

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

*Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління*

"На правах рукопису"
УДК 519.854.2

До захисту допущено
В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

“ _____ ” _____ 20 21 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

**на здобуття ступеня магістра
за освітньо-науковою програмою
«Інформаційні управляючі системи та технології»
зі спеціальності 126 «Інформаційні системи та технології»
на тему:**

**«Оптимізація маршрутів безпілотних літальних апаратів за наявності декількох
депо»**

Виконала:

студентка VI курсу, групи ІС-91мн
Коткова Ангеліна Андріївна

Керівник:

професор, д.т.н., с.н.с.,
Гуляницький Леонід Федорович

Консультант:

доцент, к.т.н., доцент,
Жданова Олена Григорівна

Рецензент:

д-р. фіз.-мат. наук, с.н.с.,
Стецюк Петро Іванович

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

Студентка _____

Київ – 2021 року

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

*Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління*

Рівень вищої освіти – *другий (магістерський)*

Спеціальність – *126 «Інформаційні системи та технології»*

Освітньо-наукова програма *«Інформаційні управляючі системи та технології»*

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

«__» _____ 2021 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Котковій Ангеліні Андріївні

1. Тема дисертації «Оптимізація маршрутів безпілотних літальних апаратів за наявності декількох депо», науковий керівник дисертації Гуляницький Леонід Федорович, проф., д.т.н., с.н.с., затверджені наказом по університету від «12» березня 2021 р. № 809-с

2. Строк подання студентом дисертації “ 10 ” травня 2021 р.

3. Об’єкт дослідження – процес побудови маршрутів за допомогою безпілотних літальних апаратів.

4. Перелік завдань, які потрібно розробити:

- виконати аналіз сучасного стану задач маршрутизації транспортних засобів та задач маршрутизації для безпілотних літальних апаратів;
- здійснити огляд наявних методів розв’язування задач маршрутизації транспортних засобів за наявності декількох депо;
- розробити алгоритми розв’язування задачі маршрутизації безпілотних літальних апаратів за наявності декількох депо (детермінований локальний пошук, пошук із заборонами, алгоритм імітації відпалу, алгоритм прискореного ймовірнісного моделювання);
- розробити програмне забезпечення;

– виконати дослідження ефективності розроблених алгоритмів.

5. Орієнтовний перелік графічного (ілюстративного) матеріалу

Схема структурна діяльності, Блок-схема методу найдешевшого включення, Блок-схема гібридного алгоритму найдешевшого включення з проміжними покращеннями, Схема структурна розгортання, Схема структурна послідовності, Діаграма класів.

6. Орієнтовний перелік публікацій

2 наукові роботи: 1 стаття в журналі категорії Б та 1 тези.

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 01 ” лютого 20 21 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	<i>Огляд наявної літератури</i>	<i>01.02.2021 – 14.02.2021</i>	
2	<i>Здійснення класифікації задач маршрутизації транспортних засобів та окремо задач маршрутизації транспортних засобів для безпілотних літальних апаратів</i>	<i>15.02.2021 – 21.02.2021</i>	
3	<i>Огляд наявних методів розв’язування задач маршрутизації транспортних засобів за наявності декількох депо</i>	<i>22.02.2021 – 28.02.2021</i>	
4	<i>Формулювання змістовної та математичної моделі задачі</i>	<i>01.03.2021 – 07.03.2021</i>	
5	<i>Розробка алгоритмів маршрутизації безпілотних літальних апаратів</i>	<i>08.03.2021 – 28.03.2021</i>	
7	<i>Розробка програмного забезпечення</i>	<i>29.03.2021 – 11.04.2021</i>	
8	<i>Проведення експериментальних досліджень для розроблених алгоритмів</i>	<i>12.04.2021 – 16.04.2021</i>	
9	<i>Оформлення документації</i>	<i>01.02.2021 – 20.04.2021</i>	
10	<i>Подання роботи на попередній захист</i>	<i>20.04.2021</i>	
11	<i>Подання роботи на основний захист</i>	<i>10.05.2021</i>	

Студент

Ангеліна КОТКОВА

Науковий керівник

Леонід ГУЛЯНИЦЬКИЙ

РЕФЕРАТ

Магістерська дисертація: 110 с., 33 рис., 44 табл., 55 джерел, 1 додаток.

Актуальність. У сучасному світі задачі маршрутизації транспортних засобів відіграють ключову роль у логістиці та інших застосуваннях. Це пов'язано з постійним збільшенням продукції на ринку та потребою в доставці великого обсягу товарів за малий час та з мінімальними витратами. Якщо раніше в якості транспортних засобів розглядали наземний транспорт, то нині стрімко набувають популярності безпілотні літальні апарати. Уже зараз їх почали розглядати не лише як засоби доставки товарів, а і як транспорт, який може дістатися до складнодоступних місць. Безпілотні літальні апарати значно скорочують витрати коштів та часу, що необхідні для обстеження та/чи обслуговування об'єктів, оскільки є дешевшими в експлуатації за традиційні транспортні засоби та можуть швидко дістатися до необхідних об'єктів.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» в рамках теми Інституту кібернетики ім. В. М. Глушкова НАН України (2017–2021 рр.): ВФ.180.11 «Розробити математичний апарат, орієнтований на створення інтелектуальних інформаційних технологій розв'язування проблем комбінаторної оптимізації та інформаційної безпеки» (№ 0117U000323).

Мета роботи – мінімізація часу обстеження або вартості маршрутів внаслідок обстеження та/чи обслуговування заданої множини цілей на місцевості безпілотними літальними апаратами, що можуть розміщуватися в декількох депо, за наявності певних додаткових обмежень.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати аналіз сучасного стану задач маршрутизації транспортних засобів та задач маршрутизації для безпілотних літальних апаратів;
- здійснити огляд наявних методів розв'язування задач маршрутизації

транспортних засобів за наявності декількох депо;

- розробити алгоритми розв’язування задачі маршрутизації безпілотних літальних апаратів за наявності декількох депо (детермінований локальний пошук, пошук із заборонами, алгоритм імітації відпалу, алгоритм прискореного ймовірнісного моделювання);
- розробити програмне забезпечення;
- виконати дослідження ефективності розроблених алгоритмів.

Об’єктом дослідження роботи є процес побудови маршрутів за допомогою безпілотних літальних апаратів.

Предметом дослідження є методи побудови маршрутів за наявності декількох депо.

Методи дослідження – емпіричні та теоретичні.

Наукова новизна – класифікація задач маршрутизації транспортних засобів, розробка нових алгоритмів маршрутизації безпілотних літальних апаратів за наявності декількох депо.

Прикладна значущість. Методи та програмне забезпечення можуть бути використані для розв’язування задач маршрутизації безпілотних літальних апаратів за наявності декількох депо в таких галузях, як пошукові та порятункові роботи, сільське господарство, моніторинг навколишнього середовища, воєнна справа, картографія, логістика.

Публікації. Матеріали роботи опубліковані в фаховому журналі категорії Б «Науковий вісник Ужгородського університету. Серія математика й інформатика» [1] та на VI Всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021, м. Київ).

ЗАДАЧА МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ, БЕЗПІЛОТНІ ЛІТАЛЬНІ АПАРАТИ, ДЕТЕРМІНОВАНИЙ ЛОКАЛЬНИЙ ПОШУК, ПОШУК ІЗ ЗАБОРОНАМИ, ТАБУ-ПОШУК, АЛГОРИТМ ІМІТАЦІЇ ВІДПАЛУ, G-АЛГОРИТМ

ABSTRACT

Master's dissertation: 110 pages, 33 figures, 44 tables, 55 sources, 1 appendix.

Topicality. In today's world, vehicle routing tasks play a key role in logistics and other applications. This is due to the constant increase in production on the market and the need to deliver a large volume of goods in a short time and with minimal costs. If previously ground vehicles were considered as vehicles, now unmanned aerial vehicles are rapidly gaining popularity. They have already begun to be seen not only as a means of delivering goods, but also as a means of transport that can reach hard-to-reach places. Unmanned aerial vehicles significantly reduce the cost and time required to inspect and / or maintain facilities, as they are cheaper to operate than traditional vehicles and can quickly reach the required facilities.

Connection of work with scientific programs, plans, themes. The work was performed at the Department of Automated Information Processing and Control Systems of the National Technical University of Ukraine “Kyiv Polytechnic Institute named after Igor Sikorsky” within the theme of the Institute of Cybernetics named after Victor Glushkov NAS of Ukraine (2017–2021): VF.180.11 “To develop a mathematical apparatus focused on the creation of intelligent information technologies for solving problems of combinatorial optimization and information security” (№ 0117U000323).

The purpose of the work is to minimize the time of inspection or the cost of routes due to the inspection and / or maintenance of a given set of targets in the field by unmanned aerial vehicles that can be located in several depots, subject to certain additional restrictions.

To achieve this purposes you must perform the following **tasks**:

- perform an analysis of the current state of vehicle routing tasks and routing tasks for unmanned aerial vehicles;
- to review the existing methods of solving vehicle routing problems in the presence of several depots;
- to develop algorithms for solving the problem of routing unmanned aerial

vehicles in the presence of several depots (deterministic local search, search with prohibitions, annealing simulation algorithm, algorithm of accelerated probabilistic modeling);

- develop software;
- perform research on the effectiveness of the developed algorithms.

The object of the study is the process of building routes using unmanned aerial vehicles.

The subject of research is methods of construction of routes in the presence of several depots.

Research methods – empirical and theoretical.

Scientific novelty – classification of routing tasks of vehicles, development of new algorithms for routing unmanned aerial vehicles in the presence of several depots.

Applied significance. Methods and software can be used to solve the problems of routing unmanned aerial vehicles in the presence of several depots in areas such as search and rescue, agriculture, environmental monitoring, military affairs, cartography, logistics.

Publications. Materials of the work were published in the professional journal category B “Scientific Bulletin of Uzhgorod University. Mathematics and Informatics Series” [1] and at the VI All-Ukrainian Scientific and Practical Conference of Young Scientists and Students “Information Systems and Management Technologies” (ISTU-2021, Kyiv).

VEHICLE ROUTING PROBLEM, UNMANNED AIRCRAFT, DETERMINED
LOCAL SEARCH, SEARCH WITH PROHIBITIONS, TABU SEARCH, IGNITION
SIMULATION ALGORITHM, G-ALGORITHM

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	11
ВСТУП.....	12
1 ОГЛЯД ЗАДАЧ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ ТА МЕТОДІВ ЇХ РОЗВ’ЯЗУВАННЯ	14
1.1 Класифікація задач маршрутизації транспортних засобів.....	14
1.1.2 Задача маршрутизації в загальному вигляді	14
1.1.3 Найбільш поширені класи задач маршрутизації транспортних засобів	15
1.2 Класифікація задач маршрутизації транспортних засобів для безпілотних літальних апаратів	21
1.3 Класифікація методів розв’язування задач комбінаторної оптимізації	26
1.3.1 Обчислювальна складність задач комбінаторної оптимізації	26
1.3.2 Класи алгоритмів комбінаторної оптимізації за точністю	26
1.3.3 Класи алгоритмів комбінаторної оптимізації за типом обчислювальної схеми	29
Висновки до розділу	30
2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ МАРШРУТИЗАЦІЇ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ.....	31
2.1 Змістовна постановка задачі	31
2.2 Критерії оцінювання розв’язку	31
2.3 Математична модель.....	32
2.4 Вхідні дані задачі	35
2.5 Вихідні дані задачі	37
Висновки до розділу	39
3 МЕТОДИ МАРШРУТИЗАЦІЇ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ.....	40
3.1 Обґрунтування методів розв’язування.....	40
3.2 Генерація початкового наближення	41
3.2.1 Метод найдешевшого включення	41
3.2.2 Гібридний алгоритм найдешевшого включення з проміжними покращеннями	42
3.3 Покращення початкового наближення	43
3.3.1 Детермінований локальний пошук.....	44

3.3.2 Пошук із заборонами	45
3.3.3 Алгоритм імітації відпалу	46
3.3.4 Алгоритм прискореного ймовірнісного моделювання	48
Висновки до розділу	50
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	51
4.1 Засоби розробки	51
4.2 Архітектура програмного забезпечення	52
4.2.1 Схема структурна розгортання	53
4.2.2 Схема структурна послідовності	53
4.2.3 Схема структурна діяльності	53
4.2.4 Діаграма класів	54
4.2.5 Специфікація функцій	55
4.3 Настанова з використання	57
Висновки до розділу	62
5 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ	63
5.1 Умови проведення досліджень	63
5.2 Дослідження впливу параметрів на роботу алгоритмів	64
5.2.1 Параметри пошуку із заборонами	64
5.2.2 Параметри алгоритму імітації відпалу	64
5.2.3 Параметри алгоритму прискореного ймовірнісного моделювання	65
5.2.4 Результати налаштування параметрів алгоритмів	66
5.3 Порівняння результатів роботи алгоритму	67
5.3.1 Дослідження кінцевих результатів	67
5.3.2 Дослідження швидкодії	74
Висновки до розділу	81
6 РОЗРОБКА СТАРТАП-ПРОЄКТУ	83
6.1 Опис ідеї проєкту	83
6.2 Технологічний аудит ідеї проєкту	87
6.3 Аналіз ринкових можливостей запуску стартап-проєкту	87
6.4 Розроблення ринкової стратегії проєкту	93
6.5 Розроблення маркетингової програми стартап-проєкту	94

Висновки до розділу	96
ВИСНОВКИ.....	97
ПЕРЕЛІК ПОСИЛАНЬ	98
ДОДАТОК А.....	104
Плакат 1 Схема структурна діяльності	105
Плакат 2 Блок-схема методу найдешевшого включення	106
Плакат 3 Блок-схема гібридного алгоритму найдешевшого включення з проміжними покращеннями	107
Плакат 4 Схема структурна розгортання	108
Плакат 5 Схема структурна послідовності	109
Плакат 6 Діаграма класів	110

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АІВ – алгоритм імітації відпалу

АКО – алгоритм комбінаторної оптимізації

БПЛА – безпілотний літальний апарат

ДЛП – детермінований локальний пошук

ЗКО – задача комбінаторної оптимізації

ЗМТЗ – задача маршрутизації транспортних засобів

ТЗ – транспортний засіб

ТП – табу-пошук, пошук із заборонами

MDVRP – Multiple Depot Vehicle Routing Problem, з англійської задача маршрутизації транспортних засобів з декількома депо

VRP – Vehicle Routing Problem, з англійської задача маршрутизації транспортних засобів

ВСТУП

Задача маршрутизації транспортних засобів (ЗМТЗ) є важливою та актуальною, оскільки має широке застосування на практиці. Уперше про ЗМТЗ заговорили George Dantzig та John Ramseur у 1959 році [2]. Вони вручну провели аналітичні розрахунки для оптимізації вантажних перевезень, використовуючи метод, заснований на лінійному програмуванні. У результаті отримано близький до оптимального розв’язок з чотирма маршрутами та дванадцятьма споживачами. З того часу обсяг досліджень у цьому напрямку значно збільшився. Так, пошук у Google Academy за запитом «Vehicle Routing Problem» за всі роки налічує близько 522 000 результатів, зокрема у 2017 році – 18 900 результатів, у 2018 році – 20 600 результатів, у 2019 році – 22 400 результатів, у 2020 році – 19 200 результатів, у 2021 році – маємо за 4 місяці 3 850 результатів. Як бачимо, ЗМТЗ нині користуються великою популярністю серед науковців і практиків. Це пов’язано з тим, що в деяких випадках, особливо при плануванні великої кількості перевезень, навіть незначне покращення маршрутів транспортних засобів (ТЗ) може дати значну економію ресурсів.

Якщо раніше в якості ТЗ розглядали наземний транспорт, то згодом стрімко почали набувати популярності безпілотні літальні апарати (БПЛА) – це такі апарати, що літають та сідають без фізичної присутності пілота на борту. У 1899 році Nikola Tesla сконструював та продемонстрував радіокероване мінісудно. У 1910 році Charles Kettering, американський військовий інженер, запропонував використовувати літальні апарати без безпосередньої участі людини в їх керуванні. Вони були призначені для ведення бойових дій: у заданій точці БПЛА мав відкидати крила та падати на ворога, як бомба. Далі активний розвиток БПЛА відбувався під час Другої світової війни. Їх використовували як літаки-мішені, керовані бомби, радіокеровані літаки з вибуховими речовинами, торпедоносці тощо. Згодом БПЛА почали переходити з військової сфери до цивільної. Їх почали застосовувати в мирних цілях. БПЛА стали користуватися попитом як у промислових масштабах для послуг доставки та

перенесення вантажів, так і в інших сферах: обороні, пошуку, порятунку, сільському господарстві, нагляді за навколишнім середовищем, картографії тощо. Наприклад, після недавнього вдосконалення технології БПЛА Amazon, DHL, Federal Express та інші великі компанії, що зацікавлені в доставці пакетів, почали досліджувати доцільність включення доставки за допомогою БПЛА у свої комерційні послуги [3]. Залежно від принципів керування розрізняють некеровані, автоматичні та дистанційно-пілотовані БПЛА. Після 2000 року в авіації стрімко зростає використання останнього типу БПЛА. Говорячи про них, вживають терміни «безпілотник», «дрон».

Основними перевагами БПЛА є: точність, значно менші витрати на обслуговування, оперативність, актуальність отриманих даних, широкий спектр застосування. БПЛА можуть за декілька днів розв'язати певні специфічні задачі, над якими люди працювали б тижнями. Також однією з переваг є безпека використання. Наприклад, деякі типи робіт у шахтах є небезпечними та становлять загрозу для життя працівників, а з БПЛА можна звести ці ризики до мінімуму.

Враховуючи, що нині відбувається активний розвиток галузі, пов'язаної з БПЛА, перед науковцями постала нова проблема – пошук оптимальних розв'язків ЗМТЗ із застосуванням БПЛА.

1 ОГЛЯД ЗАДАЧ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ ТА МЕТОДІВ ЇХ РОЗВ'ЯЗУВАННЯ

1.1 Класифікація задач маршрутизації транспортних засобів

ЗМТЗ (Vehicle Routing Problem, VRP) – це задачі знаходження оптимальних маршрутів перевезення вантажів із пунктів відправлення до пунктів призначення, обстеження об'єктів тощо. Метою таких задач є знаходження множини маршрутів для обслуговування клієнтів певною кількістю ТЗ у заданому середовищі.

1.1.2 Задача маршрутизації в загальному вигляді

Класична ЗМТЗ відноситься до класу комбінаторних задач [4]. Її можна подати у вигляді графа $G(V, E)$,

де $V = \{v_0, v_1, \dots, v_n\}$ – множина вершин (v_0 – депо, $v_1, \dots, v_n$ – точки доставки, клієнти);

E – множина ребер $\{(v_i, v_j) | i \neq j\}$, що відповідають шляхам;

C – матриця ваг ребер (вартості шляхів) c_{ij} між вершинами;

m – кількість ТЗ, задіяних у доставці товару;

R_i – маршрут i -го ТЗ ($i = \overline{1, m}$);

$C(R_i)$ – вартість маршруту R_i ;

q_i – обсяг товару, що має постачатися в i -ту точку доставки.

Кожній вершині v_i ставиться у відповідність деяка кількість товарів, які мають бути доставлені клієнтам у відповідну точку доставки. Задача маршрутизації полягає у визначенні такої множини маршрутів мінімальної вартості (відповідно до величини m – кількості ТЗ), щоб кожна вершина відвідана ТЗ тільки один раз. Усі маршрути повинні починатися й закінчуватися в депо v_0 .

Розв'язком задачі є розбиття множини V на підмножини (маршрути) й задавання порядку обходу вершин на кожній підмножині (перестановка вершин маршруту). Розв'язок задачі є допустимим, якщо всі маршрути задовольняють додатковим обмеженням задачі. Цільовою функцією (ЦФ) є сумарна вартість перевезень:

$$F_{VRP} = \sum_{i=1}^m C(R_i), \quad (1.1)$$

де $C(R_i)$ – сума вартостей шляхів (вага ребер) маршруту R_i .

У класичному варіанті ЗМТЗ потрібно знайти допустимий розв'язок з мінімальною такою вартістю. Зазвичай у реальних ЗМТЗ виникає низка додаткових обмежень, які породжують цілий спектр подібних задач.

1.1.3 Найбільш поширені класи задач маршрутизації транспортних засобів

Однією з найпростіших ЗМТЗ є задача комівояжера (Travelling Salesman Problem, TSP). Вона полягає в знаходженні найкоротшого Гамільтонового циклу на графі. Тобто агент повинен пройти через усі вершини графу по одному разу та повернутися в початкову, пройшовши мінімальну відстань або затративши мінімальну кількість часу.

Існує багато різновидів ЗМТЗ: з обмеженням по вантажопідйомності, «часовими вікнами», альтернативними динамічними депо, поверненням і доставкою товарів, різним транспортом, періодичною маршрутизацією тощо. У літературі існує багато різних класифікацій класів ЗМТЗ. На рисунку 1.1 наведено одну із таких можливих ієрархій.

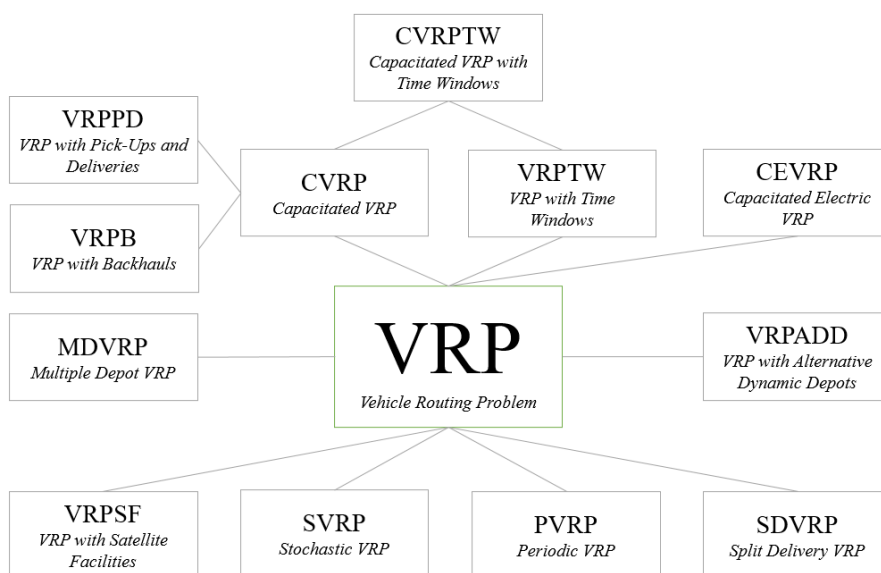


Рисунок 1.1 – Ієрархія основних класів ЗМТЗ

Розглянемо більш детально кожен клас задач, указаний на рисунку 1.1.

ЗМТЗ з обмеженнями на вантажопідйомність (Capacitated VRP, CVRP) [5]. Додатковим обмеженням цієї задачі є те, що маса товарів на кожному маршруті не повинна перевищувати заданої величини обсягу товару, який має постачатися в певну точку. Ціллю такої задачі є мінімізація загальної вартості перевезень. Задачу маршрутизації з обмеженнями на вантажопідйомність можна розширити додатковими обмеженнями та отримати принципово нові класи задач.

ЗМТЗ з поверненням і доставкою товарів (VRP with Pick-Ups and Deliveries, VRPPD) [6, 7]. Відмінність цієї задачі від класичної в тому, що клієнти в точках доставки можуть повертати товари, які ТЗ має привезти назад у депо. Умовою повернення товару є те, що він не повинен перевантажувати ТЗ, тобто необхідно враховувати вантажопідйомність. Такі ЗМТЗ є складними в розв'язанні, адже викликають ряд проблем: імовірність неефективного використання транспорту, збільшення вартості маршрутів та загальної кількості ТЗ у депо тощо. Для простоти зазвичай розглядають задачі з додатковими обмеженнями. Наприклад, доставка товарів відбувається тільки до точок доставки, а повернення – тільки в депо, тобто немає обміну товарами між точками доставки. Інший спосіб спрощення полягає у відміні обмеження, що всі точки доставки можуть бути відвідані лише один раз. Існує також ще один спосіб: припустити, що ТЗ спочатку розвозить усі товари, а лише потім починає забирати товар з точок доставки (VRP with Backhauls, VRPB, що описана нижче). Ціль задачі: мінімізувати вартість перевезень і розмір парку ТЗ. Обмеження: кількість товару, який потрібно доставити до клієнтів, і товару, який потрібно забрати від клієнтів, не повинна перевищувати вантажопідйомність ТЗ у жодній точці маршруту. Однією зі складних задач VRPPD є ЗМТЗ з гетерогенним транспортом, змішаним поверненням та часовими вікнами (Heterogeneous VRP with Mixed Backhauls and Time Windows, HVRPMBTW) [8]. Ця задача характеризується обмеженою кількістю ТЗ, різною місткістю та витратами.

ЗМТЗ зі зворотними перевезеннями (VRP with Backhauls, VRPB) [9, 10]. У попередній задачі VRPPD потрібно враховувати, що всі товари, які повернуть клієнти, повинні вміститися до ТЗ. Але задача VRPB відрізняється тим, що спершу всі товари мають бути доставлені, а вже потім клієнти зможуть робити повернення. Це пов'язано з тим, що більшість ТЗ зазвичай завантажуються ззаду або зверху, утворюючи стек. Тому перестановка вантажу під час кожної доставки є економічно не вигідною.

ЗМТЗ з «часовими вікнами» (VRP with Time Windows, VRPTW) [11]. Основним обмеженням для цієї задачі є те, що ТЗ повинен доставити товар у конкретний проміжок часу. У таких задачах є верхня та нижня границі часу. Так, якщо товар доставлено в точку після верхньої часової границі, то його не можуть прийняти або накладають на його прийняття штрафні санкції. Якщо ТЗ привіз товар раніше нижньої часової границі, то він має очікувати. Ціллю задачі є мінімізація загальної вартості перевезень, розміру парку ТЗ та загального часу очікування. Такі задачі допомагають визначити оптимальний час виїзду ТЗ із депо та уникнути їх простоїв. Подальший розвиток для задачі VRPTW представлено розв'язуванням ЗМТЗ за допомогою зворотних передач та часових вікон (VRP with Backhauls and Time Windows, VRPBW). Ця задача включає в себе вантажопідйомність, повернення товарів, часові вікна та розв'язується за допомогою гібридного метаевристичного алгоритму. Метою є мінімізація загальної відстані перевезень [12]. Ще однією задачею є VRP з м'якими часовими вікнами для декількох типів ТЗ з контролем забруднення навколишнього середовища від перевезень. Метою є дослідження взаємозв'язку між вартістю доставок, задоволеністю споживачів та забрудненістю навколишнього середовища. Для розв'язування такої задачі розроблено гібридний генетичний алгоритм, результати якого показують, що швидкість ТЗ має сильну кореляцію з експлуатаційними витратами та забрудненням навколишнього середовища, тоді як вантажопідйомність впливає на експлуатаційні витрати, задоволеність споживачів та забруднення навколишнього середовища [13].

