Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: *Методи та засоби оптимального розподілу завдань в комп'ютеризованій системі безпілотної доставки*

Виконав: студент *VI* курсу, групи *СІм-62*
спеціальності *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

(підпис) *Олійник М. Є.* (прізвище та ініціали)

Керівник (підпис) *Луцик Н. С.* (прізвище та ініціали)

Нормоконтроль (підпис) *Тиш Є. В.* (прізвище та ініціали)

Завідувач кафедри (підпис) *Осухівська Г. М.* (прізвище та ініціали)

Рецензент (підпис) *Бревус В. М.* (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
« 11 » листопада 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *магістр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Олійнику Михайлу Євгеновичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Методи та засоби оптимального розподілу завдань в комп'ютеризованій системі безпілотної доставки*

Керівник роботи *Луцик Надія Степанівна, докт. філософ. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « *29 » листопада 2024 року № 4/7-1144*
2. Термін подання студентом завершеної роботи *16.12.2024 р.*
3. Вихідні дані до роботи *Наукові літературні джерела*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ.
1. Аналіз існуючих комп'ютеризованих систем безпілотної доставки.
2. Методи динамічного розподілу завдань в комп'ютеризованих системах безпілотної доставки.
3. Реалізація оптимального розподілу завдань в комп'ютеризованій системі.
4. Охорона праці та безпека в надзвичайних ситуаціях.
Висновки. Список використаних джерел.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність і мета дослідження.
2. Завдання, об'єкт і предмет, наукова новизна і практична цінність дослідження.
3. Архітектура комп'ютеризованої системи безпілотної доставки.
4. Предметна область комп'ютеризованої системи оптимального розподілу завдань.
5. Блок-схеми алгоритмів роботи основних частин комп'ютеризованої системи оптимального розподілу завдань.
6. Результати тестування роботи алгоритму оптимального розподілу завдань для сценарію №1.
7. Результати тестування роботи алгоритму оптимального розподілу завдань для сценарію №2.
8. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	*Осухівська Г.М., зав.каф. КС*	*05.12.2024*	
Безпека в надзвичайних ситуаціях	*Стручок В.С., ст. викладач каф. Обладнання харчових технологій*	*21.11.2024*	

7. Дата видачі завдання *11.11.2024*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Аналіз існуючих методів та засобів розв'язання науково-технічних проблем, що відповідають тематиці кваліфікаційної роботи.*	*20.11.2024р. – 30.11.2024р.*	*Викон.*
2	*Проектування архітектури комп'ютеризованої системи безпілотної доставки*	*21.11.2024р. – 23.11.2024р*	*Викон.*
3	*Розробка структури системи оптимального розподілу завдань*	*24.11.2024р.*	*Викон.*
4	*Реалізація алгоритмічного забезпечення методу оптимального розподілу завдань*	*25.11.2024р.*	*Викон.*
5	*Реалізація програмного забезпечення методу оптимального розподілу завдань*	*26.11.2024р. – 06.12.2024р.*	*Викон.*
6	*Охорона праці та безпека в надзвичайних ситуаціях*	*07.12.2024р*	*Викон.*
7	*Оформлення пояснювальної записки і графічного матеріалу*	*08.12.2024р.- 15.12.2024р.*	*Викон.*
8	*Попередній захист кваліфікаційної роботи магістра*	*16.12.2024р.*	*Викон.*
9	*Захист кваліфікаційної роботи магістра*	*23.12.2024р.*	*Викон.*

Студент _____ *Олійник М. Є.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Луцик Н. С.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Олійник М. Є. Методи та засоби оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки: робота на здобуття кваліфікаційного ступеня магістра: спец. 123 — комп’ютерна інженерія / наук.кер. Н. С. Луцик. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. — 70 с.

Ключові слова: комп’ютеризована система, безпілотна доставка, розподіл завдань, генетичний алгоритм.

Кваліфікаційна робота присвячена дослідженню методів та засобів оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки. Робота включає в себе аналіз існуючих досліджень у сфері безпілотної доставки, а також зосереджує увагу на методах динамічного розподілу завдань в системах де враховуються різні обмеження.

