відвідування усіх пунктів обльоту, враховуючи, що через обмеження на максимальну довжину маршруту, загальний план відвідування пунктів обльоту розбиватиметься на підмаршрути, і між ними буде перезарядка БПЛА, що також вимагає певного часу. Відповідно, задля досягнення цілі мінімізації загального витраченого часу, необхідно мінімізувати кількість маршрутів, щоб довелося виконувати менше перезарядок між ними, та мінімізувати відстані, пройдені між цілями у маршрутах. Схематично, структура розв'язку представлена на рисунку 3.1.

					<i>IC92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

Необхідно знайти план обльоту цілей, який складається з Q маршрутів (Q є невідомою на початку величиною, тобто такою що визначається у процесі розв'язання). При цьому кожен з маршрутів починається та закінчується у одній з двох баз P (але перший маршрут розпочинається в базі p_1).

Отже, необхідно розбити множину пунктів обльоту K на Q впорядкованих підмножин:

$$\{K_1 = \{k_0^1, k_1^1 \dots k_{n_1-1}^1\}, \dots, K_Q = \{k_0^Q, k_1^Q \dots k_{n_Q-1}^Q\}\},$$

таких що:

$$\cup_{q=1}^Q K_q = K \quad (\sum_{q=1}^Q (n-1)_q = (n-1))$$

$$K_i \cap K_j = \emptyset, \text{ де } i \neq j$$

Наведені вище умови забезпечують те, що кожен пункт обльоту буде включено у маршрут, але лише у один маршрут плану обльотів і лише один раз, при цьому:

$K_1 = \{k_0^1, k_1^1 \dots k_{n_1-1}^1\}$ – пункти обльоту що належать першому з побудованих маршрутів;

...

$K_Q = \{k_0^Q, k_1^Q \dots k_{n_Q-1}^Q\}$ – пункти обльоту що належать останньому з побудованих маршрутів Q .

При тому, що кожна з підмножин пунктів обльоту не містить баз. Підмножини пунктів обльоту розділяються відвідуванням баз. Перед першою підмножиною пунктів обльоту БПЛА вилітає з першої бази. Далі, після відвідування кожної з підмножин пунктів обльоту, БПЛА перезаряджається на першій або другій базі. Проходження усіх підмножин пунктів обльоту завершується на другій або першій базі.

Позначимо через T_q – сумарний час проходження маршруту q ($q = 1, \dots, Q$):

									Арк.
									21
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220BAK.004 ПЗ</i>				

$$T_1 = t_{p_1, k_0^1} + \sum_{i=0}^{n_1-1} t_{k_i^1, k_{i+1}^1} + t_{k_{n_1-1, P}^1}, P \in \{p_1, p_2\} \quad (3.1)$$

$$T_q = t_{P, k_0^q} + \sum_{i=0}^{n_q-1} t_{k_i^q, k_{i+1}^q} + t_{k_{n_q-1, P}^q}, q = 2, \dots, Q, P \in \{p_1, p_2\}. \quad (3.2)$$

Відштовхуючись від змістовної постановки задачі, виділено цільову функцію. Мінімізація сумарного часу, витраченого на обліт усіх пунктів (тобто пунктів обльоту та баз) побудованих маршрутів плану обльоту, та на перезарядки між маршрутами:

$$z = (Q - 1)R + \sum_{q=1}^Q T_q \rightarrow \min. \quad (3.3)$$

Ця цільова функція враховує час, що потрібен на перезарядку та кількість перезарядок, а відповідно і кількість маршрутів у розв'язку та мінімізації кількості маршрутів, що входять до плану обльоту.

Сумарний час відвідування усіх пунктів (пунктів обльоту, з завершенням у одній з баз) кожного з маршрутів не повинен перевищувати максимальний час автономної роботи БПЛА:

$$T_q \leq M, \text{ де } q = 1, \dots, Q, \quad (3.4)$$

де

$$T_q = \sum_{k=0}^{n+1_q} \sum_{b=0}^{n+1_q} t_{kb} x_{kbq}. \quad (3.5)$$

Обмеження, що гарантує, план об'їзду пунктів об'їзду складається з 1 і більше маршрутів, має вигляд:

$$Q \geq 1 \quad (3.6)$$

Тоді враховуючи змістовну та математичну постановки задач, маємо таку структуру розв'язку:

$$p_1, k_0^1, k_1^1, \dots, k_{n_1-1}^1 P, \dots, P, k_0^Q, k_1^Q, \dots, k_{n_Q-1}^Q P,$$

де $P = \{p_1, p_2\}$.

3.3 Обґрунтування методу розв'язання

3.3.1 Класифікація задач транспортної маршрутизації

Класифікація задач транспортної маршрутизації дає можливість більш системно підходити до вивчення і розв'язання цих задач. Кожен вид задачі має свої особливості і вимагає використання відповідних методів і інструментів для розв'язання. Класифікація також допомагає краще розуміти суть задачі і знаходити оптимальні розв'язки.

Знання класифікації задач транспортної маршрутизації дозволяє ефективніше планувати та оптимізувати логістику та транспортні процеси, знижуючи витрати та час на доставку товарів або послуг.

Існує багато різних типів задач маршрутизації транспорту через те, що вони можуть мати різні характеристики та обмеження, які впливають на їх розв'язання. Наприклад, деякі задачі можуть мати обмеження на кількість транспортних засобів, їх максимальну потужність або обсяг перевезення, тоді як інші задачі можуть включати обмеження на час або відстань, яку треба пройти. Крім того, різні типи задач можуть вимагати різних методів розв'язання, які можуть бути більш або менш ефективними залежно від контексту та обмежень.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

Тому, для того, щоб ефективно розв'язувати задачі маршрутизації транспорту, необхідно розуміти різні типи задач та їх характеристики. В залежності від конкретної формулювання задачі, існує кілька типів VRP, які можна класифікувати за певними ознаками [3]:

а) залежно від кількості точок доставки;

1) VRP з одним депо (Single Depot VRP) – задача полягає в знаходженні найкоротшого маршруту для кожного транспортного засобу з одним вибраним депо;

2) VRP з багатьма депо (Multiple Depot VRP) – ця задача полягає в знаходженні найкоротших маршрутів для кожного транспортного засобу з кількох депо;

б) залежно від обмежень, накладених на ресурси;

1) VRP з обмеженнями на час (Time– Constrained VRP) – це задача, в якій додатково до знаходження найкоротших маршрутів для транспортних засобів необхідно дотримуватися певних обмежень на час доставки;

2) VRP з обмеженнями на потужність (Capacity–Constrained VRP) – в цій задачі додатково до знаходження найкоротших маршрутів необхідно дотримуватися обмежень на максимальну потужність кожного транспортного засобу;

3) VRP з обмеженнями на відстань (Distance– Constrained VRP) – ця задача вимагає знаходження найкоротших маршрутів для транспортних засобів з обмеженнями на максимальну дистанцію, яку транспортний засіб може проїхати;

4) VRP з комбінованими обмеженнями (Mixed VRP) – ця задача об'єднує кілька типів VRP і включає обмеження на час, потужність, відстань та інші параметри;

в) залежно від умов та обмежень, накладених на час;

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

1) VRP з часовими вікнами (Vehicle Routing Problem with Time Windows VRPTW) – у цій задачі кожне місце призначення має визначений часовий проміжок, протягом якого можна здійснити доставку;

2) VRP з часовими вікнами та роздільною доставкою (Vehicle Routing Problem with Time Windows and Split Deliveries VRPTWSD): включає в себе відвезення частини товару на початку дня, а іншої частини в певний час пізніше;

3) VRP з транзитними перевезеннями і часовими вікнами (Vehicle Routing Problem with Backhauls and Time Windows VRPBW): включає в себе не тільки відвезення товару до клієнтів, але також забирання вантажу або виконання інших завдань на поверненні до бази;

г) залежно від динамічності вихідних даних (параметрів):

1) статична задача VRP (Static VRP) – це задача, де умови та параметри є статичними, тобто не змінюються з часом;

2) динамічна задача VRP з попередньою інформацією (VRPPD) – це задача, де деякі параметри можуть змінюватися з часом, але ця зміна передбачена заздалегідь, і інформація про неї відома;

3) динамічна задача VRP з невідомим майбутнім (VRPUD) – це задача, де деякі параметри можуть змінюватися з часом, але ці зміни не передбачені заздалегідь, тому інформація про них не відома заздалегідь;

4) змішана динамічна задача VRP (MDVRP) – це задача, де деякі параметри можуть змінюватися з часом, але деякі зміни передбачені заздалегідь, а деякі – ні;

д) залежно від метрики відстані:

1) евклідова VRP: задача, в якій відстань між точками вимірюється у прямих лініях.

2) манхетенська VRP: задача, в якій відстань між точками вимірюється вздовж вулиць (за прямими кутами).

е) залежно від кількості критеріїв оптимізації:

					<i>IC92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

1) single objective VRP: задача, в якій є один критерій оптимізації, наприклад, мінімізація відстані або мінімізація кількості транспортних засобів.

2) multi-objective VRP: задача, в якій є декілька критеріїв оптимізації, наприклад, мінімізація відстані та мінімізація витрат на паливо [3].

Окрім цього переліку класів задачі транспортно маршрутизації існує багато інших видів транспортної маршрутизації, які відрізняються за різними критеріями і мають свої особливості. Це основна класифікація, яка зустрічається у багатьох джерелах і найчастіше використовується на практиці.