ЗМТЗ (Capacitated VRP with Time Windows, CVRPTW). Якщо об'єднати між собою ЗМТЗ з обмеженнями на вантажопідйомність та з «часовими вікнами», то отримаємо цей клас задач. У них беруть до уваги як вантажопідйомність ТЗ, так і часові межі, у яких вони повинні доставляти товар.

ЗМТЗ з декількома депо (Multiple Depot VRP, MDVRP) [14, 15, 16, 17, 18, 19, 20, 21, 22]. Ця задача потребує найбільш вигідного розподілення клієнтів по різних депо та визначення оптимального маршруту, за якого вони всі будуть відвідані. У кожному депо є свій парк ТЗ, де кожен ТЗ обслуговує клієнтів, а потім повертається назад. Ціллю такої задачі є мінімізація вартості перевезень та розміру парку ТЗ. У деяких випадках може бути кілька депо, які обслуговують клієнтів. Якщо клієнти зосереджені біля кожного депо, то задача може бути розбита на кілька незалежних задач. Проте, якщо клієнти й депо розташовані довільним чином, то потрібно шукати розв'язок ЗМТЗ з декількома депо.

ЗМТЗ з можливістю дозавантаження (VRP with Satellite Facilities, VRPSF) [23]. Для класичної ЗМТЗ є характерним те, що при проходженні маршруту ТЗ повинен почати й закінчити маршрут у депо. Це пов'язано з обмеженою вантажопідйомністю: ТЗ повинен дозавантажитися, коли доставить усі товари. Для деяких задач є вигідним дозавантаження ТЗ без повернення в депо, а безпосередньо на маршруті. Це можливо реалізувати за допомогою додаткових ТЗ. Типовим є випадок, коли велика кількість клієнтів очікує регулярних поставок від одного центрального поставника. Метою задачі є мінімізація витрат на перевезення у визначений час. Можливим недоліком такої задачі є більша загальна вартість розв'язання задачі в короткостроковій перспективі, ніж для класичної VRP, оскільки необхідні додаткові витрати на додаткові ТЗ.

ЗМТЗ з випадковими даними (Stochastic VRP, SVRP) [24, 25]. Для цього класу ЗМТЗ характерною є випадкова поведінка для деяких компонентів задачі. Випадковість проявляється так: випадкові клієнти — кожна точка доставки є

ймовірнісною величиною; випадкові запити – кількість товарів, які необхідно доставити клієнту, та відстань між точками доставки є випадковими величинами. Розв’язування таких задач відбувається у два етапи. На першому етапі отримується розв’язок без урахування додаткових змінних. На другому етапі відбувається корекція отриманого раніше розв’язку на основі випадкових значень. Метою задачі є мінімізація загальної вартості перевезень. Якщо деякі точки доставки невідомі, необхідно накладати додаткові обмеження: деякі умови будуть виконуватися із заданою імовірністю або потрібно будувати модель корекції, що виконуватиметься при порушенні деяких обмежень. Наприклад, у задачі SVRP з поверненням товарів і обмеженням по вантажопідйомності ТЗ способами корекції будуть: повернутися в депо, коли ТЗ перевантажений, розвантажитися й повернутися на маршрут; повернутися в депо, коли ТЗ перевантажений, а потім заново оптимізувати залишок маршруту; запланувати дострокове повернення в депо, навіть якщо ТЗ завантажений не повністю. Для такої моделі корекції розв’язок задачі може залежати від кількості зібраного товару й відстані до депо. ТЗ може повернутися до депо за умови, коли збирання товару від наступного клієнта призведе до перевищення вантажопідйомності ТЗ.

ЗМТЗ з періодичною маршрутизацією (Periodic VRP, PVRP) [26]. Для цього класу задач характерним є розширення періоду планування від одного дня до кількох днів. Метою задачі є мінімізація вартості перевезень та розміру парку ТЗ. Додаткові обмеження: ТЗ може повернутися в депо не в той день, коли виїхав; за M -денний період кожна точка доставки повинна бути відвідана, як мінімум, один раз. Запити кожного клієнта повинні бути виконані за один візит одним ТЗ. Якщо період планування $M = 1$, то задача зводиться до класичної VRP. Кожна точка доставки в задачі з періодичною маршрутизацією має бути відвідана k раз, причому $1 \leq k \leq M$. У класичній PVRP щоденне замовлення клієнта є фіксованим. Тому таку задачу можна розглядати як задачу компонування групи маршрутів на кожен день за умови,

що маршрути задовольняють обмеження, а загальна вартість перевезень є мінімальною.

ЗМТЗ з різним транспортом (Split Delivery VRP, SDVRP) [27, 28]. Принциповою відмінністю цього класу задач є те, що одну точку доставки можуть обслуговувати декілька ТЗ за умови, що це зменшить загальну вартість перевезень. Типовим прикладом такої задачі є великий об'єм замовленого товару при малій вантажопідйомності ТЗ. Метою задачі є мінімізація затрат на перевезення та розміру парку ТЗ.

ЗМТЗ з альтернативними динамічними депо (VRP with Alternative Dynamic Depots, VRPADD) [29, 30]. Сферою застосування цього класу задач є ЗМТЗ, де в якості ТЗ виступають безпілотні літальні апарати чи інші рухомі роботизовані системи. Особливістю цього класу задач є те, що ТЗ можуть стартувати в рухомих депо та завершувати маршрути в рухомих депо. Такі задачі актуальні для військової та морської діяльності. При оптимізації маршрутів вантажопідйомність не грає ключову роль (наприклад, при аерофотозніманні). Суттєве значення мають обмеження на ресурси системи: обсяг пального чи ємність акумулятора;

ЗМТЗ з використанням електромобілів та обмеженнями на вантажопідйомність (Capacitated Electric VPR, CEVRP). У зв'язку з розвитком електромобілів постала проблема маршрутизації, що слідує з обмеженої кількості зарядних станцій та обмеженого запасу ходу електромобілів. У таких задачах необхідно враховувати не лише порядок обслуговування клієнтів, але й графіки підзарядки електромобілів. Розв'язати ці дві задачі одночасно досить складно. Тому її розглядають як задачу дворівневої оптимізації. На першому рівні розглядається ємнісна задача маршрутизації (CVRP), а на другому рівні – проблема зарядки транспортного засобу за вже відомим маршрутом. Для підзадачі першого рівня ігноруються обмеження на електроспоживання, а на другому рівні розробляється нова евристика, щоб задовольнити обмеження по електроенергії для вже згенерованих маршрутів.

На рисунку 1.2 показано місце досліджуваної проблеми маршрутизації серед множини класів ЗМТЗ.

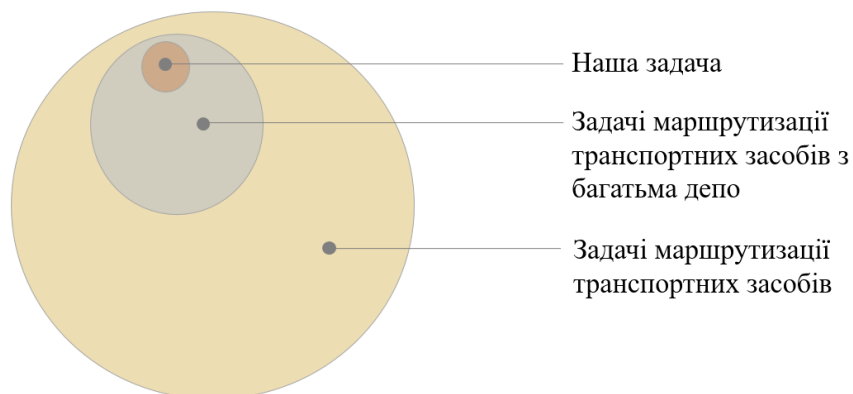


Рисунок 1.2 – Місце досліджуваної проблеми серед множини класів ЗМТЗ

Провівши аналіз близьких постановок, можна стверджувати, що досліджувана проблема входить до класу задач MDVRP, для яких є характерним наявність декількох депо, між якими розподілені ТЗ.

1.2 Класифікація задач маршрутизації транспортних засобів для безпілотних літальних апаратів

Задачі маршрутизації БПЛА переходять від нової теми до зростаючої дослідницької області. Сучасний розвиток технологій поступово прямує до того, щоб БПЛА відігравали ключову роль у міській логістиці. Уже зараз задачі маршрутизації БПЛА перетворилися на зрілу технологію, що застосовується в провідних галузях. БПЛА можуть значно скоротити фінансові витрати та час, що необхідний для доставки матеріалів, оскільки вони дешевші в обслуговуванні, ніж традиційні ТЗ, наприклад, БПЛА можуть знизити витрати на робочу силу, виконуючи завдання автономно. На рисунку 1.3 наведено класифікацію сфер застосування задач маршрутизації БПЛА. У цілому їх можна розділити на дві групи: цивільна та військова.

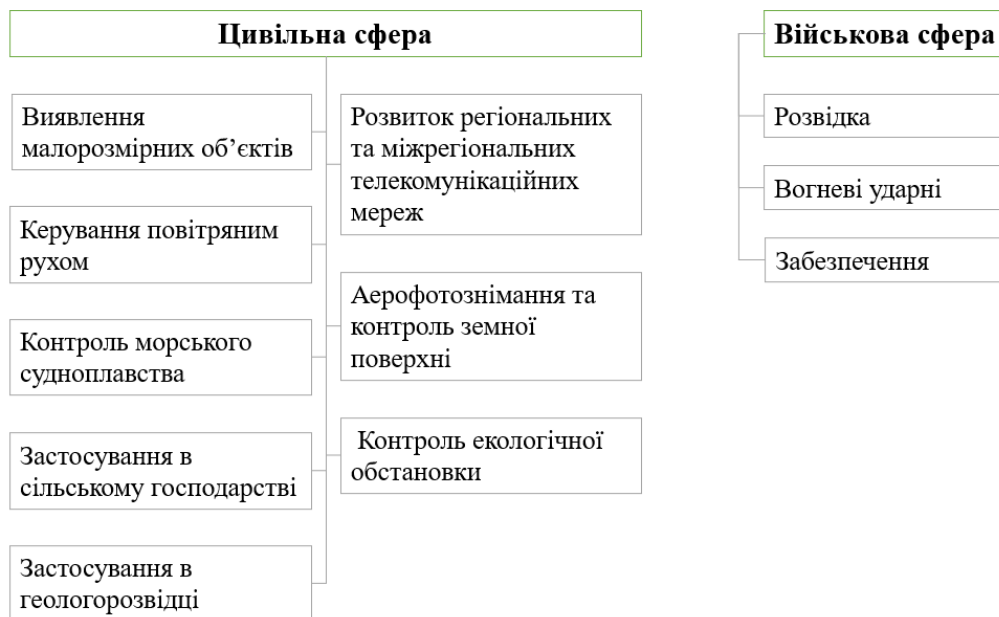


Рисунок 1.3 – Сфери застосування БПЛА

Розглянемо більш детально сфери застосування БПЛА, наведені на рисунку 1.3. До цивільної сфери входять виконання таких задач:

- виявлення малорозмірних об'єктів (повітряних, наводних, наземних, пошук і порятунк, допомога в надзвичайних ситуаціях);
- керування повітряним рухом (у важкодоступних районах, при стихійних лихах і аваріях, на тимчасових авіаційних трасах при виконанні авіаційних робіт);
- контроль морського судноплавства (пошук і виявлення суден, попередження аварійних ситуацій у портах, контроль морських кордонів, контроль правил рибальства);
- застосування в сільському господарстві (виконання авіаційно-хімічних робіт);
- застосування в геологорозвідці (визначення характеристик ґрунту, розвідка корисних копалин, підповерхневе зондування Землі);

- розвиток регіональних та міжрегіональних телекомунікаційних мереж (у системах зв'язку, телерадіомовленні, ретрансляції, навігаційних системах, рекламі, телебаченні, кіно);
- аерофотознімання та контроль земної поверхні (картографія, контроль гідро- та метеообстановки, правоохоронна діяльність, виявлення лісових пожеж, спостереження периметрів об'єкта, контроль залізничних колій);
- контроль екологічної обстановки (радіаційний контроль, газохімічний контроль, контроль стану газо- та нафтопроводів, контроль стану лавин) [31].

До військової сфери входять виконання таких задач:

- розвідка (ведення повітряної розвідки з малих і середніх висот в умовах сильної протидії природних умов та складної радіоелектронної обстановки з можливістю передачі отриманої розвідувальної інформації із захищених радіоканалів у реальному масштабі часу; виявлення та ідентифікація наземних, наводних та повітряних цілей, угруповань військ та техніки, вогневих точок та фортифікаційних споруд; виявлення активно випромінюючих об'єктів);
- вогневі ударні (ураження техніки та живої сили противника шляхом нанесення ударів по наземних та морських цілях; боротьба з повітряними цілями та знищення боеголовок балістичних ракет; дистанційне встановлення мінних полів, мінування територій);
- забезпечення (перевезення вантажів у різних умовах бойової обстановки, у тому числі під час активного ведення бойових дій та доставка в райони, заражені засобами радіоактивного, хімічного та біологічного впливу; управління наземними, повітряними, морськими вогняними засобами; оцінка результатів удару, завданого противнику; ретрансляція повідомлень і даних; пошук та евакуація поранених із поля бою) [31].

Ієрархію основних класів ЗМТЗ можна розширити, додавши до неї задачі маршрутизації з використанням БПЛА.

Задача комівояжера з використанням БПЛА (TSP with Unmanned Aerial Vehicles, TSP-UAV) [32, 33]. Для цього класу задач характерні наявність одного депо, одного БПЛА та показників витрат. БПЛА має відвідати кожну ціль лише один раз і повернутися назад до депо. Метою задачі є мінімізація загальної вартості маршруту. Критеріями завершення можуть бути, наприклад, пройдена відстань або час виконання завдання.

Задача комівояжера з використанням декількох БПЛА (TSP with Multi Unmanned Aerial Vehicles, TSP-MUAV) [34, 35]. Ця задача є узагальненням задачі TSP-UAV. Її принципова відмінність полягає в тому, що цілі може відвідувати не один БПЛА, а декілька. При чому кожна ціль має бути відвідана лише одним БПЛА. Метою такої задачі є визначення набору маршрутів таким чином, щоб їх вартість була мінімальною.

Задача комівояжера з використанням БПЛА та вартості маршруту (TSP with Unmanned Aerial Vehicles and the Cost of the Route, TSP-UAVCR). Ця задача також є узагальненням задачі TSP-UAV. Основною її відмінністю є те, що для кожного маршруту може бути задана різна вартість у залежності від напрямку. Метою задачі також є мінімізація загальної вартості маршруту.

Класична ЗМТЗ з використанням БПЛА (VRP with Unmanned Aerial Vehicles, VRP-UAV) [36, 37, 38, 39, 40, 41, 42]. Необхідно визначити набір маршрутів для парку дронів, які будуть відвідувати цілі. Як і для класичної ЗМТЗ, метою цієї задачі є мінімізація загальної вартості маршрутів.

ЗМТЗ з використанням БПЛА та з «часовими вікнами» (VRP with Unmanned Aerial Vehicles and Time Windows, VRPTW-UAV) [43, 44, 45, 46]. Ця задача є узагальненням задачі VRP-UAV. Принциповою відмінністю цієї задачі є те, що обслуговування цілей відбувається в заданий час, що зветься «часовим вікном». Таке «часове вікно» задається окремо для кожного об'єкта. Якщо БПЛА прилетів до цілі

раніше, він повинен почекати, а якщо пізніше, то має сплатити штраф. Метою задачі також є мінімізація загальної вартості маршрутів.

ЗМТЗ з використанням БПЛА та можливістю дозавантаження (VRP with Unmanned Aerial Vehicles and Satellite Facilities, VRPSF-UAV) [47]. Особливістю цієї задачі є те, що за рахунок дозавантаження можна зменшити використання БПЛА. Метою також є мінімізація вартості маршрутів.

ЗМТЗ з використанням БПЛА та обмеженнями на вантажопідйомність (Capacitated VRP with Unmanned Aerial Vehicles, CVRP-UAV) [48]. Особливістю цієї задачі є обмежена вантажопідйомність БПЛА. При цьому кожен дрон може облетіти лише ту відстань, на яку вистачає його потужності без можливості дозаправлення. Метою задачі також є мінімізація вартості маршрутів.

ЗМТЗ з використанням БПЛА, обмеженнями на вантажопідйомність та можливістю дозаправлення (Capacitated VRP with Unmanned Aerial Vehicles and the Possibility of Refueling, CVRPPR-UAV). Ця задача є розширенням задачі CVRP-UAV, але з можливістю повернення в депо для дозаправки. Таким чином БПЛА може здійснювати багаторазові польоти.

ЗМТЗ з використанням БПЛА та багатьма депо (VRP with Unmanned Aerial Vehicles and Multi Depo, MDVRP-UAV). Ця задача дуже схожа на звичайну MDVRP, за винятком того, що тут у якості ТЗ використовуються БПЛА. Для цієї задачі характерними є наявність декількох депо, декількох цілей та декількох БПЛА. Кожен БПЛА може відвідати лише одну ціль, після чого він має повернутися в те саме депо, звідки вилетів. Метою задачі є знаходження маршруту, за якого вартість обслуговування цілей буде мінімальною.

Як бачимо із наведеної класифікації, класи ЗМТЗ з використанням БПЛА схожі з класами ЗМТЗ, де БПЛА не задіяні, але все ж таки мають певні відмінності.

1.3 Класифікація методів розв'язування задач комбінаторної оптимізації

Для простоти викладу вважаємо, що задача комбінаторної оптимізації (ЗКО) не має додаткових обмежень, тобто $D \equiv X$. Необхідно знайти $x_* = X$ такий, що

$$x_* = \arg \min_{x \in X} f(x). \quad (1.2)$$

Вважаємо, що алгоритм комбінаторної оптимізації (АКО) – це певна процедура, що переводить задану підмножину $Z \subset X$ (початкові наближення) у множину $X_* \subset X$:

$$AZ = X_*, \quad (1.3)$$

де X_* – множина знайдених розв'язків задачі (1.2).

Для розв'язування ЗМТЗ застосовують чимало методів, для яких є класифікації за різними ознаками: за точністю, типом використання просторів, структурою обчислювальної схеми, отримуваним розв'язком тощо. Розглянемо дві можливі класифікації основних методів: за точністю та за типом обчислювальної схеми.

1.3.1 Обчислювальна складність задач комбінаторної оптимізації

Розрізняють два класи проблем: P та NP.

Задачі класу P – це такі задачі, які можна вирішити за поліноміальний час, а задачі класу NP – такі, для яких час вирішення не обмежений поліномом та точний розв'язок може бути знайдений лише перебором усіх варіантів.

Для задач класу NP не доцільно застосовувати точні алгоритми, оскільки їх обчислювальна складність є занадто великою. Для них використовуються наближені методи, як от: евристики, метаевристики тощо.

1.3.2 Класи алгоритмів комбінаторної оптимізації за точністю

На рисунку 1.4 наведено основні методи розв'язування ЗМТЗ, класифіковані за точністю.



Рисунок 1.4 – Класифікація АКО за точністю

Точні АКО знаходять глобальний розв’язок, тобто для них $X_* \subset \text{Arg ext}$. Наближені АКО діляться на алгоритми з апіорною та апостеріорною оцінками точності [4]. Евристичні алгоритми побудовані на основі правдоподібних міркувань, наприклад, метод включення найближчого міста, застосовують для знаходження розв’язків задачі комівояжера. Усе ж такі методи не дають бажаної точності для отриманих результатів.

Часто на практиці дослідники об’єднують наближені алгоритми з оцінкою точності та евристичні в один клас, який називають просто «наближені алгоритми». Тоді АКО можна розділити на дві великі групи: точні та наближені алгоритми.

Точні алгоритми можна розділити на загальні методи та спеціальні алгоритми. До загальних методів належать повний перебір, метод гілок та меж, метод гілок та відтинань, послідовний аналіз варіантів та динамічне програмування. До спеціальних алгоритмів належать ті, що розробляються під конкретну задачу, а тому мають вузький спектр застосування.

Наближені алгоритми можна розділити на сім класів (рисунок 1.5).

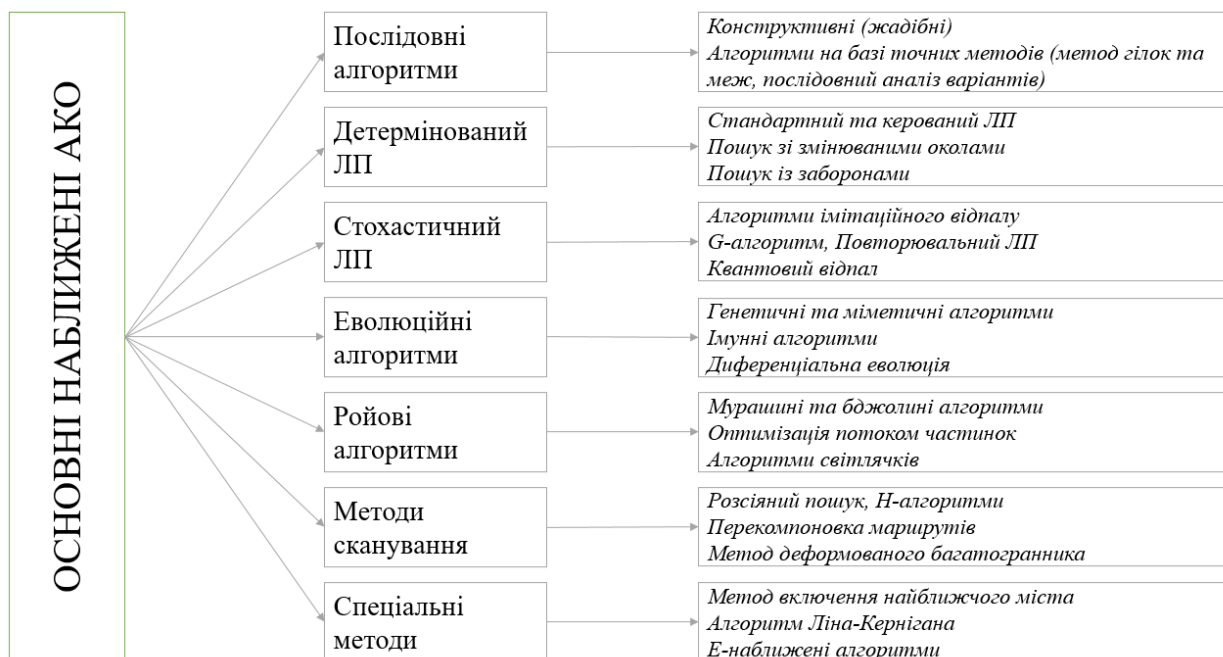


Рисунок 1.5 – Класифікація основних наближених АКО

Важливо зазначити, що в наведеній класифікації точні алгоритми, наприклад, метод гілок та меж, використовуються для породження наближених обчислювальних схем. Звісно, нині існує набагато більше АКО, ніж наведено на рисунку 1.5, однак, як свідчить аналіз наукових публікацій у цьому напрямку, саме зазначені алгоритми є найбільш уживаними при розв’язанні ЗКО.

У переважній більшості для розв’язання ЗКО застосовують саме наближені алгоритми, що пов’язано з низкою причин: практично всі ЗКО належать до класу NP-складних задач, тож їх точне розв’язання є проблематичним навіть із використанням сучасних комп’ютерів; зазвичай ЦФ ЗКО багатоекстремальні, тобто мають декілька локальних мінімумів або максимумів; у багатьох прикладних проблемах дані задаються з певними похибками; ідеї, що покладені в основу розробки наближених АКО дозволяють створювати алгоритми, які можуть розв’язувати не одну задачу, а цілий клас близьких за постановкою ЗКО.

1.3.3 Класи алгоритмів комбінаторної оптимізації за типом обчислювальної схеми

За типом обчислювальної схеми АКО поділяються на конструктивні (прямі, послідовні) та ітераційні. Нехай маємо певну множину $Y \supseteq X$.

Конструктивні алгоритми – це такі алгоритми, у яких $Y \supset X$ ($Y \neq X$), тобто вони оперують у просторі, що є розширенням простору розв’язків X . Починаючи «з нуля» чи якогось іншого фрагменту розв’язку, вони поступово формують повний розв’язок. Розглянемо приклади таких алгоритмів для різних ЗКО.

Задача комівояжера розв’язується алгоритмом включення найближчого міста, де на кожному наступному кроці до маршруту додається вершина, відстань до якої є найменшою з усіх можливих. Він отримав назву «іди в найближче» й відноситься до жадібних алгоритмів.

Для квадратичної задачі про призначення характерним є те, що об’єкти (елементи) з більш інтенсивними потоками розміщуються в першу чергу.

Для пошуку мінімального кістякового дерева застосовується евристика, відповідно до якої на кожному наступному кроці в побудований фрагмент кістякового дерева включається та вершина, яка мінімально збільшує його сумарну довжину й не утворює підцикл.

Ітераційні алгоритми – це такі алгоритми, які на кожному кроці опрацьовують «повні» розв’язки, тобто для них простір пошуку $Y \equiv X$. Починаючи з деякого $x^0 \in X$, ітераційні алгоритми намагаються його покроково поліпшувати:

$$x^{(h+1)} = A^{(h)}x^{(h)}, h = 0, 1, \dots,$$

де $A^{(h)}$ – ітераційна процедура, яка в більшості випадків не залежить від кроку h , тобто $A^{(h)} = A$.

Серед ітераційних алгоритмів виділяють два класи: траєкторні та популяційні. Траєкторними називають такі ітераційні методи, які оперують на кожному кроці одним (поточним) розв’язком. Популяційними називають такі ітераційні методи, які

на кожній ітерації опрацьовують не один, а кілька розв'язків одночасно. Тож для траєкторних алгоритмів $\|Z\| = \|X_*\| = 1$, а для популяційних $\|Z\| > 1, \|X_*\| > 1$.

Висновки до розділу

У першому розділі розглянуто ЗМТЗ і визначено, що нині існує багато їх різновидів: з обмеженням по вантажопідйомності, «часовими вікнами», альтернативними динамічними депо, поверненням і доставкою товарів, різним транспортом, періодичною маршрутизацією, з можливістю дозавантаження тощо. У результаті наведено класифікацію найбільш поширених ЗМТЗ.

Провівши аналіз близьких постановок, визначено, що досліджувана проблема входить до класу ЗМТЗ з багатьма депо, та показано місце досліджуваної проблеми серед множини класів ЗМТЗ. У ході аналізу сучасного стану ЗМТЗ виявлено, що замість звичайних транспортних засобів у багатьох випадках доцільно використовувати БПЛА. Сучасний розвиток технологій поступово прямує до того, щоб БПЛА відігравали ключову роль у різних логістичних схемах. Уже зараз задачі маршрутизації БПЛА перетворились на зрілу технологію, що застосовується в провідних галузях. Аналіз сфер застосування БПЛА показав, що в цілому їх можна розділити на дві групи: цивільна та військова. БПЛА можуть значно скоротити фінансові витрати та час, що необхідний для доставки матеріалів, оскільки вони дешевші в обслуговуванні, ніж традиційні ТЗ.

