

**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 методом локального пошуку»**

Виконала:

студентка IV курсу, групи ІС-92

Зінкова Ксенія В'ячелавівна _____

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.
Студентка _____

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

Зінковій Ксенії В'ячеславівні

1. Тема проєкту «Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів методом локального 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: мови програмування TypeScript, JavaScript, редактор вихідного коду Visual Studio Code, бібліотеки MobX, React.js.

4. Зміст пояснювальної записки:

Проаналізувати поставлену задачу, розробити математичну та змістовні постановки, проаналізувати алгоритми розв'язання задачі, розробити алгоритм локального пошуку для розв'язання поставленої задачі, проаналізувати його ефективність та підібрати параметри для кращої роботи алгоритму, імплементувати клієнську частину компоненти проведення експериментів інформаційної системи маршрутизації БПЛА.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

- діаграма варіантів використання;
- діаграма компонентів;

- діаграма діяльності;
- схема алгоритму штучної бджолоїної колонії.

6. Дата видачі завдання 13.02.2023

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз та уточнення завдання на дипломний проєкт	24.04.2023	
2	Аналіз предметної області, ознайомлення з аналогами, розробка функціональної моделі	26.04.2023	
3	Аналіз та проєктування математичного забезпечення	01.05.2023	
4	Проєктування та опис інформаційного забезпечення, реалізація програмного забезпечення	04.05.2023	
5	Реалізація програмного забезпечення	08.05.2023	
6	Опис програмного та технічного забезпечення	10.05.2023	
7	Оформлення звіту дипломного проєкту	17.05.2023	
8	Подання ДП на попередній захист	05.05.2023	
9	Подання ДП на основний захист	12.05.2023	
10	Подання ДП рецензенту	13.05.2023	

Студент

Ксенія ЗІНКОВА

Керівник

Олена ЖДАНОВА

АНОТАЦІЯ

Зінкова К.В. Система маршрутизації БПЛА. Підсистема складання маршрутів методом локального пошуку. КПІ ім. Ігоря Сікорського, Київ, 2023.

Пояснювальна записка дипломного проєкту складається з п'яти розділів, а також 18 рисунків, 18 таблиць, 4 графічні матеріали, 37 джерел.

Ключові слова: безпілотний літальний апарат, задача маршрутизації транспортних засобів, локальний пошук, метод штучної бджолоїної колонії, план обльоту цілей, схрещування маршрутів.

Об'єктом розробки є система маршрутизації БПЛА.

Метою розробки є складання маршрутів польоту БПЛА, за яких досягає мінімуму час обльоту заданої множини цілей.

Дипломний проєкт присвячений розробці частини клієнтської сторони комплексної системи маршрутизації БПЛА, а також проєктуванню, реалізації, та аналізу алгоритму локального пошуку. На основі аналізу класифікації задач транспортної маршрутизації та алгоритмів їх розв'язання, обрано алгоритм штучної бджолоїної колонії. Алгоритм адаптовано під математичну постановку задачі та реалізовано. Проведення підбору параметрів дозволило підібрати такі параметри алгоритму, що працюють краще саме для задачі маршрутизації БПЛА. З підібраними параметрами було проведено дослідження роботи алгоритму, що показало ефективні результати роботи на обраних (на основі змістовної постановки задачі) розмірностях.

Розроблена система може бути корисною для використання в реальних умовах, а також при автоматизації процесу складання маршрутів для подібних чи аналогічних систем.

SUMMARY

Zinkova K.V. UAV routing system. The subsystem of route planning by local search. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

Explanatory note of the diploma project consists of five chapters, as well as 18 figures, 18 tables, 4 graphic materials, 37 sources.

Keywords: artificial bee colony method, target flight plan, local search, route crossing, unmanned aerial vehicle, vehicle routing problem.

The object of development is a UAV routing system.

The purpose of the development is to compile UAV flight routes that minimize the time of flying around a given set of targets.

The diploma project is devoted to the development of the client side of an integrated UAV routing system, as well as the design, implementation, and analysis of a local search algorithm. Based on the analysis of the classification of transport routing problems and algorithms for solving them, the artificial bee colony algorithm was chosen. The algorithm is adapted to the mathematical formulation of the problem and implemented. Parameter selection allowed us to select the algorithm parameters that work best for the UAV routing problem. With the selected parameters, the algorithm was studied, which showed effective results on the selected (based on the meaningful statement of the problem) dimensions.

The developed system can be useful for use in real conditions, as well as for automating the process of route planning for similar or similar systems.

Номер рядка	Формат	Позначення	Найменування	Кількість аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC92.050БАК.004 ПЗ	Пояснювальна записка	102		
6	A3	IC92.050БАК.004 Д1	Діаграма варіантів	1		
7			використання			
8	A3	IC92.050БАК.004 Д2	Діаграма компонентів	1		
9	A3	IC92.050БАК.004 Д3	Діаграма діяльності	1		
10	A3	IC92.050БАК.004 Д4	Схема алгоритму	1		
11			штучної бджолоїної колонії			
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						

					IC92.050БАК.004 ТП		
Зм.	Аркуш	№ докум.	Підп.	Дата			
Розроб.		Зінкова К.В.			Літ.	Аркуш	Аркушів
Керівн.		Жданова О.Г.			т	1	1
Затв.					КПІ ім. Ігоря Сікорського Група IC-92		
					Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів методом локального пошуку. Відомість проекту		

**Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система маршрутизації
БПЛА. Підсистема складання маршрутів методом
локального пошуку»**

ЗМІСТ

ВСТУП.....	3
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	8
1.1 Опис предметного середовища	8
1.1.1 Опис процесу діяльності.....	10
1.1.2 Опис функціональної моделі.....	11
1.2 Огляд наявних аналогів.....	12
1.3 Постановка задачі	15
1.3.1 Призначення розробки	15
1.3.2 Цілі та задачі розробки	15
Висновок до розділу	16
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	18
2.1 Вхідні дані	18
2.2 Вихідні дані	19
2.3 Опис структури масивів інформації та бази даних.....	19
Висновок до розділу	20
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	21
3.1 Змістовна постановка задачі.....	21
3.2 Математична постановка задачі.....	22
3.3 Обґрунтування методу розв'язання.....	25
3.3.1 Класифікація задач транспортної маршрутизації	25
3.3.2 Комбінаторна складність задачі транспортної маршрутизації.....	33
3.3.3 Класифікація методів розв'язання задачі транспортної маршрутизації	34
3.3.4 Вибір методу розв'язання для задачі маршрутизації БПЛА.....	36
3.4 Опис методу розв'язання	44

					<i>IS92.050BAK.004 ПЗ</i>			
Зм.	Лист	№ докум.	Підпис		Інформаційна система маршрутизації БПЛА. Підсистема складання маршрутів методом локального пошуку. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Зінкова К.В.					Т	2	102
Перевірив	Жданова О.Г.					КПІ ім. Ігоря Сікорського		
Затв.								

3.4.1	Підбір параметрів методу розв'язання задачі	51
3.4.1.1	Аналіз результатів роботи розробленого методу розв'язання	55
	Висновок до розділу	57
4	ПРОГРАМНЕ І ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	59
4.1	Засоби розробки	59
4.1.1	Мова розробки програмного забезпечення	59
4.1.2	Середовище розробки програмного забезпечення	62
4.1.3	Технології розробки програмного забезпечення	64
4.2	Вимоги до технічного забезпечення	67
4.3	Архітектура програмного забезпечення	68
4.3.1	Діаграма компонентів	68
4.3.2	Специфікація функцій	70
	Висновок до розділу	77
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	78
5.1	Керівництво користувача	78
5.2	Випробування програмного продукту	84
5.2.1	Мета випробувань	85
5.2.2	Результати випробувань	85
	Висновок до розділу	95
	ВИСНОВКИ	96
	ПЕРЕЛІК ПОСИЛАНЬ	98

ВСТУП

На сьогодні, питання оптимізації руху та розподілу транспортних потоків в Україні актуальне в першу чергу для організації перевезень вантажу у внутрішньому та міжнародному сполученнях. Завдяки автоматизації процесів налагодження взаємодії різних видів транспорту, оптимізації маршрутів транспортних перевезень, можна досягнути зменшення витрат ресурсів – об'ємів пального або електроенергії, кількості водіїв та транспортних засобів, а також пришвидшення процесів перевезень.

Задачі транспортної маршрутизації все більше привертають увагу науковців, оскільки зараз сфера логістики стикається з великою кількістю проблем.

Так, з початку широкомасштабного вторгнення в Україну росії, частково заблокована робота морських портів України, а саме через морські порти експортувалося 70% усіх вантажів на суму близько 47 мільярдів доларів [1]. Відповідно, зростає навантаження на інші види транспорту для перевезення вантажу, зокрема на залізничні перевезення.

Частка вантажів, що транспортувалися по морю, на період заблокування портів перевозиться залізницею, але пропускна здатність пунктів перетину кордону України через це зазнає значних навантажень, через такі перешкоди як необхідність витратити на технічний огляд поїзда на кордоні в середньому п'ять годин [2], необхідність заміни пристроїв зчеплення вагонів, колісних візків. Так, лише п'ять із тринадцяти переходів державного кордону України підтримують перехід між українською колією, ширина якої складає 1520 мм та євроколією, завширшки 1435 мм [3].

Крім цього, європейська залізнична система не розрахована на таку пропускну спроможність, оскільки у Європі частка залізничних перевезень складає 15 – 35% з загальної структури транспортної системи, у той час, як в Україні ця цифра складає 65% [4]. Також, у 2020 році спостерігається спад

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		3

обсягів перевезення вантажів за різними видами транспорту в Україні, що може бути зумовлено пандемією COVID-19 [1].

Крім того, як ніколи актуальною на цей момент є проблема з живленням транспортних засобів. Ринок палива суттєво зменшився, оскільки до широкомасштабного вторгнення в Україну росії, на рік Україна споживала 12 млн тонн нафтопродуктів, з них 9,5 млн тонн Україна імпортувала. Значна частка цих нафтопродуктів – 7 млн тонн це дизпаливо, з якого з Білорусі постачалося 43%, з росії – 30%, з Литви – 10%, а 17% постачалося через морські порти, що зараз частково заблоковані [5]. Зараз частина каналів імпорту перекрито. Отже, через брак палива необхідно з підвищеною уважністю ставитися до побудови найкоротших маршрутів та до мінімізації холостого пробігу транспортних засобів.

Задача транспортної маршрутизації є актуальною для багатьох компаній та організацій у всіх галузях, які залежать від перевезень та доставлення. Ось кілька прикладів, чому ця задача є актуальною:

— електронна комерція – компанії електронної комерції, які потребують оптимальної організації доставлення товарів до клієнтів; це може включати в себе визначення маршрутів для доставлення великої кількості замовлень в різних місцях, що дозволяє економити на витратах та забезпечує більш ефективну доставку;

— логістика – вона використовується для планування маршрутів транспортних засобів та розподілу вантажів між різними місцями; це допомагає економити на часі, паливі та інших витратах;

— транспорт та доставка – компанії, які займаються транспортом та доставкою можуть використовувати транспортну маршрутизацію для оптимального розподілу ресурсів та оптимізації маршрутів доставки, що забезпечує економію на витратах та покращує якість обслуговування.

Крім транспортних перевезень, задача маршрутизації транспортних засобів постає перед суспільством у сферах перевезення людей, наприклад

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

у таких областях:

- регулярні перевезення громадським, міжміським та міжнародним транспортом;
- служби таксі;
- перевезення працівників сервісного обслуговування;
- спеціалізовані сервіси транспортування тварин;
- військові перевезення.

Логістика, невід’ємною частиною якої є транспортна маршрутизація, в сучасному світі має важливе значення для успішної діяльності бізнесу, особливо в контексті глобалізації та електронної комерції. Основні причини, чому логістика зараз актуальна як ніколи, включають:

— зростання глобалізації – вільний рух товарів та послуг між країнами вимагає від підприємств складніших логістичних рішень для забезпечення постачання товарів та послуг своїм клієнтам вчасно та з заданим якісним рівнем;

— збільшення обсягу електронної комерції – зростання популярності онлайн-шопінгу вимагає від бізнесу швидкого та ефективного доставлення товарів клієнтам, що також потребує складних логістичних процесів;

— складніші вимоги споживачів – сучасні споживачі очікують не тільки вчасної доставки, але й індивідуального підходу до своїх потреб та бажань; це вимагає від бізнесу складніших логістичних рішень для забезпечення персоналізованого обслуговування клієнтів;

— постійні зміни в технологіях та тенденціях – розвиток технологій та зміни в уявленнях про зручність та ефективність процесів змушують бізнес адаптуватись та шукати нові, більш ефективні способи логістики.

Враховуючи вищезазначене, все більше підвищується інтерес до задач транспортної маршрутизації та необхідність розробки новітніх підходів для їх розв’язання.

Як державні, так і приватні компанії зацікавлені у мінімізації витрат на

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

логістику шляхом впровадженні у бізнес-процеси автоматизованих систем розрахунку оптимальних маршрутів, а також оптимізації витрат на ресурси, що необхідні для забезпечення перевезень.

Даний дипломний проєкт присвячений задачі маршрутизації БПЛА, яка є актуальною як для транспортної маршрутизації загалом, так при використанні БПЛА у різних галузях.

Актуальність проілюстрована вище конкретними прикладами, досягненнями, проблемами та викликами, що постають зараз перед суспільством у сфері транспортної маршрутизації загалом, та застосуванні БПЛА у промисловості. Використання БПЛА для розв'язання задачі транспортної маршрутизації є перспективним напрямом розвитку галузі. Це забезпечує можливість зниження вартості доставки, скорочення часу перевезення, покращення безпеки та зниження впливу на довкілля. Однак, для ефективного використання БПЛА в транспортній маршрутизації потрібні алгоритми, які враховують особливості цих систем та обмеження, пов'язані з їх використанням. Все вищезазначене детально розглядається у проєкті.

Метою розробки інформаційної системи є складання маршрутів польоту БПЛА, за яких досягає мінімуму час обльоту заданої множини цілей.

Даний проєкт є частиною комплексної роботи. Він присвячений розробці частини функціоналу сторони клієнта. Для досягнення мети необхідно реалізувати наступні модулі:

- а) модуль дослідження алгоритмів розв'язання задачі;
 - 1) ведення параметрів алгоритмів;
 - 2) ведення параметрів експериментів;
 - 3) проведення експериментів з дослідження ефективності алгоритмів розв'язання задачі;
 - 4) візуалізація результатів проведення експериментів;
- б) модуль складання маршрутів польоту БПЛА для заданої множини цілей методом локального пошуку.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

Завдання дипломного проектування такі:

— провести аналіз предметної області, існуючих підходів до розв’язання, алгоритмів та їх поєднань, аналогів програмного забезпечення транспортної маршрутизації;

— розробити алгоритм локального пошуку розв’язання поставленої задачі, що враховує особливості розв’язуваної задачі;

— виконати програмну реалізацію алгоритму локального пошуку;

— підібрати параметри алгоритму, що забезпечують його ефективну роботу;

— імплементувати реалізацію алгоритму як частину функціоналу інформаційної системи маршрутизації БПЛА;

— провести тестування програмного забезпечення на згенерованих даних та проаналізувавши результати, дослідити релевантність результатів роботи розробленого алгоритму.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

На сьогодні, все ширшого застосування набуває використання безпілотних літальних засобів (БПЛА). Наприклад, використання БПЛА зараз є розповсюдженою практикою у таких сферах:

— мілітарний сектор – БПЛА використовуються для збору розвідувальної інформації, проведення навчань, пошуку та знищення цілей, контролю над територіями та інших військових завдань;

— громадська безпека – БПЛА застосовуються для нагляду за громадською безпекою, таких як спостереження за дорожнім рухом, пожежами, повеннями, катастрофами, пошуком загублених людей та ін.;

— агропромисловість – БПЛА можуть допомогти аграрним виробникам в зборі даних про стан ґрунту та рослин, контролі над зрошенням, виявленні шкідників та іншому;

— транспорт – БПЛА можуть використовуватися для транспортування товарів та пошти, допомоги в рятувальних операціях, охороні меж та ін.;

— медицина – БПЛА можуть використовуватися для швидкої доставки медикаментів, вакцин, апаратів штучної вентиляції легенів тощо у віддалені та важкодоступні місця;

— промисловість – БПЛА можуть використовуватися для інспекції труднодоступних і небезпечних місць, включаючи промислові споруди та інфраструктуру, які підлягають регулярному обслуговуванню та ремонту.

Для дипломного проєкту було обрано мілітарний сектор, але задача може бути адаптована під частину з вищеперелічених галузей.

Є деякий військовий підрозділ, якому для виконання задач потрібно на регулярні основі обстежувати певні набори цілей. Оскільки єдиним можливим способом є обстеження з повітря, то для цього процесу застосовується БПЛА.

Набори цілей довільно розташовуються на мапі, задаються їх

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

координати.

Також задаються координати двох довільно розташованих депо, які відповідають за зберігання, та перезарядку БПЛА. Під перезарядкою мається на увазі комплекс дій, спрямований на повну готовність БПЛА до автономного польоту, включаючи заміну батареї. Перезарядка займає фіксований час, що залежить від особливостей моделі БПЛА, та особливостей організації роботи в депо. Тому, час перезарядки є параметром, який надає користувач. Вважається, що перед початком обстеження цілей БПЛА заряджений.

Користувач задає середню швидкість БПЛА та час автономної роботи БПЛА, тобто максимальний час, який БПЛА може провести у повітрі без перезарядки. Ці характеристики можуть відрізнятися для різних БПЛА, враховуючи модель та технічний стан.

Враховуючи обмеження на максимальний час автономної роботи БПЛА, відвідати усі цілі одним цілісним (без проміжних повернень у депо) маршрутом неможливо, а отже загальний маршрут БПЛА розбивається на підмаршрути, між якими БПЛА повертається у одне з депо для перезарядки.

Для військового підрозділу несуттєво на якому з двох депо відбуватиметься перезарядка БПЛА, і на якому з двох депо він завершить маршрут. Але потрібно враховувати, що БПЛА зберігається в одному депо, а отже саме з нього він починає проходження плану відвідування цілей.

Для того, щоб задача була розв'язувана, цілі повинні бути розташовані враховуючи таке обмеження: координати кожної цілі обираються так, щоб часу автономної роботи БПЛА, з врахуванням середньої швидкості БПЛА, вистачило на проходження БПЛА шляху від найближчого до цілі депо безпосередньо до самої цілі, і назад до депо.

Також, для того, щоб задача була розв'язувана, депо повинні бути розташовані враховуючи таке обмеження: координати депо обираються так, щоб щоб часу автономної роботи БПЛА, з врахуванням середньої швидкості

БПЛА, вистачило на проходження БПЛА шляху від одного депо до

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

іншого.

Необхідно спланувати план відвідування цілей так, щоб мінімізувати ресурси, що витратить БПЛА на проходження цього плану. Відповідно, потрібно мінімізувати кількість перезарядок. Надалі використовуватимемо назву мінімізації сумарного часу проходження плану відвідування цілей (що складається з часу, витраченого на безпосереднє відвідування цілей, а також часу, витраченого не перезарядку між маршрутами).

1.1.1 Опис процесу діяльності

Використовуючи UML-діаграму процесу діяльності, опишемо послідовність та альтернативні потоки дій, які виконуються при роботі з системою.

Для того щоб розпочати роботу з системою, користувач повинен увійти у свій обліковий запис. Для цього користувач вводить дані для входу – логін та пароль та відправляє їх системі на валідацію. Якщо система успішно провалідувала дані для входу, користувач заходить у систему. Інакше, при виникненні помилок, система повертає користувачу помилку та не дозволяє увійти в акаунт використовуючи ці дані для входу.

Якщо у користувача немає облікового запису, обліковий запис створюється шляхом реєстрації у системі. Для цього користувач повинен задати унікальний логін, пароль до акаунту та обрати роль для створюваного облікового запису (користувач або дослідник).

У системі є дві ролі – звичайний користувач, та особлива роль звичайного користувача, дослідник. Детальніше про функціонал, що доступний кожній з ролей, розповідається у пункті 1.1.2 дипломного проєкту.

Загалом, функціонал системи можна розділити на два незалежні для користувача модулі – модуль розв'язання задачі транспортної маршрутизації та модуль проведення експериментів. Опишемо процес діяльності для модуля

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

проведення експериментів, реалізація клієнтської частини якого з задачею даної індивідуальної частини комплексного дипломного проєкту.

Процес роботи з модулем експериментів починається з того, що досліднику потрібно задати такі параметри експерименту як кількість прогонів (тобто кількість задач, які буде розв'язано під час проведення експерименту), розмірність задач (усі задачі для експерименту генеруватимуться однакової розмірності, під розмірністю мається на увазі кількість цілей, яку потрібно відвідати) та обирається алгоритм.

Після того як користувач ініціює розв'язання задачі, система зчитує надані користувачем дані, випадковим чином генерує на основі них дані для експерименту, а саме потрібну кількість наборів цілей з координатами для них та для депо і розв'язує усі згенеровані задачі.

Після розв'язання, система виводить на екран для користувача результати, а саме час, за який БПЛА пройде маршрут для кожної зі згенерованих задач, середній час проходження маршруту по усім розв'язкам задач, та середньоквадратичне відхилення часу проходження маршруту.

Аналогічно до модуля розв'язання задачі маршрутизації, користувач за бажанням може ініціювати збереження результатів розв'язання у файл. Тоді система сформує відповідний файл з результатами і збереже його на пристрій користувача.

Діаграма діяльності для роботи з модулем проведення експериментів представлена у графічних матеріалах, кресленик IC92.050БАК.004 ДЗ.

1.1.2 Опис функціональної моделі

Головним актором, що взаємодіє з системою, є Користувач. Метою користувача є розв'язання задачі транспортної маршрутизації БПЛА.

Користувачу може бути надана особлива роль, роль Дослідника. Метою дослідника є аналіз ефективності розроблених алгоритмів системи шляхом проведення експериментів.