Отже, задачі маршрутизації транспорту (VRP) є важливими в контексті оптимізації логістичних процесів та транспортних маршрутів. Вивчення різних типів VRP дозволяє знаходити ефективні розв'язки для різноманітних завдань, таких як розвезення товарів, забір відходів, евакуація населення тощо. Існує безліч типів VRP, включаючи базові, які описують простіші сценарії, а також складніші, які дозволяють враховувати багато різних факторів, таких як обмеження на час, обсяг та вагу вантажу, вартість перевезення тощо. Дослідження різних типів VRP має великий практичний інтерес і може сприяти покращенню транспортної логістики та ефективності бізнесу.

3.3.2 Аналіз методів розв'язання задач оптимізації

Задача оптимізації маршруту транспортних засобів є NP-повною, тобто ми можемо говорити про складність її розв'язання точними алгоритмами. Аби знайти оптимальний розв'язок потрібно обрахувати усі можливі розв'язки. Час розв'язку задачі збільшується за експонентою в залежності від кількості пунктів. Проте для задач з невеликою кількістю пунктів обслуговування точні методи можуть бути застосовані. Існує багато методів для розв'язання, деякі з них описані нижче.

3.3.2.1 Точні методи

Динамічне програмування – це алгоритмічний підхід до розв'язування задач шляхом розбиття їх на менші підзадачі і знаходження оптимального розв'язку шляхом обчислення оптимальних рішень для кожної підзадачі.

Основна ідея динамічного програмування полягає в тому, щоб зберігати результати попередніх обчислень та використовувати їх для подальших обчислень. Це дозволяє уникнути повторних обчислень та скоротити час виконання алгоритму.

Метод гілок та меж – це алгоритм для пошуку оптимального розв'язку задачі оптимізації у випадку, коли задача має складну структуру та вимагає перебору великої кількості можливих варіантів розв'язку.

Метод гілок та меж розбиває задачу на підзадачі та здійснює перебір можливих варіантів розв'язку цих підзадач. При цьому здійснюється відсіювання неперспективних варіантів за допомогою оцінювання верхніх та нижніх меж для кожної підзадачі.

Кожна підзадача розглядається як гілка дерева, де кожен вузол є певним варіантом розв'язку попередньої підзадачі. Кожна гілка поступово розбивається на дочірні вузли до тих пір, поки не буде знайдено оптимальний розв'язок задачі.

Нижче наведений загальний псевдокод методу гілок та меж для задач маршрутизації транспортних засобів:

КРОК 1 Ініціалізуємо початкову підзадачу.

КРОК 2 Знаходимо найкращий розв'язок для початкової підзадачі.

КРОК 3 Якщо найкращий розв'язок для початкової підзадачі не задовольняє умову оптимальності, розбиваємо початкову підзадачу на менші підзадачі.

КРОК 4 Для кожної нової підзадачі повторюємо кроки 2– 4.

КРОК 5 Якщо всі підзадачі задовольняють умову оптимальності, знаходимо найкращий розв'язок для всієї задачі [3].

Метод гілок та меж може бути застосований для розв'язування різноманітних задач, включаючи задачі комівояжера, задачі розкладу, задачі управління запасами тощо.

Метод гілок та меж з відсіканнями додатково використовує інформацію про граф залежностей обмежень задачі та відсікання, які дозволяють позбутися непотрібних підзадач та скоротити час розв'язання [3].

3.3.1.2 Евристичні методи

Метод "Sweep" – це простий евристичний метод для розв'язування задачі маршрутизації транспортних засобів, який базується на тому, щоб згрупувати замовлення, які знаходяться поруч з одним зв'язком на маршруті.

Алгоритм полягає у наступних кроках:

КРОК 1 Вибрати точку старту та напрямок руху транспортного засобу.

КРОК 2 Визначити всі замовлення, які знаходяться по обраному напрямку, та відсортувати їх за відстанню від точки старту.

КРОК 3 Починаючи з першого замовлення, додавати їх до поточного маршруту, доки не буде досягнуто максимальну ємність транспортного засобу або не будуть додані всі замовлення, які знаходяться на цьому напрямку.

КРОК 4 Змінити напрямок руху транспортного засобу та повторити кроки 2–3, доки всі замовлення не будуть включені до маршруту.

Цей метод є простим та швидким способом розв'язання задачі VRP, але може не давати оптимальних рішень. Також, він може бути обмежений у випадку, коли замовлення розташовані далеко від точки старту або коли кількість замовлень є дуже великою [21].

Метод збереження є одним з простих евристичних методів для розв'язання задачі маршрутизації транспортних засобів (VRP). Він був запропонований в 1963 році французьким вченим Клодом Гроувесом (Claude Grouve).

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

Метод збереження базується на ідеї об'єднання двох клієнтів у один маршрут з метою зменшення відстані, що проходить транспортний засіб. Суть методу полягає у наступному:

КРОК 1 Визначається початковий маршрут, наприклад, з використанням методу сусіднього клієнта (nearest neighbor method).

КРОК 2 Для кожної пари клієнтів (i, j) обчислюється зменшення відстані, що можливе при їх об'єднанні в один маршрут: $\text{Saving}(i,j) = d(0,i) + d(i,j) + d(j,0) - d(0,i) - d(0,j)$.

КРОК 3 Клієнти відсортовуються за зменшенням значення $\text{Saving}(i,j)$.

КРОК 4 Кожна пара клієнтів додається до поточного маршруту, якщо вона не порушує обмеження на кількість відвідуваних клієнтів транспортним засобом та не перетинається з іншими маршрутами.

КРОК 5 Процес додавання пар клієнтів продовжується доти, поки всі клієнти не будуть розміщені в маршрути.

Хоча метод збереження є простим і легко реалізовується, він не завжди дає оптимальні розв'язки задачі маршрутизації транспорту, адже не враховує всі можливі варіанти комбінування клієнтів у маршрути [21].

3.3.1.3 Метаевристичні методи

Табу пошук (Tabu search) є метаевристичним методом оптимізації, який використовується для розв'язання складних задач оптимізації, таких як VRP. Метод табу пошуку був розроблений Фредом Гловером в 1986 році.

Основна ідея табу пошуку полягає в тому, щоб уникати повторення кроків, які ведуть до погіршення розв'язку. Для цього використовується список табу (Tabu list), який містить описи останніх кроків пошуку, які були виконані. Кожний крок пошуку, який призводить до погіршення розв'язку, додається до списку табу на певний час (Tabu period), після чого його можна повторно виконати.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		29

Табу пошук використовує різні стратегії для знаходження кращого розв'язку. Одна з них – це заборона зміни деяких компонентів поточного розв'язку на певний час. Інші стратегії включають вибір кращого можливого розв'язку серед всіх доступних варіантів та виконання деяких обмежень на кількість кроків пошуку.

Табу пошук може бути досить ефективним методом для розв'язання VRP, оскільки він дозволяє уникнути повторення поганої дії та забезпечує збереження попереднього розв'язку. Однак, як і з будь-яким методом оптимізації, ефективність табу пошуку залежить від властивостей задачі та параметрів пошуку [21].

Метод імітації відпалу (Simulated annealing) – це метаевристичний алгоритм пошуку, який використовується для глобальної оптимізації складних функцій. Цей метод був розроблений Скоттом Кіркпатріком та Міхаелом Гелаті в 1983 році.

Метод імітації відпалу походить від фізичного процесом відпалу, коли матеріал, наприклад, сталь або скло, нагрівається, а потім охолоджується. Шляхом імітації цього фізичного процесу кожен крок алгоритму імітації відпалу замінює поточне розв'язку на сусіднє. Сусідній розв'язок генерується випадковим чином за допомогою функцій/операторів, які розроблені спеціально для конкретної проблеми.

Метод імітації відпалу для розв'язання VRPTW вимагає трьох основних процесів:

КРОК1 Генерація випадкового початкового розв'язку.

КРОК 2 Створення нового розв'язку за допомогою спеціальних операторів, які працюють з сусідніми розв'язками.

КРОК 3 Застосування функції для зменшення ймовірності прийняття гіршого нового розв'язку, що допомагає уникнути локальних оптимумів під час пошуку глобального оптимуму [21].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		30

Генетичний алгоритм (Genetic Algorithm) – це метод еволюційної оптимізації, який моделює природний процес еволюції для розв’язання задач оптимізації. У генетичному алгоритмі, починаючи зі згенерованої випадкової популяції, кожне нове покоління створюється шляхом застосування генетичних операторів, таких як мутація, кросовер та селекція. У процесі мутації випадковим чином змінюється частина генотипу, а кросовер використовує два батьківські генотипи для створення нового потомства. Після створення нового покоління, здійснюється селекція, де відбираються найкращі індивіди для створення наступного покоління. Генетичний алгоритм є ефективним методом для розв’язання задач оптимізації, особливо тих, які мають складну функцію мети та велику кількість змінних.

Генетичні алгоритми можуть використатися для пошуку рішень в дуже великих і тяжких просторах пошуку.

Можна виділити наступні кроки генетичного алгоритму:

КРОК 1 Створюємо початкову популяцію.

КРОК 2 Обчислюємо функцію пристосованості для осіб популяції.

КРОК 3 Повторюємо до виконання критерію зупинки алгоритму.

КРОК 4 Вибираємо індивідів із поточної популяції (селекція).

КРОК 5 Схрещуємо або/та мутація.