Так, як і для звичайних ЗМТЗ, для ЗМЗТ з використанням БПЛА нині існує чимало різновидів: задача Комівояжера з використанням декількох БПЛА, задача Комівояжера з використанням БПЛА та вартості маршруту тощо. Також розглянуто методи розв'язування ЗКО, проаналізовано обчислювальні складності ЗКО та виділено основні класи АКО за точністю та типом обчислювальної схеми. Так, за точністю АКО поділяють на точні, наближені (з оцінкою точності) та евристичні, а за типом обчислювальної схеми – на конструктивні та ітераційні. Серед ітераційних алгоритмів виділено ще два класи: траєкторні та популяційні.

2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ МАРШРУТИЗАЦІЇ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

2.1 Змістовна постановка задачі

У загальному розглядається проблема планування місії команди гетерогенних БПЛА. Вважаються заданими кількість об'єктів дослідження, час їх обстеження та координати на площині; кількість депо та їх координати, кількість БПЛА, швидкість кожного БПЛА, вартість одиниці пройденого шляху та польотний ресурс. Цілями є мінімізація загальної вартості місії або часу її виконання.

Проблема полягає в обстеженні та/чи обслуговуванні заданої множини об'єктів на місцевості. При цьому БПЛА не закріплені за конкретним депо – місцем базування. Метою є мінімізація затрат часу або вартості маршруту при використанні мінімальної кількості БПЛА або будь-якої кількості з доступних за наявності певних додаткових обмежень:

- кожен об'єкт має бути відвіданий лише один раз одним БПЛА;
- кожен БПЛА має обмежений польотний ресурс;
- довжина маршруту кожного БПЛА не має перевищувати його польотний ресурс;
- кожен БПЛА після завершення місії має повернутися в депо, з якого почав свій маршрут.

Якщо ресурсів БПЛА недостатньо для обстеження та/чи обслуговування всіх об'єктів, то задача розв'язку не має. У цьому разі слід збільшити кількість задіяних БПЛА або використати БПЛА зі збільшеними параметрами ресурсів.

2.2 Критерії оцінювання розв'язку

Для поставленої задачі виділено два класи: задача покращення заданого критерію при задійній кількості БПЛА та мінімізація кількості БПЛА з подальшим покращенням критерію. При цьому критерії такі: час виконання завдання та вартість маршруту. У результаті маємо чотири різні задачі.

Користувач має можливість вибрати один із двох класів задач. Якщо обрано клас із мінімізацією часу виконання чи вартості, то задача алгоритму – покращити критерій не залежно від кількості БПЛА. Якщо обрано клас із мінімізацією кількості БПЛА та подальшим покращенням критерію, то першочерговим завданням алгоритму є знаходження мінімальної кількості БПЛА, а вже потім необхідно покращити критерій. Перший клас виникає за наявності парку БПЛА, тоді потрібно лише знайти оптимальні маршрути за часом або вартістю. Другий клас задач корисний клієнтам, яким необхідно здійснити закупку БПЛА / збільшити парк БПЛА перед виконанням місії, а тому їм актуально дізнатися мінімальну кількість БПЛА для побудови оптимальних маршрутів.

Розглянемо критерій мінімізації часу виконання завдання. Для кожного БПЛА повинні бути задані його швидкість та польотний ресурс, для кожного об'єкту заданий час його обстеження. Час виконання завдання знаходиться як час виконання максимального маршруту. Час виконання маршруту обчислюється як сума добутку сумарної довжини маршруту й швидкості БПЛА та сумарного часу обстеження об'єктів.

Розглянемо критерій мінімізації вартості маршруту. Для кожного БПЛА задано польотний ресурс та вартість одиниці пройденого шляху. Вартість маршруту – це добуток його довжини та вартості одиниці пройденого шляху БПЛА, який його виконує.

2.3 Математична модель

Для поставленої оптимізаційної проблеми можна запропонувати як цілочислову постановку (булеву), так і комбінаторну. Розглянемо спочатку модель із булевими змінними. Для цього введемо позначення:

$M = \{n + 1, \dots, n + m\}$ – множина депо, m – їх кількість;

$N = \{1, \dots, n\}$ – множина цілей, які слід відвідати, n – їх кількість;

$P = M \cup N$ – множина точок (депо та цілей);

$B = \{1, \dots, b\}$ – множина БПЛА;

b – кількість БПЛА;

$D = d_{ij}, i, j = \overline{1, n + m}$ – матриця відстаней між депо та об'єктами;

$T = t_i, i \in N$ – час обстеження i -ї цілі;

$C = c_k, k \in B$ – вартість одиниці пройденого шляху k -им БПЛА;

$V = v_k, k \in B$ – середня швидкість k -го БПЛА;

$R = r_k, k \in B$ – польотний ресурс k -го БПЛА.

Включення перельоту з точки (депо, об'єкта) i в точку (депо, об'єкт) j для k -го БПЛА задається змінними x_{ijk} , що обчислюється за формулою (2.1):

$$x_{ijk} = \begin{cases} 0 & \text{– переліт не виконується} \\ 1 & \text{– переліт виконується} \end{cases}, \quad i, j \in P, k \in B. \quad (2.1)$$

Залежно від критерію задача полягає в мінімізації однієї з цільових функцій: загальної вартості місії, що обчислюється за формулою (2.2), або часу виконання місії, що обчислюється за формулою (2.3):

$$F_1 = \sum_{k \in B} \sum_{i \in P} \sum_{j \in P} c_k d_{ij} x_{ijk}; \quad (2.2)$$

$$F_2 = \max_{k \in B} \left(\sum_{i \in P} \sum_{j \in P} \frac{d_{ij} x_{ijk}}{v_k} + \sum_{i \in M} \sum_{j \in P} t_j x_{ijk} \right). \quad (2.3)$$

Обмеження:

$$\sum_{k \in B} \sum_{i \in P} x_{ijk} = 1, j \in N; \quad (2.4)$$

$$\sum_{k \in B} \sum_{j \in P} x_{ijk} = 1, i \in N; \quad (2.5)$$

$$\sum_{i \in M} \sum_{j \in N} x_{ijk} = 1, k \in B; \quad (2.6)$$

$$\sum_{i \in N} \sum_{j \in M} x_{ijk} = 1, k \in B; \quad (2.7)$$

$$\sum_{k \in B} \sum_{i \in M} \sum_{j \in M} x_{ijk} = 0; \quad (2.8)$$

$$\sum_{k \in B} \sum_{i \in P} x_{iik} = 0; \quad (2.9)$$

$$\sum_{i \in P} \sum_{j \in P} d_{ij} x_{ijk} \leq r_k, k \in B. \quad (2.10)$$

Цільова функція залежно від критерію визначає сумарні затрати від планування маршрутів для об'їзду об'єктів, як подано у формулі (2.2), або час виконання місії обстеження об'єктів, як подано у формулі (2.3).

Область визначення змінних задачі задана формулою (2.1). Кожен БПЛА відвідує певну кількість об'єктів, але прибуття до кожного об'єкту та відправлення з нього здійснюється лише один раз одним БПЛА, що обумовлюють формули (2.4) та (2.5). Формула (2.4) означає, що лише один БПЛА залишає об'єкт j , а формула (2.5) – що лише один БПЛА прибуває до об'єкта. Формула (2.6) задає умову вильоту БПЛА з депо, а формула (2.7) – умову повернення БПЛА в депо. Вимога не перелітати з одного депо безпосередньо до цього або іншого депо відображена в формулі (2.8). У формулі (2.10) відображено обмеження на польотні ресурси БПЛА.

Розглянемо комбінаторну модель задачі. Вона є більш компактною й зручною для програмної реалізації. Дано:

n – кількість досліджуваних об'єктів;

m – кількість депо;

b – кількість БПЛА;

T^i – тривалість обстеження i -го об'єкта, $i = \overline{1, n}$;

R^k – польотний ресурс k -го БПЛА, $k = \overline{1, b}$;

S^k – швидкість k -го БПЛА, $k = \overline{1, b}$;

C^k – вартість одиниці пройденого шляху для k -го БПЛА, $k = \overline{1, b}$.

Нехай $V = \{v_1, \dots, v_n\}$ – множина об'єктів дослідження, $E = \{e_1, \dots, e_m\}$ – множина депо. Розглянемо розбиття множини V на спеціальні блоки розбиття $V^i, i = \overline{1, b}$, що є упорядкованими послідовностями елементів V і задовольняють умовам (2.11) та (2.12):

$$\bigcup_{i=1}^b V^i = V; \quad (2.11)$$

$$\bigcap_{i=1}^b V^i = \emptyset. \quad (2.12)$$

Такі спеціальні блоки розбиття визначають елементи та порядок їх обходу для маршруту кожного БПЛА. Маршрут i -го БПЛА задано вектором $X^i = (x_1^i, \dots, x_l^i)$, $i = \overline{1, b}$, $l = \overline{1, |X^i|}$, де $x_1^i = x_l^i = e_i$, $e_i \in E$ – стартове депо i -го БПЛА; $(x_2^i, \dots, x_{l-1}^i) = V^i$, $i = \overline{1, b}$ – один із зазначених упорядкованих блоків розбиття. Сукупність усіх маршрутів позначимо $X = \{X^1, \dots, X^b\}$. У цих позначеннях довжина маршруту i -го БПЛА обчислюється за формулою (2.13):

$$L(X^i) = \begin{cases} \sum_{j=1}^{|X^i|} d(x_j^i, x_{j+1}^i), & \text{якщо } V^i \neq \emptyset, i = \overline{1, b} \\ 0, & \text{в іншому випадку,} \end{cases} \quad (2.13)$$

де $d(x_j^i, x_{j+1}^i)$ – відстань між відповідними пунктами (депо, об'єктами).

Залежно від критерію задача полягає в мінімізації однієї з цільових функцій. Вартість маршруту обчислюється як добуток вартості одиниці пройденого шляху БПЛА та довжини маршруту, а цільова функція обчислюється за формулою (2.14):

$$F_1(X) = \sum_{i=1}^m C^i L(X^i). \quad (2.14)$$

Тривалість виконання маршруту обчислюється як сума відношення довжини маршруту до швидкості БПЛА та сумарного часу обслуговування всіх об'єктів цього маршруту. Цільова функція для критерію часу виконання місії обчислюється за формулою (2.15):

$$F_2(X) = \max_{i=1, \overline{b}} \left(\frac{1}{s^i} L(X^i) + \sum_{j \in V^i} T^j \right). \quad (2.15)$$

Виконання умови достатності польотного ресурсу подається формулою (2.16):

$$\sum_{j=1}^{|X^i|} d(x_j^i, x_{j+1}^i) < R^i, i = \overline{1, b}. \quad (2.16)$$

Виконання умови одноразового обстеження всіх об'єктів та обстеження кожного об'єкту в результаті виконання місії описується формулами (2.11) та (2.12).

2.4 Вхідні дані задачі

Усі вхідні дані можна розділити на дві категорії: вхідні документи та вхідні дані користувача. Розглянемо більш детально кожен категорію.

Вхідними документами є файли формату .json, де міститься інформація про об'єкти дослідження, депо та БПЛА. На рисунку 2.1 наведено фрагмент вхідного файлу, де міститься інформація про об'єкти дослідження.

```
[
  "customers": [
    {
      "idx": 1,
      "duration": 2.0,
      "x": -37.67,
      "y": 53.89
    },
    {
      "idx": 2,
      "duration": 4.0,
      "x": -1.32,
      "y": -33.5
    },
    {
      "idx": 3,
      "duration": 3.0,
      "x": -7.91,
      "y": 1.86
    }
  ],
]
```

Рисунок 2.1 – Фрагмент вхідного файлу з інформацією про об'єкти

Кожне поле містить наступні атрибути:

- idx – ідентифікатор об'єкта дослідження;
- duration – час обстеження об'єкта дослідження;
- x – абсциса об'єкта дослідження на площині;
- y – ордината об'єкта дослідження на площині.

На рисунку 2.2 наведено фрагмент вхідного файлу, де міститься інформація про депо.

```
"depots": [
  {
    "idx": 151,
    "x": 98.3,
    "y": -11.39
  },
  {
    "idx": 152,
    "x": -29.36,
    "y": 67.89
  },
  {
    "idx": 153,
    "x": -63.26,
    "y": -85.15
  }
],
]
```

Рисунок 2.2 – Фрагмент вхідного файлу з інформацією про депо

Кожне поле містить наступні атрибути:

- idx – ідентифікатор депо;

- x – абсциса депо на площині;
- y – ордината депо на площині.

На рисунку 2.3 наведено фрагмент вхідного файлу, де міститься інформація про БПЛА.

```
"uavs": [
  {
    "idx": 1,
    "speed": 30.0,
    "travel_cost": 5.0,
    "max_distance": 500
  },
  {
    "idx": 2,
    "speed": 50.0,
    "travel_cost": 6.0,
    "max_distance": 300
  },
  {
    "idx": 3,
    "speed": 40.0,
    "travel_cost": 4.0,
    "max_distance": 700
  },
],
```

Рисунок 2.3 – Фрагмент вхідного файлу з інформацією про БПЛА

Кожне поле містить наступні атрибути:

- idx – ідентифікатор БПЛА;
- $speed$ – швидкість БПЛА;
- $travel_cost$ – вартість одиниці пройденого шляху;
- $max_distance$ – польотний ресурс БПЛА.

Вхідними даними користувача є вибір методу розв’язку задачі та критеріїв. Користувач може обрати один із запропонованих алгоритмів (детермінований локальний пошук (ДЛП), пошук із заборонами (табу-пошук, ТП), алгоритм імітації відпалу (АІВ), або алгоритм прискореного ймовірнісного моделювання) та один із критеріїв (час виконання завдання або вартість маршруту та необхідність мінімізації кількості БПЛА).

2.5 Вихідні дані задачі

Вихідними даними задачі є візуалізовані результати виконання алгоритмів на площині декартових координат, які користувач може переглядати для кожного окремо вибраного алгоритму й критерію, та файли формату `.json`, де міститься інформація про маршрути.

На рисунку 2.4 наведено приклад візуалізованих результатів для ДЛП за критерієм сумарної вартості місії із необхідністю мінімізації кількості задіяних БПЛА.

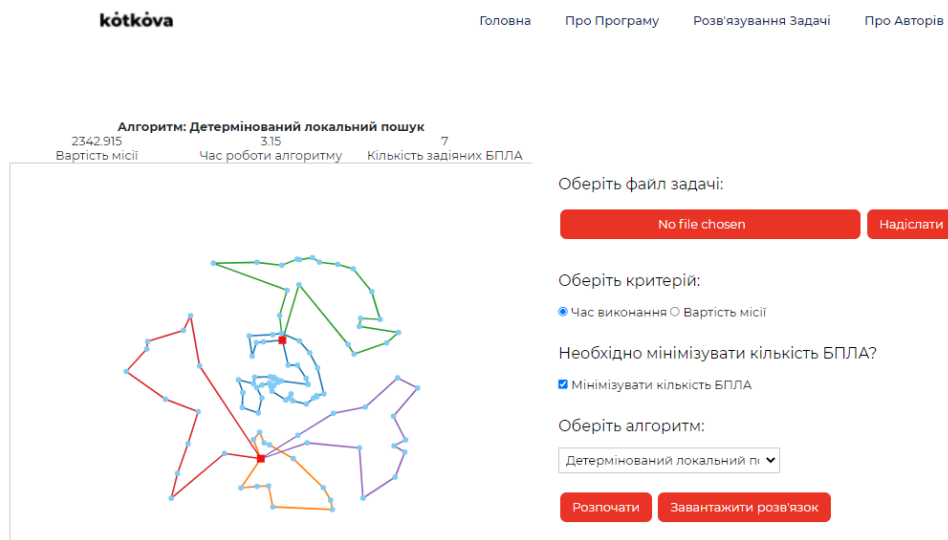


Рисунок 2.4 – Приклад візуалізованих результатів виконання алгоритмів

Основні результати роботи програми виводяться також у табличному вигляді, де зазначено сумарну вартість маршруту, час роботи алгоритму та кількість маршрутів, тобто кількість задіяних БПЛА. Після побудови маршруту користувач має змогу зберегти результати у файл формату .json з інформацією про маршрути. На рисунку 2.5 наведено приклад вихідного файлу.

```
{
  "routes": [
    {
      "uav": 1,
      "depot": 152,
      "customers": [23, 58, 104, 3, 148, 2, 10]
    },
    {
      "uav": 3,
      "depot": 151,
      "customers": [76, 102, 16, 30, 20, 4, 111, 98, 79]
    },
    {
      "uav": 6,
      "depot": 153,
      "customers": [22, 101, 70, 17, 1, 11, 68]
    }
  ]
}
```

Рисунок 2.5 – Приклад вихідного файлу з інформацією про маршрути

Кожне поле містить наступні атрибути:

- uav – ідентифікатор БПЛА;
- depot – ідентифікатор депо;
- customers – список ідентифікаторів об'єктів дослідження.

Висновки до розділу

У другому розділі наведено змістовну постановку задачі. Розглянуто проблему планування місії команди гетерогенних БПЛА, яка полягає в обстеженні та/чи обслуговуванні заданої множини об'єктів на місцевості. При цьому БПЛА не мають апріорного закріплення за конкретним депо, а метою є мінімізація затрат часу або вартості маршруту при використанні мінімальної кількості БПЛА або будь-якої кількості з доступних за наявності певних додаткових обмежень.

Для поставленої задачі виділено два класи: задача мінімізації критерію при заданій кількості БПЛА та мінімізація кількості БПЛА із подальшим покращенням критерію. При цьому розглянуто такі критерії: час виконання завдання та вартість маршруту. У результаті отримано чотири різні задачі. Якщо обрано клас із мінімізацією часу виконання чи вартості, то задача алгоритму – покращити критерій при заданій кількості БПЛА. Якщо обрано клас із мінімізацією кількості БПЛА та подальшим покращенням критерію, то першочерговим завданням алгоритму є знаходження мінімальної кількості БПЛА, а вже потім необхідно покращити критерій.

Для поставленої оптимізаційної проблеми запропоновано як булеву, так і комбінаторну постановки. Однак більш компактною й зручною для програмної реалізації є саме комбінаторна постановка. У ході побудови цих постановок введено основні позначення, описано обмеження та сформульовано дві ЦФ: мінімізація загальної вартості місії або часу її виконання.

Усі вхідні дані розділено на дві категорії: вхідні документи та вхідні дані користувача. Вхідними документами є файли формату .json, де міститься інформація про об'єкти дослідження, депо та БПЛА. Вхідними даними користувача є вибір методу розв'язування задачі та критеріїв. Вихідними даними задачі є візуалізовані результати формування маршрутів із наведенням відповідних декартових координат, які користувач може переглядати для кожного окремо вибраного алгоритму й критерію, та файли формату .json, де міститься інформація про маршрути.

3 МЕТОДИ МАРШРУТИЗАЦІЇ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

3.1 Обґрунтування методів розв’язування

Важливо зазначити, що, з точки зору програмування, реалізація алгоритмів не залежить від вибору критерію. Тому критерії «час виконання завдання» та «вартість маршруту» можна розглядати як одну задачу з різними ЦФ. Оскільки ЗМТЗ за наявності декількох депо є NP складною, то для неї доцільно застосовувати наближені алгоритми. Більшість таких алгоритмів складається з двох етапів – генерації початкового наближення та його покращення. На рисунку 3.1 наведено загальну поетапну схему роботи алгоритмів.

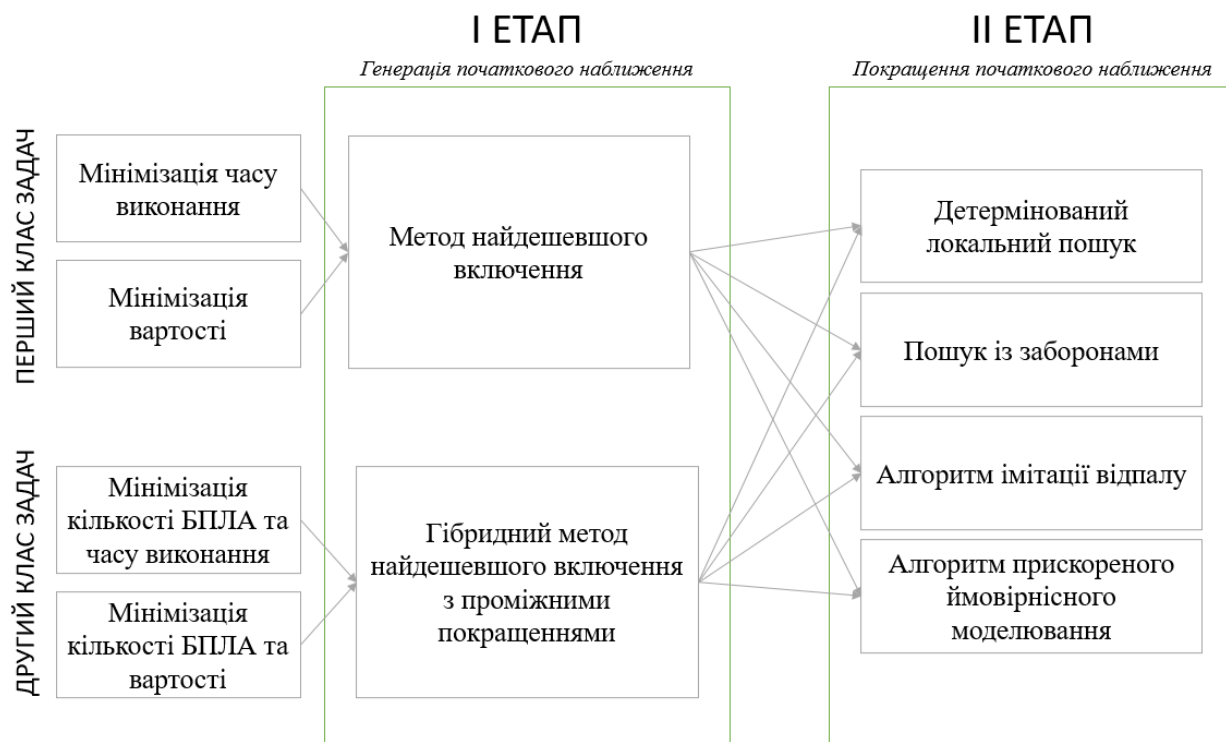


Рисунок 3.1 – Загальна поетапна схема роботи алгоритмів маршрутизації БПЛА за наявності декількох депо

Перший етап методів залежить від класу задач та критерію, а другий – лише від критерію. Як бачимо, на першому етапі реалізовується або мінімізація кількості БПЛА з подальшою мінімізацією ЦФ, або задача мінімізації ЦФ не залежно від кількості

БПЛА. А на другому етапі відбувається лише покращення ЦФ без збільшення кількості БПЛА.

3.2 Генерація початкового наближення

Розглянемо генерацію початкового наближення для двох різних класів задач: мінімізації кількості БПЛА та ЦФ. Для першого класу задач якість початкового наближення майже не впливає на кінцевий результат, тому доцільно використати алгоритм найдешевшого включення, оскільки він має низьку обчислювальну складність та простий у реалізації. Для другого класу задач необхідно знайти мінімальну кількість БПЛА на першому етапі, оскільки алгоритми другого етапу не пристосовані для цього. Тому розробляємо гібридний алгоритм генерації початкового наближення, побудований на методі найдешевшого включення та ДЛП.

3.2.1 Метод найдешевшого включення

Метод найдешевшого включення є конструктивним жадібним алгоритмом. У графічному матеріалі наведено блок-схему роботи алгоритму.

Опишемо покрокову реалізацію методу найдешевшого включення.

КРОК 1. Створюємо найкоротший із можливих маршрутів, що складаються з депо та одного об'єкту.

КРОК 2. Визначаємо найдешевше включення об'єкту в маршрут, що існує.

КРОК 3. Визначаємо найдешевше створення нового маршруту.

ЯКЩО створення маршруту дешевше, ніж включення, то КРОК 4.

КРОК 4. Створюємо найдешевший маршрут.

ІНАКШЕ КРОК 5.

КРОК 5. Виконуємо найдешевше включення об'єкту в маршрут, що існує.

ЯКЩО є об'єкти, які не включені в маршрут, то КРОК 2.

ІНАКШЕ КРОК 6.

КРОК 6. Кінець алгоритму.

На рисунку 3.2 наведено псевдокод для методу найдешевшого включення.

```

procedure НайдешевшеВключення
    невикористані об'єкти := всі об'єкти
    створити найдешевший маршрут
    видалити об'єкт зі списку невикористаних
    while (існують невикористані об'єкти)
        x := ціна найдешевшого можливого включення
        y := ціна створення найдешевшого маршруту
        if (x > y)
            здійснити найдешевше можливе включення
        else
            створити найдешевший маршрут
        endelse
        видалити об'єкт зі списку невикористаних
    endwhile
end

```

Рисунок 3.2 – Псевдокод для методу найдешевшого включення

Створення найдешевшого маршруту полягає в тому, що необхідно знайти об'єкт, розташований найближче до депо, та сформувати маршрут, що проходить з депо до цього об'єкту та назад. Найдешевше можливе включення – це включення об'єкту в маршрут, при якому значення ЦФ найменше, і не вичерпано польотний ресурс БПЛА, що виконує цей маршрут.

3.2.2 Гібридний алгоритм найдешевшого включення з проміжними покращеннями

Гібридний алгоритм найдешевшого включення з проміжними покращеннями є комбінацією алгоритму найдешевшого включення та ДЛП. У графічному матеріалі наведено блок-схему роботи алгоритму.

Опишемо покрокову реалізацію.

КРОК 1. Створюємо найкоротший із можливих маршрутів, що складаються з депо та одного об'єкту.

КРОК 2. Шукаємо найдешевше включення в маршрути, що існують.

ЯКЩО існує найдешевше включення, то КРОК 3.

КРОК 3. Здійснюємо найдешевше включення та переходимо на КРОК 2.

ІНАКШЕ КРОК 4.

КРОК 4. Перевіряємо, чи всі об'єкти включені в маршрути.

ЯКЩО всі об'єкти включені в маршрути, то КРОК 6.

ІНАКШЕ КРОК 5.

КРОК 5. Здійснюємо ДЛП для маршрутів, що існують.

ЯКЩО існує включення в маршрути, що існують, то КРОК 2.

ІНАКШЕ КРОК 1.

КРОК 6. Кінець алгоритму.

На рисунку 3.3 наведено псевдокод для гібридного алгоритму найдешевшого включення з проміжними покращеннями.

```

procedure НайдешевшеВключенняЗПокращенням
  невикористані об'єкти := всі об'єкти
  while (існують невикористані об'єкти)
    створити найдешевший маршрут
    while (існують можливі включення в маршрути що існують)
      виконати найдешевше включення
      if (немає можливих включень)
        виконати ДЛП для існуючих маршрутів
      endif
    endwhile
  endwhile
end

```

Рисунок 3.3 – Псевдокод для гібридного алгоритму найдешевшого включення з проміжними покращеннями

Гібридний алгоритм найдешевшого включення з проміжними покращеннями є модифікацією алгоритму найкращого включення, у якому створення нового маршруту відбувається тільки в тому випадку, коли неможливо здійснити включення в маршрути, що існують, які були покращені за допомогою ДЛП.