					<i>IC92.050БАК.004 ПЗ</i>	Арк.
						11
Зм.	Лист	№ докум.	Підпис	Дата		

Користувачу (включно з Дослідником) доступний наступний функціонал системи:

- а) вхід у систему;
- б) реєстрація у системі;
- в) розв'язання задачі.

При цьому функціонал розв'язання задачі розроблено у частині комплексного диплому Карпельової Ірини, у даному дипломному проєкті він використовується для розв'язання задачі маршрутизації БПЛА розробленим алгоритмом.

Користувачу з роллю Дослідника, окрім функціоналу, що доступний звичайному Користувачу, також доступні наступні функції системи:

- а) проведення експерименту;
 - 1) вибір алгоритму для розв'язання;
 - 2) задання кількості прогонів розв'язання;
 - 3) генерація випадкових даних для експерименту;
 - 4) перегляд результатів експерименту;
 - 5) завантаження результатів експерименту у файл формату

JSON.

Перелічені функції проілюстровані на діаграмі варіантів використання, розміщеній у графічних матеріалах, кресленик ІС92.050БАК.004 Д1. З діаграми можна побачити, як пов'язані між собою функції, які підфункції є обов'язковими для певних функцій (структурний зв'язок <include>, включає), а які є опціональними (структурний зв'язок <extend>, розширює). Також чітко видно, як співвідносяться між собою ролі звичайного користувача та дослідника. Права поступу до функціоналу дослідника розширюють права доступу до функціоналу звичайного користувача.

1.2 Огляд наявних аналогів

Один з прикладів програмного забезпечення для складання маршрутів

					<i>ІС92.050БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

дрона – Mission Planner. Це безкоштовне програмне забезпечення з відкритим вихідним кодом, яке дозволяє планувати та виконувати автономні місії з дронами, зокрема складати маршрути.

Mission Planner підтримує багато різних дронів, включаючи дрони з використанням проекту ArduPilot, такі як Pixhawk, APM та інші [32].

ArduPilot – це безкоштовне програмне забезпечення з відкритим вихідним кодом для автономних систем, зокрема для БПЛА, для автономної системи управління, що дозволяє контролювати різні параметри польоту дрону та виконувати автономні місії, включаючи складання маршрутів..Воно розробляється спільнотою розробників та підтримується міжнародною організацією ArduPilot, що забезпечує його постійну підтримку та оновлення [33].

ArduPilot може бути встановлене на багато різних типів БПЛА, включаючи мультикоптери, літаки, квадрокоптери, вертольоти та інші. Воно дозволяє контролювати багато різних параметрів польоту, таких як висота, швидкість, кут нахилу та інші, а також виконувати автономні місії та складати маршрути.

ArduPilot має вбудовану систему навігації, що дозволяє дрону самостійно навігувати в просторі, використовуючи GPS та інші датчики. Воно також підтримує різні режими польоту, включаючи ручне керування, автономний режим, підтримку місій та інші [33].

ArduPilot має відкритий вихідний код та активну спільноту розробників, що дозволяє користувачам змінювати та доповнювати функціональність програми, що дозволяє забезпечувати постійний розвиток та підтримку.

Mission Planner дозволяє користувачам визначати точки на мапі та створювати маршрути, які можуть бути виконані автоматично дроном. Крім того, Mission Planner дозволяє встановлювати параметри польоту, такі як висота та швидкість, а також забезпечує контроль за життєвим циклом дрона, включаючи стан батареї, обладнання та інші параметри [32].

Основними функціями Mission Planner є:

- введення маршрутних точок за допомогою Google Maps/Bing/відкритих карт вулиць/кастомних WMS;
- вибір команд та характеристик місії з випадаючих меню;
- завантаження файлів журналу місії для подальшого аналізу;
- налаштування параметрів автопілота для транспортного засобу;
- можливість взаємодії з комп'ютерним авіасимулятором для створення повноцінного програмного симулятора транспортного засобу;
- запуск власної симуляції з багатьма типами кадрів для всіх доступних апаратів ArduPilot [32].

Mission Planner має інтуїтивний інтерфейс користувача та детальну документацію, яка допомагає користувачам ознайомитися з усіма можливостями програмного забезпечення. На рисунку 1.1 наведено знімок інтерфейсу системи [32]. Воно може бути використане для виконання різних завдань, включаючи зйомку високоякісних фото та відео, обстеження місцевості, пошук втрачених людей та інші.



Рисунок 1.1 – Інтерфейс системи Mission Planner

Отже, Mission Planner є прикладом програмного забезпечення, яке дозволяє складати маршрути та виконувати автономні місії з дроном, а також контролювати різні параметри польоту для досягнення бажаних результатів.

На відміну від даної системи, проєктована у дипломній роботі у дипломному проєкті розробляється система, що націлена на планування складних маршрутів, тобто таких, що складаються з кількох підмаршрутів та за наявності кількох депо для БПЛА.

Також, система дипломного проєкту розроблена під конкретну постановку задачі, а отже відсутність опціонального функціоналу забезпечує швидке навчання користувача роботі з системою. Mission Planner має велику кількість функцій і налаштувань, що може ускладнити його освоєння для новачків. Інтерфейс програми може здаватися складним і заплутаним, що може викликати певні труднощі при першому знайомстві з нею.

І ще однією перевагою проєктованої системи перед Mission Planner є підтримуваність. Веб-застосунок, завдяки використанню сучасних засобів розробки, адаптований під більшість браузерів, а Mission Planner, як десктопний застосунок, підтримує Windows, але не надає офіційної підтримки для інших операційних систем, таких як macOS або Linux.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розроблюваної інформаційної системи є складання планів обльоту заданої множини цілей БПЛА з двома депо, для якого досягає мінімуму час відвідування цілей. У даній частині комплексного дипломного проєкту реалізована частина функціоналу клієнтської сторони системи.

1.3.2 Цілі та задачі розробки

Метою інформаційної системи є складання маршрутів польоту БПЛА, за яких досягає мінімуму час обльоту заданої множини цілей, що дозволить

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

підвищити швидкість відвідування усіх цілей та ефективне використання палива.

Для досягнення мети необхідно реалізувати наступні задачі у даній частині комплексного дипломного проєкту клієнтської сторони системи:

а) модуль дослідження алгоритмів розв'язання задачі;

1) ведення параметрів алгоритмів;

2) ведення параметрів експериментів;

3) проведення експериментів з дослідження ефективності алгоритмів розв'язання задачі;

4) візуалізація результатів проведення експериментів;

б) модуль складання маршрутів польоту БПЛА для заданої множини цілей методом локального пошуку.

В ході дипломного проєктування необхідно виконати наступні завдання:

— провести аналіз предметної області, існуючих підходів до розв'язання, алгоритмів та їх поєднань, аналогів програмного забезпечення транспортної маршрутизації;

— розробити алгоритм локального пошуку розв'язання поставленої задачі, що враховує особливості розв'язуваної задачі;

— виконати програмну реалізацію алгоритму локального пошуку;

— підібрати параметри алгоритму, що забезпечують його ефективну роботу;

— імплементувати реалізацію алгоритму як частину функціоналу інформаційної системи маршрутизації БПЛА;

— провести тестування програмного забезпечення на згенерованих даних та проаналізувавши результати, дослідити релевантність результатів роботи розробленого алгоритму.

Висновок до розділу

Виконано проєктування інформаційної системи маршрутизації БПЛА;

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						16
Зм.	Лист	№ докум.	Підпис	Дата		

описане предметне середовище, описані як загалом сфери застосування маршрутизації БПЛА, так і конкретно використання маршрутизації БПЛА для обстеження військових цілей; розглянуто та схематично проілюстровано діаграмою діяльності процес роботи системи та її взаємодію з користувачем, окремо для модуля розв'язання задачі маршрутизації БПЛА та для модуля проведення експериментів; розглянуто та проілюстровано на діаграмі використання ролі користувачів – звичайний користувач та дослідник, варіанти їхньої взаємодії з системою та їх співвідношення; проаналізовано аналог системи Mission Planner, виявлено переваги проєктованої системи над ним; поставлені цілі розробки, сформована мета, виокремлено задачі дипломного проєктування; визначено функції системи, які необхідно імплементувати.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Оскільки функціонал системи розподілено на два модулі: розв'язувач задачі маршрутизації БПЛА, та проведення експериментів, описуватимемо вихідні дані окремо для кожного з модулів.

Для модуля розв'язувача задачі маршрутизації БПЛА вхідними даними є:

- координати кожної цілі з набору цілей;
- координати кожного з двох депо;
- середня швидкість БПЛА;
- час автономної роботи БПЛА;
- час перезарядки БПЛА (однаковий для обох депо).

Через інтерфейс веб-застосунку користувач може ініціювати зчитування вхідних даних модуля розв'язувача з файлу JSON формату, або ввести вхідні дані вручну.

Для модуля проведення експериментів вхідними даними є:

- вибраний алгоритм;
- кількість прогонів розв'язувача (кожен прогон виконується на новому наборі пунктів);
- розмірність задачі (кількість цілей, спільне значення для кожного з прогонів).

Значення середньої швидкості БПЛА, часу автономної роботи БПЛА, та часу перезарядки БПЛА (однаковий для обох депо) для експерименту є статичними параметрами, оскільки метою експериментів буде порівняння параметрів алгоритму, відповідно порівняння результатів від прогону до прогону, і на результати не повинна впливати зміна цих трьох параметрів. Координати цілей та депо генеруються автоматично генератором.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

2.2 Вихідні дані

Оскільки функціонал системи розподілено на два модулі: розв'язувач задачі маршрутизації БПЛА, та проведення експериментів, описуватимемо вихідні дані окремо для кожного з модулів.

Для модуля розв'язувача задачі маршрутизації БПЛА вихідними даними є графічне представлення плану відвідування цілей, упорядкована множина цілей, у якій підмаршрути розмежовуються депо, та такі параметри як час проходження маршруту (включно і з сумарним часом, що потрібен на перезарядку БПЛА між маршрутами, і з сумарним часом, що потрібен на відвідування усіх цілей) і довжину маршруту.

Для модуля проведення експериментів вихідними даними є час проходження кожного з маршрутів (представлено списком, у якому наведено номер згенерованої задачі та час походження маршруту для неї відповідно), рекорд даного експерименту (тобто найкраще значення цільової функції по розв'язках усіх згенерованих задач) середній час проходження усіх маршрутів та середньоквадратичне відхилення часу проходження маршрутів прогону експерименту.

Додатково, як частину (окрім візуалізації графічного представлення побудованого плану відвідування цілей) вихідних даних розв'язувача задачі БПЛА, так і частину вихідних даних модуля проведення експериментів (окрім часу проходження кожного з маршрутів), користувач може зберегти у форматі JSON файлу.

2.3 Опис структури масивів інформації та бази даних

Оскільки я працюю над клієнтською складовою системи, а серверна частина розробляється у індивідуальній частині комплексного диплому проєкті Олександра Чорного, опис структури масивів інформації та бази даних наведено у підрозділі 2.3 та 2.4 дипломного проєкту Олександра Чорного.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						19
Зм.	Лист	№ докум.	Підпис	Дата		

У своїй роботі я використовую ці напрацювання.

Зі сторони клієнтської частини системи, можна зазначити, що вхідні дані можна ввести двома способами: вручну, або зчитати з файлу формату JSON.

Результати як звичайного розв'язання задачі маршрутизації БПЛА користувачем, так і проведення експерименту, система виводить на екран та надає посилання на завантаження файлів (з відповідною структурою для результатів розв'язання, та результатів експерименту, описаною у дипломному проєкті) на пристрій користувача.

Висновок до розділу

У розділі визначено вхідні та вихідні дані для роботи системи, описано процес взаємодії користувача з файлами масивів інформації зі сторони клієнтської частини.

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		20

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Для дослідження було обрано задачу маршрутизації БПЛА для обльоту пунктів споживання (надалі називатимемо їх цілями). Ключова різниця цієї постановки від класичної, це те що немає обмеження на вантажопідйомність, оскільки цілі лише обстежуються, і відсутня потреба у транспортуванні товару до них. Виконується лише обліт цілей, без затримки у жодній з них.

Також, особливістю є те, що розглядатиметься задача транспортної маршрутизації з кількома пунктами виробництва (надалі називатимемо їх депо). Маємо два фіксовані та рівноправні депо, БПЛА повинен розпочати маршрут у визначеному депо. Заряджатися між маршрутами та завершити план обльоту цілей можна у будь-якому депо. Припускається, що перед початком обльоту цілей БПЛА вже заряджений та у розрахунки на враховується підзарядка в кінці обльоту маршрутів. Кожна ціль відвідується один і лише один раз. Приклади розв'язків задачі проілюстровано на рисунку 3.1.

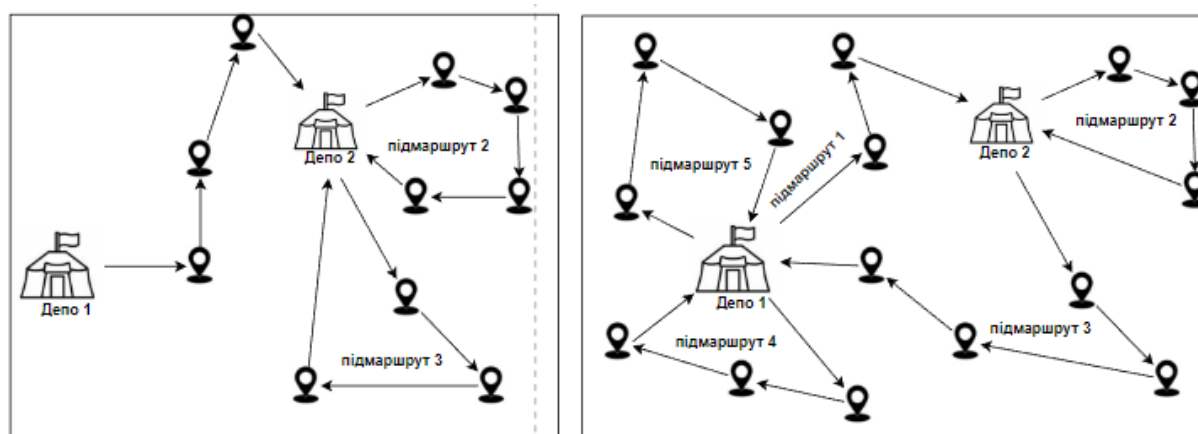


Рисунок 3.1 – Приклади розв'язків задачі маршрутизації БПЛА

Оскільки довжина кожного маршруту не повинна перевищувати певне значення, що залежить від характеристик БПЛА, для якого розв'язується

Зм.	Лист	№ докум.	Підпис	Дата

задача, маємо обмеження для кожного з маршрутів. Також, враховуємо, що між маршрутами повинна бути підзарядка, що займає час і відповідно погіршує розв'язок, а отже необхідно прагнути до мінімізації кількості маршрутів у загальному плані обльоту цілей.

3.2 Математична постановка задачі

Відштовхуючись від змістовної постановки задачі маршрутизації БПЛА, яка була сформульована у попередніх розділах, сформулюємо математичну постановку задачі. Вихідними параметрами задачі є:

- S – середня швидкість руху БПЛА;
- H – час, за який БПЛА перезаряджається у депо (однаковий для обох депо);
- F – час роботи БПЛА без перезарядки (максимальний час, який БПЛА може провести у польоті).

Дано множину пунктів, які БПЛА повинен відвідати, що складається з n цілей $W = \{1, 2, \dots, n\}$. Також дано множину пунктів, які є депо БПЛА, з $B = \{b_1, b_2\}$.

Також дано матрицю $\{d_{ab}\}$ відстаней перельотів між усіма пунктами, тобто і між депо, і між цілями:

$$d_{ab} = \{a, b \in \{1, 2, \dots, (n + 2)\}\},$$

де $n+1$ це перше депо b_1 , а $n+2$ це друге депо b_2 .

На основі значень у матриці відстаней перельотів $\{d_{ab}\}$ та вихідного параметру середньої швидкості польоту БПЛА S , розраховуються значення матриці часу перельотів:

$$t_{ab} = \{a, b \in \{1, 2, \dots, (n + 2)\}\}.$$

Потрібно скласти план відвідування пунктів. Для цього розбиваємо цілі W на M впорядкованих підмножин, кожна з яких має розмірність $n_i, i = \overline{1, \dots, M}$:

$$\{W_1 = \{w_1^1, w_2^1, \dots, w_{n_1}^1\}, \dots, W_M = \{w_1^M, w_2^M \dots w_{n_M}^M\}\},$$

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						22
Зм.	Лист	№ докум.	Підпис	Дата		

де $W_1 = \{w_1^1, w_2^1, \dots, w_{n_1}^1\}$ – це впорядковані цілі, які БПЛА відвідає у першому маршруті;

$W_M = \{w_1^M, w_2^M \dots w_{n_M}^M\}$ – це впорядковані цілі, які БПЛА відвідає у останньому маршруті M .

У структурі вихідного розв'язку впорядковані підмножини цілей $W_i, i = \overline{1, \dots, M}$ розмежовуються депо P , при цьому перед першою підмножиною стоїть лише перше депо b_1 між усіма множинами та після останньої стоїть будь-яке з депо $B = \{b_1, b_2\}$.

При цьому, наступні умови гарантують те, що кожна з цілей буде включена у один та лише один маршрут:

$$\bigcup_{m=1}^M W_m = W \quad \sum_{m=1}^M n_m = n$$

$$W_i \cap W_j = \emptyset, \text{ де } i \neq j.$$

Нехай $T_m, m = 1, \dots, M$ – сумарний час повного проходження (від депо до депо, включно з обльотом цілей, що належать підмаршруту) підмаршруту m :

$$T_1 = t_{b_1 w_1^1} + \sum_{i=1}^{n_1-1} t_{w_i^1 w_{i+1}^1} + t_{w_{n_1}^1 B}, \quad (3.1)$$

$$B \in \{b_1, b_2\}$$

$$T_m = t_{B w_1^m} + \sum_{i=1}^{n_m-1} t_{w_i^m w_{i+1}^m} + t_{w_{n_m}^m B}, m = 2, \dots, M, \quad (3.2)$$

$$B \in \{b_1, b_2\}$$

При цьому, з зі структури формування T_1 (формула 3.1) бачимо, що перший підмаршрут має початися у першому депо, а завершитися у першому чи у другому. Тоді як зі структури T_m , де $m = 2, \dots, M$ (формула 3.2) бачимо, що другий та усі подальші маршрути можуть як розпочатися, так і завершитися у першому або у другому депо.

Нехай $D_m, m = 1, \dots, M$ – сумарна відстань повного проходження (від

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

депо до депо, включно з обльотом цілей, що належать підмаршруту) підмаршруту m :

$$D_1 = d_{b_1 w_1^1} + \sum_{i=1}^{n_1-1} d_{w_i^1 w_{i+1}^1} + d_{w_{n_1}^1 B}, B \in \{b_1, b_2\} \quad (3.3)$$

$$D_m = d_{B w_1^m} + \sum_{i=1}^{n_m-1} d_{w_i^m w_{i+1}^m} + d_{w_{n_m}^m B}, m = 2, \dots, M, B \in \{b_1, b_2\} \quad (3.4)$$

При цьому, з зі структури формування D_1 (формула 3.3) бачимо, що перший підмаршрут має початися у першому депо, а завершитися у першій чи у другій. Тоді як зі структури D_m , де $m = 2, \dots, M$ (формула 3.4) бачимо, що другий та усі подальші маршрути можуть як розпочатися, так і завершитися у першому або у другому депо.

Тепер перейдемо до визначення цільової функції. Цільовою функцією є мінімізація часу (формула 3.5), який витрачено на проходження усього плану обльоту пунктів, тобто включає усі перельоти між депо та цілями, що входять до підмаршрутів, а також час перезарядки БПЛА між підмаршрутами:

$$z = (M - 1)H + \sum_{m=1}^M T_m \rightarrow \min \quad (3.5)$$

Ця цільова функція підлягатиме оптимізації у процесі розв'язання задачі.

Наступне обмеження (формула 3.6) гарантує те, що для кожного підмаршруту плану відвідування цілей сумарний час обльоту пунктів (включно з часом на переліт з одного з двох депо до першої цілі підмаршруту, та часом на переліт з останньої цілі підмаршруту до одного з двох депо) не

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

перевищуватиме час роботи БПЛА без перезарядки:

$$T_m \leq F, \text{ де } m = 1, \dots, M \quad (3.6)$$

Дане обмеження (формула 3.7) гарантує те, що план відвідування цілей складається мінімум з одного підмаршруту і не обмежує кількість підмаршрутів зверху:

$$M \geq 1. \quad (3.7)$$

Отже, відштовхуючись від змістовної та математичної постановок задач, маємо таку структуру розв'язку (3.8):

$$\mathbf{b}_1, w_1^1, w_2^1, \dots, w_{n_1}^1, \mathbf{B}, \dots, \mathbf{B}, w_1^M, w_2^M, \dots, w_{n-M}^M, \mathbf{B} \quad (3.8)$$

де $B = b_1$ або b_2 .

3.3 Обґрунтування методу розв'язання

3.3.1 Класифікація задач транспортної маршрутизації

На практиці, існує ряд обмежень, що впливає на підхід до розв'язання задач транспортної маршрутизації. Саме ці обмеження, що накладаються на постановку задачі та на розв'язок, спричиняють появу та розвиток нових класів задач маршрутизації.

Багатофакторність задач на практиці вимагає пошуку унікальних підходів до розв'язку, а отже, розробки складніших та більш узагальнених задач маршрутизації, що враховують ці особливості та обмеження реальних умов. Як будь-які алгоритми, підходи до розв'язання класичних задач маршрутизації обмежені, вимагають чітких початкових умов та ряду обмежень, а отже, для розв'язання реальних практичних задач їх недостатньо.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						25
Зм.	Лист	№ докум.	Підпис	Дата		

Розробляють модифікації алгоритмів розв'язання базових типів задач маршрутизації, залежно від потреб поставленої задачі, підходів до розв'язання задач, а також обмежень ресурсів, таких як доступність пам'яті та швидкості процесора, для досягнення найкращої продуктивності та ефективності програми [6].