КРОК 6 Обчислюємо функцію пристосованості для всіх осіб.

КРОК 7 Формуємо нове покоління [21].

Адаптивний пошук з пам'яттю (Adaptive Memory Search) – це клас метаевристичних методів оптимізації, які комбінують в собі ідеї інтенсифікації та диверсифікації для ефективного пошуку глобального оптимуму.

Інтенсифікація – це процес зосередження пошукових зусиль на найбільш перспективних областях простору розв'язків з метою знайти найкращий розв'язок проблеми. Це важлива складова багатьох методів оптимізації і пошуку, включаючи метаевристики.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

Інтенсифікація допомагає підвищити швидкість збіжності алгоритму та забезпечує більш точний результат. Вона зокрема корисна у випадках, коли простір розв'язків має групу областей з високою якістю та областей з низькою якістю, оскільки інтенсифікація дозволяє зосередити зусилля на найбільш перспективних областях.

Диверсифікація – це процес розширення пошукового простору з метою уникнення застрягання в локальних оптимумах і покращення шансів знайти глобальний оптимум.

У контексті оптимізації, диверсифікація використовується для створення різних початкових розв'язків та пошуку віддалених регіонів простору розв'язків. Вона зокрема корисна у випадках, коли простір розв'язків має багато локальних оптимумів або коли не відомо, які області простору розв'язків мають високу якість.

Цей метод пошуку використовує пам'ять, щоб зберігати інформацію про попередні розв'язки та їх якість, і використовує її для генерації нових рішень. Він поєднує в собі інформацію про попередні розв'язки та структуру проблеми для покращення результатів пошуку.

Одним з ключових елементів адаптивного пошуку з пам'яттю є здатність алгоритму до адаптації до змін у функції мети та структурі проблеми. Адаптивний пошук з пам'яттю включає в себе методи, які дозволяють контролювати інтенсивність та диверсифікацію пошуку, щоб забезпечити баланс між інтенсивністю та диверсифікацією.

Метод адаптивного пошуку з пам'яттю використовується для розв'язання різноманітних задач оптимізації, таких як планування маршрутів, управління виробництвом, а також для розробки алгоритмів машинного навчання. Його ефективність полягає в тому, що він може знайти оптимальні розв'язки навіть в складних задачах з багатьма обмеженнями [21].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

Нейронні мережі (Neural networks) можуть бути використані в задачах маршрутизації транспортних засобів (VRP) різними способами. Один зі способів – використання їх для безпосереднього розв'язання VRP, де мережа навчається виводити оптимальний розв'язок для заданого випадку VRP. Інший підхід полягає в використанні нейронних мереж як частини більшої оптимізаційної алгоритмічної системи, такої як метаевристика, для покращення якості знайдених рішень.

У першому підході нейронна мережа навчається на великому наборі випадків VRP, де кожен випадок містить інформацію, таку як кількість клієнтів, їхні місця розташування та місткість транспортних засобів. Потім мережа може отримувати нові випадки та виводити оптимальний розв'язок, який може бути використаний як вихідна точка для подальшої оптимізації.

У другому підході нейронна мережа використовується для генерації початкових розв'язків або для направлення процесу пошуку в перспективні напрямки розв'язків. Наприклад, мережу можна використовувати для генерації набору кандидатських розв'язків, які потім можна уточнити за допомогою алгоритмів локального пошуку.

В цілому використання нейронних мереж в VRP може покращити якість розв'язків та зменшити час, необхідний для їхнього знаходження. Однак проектування та навчання нейронних мереж для VRP може бути складним та вимагає ретельного вивчення структури задачі та наявних даних.

Мурашиний алгоритм (Ant Colony Optimization (ACO)) – це метаевристичний алгоритм, що схожий на поведінку мурах, що шукають їжу. У природі мурахи відкладають феромони на землю під час пошуку їжі, які привертають інших мурах слідувати по тому ж самому шляху. З часом, шлях з найвищою концентрацією феромонів стає найбільш відвідуюваною маршрутом [21].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

3.3.2 Опис табу пошуку для розв'язання VRP

У попередніх розділах роботи була детально розглянута класифікація алгоритмів для розв'язання задач комбінаторної оптимізації на прикладах застосування для задачі маршрутизації транспортних засобів. Окремо розглянуто точні, евристичні та метаевристичні методи, їх переваги та недоліки. Оскільки для обраної задачі важлива як точність (яка є низькою при застосуванні евристичних алгоритмів), так і можливість розв'язання задачі на середніх розмірностях даних (а для точних методів переважно доступне розв'язання задач лише на невеликих розмірностях), оберемо для проектування один з проаналізованих метаевристичних алгоритмів. Реалізація алгоритмів даного класу складніша, порівняно з реалізацією точних та евристичних методів, але при роботі з ними можна провести аналіз вхідних параметрів алгоритму, шляхом експериментів та аналізу наборів вихідних даних. Недоліком такого вибору може бути час розв'язання задач. Так, наприклад, евристичні алгоритми зазвичай швидше знаходять розв'язки, оскільки вони використовують спрощені правила та методи, а не поєднують у собі кілька підходів, що дозволяє їм пропустити певний об'єм розрахунків за невеликий проміжок часу.

Серед метаевристичних методів для проектування, імплементації та аналізу обрано табу пошук. Основні принципи табу пошуку було наведено у попередніх розділах, при аналізі методів, розглянемо алгоритм детальніше.

Табу пошук був вперше введений Фредом Гловером у 1986 році в статті, яку він опублікував. Табу пошук був винайдений через велику кількість складних комбінаторних задач оптимізації в різних сферах, а саме: логістика, інженерія, економіка бізнес та інші. Згідно з Гловером, Табу пошук – це метаевристика, яка використовує локальні евристики для аналізу простору можливих рішень. Пошук табу розпочинається як звичайний локальний пошук, ітеративно переходячи від одного розв'язку до наступного, поки не буде задоволено певний критерій зупинки. В процесі використовуються різні

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		34

стратегії, щоб уникнути потрапляння в локальні оптимуми. Фактично, локальний пошук, наприклад, метод спуску, часто призводить до досягнення локальних оптимумів, які можуть бути або не бути глобальним оптимумом для мінімізації цільової функції.

Однією з характерних особливостей табу пошуку, відмінною від інших методів пошуку, є використання пам'яті, яка може бути явною або атрибутивною. Явна пам'ять зберігає повні розв'язки, у тому числі рекордні розв'язки, які були відвідані під час пошуку. Крім того, ця пам'ять може фіксувати нестандартні не відвідані сусідні розв'язки рекордних розв'язків. Локальний пошук розширюється, використовуючи збереження рекордних рішень або їх нестандартних сусідів (Glover and Laguna, 1997). З іншого боку, атрибутивна пам'ять використовується для керування процесом пошуку. Вона зберігає інформацію про зміну атрибутів розв'язків під час переходу від одного розв'язку до іншого. Наприклад, це можуть бути вузли та дуги, що видаляються, додаються або переміщуються на графі згідно з механізмом переміщення. В нашому випадку ми використовуємо явну пам'ять, де зберігаємо рішення, які відвідуються під час пошуку.

Відповідно до Brassy(2001), існує два типи структур сусідства, а саме:

- проста структура сусідства;
- складна структура сусідства.

Проста структура сусідства створюється шляхом виконання простих операцій переміщення. Наприклад, це передбачає переміщення лише одного пункту обльоту на кожній ітерації з одного маршруту та розміщення останнього в іншому маршруті або десь ще в тому ж маршруті. Наприклад, на рисунку 3.2. пункт 2 було переміщено зі свого поточного положення (між пунктом 1 і пунктом 3) і вставлено між пунктами 3 і 4.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		35

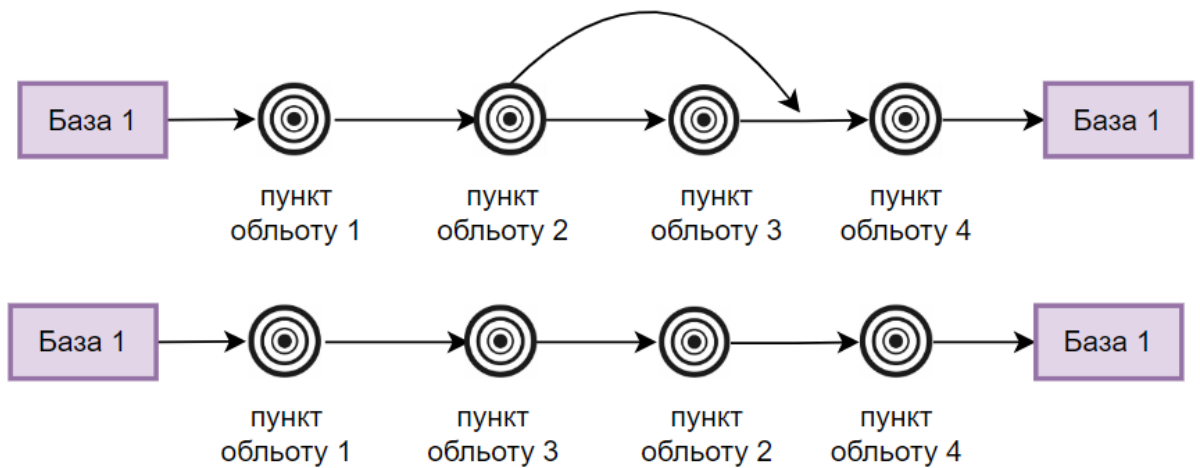


Рисунок 3.2 – Схема простої структури сусідства

Побудова складних структур сусідства передбачає складніші операції переміщення. Такі переміщення можуть включати в себе оператор переміщення, оператор обміну, оператор 2–орт або навіть їх комбінацію. На рисунку 3.3 зображено оператор переміщення обміну [5]. У цьому прикладі є два маршрута, а саме: маршрут 1, який починається у базі 1 та закінчується у базі 1 та маршрут 2, який починається у базі 1, а закінчується у базі 2. Для прикладу пункт обльоту 2 перевели з маршруту 1 на маршрут 2, а пункт обльоту 4 – на маршрут 1.