3.3 Покращення початкового наближення

Покращення початкового наближення – це другий етап роботи наближених алгоритмів, що полягає в оптимізації ЦФ. Розглянемо чотири найбільш ефективні методи розв'язування ЗКО за швидкістю обчислень та точністю: ДЛП, ТП, АІВ, G-алгоритм. Кожен із цих алгоритмів адаптуємо для розв'язку поставленої задачі.

3.3.1 Детермінований локальний пошук

ДЛП є одним із найпростіших алгоритмів локального пошуку. За точністю ДЛП належить до наближених АКО, а за типом обчислювальної схеми – до ітераційних траєкторних алгоритмів. Суть ДЛП полягає в пошуку кращих розв’язків в околі поточного. На рисунку 3.4 наведено псевдокод для ДЛП.

```

procedure ДЛП
  x := початкове наближення
  d := 1
  while d > 0 do
    X := окіл x
    for i=0 to |X| do
      d :=  $\Delta(x, X[i])$ 
      if d > 0 and do
        x := X[i]
      endif
    endfor
  endwhile
end

```

Рисунок 3.4 – Псевдокод для ДЛП

Робота ДЛП починається з початкового наближення x . На кожній ітерації формується окіл поточного розв’язку $O(x)$. Для критерію вартості місії оком розв’язку x є множина допустимих розв’язків, утворена перестановкою одного об’єкта в інше місце поточного маршруту або в будь-яке місце іншого маршруту. Для критерію часу виконання завдання оком розв’язку x є множина допустимих розв’язків, утворена перестановкою одного об’єкта з найдовшого маршруту в інше місце цього ж або іншого маршруту. В околі $O(x)$ відбувається пошук розв’язку $y \in O(x)$, для якого значення Δ -функції за формулою (3.1) найбільше.

$$\Delta = \Delta(x, y) = F(x) - F(y), \quad (3.1)$$

де F – ЦФ.

Після цього розв’язок y приймається за поточний ($x = y$). Якщо $\Delta \leq 0$, то пошук завершується, якщо $\Delta > 0$, то відбувається перехід до наступної ітерації.

3.3.2 Пошук із заборонами

ТП є різновидом ДЛП. За точністю він належить до наближених АКО, а за типом обчислювальної схеми – до ітераційних траєкторних алгоритмів. Покращення ТП полягає в тому, що під час роботи алгоритму зберігається деяка кількість попередніх розв’язків і на наступній ітерації алгоритм не може в них повернутися. Так зменшується імовірність передчасної зупинки алгоритму в локальному мінімумі. На рисунку 3.5 наведено псевдокод для ТП.

```

procedure ПошукІзЗаборонами
  s := початкове наближення
  x := початкове наближення
  tabu_list := []
  while не виконується умова завершення do
    X := окіл x
    for i=0 to |X| do
      d =  $\Delta(x, X[i])$ 
      if  $d > 0$  and  $X[i] \notin \text{tabu\_list}$  do
        x := X[i]
      endif
    endfor
    if  $\Delta(s, x) > 0$  do
      s := x
    endif
    tabu_list.append(x)
    if |tabu_list| > maxTabuSize do
      tabu_list.removeFirst()
    endif
  endwhile
end

```

Рисунок 3.5 – Псевдокод для ТП

Робота ТП починається з початкового наближення x та порожнього списку заборон. За поточний найкращий розв’язок приймається $s = x$. На кожній ітерації формується окіл поточного розв’язку $O(x)$. В околі $O(x)$ відбувається пошук розв’язку $y \in O(x)$, для якого значення Δ -функції (3.1) найбільше та який не є в списку заборон. Після цього розв’язок y приймається за поточний ($x = y$). Якщо x краще за s , то $s = x$. Поточний розв’язок x додається в кінець списку заборон. Якщо довжина списку заборон перевищує максимальну, то зі списку заборон видаляється

перший елемент. Якщо не виконується умова завершення, то необхідно перейти до наступної ітерації. Умова завершення: якщо найкращий поточний розв'язок s не покращується протягом максимальної кількості ітерацій, то робота алгоритму завершується. Результатом роботи є найкращий розв'язок s .

3.3.3 Алгоритм імітації відпалу

AIB є різновидом стохастичного локального пошуку. За точністю він належить до наближених АКО, а за типом обчислювальної схеми – до ітераційних траєкторних алгоритмів. AIB є аналогом фізичного процесу контрольованого охолодження, який також називається «відпалом» та використовує впорядкований випадковий пошук. З термодинаміки відомо, що ймовірність переходу системи з одного стану в інший обчислюється за формулою Больцмана-Гіббса (3.2):

$$p = e^{-\frac{\Delta E}{kT}}, \quad (3.2)$$

де ΔE – зміна узагальненої енергії;

T – температура;

k – стала Больцмана.

У процесі повільного керованого охолодження розплавленого металу кристалізація розплаву супроводжується глобальним зменшенням його енергії E . Однак допускаються ситуації, у яких вона може тимчасово збільшуватися. Завдяки можливості короточасного підвищення енергії можливий вихід з локальних мінімумів, які виникають при реалізації процесу. Зниження температури до абсолютного нуля робить неможливим самостійне збільшення енергії.

Для поставленої задачі аналогом фізичної системи є ЗКО, а аналогом енергії є ЦФ. Імовірність переходу обчислюється за формулою (3.3):

$$p(\Delta, T) = e^{-\frac{\Delta}{T}}; \quad (3.3)$$

$$\text{де } \Delta = F(y) - F(x). \quad (3.4)$$

При високій температурі значення імовірності переходу при $\Delta < 0$ доволі високе, тому робота алгоритму схожа на випадковий пошук. При зниженні

температури ймовірність переходів зі збільшенням ЦФ знижується. На рисунку 3.6 наведено псевдокод для AIB.

```

procedure ІмітаціяВідпалу
  x := початкове наближення
  T := початкова температура
  s := x
  while не виконується умова завершення do
    while не досягнута рівновага do
      for i = 0 to k do
        y := випадковий елемент з  $O(x)$ 
        d =  $\Delta(x, y)$ 
        P :=  $\min(1, p(d, T))$ 
        if P > random(0, 1) then
          x := y
          if  $\Delta(s, x) > 0$  do
            s := x
          endif
        endif
      endfor
    endwhile
    T := нове значення температури
  endwhile
end

```

Рисунок 3.6 – Псевдокод для AIB

Робота AIB починається з початкового наближення x , а поточний найкращий розв’язок приймається $s = x$. Встановлюється початкова температура T . Починається прогін, що складається з k ітерацій. Вибирається випадковий розв’язок $y \in O(x)$. Обчислюється Δ -функція за формулою (3.4) та ймовірність переходу $P = \min(1, p(\Delta, T))$, при цьому $p(\Delta, T)$ обчислюється за формулою (3.3). Якщо $P > \text{random}[0,1]$, то за поточний розв’язок приймається $x = y$, та якщо $\Delta(s, x) > 0$, то приймається поточний найкращий розв’язок $s = x$. Після кожного прогону необхідно перевірити, чи досягнута рівновага. Якщо так – перевірити умову завершення. Якщо умова завершення не виконується, то зменшити температуру T і почати наступний прогін. Інакше робота алгоритму завершується, а результатом його роботи є найкращий поточний розв’язок s .

Температурний розклад є геометричною прогресією з $\alpha \in [0,5; 0,99]$, тобто наступне значення температури обчислюється за формулою (3.5):

$$T_{i+1} = \alpha T_i. \quad (3.5)$$

Розглянемо умову рівноваги. Після кожного прогону перевіряється зважена зміна ЦФ відносно кожного попереднього прогону за формулою (3.6):

$$d = \frac{|f_i - f_k|}{f_i}, \quad (3.6)$$

де f_i – значення ЦФ i -го прогону;

f_k – значення ЦФ поточного прогону.

Якщо $\exists i = \overline{0, k-1}: d \leq \varepsilon$, то вважається, що рівновага досягнута.

Умовою завершення роботи алгоритму є відсутність покращення результату протягом $n > 1$ ітерацій.

3.3.4 Алгоритм прискореного ймовірнісного моделювання

Ще одним представником алгоритмів стохастичного локального пошуку є алгоритм прискореного ймовірнісного моделювання, або G-алгоритм. Він має нижчу обчислювальну складність та вищу стабільність результатів, ніж AIB. Як і в AIB, у G-алгоритмах здійснюється пошук в околі поточного наближення. Кращі наближення одразу приймаються за поточні, а гірші можуть стати поточними з певною ймовірністю.

Усе ж на відміну від AIB, ймовірність переходу до гірших варіантів залежить лише від зміни ЦФ, а фактором обмеження переходів зі збільшенням ЦФ є порогове значення, яке зменшується протягом роботи алгоритму. На рисунку 3.7 наведено псевдокод для G-алгоритму.

```

procedure Галгоритм
  x := початкове наближення
  μ := 0
  s := x
  while O(x) не обстежений повністю do
    while не досягнута рівновага do
      for i = 0 to k do
        y := випадковий елемент з O(x)
        P := p(x, y)
        ξ = μ + random[0, 1] * (1 - μ)
        if P > ξ then
          x := y
          if Δ(s, x) > 0 do
            s := x
          endif
        endif
      endfor
    endwhile
    μ := наступне значення
  endwhile
end

```

Рисунок 3.7 – Псевдокод для G-алгоритму

Робота G-алгоритму починається з початкового наближення x . За поточний найкращий розв’язок приймається $s = x$. Встановлюється початкова температура T . Починається прогін, що складається з k ітерацій. Вибирається випадковий розв’язок $y \in O(x)$. Обчислюється ймовірність переходу p з x в y за формулою (3.7):

$$p(x, y) = \begin{cases} \min\{1, \varphi(x, y)\}, & \varphi(x, y) \geq 0, \\ 0, & \text{у іншому разі,} \end{cases} \quad (3.7)$$

$$\text{де } \varphi(x, y) = 1 - \frac{F(y) - F(x)}{F(x)}. \quad (3.8)$$

Порогове значення переходу ξ обчислюється за формулою (3.9):

$$\xi = \mu + \text{random}[0, 1](1 - \mu), \quad (3.9)$$

де μ – це значення, яке зростає на кожній ітерації, при цьому наступне його значення формується на основі попереднього за допомогою G -функції, як показано на формулі (3.10):

$$\mu_{i+1} = G(\mu_i). \quad (3.10)$$

Типовою G -функцією є (3.11):

$$G_i(x) = \left(x^{\frac{1}{i}} + b\right)^k, i \in \mathbb{N}, 0 < b < 1. \quad (3.11)$$

Якщо ймовірність переходу $p > \xi$, то за поточний розв'язок приймається $x = y$, та якщо $F(s) - F(x) > 0$, то приймаємо найкраще поточне наближення $s = x$. Після кожного прогону перевіряється, чи досягнута рівновага. Якщо так, то перевіряється умова завершення. Якщо умова завершення не виконується, то необхідно змінити значення μ на наступне й почати наступний прогін. Інакше робота алгоритму завершується, а результатом роботи алгоритму є найкращий поточний розв'язок s .

Умова рівноваги та критерій завершення для G -алгоритму такі ж, як для AIB. Їх описано в підрозділі 3.3.3.

Висновки до розділу

У третьому розділі визначено, що для ЗМТЗ доцільно застосовувати наближені алгоритми, оскільки за наявності декількох депо вона є NP складною. Більшість наближених алгоритмів складаються з двох етапів – генерації початкового наближення та його покращення.

Генерацію початкового наближення розглянуто для двох різних класів задач: мінімізації кількості БПЛА та ЦФ. Для першого класу задач якість початкового наближення майже не впливає на кінцевий результат, тому доцільно використовувати алгоритм найдешевшого включення, оскільки він має низьку обчислювальну складність та простий у реалізації. Для другого класу задач необхідно знайти мінімальну кількість БПЛА на першому етапі, оскільки алгоритми другого етапу не пристосовані для цього. Тому розроблено гібридний алгоритм генерації початкового наближення, побудований на методі найдешевшого включення та ДЛП. Для кожного алгоритму наведено блок-схеми їх роботи, покрокову реалізацію та псевдокод.

Покращення початкового наближення – це другий етап роботи наближених алгоритмів, що полягає в оптимізації ЦФ. Розроблено чотири найбільш ефективні методи розв'язування ЗКО за швидкістю обчислень та точністю: ДЛП, ТП, AIB, G -алгоритм. Для кожного алгоритму наведено покрокову реалізацію та псевдокод.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При розробці програмного продукту використано такі засоби розробки, як операційну систему Ubuntu 20.04, текстовий редактор Sublime Text 3, мови програмування Python та JavaScript, мову розмітки веб-сторінок HTML5, таблицю стилів CSS3 та бібліотеки NumPy, Flask, WebGL.

Ubuntu – це операційна система для робочих станцій, комп'ютерів і серверів; найпопулярніший у світі дистрибутив Linux. Це заснована на ядрі Linux операційна система, яка підходить для використання на персональних комп'ютерах, ноутбуках і серверах. Вона містить всі необхідні програми: перегляду Інтернет, офісний пакет для роботи з текстами, електронними таблицями і презентаціями, програми для спілкування в Інтернет тощо [49].

Sublime Text 3 – швидкий кросплатформенний текстовий редактор. Він має потужний API Python, який дозволяє плагінам розширити вбудовану функціональність. Контроль пакетів можна встановити за допомогою командної палітри, забезпечуючи простий доступ до тисяч пакетів, створених спільнотою [50].

Python – це універсальна інтерпретована, об'єктно-орієнтована високорівнева мова програмування сценаріїв із динамічною семантикою. Розвинені вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять її дуже привабливою для швидкої розробки застосувань, а також для використання як скриптової або мови, що «склеює» разом наявні компоненти [51].

JavaScript – високорівнева мова програмування що розроблена компанією Netscape Communication разом з Sun. JavaScript побудована на основі Java, але не має строгої типізації. При цьому вона підтримує багато синтаксичних конструкцій Java, але не має поняття класів, використовуючи лише невеликий набір типів даних: числові, строкові й булевські [52].

HTML 5 – це удосконалена версія стандарту HTML, робота над якою розпочалася WHATWG у червні 2004 року та специфікація якої отримала назву «Web Application 1.0» [53]. HTML – це мова розмітки веб-сторінок, що застосовується для форматування контенту.

CSS – специфікація, розроблена W3C для розширення можливості форматування XML- та HTML-документів з метою спрощення керування процесом їх відображення у веб-браузерах. У створюваній об'єктній моделі документа останній уявляється таким, що складається з набору деяких об'єктів. Усі об'єкти в документі взаємозалежні й об'єднані загальною деревовидною архітектурою. Використання CSS дозволяє відокремити структуру й зміст документа від рівня його подання користувачеві. Одна таблиця може використовуватися в безлічі документів [52].

NumPy – пакет, що додає Python підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Основні пакети, які доповнюють NumPy, це: SciPy і Matplotlib [54].

Flask – фреймворк для створення веб-додатків, що виступає в ролі веб-сервера. Він не потребує підключення додаткових бібліотек та розроблений за допомогою мови Python.

WebGL є реалізацією інтерфейсу для роботи з графічними елементами для мови JavaScript. WebGL є частиною HTML 5, створений на основі стандарту OpenGL ES 2.0 та підтримується всіма сучасними браузерами, наприклад, Chrome, Opera, Mozilla Firefox, Safari, Chromium тощо.

4.2 Архітектура програмного забезпечення

Наведемо схеми структурні розгортання, послідовності, діяльності та класів для розуміння логіки роботи програми. Також опишемо специфікацію функцій, що знаходяться в окремих файлах.

4.2.1 Схема структурна розгортання

У графічному матеріалі наведено схему структурну розгортання. Опишемо її більш детально. Програма складається із серверної та клієнтської частини.

Ієрархія наступна:

- Server, у який входить Server OS; у нього, у свою чергу, входить Web-server; Web-server складається з двох компонентів: Api та Routing;
- Client, у який входить Client OS; у нього, у свою чергу, входить Client Browser.

4.2.2 Схема структурна послідовності

У графічному матеріалі наведено схему структурну послідовності. Опишемо детальніше послідовність дій:

- від клієнта до порталу: «Завантажити файл з умовою задачі»;
- від клієнта до порталу: «Обрати параметри задачі»;
- від клієнта до порталу: «Обрати критерії задачі»;
- від порталу до сервера: «Надіслати запит про пошук розв'язку»;
- на сервері: «Почати пошук розв'язку задачі»;
- від порталу до сервера: «Надіслати запит про найкращий поточний розв'язок»;
- від порталу до клієнта: «Відобразити найкращий поточний розв'язок»;
- від сервера до порталу: «Надіслати кінцевий розв'язок задачі»;
- від порталу до клієнта: «Відобразити кінцевий розв'язок»;
- від клієнта до порталу: «Надіслати запит на завантаження файлу з кінцевим розв'язком»;
- від порталу до клієнта: «Надіслати файл із кінцевим розв'язком».

4.2.3 Схема структурна діяльності

У графічному матеріалі наведено схему структурну діяльності. Клієнту необхідно:

- завантажити файл із вхідними даними;
- обрати один із алгоритмів: ДЛП, ТП, AIB або G-алгоритм;
- обрати критерій: час виконання або вартість маршруту;
- обрати необхідність мінімізації кількості БПЛА.

У відповідь на дії клієнта порталу необхідно надати кінцевий розв'язок задачі. Після цього клієнт може переглянути та завантажити отриманий розв'язок, на чому роботу програми завершується.

4.2.4 Діаграма класів

У графічному матеріалі наведено діаграму класів. Є базовий клас Node, він же вершина графу, від якого наслідуються класи Customer та Depot. Клас Node, у свою чергу, наслідується від класу Position. Клас Graph містить у собі списки об'єктів класів Depot, Customer та UAV. Клас Route містить у собі об'єкт класу UAV та список об'єктів класу Customer. Клас Solution містить у собі список об'єктів класу Route.

- Клас Position має поля x та y типу float та метод distance, що повертає значення типу float.
- Клас Node має метод distance, що повертає значення типу float.
- Класи Depot та Customer є нащадками класу Node та не мають своїх полів та методів.
- Клас Graph має поля customers (список об'єктів класу Customer), depots (список об'єктів класу Depot), uavs (список об'єктів класу UAV) та методи irc_greedy, irc_hybrid, що повертають об'єкт класу Solution, та validate, що повертає булеве значення.
- Клас UAV має поля depot (об'єкт класу Depot), resource та speed типу float.
- Клас Route має поля targets (список об'єктів класу Customer), uav (об'єкт класу UAV) та методи equals (повертає булеве значення), length та score (повертають значення float), update (не повертає ніяких значень).

– Клас Solution має поле routes (список об'єктів класу Routes), методи DLP (не повертає ніяких значень), equals (повертає булеве значення), Galgorithm (не повертає ніяких значень), is_valid (не повертає ніяких значень), length (повертає значення типу float), SA (не повертає ніяких значень), score (повертає значення типу float), TabuSearch (не повертає ніяких значень).

4.2.5 Специфікація функцій

Функції згруповані за їх призначенням у окремих файлах: irc.py, search_utils.py, algorithms.py, solution.py, graph.py, app.py, api.py. Розглянемо окремо кожен файл та функції, які в ньому знаходяться. irc.py У таблиці 4.1 наведено детальний опис функцій, що знаходяться у файлі irc.py, що містить у собі функції, за допомогою яких формується початкове наближення.

Таблиця 4.1 – Опис функцій із файлу irc.py

Назва функції	Опис
_get_uav()	Функція знаходження найкращого БПЛА для створення маршруту
_get_best_insertion()	Функція знаходження найкращого включення в існуючий маршрут
less_uavs()	Метод найдешевшого включення
shorter_routes()	Метод найдешевшого включення без оптимізації кількості БПЛА
get_initial_solution()	Інтерфейс для створення початкового наближення

У таблиці 4.2 наведено детальний опис функцій, що знаходяться у файлі search_utils.py, що містить у собі допоміжні функції для алгоритмів покращення початкового наближення.

Таблиця 4.2 – Опис функцій із файлу search_utils.py

Назва функції	Опис
get_benefit()	Зменшення ЦФ при видаленні об'єкта з маршруту
get_cost()	Збільшення ЦФ при додаванні об'єкту до маршруту
get_random_neighbour()	Вибір випадкового наближення з околу поточного

У таблиці 4.3 наведено детальний опис функцій, що знаходяться у файлі algorithms.py, що містить у собі алгоритми покращення початкового наближення.

Таблиця 4.3 – Опис функцій із файлу *algorithms.py*

Назва функції	Опис
local_search()	ДЛП
local_search.get_better_solution()	Функція знаходження найкращого розв'язку з околу поточного
tabu_search()	ТП
tabu_search.get_better_solution()	Функція знаходження найкращого розв'язку з околу поточного з урахуванням списку заборон
simulated_annealing()	АІВ
g_algorithm()	G-алгоритм

У таблиці 4.4 наведено детальний опис функцій, що знаходяться у файлі *solution.py*, що містить у собі клас розв'язку та допоміжні класи.

Таблиця 4.4 – Опис функцій із файлу *solution.py*

Назва функції	Опис
solution.score()	Значення ЦФ для поточного розв'язку
solution.update()	Оновлення ЦФ поточного розв'язку
solution.for_render()	Представлення розв'язку для відображення
route.update()	Значення ЦФ для маршруту
route.score()	Оновлення ЦФ маршруту
node.distance()	Відстань між точками графу

У таблиці 4.5 наведено детальний опис функцій, що знаходяться у файлі *graph.py*, що містить у собі клас графу.

Таблиця 4.5 – Опис функцій із файлу *graph.py*

Назва функції	Опис
graph.from_json()	Зчитування графу з файлу формату .json
graph.for_render()	Представлення графу для відображення

У таблиці 4.6 наведено детальний опис функцій, що знаходяться у файлі *app.py*, що містить у собі функції, які використовуються веб-сервером.

Таблиця 4.6 – Опис функцій із файлу *app.py*

Назва функції	Опис
main()	Функція для відображення головної сторінки сайту
about()	Функція для відображення сторінки з інформацією про програму

Продовження таблиці 4.6

<code>solver()</code>	Функція для відображення сторінки з побудовою маршрутів
<code>authors()</code>	Функція для відображення сторінки з інформацією про авторів
<code>solution()</code>	Функція, що повертає найкращий поточний розв'язок

У таблиці 4.7 наведено детальний опис функцій, що знаходяться у файлі `api.py`, що містить у собі функції для зв'язку веб-сервера з алгоритмами.

Таблиця 4.7 – Опис функцій із файлу `api.py`

Назва функції	Опис
<code>start_solving()</code>	Функція для початку розв'язання задачі
<code>get_current_solution()</code>	Функція, що повертає найкращий поточний розв'язок для відображення
<code>get_graph()</code>	Функція, що повертає граф для відображення

На HTML-сторінці з побудовою маршрутів є функція `plot()` для відображення об'єктів, депо та маршрутів на декартовій системі координат.

4.3 Настанова з використання

Для роботи із системою користувачу необхідно мати мінімальні навички роботи з комп'ютером. Для зручності розуміння функціонування програмного продукту на рисунку 4.1 наведено схему переходу між сторінками.



Рисунок 4.1 – Схема переходу між сторінками програмного продукту

Робота з програмою починається з головної сторінки. Звідси користувач має змогу перейти на сторінку з можливістю розв'язування задач, сторінку з порівнянням роботи алгоритмів, а також до сторінок з інформацією про програму та авторів.

Опишемо детальніше основні дії, що користувач може виконувати із запропонованою системою. На рисунку 4.2 представлено головну сторінку програмного продукту.

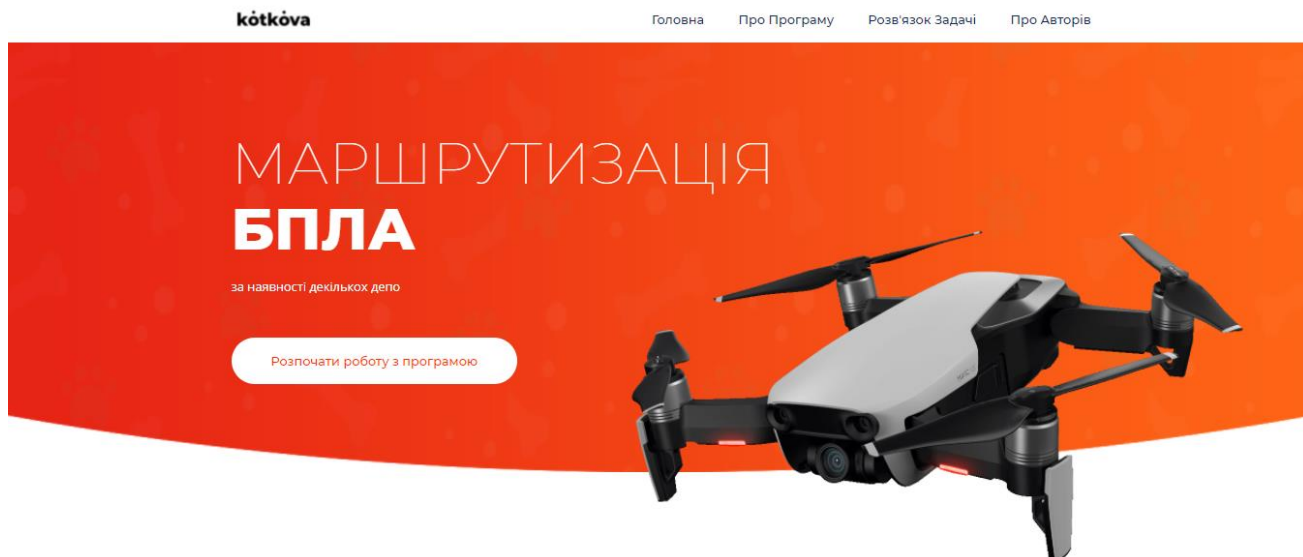


Рисунок 4.2 – Головна сторінка

На головній сторінці є меню, що використовується на всіх сторінках, назва програмного продукту та кнопка, за допомогою якої можна перейти до сторінки «Розв'язування задачі». Розглянемо головне меню програмного продукту (рисунок 4.3).

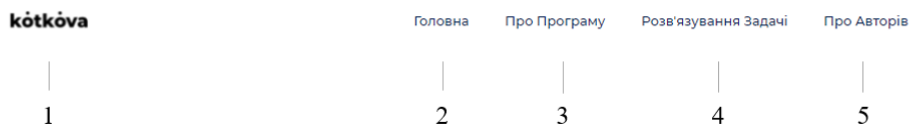


Рисунок 4.3 – Головне меню

Реакцію системи на дії користувача при роботі з головним меню наведено в таблиці 4.8.