Прикладами таких обмежень можуть бути:

- обмеження на кількість пунктів виробництва (наприклад, склад, депо);
- обмеження на кількість пунктів споживання (наприклад, точка, куди необхідно доставити товар);
- обмеження на довжину маршруту;
- обмеження на час, витрачений на маршрут;
- обмеження на відстань між пунктами виробництва та споживання (наприклад, служба доставки працює лише у чітко визначеному радіусі);
- обмеження на вагу або об'єм або кількість вантажу, що перевозиться;
- пріоритет відвідування пунктів (наприклад, лікарню необхідно відвідати перш ніж магазин);
- часові вікна, тобто пункт повинен бути відвіданим у певний проміжок часу, при цьому може бути визначено відхилення на затримку або на передування часу;
- типи та види транспорту, що впливає на вибір маршруту, швидкість перевезень, та допустимість доступних доріг;
- кількість транспортних засобів;
- порядок відвідування пунктів (наприклад, пункт обслуговування А потрібно відвідати перед пунктом Б, оскільки з пункту А до пункту Б потрібно перевезти певний вантаж);
- необхідність відвідування депо після певної кількості відвіданих точок споживання;
- динамічність задачі транспортної маршрутизації (чи змінюються

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

деякі параметри у ході розв'язання).

Зважаючи на ці обмеження було виокремлено набір різновидів задач маршрутизації транспорту. У назвах більшості з них можна відслідкувати обмеження, які вони покривають у першу чергу. На практиці, ці різновиди комбінують, модифікують та оптимізують в залежності від умов задачі, що стоїть перед дослідником.

Отже, розглянемо найрозповсюдженіші задачі маршрутизації транспортних засобів, що найчастіше використовуються у наукових дослідженнях:

Capacitated Vehicle Routing Problem, CVRP – маршрутизація з обмеженням по вантажопідйомності транспортних засобів. Характеризується тим, що в задачах цього типу вводиться додаткове обмеження: обсяг вантажів на кожному маршруті R_i не повинен перевищувати заданої величини Q , котра повинна бути однаковою для всіх транспортних засобів.

Потрібно враховувати потужність транспортного парку, він є фіксованим, усі транспортні засоби мають однакову місткість та дозволяють задовольнити для кожного клієнта попит на товар, при цьому мінімізувавши витрати ресурсів і залишившись у рамках власної вантажопідйомності [7].

Метою є мінімізація часу, витраченого на повний маршрут та кількості транспортних засобів, залучених у роботі.

VRP with Time Windows, VRPTW – задача маршрутизації з обмеженнями по часу відвідування пунктів споживання, або ж задача з часовими вікнами. Дана постановка задачі характеризується тим, що для кожного пункту споживання V_i існує заздалегідь визначений проміжок часу $[x_i, y_i]$, в який необхідно здійснити його обслуговування. В залежності від особливостей постановки задачі цей проміжок може бути строгим, і обслуговування пункту споживання поза ним недопустиме, або ж не строгий, коли допустимі відхилення від зазначеного проміжку часу. У деяких випадках, значення даного відхилення обмежене. Відповідно, визначено два варіанти

задачі часових вікон – строгі часові вікна (hard time window), та нестрогі часові вікна (soft time window).

Порушення визначеного проміжку часу може стосуватися як верхньої, так і нижньої його межі. При порушенні верхньої межі обраховується часова затримка, а при порушенні нижньої – очікування настання обслуговування пункту споживання. У випадку строгості часового проміжку обслуговування і недопустимості відхилень, якщо транспортний засіб прибуває раніше визначеного часу, відбувається простій – транспортний засіб очікує на настання визначеного часового проміжку. Якщо транспорт прибуває пізніше визначеного часу, в залежності від особливостей постановки задачі, пункт споживання або пропускається і вважається не обслугованим, або до цільової функції додається додаткове значення, як штраф за невиконання умови. Додаткові значення може бути сталим, або пропорційним часовій затримці [8].

Метою даної задачі є мінімізація кількості залучених транспортних засобів, а також загального часу роботи та часових затримок, що виникають при обслуговуванні пунктів споживання у визначені інтервали часу.

Multiple Depot VRP, MDVRP – задача маршрутизації з кількома пунктами виробництва (складів, депо). Особливістю даної постановки є те, що кожен транспортний засіб закріплений за одним з кількох існуючих пунктів виробництва. Відповідно, маршрут кожного транспортного засобу повинен розпочатися і завершитися в одній і тій же самій точці, яка може відрізнятися для кожного з транспортних засобів. Пункти споживання можуть бути закріплені за відповідними пунктами виробництва, або ж, пункт виробництва, транспортний засіб якого обслуговуватиме пункт споживання, визначається безпосередньо перед побудовою маршруту.

Метою даної задачі є мінімізація кількості залучених транспортних засобів, а також загального часу роботи [8].

Multi-Depot Vehicle Routing Problem with Inter-Depot Routes – розширена задача маршрутизації з кількома пунктами виробництва (складів, депо). Від

описаної вище задачі маршрутизації з кількома пунктами виробництва відрізняється тим, що може передбачати дозавантаження на маршруті транспортного засобу в одному з наявних депо. Також у розширеному варіанті даної задачі допускається, що маршрут може розпочинатися в одному з пунктів виробництва, а завершуватися в іншому [8].

При цьому, мета розширеної задачі не відрізняється від мети класичної задачі маршрутизації з кількома пунктами виробництва.

Dynamic VRP, DVRP – динамічна задача маршрутизації транспортних засобів. Особливістю даної задачі є те, що програмі наперед не відомі усі вихідні дані.

Наприклад, у процесі виконання задачі, після того як транспортний засіб покине пункт обслуговування може змінюватися кількість пунктів обслуговування, їх місцезнаходження, часове вікно, у яке потрібно обслужити пункт споживання на маршруті можуть виникати затори. Тобто, розв'язання задачі відбувається паралельно з плануванням та надходженням вхідних даних, а отже програма динамічно знаходить новий розв'язок, або ж покращує

поточний у процесі роботи. Проаналізувавши вихідні дані та результати роботи алгоритмів, для їх покращення, проводять імітаційне моделювання системи, що допомагає відслідковувати залежності між показниками системи, що динамічно змінюються у часі та адаптувати алгоритм під частоту виникнення змін та їх характер [9].

Asymmetric cost matrix VRP, AVRP – асиметрична задача маршрутизації транспортних засобів. Особливістю цієї задачі є те, що вартість переходу від i -го пункту споживання або обслуговування до j -го є нерівноцінною для одного й того ж транспортного засобу. Отже, моделювання даної задачі проводиться орієнтовним зваженим графом. Наприклад, час переміщення у різні сторони може відрізнятися через особливості інфраструктури доріг, швидкісних обмежень, заторів. Отже, при розв'язанні даної задачі варто враховувати асиметричність вартості переміщень між пунктами споживання та

обслуговування [8].

Метою даної задачі є мінімізація кількості залучених транспортних засобів, а також загального часу роботи.

Distance-Constrained VRP, DCVRP – задача маршрутизації транспортних засобів з обмеженням по відстані. Особливістю даної задачі є те, що при знаходженні оптимального маршруту враховується те, що існує деяке обмеження на максимальну відстань, що може пройти певний транспортний засіб.

Наприклад, при обмеженому запасі пального на транспортному засобі необхідно обмежити суму дуг між пунктами споживання та обслуговування на маршруті [7].

Метою даної задачі є мінімізація обсягів витрати ресурсів пального шляхом мінімізації довжини пройденого маршруту.

Heterogeneous fleet VRP, HVRP – задача маршрутизації транспортних засобів з неоднорідним парком транспорту. Особливістю цієї задачі є те, що при побудові маршрутів враховуються обмеження та особливості кожного з

транспортних засобів. Наприклад, може відрізнятися місткість, вантажопідйомність, швидкість, розхід палива транспортних засобів. Також, обмеження можуть накладатися на Можливість Обслуговування певним транспортним засобом певного пункту обслуговування. Наприклад, при доставці до пункту, що знаходиться на острові, не можна дістатися легковим автомобілем [6].

Метою даної задачі є мінімізація кількості залучених транспортних засобів, а також загального часу роботи, враховуючи особливості та можливості транспортних засобів.

VRP with Pick-Ups and Deliveries, VRPPD – задача маршрутизації з невідкладним поверненням товару. Задача відрізняється тим, що окрім доставки товарів може бути запит на повернення товару з пункту обслуговування на пункт виробництва. Тут необхідно уважно враховувати

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

вантажопідйомність машини, оскільки кількість товару, що перевозиться, в даній задачі може не лише зменшуватися, а і збільшуватися.

Крім того, вантаж може бути забраний або доставлений за певний час або у визначений термін. Таким чином, у VRPPD маємо не тільки визначити найкоротший шлях для доставки вантажу до кожного місця призначення, але й знайти оптимальну послідовність забору та доставки вантажів.

VRPPD є досить складною задачею, оскільки має декілька рівнів складності, такі як обмеження на вантажопідйомність, обмеження на час та відстань, інтеракція між вантажами та різні пріоритети доставки та забору товарів.

Для спрощення цієї модифікації задачі маршрутизації вводять додаткові умови, такі як скасування вимоги, що кожен пункт обслуговування має трапитися лише один раз на маршруті та відсутність можливості обміну товарів між пунктами споживання (повернений товар має бути транспортовано в пункт виробництва, а не передано в інший пункт обслуговування) [8].

Метою даної задачі є мінімізація транспорту та сумарного часу проходження маршрутів.

Split Delivery VRP, SDVRP – задача транспортної маршрутизації з розбиттям обслуговування пункту на кілька транспортних засобів. Особливістю задачі є те, що один вузол може бути обслугований декількома транспортними засобами.

Основна ідея SDVRP полягає в тому, що вантаж може бути розділений між декількома транспортними засобами. Наприклад, якщо вантаж потрібно доставити на 2 різні адреси, то цей вантаж може бути розділений між двома транспортними засобами.

Отже, як один вузол може бути обслугований декількома транспортними засобами, так і один вантаж може бути розділений між декількома транспортними засобами. Відповідно, Кількість товару, який може

					<i>IC92.050БАК.004 ПЗ</i>	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

бути доставлений одним транспортним засобом, може бути меншою, ніж сумарна кількість товару, який може бути доставлений на одному маршруті у VRP [8].

Метою даної задачі є мінімізація транспорту та сумарного часу об'їзду пунктів обслуговування.

Stochastic VRP, SVRP – стохастична задача маршрутизації транспорту. Це такий тип задачі транспортної маршрутизації, в якому певні фактори є випадковими та невідомими завчасно.

Наприклад, до таких факторів можуть відноситися наявність чи відсутність певного пункту обслуговування на маршруті, запит пункту обслуговування, час переміщення між пунктами обслуговування та виробництва. У задачі SVRP метою є розробка набору маршрутів, які будуть стійкими до цих невизначеностей і можуть адаптуватися до змін у реальному часі.

SVRP є складною задачею, оскільки потрібно вирішувати кілька оптимізаційних задач одночасно та приймати рішення в умовах невизначеності. Один з підходів до SVRP – використання ймовірнісних методів, таких як метод Монте-Карло, для моделювання невизначеності та генерування сценаріїв, які представляють різні можливі варіанти. Ці сценарії можуть бути використані для оцінки стійкості різних стратегій маршрутизації та вибору найкращої.

Інший підхід до SVRP – використання методів онлайн-оптимізації, коли рішення приймаються в реальному часі на основі доступної інформації. У цьому випадку розв'язок задачі маршрутизації транспорту оновлюється по мірі того, як з'являється нова інформація, що дозволяє системі адаптуватися до змінних умов.

Метою даної задачі є мінімізація транспорту та сумарного часу об'їзду пунктів обслуговування.

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

3.3.2 Комбінаторна складність задачі транспортної маршрутизації

Транспортна маршрутизація є складною комбінаторною задачею.

Комбінаторні задачі поділяються на два класи, а саме – P та NP. До класу P входять задачі, які можна вирішити за допомогою алгоритмів, що мають поліноміальну складність. Тобто, за допомогою таких, складність яких обмежується поліномом, у степені, що залежить від розмірності задачі. Ці алгоритми ефективні, а задачі класу простіші за задачі класу NP. До класу NP (nondeterministic polynomial) належать задачі, що можна вирішити недетермінованим поліноміальним алгоритмом. Тобто, таким алгоритмом, що розбивається на два етапи. Перший етап – це знаходження допустимого варіанту розв'язку, а другий етап – це перевірка знайденого на попередньому етапі розв'язку поліноміальним алгоритмом [10].

Своєю чергою, NP задачі поділяються на NP-повні (NP-complete) та NP-складні (NP-hard). NP-повна задача – це така задача комбінаторної оптимізації, що належить до класу NP, і при цьому будь-яка NP задача може бути зведена до неї, використовуючи поліноміальний алгоритм. У випадку виконання другої але невиконання першої умови, задача вважається NP-складною. NP-складна задача – це така задача комбінаторної оптимізації, до якої можуть бути зведені усі задачі NP класу [10]. Тобто, основна відмінність між NP-складними та NP-повними задачами полягає в тому, що NP-повні – це підмножина NP-складних, яка включає задачі, які належать до NP і є найскладнішими в класі NP.

Задача транспортної маршрутизації відноситься до класу NP-складних задач. Вона може бути вирішена за поліноміальний час з перевіркою правильності розв'язку, але її оптимальне розв'язку, як правило, вимагає часових та обчислювальних ресурсів, що зростають експоненційно зі збільшенням обсягу даних.

Задача транспортної маршрутизації не може бути віднесена до класу NP-

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

повних, оскільки для цього потрібно, щоб вона була і NP-складною, і NP-повною. NP-складність вже була обговорена – це вимога того, що задачу можна вирішити за поліноміальний час з перевіркою правильності розв'язку. Для того, щоб задача була NP-повною, потрібно ще довести, що будь-яка інша NP-складна задача може бути зведена до неї за поліноміальний час. Цього не вдалося довести для задачі транспортної маршрутизації. Хоча вона є складною та оптимізаційною, існує декілька алгоритмів, які можуть розв'язати її для невеликих обсягів даних за прийнятний час.

3.3.3 Класифікація методів розв'язання задачі транспортної маршрутизації

Як і для будь-якої задачі комбінаторної оптимізації, для розв'язання задачі транспортної маршрутизації можна застосувати точні або наближені методи розв'язання.

За допомогою точних методів можна знайти глобальний розв'язок. Хоч точні методи і гарантують знайти оптимальний розв'язок, але вони можуть бути дуже часозатратними та ресурсозатратними. Ці методи можуть бути використані для розв'язання невеликих задач, коли час розв'язання не є критичним фактором.

Приклади точних методів, що можуть бути застосовані для розв'язання задачі транспортної маршрутизації:

- метод гілок та меж (англ. Branch and Bound);
- лінійне програмування (Linear Programming);
- динамічне програмування (Dynamic programming).

Наближені методи, своєю чергою, поділяють на евристичні та метаевристичні.

Евристичні методи базуються на досвіді та експертному знанні про розв'язання подібних задач та мають тенденцію шукати розв'язок шляхом знаходження більш або менш оптимального розв'язку за допомогою евристик

або емпіричного пошуку. Такі методи можуть бути корисними для задач, в яких точний або повний перебір може бути неможливим або занадто складним з точки зору обчислювальної складності. Евристичні методи можуть допомогти знайти прийнятний розв'язок в таких ситуаціях, надаючи швидкі та ефективні алгоритми для розв'язання задач.

Евристичні методи, зазвичай працюють швидше за точні, але не гарантують знайти оптимальний розв'язок. Вони можуть бути використані для великих задач, де важливішим є швидкість розв'язання, ніж точність.

Евристичні методи можна класифікувати на конструктивні, двофазні (кластерні) та покращуючі. Конструктивні методи ґрунтуються на побудові розв'язку з нуля шляхом поступового додавання клієнтів та транспортних засобів до маршруту, не мають подальшого поліпшення. Двофазні методи складаються з двох етапів: кластеризація пунктів споживачів та розв'язання задачі транспортної маршрутизації в межах кожного з кластерів. Покращуючі методи ґрунтуються на поступовому покращенні вже існуючого розв'язку, шляхом модифікації маршрутів всередині кожного, або між ними [10].

Приклади евристичних методів, що застосовують для розв'язання задачі маршрутизації транспортних засобів:

а) конструктивні методи;

- 1) жадібний алгоритм (Greedy);
- 2) метод найближчого сусіда (Nearest Neighbour);

б) двофазні (кластерні) методи;

- 1) алгоритм пелюсток (Petal Algorithm);
- 2) алгоритм Ліна-Кернігана (Lin-Kernighan);
- 3) алгоритм замітання (Sweep Algorithm);

в) покращуючі методи;

- 1) внутрішньомаршрутні алгоритми (Inner route, Intra route);
- 2) алгоритми k-оптимізації, оптимізації першого порядку, оптимізації другого порядку (k-opt, 2-opt, 3-opt).

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						35
Зм.	Лист	№ докум.	Підпис	Дата		

Метаевристичні методи поєднують в собі переваги обох попередніх методів, це такі методи, що знаходять розв'язок використовуючи комбінації інших евристик чи алгоритмів, при цьому один є провідним, а інший (або ж інші) є підлеглими. Однією з ключових особливостей метаевристичних методів є їх здатність обходити локальні максимуми та мінімуми, що дозволяє їм знаходити оптимальні розв'язки, які можуть бути недосяжні для інших методів. Вони швидші за точні методи та більш точні за евристичні методи. Однак, ці методи можуть бути складні для реалізації та потребувати більш високої кваліфікації виконавців [10].

Наведемо такі приклади метаевристичних методів, що можуть бути застосовані для розв'язання задачі маршрутизації транспортних засобів:

- метод імітаційного відпалу (Simulated Annealing);
- метод оптимізації мурашиними колоніями (Ant Colony Optimization);
- бджолині алгоритми й методи штучних бджолиних колоній (Bees Algorithms, Artificial Bee Colony);
- генетичний алгоритм (Genetic Algorithm);
- табу пошук (Tabu Search);

Отже, використання різних методів дозволяє знайти розв'язок задачі транспортної маршрутизації, який відповідає вимогам та обмеженням, залежно від умов та потреб.

3.3.4 Вибір методу розв'язання для задачі маршрутизації БПЛА

У попередньому підрозділі була розглянута класифікація алгоритмів комбінаторної оптимізації, визначено недоліки та переваги кожного з класів, а також окремо розглянуто деякі приклади розповсюджених алгоритмів кожного з класів у контексті задачі транспортної маршрутизації. Визначено переваги та недоліки як класів, так і окремих методів, тож підсумуємо виокремивши головне для поставленої задачі з проаналізованого матеріалу.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

Відштовхуючись від постановки задачі дослідження, що була також детально розібрана в одному з попередніх розділів, можемо виокремити те, що на першому місці стоїть задача підвищення швидкості роботи обраного алгоритму, оскільки розраховується, що систему використовуватимуть у польових умовах, або ж для обслуговування критичної інфраструктури. Звичайно, точність є також дуже важливим критерієм при виборі методу розв'язання. Отже, звузимо коло вибору до метаевристичних алгоритмів, оскільки, як було визначено у попередніх розділах, саме вони поєднують у собі переваги швидкості евристичних та точності точних методів. А складність розробки та реалізації цих алгоритмів буде простором для покращення обраного методу.

Отже, тепер оберемо метод серед метаевристичних алгоритмів. Для розробки та реалізації я обрала біологічно навіяний бджолиний алгоритм комбінаторної оптимізації, який також відноситься до локальних. Основна перевага бджолиного алгоритму полягає в його здатності досліджувати глобальний простір рішень та знаходити оптимальні розв'язки без застрягання в локальних оптимумах. Крім того, бджолиний алгоритм може працювати зі змінними обмеженнями, що дозволяє враховувати різноманітні фактори, такі як вартість, час та обмеження щодо кількості ресурсів. Зокрема, для задачі транспортної маршрутизації бджолиний алгоритм може ефективно генерувати та оновлювати маршрути, враховувати обмеження при визначенні оптимальних шляхів між пунктами.

Отже, тепер оберемо метод серед бджолиних алгоритмів. У літературі в основному зустрічається чотири різні бджолині алгоритми, які є основою для всіх підходів до розв'язання проблем маршрутизації транспортних засобів. Ці алгоритми можна розділити на два підкласи на основі поведінки бджіл у природньому середовищі, яка використовується як натхнення для цих алгоритмів:

а) заготовка корму (Foraging);

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						37
Зм.	Лист	№ докум.	Підпис	Дата		

- 1) алгоритм штучної бджолоїної колонії (Artificial Bee Colony, ABC);
 - 2) алгоритм оптимізації бджолиними колоніями, (Bee Colony Optimization, BCO);
 - 3) алгоритм бджолоїної системи (Bee System, BS);
- б) одруження (Marriage);

1) алгоритм шлюбу медоносних бджіл (Marriage in Honeybees Optimization, МНВО) [24].

Розглянемо основні відомості про кожен з цих алгоритмів.

Натхненням для алгоритму штучної бджолоїної колонії, Artificial Bee Colony (ABC) слугує харчова поведінка медоносних бджіл. Першу модифікацію алгоритму для розв'язання задачі маршрутизації транспортних засобів була запропонована для задачі комівояжера. Алгоритм ABC працює з трьома типами агентів (бджіл):

- бджоли-працівники – бджоли, які вже пов'язані з джерелом їжі, що є розв'язком;
- бджоли-спостерігачі – бджоли, які спостерігатимуть за танцем зайнятих бджіл і слідуватимуть за ними до джерела їжі;
- бджоли-розвідники – бджоли, які блукають у пошуковому просторі, щоб знайти нові джерела їжі, це може бути випадковий або керований процес [25].

Алгоритм починається з ініціалізації, і доки не досягнуто критерій завершення, послідовно відбуваються фази роботи кожного з типу бджіл.

Під час ініціалізації бджоли розвідники стартові джерела їжі, кількість яких дорівнює кількості бджіл-працівників та кожне джерело одразу асоціюється з робочою бджолою.