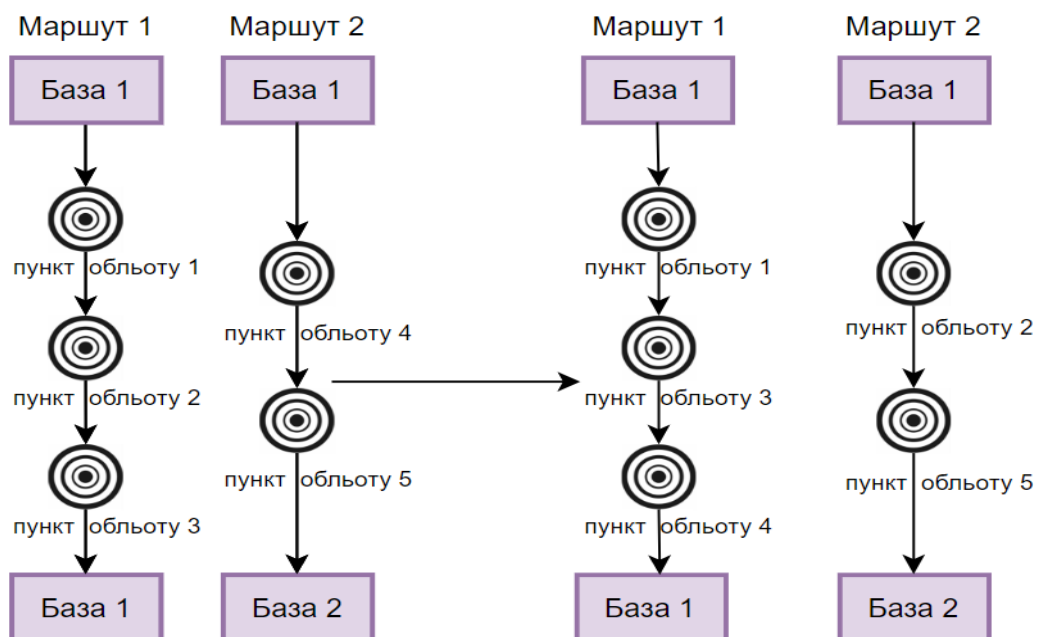


Рисунок 3.3 – Схема складної структури сусідства

пам'яттю, особливо для великих задач, але ця проблема може бути вирішена підходами до формування табу списку, що розглядаються далі у цьому розділі;

– залежність від налаштувань – як вже було визначено раніше, табу пошук має багато параметрів, які потрібно налаштувати для оптимального результату, це є як перевагою, оскільки коректне налаштування параметрів дозволяє адаптувати алгоритм конкретно до поставленої задачі, так і недоліком, оскільки належні налаштування можуть привести до невдачі у пошуку, а також вимагають часу на експерименти, їх аналіз так налаштування.

3.4 Опис методу розв'язання

Опишемо алгоритм табу пошуку для маршрутизації БПЛА формально. Покроково, формально табу пошук можна представити так:

КРОК 0 Ініціалізація вхідних параметрів алгоритму:

– I – кількість ітерацій, протягом яких розв'язок не покращується;
– X – кількість поточних розв'язків, знайдених локальною оптимізацією;
– O – окіл, у якому проводиться локальна оптимізація, задається відсотком від поточної кількості розв'язків.

КРОК 1 Генерація випадкового розв'язку та призначення його рекордним.

КРОК 2 Створення табу списку розв'язків, порожній на початку.

КРОК 3 ПОКИ рекордний розв'язок не покращується впродовж I ітерацій.

КРОК 3.1 локальна оптимізація (в околі O рекордного розв'язку визначити m розв'язків, що виключають розв'язки, що входять до табу списку).

КРОК 3.2 Оновлення рекордного розв'язку (за потребою, якщо найкращий знайдений локальною оптимізацією розв'язок кращий за поточний рекордний).

КРОК 3.3 Оновлення табу списку.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		38

На нульовому кроці проводитиметься адаптація імплементованого алгоритму під вихідні дані, на яких проводитиметься експеримент, шляхом налаштування вхідних параметрів алгоритму. Саме тому для експериментів потрібно використовувати дані, що є максимально наближеними до реальних.

На першому кроці, при генерації випадкового розв'язку використано стохастичний алгоритм, це дозволяє з самого початку обрати прийнятний розв'язок та не витрачати зайві ресурси на додаткові ітерації по покращенню поганого розв'язку.

Стохастичний алгоритм, покроково обирає з масиву пунктів обльоту досяжні з поточного пункту (бази чи пункту обльоту відповідно), при цьому при визначенні враховуючи не лише те що БПЛА вистачить заряду на те, щоб долетіти до пункту, а й на те, щоб долетіти до бази на підзарядку після відвідування цього пункту. Залежно від віддаленості досяжних пунктів від поточного, кожному з них пропорційно присвоюється ймовірність відвідування. Усі ймовірності відвідування пунктів обльоту з поточного пункту у сумі дають одиницю. Випадковим чином, але враховуючи ймовірність, стохастичний алгоритм обирає який пункт відвідати і переходить до пошуку сусіда вже для нього. Якщо на певному етапі заряду БПЛА вистачає лише на те, щоб долетіти до бази, БПЛА летить до бази. Таким чином, вже на етапі генерації початкового розв'язку отримуємо прийнятний.

Далі впродовж певної кількості ітерацій проводиться локальна оптимізація та поповнення табу списку. Кількість ітерацій, що необхідні для розв'язання задачі визначається вихідним параметром, що регламентує, яка кількість ітерацій повинна бути виконана з умовою того що рекордний розв'язок не оновлюється.

Локальна оптимізація для кожного з побудованих планів обльоту виконується шляхом побудови можливих (тобто таких, що задовільняють обмеження математичної постановки, зокрема на максимальний час, що БПЛА може провести у повітрі, тобто максимальну довжину підмаршруту від бази до бази) перестановок пунктів обльоту, можливих перестановок баз і спроби зменшити кількість маршрутів у плані обльоту, прибравши зайві відвідування баз в околі, який задано у вхідних параметрах алгоритму.

Після локальної оптимізації відбувається пошук серед знайдених розв'язків найкращого та його порівняння з рекордом. Якщо цей розв'язок кращий за рекорд, а також якщо цього розв'язку немає в табу списку, то рекорд оновлюється та замінюється ним, при чому попередній рекорд заноситься до табу списку.

Табу список оновлюється після кожної ітерації і доповнюється проаналізованими розв'язками, окрім поточного рекордного розв'язку (включно і з попереднім рекордним розв'язком, якщо він оновлювався, і з розв'язками, отриманими завдяки локальній оптимізації).

3.4.1 Підбір параметрів для розробленого алгоритму

Для дослідження розробленого алгоритму необхідно спочатку підібрати параметри алгоритму.

Підбір оптимальних параметрів алгоритму є важливим етапом у розробці. Розглянемо кілька методів для підбору параметрів, які є найвживанішими.

Grid Search (Пошук по сітці): цей метод включає в себе визначення сітки значень для кожного параметра алгоритму. Алгоритм потім перебирає всі можливі комбінації параметрів, обчислює метрику якості і вибирає набір параметрів з найкращими результатами. Grid Search може бути часом витратним, особливо якщо маємо велику кількість параметрів або широкий діапазон значень для кожного параметра [19].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		40

логічні обмеження накладені на параметри (наприклад, параметри табу– пошуку не можуть бути від’ємні) а також орієнтовні границі параметрів (наприклад, табу– список може бути досить великим, щоб працювати ефективно, збільшення кількості ітерацій алгоритму підвищить точність отриманих результатів, але сильно вплине на використання ресурсів системи).

Після задання простору підбору параметрів вручну, та проведення самого підбору, результати аналізуються, та якщо значення досить близьке до нижньої або верхньої границі простору підбору параметрів, то границі простору розширюються, в рамках допустимих. В результаті отримано такі значення простору підбору параметрів:

- кількість ітерацій без покращення – від 1 до 50;
- довжина табу списку – від 1 до 1000.

Проведення випадкового пошуку параметрів відбувається на задачах розмірності 20. Генератор задач випадковим чином генерує 15 задач заданої розмірності. Ці задачі вирішуються 100 раз, кожного разу з новими, згенерованими системою, значеннями параметрів. Для кожного зі 100 наборів параметрів обчислюється середнє значення цільової функції по 15 вирішеним задачам. Обирається найкраще середнє значення цільової функції і відповідні параметри вважаються ефективними. У таблиці 3.1 наведено результати проведення випадкового пошуку параметрів.