Таблиця 4.8 – Реакція системи на дії користувача на головному меню

№	Опис-реакція системи
1	При натисканні на логотип система відкриває головну сторінку.
2	При натисканні на кнопку «Головна» система відкриває головну сторінку.

Кінець таблиці 4.8

3	При натисканні на кнопку «Про програму» система відкриває сторінку, де користувач має змогу ознайомитися з інформацією про запропонований програмний продукт
4	При натисканні на кнопку «Розв’язування задачі» система відкриває сторінку, де користувач має змогу розв’язати свою задачу.
5	При натисканні на кнопку «Про авторів» система відкриває сторінку, де користувач має змогу ознайомитися з інформацією про авторів.

На рисунку 4.4 представлено сторінку з інформацією про програму. На цій сторінці користувач має змогу ознайомитися з інформацією про запропонований програмний продукт.

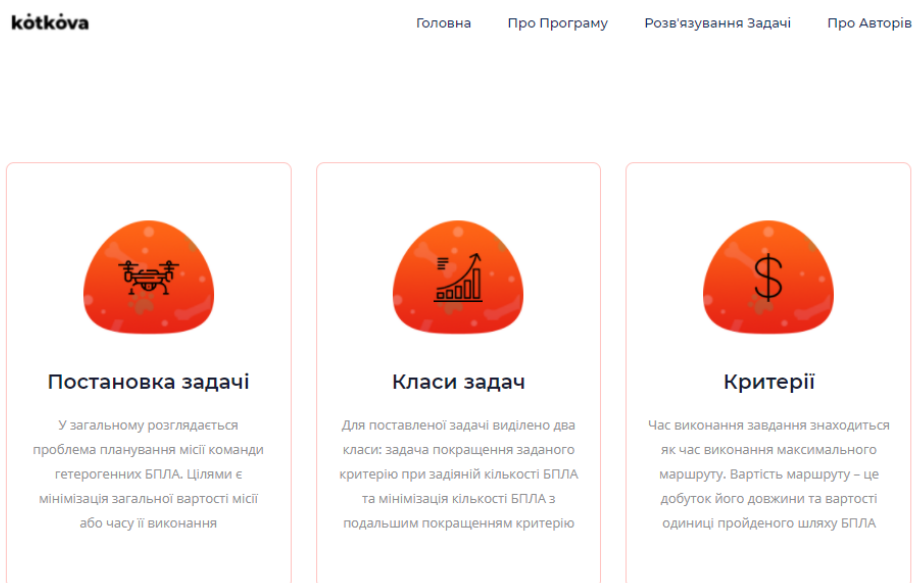


Рисунок 4.4 – Сторінка з інформацією про програму

На сторінці «Розв’язування задачі» (рисунок 4.5) користувач може завантажити свою задачу, розв’язати її одним із запропонованих алгоритмів за одним із критеріїв, а також переглянути візуалізовані на декартовій системі координат розв’язки та, за необхідності, завантажити файл із результатами задачі собі на комп’ютер.

Рисунок 4.5 – Сторінка «Розв’язування задачі»

При натисканні на кнопку «No file chosen» система дає можливість користувачу завантажити необхідний файл формату .json, який необхідний для початку роботи програми. Після натискання на кнопку «Надіслати» на декартовій системі координат відображається карта з об’єктами та депо. Після цього стає активним поле, де користувачу необхідно обрати критерій (час виконання або вартість місії), вказати необхідність мінімізації кількості БПЛА та обрати алгоритм (ДЛП, ТП, AIV або G-алгоритм). У разі, якщо критерій не вибрано, за замовчуванням для розв’язку обирається критерій час виконання, якщо алгоритм не вибрано, то за замовчуванням для розв’язку обирається ДЛП, якщо рішення щодо мінімізації не прийнято, за замовчуванням обирається варіант «Так». Після того, як користувач заповнив усі поля, йому необхідно натиснути кнопку «Розпочати», після чого система надає результати роботи. Після того, як програма закінчить роботу, з’являється кнопка «Завантажити розв’язок», вона надає користувачу можливість зберегти файл формату .json з інформацією про маршрути на свій комп’ютер. На рисунку 4.6 наведено приклад результату роботи програми для випадку, коли користувач обрав критерій

«Час виконання», вирішив, що необхідно мінімізувати кількість БПЛА та обрав алгоритмом для розв’язування задачі ДЛП.

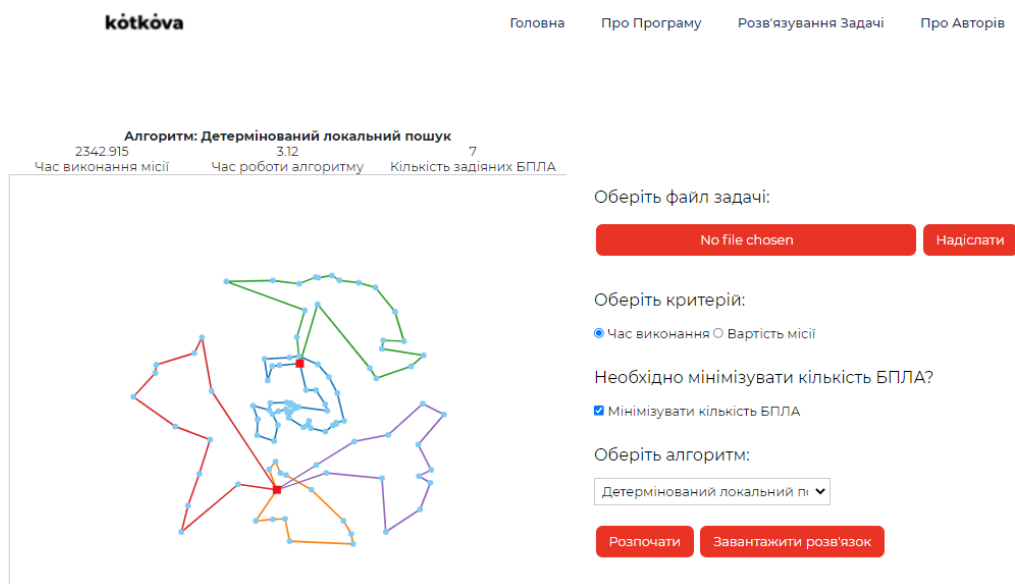


Рисунок 4.6 – Приклад результату роботи програми

Результат роботи програми надано в наступному вигляді: назва алгоритму, сумарний час виконання місії, час роботи алгоритму, кількість маршрутів, тобто кількість задіяних БПЛА, та візуалізовані результати.

На рисунку 4.7 наведено сторінку з інформацією про авторів.

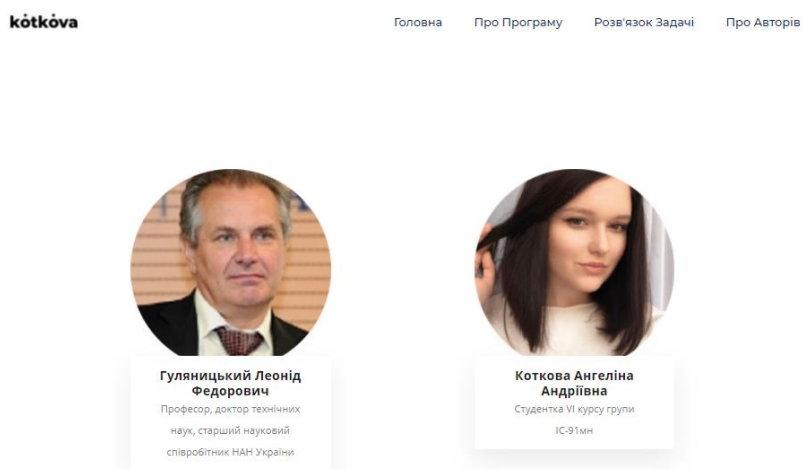


Рисунок 4.7 – Сторінка з інформацією про авторів

На сторінці «Про авторів» користувач має змогу ознайомитися з короткою інформацією про розробників запропонованого програмного продукту.

Висновки до розділу

У четвертому розділі розглянуто основні засоби розробки та наведено їх детальну характеристику. Так, при написанні програми використано операційну систему Ubuntu 20.04, текстовий редактор Sublime Text 3, мови програмування Python та JavaScript, мову розмітки веб-сторінок HTML5, таблицю стилів CSS3 та бібліотеки NumPy, Flask та WebGL.

Наведено архітектуру програмного забезпечення за допомогою схем структурних розгортання, послідовності, діяльності та діаграми класів, які наведено в графічному матеріалі. Програма складається із серверної та клієнтської частин.

Описано специфікацію функцій. Так, функції згруповані за їх призначенням у окремих файлах: `irc.py`, `search_utils.py`, `algorithms.py`, `solution.py`, `graph.py`, `app.py`, `api.py`. На HTML-сторінці з побудовою маршрутів є окрема функція для відображення об'єктів, депо та маршрутів на декартовій системі координат.

Для роботи із системою користувачу необхідно мати мінімальні навички роботи з комп'ютером. Програмний продукт складається з 4 сторінок, між якими користувач може вільно переключатися. Робота програми починається з головної сторінки. З неї користувач може здійснювати перехід на сторінки з інформацією про програму, з можливістю розв'язування задач та з інформацією про авторів програми. На сторінці «Розв'язування задач» користувач має можливість завантажити файл зі своєю задачею, обрати потрібний критерій, необхідність мінімізації кількості БПЛА та алгоритм. Після цього система починає обрахунок та видає результат у вигляді візуалізованих результатів на декартовій системі координат. Також користувач має змогу завантажити файл формату `.json` з інформацією про маршрути на свій комп'ютер.

5 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ

5.1 Умови проведення досліджень

Усі дослідження проведено на ноутбуці HP ProBook 4540s з такими характеристиками:

- процесор Intel® Core™ i5-3210M з двома ядрами, чотирма потоками та тактовою частотою ядра 2,6 ГГц;
- обсяг оперативної пам'яті 8 Гб.

Для проведення експериментів згенеровано 9 задач:

- 3 задачі з 2 депо та 100 об'єктами;
- 3 задачі з 3 депо та 150 об'єктами;
- 3 задачі з 4 депо та 200 об'єктами.

При порівнянні ефективності всіх алгоритмів використовується усереднене значення ЦФ, що розраховується як середнє гармонійне, та часу виконання алгоритму за 10 запусків. Середнє гармонійне розраховується за формулою (5.1):

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}, x_1, x_2, \dots, x_n > 0. \quad (5.1)$$

Середнє гармонійне доцільно використовувати, оскільки воно більш чутливе до менших значень, ніж середнє арифметичне та середнє геометричне. Це дозволяє врахувати стабільність роботи алгоритму, оскільки алгоритм з меншою стабільністю (більшим розкидом) результатів матиме нижче усереднене середнє арифметичне значень ЦФ.

Для всіх алгоритмів максимальний час роботи обмежений та складає 60 с. Якщо алгоритм працює довше, то результатом його роботи вважається найкращий поточний розв'язок на цей момент часу. Для підбору параметрів алгоритмів застосовано локальний пошук. Експерименти проведено окремо для різних критеріїв: Критерій 1 – час виконання завдання; Критерій 2 – вартість маршруту.

5.2 Дослідження впливу параметрів на роботу алгоритмів

Для різних алгоритмів є різні параметри. Наприклад, для ДЛП параметрів немає взагалі, а от параметри для ТП, АІВ та G-алгоритму розглянемо в наступних підрозділах. Для налаштування параметрів використано ДЛП.

5.2.1 Параметри пошуку із заборонами

Для ТП є два параметри: довжина списку заборон та максимальна кількість ітерацій без покращення результату. У якості простору для локального пошуку вибрано значення параметрів з певним кроком:

- довжина списку заборон: від 5 до 50 з кроком 5;
- максимальна кількість ітерацій без покращення результату: від 10 до 100 з кроком 10.

У якості початкових параметрів алгоритму вибрано середнє значення в наведеному діапазоні. Початкові та кінцеві значення параметрів для ТП наведено в таблиці 5.1.

Таблиця 5.1 – Початкові та кінцеві параметри ТП

Параметр	Початкове значення	Кінцеве значення Критерію 1	Кінцеве значення Критерію 2
Довжина списку заборон	30	35	35
Максимальна кількість ітерацій без покращення результату	50	70	60

Для ТП середнє значення ЦФ для Критерію 1 при запуску на всіх задачах при початкових значеннях параметрів становить 345,62, а для кінцевих значень параметрів – 337,02. Середнє значення ЦФ для Критерію 2 при запуску на всіх задачах при початкових значеннях параметрів становить 2434,35, а для кінцевих значень параметрів – 2423,72.

5.2.2 Параметри алгоритму імітації відпалу

Для АІВ є п'ять параметрів: початкова температура, тривалість прогону, коефіцієнт зниження температури, поріг досягнення рівноваги та максимальна

кількість ітерацій без покращення результату. У якості простору для локального пошуку вибрано значення параметрів з певним кроком:

- початкова температура: від 1 до 10 з кроком 1;
- тривалість прогону: від 5 до 30 з кроком 5;
- коефіцієнт зниження температури: від 0,9 до 0,99 з кроком 0,01;
- поріг досягнення рівноваги: від 0,001 до 0,01 з кроком 0,001;
- максимальна кількість ітерацій без покращення результату: від 10 до 100

з кроком 10.

Початкові та кінцеві значення параметрів для AIB наведено в таблиці 5.2.

Таблиця 5.2 – Початкові та кінцеві параметри AIB

Параметр	Початкове значення	Кінцеве значення Критерію 1	Кінцеве значення Критерію 2
Початкова температура	5	7	7
Тривалість прогону	20	20	15
Коефіцієнт зниження температури	0,95	0,98	0,99
Поріг досягнення рівноваги	0,005	0,01	0,01
Максимальна кількість ітерацій без покращення результату	50	30	40

Для AIB середнє значення ЦФ для Критерію 1 при запуску на всіх задачах при початкових значеннях параметрів становить 334,23, а для кінцевих значень параметрів – 319,78. Середнє значення ЦФ для Критерію 2 при запуску на всіх задачах при початкових значеннях параметрів становить 2421,45, а для кінцевих значень параметрів – 2372,12.

5.2.3 Параметри алгоритму прискореного ймовірнісного моделювання

Для G-алгоритму є чотири параметри: довжина прогону, коефіцієнт b для G -функції, поріг досягнення рівноваги та максимальна кількість ітерацій без покращення результату. У якості простору для локального пошуку вибрано значення параметрів з певним кроком:

- довжина прогону: від 5 до 30 з кроком 5;

- коефіцієнт b для G -функції: від 0,0001 до 0,001 з кроком 0,0001;
- поріг досягнення рівноваги: від 0,001 до 0,01 з кроком 0,001;
- максимальна кількість ітерацій без покращення результату: від 10 до 100 з кроком 10.

Початкові та кінцеві значення параметрів для G -алгоритму наведено в таблиці 5.3.

Таблиця 5.3 – Початкові та кінцеві параметри G -алгоритму

Параметр	Початкове значення	Кінцеве значення Критерію 1	Кінцеве значення Критерію 2
Довжина прогону	20	15	20
Коефіцієнт b для G -функції	0,0005	0,001	0,001
Поріг досягнення рівноваги	0,005	0,003	0,002
Максимальна кількість ітерацій без покращення результату	50	40	30

Для G -алгоритму середнє значення ЦФ для Критерію 1 при запуску на всіх задачах при початкових значеннях параметрів становить 329,5, а для кінцевих значень параметрів – 315,61. Середнє значення ЦФ для Критерію 2 при запуску на всіх задачах при початкових значеннях параметрів становить 2407,77, а для кінцевих значень параметрів – 2316,43.

5.2.4 Результати налаштування параметрів алгоритмів

У результаті налаштування параметрів за допомогою алгоритму локального пошуку значення ЦФ зменшується. У таблиці 5.4 та 5.5 наведено результати налаштування параметрів алгоритмів для Критерію 1 та Критерію 2 відповідно.

Таблиця 5.4 – Результати налаштування параметрів алгоритмів для Критерію 1

Алгоритм	ТП	АІВ	G -алгоритм
Початкове значення ЦФ	345,62	334,23	329,5
Кінцеве значення ЦФ	337,02	319,78	315,61
Відсоток покращення	2,55%	4,5%	4,4%

Таблиця 5.5 – Результати налаштування параметрів алгоритмів для Критерію 2

Алгоритм	ТП	АІВ	Г-алгоритм
Початкове значення ЦФ	2434,35	2421,45	2407,77
Кінцеве значення ЦФ	2423,72	2372,12	2316,43
Відсоток покращення	0,43%	2,08%	3,9%

Після налаштування параметрів алгоритму результат покращився в середньому на 3,8% для Критерію 1 та на 2,1% для Критерію 2.

5.3 Порівняння результатів роботи алгоритму

Необхідно запустити кожен алгоритм для кожного з 9 файлів. Робота алгоритмів оцінюється за двома критеріями оцінки алгоритмів: якість кінцевого розв'язку та час роботи алгоритму. Кожен алгоритм перевіряємо за цими критеріями.

5.3.1 Дослідження кінцевих результатів

Проведемо дослідження значень ЦФ для критерію часу виконання завдання (Критерій 1) з необхідністю мінімізації кількості БПЛА. У таблиці 5.6 наведено значення ЦФ за Критерієм 1 з мінімізацією по БПЛА.

Таблиця 5.6 – Значення ЦФ для Критерію 1 з мінімізацією по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	Г-алгоритм
1	100	2	492,76	425,58	485,78	405,31
2	100	2	438,27	438,27	430,29	414,48
3	100	2	466,85	431,76	455,80	408,11
Середнє значення ЦФ			465,96	431,87	457,25	409,3
4	150	3	460,04	437,13	438,20	420,17
5	150	3	473,74	448,92	459,84	421,08
6	150	3	488,39	462,48	486,12	460,98
Середнє значення ЦФ			474,06	449,51	461,39	434,08
7	200	4	484,77	453,59	469,47	426,39
8	200	4	474,23	466,43	466,90	444,70
9	200	4	477,65	443,57	474,70	422,89
Середнє значення ЦФ			478,88	454,53	470,36	431,33
Загальне середнє значення ЦФ			472,99	445,30	463,01	424,90

Як бачимо із таблиці 5.6, із загального середнього значення ЦФ G-алгоритм має найменше значення ЦФ, а тому він найкраще працює для запропонованого типу задач.

Для наочності отриманих результатів дослідження побудуємо графік залежності середнього значення ЦФ для різних алгоритмів у залежності від кількості об'єктів (рисунок 5.1) для критерію часу виконання завдання з необхідністю мінімізації кількості БПЛА.

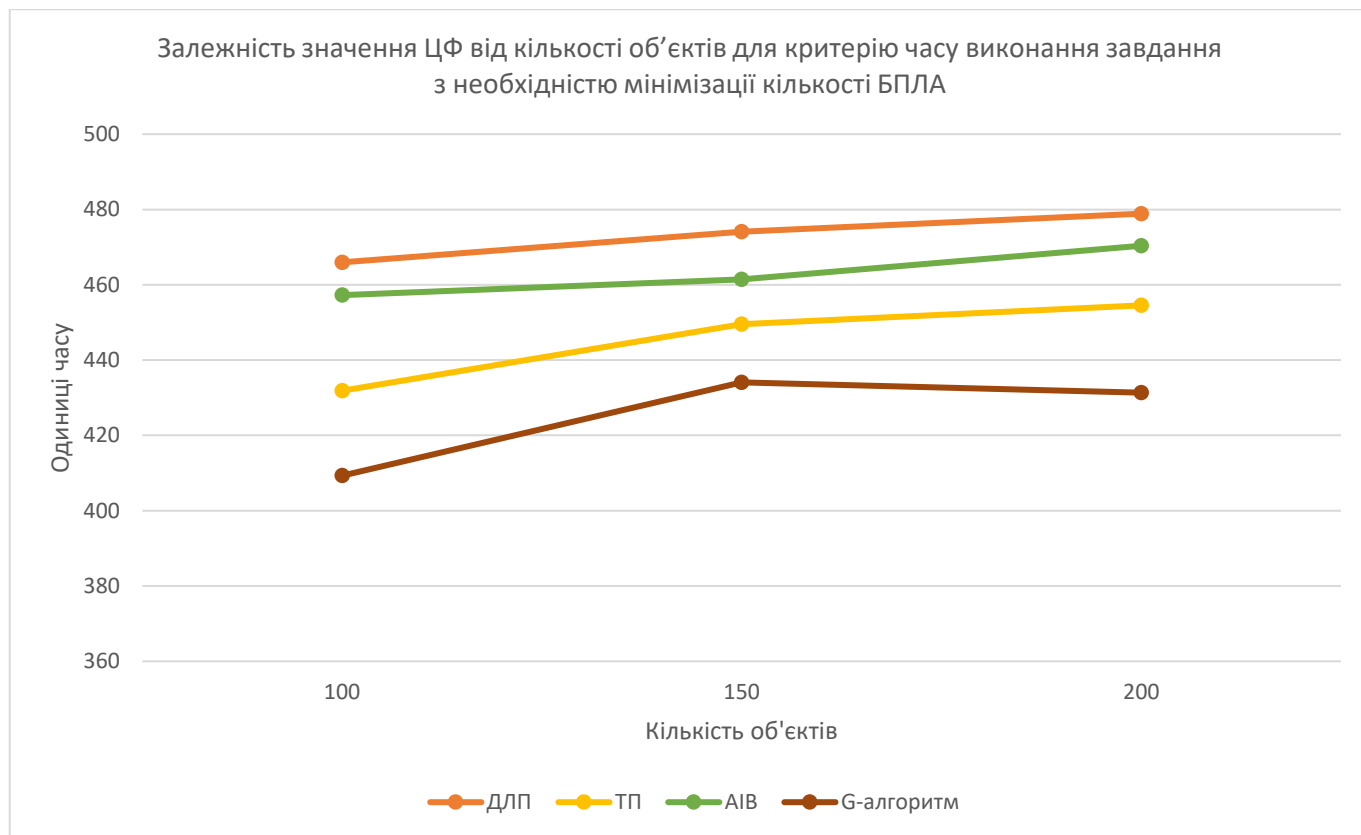


Рисунок 5.1 – Залежність значення ЦФ від кількості об'єктів для критерію часу виконання завдання з необхідністю мінімізації кількості БПЛА

На рисунку 5.1 бачимо графіки залежності ЦФ від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення ЦФ. Для побудови графіку обчислено середні значення ЦФ для різних кількостей об'єктів та для кожного алгоритму. Як

бачимо з графіку, G-алгоритм має кращі результати для будь-якої кількості об'єктів, далі йде ТП, на третьому місці – АІВ, а найгіршим виявився ДЛП.

Проведемо дослідження значень ЦФ для критерію вартості маршруту (Критерій 2) з необхідністю мінімізації кількості БПЛА. У таблиці 5.7 наведено значення ЦФ за Критерієм 2 з мінімізацією по БПЛА.

Таблиця 5.7 – Значення ЦФ для Критерію 2 з мінімізацією по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	G-алгоритм
1	100	2	2327,18	2327,18	2246,18	2178,08
2	100	2	2342,91	2342,91	2307,08	2269,46
3	100	2	2339,84	2334,54	2226,85	2227,90
Середнє значення ЦФ			2336,64	2334,87	2260,03	2225,14
4	150	3	2977,90	2950,72	2865,42	2802,85
5	150	3	3079,57	3079,57	3011,28	2942,18
6	150	3	2926,06	2921,68	2783,88	2758,31
Середнє значення ЦФ			2994,51	2983,99	2886,86	2834,44
7	200	4	3300,84	3300,84	3218,72	3296,69
8	200	4	3298,21	3298,21	3182,59	3148,07
9	200	4	3231,36	3229,56	3119,30	3045,26
Середнє значення ЦФ			3276,80	3276,20	3173,53	3163,34
Загальне середнє значення ЦФ			2869,32	2865,02	2773,48	2740,98

Як бачимо із таблиці 5.7, із загального середнього значення ЦФ G-алгоритм має найменше значення ЦФ, а тому він найкраще працює для запропонованого типу задач.

Побудуємо графік залежності значення ЦФ для різних алгоритмів в залежності від кількості об'єктів (рисунок 5.2) для критерію вартості маршруту з необхідністю мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення ЦФ для різних кількостей об'єктів та для кожного алгоритму.

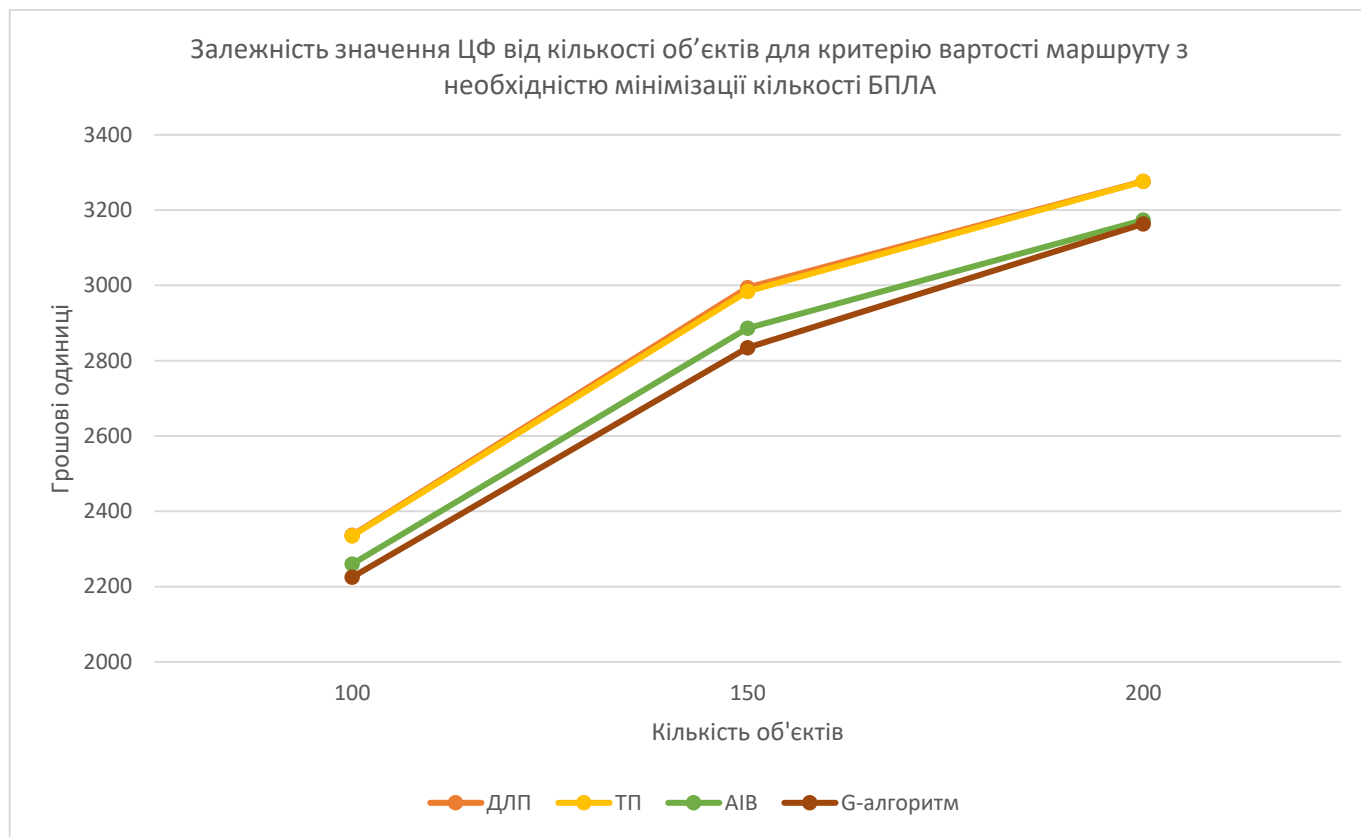


Рисунок 5.2 – Залежність значення ЦФ від кількості об'єктів для критерію вартості маршруту з необхідністю мінімізації кількості БПЛА

На рисунку 5.2 бачимо графіки залежності ЦФ від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення ЦФ. Як бачимо з графіку, G-алгоритм має кращі результати для будь-якої кількості об'єктів, далі йде АІВ, і приблизно однаковими виявилися ДЛП та ТП. Для 150 об'єктів ТП працює трохи краще, але це покращення є незначним.