Результати даної роботи можуть бути застосованими в реальній комп’ютеризованій системі безпілотної доставки. Розроблений метод оптимального розподілу завдань дозволить суттєво підвищити ефективність систем безпілотної доставки, враховуючи типи пристроїв, їхні характеристики та робочі зони.

ANNOTATION

Oliinyk M. Methods and tools for optimal task allocation in a computerized unmanned delivery system. Master's Graduation Thesis: speciality 123 — Computer engineering / supervisor N. Lutsyk. Ternopil: Ternopil Ivan Puluj National Technical University, 2024. — 70 p.

Keywords: computer system, unmanned delivery, task allocation, genetic algorithm.

The qualification work is dedicated to the study of methods and tools for optimal task allocation in a computerized unmanned delivery system. The work includes an analysis of existing research in the field of unmanned delivery and focuses on methods of dynamic task allocation in systems that take various constraints into account.

The results of this work can be applied to real computerized unmanned delivery systems. The developed method for optimal task allocation will significantly enhance the efficiency of unmanned delivery systems, considering the types of devices, their characteristics, and operational zones.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ БЕЗПІЛОТНОЇ ДОСТАВКИ 12
1.1. Комп'ютеризовані системи безпілотної доставки 12
1.1.1. Огляд апаратних засобів безпілотної доставки 13
1.2. Застосування безпілотних систем доставки 16
1.3. Сучасний стан досліджень у галузі безпілотної доставки 19
1.4. Висновки до розділу 25
РОЗДІЛ 2 МЕТОДИ ДИНАМІЧНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНИХ СИСТЕМАХ БЕЗПІЛОТНОЇ ДОСТАВКИ 26
2.1. Алгоритми динамічного розподілу завдань 26
2.2. Методи визначення енергетичних витрат безпілотних пристроїв 30
2.3. Аналіз існуючих методів динамічного розподілу завдань 31
2.4. Застосування генетичного алгоритму в системах безпілотної доставки 34
2.4.1. Додавання протилежної популяції 36
2.4.2. Операції схрещування 36
2.4.3. Операції кількісної мутації. 38
2.5. Висновки до розділу 40
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ОПТИМАЛЬНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНІЙ СИСТЕМІ 41
3.1. Врахування типу безпілотників та зони їхньої досяжності 41
3.2. Архітектура комп'ютеризованої системи безпілотної доставки 42
3.3. Структура системи оптимального розподілу завдань 43
3.4. Розробка алгоритмічного та програмного забезпечення 46
3.4.1. Алгоритмічне забезпечення системи 46
3.4.2. Програмне забезпечення системи 49

3.5. Аналіз результатів оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки 52
3.6. Висновки до розділу 60
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 61
4.1. Охорона праці 61
4.2. Заходи та засоби захисту комп'ютерних систем до впливу надзвичайних ситуацій воєнного характеру 64
ВИСНОВКИ 67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 68
Додаток А Тези конференцій 71
Додаток Б Лістинг коду генетичного алгоритму 77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ Програмне забезпечення

PSO (англ. Particle Swarm Optimization) метод рою частинок

CNP (англ. Contract Net Protocol) Смарт-контракт

GA (англ. Genetic Algorithm) генетичний алгоритм

OGA-DEMMO (англ. Opposition-based Genetic Algorithm using Double-chromosomes Encoding and Multiple Mutation Operators) генетичний алгоритм на основі протилежних хромосом з подвійним кодуванням та множинних операторів мутації

ВСТУП

Актуальність роботи. Розвиток сучасних технологій суттєво впливає на логістичну сферу, забезпечуючи нові можливості для оптимізації процесів доставки. Одним із перспективних напрямків є впровадження безпілотних систем доставки, що включає використання дронів і наземних роботів. Дані системи здатні досягати важкодоступні регіони, мінімізуючи енергетичні витрати та необхідність людського втручання.

Ринок безпілотних систем доставки демонструє стабільний ріст, який обумовлений потребою в автоматизації логістичних процесів. Швидкість і точність доставки набувають вирішального значення для економічного розвитку, а технології безпілотної доставки привертають усе більше уваги серед дослідників. Завдяки своїй різноманітності, безпілотні пристрої вже знаходять застосування в таких сферах, як доставка їжі, медикаментів, комерційних вантажів та гуманітарної допомоги, що підкреслює їхній потенціал.