Таблиця 3.1 – Результати проведення підбору параметрів

Нумерація генерації параметрів	Середнє значення цільової функції	Довжина табу списку	Кількість прогонів алгоритму	Кількість ітерацій без покращення
1	82.73	912	50	50
2	81.79	334	50	7
3	83.57	365	50	9
4	82.18	879	50	42
5	82.74	756	50	36
6	81.75	125	50	39

7	81.14	323	50	36
8	79.87	12	50	46
9	81.65	275	50	39
10	80.99	196	50	40
11	82.73	214	50	6
12	82.24	699	50	48
13	83.93	279	50	28
14	82.62	883	50	9
15	83.9	155	50	40
16	82.41	28	50	2
17	82.16	295	50	15
18	82.81	838	50	14
19	82.74	97	50	21
20	82.59	875	50	23
21	81.78	42	50	28
22	82.61	933	50	29
23	83.14	293	50	27
24	83.66	221	50	19
25	82	132	50	26
26	82.68	1	50	10
27	80.76	53	50	25
28	83.43	737	50	17
29	82.82	331	50	44
30	82.59	184	50	11
31	83.34	511	50	37
32	82.59	312	50	49
33	82.04	160	50	45
34	82.49	53	50	27
35	82.62	725	50	39
36	79.35	38	50	41
37	82.82	772	50	49
38	83.25	177	50	25
39	81.14	893	50	1
40	82.58	611	50	18
41	81.55	94	50	39
42	82.61	214	50	12
43	83.32	642	50	6

Зм.	Лист	№ докум.	Підпис	Дата

IC92.220БАК.004 ПЗ

Арк.

43

44	83.05	353	50	44
45	81.84	141	50	9
46	81.47	853	50	32
47	83.7	253	50	9
48	82.21	170	50	12
49	81.72	625	50	31
50	82.63	305	50	48
51	82.75	265	50	47
52	82.36	206	50	25
53	80.97	890	50	17
54	82.27	562	50	1
55	83.1	453	50	38
56	84.1	410	50	7
57	82.12	293	50	4
58	81.29	604	50	3
59	80.62	793	50	21
60	83.16	402	50	48
61	84.13	269	50	8
62	82.75	784	50	49
63	82.19	978	50	38
64	83.68	737	50	16
65	81.34	129	50	13
66	82.76	814	50	25
67	81.9	849	50	10
68	83.08	447	50	35
69	82.26	570	50	37
70	81.72	842	50	45
71	83.45	194	50	47
72	81.42	190	50	12
73	82.25	914	50	2
74	82.32	836	50	30
75	83.16	590	50	38
76	82.64	912	50	21
77	82.53	92	50	2
78	82.95	816	50	1
79	81.69	36	50	4
80	81.78	570	50	24

Зм.	Лист	№ докум.	Підпис	Дата

IC92.220БАК.004 ПЗ

Арк.

44

81	83.42	360	50	41
82	83.49	260	50	8
83	82.01	201	50	6
84	82.19	596	50	42
85	83.73	730	50	47
86	82.18	670	50	39
87	82.58	407	50	41
88	83.41	775	50	48
89	81.82	465	50	17
90	82.75	829	50	12
91	81.29	398	50	46
92	82.66	41	50	46
93	83.54	854	50	9
94	81.62	802	50	3
95	81.85	642	50	17
96	81.52	243	50	45
97	83	691	50	50
98	80.35	579	50	16
99	82.49	32	50	6
100	82.04	676	50	40

Після проведення випадкового пошуку параметрів у вже визначеному просторі підбору параметрів, отримано такі значення ефективних параметрів:

- кількість ітерацій без покращення ~50;
- довжина табу списку ~38.

3.4.2 Результати роботи розробленого алгоритму

Проведемо перевірку роботи табу пошуку. Для кожної з обраних розмірностей задач експерименти на 10, 25 та 50 прогонів, тобто 10, 25, або 100 згенерованих задач. Визначені середні значення цільової функції, краще значення цільової функції та середньоквадратичне відхилення цільової функції по усім прогонам. Результати перевірки наведено у таблиці 3.2.

Таблиця 3.2 – Результати роботи табу-пошуку

Кількість прогонів	Розмірність задачі	Краще значення цільової функції	Середнє значення цільової функції	Середньоквадратичне відхилення
10	10	45,58	53,65	5,33
	15	52,17	64,6	5,86
	20	75,72	83,06	5,43
	25	85,14	93,41	4,85
	30	103,2	113,85	9,36
25	10	44,26	53,83	5,3
	15	44,7	63,99	8,97
	20	67,19	82,58	9,21
	25	76,94	92,21	6,54
	30	87,16	108,36	9,03
50	10	41,58	50,58	6,28
	15	43,37	63,34	7,13
	20	62,59	81,81	7,74
	25	75,39	94,82	9,62
	30	83,67	108,79	10,28

для підбору параметрів алгоритму, з якими він розв'язує задачі з кращими результатами.

					<i>IS92.220БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

4 ПРОГРАМНЕ І ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

4.1.1 Програмне середовище розробки системи

Для розробки було обрано програмне середовище Visual Studio Code – інтегроване середовище розробки (IDE), це безкоштовне сучасне програмне забезпечення. Дослідження Stack Overflow, популярної мережі для розробників, показують, що у 2022 році саме Visual Studio Code займає перше місце у рейтингу найвживаніших IDE – 74.48% респондентів надали перевагу саме цій IDE. При цьому, розрив з її наступником значний, друге місце посідає Visual Studio з 32.1% [6]. Також з досліджень видно, що останні кілька років популярність Visual Studio Code стрімко зростає. Це призводить до активного підтримання програмного забезпечення і розширення його функціоналу сучасним.

Ключовими перевагами, що вплинули на вибір саме цієї IDE є:

- підтримка розширень – VS Code має багато розширень, що дозволяють доповнювати його можливості та функціональність; це може бути корисним для додавання додаткових інструментів та функцій, що полегшують розробку системи;
- інтеграція з Git – VS Code має вбудовану підтримку Git, що дозволяє легко працювати з репозиторіями Git; це може бути корисним для розробки та збереження коду системи, та спільної роботи над проектом паралельно;
- підтримка налагоджування коду – VS Code має вбудовані інструменти для налагоджування коду, що дозволяє швидко знаходити та виправляти одруківки;
- підтримка автодоповнення коду – VS Code має підтримку автодоповнення коду, що дозволяє швидко та легко писати код, скорочуючи час на написання та запам'ятовування синтаксису.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		49

– легкість використання – VS Code має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє швидко засвоїти середовище та почати розробляти систему, також ми вже маємо досвід роботи з цим середовищем.

4.1.2 Мова розробки системи

Як основну мову для програмування обрано TypeScript – це потужна та універсальна мова програмування, яка є розширенням JavaScript, дозволяє використовувати статичну типізацію, об'єктно– орієнтовані та функціональні конструкції, інтерфейси та інші функції, що полегшують розробку великих та складних проектів з багатьма залежностями. TypeScript був створений компанією Microsoft і видано у доступ у 2012 році [8].

Однією з ключових особливостей TypeScript є статична типізація, яка дозволяє виявляти помилки на етапі компіляції, що допомагає запобігти помилкам та полегшити розробку. TypeScript дозволяє використовувати типи для змінних, функцій, параметрів та повернених значень, що дозволяє зробити код більш читабельним та більш прогнозованим [8].

TypeScript також підтримує об'єктно– орієнтоване програмування, включаючи класи, наслідування, інтерфейси та поліморфізм. Зокрема, TypeScript дозволяє використовувати інтерфейси для опису структури даних, що забезпечує більшу стабільність та компонованість коду [8].

Ще одним корисним функціоналом TypeScript є можливість використовувати нові функції JavaScript, що не підтримуються старішими браузерами. TypeScript підтримує ES6, ES7, ES8 та більшість функцій ES9 та ES10. ES6, ES7, ES8, ES9 та ES10 – це різні версії стандарту ECMAScript, який визначає синтаксис та функціональність мови JavaScript. Кожна нова версія включає нові функції та можливості для покращення розробки веб– додатків [9].

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

впроваджує нові функції та можливості, що передбачені у нових версіях стандарту ECMAScript, тому TypeScript завжди буде на передовій розробки;

– розширення можливостей – TypeScript можна використовувати не тільки для веб– розробки, але й для розробки додатків для мобільних пристроїв та десктопних додатків; це дозволяє розробникам писати код на TypeScript для широкого спектру проектів;

– вдосконалення інструментів розробки: TypeScript має потужну екосистему інструментів розробки, таких як Visual Studio Code, WebStorm, Sublime Text та інші; ці інструменти вдосконалюються та отримують нові можливості, що дозволяє розробникам працювати більш продуктивно та ефективно;

– використання в навчанні: TypeScript є потужним інструментом для навчання програмування; він дозволяє поєднувати навчання основ програмування з вивченням сучасних технологій, що забезпечує студентам ширший досвід та допомагає їм бути готовими до роботи в індустрії [12, 13].

Таким чином, обрання TypeScript як основної мови для програмування є виправданим, оскільки вона є потужною та універсальною мовою програмування, що дозволяє використовувати статичну типізацію, об'єктно–орієнтовані та функціональні конструкції, інтерфейси та інші функції, що полегшують розробку великих та складних проектів з багатьма залежностями. TypeScript також підтримує нові функції JavaScript, що дозволяє використовувати його для розробки сучасних веб– додатків.

4.1.3 Фреймворки використані для розробки системи

Аналіз, вибір та застосування фреймворків є невід'ємною частиною роботи проектування та розробкою системи.