Проведемо дослідження значень ЦФ для критерію часу виконання завдання (Критерій 1) без необхідності мінімізації кількості БПЛА. У таблиці 5.8 наведено значення ЦФ за Критерієм 1 без мінімізації по БПЛА.

Таблиця 5.8 – Значення ЦФ для Критерію 1 без мінімізації по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	AIB	G-алгоритм
1	100	2	374,94	339,12	306,98	320,83
2	100	2	279,37	332,99	327,48	259,11
3	100	2	347,75	281,65	347,48	324,26
Середнє значення ЦФ			334,02	317,92	327,32	301,4
4	150	3	289,38	272,26	327,59	254,89
5	150	3	347,97	300,76	309,81	304,64
6	150	3	375,92	306,14	327,94	300,76
Середнє значення ЦФ			337,75	293,05	321,78	286,76
7	200	4	358,92	358,9	359,7	331,19
8	200	4	369,76	356,88	289,08	269,51
9	200	4	298,07	299,78	304,67	265,81
Середнє значення ЦФ			342,25	338,52	317,82	288,84
Загальне середнє значення ЦФ			338,01	316,50	322,30	292,33

Як бачимо із таблиці 5.8, із загального середнього значення ЦФ G-алгоритм має найменше значення ЦФ, а тому він найкраще працює для запропонованого типу задач.

Побудуємо графік залежності значення ЦФ для різних алгоритмів у залежності від кількості об'єктів (рисунок 5.3) для критерію часу виконання завдання без необхідності мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення ЦФ для різних кількостей об'єктів та для кожного алгоритму.

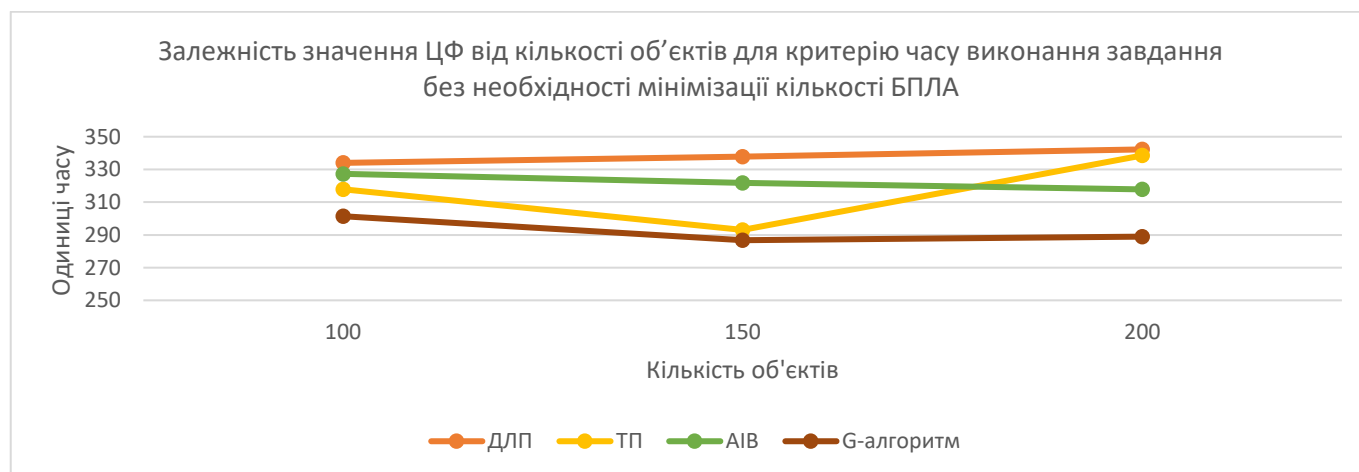


Рисунок 5.3 – Залежність значення ЦФ від кількості об'єктів для критерію часу виконання завдання без необхідності мінімізації кількості БПЛА

На рисунку 5.3 бачимо графіки залежності ЦФ від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення ЦФ. Як бачимо з графіку, G-алгоритм має кращі результати для будь-якої кількості об'єктів, далі йде ТП, але на великих даних він працює значно гірше, ніж АІВ. Для невеликих даних на третьому місці знаходиться АІВ, а найгірше для всіх задач працює ДЛП.

Проведемо дослідження значень ЦФ для критерію вартості маршруту (Критерій 2) без необхідності мінімізації кількості БПЛА. У таблиці 5.9 наведено значення ЦФ за Критерієм 2 без мінімізації по БПЛА.

Таблиця 5.9 – Значення ЦФ для Критерію 2 без мінімізації по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	G-алгоритм
1	100	2	1825,99	1681,76	1413,79	1347,71
2	100	2	1555,31	1857,02	1760,62	1425,45
3	100	2	1745,28	1661,92	1372,53	1370,36
Середнє значення ЦФ			1708,86	1733,57	1515,65	1381,17
4	150	3	2156,4	1972,43	1929,99	1900,13
5	150	3	2287,51	1991,2	2252,11	2318,53
6	150	3	1890,92	2198,33	1716,51	1726,87
Середнє значення ЦФ			2111,61	2053,99	1966,20	1981,84
7	200	4	2275,6	1992,13	2152,77	2041,46
8	200	4	2036,06	2230,55	2454,21	2166,70
9	200	4	1985,95	2471,23	2338,88	1943,27
Середнє значення ЦФ			2099,20	2231,30	2315,29	2050,48
Загальне середнє значення ЦФ			1973,22	2006,29	1932,38	1804,50

Як бачимо із таблиці 5.9, із загального середнього значення ЦФ G-алгоритм має найменше значення ЦФ, а тому він найкраще працює для запропонованого типу задач.

Побудуємо графік залежності значення ЦФ для різних алгоритмів у залежності від кількості об'єктів (рисунок 5.4) для критерію вартості маршруту без необхідності

мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення ЦФ для різних кількостей об'єктів та для кожного алгоритму.

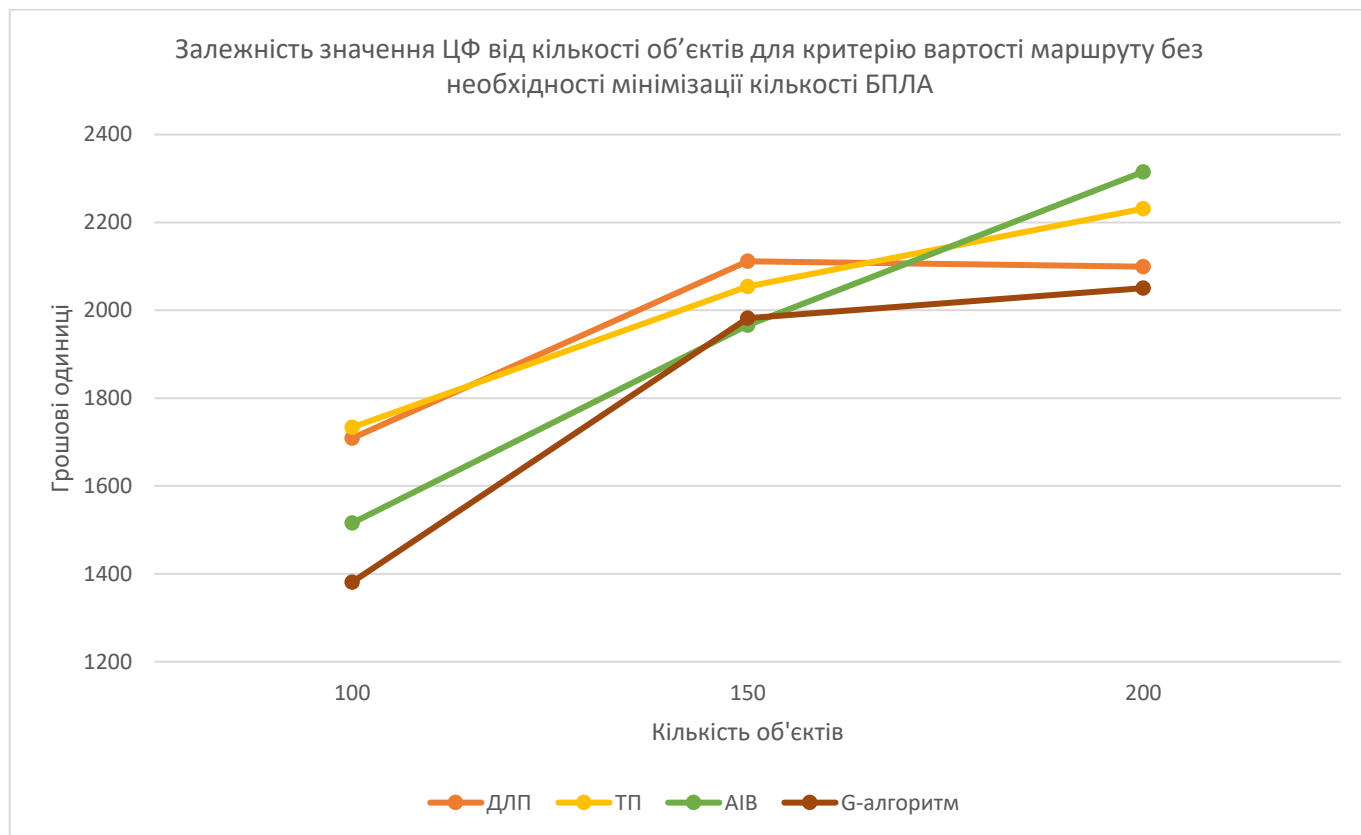


Рисунок 5.4 – Залежність значення ЦФ від кількості об'єктів для критерію вартості маршруту без необхідності мінімізації кількості БПЛА

На рисунку 5.4 бачимо графіки залежності ЦФ від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення ЦФ. Як бачимо з графіку, для різної кількості об'єктів краще працюють різні алгоритми. Так, для невеликої кількості об'єктів найкращим є G-алгоритм, а найгіршим – ТП, для середньої кількості об'єктів найкращим є АІВ, а найгіршим – ДЛП, а для великої кількості об'єктів найкращим є G-алгоритм, а найгіршим АІВ. При цьому АІВ дуже добре працює на малих та середніх даних, а на великих він дає найгірші результати.

5.3.2 Дослідження швидкодії

Проведемо дослідження значень швидкості роботи алгоритму для критерію часу виконання завдання (Критерій 1) з необхідністю мінімізації кількості БПЛА. У таблиці 5.10 наведено значення швидкості роботи алгоритму за Критерієм 1 з мінімізацією по БПЛА.

Таблиця 5.10 – Значення швидкості роботи алгоритму для Критерію 1 з мінімізацією по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	G-алгоритм
1	100	2	2,4	5,9	6,1	5,0
2	100	2	5,7	6,9	6,3	6,5
3	100	2	4,9	8,9	5,4	6,2
Середнє значення швидкості			4,3	7,2	5,9	5,9
4	150	3	6,9	12,3	7,1	10,9
5	150	3	7,7	29,3	11,1	11,4
6	150	3	5,0	13,7	11,5	14,2
Середнє значення швидкості			6,5	18,4	9,9	12,2
7	200	4	9,8	21,9	22,3	20,8
8	200	4	19,3	28,3	19,0	15,6
9	200	4	10,9	29,0	16,6	14,8
Середнє значення швидкості			13,3	26,4	19,3	17,1
Загальне середнє значення			8,1	17,4	11,7	11,7

Як бачимо із таблиці 5.10, із загального середнього значення швидкості роботи алгоритму ДЛП має найменше значення швидкості роботи алгоритму.

Для наочності отриманих результатів дослідження побудуємо графік залежності значення швидкості роботи алгоритму для різних алгоритмів у залежності від кількості об'єктів (рисунок 5.5) для критерію часу виконання завдання з необхідністю мінімізації кількості БПЛА.

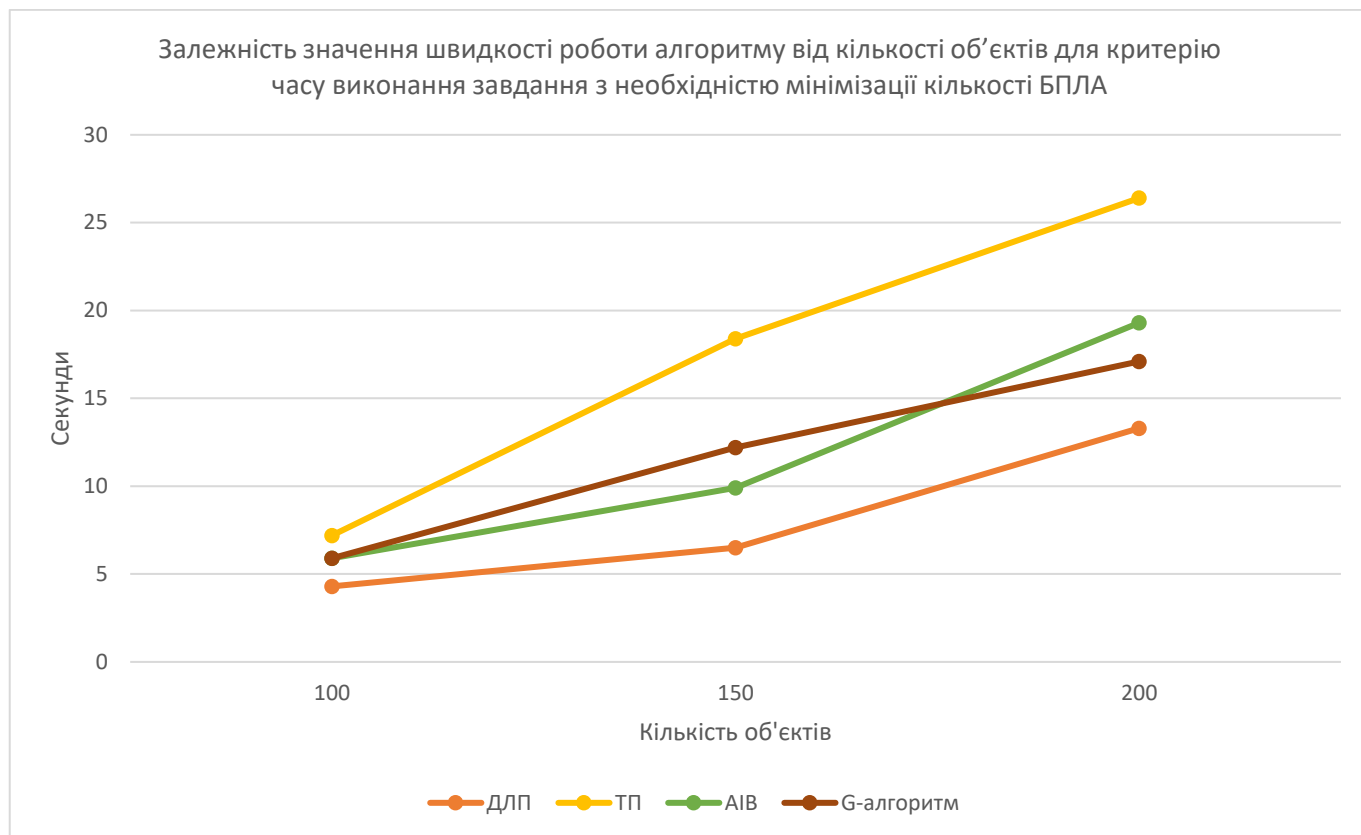


Рисунок 5.5 – Залежність значення швидкості роботи алгоритму від кількості об'єктів для критерію часу виконання завдання з необхідністю мінімізації кількості БПЛА

На рисунку 5.5 бачимо графіки залежності ЦФ від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення швидкості роботи алгоритму. Для побудови графіку обчислено середні значення швидкості роботи алгоритму для різних кількостей об'єктів та для кожного алгоритму. Як бачимо з графіку, ДЛП має найкращі результати для будь-якої кількості об'єктів, а найдовше виконує роботу ТП. На великих даних АІВ працює краще, ніж G-алгоритм, але на середніх даних – навпаки.

Проведемо дослідження значень швидкості роботи алгоритму для критерію вартості маршруту (Критерій 2) з необхідністю мінімізації кількості БПЛА. У таблиці 5.11 наведено значення швидкості роботи алгоритму за Критерієм 2 з мінімізацією по БПЛА.

Таблиця 5.11 – Значення швидкості роботи алгоритму для Критерію 2 з мінімізацією по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	AIB	G-алгоритм
1	100	2	9,0	20,3	25,2	27,8
2	100	2	4,9	16,1	17,5	23,2
3	100	2	8,4	26,3	24,1	25,1
Середнє значення швидкості			7,4	20,9	22,3	25,4
4	150	3	11,9	58,4	30,8	40,2
5	150	3	18,6	48,6	42,0	39,9
6	150	3	17,2	45,9	29,9	34,6
Середнє значення швидкості			15,9	51,0	34,2	38,2
7	200	4	54,0	60,0	50,3	53,5
8	200	4	28,1	60,0	47,4	45,1
9	200	4	47,1	60,0	60,0	60,0
Середнє значення швидкості			43,1	60,0	52,6	52,9
Загальне середнє значення			22,1	44,0	36,4	38,8

Як бачимо із таблиці 5.11, із загального середнього значення швидкості роботи алгоритму ДЛП має найменше значення швидкості роботи алгоритму.

Побудуємо графік залежності значення швидкості роботи алгоритму для різних алгоритмів у залежності від кількості об'єктів (рисунок 5.6) для критерію вартості маршруту з необхідністю мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення швидкості роботи алгоритму для різних кількостей об'єктів та для кожного алгоритму.

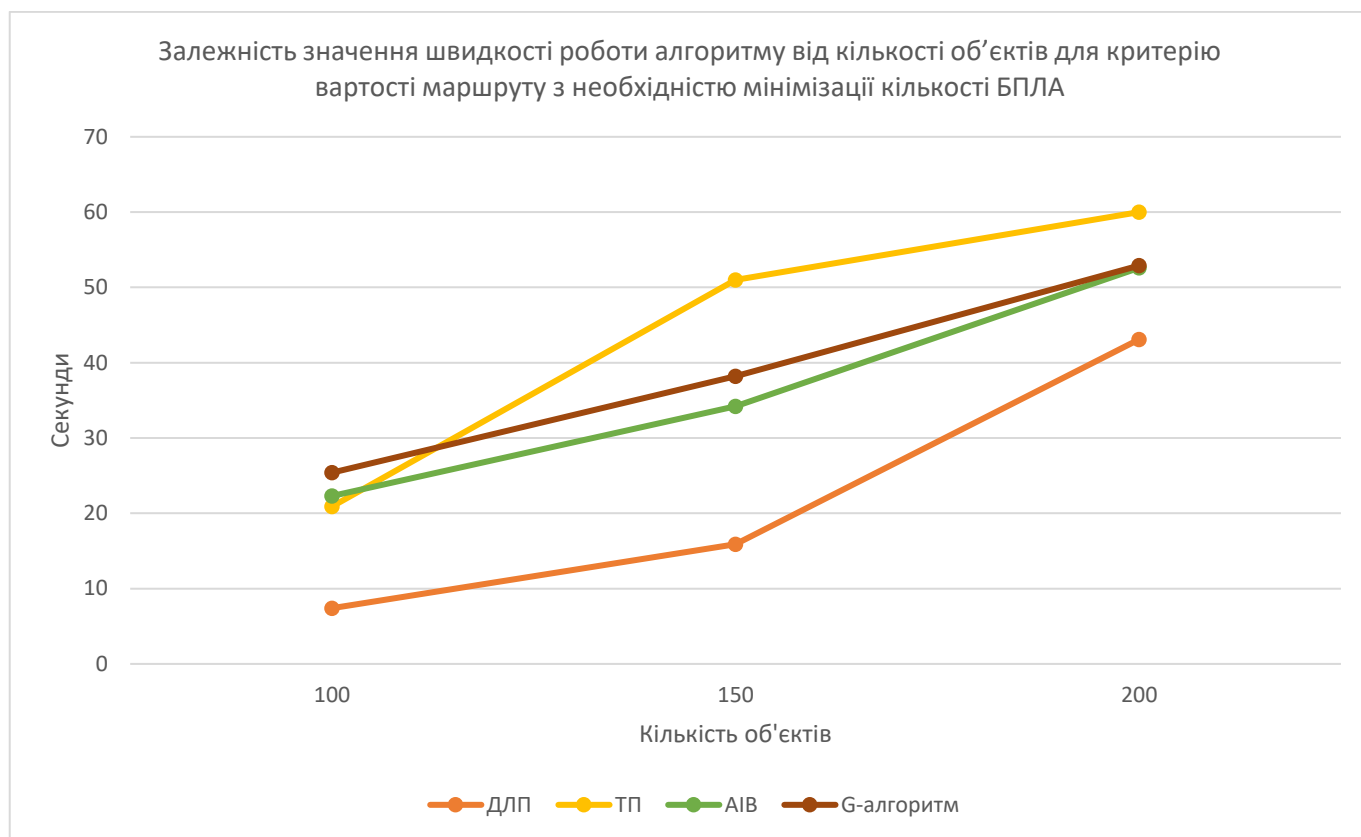


Рисунок 5.6 – Залежність значення швидкості роботи алгоритму від кількості об'єктів для критерію вартості маршруту з необхідністю мінімізації кількості БПЛА

На рисунку 5.6 бачимо графіки залежності швидкості роботи алгоритму від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення швидкості роботи алгоритму. Як бачимо з графіку, ДЛП має найкращі результати для будь-якої кількості об'єктів. Далі приблизно однаковими є АІВ та G-алгоритм. Усе ж на малих і середніх даних G-алгоритм працює повільніше. ТП працює швидше на малих даних, однак на середніх та великих він є найгіршим.

Проведемо дослідження значень швидкості роботи алгоритму для критерію часу виконання завдання (Критерій 1) без необхідності мінімізації кількості БПЛА. У

таблиці 5.12 наведено значення швидкості роботи алгоритму за Критерієм 1 без мінімізації по БПЛА.

Таблиця 5.12 – Значення швидкості роботи алгоритму для Критерію 1 без мінімізації по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	G-алгоритм
1	100	2	1,1	2,5	3,0	1,8
2	100	2	2,2	2,9	2,6	2,5
3	100	2	2,3	3,2	2,5	2,5
Середнє значення швидкості			1,9	2,9	2,7	2,3
4	150	3	2,3	4,8	2,3	3,4
5	150	3	3,8	11,2	4,8	4,8
6	150	3	2,5	6,4	4,3	6,4
Середнє значення швидкості			2,8	7,2	3,8	4,9
7	200	4	4,7	8,5	9,8	7,7
8	200	4	5,8	10,2	7,9	5,1
9	200	4	4,7	9,9	7,4	4,5
Середнє значення швидкості			5,1	9,5	8,4	5,8
Загальне середнє значення			3,3	6,6	4,9	4,3

Як бачимо із таблиці 5.12, із загального середнього значення швидкості роботи алгоритму ДЛП має найменше значення швидкості роботи алгоритму.

Побудуємо графік залежності значення швидкості роботи алгоритмів від кількості об'єктів (рисунок 5.7) для критерію часу виконання завдання без необхідності мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення швидкості роботи алгоритму для різних кількостей об'єктів та для кожного алгоритму.

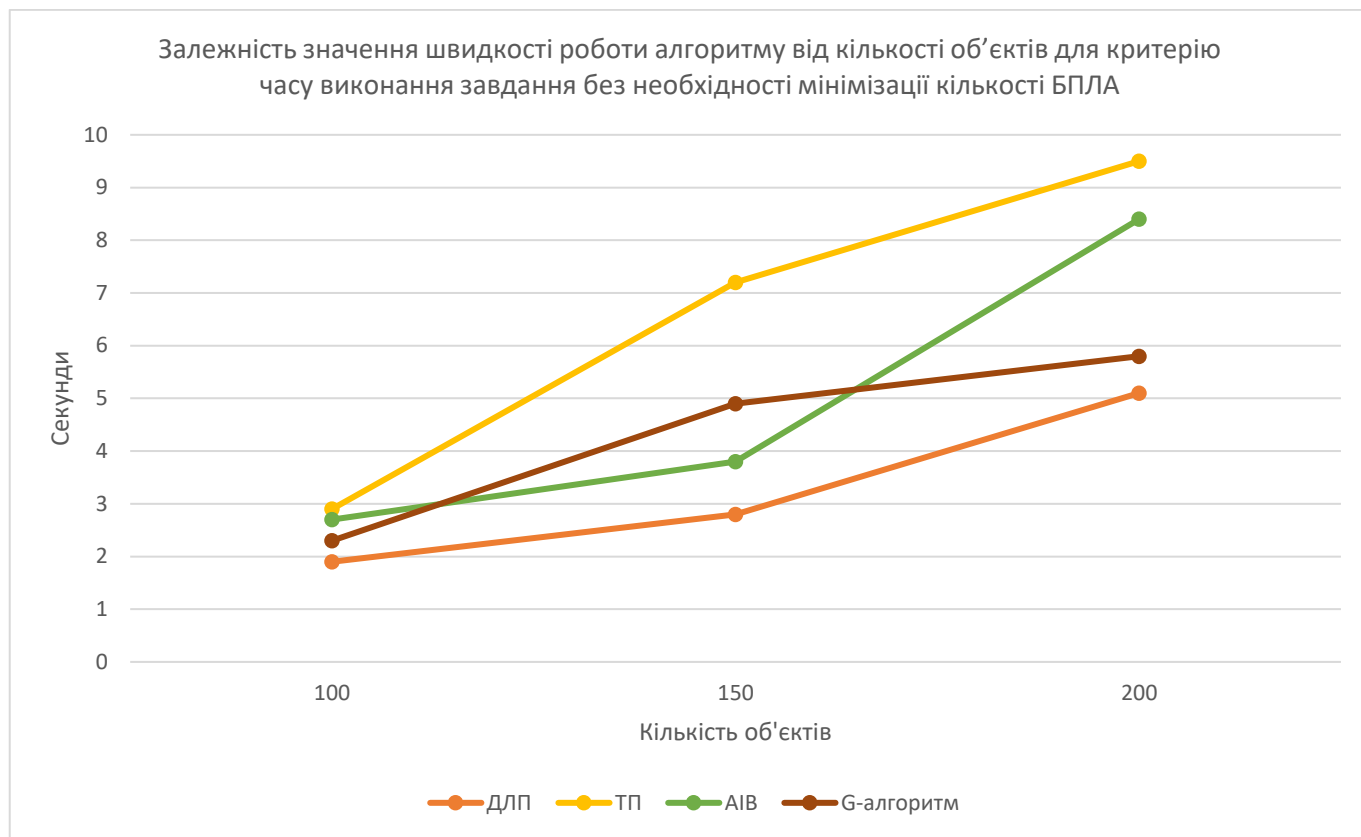


Рисунок 5.7 – Залежність значення швидкості роботи алгоритму від кількості об'єктів для критерію часу виконання завдання без необхідності мінімізації кількості БПЛА

На рисунку 5.7 бачимо графіки залежності швидкості роботи алгоритму від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення швидкості роботи алгоритму. Як бачимо з графіку, ДЛП має найкращі результати для будь-якої кількості об'єктів. Найгіршим за швидкістю є ТП. Між цими двома алгоритмами знаходяться АІВ та G-алгоритм. При цьому на малих та особливо на великих даних АІВ працює гірше за G-алгоритм, а на середніх навпаки – G-алгоритм дає результат швидше.