Однак, попри явні переваги, системи безпілотної доставки мають низку невирішених питань, пов’язаних із їхньою взаємодією. Це обумовлено різноманітністю пристроїв і відсутністю єдиного підходу до їхньої взаємодії. Особливо це актуально у випадках, коли в межах однієї системи використовуються різні типи пристроїв. Для ефективного розподілу завдань в таких системах є необхідність в застосуванні алгоритмів, які здатні враховувати типи безпілотних пристроїв та їхні робочі зони.

Таким чином, аналіз методів і алгоритмів розподілу завдань є ключовим кроком до підвищення енергоефективності та доступності безпілотних систем доставки. Врахування різних типів пристроїв, їхніх характеристик та робочих зон дозволить ефективніше розподіляти завдання та зменшити загальне енергоспоживання.

Метою кваліфікаційної роботи є створення методів та засобів для оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки з врахуванням їхнього типу та робочих зон.

Завдання кваліфікаційної роботи:

– провести огляд літератури та аналіз існуючих комп’ютеризованих систем безпілотної доставки;

– проаналізувати існуючі методи та алгоритми динамічного розподілу завдань;

– розробити архітектуру комп’ютеризованої системи безпілотної доставки, що дозволяє враховувати різні типи і характеристики безпілотних засобів та їхніх робочих зон для найбільш ефективної їх роботи;

– розробити метод оптимального розподілу завдань для комп’ютеризованої системи безпілотної доставки з врахуванням типу пристрою та робочих зон;

– розробити веб-застосунок для моделювання процесу розподілу та проведення аналізу роботи розробленої системи;

– провести тестування роботи алгоритму оптимального розподілу завдань для комп’ютеризованої системи безпілотної доставки.

Відповідно до мети та завдань кваліфікаційної роботи визначено її об'єкт та предмет.

Об’єкт дослідження: процес оптимального розподілу задач в комп’ютеризованих системах безпілотної доставки.

Предмет дослідження: методи та засоби оптимального розподілу завдань в системах безпілотної доставки з врахуванням типів і характеристик пристроїв та їхніх робочих зон.

Методи дослідження: моделювання для імітації роботи комп’ютеризованої системи безпілотної доставки з набором робочих зон, безпілотних засобів та завдань для обчислення їхнього розподілу, евристичні методи для пошуку оптимального рішення у задачі оптимального розподілу завдань, методи математичного моделювання для розрахунку енергетичних витрат, методи

системного аналізу для комплексного розгляду взаємозв'язків між різними елементами системи та їх впливу на загальну ефективність.

Наукова новизна дослідження:

1. Вперше представлено архітектуру комп'ютеризованої системи безпілотної доставки на основі методу оптимального розподілу завдань шляхом створення універсального підходу, що дозволяє враховувати різні типи і характеристики безпілотних засобів та їхніх робочих зон для найбільш ефективної їх роботи.

2. Запропоновано використання розробленого веб-застосуноку для моделювання процесу розподілу завдань та проведення аналізу роботи досліджуваної комп'ютеризованої системи.

Практичне значення одержаних результатів кваліфікаційної роботи полягає у можливості впровадження запропонованого методу в реальній комп'ютеризовані системи безпілотної доставки, для оптимального розподілу завдань з врахуванням типів пристроїв, їхніх характеристик та робочих зон.

Публікації. Результати дослідження апробовано на XII науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології», XIII міжнародній науково-технічній конференція молодих учених та студентів «Актуальні задачі сучасних технологій», у вигляді тез конференцій.

1. Луцик Н. С., Олійник М. Є., Мудрий І. В. Аналіз комп'ютеризованих систем безпілотної доставки. Актуальні задачі сучасних технологій: Праці XIII наук.-техн. конф. (Тернопіль, 11-12 грудня 2024 р.), Тернопіль, 2024. С. 465.