По-перше, фреймворки надають готові рішення для типових задач, що дозволяє значно зменшити час, витрачений на написання коду вручну.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		52

4.1.4 Бібліотеки використані для розробки системи

Бібліотеки – це збірники функцій, підпрограм та інших ресурсів, зазвичай згруповані по функціоналу, які можуть бути використані у розробці програмного забезпечення [12].

Використання бібліотек дозволяє економити час і зусилля розробника. Замість написання всього коду з нуля, розробник може використовувати готові рішення з бібліотек. Це дозволяє скоротити час розробки та зосередитися на розробці більш важливих функцій та функціональності продукту. Бібліотеки часто мають широку спільноту користувачів та розробників, які активно підтримують та оновлюють їх. Це означає, вони проходять більш детальне тестування та перевірку. Також використання бібліотек дозволяє зменшити розмір програми та споживання ресурсів комп'ютера [12].

Бібліотеки, використані для розробки серверної частини:

– nestjs/common (версія 9.0.0)– це основна бібліотека фреймворку NestJS. Вона надає загальні функції та декоратори, необхідні для розробки модулів, контролерів, провайдерів та інших складових частин NestJS– додатків.

– nestjs/core (версія 9.0.0) містить основний функціонал ядра NestJS. Вона включає в себе контроль циклу життя додатків, механізми ініціалізації, інверсії керування та керування модулями.

– nestjs/jwt (версія 10.0.3) – це розширення NestJS, яке надає підтримку JWT (JSON Web Tokens); воно містить інструменти для генерації та перевірки JWT, а також підтримку аутентифікації і авторизації на основі JWT.

– nestjs/passport (версія 9.0.3) – це розширення NestJS, яке надає підтримку аутентифікації на основі Passport; вона дозволяє легко налаштовувати різні стратегії аутентифікації, такі як аутентифікація з використанням локального пароля, JWT, соціальних мереж тощо.

									Арк.
									55
Зм.	Лист	№ докум.	Підпис	Дата	<i>IC92.220BAK.004 ПЗ</i>				

– nestjs/platform– express (версія 9.0.0) – це розширення NestJS, яке надає інтеграцію з фреймворком Express; воно дозволяє створювати веб– сервери з використанням Express та використовувати його функціонал разом з NestJS.

– fs– extra (версія 11.1.1) – це розширена версія вбудованого модуля Node.js fs, яка надає додаткові функції для роботи з файловою системою; вона має зручний API для виконання операцій з файлами.

– "passport" (версія 0.6.0): passport – це популярна бібліотека аутентифікації для Node.js; вона надає широкий спектр стратегій аутентифікації, таких як аутентифікація з використанням пароля, JWT, соціальних мереж та багато інших; Passport спрощує процес аутентифікації та авторизації в додатках.

– passport– jwt (версія 4.0.1) – це розширення Passport, яке надає підтримку аутентифікації на основі JSON Web Tokens (JWT); воно дозволяє перевіряти та верифікувати JWT для аутентифікації користувачів в додатках.

– passport– local (версія 1.0.0) – це розширення Passport, яке надає підтримку аутентифікації з використанням локального пароля; воно дозволяє перевіряти введений користувачем логін та пароль для аутентифікації.

– reflect– metadata (версія 0.1.13) – це бібліотека, яка надає підтримку метаданих в TypeScript; вона дозволяє використовувати декоратори та отримувати доступ до метаданих класів, методів та властивостей під час виконання додатка; використовується у фреймворку NestJS для реалізації інверсії керування та рефлексії.

4.2 Вимоги до технічного забезпечення

Для роботи даної системи необхідно, щоб комп'ютер користувача відповідав наступним вимогам або перевищував їх:

- об'єм RAM – 2 Гб;
- об'єм HDD/SSD – 128 Гб;
- CPU – Core i3– 5xx, AMD A4/A6 або краще.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

- наявність встановленого браузерa останньої версії на комп'ютері.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма компонентів

У даній інформаційній системі присутні два основних компонента, а саме: інтерфейс користувача та сервер. Оскільки у даному дипломному проєкті реалізована серверна частина системи, у графічних матеріалах наведено діаграму компонентів, на якій зеленим кольором виділено компоненти системи, реалізовані в рамках дипломного проєкту.

На діаграмі представлено взаємодію компонентів між собою та те як вони залежать одна від одної.

Інформаційна система складання маршруту польоту БПЛА складається з трьох рівнів: рівень представлення – візуалізація даних, рівень компонентів – управління підзадачами та маршрутизація даних, та логічний рівень – математичні розрахунки та рівень зберігання даних, як вхідних, так і вихідних.

Для візуалізації даних на рівні представлення система використовує рівень компонентів. Для цього рівень компонентів користується рівнем зберігання даних, що містить результати роботи алгоритмів на логічному рівні у стандартизованому форматі. На рівні компонентів відбувається взаємодія користувача з програмою, ведення даних. Система приводить введені користувачем дані у стандартизований формат та передає на рівень зберігання даних. Система передає збережені вхідні дані через рівень компонентів у логічний рівень. Після роботи алгоритмів результат повертається на логічний рівень. Згодом рівень представлення візуалізує наш результат за допомогою рівня компонентів, що використовує логічний рівень. Окрім використання системою розроблених алгоритмів розв'язання поставленої задачі маршрутизації польоту БПЛА, ще однією підзадачею цього рівня є аналіз результатів роботи розроблених алгоритмів. Дані, отримані при роботі з компонентою аналізу

результатів роботи розроблених алгоритмів, також передаються на рівень зберігання даних, звідки рівень компонентів передає їх на рівень представлення даних на якому відбувається візуалізація результатів аналізу.

Для дипломного проєкту ми проводимо дослідження розроблених алгоритмів розв'язання поставленої задачі, їх аналіз, порівняння та візуалізацію результатів аналізу. В реальних умовах застосовуватиметься лише один алгоритм розв'язання – той, застосування якого дасть найкращі порівняно з іншими розробленими алгоритмами результати. Отже, не будуть використовуватися усі компоненти експерименту та компоненти алгоритмів, що показали гірші результати ніж найкращий розроблений алгоритм.

4.3.2 Діаграма послідовності

Діаграма послідовностей відображає дії двох акторів, а саме: звичайного користувача та дослідника.

Для початку побудови маршрутів, користувач повинен ввести вхідні дані та натиснути кнопку на інтерфейсі. Після цього вхідні дані передаються до модуля серверної частини Solver, який певним алгоритмом буде оптимальний маршрут та передає дані до бази даних для збереження. Solver повертає результат або інформує користувача про помилку. Користувач має можливість зберегти результат планування маршрутів. Якщо користувач натискає кнопку для збереження даних, тоді система відправляє запит на збереження файлу з результатами розв'язання до серверної частини. Серверна частина робить запит до бази даних і повертає потрібний файл з даними.

Розглянемо тепер дії дослідника.

Дослідник вводить дані для експерименту, а саме: алгоритм, над яким він може проводити експеримент, параметри обраного алгоритму та розмірність задачі. Для початку проведення експерименту, дослідник повинен натиснути кнопку на інтерфейсі. Вхідні дані надсилаються до серверної частини, Solver починає обробку даних та надсилає запит на генерування задачі до Input data

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

	маршруту		видалення заданої бази з вихідного маршруту
changeBase	route – маршрут indexOfBaseToChange – індекс бази, яку потрібно замінити replacement – нова база	Новий маршрут	Дана функція повертає новий маршрут, отриманий в результаті заміни заданої бази на іншу
createCalculateTimeFitness	speed – швидкість руху maxFlightTime – максимальний час польоту без необхідності в обслуговуванні chargeTime – час обслуговування між польотами	Повертає функцію, яка обчислює значення сумарного часу польотів та обслуговувань	Дана функція створює і повертає функцію, яка обчислює сумарний час польотів та обслуговувань
Функція, яку повернув createCalculateTimeFitness	route – маршрут	Значення сумарного часу обслуговувань та польотів	Дана функція обчислює сумарний час обслуговувань та польотів
isValidRoute	route – маршрут maxFlightTime – максимальний час польоту без необхідності в обслуговуванні speed – швидкість польоту pointsToObserve – масив точок, що потрібно облетіти	Булеве значення	Дана функція визначає чи є заданий маршрут валідним, тобто таким, що відповідає обмеженням задачі

getKNear estPoints	point – точка allPoints – масив всіх точок k – число в інтервалі [0, 1], яке відображає долю найближчих до заданої точки точок	Повертає масив точок	Дана функція для заданої точки знаходить найближчі точки у кількості, яка задається коефіцієнтом k
buildValidRoute	pointsToObserve – масив точок, які потрібно облетіти startBase – початкова база anotherBase – друга база chargeTime – час обслуговування між польотами (у хвилинах) maxFlightTime – максимальний польотний час без обслуговування (у хвилинах) speed – швидкість польоту	Повертає об'єкт, який містить поля: route – побудований маршрут fitness – значення цільової функції	Дана функція будує і повертає випадковий розв'язок такий, що відповідає обмеженням задачі. Алгоритм має стохастичних елемент, але розв'язок не є повністю випадковим, оскільки вірогідність обрання кожної доступної точки обернено пропорційна відстані до неї
findAvailablePoint	currentPoint – поточна точка unvisitedPoints – точки, що ще не були відвідані bases – масив баз restOfFlightTime – залишок часу польоту без обслуговування	Повертає об'єкт, який містить обрану точку, новий залишок часу польоту та оновлений масив невідвіданих точок	Дана функція обирає наступну точку, яка буде відвідана при побудові маршруту функцією buildValidRoute і повертає дані, які характеризують поточний стан побудови маршруту

	speed – швидкість польоту maxFlightTime – максимальний час польоту без обслуговування		
isAvailablePoint	currentPoint – поточна точка unvisitedPoints – інша точка bases – масив баз restOfFlightTime – залишок часу польоту без обслуговування speed – швидкість польоту	Булеве значення	Дана функція для пари точок повертає булеве значення, що показує чи доступна ця точка для польоту з заданої з урахуванням обмежень задачі
isAvailableBase	currentPoint – поточна точка unvisitedPoints – інша точка base – база restOfFlightTime – залишок часу польоту без обслуговування speed – швидкість польоту	Булеве значення	Дана функція повертає булеве значення, що показує чи доступна задана база для польоту із заданої точки з урахуванням залишку польотного часу
determineOutcome	probabilities – масив нормалізованих ймовірностей	Індекс ймовірності, яка справдилася	Дана функція випадковим чином обирає і повертає індекс ймовірності, яка справдилася