Проведемо дослідження значень швидкості роботи алгоритму для критерію вартості маршруту (Критерій 2) без необхідності мінімізації кількості БПЛА. У таблиці 5.13 наведено значення швидкості роботи алгоритму за Критерієм 2 без мінімізації по БПЛА.

Таблиця 5.13 – Значення швидкості роботи алгоритму для Критерію 2 без мінімізації по БПЛА

Задача	Кількість об'єктів	Кількість депо	ДЛП	ТП	АІВ	G-алгоритм
1	100	2	4,2	9,7	11,3	8,8
2	100	2	2,3	7,4	6,9	9,9
3	100	2	3,6	12,4	8,5	10,9
Середнє значення швидкості			3,4	9,8	8,9	9,9
4	150	3	4,6	23,3	11,4	12,1
5	150	3	8,1	20,7	19,6	16,7
6	150	3	7,3	21,6	13,4	16,0
Середнє значення швидкості			6,3	21,9	14,8	14,9
7	200	4	23,6	27,1	19,2	20,4
8	200	4	12,6	32,4	17,8	16,8
9	200	4	21,3	29,2	26,6	32,7
Середнє значення швидкості			19,2	29,6	21,2	23,3
Загальне середнє значення			9,7	20,4	15,0	16,0

Як бачимо із таблиці 5.13 із загального середнього значення швидкості роботи алгоритму, ДЛП має найменше значення швидкості роботи алгоритму.

Побудуємо графік залежності значення швидкості роботи алгоритму для різних алгоритмів в залежності від кількості об'єктів (рисунок 5.8) для критерію вартості маршруту без необхідності мінімізації кількості БПЛА. Для побудови графіку обчислено середні значення швидкості роботи алгоритму для різних кількостей об'єктів та для кожного алгоритму.

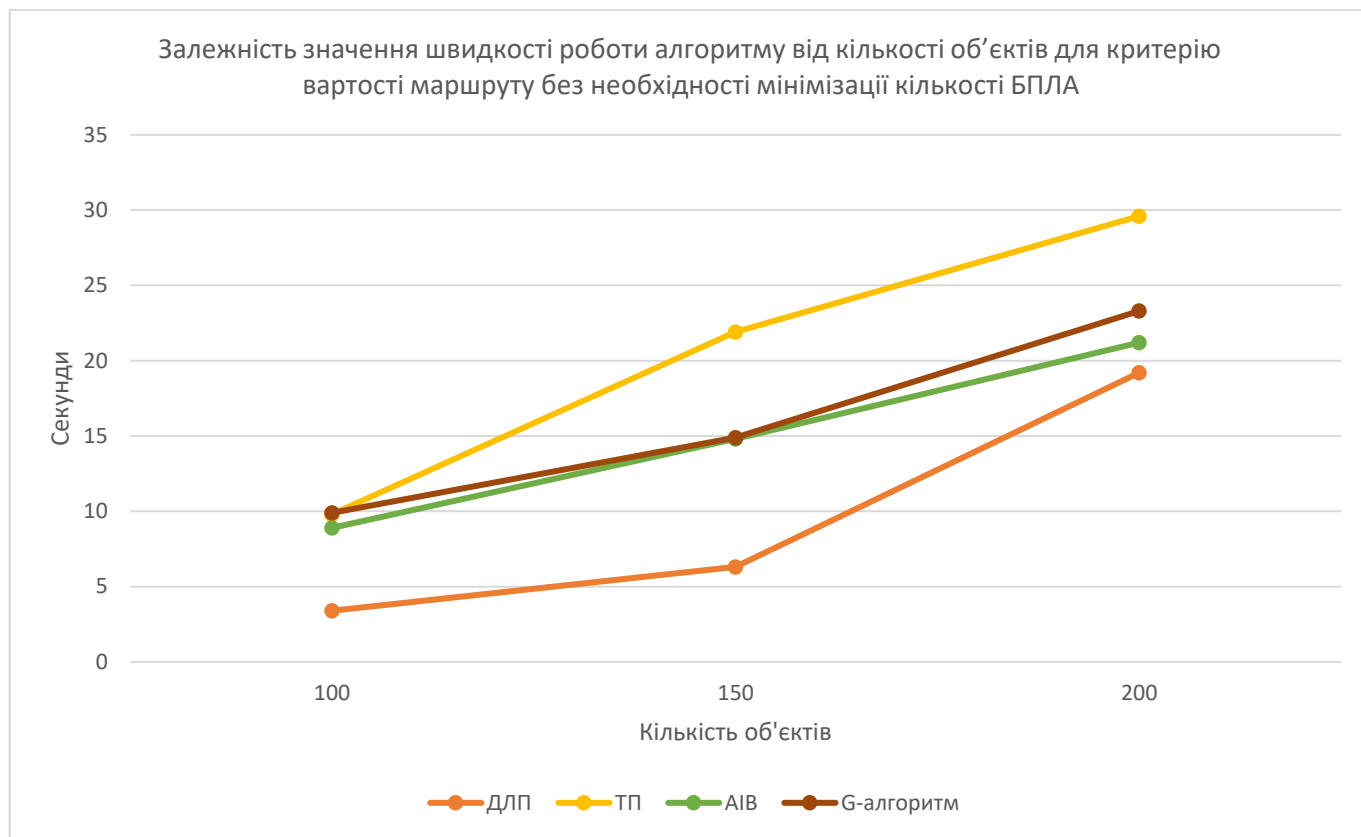


Рисунок 5.8 – Залежність значення швидкості роботи алгоритму від кількості об'єктів для критерію вартості маршруту без необхідності мінімізації кількості БПЛА

На рисунку 5.8 бачимо графіки залежності швидкості роботи алгоритму від кількості об'єктів для різних алгоритмів. Оранжевим кольором тут позначено ДЛП, жовтим – ТП, зеленим – АІВ, коричневим – G-алгоритм. На осі абсцис позначено кількість об'єктів, для яких розв'язано задачу, а на осі ординат – значення ЦФ. Як бачимо з графіку, найкращий результат для будь-яких даних дає ДЛП. На другому місці знаходиться АІВ, на третьому – G-алгоритм, а на останньому – ТП.

Висновки до розділу

У п'ятому розділі розглянуто дослідження ефективності розроблених методів, а саме: описано умови проведення досліджень, досліджено вплив параметрів на роботу алгоритмів та порівняно результати роботи алгоритмів.

Для проведення експериментів згенеровано 9 задач: 3 задачі на 2 депо та 100 об'єктів; 3 задачі на 3 депо та 150 об'єктів; 3 задачі на 4 депо та 200 об'єктів. При порівнянні ефективності всіх алгоритмів використовувалося усереднене значення ЦФ та часу виконання алгоритму за 10 запусків. Для всіх алгоритмів максимальний час роботи обмежений та склав 60 с. Для підбору параметрів алгоритмів застосовано локальний пошук. Експерименти проводилися окремо для різних критеріїв: час виконання завдання та вартість маршруту.

Для різних алгоритмів є різні параметри. Наприклад, для ДЛП параметрів немає взагалі, а для ТП, АІВ та G-алгоритму вони існують. Для ТП є два параметри: довжина списку заборон та максимальна кількість ітерацій без покращення результату. Для АІВ є п'ять параметрів: початкова температура, тривалість прогону, коефіцієнт зниження температури, поріг досягнення рівноваги та максимальна кількість ітерацій без покращення результату. Для G-алгоритму є чотири параметри: довжина прогону, коефіцієнт b для G -функції, поріг досягнення рівноваги та максимальна кількість ітерацій без покращення результату. У результаті налаштування параметрів за допомогою алгоритму локального пошуку визначено, що значення ЦФ зменшується. Після налаштування параметрів алгоритму результат покращився в середньому на 3,8% для Критерію 1 та на 2,1% для Критерію 2.

Роботу алгоритмів оцінено за двома критеріями оцінки алгоритмів: якість кінцевого розв'язку та час роботи алгоритму. Кожен алгоритм перевірено за цими критеріями. При дослідженні кінцевих результатів визначено, що G-алгоритм має найменше значення ЦФ для всіх критеріїв та є досить швидким. При дослідженні швидкодії визначено, що ДЛП працює найшвидше, усе ж він найгірший за точністю.

6 РОЗРОБКА СТАРТАП-ПРОЄКТУ

6.1 Опис ідеї проєкту

Основною ідеєю проєкту є надання програмного забезпечення для розв’язування задач маршрутизації БПЛА за наявності декількох депо. Узагальнений опис ідей стартап-проєкту наведено в таблиці 6.1.

Таблиця 6.1 – Опис ідеї проєкту

Зміст ідеї	Напрямки застосування
Надання програмного забезпечення для розв’язування задач маршрутизації безпілотних літальних апаратів за наявності декількох депо	1. При необхідності мінімізації часу обстеження об’єктів, якщо роботу потрібно виконати якнайшвидше.
	2. При необхідності мінімізації затрат грошових засобів на обслуговування БПЛА при обстеженні об’єктів, наприклад, для бензинових БПЛА.

Для того, щоб визначити потенційні техніко-економічні переваги ідеї порівняно з пропозиціями конкурентів, необхідно сформувати перелік техніко-економічних властивостей та характеристик ідей. До цього переліку входять:

- вартість;
- швидкодія;
- зручність інтерфейсу;
- складність у експлуатації;
- можливість роботи без інтернету;
- надійність;
- унікальність алгоритму;
- документованість.

Розглянемо аналоги, з якими далі будемо проводити порівняння. Існує велика кількість програмного забезпечення для побудови маршрутів ТЗ.

1) ABM Rinkai TMS. Це програма планування оптимальних маршрутів доставки для сегментів B2B, B2C, транспортних компаній тощо.

Особливості та переваги:

- система дозволяє зберігати інформацію про адреси клієнтів, завантажених із системи 1С; їх можна побачити на карті та відредагувати, якщо це необхідно;
- для кожного автомобіля чи співробітника, який здійснює доставку товарів, можна вказати ідентифікатор трекера чи мобільного застосунку для отримання даних GPS по фактичному пересуванні;
- у єдиному вікні планування відображається інформація про роботу складу, доступні автомобілі та заплановані замовлення. У результаті складання оптимальних маршрутів система видає до 10 оптимальних маршрутів, відсортованих за сумою розходів;
- система надає можливість задавати назву продукції, яку необхідно доставити, наприклад, пляшки, ящики тощо, щоб враховувати її об'єм та масу з метою розрахунку часу її завантаження на ТЗ та розвантаження.

2) TSM. Це хмарна система управління транспортом, що забезпечує автоматичне планування маршрутів, оцінку рентабельності точок доставки, GPS контроль маршрутів руху та аналітичну звітність.

Особливості та переваги:

- при плануванні маршрутів система враховує трафік доріг;
- під час розрахунку маршрутів доставки враховуються габарити та вантажопідйомність автомобіля, тип замовлених товарів, їх вага та габарити, часові обмеження на доставку;
- система оптимізує маршрути за ціною та часом;
- система надає 7 різних рівнів доступу: адміністратор, логіст, водій, спостерігач, контролер, диспетчер, аналітик;
- система пропонує переглядати точки доставки та маршрути на карті.

3) **Max Plus.** Це програма для автоматизації транспортної логістики, планування оптимальних маршрутів та проведення план-фактового аналізу.

Особливості та переваги:

- програмний комплекс може бути адаптований під індивідуальні вимоги кожного клієнта;
- система планує маршрути доставки товарів з одного чи декількох складів на множину торгових точок з мінімальними затратами;
- можливо здійснювати планування на основі не лише свого автопарку, а й із залученням додаткових;
- система виявляє торгові точки, доставка товарів у які є не вигідною;
- система визначає оптимальний вибір місць, де можливо перезавантажити ТЗ для подальшої доставки товарів.

4) **Drone UA.** Це міжнародний системний інтегратор безпілотних рішень, що вирішує питання, пов'язані з безпілотниками. У структурі компанії функціонують власні інженерні та виробничі підрозділи, відкритий центр з обробки даних. Ця програма не надає можливості побудови маршрутів, а лише здійснює аналіз отриманих результатів, наприклад, знімків, на основі виконання місій БПЛА.

Особливості та переваги:

- програма корисна для сільського господарства, оперативної картографії, телекомунікацій, енергетики, нафтової промисловості;
- програма пропонує зчитування та обробку даних із дронів.

У таблиці 6.2 наведено результати аналізу характеристик ідей проєкту.

Таблиця 6.2 – Характеристики ідей проєкту

№ п/п	Техніко-економічні характеристики ідей	Наш проєкт	ABM Rinkai TMS	TSM	Max Plus	Drone UA
1	Вартість, тис грн	–	10	12	18	4
2	Швидкодія	Середня	Висока	Висока	Висока	Середня

Кінець таблиці 6.2

№ п/п	Техніко- економічні характеристики ідеї	Наш проект	ABM Rinkai TMS	TSM	Max Plus	Drone UA
3	Зручність інтерфейсу	Висока	Висока	Висока	Висока	Висока
4	Складність експлуатації	Інтуїтивно зрозуміло	Інтуїтивно зрозуміло	Високий поріг входження	Високий поріг входження	Інтуїтивно зрозуміло
5	Можливість роботи без інтернету	Наявна	Наявна	Відсутня	Відсутня	Відсутня
6	Надійність	Висока	Невідомо	Невідомо	Невідомо	Невідомо
7	Унікальність алгоритму	Так	Невідомо	Невідомо	Невідомо	Невідомо
8	Документованість	Часткова	Часткова	Повна	Часткова	Повна

У таблиці 6.3 наведено визначення сильних, слабких та нейтральних характеристик ідей проекту.

Таблиця 6.3 – Сильні, слабкі та нейтральні характеристики ідей проекту

№ п/п	Техніко-економічні характеристики ідеї	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1	Вартість, тис грн			+
2	Швидкодія		+	
3	Зручність інтерфейсу			+
4	Складність експлуатації			+
5	Можливість роботи без інтернету			+
6	Надійність			+
7	Унікальність алгоритму			+
8	Документованість	+		

Як показало дослідження, нині на ринку існує чимало програм, які забезпечують побудову оптимальних маршрутів для наземного транспорту. Усе ж готових інформаційних систем, які можуть будувати оптимальні маршрути для БПЛА не виявлено. існують програми, за допомогою яких можливо задавати необхідні точки

обльоту безпосередньо для БПЛА, але які саме точки мають входити в маршрут та в якому порядку – це вже окрема задача, розв’язуванням якої займається наш програмний продукт.

6.2 Технологічний аудит ідеї проєкту

Проведемо аудит технології, за допомогою якої можна реалізувати ідею проєкту. Визначення технологічної здійсненності проєкту передбачає аналіз таких складових:

- за якою технологією буде виготовлено товар згідно з ідеєю проєкту;
- чи існують такі технології, чи їх потрібно доробити/розробити;
- чи доступні такі технології авторам проєкту.

Аналіз складових для визначення технологічної здійсненності проєкту наведено у таблиці 6.4.

Таблиця 6.4 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології реалізації	Наявність технології	Доступність технології
1	Візуалізація маршруту	Бібліотека WebGL	Наявна	У вільному доступі
2	Зберігання маршруту	Бібліотека Json	Необхідно розробити	У вільному доступі
3	Використання алгоритмів побудови оптимальних маршрутів	ДЛП, ТП, АІВ, G-алгоритм	Необхідно розробити	Доступна

Оскільки всі технології для реалізації ідей доступні, то можливо реалізувати всі три ідеї. Дві з чотирьох технологій необхідно розробити, але є розуміння, які сімейства алгоритмів треба взяти за основу для створення власних.

6.3 Аналіз ринкових можливостей запуску стартап-проєкту

Наведемо попередню характеристику потенційного ринку стартап-проєкту (таблиця 6.5).

Таблиця 6.5 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум. од.	Невідомий
3	Динаміка ринку (якісна оцінка)	Зростає
4	Специфічні вимоги до стандартизації та сертифікації	Юридичні
5	Середня норма рентабельності в галузі, %	52%

Ринок програмного забезпечення для оптимізації маршрутів БПЛА за наявності декількох депо є достатньо привабливим, оскільки використовується в різних сферах діяльності. Тому існує потреба у створенні програмного продукту для оптимізації маршрутів. Далі визначимо потенційні групи клієнтів, їх характеристики та сформуємо орієнтовний перелік вимог до споживачів ринку. Результати наведено в таблиці 6.6.

Таблиця 6.6 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності в поведінці різних цільових груп клієнтів	Вимоги споживачів до товару
1	Пошук оптимальних маршрутів для групи БПЛА	Керівники виробництва	Пошук автоматизації побудови маршрутів	Безпека даних, інтуїтивна зрозумілість, простота використання

Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища: складемо таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 6.7 – 6.8).

Таблиця 6.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Нестабільність ринку	Нестабільна політична ситуація, війна, низький рівень економік	Вихід на європейський ринок
2	Швидкість розвитку технологій	Поява нових технологій	Періодичне впровадження нових технологій
3	Зростання конкуренції	Поява нових конкурентів	Створення оригінального функціоналу

Таблиця 6.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Інвестиції	Зацікавленість інших осіб у стартапі	Виконати вимоги інвесторів
2	Запровадження унікальної ідеї	Стати універсальним застосуванням для різних підприємств	Реалізувати функціонал і стати першими на ринку
3	Розвиток ідеї	Запроваджувати нові технології, які зацікавлять користувачів та інвесторів	Розвивати далі ідею

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку. Результати аналізу наведено в таблиці 6.9.

Таблиця 6.9 – Ступеневий аналіз конкуренції на ринку

№ п/п	Особливості конкурентного середовища	У чому проявляється характеристика	Вплив на діяльність підприємства
1	Тип конкуренції – чиста	Існують схожі програмні засоби, які широко направлені та досить складні у налаштуванні	Запропонувати новий розв'язок, якого немає на ринку
2	За рівнем конкурентної боротьби – національний	Багато фірм розробляють програми покращення формування карт розкредиту	Охопити ринок України
3	За галузевою ознакою – міжгалузева	Галузь ІТ	Впроваджувати нові технології
4	Конкуренція за видами товарів – товарно-видова	Існують схожі програми, але для вузьких сфер	Аналізувати та впроваджувати бажання клієнтів

Кінець таблиці 6.9

№ п/п	Особливості конкурентного середовища	У чому проявляється характеристика	Вплив на діяльність підприємства
5	За характером конкурентних переваг – нецінова	Програмне забезпечення буде суттєво відрізнятися від звичайних програм створення карт розкрою	Підтримувати сучасний та зручний інтерфейс, впроваджувати новий функціонал
6	За інтенсивністю – не марочна	Фірма нова і невідома в Україні	Стати брендом

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі (таблиця 6.10).

Таблиця 6.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	ABM Rinkai TMS, TMS, Max Plus	Патенти на продукти	Відсутні	Наявність документації про продукт, контроль швидкодії	Є часткова заміна
Висновки			Відсутні	Встановлюються вимоги до швидкодії	Частково перекривають функціонал програмного продукту за рахунок складних налаштувань, що не є основним функціоналом

У результаті аналізу таблиці 6.10 робимо висновок, що проєкт є конкурентоспроможним, незважаючи на існування аналогів. Сильними сторонами проєкту є вартість, зручність інтерфейсу, складність експлуатації, можливість роботи без інтернету, надійність та унікальність алгоритмів.

На основі аналізу конкуренції, а також із урахуванням характеристик ідеї проєкту, вимог споживачів до товару та факторів маркетингового середовища визначимо перелік факторів конкурентоспроможності та наведемо їх у таблиці 6.11.

Таблиця 6.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор	Обґрунтування
1	Динаміка галузі	Кожного дня з'являються нові технології та стрімкий розвиток конкурентів
2	Постійні витрати	Необхідні постійні витрати на сервера, підтримка програмного продукту
3	Наявність торгівельних знаків	Мати унікальну і відому
4	Рівень автоматизації	Чим більший рівень автоматизації, тим менше затрат на підтримку та роботу

За визначеними факторами проведемо аналіз сильних та слабких сторін стартап-проєкту. Результати наведено в таблиці 6.12.

Таблиця 6.12 – Порівняльний аналіз сильних та слабких сторін нашого проєкту

№ п/п	Фактор конкурентоспроможності	Бали	Рейтинг конкурентів						
			-3	-2	-1	0	1	2	3
1	Динаміка галузі	16		+					
2	Постійні витрати	19					+		
3	Наявність торгівельних знаків	17						+	
4	Рівень автоматизації	13				+			

Проведемо оцінку сильних і слабких сторін компанії за допомогою SWOT-аналізу. SWOT – аббревіатура:

S – strengths (сильні сторони);

W – weaknesses (слабкі сторони);

O – opportunities (можливості);

T – threats (загрози).

У готовій матриці необхідно поставити «+» біля факторів у списках сильних сторін та «-» біля факторів, що є загрозою для бізнесу. Так само необхідно оцінити

список слабких сторін та можливостей. У таблиці 6.13 наведено SWOT-аналіз стартап-проєкту.

Таблиця 6.13 – SWOT-аналіз стартап-проєкту

Сильні сторони	Оцінка	Можливості	Оцінка
Низька ціна	+	Зацікавлення представників різних сфер діяльності	+
Доступний та інтуїтивний інтерфейс	+	Поява нових партнерів	+
Задоволення потреб компанії	+	Вихід на ринки регіону	+
Слабкі сторони	Оцінка	Загрози	Оцінка
Мало ресурсів та часу для створення дійсно якісного продукту	–	Задоволення потреб клієнтів	+
Малий бюджет на маркетинг	–	Агресивні дії великих конкурентів у галузі	–

Розробимо альтернативи ринкової поведінки для виведення стартап-проєкту на ринок та орієнтовний оптимальний час їх ринкової організації з огляду на потенційні проєкти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів. Результати наведено в таблиці 6.14.

Таблиця 6.14 – Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива	Ймовірність отримання результатів	Строки реалізації
1	Стратегія спеціалізації	90%	1,5 роки
2	Стратегія диференціації	50%	2 роки
3	Позиціонування за співвідношенням «ціна-якість»	93%	1 рік
4	Стратегія зайняття конкурентної ніші	60%	1 рік

Найкращим варіантом буде стратегія позиціонування співвідношенням «ціна-якість», оскільки ймовірність отримання результатів та строки реалізації є найкращими.

6.4 Розроблення ринкової стратегії проєкту

Для початку необхідно визначити цільові групи потенційних споживачів. Їх опис наведено в таблиці 6.15.

Таблиця 6.15 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	Військові логісти	90%	100%	Середня	Середня
2	Цивільні логісти	80%	100%	Висока	Низька

Проєкт направлений в основному на цивільну сферу, але тут є різні галузі, де можна його застосовувати. Тому доцільно використовувати стратегію масового маркетингу. Сформуємо базову стратегію розвитку для роботи в обраних сегментах ринку та наведемо результати в таблиці 6.16.

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Зосередження на цивільній сфері	Стратегія концентрованого маркетингу	Високий рівень автоматизації, швидкодія	Стратегія спеціалізації

Далі визначимо базову стратегію конкурентної поведінки (таблиця 6.17).

Таблиця 6.17 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента?	Стратегія конкурентної поведінки
1	Так (на ринку України)	Так	Ні	Зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до стартап-компанії та продукту та в залежності від обраної базової стратегії розвитку й конкурентної поведінки розробимо стратегію позиціонування, що наведена в таблиці 6.18.

Таблиця 6.18 – Стратегія позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувану комплексну позицію власного проєкту
1	Якість, зручний інтерфейс, простота у використанні, новий функціонал: формування розміщення різними алгоритмам	Задоволення потреб вибраного цільового сегменту краще, ніж конкуренти	Оптимальне співвідношення ціна-якість проєкту	Зручність інтерфейсу, високий рівень автоматизації, задоволення потреб клієнтів

Для проєкту необхідна стратегія концентрованого маркетингу. Як базову стратегію розвитку обрано задоволення потреб вибраного цільового сегменту краще, ніж конкуренти. У результаті дотримання обраних стратегій у компанії буде можливість стати лідером на ринку.

6.5 Розроблення маркетингової програми стартап-проєкту

Сформуємо маркетингову концепцію продукту, який отримає споживач. Для цього необхідно підсумувати результати попередніх аналізів конкурентоспроможності товару. У таблиці 6.19 визначено ключові переваги концепції потенційного товару.

Таблиця 6.19 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Пошук оптимальних маршрутів для групи БПЛА	Зменшення витрат на обслуговування бензинових БПЛА, зменшення витрат часу на обстеження об'єктів	Співвідношення якість-ціна

Наведемо трьохрівневу маркетингову модель товару в таблиці 6.20.

Таблиця 6.20 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система побудови оптимальних маршрутів БПЛА за наявності декількох депо		
II. Товар у реальному виконанні	Властивості та характеристики	М/Нм	Вр/Тх/ Тл/Е/Щр
	1. Якість	—	—
	2. Простота у використанні		
	3. Видача оптимальних розкладів маршрутів		
	4. Низька ціна		
	Якість: згідно зі стандартом IEEE 829-2008 буде проведено тестування		
Пакування нема			
Марка (власна): Your UAV			
III. Товар із підкріпленням	До продажу: тріал версія з обмеженням розмірності задачі		
	Після продажу: підтримка та додавання нового функціоналу відповідно до замовлень		

Далі необхідно визначити межі встановлення ціни (таблиця 6.21).