2. Луцик Н. С., Олійник М. Є., Мудрий І. В. Методи та засоби оптимального розподілу завдань в комп'ютеризованій системі безпілотної доставки. Інформаційні моделі, системи та технології: Праці XII наук.-техн. конф. (Тернопіль, 18-19 грудня 2024 р.), Тернопіль, 2024. С. 141.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, 4 розділів,

висновків, списку використаних джерел та додатку (-ів). Обсяг роботи: пояснювальна записка – 70 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ БЕЗПІЛОТНОЇ ДОСТАВКИ

1.1. Комп'ютеризовані системи безпілотної доставки

Системи безпілотної доставки базуються на використанні автономних транспортних засобів, які здатні виконувати завдання з перевезення вантажів без участі людини або з мінімальним втручанням. У рамках дослідження комп'ютеризованих систем безпілотної доставки важливим є виділити його основні структурні елементи, які є важливими для реалізації та функціонування обраного алгоритму розподілення завдань та методів взаємодії з безпілотниками. На рис. 1.1 зображена загальна архітектура системи безпілотної доставки.

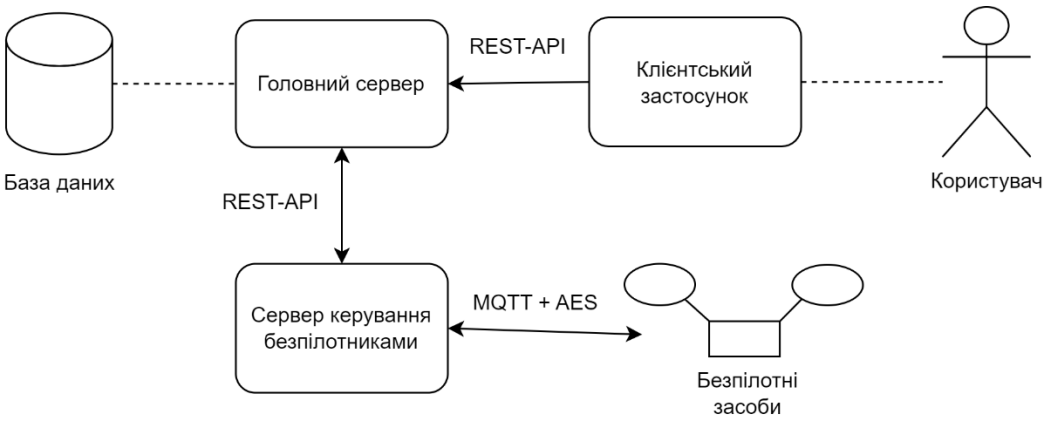

Рис. 1.1. Загальна архітектура системи безпілотної доставки

У даній системі клієнт представлений користувацьким додатком, призначеним для спрощеної взаємодії користувачів із системою. Через REST-API інтерфейс додаток надсилає HTTPS-запити до веб-сервера, який опрацьовує їх та виконує необхідні операції з даними, включаючи операції читання, створення, оновлення та видалення. Усі дії, що стосуються роботи із системою, а саме реєстрація, створення та перегляди списків відправлень і статусів здійснюються серверним модулем.

Модуль керування безпілотниками виконує централізоване управління групою роботів, а саме розраховує оптимальні маршрути враховуючи трафік, погодні умови та доступність роботів. Для організації взаємодії з безпілотниками використовується протокол MQTT у форматі “Publisher-Subscriber”. Такий підхід дозволяє серверу та апаратному забезпеченню здійснювати постійний обмін інформацією та активно використовується у сфері інтернету речей.

Дана розподілена архітектура дозволяє інтегрувати інші користувацькі сервіси до модуля керування безпілотниками, що розширює можливості застосування таких систем.

В динамічному середовищі вибір конкретного типу транспортного засобу залежить від завдань, що стоять перед компанією, характеристик вантажу, умов навколишнього середовища, а також від інфраструктури. Відкритість сервера керування безпілотників до інтеграції в інші системи, є критично важливим, оскільки це забезпечує гнучкість і масштабованість таких систем. Це дозволяє об’єднувати різні типи безпілотників в єдину систему розподілу завдань, що сприяє оптимальному підбору апаратів для виконання конкретних задач доставки, підвищуючи ефективність та адаптивність всієї системи.

1.1.1. Огляд апаратних засобів безпілотної доставки. Розвиток технологій автономного транспорту спричинив появу різноманітних рішень – від повітряних дронів до наземних роботів та гібридних платформ. Кожен з цих типів має свої особливості та сферу застосування, які впливають на ефективність виконання доставки.