Під час фази бджіл-працівників, кожна з розподілених на джерело їжі бджіл шукає краще джерело їжі поміж сусідніх застосовуючи жадібний алгоритм. Якщо в околі бджоли-працівниці знайдено краще за поточне

джерело їжі, то воно заміняє поточне і асоціюється з даною бджолою.

На фазі бджіл-спостерігачів бджоли-працівники передають свої поточні джерела їжі бджолам-спостерігачам за допомогою танцю виляння. Бджоли-спостерігачі спостерігають за цим танцем і ймовірно обирають одну з бджіл, щоб прослідувати до її джерела їжі. Іноді використовують колесо рулетки як спосіб обрати за якою бджолою слідувати, використовуючи якість джерела їжі як критерій придатності для цього процесу. Прослідувавши за обраною бджолою-працівницею, бджола-спостерігач шукатиме кращі джерела їжі по сусідству з джерелом їжі робочої бджоли. Використовується операція, аналогічна до операції на фазі робочих бджіл [25].

На фазі бджіл-розвідників, бджоли-працівники перетворюються на бджіл-розвідників, у випадку якщо джерело їх їжі не змінювалося протягом кількох попередніх ітерацій, тобто не покращувалося. Джерело запам'ятовується, а бджола-працівник відлітає від попереднього джерела їжі та починає пошук заново.

Алгоритм оптимізації бджолиних колоній, Bee Colony Optimization (BCO) – це також модель, натхненна харчовою поведінкою медоносних бджіл. Він базується на роботі, запропонованій для вирішення проблеми розподілу серверів у центрах обробки даних. Був запропонований алгоритм розв'язання TSP, названий алгоритмом збору бджолиного пилку. Далі на цій базі з'явився алгоритм оптимізації бджолиних колоній.

Робота алгоритму починається з ініціалізації, і далі, доки не досягнуто критерій зупинки, повторюють такі фази, як спостереження танцю виляння, виконання пошуку корму та виконання танцю виляння.

На ініціалізації кожна бджола популяції представляє кандидата на розв'язок, що побудовано за допомогою метода найближчого сусіда або випадкового підходу.

На фазі спостереження за танцем виляння перед тим, як вилетіти з вулика на пошук їжі, бджола вирішує, чи буде вона спостерігати за танцем

виляння інших бджіл. Бджола з більшою ймовірністю буде спостерігати за танцем інших бджіл, якщо її коефіцієнт її пилку не такий хороший, як середній коефіцієнт у вулику. Спостерігаючи за танцем виляння, бджола вибере танець іншої бджоли для наслідування. Коефіцієнт цієї обраної бджоли стане її улюбленим пилком, який вона використовуватиме під час пошуку корму. Під час першої ітерації, оскільки бджоли не мають танцю для наслідування, дозволяється досліджувати простір пошуку без обраного розв'язку.

На фазі виконання пошуку корму бджола ітеративно будує повний розв'язок, переміщуючись від одного міста до іншого. Бджола вибирає наступне місто для відвідування на основі ймовірнісного правила переходу, яке враховує відстань між двома містами і чи присутній цей переїзд в поточному кращому розв'язку.

На фазі виконання танцю вихляння бджола, яка знайшла розв'язок краще, ніж попереднє, показує іншим новий маршрут за допомогою танцю вихляння. Тривалість танцю визначається якістю знайденого бджолою розв'язку та середньою якістю вулика [25].

Алгоритм бджолої системи, Bee System передбачає після зміни позицій вуликів проходження для кожного вулика фаз пошуку джерел їжі, відсікання джерел їжі та танець вихляння, після цього застосовується локальний пошук, і ця процедура повторюється до тих пір, доки не досягнуто критерію завершення.

На початку кожної ітерації місце розташування вулика змінюється. Вулик є початковим містом для кожної бджоли. Основна основною ідеєю є зміна на кожній ітерації початкового міста маршруту, який будується [25].

Етап вважається дискретною одиницею часу в середовищі бджоли. Кожна ітерація має заздалегідь визначену кількість етапів, на яких бджоли будуть виконувати певні завдання.

На кожному етапі бджолі дозволяється відвідати заздалегідь визначену кількість міст під час процесу збору корму. На кожному етапі бджола

ітеративно будує підпункти, поки не досягне повного розв'язку задачі. Не всі бджоли починають шукати корм на одному і тому ж етапі. Ті, що не починають на початковому етапі, підуть за іншими бджолами, коли вирішать вилетіти з вулика. Бджоли будують підмаршрути, відвідуючи одне місто за іншим. Чим більша відстань між містами, тим менша ймовірність що бджола полетить у це місто наступним; чим більша кількість ітерацій вже проведена, тим більший коефіцієнт впливу відстані; чим більша кількість бджіл, які виконували той самий переліт в минулому, тим більша ймовірність [25].

На етапі відмови від джерела їжі, бджола вирішує, чи продовжувати збирати корм з того ж місця на наступному етапі, чи покинути свій підмаршрут. Чим від довший підмаршрут порівняно з розв'язками, знайденими іншими бджолами, тим вища ймовірність відмови від поточного розв'язку.

Бджоли, які зберігають свої джерела їжі, вирішують, чи будуть вони залучати до цього джерела інших бджіл за допомогою танцю виляння. Рішення танцювати для просування свого розв'язку є імовірнісною функцією з високою ймовірністю того, що бджола буде танцювати. Якщо бджола вирішила відмовитись від свого розв'язку, вона почне слідувати за іншою бджолою, перед цим спостерігаючи за танцем виляння інших бджіл, щоб вирішити, якій з них слідувати. Чим нижча вартість підмаршруту що просувається, порівняно з іншими розв'язками, тим вища ймовірність що бджола його обере; чим більша кількість бджіл рекламує той самий підмаршрут, тим вища ймовірність.

В кінці кожної ітерації зміни вулика проходить локальний пошук. Використовується 2орт або 3орт, залежно від розмірності задачі транспортної маршрутизації [25].

Алгоритм оптимізації шлюбів у медоносних бджіл, Marriage in Honeybees Optimization, був представлений у 2001 році. Алгоритм має чотири типи агентів (бджіл), кожен з яких виконує свої завдання в процесі оптимізації:

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						41
Зм.	Лист	№ докум.	Підпис	Дата		

- матки – представляють найкращі розв’язки, знайдені на поточний момент, і використовуються для операції кросинговеру;
- трутні – представляють розв’язки по всьому простору пошуку, які використовуються для кросинговеру;
- виводки – розв’язки, отримані в результаті операцій кросинговеру та мутації між матками та трутнями;
- робочі бджоли – набір евристик, що використовуються для покращення виводків.

Алгоритм починається з ініціалізації, і далі, доки не досягнуто критерій завершення, ітеративно повторюється послідовність таких фаз як шлюбний політ, формування розплоду, покращення виводків, та формування нового набору маток [25].

При ініціалізації випадковим чином генерується набір маток і до кожної з них застосовується випадкова робоча бджола.

На початку фази шлюбного польоту матці присвоюється певна енергія та швидкість. На кожній точці польоту матка зустрічається з трутнем, який представляє випадково згенерований розв'язок. Матка вирішує, чи обере вона трутня, при цьому чим вища придатність трутня, тим вища ймовірність, що матка його обере. Якщо трутень обраний для спарювання, його речовина додається до банку речовини матки. Речовина трутня являє собою лише частину її розчину. Матка витрачає на політ енергію, з часом енергія та відповідно швидкість матки зменшуються, і шлюбний політ закінчується, коли енергія матки опускається нижче встановленого порогу. Шлюбний політ проходить для кожної матки.

На фазі формування виводку генерується виводок шляхом застосування операторів кросинговеру та мутації між матками та речовинами в їхніх банках речовини.

На фазі покращення виводків для покращення кожного розплоду використовується робоча бджола. Кожній бджолі присвоюється коефіцієнт

фітнес, який керує вибором робочої бджоли. Працівник, який покращує виводок, має вищу ймовірність на те, що його оберуть. Коефіцієнт перераховується протягом усього процесу пошуку на основі результатів поліпшення, виконаного для кожного виводку.

На фазі визначення нового набору маток для найкращого розплоду перевіряється, чи є він кращим за найгіршу матку. Якщо так, то цей розплід замінює відповідну матку. Це повторюється, доки найкращий розплід не стане не гіршим за найгіршу матку [25].

З вищеперелічених бджолиних алгоритмів я обрала для аналізу та реалізації алгоритм штучної бджолиної колонії. До його переваг відноситься:

- швидкість збіжності – алгоритм має швидку збіжність, що дозволяє знайти розв'язок відносно швидко; наприклад, алгоритм оптимізації бджолиними колоніями має повільнішу збіжність, оскільки використовується обмежена кількість бджіл;

- можливість врахування обмежень – алгоритм може легко врахувати обмеження, такі як обмеження на максимальну кількість підмаршрутів, час проходження маршрутів, що є важливим для задачі транспортної маршрутизації; наприклад, алгоритм оптимізації бджолиними колоніями не завжди може легко врахувати обмеження;

- інтуїтивна зрозумілість – алгоритм штучної бджолиної колонії простіший у реалізації та розумінні порівняно з іншими бджолиними алгоритмами, які вимагають вищої математичної складності;

- стійкість до локальних мінімумів – алгоритм не зациклюється на локальних мінімумах, як це може статися з іншими еволюційними алгоритмами, що забезпечує знаходження оптимального розв'язку; інші бджолині алгоритми менш стійкі до локальних мінімумів.

До недоліків обраного алгоритму штучної бджолиної колонії можна віднести:

- можливі проблеми зі збіжністю до глобального оптимуму в деяких

складних задачах; це пов'язано з тим, що алгоритм не завжди може досить ефективно досліджувати всі області простору параметрів;

— залежно від області застосування, може потребувати деякого налаштування параметрів для досягнення кращих результатів; наприклад, кількість штучних бджіл, розмір популяції і швидкість збіжності можуть бути важливими параметрами, які потребують налаштування для кожної конкретної задачі.

— може бути вразливим до шуму в даних або вхідних параметрів, що може вплинути на якість його розв'язку.

3.4 Опис методу розв'язання

Для розв'язання задачі було обрано алгоритм штучної бджолиної колонії. Загалом, алгоритм складається з таких основних етапів як ініціалізація параметрів, та локальна оптимізація і схрещування впродовж певної кількості ітерацій.

Є різні підходи до вибору кількості ітерацій, які має виконати алгоритм.

До популярних відносяться:

- фіксована кількість ітерацій;
- достатньо малий приріст функції метрики, наприклад менший за заданий поріг;
- досягнення заданої точності розв'язку;
- максимальний час виконання алгоритму;
- пропорція розміру вхідних даних до кількості ітерацій.

Для досягнення найкращого результату, я обираю критерій, що розв'язок не змінюється впродовж заданої кількості ітерацій. Цей критерій на мою думку найкраще підходить під постановку задачі, оскільки він, порівняно з іншими, дозволяє досягти найвищої точності, при цьому обмеження на ресурси (час або пам'ять) у нас відсутні, оскільки на практиці задача розв'язуватиметься для даних відносно невеликої розмірності, необхідності відвідувати БПЛА понад

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

200 пунктів за один план обльоту не буде.

Для генерації випадкових розв'язків на нульовому кроці алгоритму застосуємо простий стохастичний алгоритмом, що базується на випадковому виборі досяжної цілі з зваженим ймовірнісним розподілом. Впродовж усієї генерації маршруту, кожній з досяжних з поточного пункту цілей присвоюється коефіцієнт, чим ближча вершина до поточної, тим він вищий. Коефіцієнт розподіляється між досяжними вершинами як певна ймовірність, тобто сумарно коефіцієнти досяжних вершин дають одиницю. Далі, на основі цих розподілених коефіцієнтів ймовірно обирається ціль. Важливо зазначити, що досяжні цілі обираються на основі обмеження про максимальний час польоту БПЛА, але не лише так, щоб БПЛА мав заряд на те щоб дістатися певної цілі, а й на те щоб БПЛА вистачило заряду дістатися до одного з депо після відвідування цієї цілі. Коли на певному кроці досяжних цілей немає, БПЛА прямує до депо.

Формальний опис стохастичного алгоритму для задачі маршрутизації БПЛА:

Крок 1. Ініціалізуємо перше депо як початкову точку плану обльоту цілей і як поточний пункт.

Крок 2. Доки усі цілі не включено у план відвідування:

Крок 2.1. Визначаємо усі досяжні з поточного пункту цілі.

Крок 2.2. На основі відстані від поточного пункту до цілей, призначаємо кожній цілі коефіцієнт вибору так, щоб сума усіх коефіцієнтів досяжних з поточного пункту цілей була одиницею.

Крок 2.3. ЯКЩО досяжних цілей немає, призначаємо поточним пунктом найближче депо.

ІНАКШЕ (якщо досяжні цілі є) ймовірно обираємо одну з досяжних цілей, на основі коефіцієнту вибору.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

Крок 2.4. Оновлюємо поточний пункт.

Крок 3. ЯКЩО досягне лише одне депо, **ТО** включаємо його в план об'їзду цілей.

ІНАКШЕ (якщо досяжні обидва депо), ймовірно включаємо одне з депо в план об'їзду цілей.

Формування множини початкових розв'язків керується певними правилами, а не повністю випадково, дозволить пришвидшити роботу алгоритму, оскільки потрібно буде менше ітерацій на стабілізацію розв'язків (розв'язки швидше припинять значно покращуватися, оскільки вже на етапі генерації початкових, до оптимізації, вони будуть прийнятні), а також підвищити точність роботи алгоритму (оскільки алгоритм з самого початку проводитиме оптимізацію прийнятних розв'язків).

Впродовж роботи алгоритму, після таких етапів як генерація розв'язків на початку, локальна оптимізація, та схрещування, за потребою оновлюємо рекордний розв'язок, якщо на відповідному етапі було знайдено розв'язок, цільова функція якого більша за поточний рекорд (цільову функцію рекордного розв'язку).

Також, після етапів генерації нових розв'язків локальною оптимізацією та схрещуванням, підтримуємо кількість розв'язків рівною початковій популяції, залишаючи для подальшої роботи лише найкращі (з найкращою цільовою функцією).

На етапі локальної оптимізації з поточних розв'язків обираємо певний відсоток (відсоток задається як параметр алгоритму) кращих розв'язків, і працюємо саме з ними. Для кожного з пунктів кожного з обраних маршрутів проводиться локальна оптимізація шляхом заміни пункту з сусідніми, при цьому розглядається певна кількість сусідніх пунктів, кількість яких дорівнює околу локальної оптимізації, тобто певному відсотку (відсоток задаємо як параметр алгоритму) від загальної кількості пунктів. Дані правила актуальні як для цілей, так і для депо. При цьому, для депо розглядається додатковий

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

варіант оновлення розв'язку – включення певних відвідувань депо з плану відвідування цілей, якщо це не призведе до порушення обмеження на максимальний допустимий час БПЛА у повітрі без перезарядки. Усі нові розв'язки запам'ятовуємо, додавши до поточної популяції. В кінці етапу локальної оптимізації розмір популяції зменшуємо до вхідного розміру, відхиливши гірші розв'язки (оцінюємо за значенням цільової функції). Це робиться для того, щоб на наступному етапі ми знову працювали лише з найкращими поточними розв'язками (після локальної оптимізації вже не беремо додатково відсоток найкращих розв'язків, лише слідкуємо, щоб розмір популяції не перевищував вхідний).

Приклад одного з варіантів локальної оптимізації для одної з цілей плану відвідування цілей, шляхом зміни порядку її відвідування з одною з точок, що належить околу локальної оптимізації, наведено на рисунку 3.2. Як видно з рисунку, перестановку не сплинула на склад цілей у підмаршрутах, лише на впорядкування цілей у першому підмаршруті. І з візуального представлення чітко видно, наскільки одна вдала перестановка може скоротити довжину плану відвідування цілей, а отже і час, що буде витрачено БПЛА на його проходження, що покращить значення цільової функції для нового знайдено розв'язку, порівняно з тим, який оптимізується.

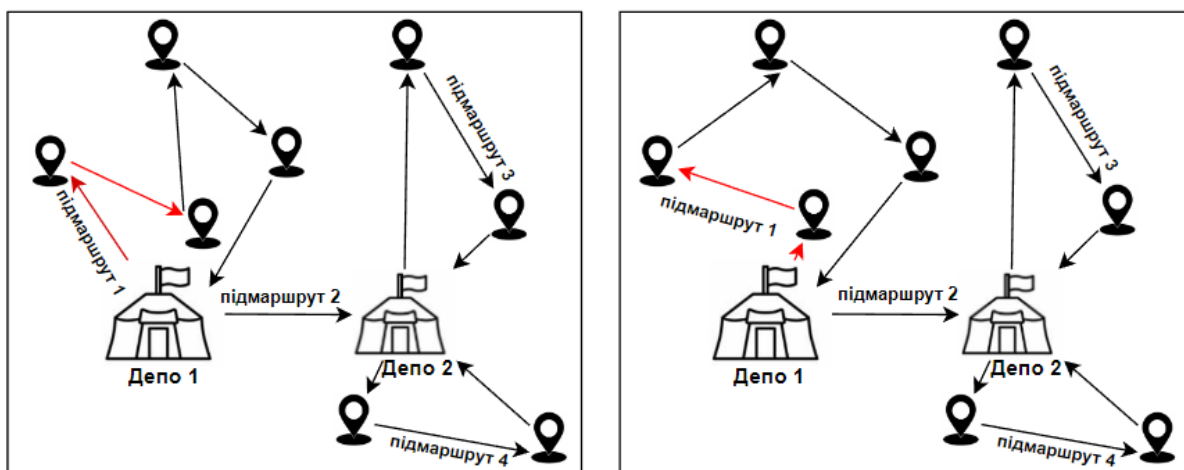


Рисунок 3.2 – Приклад одного з варіантів локальної оптимізації

На етапі схрещування популяція, оновлена на попередньому етапі локальної оптимізації, випадковим чином розбивається на пари розв'язків та проводиться схрещування. Для кожної з пар проводимо до околу для локальної оптимізації помноженого на 10 схрещувань, шляхом перестановки порядку відвідування обраних випадковим чином цілей одного маршруту таким чином, як вони розташовані у другому маршруті, і навпаки, для другого маршруту обрані випадковим чином цілі переставляємо на ті місця, на яких вони стоять у першому маршруті.

На рисунку 3.5 наведено приклад результату одного з можливих схрещувань першого маршруту, який проілюстровано на рисунку 3.3 з другим маршрутом, який проілюстровано на рисунку 3.4. Аналогічні схрещування проводяться і для другого маршруту з першим. На рисунку чітко видно, що порядок обльоту однієї з цілей першого плану відвідування цілей було замінено на такий, при якому вона облітається у другому маршруті, БПЛА після відвідування цієї цілі летить одразу в друге депо. Завдяки цьому вдалося не лише обрати коротший переліт з цілі до наступного пункту, а й уникнути зайвого перельоту з першого до другого депо, відповідно, уникнути зайвої підзарядки, на яку БПЛА витрачає час, що погіршує цільову функцію.

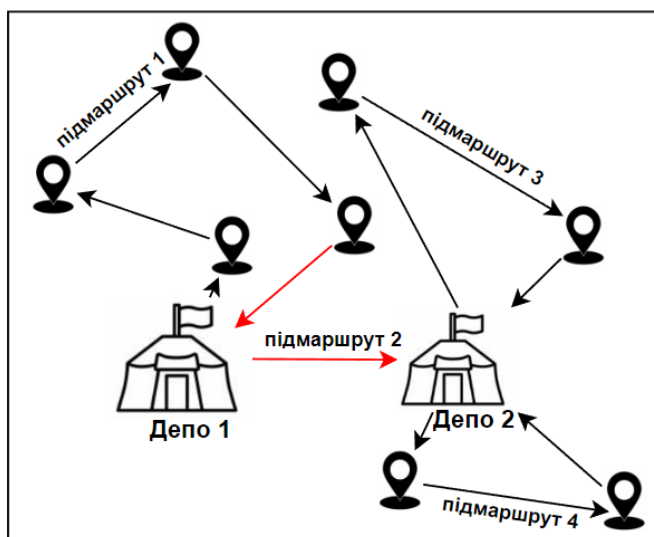


Рисунок 3.3 – План відвідування цілей, що схрещується

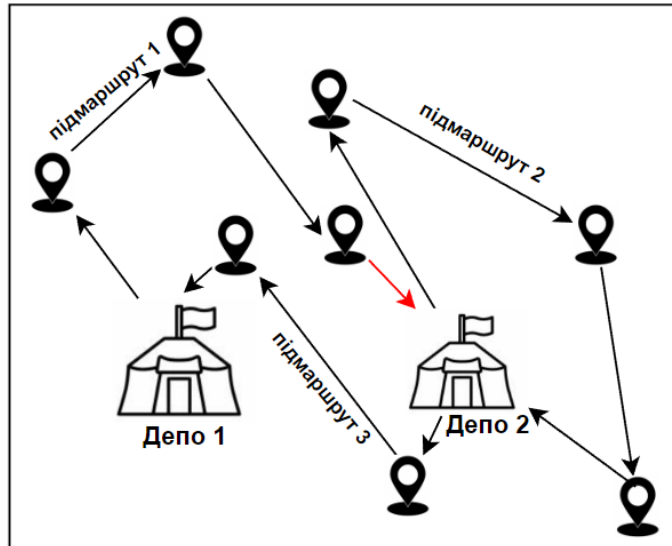


Рисунок 3.4 – План відвідування цілей, з яким схрещують

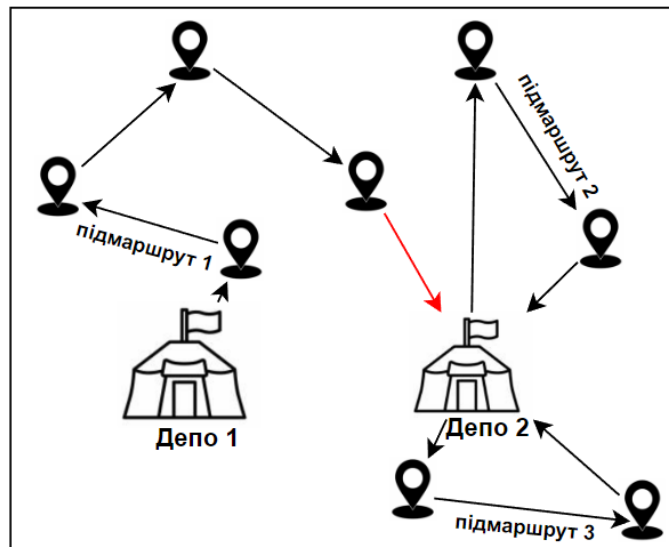


Рисунок 3.5 – План відвідування цілей, отриманий в результаті одного з варіантів схрещування

Перед переходом до наступної ітерації, необхідно залишити у поточній популяції не більше ніж відсоток (відсоток задається як параметр алгоритму) від початкової популяції кращих розв'язків.