В реалізації алгоритму табу– пошуку використовується один клас – TabuList. Цей клас є репрезентацією табу– списку і містить поточні заборонені розв’язки. Конструктор класу TabuList приймає параметр *tenure*, який визначає наскільки довго розв’язок буде знаходитися в табу– списку. Клас має 2 публічні методи, опис яких представлено у таблиці 4.2.

Таблиця 4.2 – Опис методів класу TabuList

Метод	Вхідні параметри	Вихідні параметри	Опис
has	elem – розв’язок	Булеве значення	Даний метод повертає булеве значення, яка показує чи є заданий розв’язок у табу списку
append	elem – розв’язок	–	Даний метод додає заданий розв’язок у табу список

У таблиці 4.3 описані функції, які використовуються для програмної реалізації алгоритму табу– пошук.

Таблиця 4.3 – Опис функцій табу– пошуку.

Функція	Вхідні параметри	Вихідні параметри	Опис
createTabuSolver	params – об'єкт, що містить максимальну кількість ітерацій без покращення, кількість запусків пошуку та величину, яка визначає як довго розв'язок буде знаходитися в табу листі	Повертає функцію, яка і є розв'язувачем задачі	Дана функція створює розв'язувач задачі з переданими параметрами
Функція (розв'язувач), яку повернув createTabuSolver	pointsToObserve – масив точок, які потрібно облетіти startBase – початкова база anotherBase – друга база chargeTime – час обслуговування між польотами (у хвиликах) maxFlightTime – максимальний польотний час без обслуговування (у хвиликах) speed –	Повертає об'єкт, який містить поля: route – побудований маршрут fitness – значення цільової функції	Дана функція отримує вхідні дані, розв'язую задачу методом табу– пошуку і повертає результат розв'язання

Зм.	Лист	№ докум.	Підпис	Дата

Висновок до розділу

У даному розділі було описане програмне та технічне забезпечення системи, а саме детально розглянуті технології розробки, програмне середовище, мову розробки, використані для розробленої серверної частини системи фреймворки та бібліотеки. Наведено аргументацію вибору саме цих технологій, тенденції розвитку, перспективи та актуальність застосування для розробки системи на основі аналізу вимог. Також сформульовано вимоги до технічного забезпечення користувачів системи. За допомогою діаграм компонентів, діаграми послідовності описана та графічно проілюстрована архітектура системи. Також описана специфікація функцій серверної частини системи.

					<i>IS92.220БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		66

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Випробування програмного продукту

5.1.1 Мета випробувань

Метою проведення тестування є оцінка якості роботи серверної частини системи шляхом перевірки коректності та повноти роботи реалізованого функціоналу, відповідно до вимог, визначених у першому та другому розділах дипломного проєкту.

5.1.2 Результати випробувань

У комплексній системі маршрутизації БПЛА реалізовано багато функціонала. Частину реалізованого функціоналу можна вважати основним, тобто те, без чого система не функціонуватиме належним чином, наприклад логін користувачів, ведення вхідних даних, розв'язання задачі алгоритмом, відображення результатів і т.д..

Частину функціонала можна вважати допоміжним, тобто таким, що підвищує зручність взаємодії користувача з системою, наприклад, графічна візуалізація маршрутів, проведення експерименту і т.д.. Проведемо тестування частини основного функціонала, де важливою є коректна робота серверної частини системи та реалізованого алгоритму табу– пошуку.

Отже, протестуємо успішні («sunny– day» сценарії) випадки взаємодії користувача з наступним функціоналом системи:

- завантаження файлу з результатами розв'язання задачі маршрутизації БПЛА, таблиця 5.1;
- завантаження файлу з результатами проведення експерименту, таблиця 5.2;
- розв'язання задачі маршрутизації БПЛА методом табу– пошуку, таблиця 5.3;
- проведення експерименту для методу табу– пошуку, таблиця 5.4;

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		67

– авторизація користувача у системі за сторони сервера – вхід у обліковий запис, таблиця 5.5;

– авторизація користувача у системі зі сторони сервера – реєстрація користувача, таблиця 5.6.

Таблиця 5.1 – Завантаження файлу з результатами розв’язання задачі маршрутизації БПЛА

Мета тестового сценарію	Перевірка функціонала завантаження файлу з результатами розв’язання задачі маршрутизації БПЛА
Початковий стан системи	Авторизований користувач (дослідник або звичайний користувач) знаходиться на сторінці розв’язання задачі маршрутизації БПЛА
Вхідні дані	Характеристики БПЛА, координати цілей

<p>Кроки тестового сценарію</p>	<ol style="list-style-type: none"> 1. Користувач задає координати пунктів (двох баз та пунктів обльоту) та характеристики БПЛА (час перезаряджання, максимальний час польоту без перезаряджання, швидкість руху) вручну або шляхом зчитування з файлу 2. Користувач ініціює розв'язання задачі 3. Після того як система надсилає вхідні дані на сервер, на сервері відбувається розв'язання задачі 4. Клієнтська частина системи отримує від серверної результат розв'язання задачі, відображає його користувачу 5. Користувач ініціює завантаження результатів на пристрій 6. Після того як сервер отримує запит від клієнтської частини системи на завантаження файлу з результатами, на серверній частині системи формується файл JSON формату з результатами розв'язання 7. Система завантажує на пристрій користувача файл з результатами розв'язання задачі
<p>Очікуваний результат</p>	<p>У завантаженому на пристрій користувача файлі JSON формату є такі дані, як впорядкований набір пунктів, що відповідає побудованому розв'язувачем маршруту, значення цільової функції (сумарно необхідного часу на проходження, враховуючи перезаряджання) побудованого маршруту. Дані, записані до файлу збігаються з результатами розв'язання задачі.</p>

Стан системи після проходження тестового сценарію	На пристрій користувача завантажено файл з такими даними, як впорядкований набір пунктів, що відповідає побудованому розв'язувачем маршруту, значення цільової функції (сумарно необхідного часу на проходження, враховуючи перезаряджання) побудованого маршруту. Дані, записані до файлу збігаються з результатами розв'язання задачі.
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.2 – Завантаження файлу з результатами проведення експерименту

Мета тестового сценарію	Перевірка функціонала завантаження файлу з результатами експерименту
Початковий стан системи	Авторизований як дослідник користувач знаходиться на сторінці проведення експерименту
Вхідні дані	Параметри експерименту та алгоритму, згенеровані задачі

Кроки тестового сценарію

1. Користувач вручну задає параметри експерименту (розмірність задачі, тобто кількість пунктів обльоту у згенерованих задачах, кількість прогонів експерименту, тобто задач для розв'язання, алгоритм розв'язання) та параметри алгоритму (свої для кожного з алгоритмів системи)
2. Користувач ініціює розв'язання задачі
3. Після того як система надсилає вхідні дані на сервер, на сервері відбувається процес проведення експерименту
4. Клієнтська частина системи отримує від серверної результат експерименту, відображає його користувачу
5. Користувач ініціює завантаження результатів на пристрій
6. Після того як сервер отримує запит від клієнтської частини системи на завантаження файлу з результатами, на серверній частині системи формується файл JSON формату з результатами проведення експерименту
7. Система завантажує на пристрій користувача файл з результатами проведення експерименту

Стан системи після проходження тестового сценарію	Система відобразила результат, який повернула серверна частина, і за результатами алгоритмом побудовано прийнятний маршрут, а з графічної візуалізації побудованого маршруту видно, що розв'язок задовільняє умови та порядок відвідування пунктів.
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.4 – Проведення експерименту для методу табу– пошуку

Мета тестового сценарію	Перевірка функціонала проведення експерименту для методу табу– пошуку
Початковий стан системи	Авторизований дослідник, знаходиться на сторінці проведення експерименту
Вхідні дані	Згенеровані вхідні дані генератором задач
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Дослідник задає такі параметри експерименту як кількість прогонів (задач, що мають бути розв'язані) та розмірність розв'язуваних у ході експерименту задач (кількість пунктів обльоту у згенерованих задачах) 2. Дослідник обирає табу– пошук як метод для проведення експерименту 3. Дослідник коректно заповнює усі параметри табу– пошуку (кількість прогонів на кожній ітерації табу– пошуку, максимальний розмір

Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Користувач вводить логін та пароль від зареєстрованого у системі облікового запису (за замовчуванням на формі авторизації відкрито вхід до акаунту, а не реєстрація) 2. Користувач ініціює вхід до акаунту відповідною кнопкою 3. На сервер приходить запит на вхід до облікового запису з логіном та паролем, для яких був ініційований логін 4. Сервер по унікальному логіну знаходить користувача у базі даних 5. Сервер валідує переданий пароль для знайденого за унікальним логіном у базі даних користувача та повертає на клієнтську частину системи відповідь про успішний вхід до системи
Очікуваний результат	Отримання клієнтською частиною системи від серверної частини системи відповіді про успішну авторизацію користувача
Стан системи після проходження тестового сценарію	Клієнтська частина системи отримала від серверної частини системи відповідь про успішну авторизацію користувача, і відповідно знавігувала зареєстрованого користувача з форми авторизації на сторінку розв'язання задачі
Результат проходження тестового сценарію	Пройдено успішно

	6. Користувача успішно зареєстровано та перенаправлено на сторінку розв'язання задачі з правами, що відповідають типу користувача, для якого була ініційована реєстрація
Очікуваний результат	Створення у базі даних нового користувача з логіном, паролем та типом користувача, для яких ініційовано тестовий сценарій. На клієнтській стороні отримано відповідь про це та перенавігровано користувача на сторінку розв'язання задачі з відповідними правами (під роллю звичайного користувача або дослідника)
Стан системи після проходження тестового сценарію	У базі даних створено користувача з логіном, паролем та типом користувача, для яких була ініційована реєстрація. Клієнтська частина системи отримала від серверної частини системи відповідь про успішну реєстрацію користувача, і перенавігувала зареєстрованого користувача з форми авторизації на сторінку розв'язання задачі з відповідними правами (дослідника або звичайного користувача)
Результат проходження тестового сценарію	Пройдено успішно

Висновок до розділу

У даному розділі було сформульоване детальне керівництво користувача, що описує процеси взаємодії з системою як звичайного користувача для

									Арк.
									78
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220BAK.004 ПЗ</i>				

розв'язання поставленої задачі, так і дослідника для проведення дослідження роботи розроблених алгоритмів. Проведено випробування розробленої системи. Відповідно до сформульованої мети випробувань описано тестові сценарії та загальні положення тестування, наведено результати випробувань. Мануальне тестування системи за складеними сценаріями тестування показало, що система працює відповідно до вимог та очікувань.

					<i>IS92.220БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		79

ВИСНОВКИ

В результаті дипломної роботи була розроблена серверна частина інформаційної системи маршрутизації для безпілотних літальних апаратів (БПЛА) з розв'язанням задачі за допомогою табу– пошуку. Застосування розробленої системи дозволяє ефективно складати маршрути БПЛА, зменшуючи час та ресурси, необхідні для цього процесу.

Було проведено аналіз предметного середовища, огляд наявних аналогів, визначено ціль розробки, задачі та завдання, які необхідно виконати для досягнення цілі. Описано інформаційне забезпечення, яке складається з вхідних і вихідних даних та бази даних.

Також було створено математичне забезпечення системи, включаючи формулювання змістовної та математичної постановок задачі. Проаналізовано методи розв'язання задач оптимізації, був обраний метод табу– пошуку для розв'язання задачі. Було описано розроблений алгоритм методу розв'язання та його параметризація, а також наведено приклад результатів його роботи.

Описано програмне та технічне забезпечення системи. Було визначено засоби розробки, вимоги до технічного забезпечення та архітектуру програмного забезпечення. Також було надано опис класів, функцій та методів програмного забезпечення. Проведено випробування програмного продукту з метою перевірки його працездатності та описано мету випробувань, загальні положення та наведені результати.

Отримані результати свідчать про можливість ефективного використання розробленої системи маршрутизації для БПЛА в реальних умовах. Вона дозволяє автоматизувати процес складання маршрутів, що призводить до зниження часу та ресурсів, необхідних для планування маршрутів.

У подальшому можливе розширення системи шляхом додавання нових алгоритмів та методів маршрутизації, а також вдосконалення інтерфейсу користувача для забезпечення зручного взаємодії з системою. Також можлива інтеграція системи з іншими компонентами автоматизованої системи управління

									Арк.
									80
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220BAK.004 ПЗ</i>				

БПЛА для створення повноцінного комплексу, що забезпечує комплексне керування та моніторинг польотів БПЛА.

Результати цього дослідження можуть бути корисні для науковців, інженерів та фахівців, які працюють у галузі авіаційних систем та безпілотних літальних апаратів. Розроблений алгоритм маршрутизації може бути використаний як основа для подальших досліджень та розробок в цій галузі.

					<i>IS92.220BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		81

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The History and State of Vehicle Routing [Електронний ресурс] // Morton Enterprises Inc. – 2021. – Режим доступу до ресурсу: <https://supplychaingamechanger.com/the-history-and-state-of-vehicle-routing/>.
2. Gilbert Laporte. Vehicle routing: historical perspective and recent contributions / Gilbert Laporte, Paolo Toth, Daniele Vigo., 2013.
3. Paolo Toth. Vehicle routing: problems, methods, and applications / Paolo Toth, Daniele Vigo. – Philadelphia: Society for Industrial and Applied Mathematics, 2015. – (Second edition).
4. Л. Ф. Гуляницький. До класифікації задач маршрутизації транспортних засобів / Л. Ф. Гуляницький, А. А. Коткова., 2020. – 12 с. – (Наук. вісник Ужгород. ун-ту). – (вип. 36, № 1).
5. Brausy O & Gendreau M. Tabu Search heuristics for the Vehicle Routing Problem with time windows. – 2010. – pp. 454– 464.
6. Девід Рамель. VS Code і Visual Studio Rock the 2022 Stack Overflow Developer Report [Електронний ресурс] / Девід Рамель // Visual Studio Magazine. – 2022. – Режим доступу до ресурсу: <https://visualstudiomagazine.com/articles/2022/06/23/stack-overflow-2022-survey.aspx>.
7. Ashwin D. What Is Nest.JS? Why Should You Use It? [Електронний ресурс] / Dua Ashwin. – 2022. – Режим доступу до ресурсу: <https://www.turing.com/blog/what-is-nest-js-why-use-it/>.
8. Dr. Axel Rauschmayer. Tackling TypeScript. Upgrading from JavaScript / Dr. Axel Rauschmayer., 2021. – 67 с.
9. Remo H. Jansen. Learning TypeScript / Remo H. Jansen., 2015. – 368 с.
10. Node.js is an open-source, cross-platform JavaScript runtime environment. [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://nodejs.org/en>.

										Арк.
										82
Зм.	Лист	№ докум.	Підпис	Дата	<i>IS92.220БАК.004 ПЗ</i>					

11. Magaji J. The History of Node.js [Электронный ресурс] / Jethro Magaji. – 2020. – Режим доступа до ресурсу: <https://www.section.io/engineering-education/history-of-nodejs/>.

12. Arek Nawo. Is TypeScript the future of web development? [Электронный ресурс] / Arek Nawo. – 2020. – Режим доступа до ресурсу: <https://dev.to/areknawo/is-typescript-the-future-of-web-development-3jl>.

13. Jason Sturges. JavaScript, TypeScript, and the Future of Modern Web [Электронный ресурс] / Jason Sturges. – 2022. – Режим доступа до ресурсу: <https://jasonsturges.medium.com/javascript-typescript-and-the-future-of-modern-web-78a97b1a2b1e>.

14. M. Weststrate, P. Podila. MobX Quick Start Guide. Packt Publishing ,2018.

15. Arancio S. ReactJS: A brief history [Электронный ресурс] / Stephen Arancio –2021. – Режим доступа до ресурсу: <https://medium.com/@sjarancio/reactjs-a-brief-history-3c1e969a477f>.

16. Anirudh Bhardwaj. Web Development Using Frameworks: Why Does It Matter? [Электронный ресурс] / Anirudh Bhardwaj // CodlesERP. – 2021. – Режим доступа до ресурсу: <https://erpsolutions.oodles.io/blog/web-development-frameworks-benefits/>.

17. DroneDeploy [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://drone.ua/anch/dronedeploy/>.

18. Wang W. Bayesian Optimization Concept Explained in Layman Terms. [Электронный ресурс] / Wei Wang – 2020. – Режим доступа до ресурсу:<https://towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f> .

19. Chen B. A Practical Introduction to Grid Search, Random Search, and Bayes Search. [Электронный ресурс] / – 2021. – Режим доступа до ресурсу:<https://towardsdatascience.com/a-practical-introduction-to-grid-search-random-search-and-bayes-search-d5580b1d941d> .

									Арк.
									83
Зм.	Лист	№ докум.	Підпис	Дата	<i>IC92.220БАК.004 ПЗ</i>				

20. Schwartz D. Why Drones Are the Future of Outdoor Search and Rescue. [Електронний ресурс] / – 2021. – Режим доступу до ресурсу:<https://www.outsideonline.com/outdoor-adventure/exploration-survival/drones-search-rescue/>.

21. Гуляницький Л.Ф., Мулеса О. Ю. Прикладні методики комбінаторної оптимізації: навч. посіб.: Видавничо-поліграфічний центр "Київський університет", 2016. – 142 с.

					<i>IS92.220БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		84