Найпоширенішим засобом безпілотної доставки є повітряні дрони, які оснащені електродвигунами, системою GPS та технологією Lidar. Найкраще підходять для доставки малогабаритних вантажів у важкодоступні або густонаселені райони. Повітряні дрони можна поділити на два основні типи: квадрокоптер (Quadcopter) та безпілотник з нерухомим крилом (Fixed-wing drone).

Квадрокоптер (рис. 1.2) – це пристрій, рух якого забезпечується довільною кількістю несучих гвинтів, розміщених в одній площині, що обертаються. Використовуються для точкової доставки в густонаселених або важкодоступних районах.

Рис. 1.2. Вигляд квадрокоптера

Безпілотник з нерухомим крилом (рис. 1.3), може здійснювати політ завдяки підйомній силі, яка виникає через аеродинамічну форму крила під час його руху вперед із визначеною швидкістю. Це підвищує їхню енергоефективність і дозволяє досягати більшого радіуса дії у порівнянні з квадрокоптерами.

Рис. 1.3. Вигляд безпілотника з нерухомим крилом

Наземні роботи є альтернативою повітряним дронам для доставки. Це автономні пристрої для транспортування вантажів на короткі або середні відстані. Вони оснащені колесами, що дозволяє їм пересуватися тротуарами, дорогами або спеціально облаштованими маршрутами. Завдяки використанню технологій штучного інтелекту, GPS і сенсорних систем, такі роботи здатні самостійно орієнтуватися в просторі та обходити перешкоди.

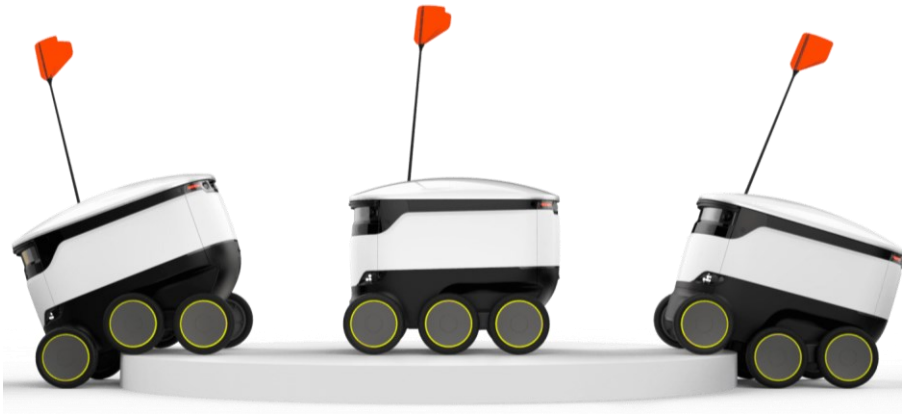

Рис. 1.4. Робот-кур’єр Starship Technologies

Вантажопідйомність роботів залежить від моделі та зазвичай не перевищує 50 кг, що робить їх ідеальними для перевезення посилок, продуктів і медикаментів на короткі відстані.

Рис. 1.5. Робот-кур’єр Clevon 1

Завдяки живленню від акумуляторів вони здатні працювати кілька годин без підзарядки. Незважаючи на деякі обмеження, зокрема чутливість до погодних умов і потребу в спеціалізованій інфраструктурі, такі роботи є перспективним рішенням для автоматизації доставки в містах і на закритих територіях.

1.2. Застосування безпілотних систем доставки

На даний момент існує відносно небагато повністю функціональних аналогів систем безпілотної доставки. Більшість таких сервісів перебувають на етапі тестування або розробки, або ж працюють у межах обмежених географічних зон. Окрім того, значна частина цих рішень орієнтована на доставку «останньої милі», а саме на вирішення логістичних складнощів доставки товарів від складу до кінцевого споживача, що супроводжуються високими витратами. Це відкриває можливість використовувати їхні послуги для комерційних потреб.

Zipline — це американський проєкт, що займається виробництвом і доставкою вантажів за допомогою повітряних дронів [7]. Основною метою компанії є вирішення проблеми доставки у важкодоступні райони. Вигляд безпілотника зображений на рис. 1.6.

Рис. 1.6. Безпілотник Zipline [7]

На сьогодні послуги Zipline доступні в Японії та США.