Отже, формальний покроковий опис алгоритму штучної бджолоїної колонії для задачі маршрутизації БПЛА:

Крок 0. Ініціалізація параметрів алгоритму:

— c – потужність множини допустимих розв’язків, початкова популяція розв’язків, яку генеруємо на початку алгоритму;

— g – кількість поточних кращих розв’язків (для локальної оптимізації, відсоток від c);

— g – кількість цілей, які необхідно відвідати;

— o – окіл в якому проводиться локальна оптимізація, тобто кількість сусідніх точок з якими проводитимемо локальну оптимізацію, відсоток від g ;

— $p = c/2$ кількість пар розв’язків (для оптимізації шляхом схрещування);

— I – кількість ітерацій, впродовж яких рекордний розв’язок не змінювався.

Крок 1. Генерація c випадкових розв’язків (стохастичним алгоритмом).

Крок 2. Визначення рекордного розв’язку.

Крок 3. ПОКИ не виконані умови завершення (впродовж I ітерацій не оновлюється рекордний розв’язок):

Крок 3.1. Робота бджіл-працівників. Локальна оптимізація для g від c найкращих розв’язків, пошук кращого розв’язку в околі, окіл дорівнює o від g сусідніх точок.

Крок 3.2. Оновлення рекордного розв’язку (за необхідності).

Крок 3.3. Робота бджіл-дослідників. Розбиття випадковим чином розв’язків на p пар, покращення шляхом схрещування.

Крок 3.4. Оновлення рекордного розв’язку (за необхідності).

Також невід’ємною частиною програмної реалізації алгоритму є впровадження усіх обмежень, що були описані в математичній постановці задачі. Як цільову функцію для визначення рекорду використовую основну цільову функцію з математичної постановки задачі, на мінімізацію часу, витраченого на проходження плану відвідування цілей (включно з часом, витраченим на обліт цілей кожного з підмаршрутів від депо до депо, та часом, сумарно витраченим на усі перезарядки БПЛА між підмаршрутами, при цьому

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

припускається, що на початку проходження плану відвідування цілей БПЛА заряджено, та в кінці БПЛА не потребує зарядки).

Блок-схему алгоритму представлена у графічних матеріалах, кресленик ІС92.050БАК.004 Д4.

3.4.1 Підбір параметрів методу розв'язання задачі

Для того, щоб розроблений алгоритм штучної бджолоїної колонії працював ефективно, необхідно правильно підібрати його параметри. Правильно підібрані значення параметрів можуть допомогти досягти швидшої збіжності, зменшити час виконання або покращити точність результатів алгоритму. Також, зважаючи на те, що кожна задача має свої особливості, підбір параметрів дозволяє адаптувати алгоритм до конкретних потреб задачі, що дозволяє досягти кращих результатів.

Загалом, існує багато різних підходів до підбору параметрів алгоритмів. Найпопулярнішими серед них є наступні:

- ручний підбір (manual search);
- сітка параметрів (grid search);
- випадковий підбір (ranom search);
- байєсівська оптимізація (bayesian optimization).

Ручний підбір дозволяє експериментувати з різними значеннями параметрів та спостерігати за їх впливом на результати. Однією з переваг ручного пошуку є можливість використання експертного знання, що дозволяє швидше знаходити оптимальні параметри. Експерт може враховувати особливості задачі, що важливі для досягнення кращих результатів. Для оптимізації, ручний пошук можна систематизувати, наприклад, спочатку робити великі стрибки між значеннями параметра, що перевіряється, а потім зменшувати у проміжках, значень які показали кращі результати [36].

Ідея сітки параметрів полягає в переборі всіх можливих комбінацій параметрів заздалегідь заданого діапазону, щоб знайти найкращі значення для

					<i>ІС92.050БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		51

досягнення оптимальної продуктивності алгоритму. Переваги сітки параметрів включають простоту реалізації, ясність та об'єктивність результатів. Вона дозволяє об'єктивно оцінювати вплив кожного параметра на результати алгоритму та знаходити комбінації параметрів, що працюють найкраще. Однак, цей метод може бути витратним з точки зору обчислювальних ресурсів, особливо при великому просторі параметрів, через велику кількість комбінацій. Також цей метод не є адаптивним до результатів попередніх експериментів [37].

Ідея випадкового перебору полягає в тому, щоб випадковим чином генерувати значення параметрів алгоритму з заданого діапазону і оцінювати продуктивність алгоритму з використанням цих випадкових значень. Цей процес повторюється декілька разів, і результати оцінок використовуються для знаходження найкращих значень параметрів [37].

Байєсівська оптимізація використовує попередні знання про функцію, яку необхідно оптимізувати, та поступово покращує свої прогнози, проводячи нові експерименти з точками у просторі параметрів. Цей метод базується на використанні апіорних ймовірностей та поновленні прогнозів з урахуванням отриманих результатів [36].

Параметри алгоритму штучної бджолоїної колонії є статичними, для їх підбору було обрано комбінацію двох методів підбору параметрів, а саме випадковий перебір та ручний підбір. Ручний підбір застосовано для параметра потужності популяції, а випадковий перебір застосовано для параметра кількості найкращих представників популяції та параметру кількості ітерацій, яка є критерієм завершення алгоритму, якщо впродовж неї алгоритм не покращується. Такий вибір зроблено відштовхуючись від ресурсів системи, особливостей параметрів т постановки задачі.

В результаті ручного підбору параметра потужності популяції було визначено, що алгоритм ефективно працює при значенні параметра 100. При цьому ручний пошук було систематизовано шляхом поступового зменшення

стрибків значень параметра для яких перевіряється ефективність алгоритму, наближуючи підбір до значень потужності популяції, з якими отримано кращі розв'язки.

Для проведення випадкового перебору необхідно спочатку для параметрів кількості найкращих представників популяції та параметру кількості ітерацій, яка є критерієм завершення алгоритму, якщо впродовж неї алгоритм не покращується, задати діапазон, у якому вони генеруватимуться, тобто простір підбору параметрів. Обираємо початковий простір підбору параметрів спираючись на логічні обмеження та припущення (наприклад, параметри невід'ємні, розв'язання задачі з такими параметрами не перевантажує ресурси системи, але водночас параметри підтримують точність алгоритмів), та потім ітеративно покращуємо простір параметрів, зміщаючи його в ту сторону, до якої ближчі значення параметрів, що забезпечують кращі результати.

У результаті проведення процесу визначення простору підбору параметрів отримано такі значення:

- кількість найкращих представників популяції – ціле число від 1 до розміру усієї популяції (проводимо підбір на популяції розміру 100);
- кількість ітерацій без покращення розв'язку, критерій завершення роботи алгоритму – ціле число від 1 до 50.

Отже, простори підбору параметрів вже покращені, а випадковий підбір параметрів цих просторів дає можливість визначити залежності між параметрами.

Випадковий підбір проведено на 15 задачах розмірності 20, загалом проведено 100 ітерацій. Для кожної з ітерацій обчислюється середнє значення цільової функції по усім розв'язаним задачам. На кожній з ітерацій система випадковим чином генерує значення параметрів, що підбираються. Серед середніх значень цільової функції усіх ітерацій обирається найкраще значення і саме комбінація згенерованих параметрів, що відповідають цій ітерації

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

вважається ефективною.

На рисунку 3.6 наведено результат проведеного експерименту. При цьому, значення параметру потужності популяції, для виначення ефективного значення якого було затосовано мануальний підбір, залишається фіксованим для усіх прогонів і дорівнює сто. З результатів бачимо, що на прогоні з індексом 30 значення цільової функції найкраще.

(index)	mean	maxOfIterWithoutImpr	numberOfBestSolutions
0	80.22	39	49
1	80.15	37	54
2	80.89	12	50
3	80.75	14	80
4	82.07	8	22
5	82.18	48	46
6	80.6	50	53
7	83.2	45	10
8	82.57	24	20
9	79.45	39	61
10	81.63	24	75
11	83.3	44	39
12	82.35	34	55
13	84.12	27	1
14	85.61	5	7
15	80.98	39	59
16	79.3	47	25
17	80.12	13	83
18	82.02	40	9
19	81.08	39	89
20	81.8	7	73
21	80.36	15	89
22	82.66	29	32
23	81.49	41	15
24	80.94	35	90
25	81.97	5	82
26	81.37	34	9
27	80.49	36	35
28	80.52	38	67
29	83.57	3	15
30	78.74	15	55
31	79.99	23	65
32	82.84	20	50
33	81.25	22	23
34	81.25	28	19
35	83.27	7	6
36	83.23	17	22
37	81.62	42	9
38	82.09	6	36
39	79.85	40	86
40	80.15	44	64
41	81.61	49	94
42	81	20	98
43	81.44	3	32
44	81.02	47	51
45	84.87	39	13
46	80.01	37	69
47	83.17	17	17
48	80.7	42	53
49	84.99	7	8

(index)	mean	maxOfIterWithoutImpr	numberOfBestSolutions
50	79.48	38	58
51	82.85	9	13
52	80	19	92
53	86.59	35	1
54	82.49	2	78
55	83.23	10	10
56	82.07	48	23
57	80.75	50	78
58	83.28	36	23
59	81.64	27	19
60	85.32	39	2
61	79.36	29	90
62	81.69	29	52
63	81.41	20	54
64	83.04	19	22
65	80.76	20	45
66	82.36	4	28
67	81.54	30	48
68	80.22	9	80
69	82.16	36	47
70	81.29	5	31
71	82.63	21	12
72	80	38	88
73	80.69	41	33
74	83.26	24	2
75	85.21	24	8
76	82.4	44	93
77	80.45	35	43
78	82.8	44	19
79	80.79	6	38
80	81.67	37	53
81	81.19	43	79
82	82.85	49	9
83	81.68	24	91
84	81.38	28	87
85	83.28	5	23
86	84.38	11	5
87	81.56	14	62
88	84.14	34	3
89	80.93	14	68
90	83.38	31	20
91	80.16	12	89
92	80.57	13	65
93	80.9	19	93
94	82.51	2	33
95	83.48	23	3
96	79.64	21	91
97	80.5	13	65
98	79.55	33	68
99	82.02	38	30

Рисунок 3.6 – Результати проведення експерименту по підбору параметрів алгоритму штучної бджолоїної колонії

Результатами випадкового перебору параметрів є наступні ефективні значення параметрів:

- кількість найкращих представників популяції $\in [54; 56]$;

– кількість ітерацій без покращення розв’язку, критерій завершення роботи алгоритму $\in [14; 16]$.

Даний експеримент по підбору параметрів було автоматизовано.

3.4.1.1 Аналіз результатів роботи розробленого методу розв’язання

Для аналізу того, наскільки ефективно працює реалізований алгоритм штучної бджолиної колонії, а також для того щоб переконатися, що у результатах роботи алгоритму відсутні аномалії, проведено експеримент.

Для цього через інтерфейс користувача, а саме через модуль проведення експериментів, проведено експерименти. При цьому параметри алгоритму незмінні, встановлено середні значення інтервалів ефективних параметрів, які було підібрано у експерименті по підборі:

- потужність популяції – 100;
- кількість найкращих представників популяції – 55;
- кількість ітерацій без покращення розв’язку, критерій завершення роботи алгоритму – 15.

Обрано кілька значень кількості прогонів, спираючись на потужність системи, а саме 10, 25 та 50 прогонів. Також обрано кілька розмірностей задач, для яких проводиться експеримент, спираючись на аналіз змістовної постановки задачі, на те, для розв’язання задач яких розмірностей використовуватиметься система, а також на потужність системи, а саме 10, 15, 20, 25, та 30 цілей.

Для кожної з комбінацій цих значень проведено експеримент, та результати занесено до таблиці 3.1.

Таблиця 3.1 – Результати роботи алгоритму штучної бджолиної колонії

Кількість прогонів	Розмірність задачі	Краще значення ЦФ	Середнє значення ЦФ	Середньо-квадратичне відхилення ЦФ
10	10	44,33	49,26	4,9

	15	52,24	63,85	7,25
	20	67,54	77,94	5,55
	25	81,46	92,38	6,33
	30	91,87	105,34	12,8
25	10	42,56	49,59	6,56
	15	50,27	62,67	8,88
	20	65,65	77,17	8,45
	25	79,46	93,1	7,3
	30	84,03	108,98	11,49
50	10	40,87	50,57	4,88
	15	49,88	63,5	6,72
	20	53,27	78,38	10,64
	25	68,91	94	8,43
	30	80,74	107,93	11,7

Отже, з результатів можна зробити наступні висновки:

- середнє значення цільової функції не залежить від кількості прогонів; це очікувано, оскільки, середнє значення варіюється попри кількість прогонів;
- краще значення цільової функції покращується зі збільшенням кількості прогонів, оскільки шанс того, що задача буде розв'язана ефективно, зростає з кількістю прогонів;
- у результатах роботи алгоритму немає аномалій, вихідні дані розподілені рівномірно;
- середньоквадратичне відхилення не залежить від кількості прогонів, але у більшості експериментів можна спостерігати закономірність

того, що чим більша розмірність задачі, тим більше середньоквадратичне відхилення.

Отже, можна вважати, що алгоритм працює коректно та ефективно для різних розмірностей задач. Візуалізація результатів дослідження та залежність значення цільової функції від розмірності задач для кожної з обраних кількостей прогонів проілюстрована діаграмою на рисунку 3.7.

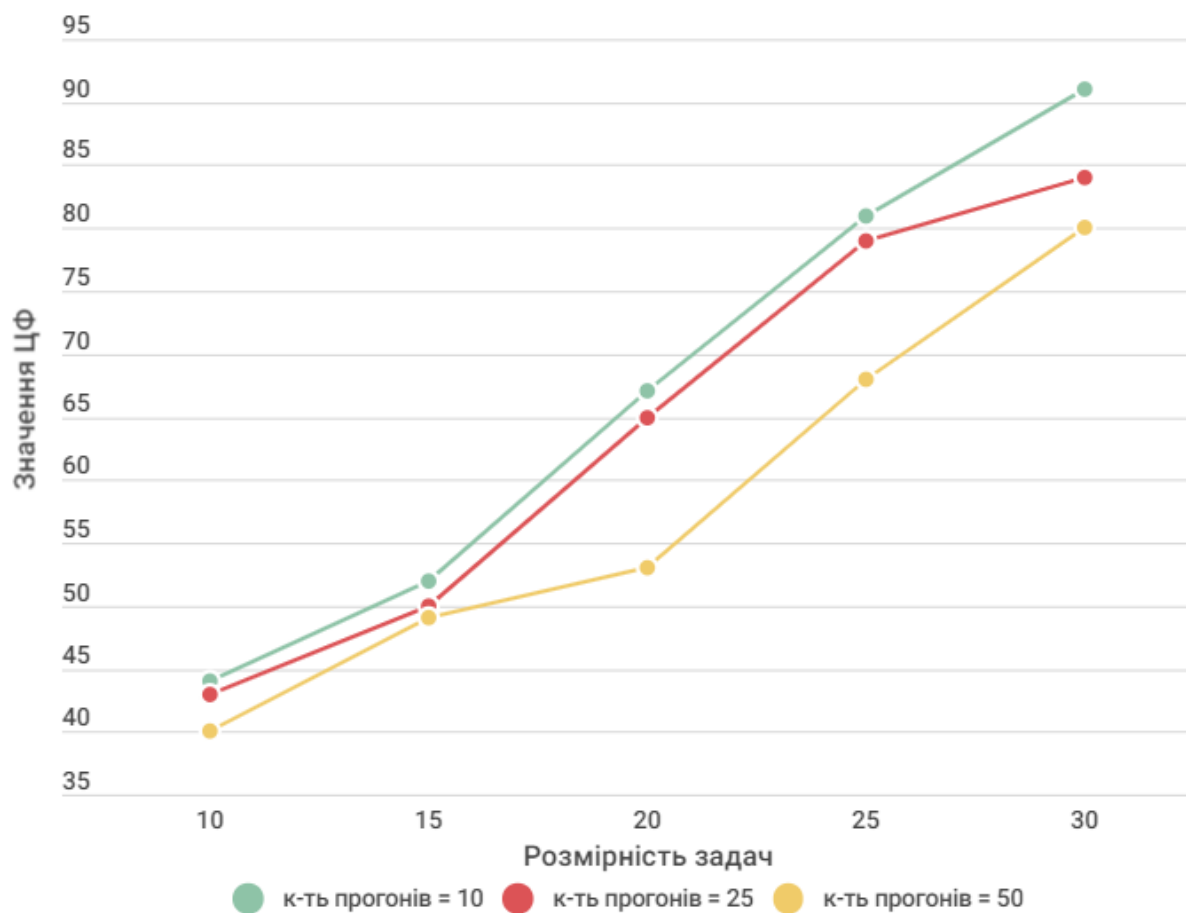


Рисунок 3.7 – Вплив кількості прогонів алгоритму штучної бджолиної колонії на результат розв’язання

Висновок до розділу

У розділі сформульовано змістовну та математичні постановки задачі маршрутизації БПЛА. Для вибору методу розв’язання проаналізовано класифікацію задач транспортної маршрутизації, розглянуто комбінаторну

складність задач транспортної маршрутизації, а також проаналізовано класифікацію методів розв'язання задачі транспортної маршрутизації.

Після вибору конкретного методу розв'язання, бджолиного алгоритму, проаналізовано його класифікацію та на основі аналізу обрано той, що найбільше підходить під постановку задачі.

Спроектовно, та детально описано модифікацію обраного методу розв'язання штучної бджолиної колонії. Алгоритм застосування методу для задачі маршрутизації БПЛА проілюстровано блок-схемою.

Адаптація класичного методу штучної бджолиної колонії під сформульовану задачу дозволить покращити результати роботи алгоритму, а отже отримувати кращі результати з використанням менших об'ємів ресурсів (пам'яті та часу). Проведено аналіз якості імплементованого алгоритму, експериментальним шляхом.

За допомогою експерименту підібрано такі параметри алгоритму, з якими алгоритм дає кращі результати.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						58
Зм.	Лист	№ докум.	Підпис	Дата		

4 ПРОГРАМНЕ І ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

4.1.1 Мова розробки програмного забезпечення

Для реалізації бджолиного алгоритму для системи маршрутизації БПЛА було обрано мову програмування TypeScript – мову програмування, розроблена Microsoft як розширення JavaScript. Вона має на меті надати розробникам сильніші можливості явної статичної типізації та додаткові функції, такі як інтерфейси та класи. Typescript також компілюється в чистий JavaScript, що робить її сумісною з будь-яким браузером або середовищем виконання, що підтримує JavaScript [26].

Мова програмування TypeScript була створена командою Microsoft у 2012 році і вперше була представлена у версії Visual Studio 2012. Одним з ключових розробників TypeScript є Андерс Гейлсберг, який раніше працював над розробкою мов C# та .NET Framework в Microsoft. Він був головним архітектором TypeScript та допоміг створити мову разом з іншими членами команди Microsoft [27].

Отже, це відносно нова мова програмування і потрібно проаналізувати динаміку її популярності, для розуміння як вона себе зарекомендувала та чи є перспективною.

Для аналізу перспективності обраної мови, посилатимемося на дані результатів дослідження JetBrains. JetBrains – це компанія-розробник програмного забезпечення зі штаб-квартирою в Празі, Чехія. Компанія заснована у 2000 році і спеціалізується на створенні інтегрованих середовищ розробки (IDE) для програмування на різних мовах програмування, таких як Java, Kotlin, Python, PHP, Ruby та інших. Найбільш відомими продуктами JetBrains є IntelliJ IDEA, PyCharm, WebStorm, PhpStorm, RubyMine, AppCode, і CLion. Ці IDE мають високу популярність серед програмістів зі всього світу завдяки своїм зручним інтерфейсам, широкому спектру функцій,

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

продуктивності та підтримки різних мов програмування. Крім розробки IDE, JetBrains також розробляє інші програмні продукти, такі як системи управління версіями (наприклад, GitLab) та інструменти для аналізу коду. Компанія також пропонує безкоштовний вебсервіс для контролю версій коду – GitHub [29].

На рисунку 4.1 наведено графік динаміки популярності мов, за останні 6 років. У опитуванні брали участь користувачі продуктами компанії JetBrains та запитання для збору даних було сформульоване як обрати мову програмування, яку користувач найчастіше використовував впродовж останнього року. У дослідженні взяли участь 29 269 розробників з усього світу, дослідження проходило у травні – липні 2022 року [28]. У результатах опитування набір мов більший, але для винесення на графік було обрано мови, які вивчалися впродовж здобуття ступеня бакалавра.