Таблиця 6.21 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	\$500	\$5000	\$2,5 млрд	\$500 – \$5000

Далі необхідно визначити оптимальний канал збуту (таблиця 6.22).

Таблиця 6.22 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Одна одиниця на підприємство	Домовлятися з керівниками підприємства про вимоги та ціну	Перший рівень	Власні сили та через посередників

Останнім етапом є формування маркетингової комунікації (таблиця 6.23).

Таблиця 6.23 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій	Ключові позиції	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Орієнтовані на зручність та якість	Контент-маркетинг, реклама	Простота використання, ціна, новизна та якість	Показати переваги продукту та заохотити користуватися ним	Показати зручність продукції

Результатом підрозділу є маркетингова програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення. Щоб заохотити споживачів користуватися продуктом необхідно задіяти рекламу, основною метою якої буде звернення уваги на переваги продукту.

Висновки до розділу

У шостому розділі проведено аналіз програмного продукту в якості стартап-проєкту. У результаті проведення аналізу конкурентного середовища визначено, що існує потреба у створенні даного програмного продукту, оскільки на нього наявний попит, а проєкт є рентабельним. У результаті аналізу ринкових можливостей запуску визначено, що на ринку наявна конкуренція, оскільки існують фірми конкуренти, тому вхід на ринок не буде легким. Усе ж великим плюсом є те, що за рахунок того, що проєкт є програмним, його розробка не потребує додаткових витрат на різноманітне обладнання та матеріали.

У ході розробки ринкової стратегії виявлено, що проєкт має багато сильних сторін, які переважають над існуючими аналогами. Для проєкту необхідна стратегія концентрованого маркетингу. Як базову стратегію розвитку обрано задоволення потреб вибраного цільового сегменту краще, ніж конкуренти. У результаті дотримання обраних стратегій у компанії буде можливість стати лідером на ринку.

Розроблення маркетингової стратегії показала, що для заохочення споживачів користуватися продуктом необхідно задіяти рекламу, основною метою якої буде звернення уваги на переваги продукту.

ВИСНОВКИ

Проведено огляд ЗМТЗ, здійснено їх класифікацію. Розглянуто найбільш поширені класи задач маршрутизації БПЛА та проведено їх класифікацію. Запропоновано математичну постановку задачі маршрутизації БПЛА за наявності декількох депо. Визначено два класи задач та два критерії для їх розв'язування, у результаті чого сформульовано дві ЦФ.

Обґрунтовано методи розв'язування та розроблено нові алгоритми розв'язування задачі маршрутизації БПЛА за наявності декількох депо, а саме:

- для формування початкового наближення: метод найдешевшого включення та гібридний алгоритм найдешевшого включення з проміжними покращеннями;

- для покращення початкового наближення: ДЛП, ТП, АІВ та G-алгоритм.

Розроблено програмне забезпечення для досліджуваної проблеми. Виконано налаштування параметрів алгоритмів та проведено дослідження ефективності розроблених алгоритмів.

Роботу алгоритмів оцінено за двома критеріями оцінки: якістю кінцевого розв'язку та часом роботи алгоритму. При дослідженні кінцевих результатів визначено, що G-алгоритм знаходить найменше значення ЦФ для всіх критеріїв та є досить швидким. При дослідженні швидкодії визначено, що ДЛП працює найшвидше, усе ж він найгірший за точністю.

За матеріалами дисертації опубліковано 2 наукові роботи: 1 стаття в журналі категорії Б та 1 тези.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гуляницький Л. Ф., Коткова А. А. До класифікації задач маршрутизації транспортних засобів. Науковий вісник Ужгородського університету. 2020. № 1 (36). С. 73–84.
2. Dantzig G. B., Ramser J. H. The truck dispatching problem. Management Science. 1959. Vol. 6, No 1. P. 80–91.
3. Radzki G., Thibbotuwawa, A., Bocewicz, G. UAVs flight routes optimization in changing weather conditions – Constraint programming approach. Applied Computer Science. 2019. Vol. 15, No 3. P. 5–17.
4. Гуляницький Л. Ф., Мулеса О. Ю. Прикладні методи комбінаторної оптимізації: навч. посіб. Київ : Видавничо-поліграфічний центр «Київський університет», 2016. 146 с.
5. Uchoa E., Pecin D., Pessoa A. New benchmark instances for the Capacitated Vehicle Routing Problem. European Journal of Operational Research. 2017. Vol. 257, No 3. P. 845–858.
6. Avci M., Topaloglu S. An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. Computers & Industrial Engineering. 2015. Vol. 83. P. 15–29.
7. Yao B., Yu B., Hu P. An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. Annals of Operations Research. 2016. Vol. 242, No 2. P. 303–320.
8. Berghida M., Boukra A. Quantum Inspired Algorithm for a VRP with Heterogeneous Fleet Mixed Backhauls and Time Windows. International Journal of Applied Metaheuristic Computing. 2016. Vol. 7, No 4. P. 18–38.
9. Koc C., Laporte G. Vehicle routing with backhauls: Review and research perspectives. Computers & Operations Research. 2018. Vol. 91. P. 79–91.

10. Reil S., Bortfeldt A., Monch L. Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints. *European Journal of Operational Research*. 2018. Vol. 266, No 3. P. 877–894.
11. Bruglieri M., Pezzella F., Pisacane O. A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows. *Electronic Notes in Discrete Mathematics*. 2015. Vol. 47. P. 221–228.
12. Kucukoglu I., Ozturk N. An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. *Computers & Industrial Engineering*. 2015. Vol. 86. P. 60–68.
13. Bin Y., Zhi-Hua H. Routing with time-windows for multiple environmental vehicle types. *Computers & Industrial Engineering*. 2015. Vol. 89. P. 150–161.
14. Cantu-Funes R., Angelica Salazar-Aguilar M., Boyer V. Multi-depot periodic vehicle routing problem with due dates and time windows. *Journal of the Operational Research Society*. 2016. Vol. 69. P. 296–306.
15. Montoya-Torres J., Franco J., Isaza S. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*. 2015. Vol. 79. P. 115–129.
16. De Oliveira F., Enayatifar R., Sadaei H. A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem. *Expert Systems with Applications*. 2016. Vol. 43. P. 117–130.
17. Lalla-Ruiz E., Exposito-Izquierdo Ch., Taheripour Sh. An improved formulation for the multi-depot open vehicle routing problem. *OR Spectrum*. 2016. Vol. 38, No 1. P. 175–187.
18. Leungsubthawee K., Saranwong S., Likasiri Ch. Multiple Depot Vehicle Routing Problems on Clustering Algorithms. *Thai Journal of Mathematics*. 2018. P. 205–216.

19. Ramos T., Gomes M., Pova A. Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*. 2019. P. 1–18.
20. Chávez J., Escobar J. A Multi-objective Pareto and Colony Algorithm for the Multi-depot Vehicle Routing Problem with Backhauls. *International Journal of Industrial Engineering Computations*. 2016. Vol. 7. P. 35–48.
21. Zhang Y., Qi M., Mia L., Wu G. A Generalized Multi-Depot Vehicle Routing Problem with Replenishment Based on LocalSolver. *International Journal of Industrial Engineering Computations*. 2015. Vol. 6. P. 81–98.
22. Braekers K., Ramaekers K., Nieuwenh I. The Vehicle Routing Problem: State of the Art Classification and Review. *Computers & Industrial Engineering*. 2016. Vol. 99. P. 300–313.
23. Markov I., Varone S., Bierlaire M. Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities. *Transportation Research Part B: Methodological*. 2016. Vol. 84. P. 256–273.
24. Ritzinger U., Puchinger J. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*. 2015. Vol. 54. P. 215–231.
25. Sarasola B., Doerner K., Schmid V. Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*. 2016. Vol. 236, No 2. P. 425–461.
26. Hernandez F., Gendreau M., Potvin J. Heuristics for tactical time slot management: a periodic vehicle routing problem view. *International Transactions in Operational Research*. 2017. Vol. 24, No 6. P. 1233–1252.
27. Lee J., Kim B. Industrial ship routing problem with split delivery and two types of vessels. *Expert Systems with Applications*. 2015. Vol. 42, No 22. P. 9012–9023.

28. Chu J., Yan S., Huang H. A Multi-Trip Split-Delivery Vehicle Routing Problem with Time Windows for Inventory Replenishment Under Stochastic Travel Times. *Networks and Spatial Economics*. 2017. Vol. 17, No 1. P. 41–68.
29. Горбулін В. П., Гуляницький Л. Ф., Сергієнко І. В. Постановки та математичні моделі проблем оптимізації маршрутів літальних апаратів із динамічними депо. *Управляючі системи й машини*. 2019. № 1. С. 3–10.
30. Гуляницький Л. Ф., Сторчевий В. В. Одна спеціальна задача маршрутизації БПЛА. *Науковий вісник Ужгородського університету. Серія математика і інформатика*. 2019. № 34 (1). С. 60–78.
31. Міночкін А. І., Сова О. Я. Аналіз використання безпілотних літальних апаратів у якості ретрансляторів тактичних мобільних радіомереж. *Збірник наукових праць ВІТІ*. 2017. № 1. С. 61–70.
32. Yu K., Budhiraja A. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Field robotics*. 2019. Vol. 36, No 3. P. 602–616.
33. Dae-Sung J., Hyeok-Joo Ch., Han-Lim Ch. Optimal control-based UAV path planning with dynamically-constrained TSP with neighborhoods. 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 18-21 October 2017. Jeju, 2017. P. 373–378.
34. Schwarzrock J., Zacarias I. Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence. *Engineering Applications of Artificial Intelligence*. 2019. Vol. 72. P. 10–20.
35. Yilmaz B., Sueda N. Multi UAV Based Traffic Control in Smart Cities. 1th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 1-3 July 2020. P. 1–7.
36. Zhen I. A vehicle routing problem arising in unmanned aerial monitoring. *Computers & Operations Research*. 2019. Vol. 105. P. 1–11.

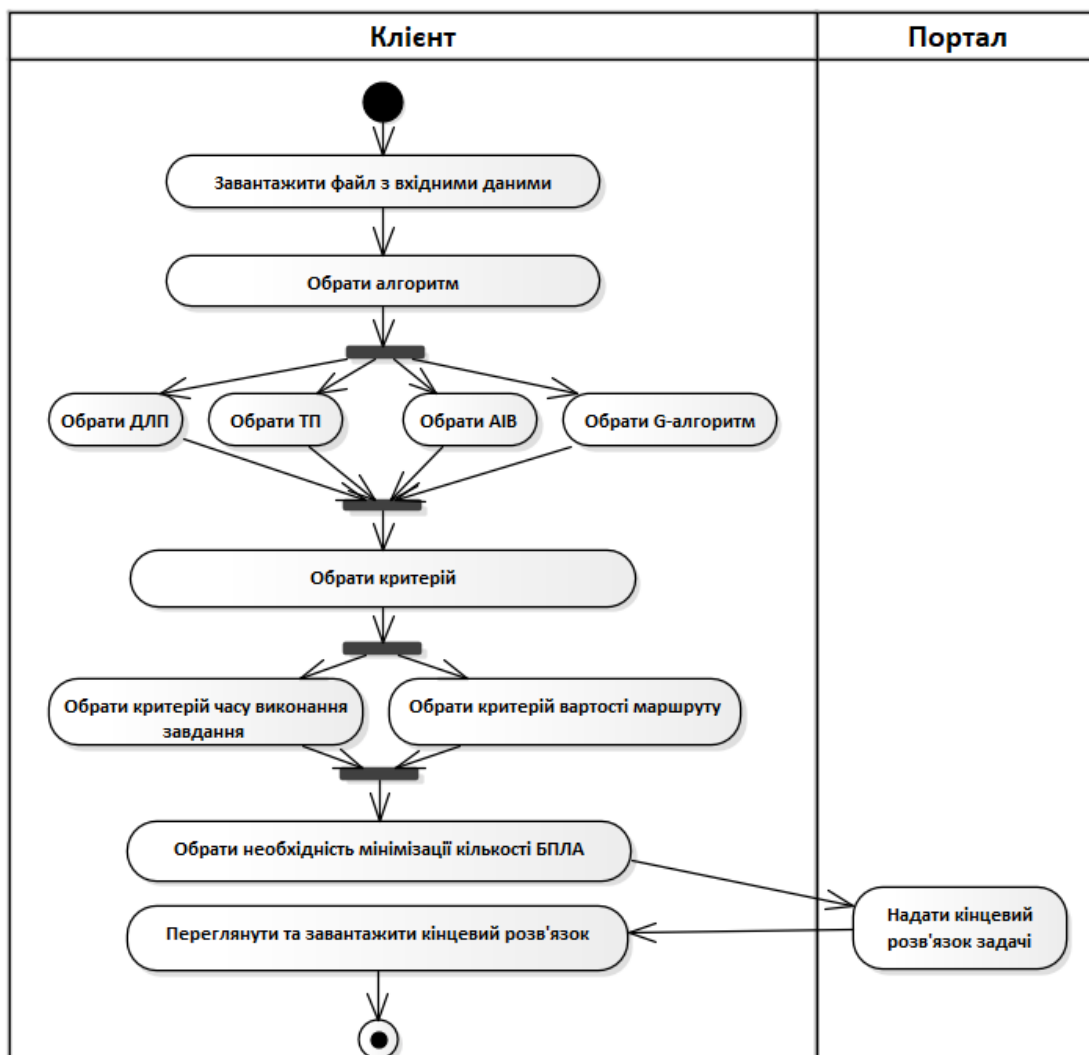
37. Rojas Vilorio D., Solano-Charnis E. Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. *International Transactions in Operational Research*. 2021. Vol. 28, No 4. P. 1626–1657.
38. Khoufi I., Laouiti A. A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones*. 2019. Vol. 3, No 3. P. 66.
39. Sacramento D., Pisinger D. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*. 2019. Vol. 102. P. 289–315.
40. Alotaibi K., Rosenbarger J. Unmanned aerial vehicle routing in the presence of threats. *Computers & Industrial Engineering*. 2018. Vol. 115. P. 190–205.
41. Wang Z., Sheu J. Vehicle routing problem with drones. *Transportation research part B: methodological*. 2019. Vol. 122. P. 350–364.
42. Li M., Zhen L. Unmanned aerial vehicle scheduling problem for traffic monitoring. *Computers & Industrial Engineering*. 2018. Vol. 122. P. 15–23.
43. Song M., Li J. Metaheuristics for solving the vehicle routing problem with the time windows and energy consumption in cold chain logistics. *Applied Soft Computing*. 2020. Vol. 95.
44. Semiz F., Polat F. Solving the area coverage problem with UAVs: A vehicle routing with time windows variation. *Robotics and Autonomous Systems*. 2020. Vol. 126.
45. Jia Z., Yu J. Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. *Aerospace Science and Technology*. 2018. Vol. 76. P. 112–125.
46. Coutinho W., Battarra M. The unmanned aerial vehicle routing and trajectory optimization problem, a taxonomic review. *Computers & Industrial Engineering*. 2018. Vol. 120. P. 116–128.

47. Li H., Wang H. Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological*. 2020. Vol. 138. P. 179–201.
48. Thibbotuwawa A., Nielsen P. Energy consumption in unmanned aerial vehicles: A review of energy consumption models and their relation to the UAV routing. In *International Conference on Information Systems Architecture and Technology*, Springer, Cham, September 2018. P. 173–184.
49. Вовкодав О. В., Ліп'яніна Х. В. Сучасні інформаційні технології : навч. посіб. Тернопіль : ТНЕУ, 2017. 500 с.
50. A sophisticated text editor for code, markup and prose. URL: <http://www.sublimetext.com/>
51. Анісімов А. В., Дорошенко А. Ю. Програмування числових методів мовою Python : навч. посіб. Київ : Видавничо-поліграфічний центр «Київський університет», 2014. 640 с.
52. Півняк Г. Г., Бусигін Б. С. Дівізінюк М. М. Тлумачний словник з інформатики. Дніпропетровськ : Національний гірничий університет, 2010. 600 с.
53. Кириченко А. В., Хрусталев А. А HTML5+CSS3. Основы современного web-дизайна. СПб. : Наука и Техника, 2018. 352 с.
54. Демків Т. М., Демків Л. С. Пакети Python для моделювання фізичних процесів. FLOSS Lviv 2015 : матеріали V міжнародної науково-практичної конференції, м. Львів, 23-26 квітня 2015 р. Львів, 2015. С. 21–22.
55. Коткова А.А. Оптимізація маршрутів безпілотних літальних апаратів за наявності декількох депо // Коткова А.А., Гуляницький Л.Ф. // Матеріали VI всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021) – м. Київ. : НТУУ «КПІ ім. Ігоря Сікорського», 22 квітня 2021 р.

ДОДАТОК А

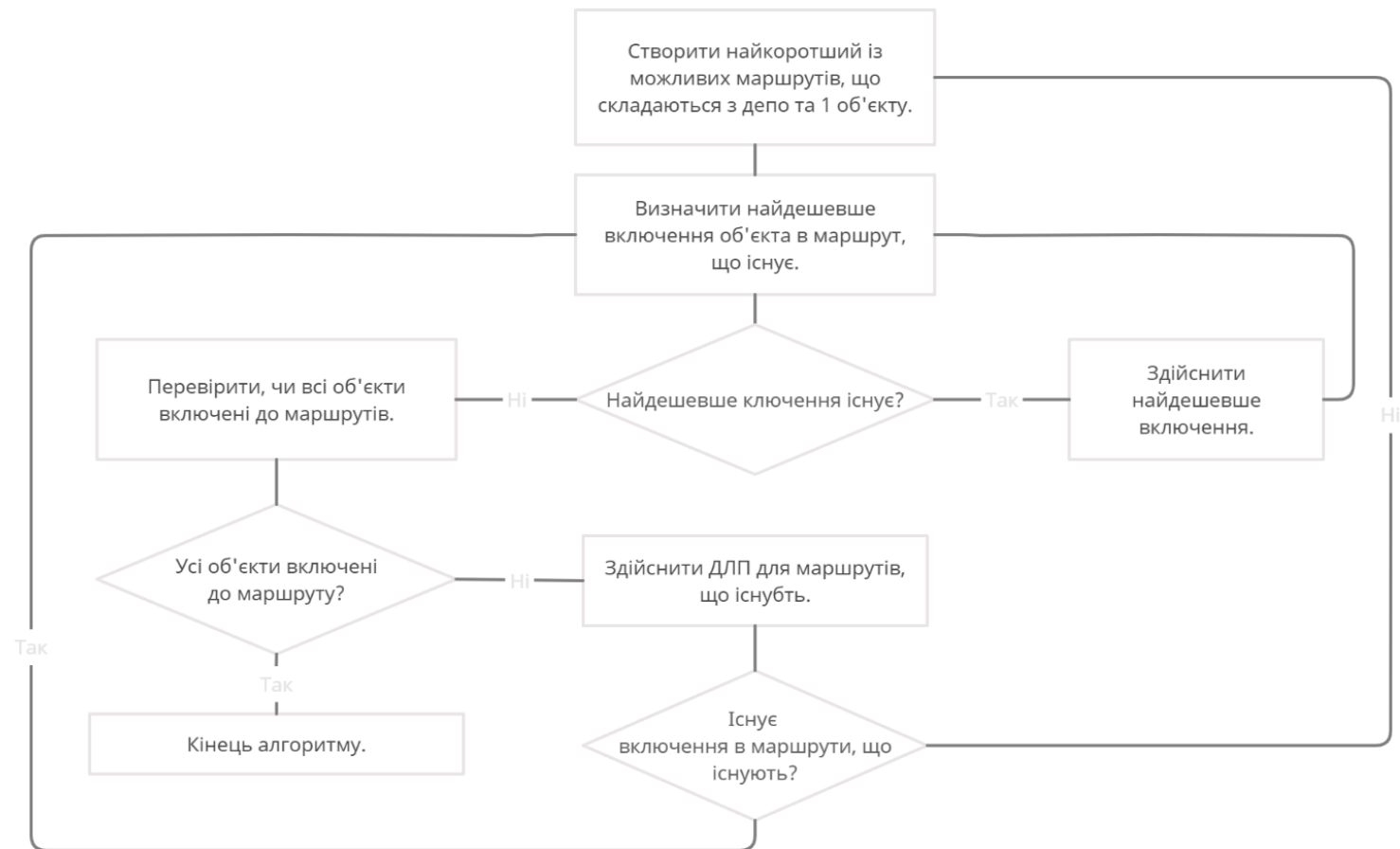
Графічний матеріал

Плакат 1 Схема структурна діяльності

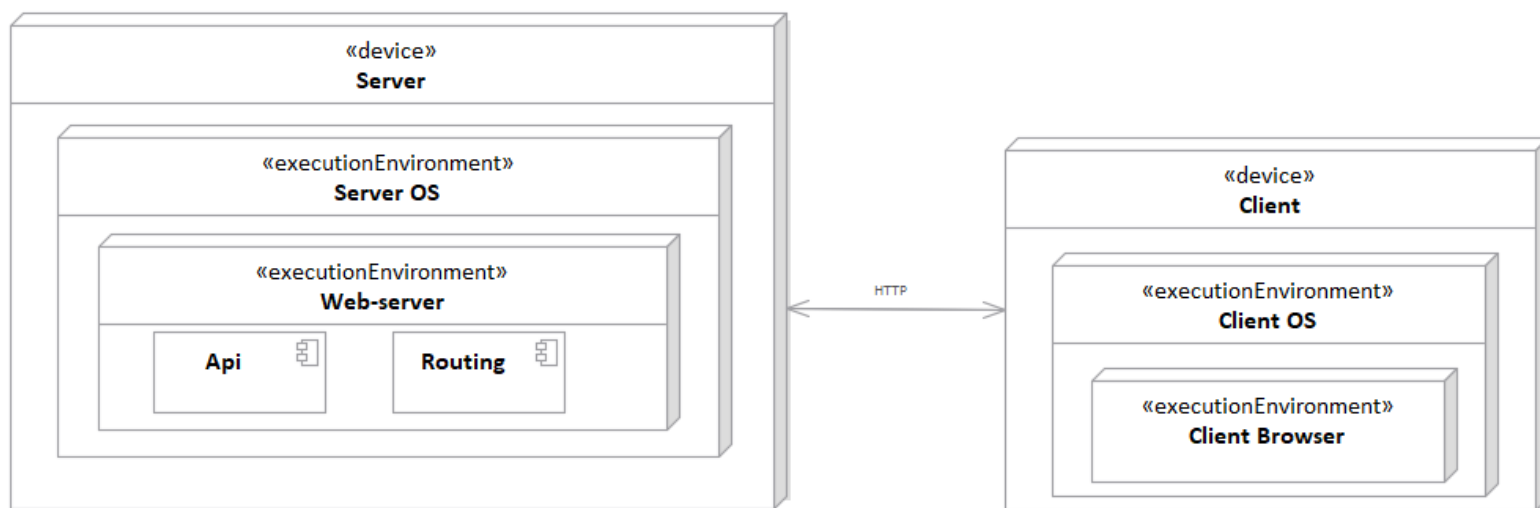


Плакат 2 Блок-схема методу найдешевшого включення

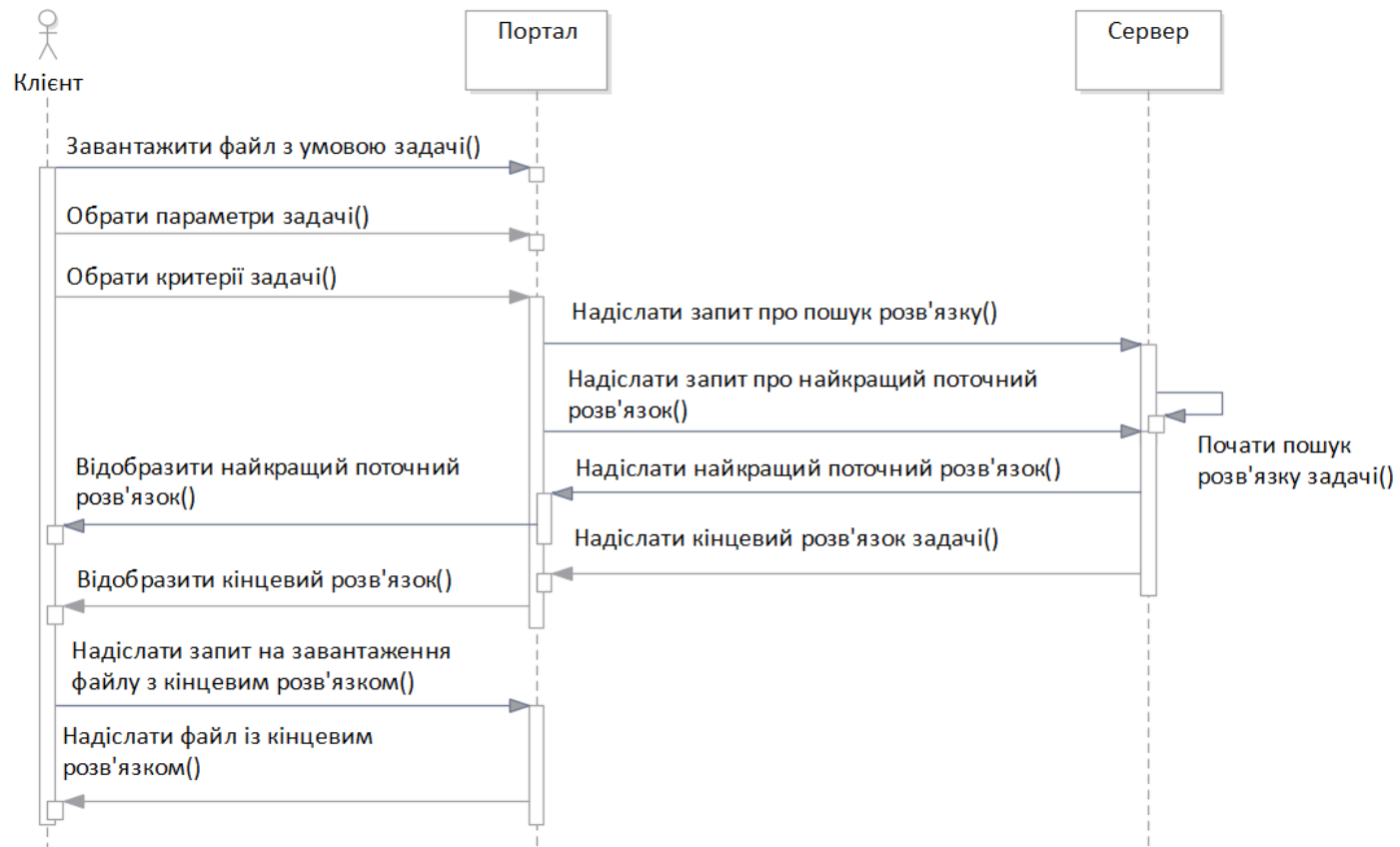


Плакат 3 Блок-схема гібридного алгоритму найдешевшого включення з проміжними покращеннями

Плакати 4 Схематична розгорнутість



Плакат 5 Схема структурна послідовності



Плакати 6 Діаграма класів