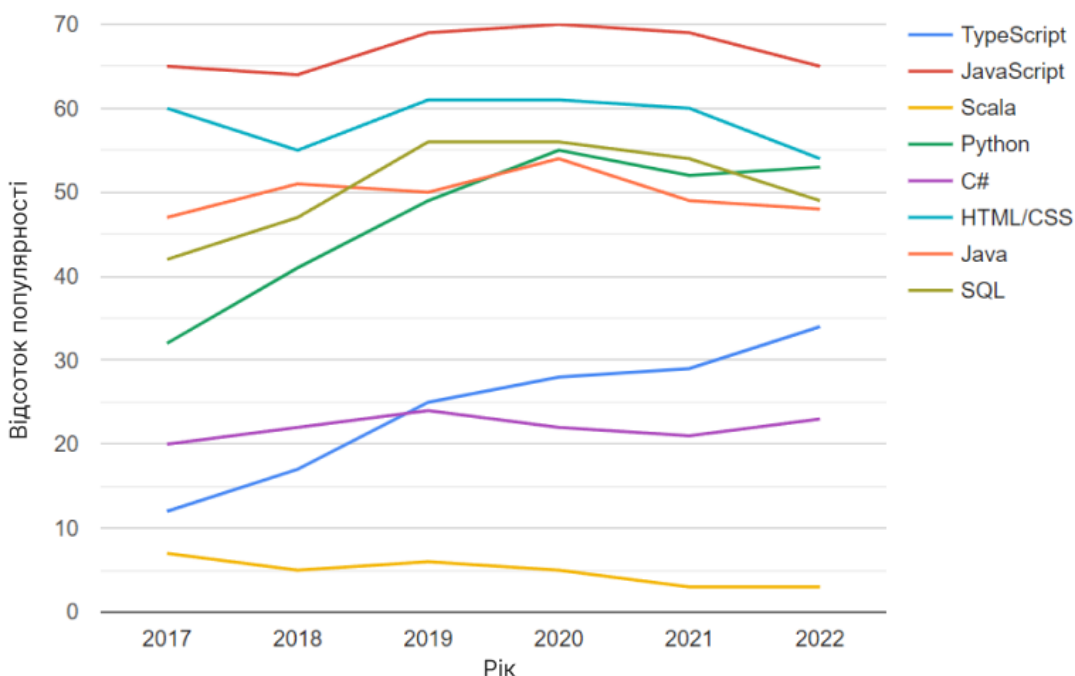


Рисунок 4.1 – Динаміка популярності використання мов за останні шість років за дослідженням JetBrains

З результатів дослідження чітко видно, що TypeScript є хоч і не

найпопулярнішою мовою, але саме тою мовою, популярність якої зростає найшвидше. JavaScript, мова від якої походить TypeScript, за результатами дослідження, є найпопулярнішою мовою програмування серед опитаних користувачів, і JavaScript все ще була потиснена TypeScript, проте використання TypeScript майже зросло майже втричі за останні 6 років.

Серед основних переваг TypeScript, окрім швидкого розвитку, можна виокремити наступні:

- розширюваність – TypeScript дозволяє легко розширювати код, оскільки він дозволяє визначати інтерфейси та класи зі строгими типами;
- спільнота – TypeScript має велику та активну спільноту розробників, яка надає різноманітні бібліотеки та інструменти для розробки, що значно полегшує процес розробки програмного забезпечення;
- підтримка останніх стандартів JavaScript – TypeScript підтримує останні стандарти JavaScript, що дозволяє використовувати нові функції та можливості мови без втрати підтримки старіших браузерів;
- інтеграція з іншими бібліотеками – TypeScript інтегрується з іншими бібліотеками та фреймворками JavaScript, що дозволяє використовувати їх в TypeScript проєктах;
- підтримка популярними IDE – TypeScript має підтримку у редакторах коду та інтегрованих середовищах розробки, оскільки мова типізована, таких як Visual Studio Code, WebStorm та інших; це полегшує написання коду, рефакторинг та налагодження програмного забезпечення;
- підтримка інтеграції з іншими бібліотеками та фреймворками – TypeScript підтримує інтеграцію з багатьма іншими бібліотеками та фреймворками, такими як React, Angular, Node.js і Express;
- зручна документація – TypeScript має документацію, що допомагає розробникам зрозуміти, як працює TypeScript та як використовувати його в своїх проєктах [27, 30].

Як і у будь-якої мови програмування, у TypeScript є свої недоліки. Ось

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						61
Зм.	Лист	№ докум.	Підпис	Дата		

деякі з них, що є значущими:

— додаткові файли – у процесі розробки у проєкт включаються додаткові файли з розширеннями .ts, .d.ts, .map, .js, що може здатися незручним і заплутувати користувача, особливо на маленьких проєктах;

— велика кількість нетривіальних класів – необхідно пам'ятати де знаходяться властивості, щоб писати код, спираючись на класи; наприклад, замість одного класу Event, як це представлено у JavaScript, у TypeScript є класи KeyboardEvent, MouseEvent, TouchEvent, та інші;

— необхідність додаткового часу для вивчення – TypeScript є складнішими і займає більше часу для вивчення, ніж, наприклад звичайний JavaScript;

— додатковий час для розробки – незважаючи на те, що TypeScript допомагає запобігти помилкам в коді на ранніх етапах розробки, він може займати більше часу для написання, через необхідність описувати типи в кожному файлі;

— великий розмір файлів – TypeScript може збільшувати розмір файлів, що може знижувати продуктивність програми, особливо на мобільних пристроях [27, 30].

Отже, на основі зібраної у розглянутому дослідженні статистики та порівняння вищезазначених переваг та недоліків TypeScript, цю мову можна вважати актуальною та перспективною.

4.1.2 Середовище розробки програмного забезпечення

Як середовище розробки було обрано Visual Studio Code (VS Code), що є одним з найпопулярніших та найбільш універсальних інтегрованих середовищ розробки (IDE) на сьогоднішній день. VS Code став популярним редактором коду завдяки своїй широкій функціональності, продуктивності та налаштуванням [31].

Деякі з найпопулярніших функцій VS Code включають наступне:

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		62

— підтримка багатьох мов програмування – VS Code підтримує багато мов програмування, включаючи JavaScript, Python, Java, C ++, PHP та інші;

— автодоповнення коду – VS Code може автоматично підставляти код, що зменшує кількість ручного введення коду;

— налагодження коду – VS Code дозволяє розробникам налагоджувати свій код за допомогою вбудованого налагоджувач;

— розширюваність – VS Code має широкий вибір розширень, що дозволяють розширити його функціональність та пристосувати його до потреб розробника;

— керування версіями – VS Code підтримує керування версіями, що дозволяє розробникам працювати з системами контролю версій, такими як Git;

— робота зі змінами – VS Code має вбудовану систему для знаходження та відстеження змін у коді, що забезпечує більш ефективну роботу з проєкт;

— робота з хмарними сервісами – VS Code може працювати з різними хмарними сервісами, такими як Microsoft Azure, Amazon Web Services (AWS) та Google Cloud Platform (GCP);

— конфігурація та налаштування – VS Code дозволяє налаштувати його з використанням конфігураційних файлів та параметрів командного рядка;

— безкоштовність – VS Code є безкоштовним та відкритим програмним забезпеченням;

— кросплатформність – VS Code підтримує операційні системи Windows, macOS та Linux, що робить його доступним для розробників з різних платформ [31].

Крім того, VS Code підтримує віддалену розробку, що дозволяє розробникам працювати з системою навіть з віддаленої локації. Також VS Code має багато інструментів для автоматизації тестування та розгортання, що

може спростити процес розробки та підвищити якість системи [31].

Отже, VS Code є дуже функціональним та налаштовуваним редактором коду, який можна адаптувати до потреб кожного розробника.

4.1.3 Технології розробки програмного забезпечення

Бібліотеки в програмуванні є наборами попередньо написаних кодів, які використовуються для вирішення певних завдань або реалізації певних функцій. Вони містять готові модулі, класи, функції та інші ресурси, які можна використовувати у своєму проєкті.

Основна мета використання бібліотек в програмуванні полягає в ефективному використанні часу та зусиль розробника. Вміст бібліотеки вже написаний, перевірений та оптимізований, що дозволяє розробнику уникнути необхідності писати всю функціональність з нуля.

Оскільки у задачею даного дипломного проєкту є реалізація саме клієнтської частини системи, опишемо бібліотеки, які використані у реалізації клієнтської частини. Основними є MobX та React.js.

MobX – це становий менеджер, який дозволяє легко та ефективно керувати станом системи. Забезпечує простоту використання та керування даними та автоматичного оновлення компонентів при зміні стану.

У MobX можна оголосити об'єкт як "стовбуровий" (observable), що дозволяє автоматично відстежувати зміни у ньому. Коли значення властивостей станового об'єкта змінюється, MobX автоматично оновлює всі компоненти, які залежать від цього значення. Також MobX дозволяє вам визначати реактивні обчислювані властивості, які автоматично оновлюються, коли змінюються їх залежності. Є можливість використання декораторів для зручного оголошення стану та обчислюваних властивостей [34].

MobX добре інтегрується з бібліотекою React і дозволяє зручно керувати станом React-компонентів. Також, переваги MobX включають:

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		64

— простота використання – MobX має простий та зрозумілий API, що дозволяє швидко розпочати роботу з ним, дозволяє декларувати станові об'єкти та обчислювані властивості всього кількома рядками коду;

— ефективність – має потужну систему відстеження залежностей, яка дозволяє автоматично оновлювати тільки ті компоненти, які дійсно залежать від змін у стані, що сприяє високій продуктивності та швидкій реакції на оновлення даних;

— гнучкість – MobX не накладає обмежень на структуру коду, дозволяє використання з React, Angular або Vue.js, як для малих проєктів, так і для великих застосунків;

— чистота коду – MobX дозволяє писати простий і зрозумілий код, без складних шаблонів керування станом та розрахунків;

— розширюваність – MobX має широкий набір доповнень та інструментів розробки, які дозволяють розширити його функціональність, наприклад для зручного налаштування логування, серіалізації, персистентності та інших функцій [34].

У системі MobX використано для синхронізації даних між клієнтом та сервером, тобто для отримання та оновлення даних з сервера, а також для відстеження змін стану на клієнті та їх відправки на сервер. Також, MobX використано у системі для управління даними на стороні клієнта, включаючи валідацію, та сортування даних.

React є бібліотекою JavaScript, підтримувана TypeScript, використовується для створення користувацьких інтерфейсів. Вона дозволяє розробникам писати компоненти, які представляють частини UI, і забезпечує ефективне управління станом та оновленням компонентів. "react" та "react-dom" – це два різні пакети, пов'язані з бібліотекою React [35].

"react" – це основний пакет React, який містить функціонал для створення та рендерингу компонентів React. Він включає реалізацію основних концепцій, таких як віртуальний DOM (Document Object Model, це програмний

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		65

інтерфейс, що надає структуроване представлення веб-сторінки або XML-документа), роботу з компонентами, станом, життєвим циклом і подіями [35].

"react-dom" – це пакет, який забезпечує інтеграцію React з DOM (Document Object Model) веб-сторінки. Він надає методи для маніпулювання DOM-елементами та рендерингу React-компонентів у реальному DOM. Використовується для монтування React-дерева компонентів у певному елементі DOM, оновлення компонентів при зміні стану та подіях, і взаємодії з реальним DOM-деревом [35].

Отже, "react" відповідає за функціональність React, а "react-dom" забезпечує інтеграцію React з DOM для відображення компонентів у браузері. Обидва пакети є необхідними для розробки React-додатків, проте "react-dom" використовується специфічно для веб-платформи.

Також, для реалізації клієнтської частини системи використано наступні бібліотеки:

— "js-cookie" – ця бібліотека дозволяє легко створювати, читати та видаляти куки (cookies) в браузері з використанням JavaScript; вона забезпечує зручний інтерфейс для роботи з куками і дозволяє зберігати дані між різними сеансами користувача; зокрема, дана бібліотека використана для імплементації процесу авторизації на стороні клієнта;

— "mobx-react-lite" – пакет реактивного прив'язування (bindings) між бібліотекою MobX та React; прив'язування – це механізм, який забезпечує зв'язок між даними та їх відображенням або між різними елементами програми; прив'язки дозволяють автоматично синхронізувати значення одного об'єкта або змінної з іншими, забезпечуючи їх оновлення при зміні; бібліотека дозволяє використовувати можливості MobX для управління станом у React-компонентах, при цьому забезпечуючи динамічне оновлення компонентів при зміні даних;

— "ramda" – функціональна бібліотекою, яка надає набір утиліт і функцій для роботи з колекціями даних; надає безліч функцій для роботи з

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		66

масивами, об'єктами, рядками і іншими типами даних; дозволяє виконувати операції, такі як фільтрація, мапування, згортання, злиття та багато інших, в функціональному стилі; дає можливість створювати композицію функцій, компоновання в логічні послідовності, що дозволяє створювати складніші операції шляхом комбінування простих функцій; полегшує створення чіткого та зрозумілого коду;

— "react-router-dom" – надає навігаційні компоненти для розробки односторінкових застосунків з використанням React; дозволяє визначати маршрути та управляти переходами між сторінками, динамічно змінювати URL і додатково забезпечує можливість створювати вкладені маршрути;

— "react-scripts" – бібліотека, що надає набір скриптів і конфігурацій, для спрощеної розробки і збірки React-застосунків; надає можливість запуску локального серверу з можливістю автоматичного перезавантаження, включає конфігурацію Webpack, яка дозволяє збирати проєкт в готову для випуску версію, автоматично оптимізує код, збирає ресурси, такі як зображення, і створює один або декілька вихідних файлів, готових для розгортання на сервері, надає стандартну конфігурацію для розробки React-додатків, що дозволяє уникнути необхідності вручну налаштовувати Webpack (модульний збирач проєкту), Babel (інструмент транспіляції коду з сучасних версій до старіших, підтримуваних більшістю браузерів) та інші інструменти.

4.2 Вимоги до технічного забезпечення

Система гарантовано працює без перебоїв та перезавантаження технічного забезпечення для пристроїв з такими характеристиками:

- місткість оперативної пам'яті (RAM) – 2 гігабайти та більше;
- місткість жорсткого дискового накопичувача або твердотілого накопичувача (HDD/SSD) – 128 гігабайтів та більше;

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
						67
Зм.	Лист	№ докум.	Підпис	Дата		

- версія центрального процесора Core i3-5xx, AMD A4/A6 або новіша;
- встановлений браузер на пристрої (Opera/ Mozilla Firefox/ Google Chrome/ Internet Explorer/ Safari) останньої версії.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма компонентів

У імplementованій інформаційній системі маршрутизації БПЛА є два рівні компонент – компоненти клієнтської та компоненти серверної частини. Взаємодія між ними відбувається за допомогою запитів API (Application Programming Interface), написаних на рівні API запитів.

У дипломному проєкті реалізована частина клієнтської сторони системи, а саме компонента клієнтської частини проведення експериментів, компонента клієнтської частини авторизації, навігаційна панель з можливістю ініціювання користувацького сценарію авторизації. Реалізована компонента розв'язувача, що відповідає за розв'язання задачі маршрутизації БПЛА алгоритмом штучної бджолоїної колонії. Також реалізовані запити на сервер, необхідні для роботи відповідних компонентів клієнтської частини системи.

На структурній схемі компонентів системи, представленої у графічних матеріалах, кресленик ІС92.050БАК.004 Д2, фіолетовим кольором виділені елементи системи, реалізовані в даному дипломному проєкті.

Отже, на діаграмі компонентів системи маршрутизації БПЛА графічно проілюстровано усі компоненти системи та взаємодію між ними. Можна виділити такі основні частини функціоналу системи:

- проведення експериментів;
- навігація з можливістю авторизації;
- авторизація;
- розв'язувач.

Залежно від складності частини функціоналу, для його реалізації використано певний набір компонент, сервіси, стори, компоненти бізнес-логіки, компоненти користувацького інтерфейсу на клієнтській частині та контролери і сервіси серверної частини.

Наприклад, для функціоналу експериментів система використовує користувацький інтерфейс, який своєю чергою використовує компоненту бізнес-логіки проведення експериментів. Компонента бізнес-логіки проведення експериментів має спільний з розв'язувачем стор розв'язувача, а також спільно з розв'язувачем використовує клієнтську частину сервісу розв'язувача. Клієнтська частина сервісу розв'язувача використовує API для виконання запитів на серверну частину.

Функціонал авторизації та навігації також тісно пов'язаний, компоненти користувацького інтерфейсу як функціоналу авторизації, так і функціоналу навігації, використовують спільний стор стану системи та компоненту клієнтської частини сервісу авторизації у системі. Своєю чергою, клієнтська частина сервісу авторизації у системі використовує API для виконання запитів на серверну частину.

Додатково, для функціоналу розв'язувача, а саме для його користувацького інтерфейсу, використовується компонента візуалізації побудованих шляхів. А для функціоналу експерименту, а саме для сервісу серверної частини використовується компонента генерації задач.

Для реалізації комплексного дипломного проєкту, спроектовано та імплементовано ще два алгоритми, окрім алгоритму штучної бджолоїної колонії. Для виконання задач дипломного проєкту, проводиться порівняльний аналіз цих алгоритмів за допомогою компонент експерименту. Як описано у підрозділі 1.1, доступ до частини експериментів є лише у дослідника, а ц звичайного користувача є доступ лише до частини розв'язання задач. У частині розв'язання задач немає можливості обрати алгоритм, яким розв'язуватиметься задача, оскільки це недоцільно для використання системи

у реальних умовах. Отже, в реальних умовах не використовуватимуться такі компоненти:

- компонента користувацького інтерфейсу експериментів;
- компонента бізнес-логіки експериментів;
- дві компоненти розв'язувачів тих алгоритмів, які показали гірші результати при порівняльному аналізі.

Для розв'язання задачі маршрутизації БПЛА у реальних умовах за замовчуванням та без можливості вибору застосовуватиметься компонента того алгоритму, який показує кращі результати.

4.3.2 Специфікація функцій

Клієнтська частина системи маршрутизації БПЛА складається з ряду класів:

- сервіси – призначені для виорання запитів до серверної частини системи та оновлення і оновити сторів отриманими даними);
- стори – призначені для зберігання даних, які потім використовуються компонентами бізнес-логіки;
- компоненти бізнес логіки (BLoC) – призначені для зберігання стану сторінки, працюють як хендлери сторінок і є способом розділення реалізації логіки сторінок на два шари, а саме шар відображення даних та шар логіки і поточного стану сторінки.

У таблицях опишемо методи для тих класів, які реалізовані в рамках імплементації частини клієнтської сторони системи у цьому дипломному проєкті.

AuthService – це сервіс, який відповідає за процес авторизації у системі. У таблиці 4.1 наведено перелік методів сервісу AuthService.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						70
Зм.	Лист	№ докум.	Підпис	Дата		

Таблиця 4.1 – Методи сервісу AuthService

Метод	Параметри		Опис
	Вхідні	Вихідні	
login	loginData - об'єкт, що містить ім'я користувача і пароль	Відсутні	Даний метод робить запит до сервера і у випадку успішної авторизації записує JWT токен в куки і оновлює стор AppStateStore даними, що прийшли з сервера.
logout	Відсутні	Відсутні	Даний метод видаляє JWT токен з куків та оновлює стан стору AppStateStore.

AppStateStore – це стор, який зберігає загальні дані про стан системи, а саме чи увійшов до облікового запису користувач та масив ролей користувача. Не має методів.

LoginScreenBLoC – це компонента бізнес-логіки (BLoC) сторінки авторизації. Компонента зберігає поточний стан форми авторизації і містить методи, кожен, описані у таблиці 4.2. Кожен з методів LoginScreenBLoC є хендлером.

Таблиця 4.2 – Методи компоненти бізнес-логіки LoginScreenBLoC

Метод	Параметри		Опис
	Вхідні	Вихідні	
setUsername	Нове значення поля імені користувача у формі логіну	Відсутні	Даний метод оновлює ім'я користувача у стані форми логіну.
setPassword	Нове значення поля паролю користувача у формі логіну	Відсутні	Даний метод оновлює пароль у стані форми логіну.

submitForm	Відсутні	Відсутні	Даний метод викликає метод login service AuthService і передає туди стан форми логіну.
------------	----------	----------	--

ExperimentScreenBLoC - це компонента бізнес-логіки сторінки проведення експерименту. Компонента містить поточні дані форми, ряд геттерів, які обчислюють дані відображення та декілька публічних методів, які описані у таблиці 4.3.

Таблиця 4.3 – Методи компоненти бізнес-логіки ExperimentScreenBLoC

Метод	Параметри		Опис
	Вхідні	Вихідні	
setNumberOfPoints	value – нове значення поля форми	Відсутні	Даний метод оновлює значення кількості точок (розмірності задачі).
setNumberOfRuns	value – нове значення поля форми	Відсутні	Даний метод оновлює значення кількості прогонів (кількість задач, які необхідно вирішити в межах експерименту).
setAlgorithm	value – нове значення поля форми	Відсутні	Даний метод змінює алгоритм, яким необхідно вирішити задачу.
submit	Відсутні	Відсутні	Даний метод приводить дані форми до необхідних типів і викликає метод SolverService, який відповідає за запит до відповідного ендпоінту проведення експерименту на сервері.

NavBarBLoC – це компонента бізнес-логіки компоненти навігаційної панелі. Компонента бізнес-логіки містить геттер, який визначає чи має користувач доступ до сторінки проведення експерименту і один публічний метод, який описано у таблиці 4.4.

Таблиця 4.4 – Методи компоненти бізнес-логіки NavBarBLoC

Метод	Параметри		Опис
	Вхідні	Вихідні	
logoutHandler	Відсутні	Відсутні	Даний метод викликає метод AuthService'у, який відповідає за вихід із застосунку.

Також частиною даного дипломного проєкту є реалізація процесу взаємодії клієнтської частини з серверною частиною, налаштування API. Для цього було реалізовано функції, що описані у таблиці 4.5.

Таблиця 4.5 – Функції клієнтської частини для роботи з API

Функція	Параметри		Опис
	Вхідні	Вихідні	
fetchAPI	url – url, за яким потрібно зробити запит; typeGuard – функція, яка перевіряє чи коректні дані прийшли з сервера options – об'єкт, що містить налаштування запиту returnAsBlob – чи	Дані, які надіслав сервер	Дана функція є обгорткою на утиліту fetch, яка робить вказаний запит до сервера, перевіряє дані на коректність і містить базову обробку помилок.

	потрібно повернути результат як Blob (необхідно для скачування файлів)		
loginAPI	body – об'єкт, що містить ім'я користувача і пароль	Повертає об'єкт, що містить JWT та масив ролей даного користувача	Дана функція робить запит, що відповідає за логін користувача.
calculateRouteAPI	Об'єкт, що містить вихідні дані задачі	Повертає об'єкт, що містить розв'язок задачі і значення функції оцінки	Дана функція робить запит, що відповідає за розв'язання заданої користувачем задачі.
Perform ExperimentAPI	Об'єкт, що містить назву алгоритму, над яким потрібно поставити експеримент, кількість запусків та розмірність задачі	Повертає масив розв'язків, які були отримані в ході проведення експерименту	Дана функція відповідає за проведення результатів експерименту.
downloadLast ResultAPI	Відсутні	Повертає останній результат розв'язання та проведення експерименту, якщо він існує в системі	Дана функція, що відповідає за запит до ендпоінту завантаження останнього результату користувача.

Для імплементації розв'язувача, що розв'язує задачу маршрутизації БПЛА алгоритмом штучної бджолоїної колонії, було реалізовано функції, описані у таблиці 4.6.

Таблиця 4.6 – Функції розв'язувача методом штучної бджолоїної колонії

Функція	Вхідні параметри	Вихідні параметри	Опис
createBees AlgorithmSolver	params – об'єкт, що містить максимальну допустиму кількість ітерацій впродовж яких рекордний розв'язок не змінюється, потужність початкової популяції, відсоток розв'язків популяції, що є кращими, окіл проведення локальної оптимізації, тобто відсоток від поточної популяції	Повертає розв'язувач задачі, представлений у вигляді функції	Дана функція створює розв'язувач задачі методом штучної бджолоїної колонії з переданими параметрами.
Функція (розв'язувач), яку повернув createBees AlgorithmSolver	pointsToObserve – масив цілей startBase – початкове депо anotherBase – додаткове депо chargeTime – час перезарядки БПЛА між польотами (у хвиликах) maxFlightTime – максимальний автономний час роботи	Повертає об'єкт, з такими полями як: route – побудований розв'язувачем маршрут fitness – значення цільової функції	Дана функція отримує вхідні дані, розв'язує задачу бджолиним алгоритмом і повертає результат розв'язання.

	БПЛА без перезарядки (у хвилинах) speed – швидкість польоту	розв'язку	
pairs	sols – масив розв'язків	Повертає масив пар розв'язків	Дана функція розбиває масив переданих розв'язків на максимальну кількість випадкових пар.
maxEvenNumber	number – число	Повертає максимальне парне число, що не більше за задане	Дана функція обчислює і повертає максимальне парне число, що не більше заданого.
rand	minimum – нижня границя maximum – верхня границя	Повертає випадкове ціле число	Дана функція обчислює випадкове ціле число у межах нижньої (включно) та верхньої границі (не включно).
getPossible Permutations	route – маршрут nearestPointsMapping – хеш-таблиця, що містить найближчі сусідні точки для	Повертає масив нових розв'язків	Дана функція обчислює і повертає масив нових розв'язків

Зм.	Лист	№ докум.	Підпис	Дата

	кожної точки valid – функція, яка перевіряє чи є розв’язок таким, що задовільняє умови задачі bases – масив депо		утворених на основі заданого таких, що відповідають обмеженням задачі.
getDescendants	route1 – перший розв’язок, route2 – другий розв’язок valid – функція, яка перевіряє чи є розв’язок таким, що задовільняє умови задачі	Повертає масив нових роз’язків	Дана функція повертає масив нових розв’язків, утворених схрещуванням двох заданих розв’язків.

Висновок до розділу

В даному розділі описані засоби розробки системи та архітектура імплементованої системи.

Серед засобів розробки розглянуто мову розробки, середовище та технології розробки програмного забезпечення.

Проаналізовано тенденції розвитку обраних засобів розробки, їх популярність. На основі переваг та недоліків аргументовано вибір засобів розробки. Для опису архітектури програмного середовища, побудовано та детально розглянуто діаграми компонентів та послідовності, а також розписано специфікацію функцій клієнтської частини системи, які розроблялися у даному дипломному проєкті.

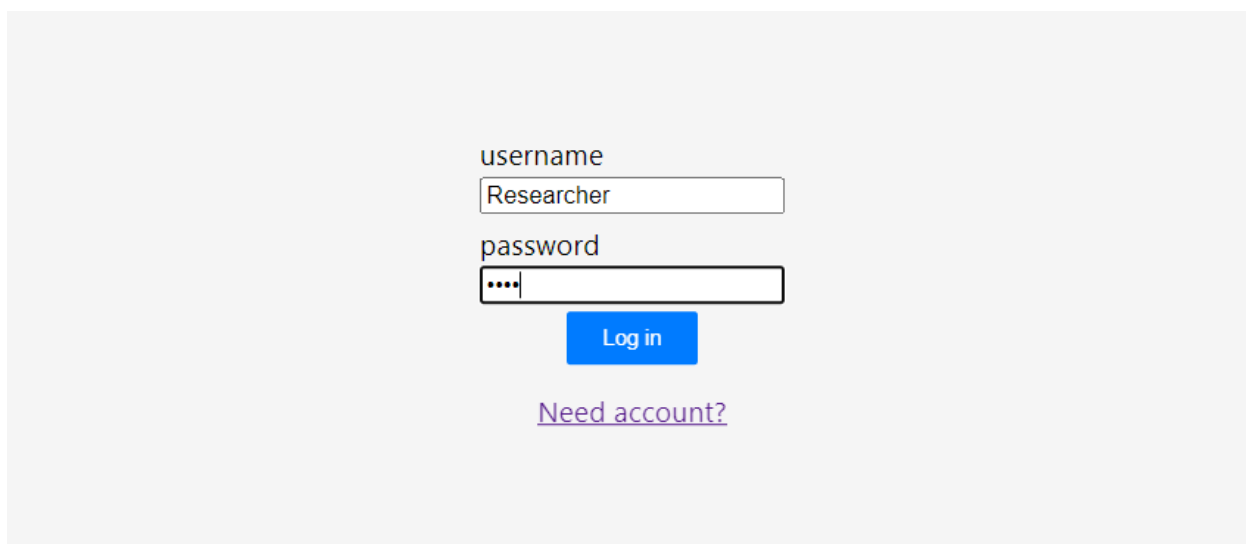
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Після запуску системи користувач за замовчуванням потрапляє на форму авторизації, стан входу у систему, зображений на рисунку 5.1. На цій сторінці користувач може увійти в обліковий запис, зареєстрований у системі, виконавши таку послідовність дій:

- ввести логін;
- ввести пароль;
- підтвердити ініціацію входу в обліковий запис з введеними логіном та паролем кнопкою <Зареєструватися>.

Після успішного входу у обліковий запис, система навігує зареєстрованого користувача на сторінку розв’язання задачі маршрутизації, при цьому з правами типу користувача, вхід до облікового запису якого було здійснено (звичайний користувач/дослідник).



The image shows a login form with the following elements:

- A label 'username' above a text input field containing the text 'Researcher'.
- A label 'password' above a text input field containing four dots '....' to indicate a masked password.
- A blue button labeled 'Log in' positioned below the password field.
- A purple underlined link labeled 'Need account?' positioned below the 'Log in' button.

Рисунок 5.1 – Форма авторизації, вхід до облікового запису

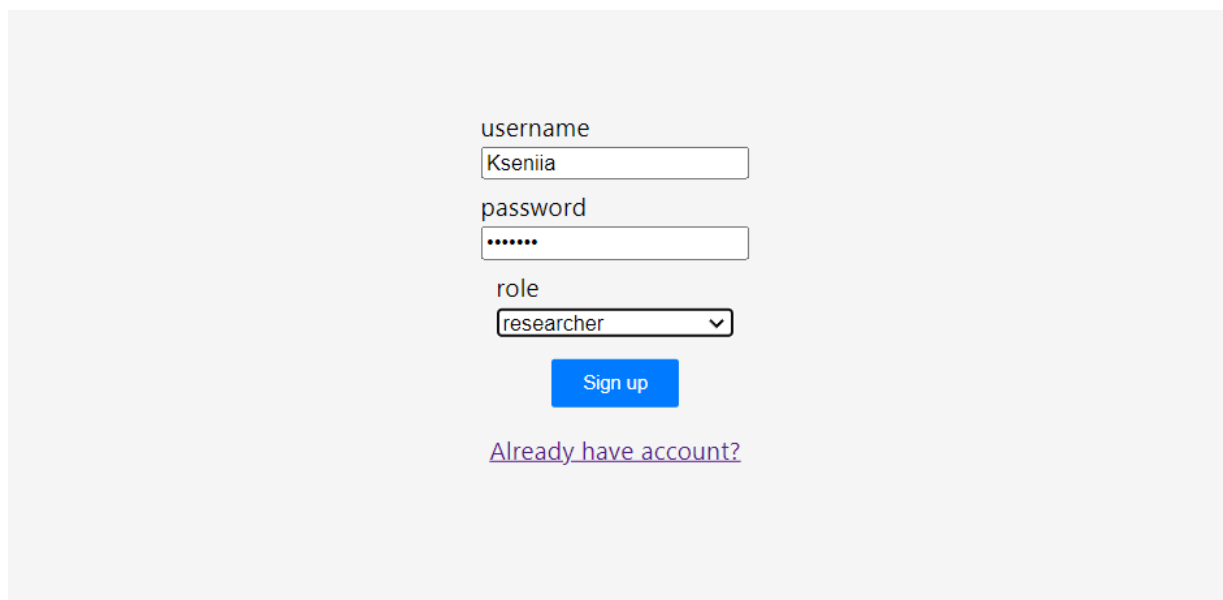
Якщо з відкритого за замовчуванням стану форми авторизації, входу в

обліковий запис, користувач ініціює реєстрацію облікового запису відповідною кнопкою, то система перенаправляє користувача на стан реєстрації форми авторизації, проілюстрований на рисунку 5.2.

Для реєстрації облікового запису, користувач виконує наступні дії:

- ввести логін (повинен бути унікальним у системі);
- ввести пароль;
- обрати тип користувача, для якого створюється обліковий запис (дослідник/звичайний користувач);
- підтвердити ініціацію реєстрації з введеним логіном, паролем, та обраним з випадаючого списку типом користувача.

При цьому, зі стану реєстрації форми авторизації, користувач може перейти назад до стану входу у обліковий запис форми авторизації.



The image shows a registration form with the following elements:

- A text input field labeled "username" containing the text "Kseniia".
- A text input field labeled "password" containing six dots ".....".
- A dropdown menu labeled "role" with "researcher" selected and a downward arrow.
- A blue button labeled "Sign up".
- A link labeled "Already have account?" in purple text.

Рисунок 5.2 – Форма авторизації, реєстрація облікового запису

Після успішної авторизації у системі, користувача навігує на сторінку розв’язання задачі маршрутизації БПЛА, зображеної на рисунку 5.3. Клієнтська частина даної сторінки розроблена у частині комплексного дипломного проекту Карпельової Ірини. Навігаційна панель на сторінці розв’язання задачі не відрізняється від навігаційної панелі сторінки

проведення експериментів, описаної нижче.

The screenshot shows a web interface for solving a flight path optimization problem. At the top right, there is a 'Log out' button. Below it, there is a 'Solver' section with a 'D3' toggle switch and a 'Google Maps' link. The interface includes several input fields: 'Point 1' (Latitude, Longitude, Label: Point 1), 'Start base' (Latitude, Longitude, Label: Base 1), and 'Another base' (Latitude, Longitude, Label: Base 2). There are also fields for 'Charge time (min)', 'Max flight time (min)', and 'Speed (km/h)'. A file upload section shows 'Choose File' and 'No file chosen'. At the bottom, there are 'Submit' and 'Download last result' buttons.

Рисунок 5.3 – Сторінка розв’язання задачі маршрутизації БПЛА

Як на сторінці розв’язання задачі, так на сторінці проведення експериментів є навігаційна панель зверху сторінки. Навігаційна панель однакова для обох сторінок системи, але відрізняється в залежності від ролі користувача (дослідник або звичайний користувач).

Навігаційна панель звичайного користувача, зображена зверху екранної форми на рисунку 5.3, містить кнопку виходу з системи. Кнопка виходу з системи закриває поточну сесію користувача та перенаправляє користувача на дефолтний стан (вхід до облікового запису) форми авторизації з попередньо заповненими даними, логіном та паролем користувача, з акаунту якого виконано вихід. Також навігаційна панель звичайного користувача містить кнопку навігації на сторінку розв’язання задачі маршрутизації БПЛА, що слугує як кнопка перезавантаження сторінки.

Функціонал проведення експерименту недоступний для звичайного користувача.

Навігаційна панель дослідника, зображена зверху екранної форми на рисунках 5.4 – 5.6, окрім кнопки виходу з облікового запису, що працює аналогічно як для звичайного користувача, та кнопки навігації на сторінку

розв'язання задачі маршрутизації БПЛА, містить кнопку навігації на сторінку проведення експерименту, оскільки цей функціонал доступний користувачу з роллю дослідник, на відміну від користувача зі звичайною роллю.

Після ініціювання навігації на сторінку проведення експерименту дослідника (звичайному користувачу функціонал проведення експериментів недоступний), користувач перенаправляється системою на сторінку проведення експериментів, зображену на рисунках 5.4 – 5.6.

Оскільки система працює не лише з алгоритмом штучної бджолоїної колонії, реалізований у даному дипломному проєкті, а й з алгоритмами табу-пошуку та мурашиного алгоритму, розроблених у частинах комплексного дипломного проєкту Олександра Чорного та Карпельової Ірини відповідно, сторінка проведення експерименту розроблена для роботи з усіма трьома алгоритмами.

Для проведення експерименту дослідник виконує наступні дії:

- вводить кількість прогонів експерименту;
- вводить розмірність задач для проведення експерименту (однакова для усіх прогонів, визначає кількість цілей у згенерованих задачах);
- обирає з трьох доступних алгоритмів з випадючого списку один, для якого буде проведено експеримент;
- вводить кожен з параметрів обраного алгоритму (параметри різні для кожного з алгоритмів);
- ініціює проведення експерименту для заданих вхідних даних, відповідною кнопкою <Підтвердити>.

Коли дослідник обрав алгоритм, для якого проводитиметься експеримент, система одразу відображає параметри, що повинні бути вказані для проведення експерименту з даним алгоритмом, без додаткового підтвердження алгоритму.

[Perform Experiments](#)
[Solver](#)
[Log out](#)

Number of points:
 Number of runs:
 Algorithm:

Number of tabu searches:

Tabu Tenure:

Max iterations without improvement:

RESULTS:

Best time: 54.51

Mean time: 62.2

Time median: 61.15

Standart deviation: 4.8

Total time (min):
 Run №1: 68.35
 Run №2: 66.13
 Run №3: 61.15
 Run №4: 60.86
 Run №5: 54.51

Рисунок 5.4 – Сторінка проведення експерименту – табу-пошук

[Perform Experiments](#)
[Solver](#)
[Log out](#)

Number of points:
 Number of runs:
 Algorithm:

Solution population size:

Number of best solutions:

Max iterations without improvement:

RESULTS:

Best time: 0

Mean time: 0

Time median: 0

Standart deviation: 0

Total time (min):

Рисунок 5.5 – Сторінка проведення експерименту – алгоритм штучної бджолоїної колонії

[Perform Experiments](#)
[Solver](#)
[Log out](#)

Number of points:
 Number of runs:
 Algorithm:

Number of ants:

Evaporations rate:

Heuristic information importance:

Pheromone importance:

Max iterations without improvement:

RESULTS:

Best time: 54.51

Mean time: 62.2

Time median: 61.15

Standart deviation: 4.8

Total time (min):
 Run №1: 68.35
 Run №2: 66.13
 Run №3: 61.15
 Run №4: 60.86
 Run №5: 54.51

Рисунок 5.6 – Сторінка проведення експерименту – мурашиний алгоритм

Після завершення експерименту система виводить на екран користувача результати проведення експерименту. Приклад результатів проведення експерименту представлено на рисунку 5.7. А саме, система відображає наступне по усім прогонам експерименту;:

- краще значення цільової функції;
- середнє значення цільової функції;
- медіану (медіани) значень цільової функції;
- середньоквадратичне відхилення цільової функції;
- значення цільової функції кожного прогону, при цьому прогони пронумеровані.

Також, система відображає кнопку <Завантажити останній результат експерименту>, що дозволяє досліднику ініціювати завантаження на пристрій файлу, з даними та результатами останнього проведеного під поточним обліковим записом експерименту.

• Perform Experiments				Log out	
• Solver					
Number of points:	Number of runs:	Algorithm:	RESULTS:		
15	5	bees	Total time (min):		
Solution population size:			Run №1: 68.35		
100			Run №2: 66.13		
Number of best solutions:			Run №3: 61.15		
44			Run №4: 60.86		
Max iterations without improvement:			Run №5: 54.51		
22			Best time:		
Submit	Download last experiment result		54.51		
			Mean time:		
			62.2		
			Time median:		
			61.15		
			Standart deviation:		
			4.8		

Рисунок 5.7 – Результати проведення експерименту

Також, на сторінці проведення експерименту для полів вводу даних, як параметрів експерименту, так і параметрів алгоритмів, на стороні клієнтської частини, реалізовані валідації, що описані у таблицях 5.2 – 5.4 тестових сценаріїв пункту дипломного проєкту 5.2.3. На рисунку 5.8 наведено приклад неуспішної валідації одного з параметрів алгоритму штучної бджолоїної

колонії. На рисунку 5.9 наведено приклад неуспішної валідації порожнього поля.

- Perform Experiments
- Solver

Number of points:
 Number of runs:
 Algorithm:

Solution population size:

Number of best solutions:

Num. of best solutions should be a natural number

Max iterations without improvement:

RESULTS:

Best time: 57.91

Mean time: 63.17

Time median: 63.02

Standart deviation: 3.31

Total time (min):

Run №1: 67.47

Run №2: 57.91

Run №3: 65.71

Run №4: 61.72

Run №5: 63.02

Рисунок 5.8 – Приклад валідації полів – некоректний формат

- Perform Experiments
- Solver

Number of points:
 Number of runs:
 Algorithm:

Number of ants:

Evaporations rate:

Heuristic information importance:

Heur. info importance is required

Pheromone importance:

Max iterations without improvement:

RESULTS:

Best time: 60.9

Mean time: 64.23

Time median: 64.66

Standart deviation: 2.27

Total time (min):

Run №1: 64.66

Run №2: 67.22

Run №3: 60.9

Run №4: 65.87

Run №5: 62.51

Рисунок 5.9 – Приклад валідації полів – порожнє поле

5.2 Випробування програмного продукту

У підрозділі описано набори тестових сценаріїв для перевірки коректності роботи системи маршрутизації БПЛА, відповідно до очікуваної роботи функціоналу системи, що описана у підрозділі 1.1, пунктах опису процесу діяльності та опису функціональної моделі системи маршрутизації БПЛА, а також відштовхуючись від цілей та задач розробки, описаних у пункті

1.3.2. Також, у цьому підрозділі описано мету та загальні положення процесу тестування імплементованої системи.

5.2.1 Мета випробувань

Метою випробувань БПЛА є перевірка працездатності реалізованої системи маршрутизації, коректності та повноти покриття очікуваного функціоналу системи, описаного у підрозділі 1.1 та підрозділі 1.3.

5.2.2 Результати випробувань

У комплексно розробленій системі маршрутизації БПЛА реалізована велика кількість основного та додаткового функціоналу. Проведемо тестування «sunny-day» сценаріїв частини основного функціоналу клієнтської сторони системи, який було імплементовано у даному дипломному проєкті, а також тестування «rainy-day» сценаріїв для валідацій полів сторінки проведення експерименту на стороні клієнтської частини системи. Протестуємо наступний функціонал:

- функціонал сторінки проведення експерименту дослідником, таблиця 5.1;
- валідація на стороні клієнта полів проведення експерименту, таблиця 5.2 – 5.4;
- проведення експерименту для аналізу алгоритму штучної бджолоїної колонії, таблиця 5.5;
- робота навігаційної панелі для авторизації, таблиця 5.6;
- робота навігаційної панелі для перемикання між сторінками розв’язання задачі та проведення експерименту, таблиця 5.7 – 5.8;
- авторизація зі сторони клієнта, таблиця 5.9 – 5.10.

Таблиця 5.1 – Проведення експерименту дослідником

Мета тестового сценарію	Перевірка функціоналу сторінки проведення експерименту
Початковий стан системи	Користувач увійшов у акаунт дослідника та перейшов на сторінку проведення експерименту
Вхідні дані	Згенеровані координати пунктів генератором задач
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Ввести розмірність задач, для яких проводитиметься експеримент (кількість цілей) 2. Ввести кількість прогонів експерименту (кількість задач, які буде згенеровано) 3. Обрати алгоритм для проведення експерименту з випадуючого списку 4. Коли система відобразить необхідні для вибраного алгоритму параметри, коректно заповнити усі параметри 5. Ініціювати проведення експерименту відповідною кнопкою
Очікуваний результат	Система успішно проводить експеримент, відображає користувачу результати експерименту, а саме час проходження маршруту для кожного з прогонів експерименту, середній час проходження маршруту та середньоквадратичне відхилення часу проходження маршруту для усіх прогонів. Система розблоковує можливість ініціювання завантаження JSON файлу з результатами проведеного експерименту.
Стан системи після проходження тестового сценарію	Система успішно провела експеримент, відобразила користувачу результати експерименту, а саме час проходження маршруту для кожного з прогонів експерименту, середній час проходження маршруту та середньоквадратичне відхилення часу проходження маршруту для усіх прогонів експерименту. Також система розблокувала можливість ініціювання завантаження JSON файлу з результатами проведеного експерименту.
Результат проходження тестового сценарію	Пройдено успішно

Зм.	Лист	№ докум.	Підпис	Дата

Таблиця 5.2 – Валідація на стороні клієнта усіх полів проведення експерименту, крім параметрів швидкості випаровування та важливості евристичної інформації мурашиного алгоритму – цілі числа строго більші за нуль

Мета тестового сценарію	Частина перевірки коректності валідацій на стороні клієнта усіх полів проведення експерименту, крім параметрів швидкості випаровування та важливості евристичної інформації мурашиного алгоритму – цілі числа строго більші за нуль
Початковий стан системи	Користувач увійшов у акаунт дослідника та перейшов на сторінку проведення експерименту
Вхідні дані	Відсутні
Кроки тестового сценарію	1. В одне з полів параметрів експерименту (кількість прогонів та розмірність задач, тобто кількість цілей у згенерованих задачах) або алгоритму (для кожного з алгоритмів свій набір параметрів), крім параметрів швидкості випаровування та важливості евристичної інформації мурашиного алгоритму, ввести будь-який символ окрім цілого числа, що строго більше нуля 2. Ініціювати проведення експерименту відповідною кнопкою
Очікуваний результат	Система ідентифікує усі некоректно заповнені поля і виводить відповідні тексти помилок для кожного з полів про те що можна вводити лише додатні цілі числові значення.
Стан системи після проходження тестового сценарію	Система ідентифікувала усі некоректно заповнені поля і вивела відповідні тексти помилок для кожного з полів про те що можна вводити лише додатні цілі числові значення.
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.3 – Валідація на стороні клієнта полів параметрів швидкості випаровування та важливості евристичної інформації мурашиного алгоритму – дробові числа від нуля до одиниці

Мета тестового сценарію	Частина перевірки коректності валідацій полів сторінки проведення експерименту на стороні клієнта, полів параметрів швидкості випаровування та важливості евристичної інформації мурашиного алгоритму – дробові числа від нуля до одиниці
Початковий стан системи	Користувач увійшов у акаунт дослідника, перейшов на сторінку проведення експерименту та обрав мурашиний алгоритм
Вхідні дані	Відсутні
Кроки тестового сценарію	<ol style="list-style-type: none"> В поле параметру швидкості випаровування або важливості евристичної інформації, ввести будь-який символ окрім десяткового (записаного через крапку або через кому) числа від нуля включно до одиниці включно, з точністю до десятитисячних Ініціювати проведення експерименту відповідною кнопкою
Очікуваний результат	Система ідентифікує усі некоректно заповнені поля і виводить відповідні тексти помилок для кожного з полів про те що можна вводити лише дробові числові значення від нуля включно до одиниці включно, з точністю до десятитисячних.
Стан системи після проходження тестового сценарію	Система ідентифікувала усі некоректно заповнені поля і вивела відповідні тексти помилок для кожного з полів про те що можна вводити лише невід’ємні числові значення від нуля включно до одиниці включно, з точністю до десятитисячних.
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.4 – Валідація на стороні клієнта полів проведення експерименту – параметр бджолиного алгоритму кращих рішень, що розглядаються менший або рівний параметру потужності популяції

Мета тестового сценарію	Частина перевірки коректності валідацій полів сторінки проведення експерименту на стороні клієнта – параметр бджолиного алгоритму кращих рішень, що розглядаються менший або рівний параметру потужності популяції
Початковий стан системи	Користувач увійшов у акаунт дослідника, перейшов на сторінку проведення експерименту та обрав бджолиний алгоритм для проведення експерименту
Вхідні дані	Відсутні
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. В поле параметру потужності популяції бджолиного алгоритму ввести певне коректне значення 2. В поле параметру бджолиного алгоритму кращих рішень, що розглядаються ввести значення, більше значення, яке введено у поле параметру потужності популяції на попередньому кроці 3. Ініціювати проведення експерименту відповідною кнопкою
Очікуваний результат	<p>Система ідентифікує що поля заповнені некоректно і виводить відповідний текст помилки про допустиме значення для цих полів.</p> <p>Якщо дана помилка валідації ініційована одночасно з помилкою валідації тестового сценарію, описаній у таблиці 5.2, то відображається лише помилка, описана у таблиці 5.2 (оскільки валідації відбуваються послідовно, до першого неуспішного результату).</p>
Стан системи після проходження тестового сценарію	<p>Система ідентифікувала усі некоректно заповнені поля і вивела відповідний текст помилки про допустиме значення для цих полів.</p> <p>Коли помилка валідації ініційована одночасно з помилкою валідації тестового сценарію, описаній у таблиці 5.2, то відобразилася лише помилка, описана у таблиці 5.2 (оскільки валідації</p>

	відбуваються послідовно, до першого неуспішного результату).
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.5 – Проведення експерименту для аналізу алгоритму штучної бджолоїної колонії

Мета тестового сценарію	Перевірка функціоналу проведення експерименту алгоритмом штучної бджолоїної колонії
Початковий стан системи	Користувач увійшов у акаунт дослідника, перенаправився на сторінку проведення експерименту та обрав бджолоїний алгоритм
Вхідні дані	Згенеровані вхідні дані генератором задач
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Коректно заповнити усі параметри експерименту (кількість прогонів та розмірність задач, тобто кількість цілей у згенерованих задачах) 2. Коректно заповнити усі параметри бджолоїного алгоритму (потужність початкової популяції, кількість кращих рішень для оптимізації, кількість ітерацій, впродовж якої розв'язок не покращується і процес розв'язання завершується) 3. Ініціювати розв'язання задачі
Очікуваний результат	Алгоритм штучної бджолоїної колонії коректно розв'язує усі згенеровані задачі, з прийнятними результатами, у результатах по усім прогонах експерименту немає значних відхилень та аномалій.
Стан системи після проходження тестового сценарію	Система відображує результати проведення експерименту алгоритмом штучної бджолоїної колонії, результати прийнятні, по усім прогонах експерименту немає значних відхилень та аномалій.
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.6 – Робота навігаційної панелі для авторизації

Мета тестового сценарію	Перевірка функціоналу навігаційної панелі для авторизації – вихід з акаунту
-------------------------	---

Початковий стан системи	Авторизований користувач (звичайний або дослідник) на будь-якій сторінці системи
Вхідні дані	Відсутні
Кроки тестового сценарію	1. Користувач ініціює вихід з акаунту
Очікуваний результат	Система анонімізує користувача та навігує на форму з авторизацією з попередньо заповненими даними для входу, що були введені для входу у акаунт, для якого було ініційовано вихід з акаунту
Стан системи після проходження тестового сценарію	Система анонімізувала користувача та перенаправила на форму з авторизацією з попередньо заповненими даними для входу, що були введені для входу у акаунт, для якого було ініційовано вихід з акаунту
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.7 – Робота навігаційної панелі для навігації між сторінками – панель звичайного користувача

Мета тестового сценарію	Перевірка функціоналу навігаційної панелі для навігації між сторінками для звичайного користувача
Початковий стан системи	Користувач (не дослідник) увійшов у акаунт та знаходиться на будь-якій сторінці системи (для звичайного користувача доступна лише сторінка розв'язання задачі)
Вхідні дані	Відсутні
Кроки тестового сценарію	Відсутні
Очікуваний результат	Система не дає можливість перенаправитися на іншу сторінку (оскільки для звичайного звичайного користувача доступна лише сторінка розв'язання задачі, сторінка проведення експериментів доступна лише досліднику)
Стан системи після проходження	Система не дає користувачу (не досліднику) можливість перенаправитися на іншу сторінку

тестового сценарію	
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.8 – Робота навігаційної панелі для навігації між сторінками – панель дослідника

Мета тестового сценарію	Перевірка функціоналу навігаційної панелі для навігації між сторінками для дослідника
Початковий стан системи	Дослідник увійшов у акаунт та знаходиться на будь-якій сторінці системи
Вхідні дані	Відсутні
Кроки тестового сценарію	1. Користувач ініціює навігацію на іншу сторінку (розв'язання задачі або проведення експерименту)
Очікуваний результат	Система навігує користувача на обрану сторінку, після чого дозволяє повторити процедуру навігації на іншу сторінку
Стан системи після проходження тестового сценарію	Система перенаправила користувача на обрану сторінку, після чого дозволила повторити процедуру навігації на іншу сторінку
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.9 – Авторизація у системі зі сторони клієнта – вхід до акаунту дослідника або звичайного користувача

Мета тестового сценарію	Перевірка функціоналу авторизації у системі – вхід до акаунту дослідника або звичайного користувача
Початковий стан системи	Анонімний користувач, знаходиться на формі авторизації у системі
Вхідні дані	Логін та пароль від акаунту звичайного користувача або дослідника
Кроки тестового сценарію	1. Ввести логін та пароль у форму авторизації (за замовчуванням на формі авторизації відкрито вхід до акаунту, а не реєстрація) 2. Ініціювати вхід до акаунту відповідною кнопкою

Очікуваний результат	Коли серверна частина повертає успішну перевірку даних для входу у акаунт та тип користувача (звичайний користувач чи дослідник), то клієнтська частина успішно виконує вхід до акаунту з правами відповідного типу користувача та навігує користувача на сторінку розв'язання задачі
Стан системи після проходження тестового сценарію	Після повернення серверною частиною успішного входу до акаунту та типу користувача (звичайний користувач чи дослідник), клієнтська частина системи успішно виконала вхід до акаунту з правами відповідного типу користувача та перенаправила користувача на сторінку розв'язання задачі
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.10 – Авторизація у системі зі сторони клієнта – реєстрація звичайного користувача

Мета тестового сценарію	Перевірка функціоналу авторизації у системі – реєстрація у системі звичайного користувача
Початковий стан системи	Анонімний користувач знаходиться на формі авторизації у системі
Вхідні дані	Відсутні
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Перейти на вкладку реєстрації у системі форми авторизації 2. Обрати з випадаючого списку ролей опцію реєстрації звичайним користувачем (обрано за замовчуванням) 3. Ввести логін та пароль для реєстрації 4. Ініціювати реєстрацію акаунту відповідною кнопкою
Очікуваний результат	Коли серверна частина повертає успішну реєстрацію (логін користувача унікальний), то клієнтська навігує користувача на сторінку розв'язання задачі з відповідними правами доступу до функціоналу системи
Стан системи після проходження	Серверна частина повернула успішну реєстрацію та тип користувача, клієнтська частина системи

тестового сценарію	перенаправила користувача на сторінку розв'язання задачі з відповідними правами доступу до функціоналу системи
Результат проходження тестового сценарію	Пройдено успішно

Таблиця 5.11 – Авторизація у системі зі сторони клієнта – реєстрація дослідника

Мета тестового сценарію	Перевірка функціоналу авторизації у системі – реєстрація у системі дослідника
Початковий стан системи	Анонімний користувач знаходиться на формі авторизації у системі
Вхідні дані	Відсутні
Кроки тестового сценарію	<ol style="list-style-type: none"> 1. Перейти на вкладку реєстрації у системі форми авторизації 2. Обрати опцію реєстрації дослідником з випадаючого списку ролей 3. Ввести логін та пароль для реєстрації. Ініціювати реєстрацію акаунту відповідною кнопкою
Очікуваний результат	Коли серверна частина системи повертає успішну реєстрацію, то клієнтська частина перенаправляє користувача на сторінку розв'язання задачі з відповідними правами доступу до функціоналу системи
Стан системи після проходження тестового сценарію	Серверна частина повернула успішну реєстрацію та тип користувача, клієнтська частина системи перенаправила користувача на сторінку розв'язання задачі з відповідними правами доступу до функціоналу системи
Результат проходження тестового сценарію	Пройдено успішно

Висновок до розділу

В даному розділі складено керівництво користувача з покроковою ілюстрацією та описом ключових користувацьких сценаріїв роботи з системою.

Також проведено випробування розробленої частини функціоналу клієнтської сторони системи. Відштовхуючись від мети та загальних положень процесу тестування, розроблено тестові сценарії основних користувацьких сценаріїв взаємодії користувача з системою. Тестові сценарії оформлено у таблиці, для кожного наведено результат випробування. Після мануального проходження тестових сценаріїв, зроблено висновок, що система функціонує відповідно до очікувань.

					<i>IC92.050BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		95

ВИСНОВКИ

Результатом роботи над дипломним проектом, що є частиною комплексного проекту, є розроблена частина функціоналу сторони клієнта системи маршрутизації БПЛА, що складається з модуля дослідження алгоритмів розв'язання задачі, а також модуля складання маршрутів польоту БПЛА для заданої множини цілей методом локального пошуку. Проаналізована предметна область та показана актуальність задачі маршрутизації БПЛА у різних сферах.

Для роботи з системою виокремлено дві ролі, а саме звичайний користувач та дослідник, задля зручності застосування системи у реальному житті. Роль дослідника має розширені права доступу до системи, та, на відміну від звичайного користувача, додатково до модуля розв'язання задачі має доступ до модуля проведення експериментів. Описані варіанти використання системи та процес діяльності з точки зору користувача та дослідника.

Проаналізовано аналог системи Mission Planner та виявлено переваги проєктованої системи над ним, а саме націленість на автоматизоване планування маршрутів, швидкість та простота використання, а також сумісність з пристроями користувача.

У розділі інформаційного забезпечення визначено вхідні та вихідні дані для роботи системи, описано процес взаємодії користувача з файлами масивів інформації та використовуваною базою даних зі сторони клієнтської частини.

Сформульовано змістовну та математичні постановки задачі маршрутизації БПЛА. Для вибору методу розв'язання проаналізовано класифікацію задач транспортної маршрутизації, та методів її розв'язання. Обрано, спроектовано, програмно реалізовано та імплементовано алгоритм штучної бджолиної колонії. Метод адаптовано під задачу, яку розв'язує проєктована система, а імплементацию протестовано та проаналізовано в експерименті.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						96
Зм.	Лист	№ докум.	Підпис	Дата		

Була наведена аргументація та опис обраних технологій розробки та описані вимоги до технічного забезпечення користувачів системи.

Архітектура програмного забезпечення проілюстрована діаграмою компонентів комплексної системи, діаграмами послідовності взаємодії користувача та компонент системи для процесів розв'язання задачі маршрутизації БПЛА та проведення експерименту, а також діаграмою структури бази даних.

Описане проілюстрована знімками розробленої системи інструкція користувача, що ілюструє основні сценарії взаємодії з системою як звичайного користувача, так і дослідника. Також проведені випробування розробленої системи маршрутизації БПЛА відповідно до положень проведення випробувань. Мануальне проходження розроблених тестових сценаріїв підтвердило коректність роботи програми та повноту покриття вимог до системи її функціоналом.

Отже, оскільки аналіз розробленого алгоритму штучної бджолоїної колонії підтвердив ефективність результатів, а тестування розробленої системи підтвердило коректність її роботи та повноту покриття розроблених у дипломному проєкті вимог, дана система маршрутизації БПЛА може бути застосована в реальних умовах як для задач обстеження певних наборів цілей на регулярній основі певним військовим підрозділом для виконання задач, що описано у постановці задачі у підрозділі 1.3, так і для розв'язання задач інших сфер людської діяльності, що підходять під математичну постановку, описану у підрозділі 3.2.

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						97
Зм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Шраменко Н. Сучасні технології в машинобудуванні та транспорті. / Н. Шраменко, В. Шраменко, О. Соларьов. – 2021. – С. 35. <https://eforum.lntu.edu.ua/index.php/jurnal-mbf/article/view/505>
2. Любов П. Україна втрачає \$170 млн на день через заблоковані порти. / П. Любов, Ш. Сергій. – 2022. – URL: <https://forbes.ua/inside/morski-vorota-na-zamku-yak-pratsyuuyut-zablokovani-cherez-viynu-porti-21042022-5552>.
3. Дмитро У. Україна розпочне будівництво колії, яка з'єднає українську залізницю з європейською. / Уляницький Дмитро. – 2022. – URL: <https://forbes.ua/inside/ukraina-rozpochne-budivnitstvo-evrokolii-abi-zednati-svoyu-zalznitsyu-z-evropeyskoju-detali-proektu-30052022-6280#:~:text=%D0%A3%20%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D0%B0%D1%85%20%D0%84%D0%A1%20%D1%88%D0%B8%D1%80%D0%B8%D0%BD%D0%B0%20%D0%BC%D1%96%D0%B6,%D0%B0%20%D0%B2%20%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%96%20%E2%80%93%20%D0%BC%D0%BC>.
4. Копилов М. УЗ планує досягти експорту 1,5 млн т зерна на місяць до осені / Микола Копилов. – 2022. – URL: <https://www.railinsider.com.ua/uz-planuye-dosyagty-eksportu-15-mln-t-zerna-na-misyacz-do-oseni/>.
5. Олійник С. Черги на АЗС зникнуть, але ціни підскочать – прогнози / Світлана Олійник. – 2022. – URL: <https://ua-energy.org/uk/posts/cherhy-na-azs-znyknut-a-ot-tsiny-pidskochat>.
6. ACM Computing Surveys - Rich Vehicle Routing Problem: Survey / Jose Caceres Cruz, Pol Arias, Daniel Guimarans, Daniel Riera. – 2014.
7. Paolo Toth. Vehicle Routing Problems, Methods, and Applications /. – Society for Industrial and Applied Mathematics, 2015. – 480 с. – (2nd Edition).
8. Golden B. Vehicle Routing Problem: Latest Advances and New Challenges / Bruce Golden, S. Ragahavan, Edward Wasil. – USA: Springer Media,

					<i>IC92.050БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		98

2008. – 583 с.

9. Dynamic Fleet Management - Dynamic And Stochastic Vehicle Routing In Practice / Truls Flatberg, Geir Hasle, Oddvar Kloster, Eivind J. Nilssen. – 2007. – С. 41–63.

10. Гуляницький Л. Ф. Прикладні методи комбінаторної оптимізації / Л. Ф. Гуляницький, О. Ю. Мулеса., 2016. – 142 с.

11. Донець Г. П. Екстремальні задачі на комбінаторних конфігураціях / Г. П. Донець, Л. М. Колечкіна. – Полтава: ПУЕТ, 2011.

12. Яровий А. А. Математичні методи дослідження операцій. / А. А. Яровий, Л. М. Ваховська, Л. В. Крилик. – Вінниця: ВНТУ, 2020. – 87 с.

13. Кутовецький В. Я. Дослідження операцій / В.Я. Кутовецький. – Миколаїв: МДГУ ім. П. Могили, 2003. – 260 с.

14. Прийма С. М. Теорія алгоритмів: Навчальний посібник. / Прийма С.М. – Мелітополь: ФОП Однорог Т.В., 2018. – 2016 с.

15. Базилевич Р. Дослідження ефективності існуючих алгоритмів для розв'язання задачі комівояжера / Базилевич Р., Кутельмах Р., 2009. – 10 с.

16. Bruwer F. Petal-shaped clustering. / Frances Bruwer. – Johannesburg, 2018. – 53 с.

17. Kreveld K. Variations on Sweep Algorithms. / Marc van Kreveld., 2020. – 14 с.

18. Marco Dorigo. Ant Colony Optimization / Marco Dorigo. – London: The MIT Press, 2004. – 519 с.

19. Кос Е. The Bees Algorithm Theory, Improvements and Applications / Ebubekir Кос. – United Kingdom: Manufacturing Engineering Centre School of Engineering, 2010. – 255 с.

20. Melanie M. An Introduction to Genetic Algorithms / Mitchell Melanie. – Mitchell, Melanie: The MIT Press, 1996. – 162 с.

21. Л. Ф. Гуляницький. Одна спеціальна задача маршрутизації БПЛА / Л. Ф. Гуляницький, В. В. Сторчевий., 2019. – (вип. 34, № 1).

					<i>IS92.050BAK.004 ПЗ</i>	Арк.
						99
Зм.	Лист	№ докум.	Підпис	Дата		

22. Dantzig, G. B. The truck dispatching problem. / Dantzig, G. B., Ramser, J. H., 1959. – (6(1)).

23. Л. Ф. Гуляницький. До класифікації задач маршрутизації транспортних засобів / Л. Ф. Гуляницький, А. А. Коткова., 2020. – (вип. 36, № 1).

24. Thiago M. Evolutionary Algorithms and Metaheuristics: Bee-Inspired Algorithms Applied to Vehicle Routing Problems. / Thiago Masutti, Leandro de Castro. // Article ID 3046830. – 2017.

25. Thiago M. Evolutionary Algorithms and Metaheuristics. / Thiago Masutti, Leandro de Castro., 2017.

26. The TypeScript Handbook. – 2023. – URL: <https://www.typescriptlang.org/docs/handbook/intro.html>.

27. Remo J. TypeScript: Modern JavaScript Development / Remo Jansen, Vilic Vane, 2016. – 841 с. – (Packt Publishing).

28. Sichkarenko A. The State of Developer Ecosystem 2022 / Anastassiya Sichkarenko //– 2023. – URL: <https://blog.jetbrains.com/blog/2023/01/17/the-state-of-developer-ecosystem-2022/>.

29. Find the right tool // 2023. – URL: <https://www.jetbrains.com/products/>.

30. Singh P. What are the advantages and disadvantages of TypeScript? / Pooja Singh // TechGig. – 2023. – URL: <https://content.techgig.com/upskilling-at-techgig/what-are-the-pros-and-cons-of-typescript/articleshow/100062248.cms>.

31. Heller M. What is Visual Studio Code? / Martin Heller // InfoWorld. – 2022. – URL: <https://www.infoworld.c>.

32. Mission Planner // ArduPilot Dev Team. – 2023. – URL: <https://ardupilot.org/planner/#>.

33. ArduPilot. Use-Cases and Applications // 2023. – URL: <https://ardupilot.org/copter/docs/common-use-cases-and-applications.html>.

34. Podila P. MobX Quick Start Guide / Pavan Podila, 2018. – 326 с.

					<i>IC92.050БАК.004 ПЗ</i>	Арк.
						100
Зм.	Лист	№ докум.	Підпис	Дата		

35. Banks A. Learning React. Modern Patterns. / Alex Banks, Eve Porcello. – Gravenstein Highway North, Sebastopol: O’Reilly Media, 2020. – 310 с. – (Second edition).

36. Gorodetski M. Hyperparameter Tuning Methods. / Maria Gorodetski // Towards Data Science. – 2021. – URL: <https://towardsdatascience.com/bayesian-optimization-for-hyperparameter-tuning-how-and-why-655b0ee0b399>.

37. Hyperparameter Tuning Methods – Grid, Random, or Bayesian Search? // Towards Data Science. – 2022. – URL: https://medium.com/@ShortHills_Tech/hyperparameter-tuning-260cce6c64ff.

					<i>IC92.050БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		101