До переваг безпілотника варто віднести: високу швидкість доставки на великі відстані завдяки мобільності безпілотників (максимальна швидкість дрона

— до 112 км/год); автономність польотів за нормальних погодних умов; зручна система завантаження посилок; можливість підбирати/доставляти посилки у відносно невеликі зони.

До недоліків варто віднести обмежену вантажопідйомність дрона (лише 3,6 кг.) та нішевий характер послуг, оскільки сервіс переважно спеціалізується на доставці медикаментів. Характеристики безпілотника Zipline представлені у табл. 1.1.

Таблиця 1.1

Характеристики платформи доставки Zipline [7]

Спрямування	Доставка додому
Максимальна вага	2.7 - 3.6 кг.
Радіус роботи	16 км.
Швидкість польоту	112 км/год.
Ділянка доставки	Невеликі ділянки: співмірних розміру зовнішнього двора будинку або невеликого столика
Система завантаження	Прості у використанні завантажувальні портали для надсилання замовлень
Механізм доставки	Опускається спеціальний портал з висоти понад 90 м. над рівнем землі

Сервіс надає змогу користувачам оформлювати замовлення на сайті компанії, а також відстежувати процес доставки через спеціалізований мобільний додаток. Zipline надає можливості для інтеграції їхнього сервісу доставки в інші системи, що робить його повністю придатним для використання в інших системах.

Starship Technologies — Естонський стартап, заснований у 2014 році, який згодом трансформувався у компанію. Його основною метою є вирішення проблеми доставки «останньої милі» за допомогою наземних дронів [8].

Сервіс вже впроваджений та активно працює в таких містах, як Лондон, Таллінн, Дюссельдорф, Гамбург, Берн, а також у кількох містах США, зокрема у Вашингтоні та у Каліфорнії.

До переваг сервісу варто віднести повну автономність роботів кур'єрів. Вони здатні самостійно пересуватись по тротуарам та оминати різного роду перешкоди. Ємність батареї дозволяє їм працювати понад 12 годин на одному заряді. Всі характеристики робота-кур'єра представлені в табл. 1.2. Пристрої оснащені міцними замками для вантажного відділу, який відкривається безпосередньо користувачу який введе персональний код безпеки. А також у разі навігаційних труднощів передбачена можливість дистанційного керування роботом.

До недоліків входить швидкість пересування, яка не перевищує 6,5 км/год, що дорівнює середній швидкості пішохода. Та вузька спеціалізація, оскільки проєкт орієнтований виключно на доставку продуктів харчування вагою до 9 кг.

Таблиця 1.2

Характеристики платформи доставки Starship Technologies [9].

Габарити робота	697 x 569 x 571 мм
Загальна вага	35 кг
Габарити вантажного відсіку	400 × 320 × 340 мм
Максимальне навантаження	10 кг
Максимальна швидкість	6 км/год
Ємність батареї	1260 Вт
Час роботи	≈12 годин їзди
Сенсори	Ультразвукові датчики, 12 камер, радар, GPS, система сигналізації, рефлектори, стереоскопічні камери
Дистанція ідентифікації об'єктів	До 60 м
FPS камери	2,000 кадрів на секунду

Starship Technologies має спеціалізований мобільний застосунок, через який користувачі можуть оформляти замовлення, оплачувати послуги, відстежувати переміщення дрона та виконувати інші операції. Важливо зазначити що даний сервіс приділив велику увагу до можливості інтеграції із іншими платформами. А також надають повний набір інструментів для розробників, документацію та підтримку.

Clevon — інноваційна компанія з виготовленням засобів безпілотної доставки. Робот-кур'єр Clevon 1 було розроблено для вирішення проблеми доставки «останньої милі». Даний пристрій інтегрований в системі логістики DHL Express Estonia для оптимізації процесу доставки. Його очевидними перевагами є швидкість переміщення, вантажопідйомність та запас ходу. Всі характеристики Clevon 1 представлені у табл. 1.3.

Таблиця 1.3

Характеристики платформи доставки CLEVON 1 [10].

Розміри робота	2470 x 1150 x 1550 мм
Вага	530 кг.
Максимальне навантаження	100 кг
Максимальна швидкість	25 км/год
Максимальна дальність	100 км на одному заряді
Система живлення	100% електрична
Особливості	Система MultiBox для перевезення кількох пакунків, система уникнення перешкод
Сенсори	GPS, радар, камери

1.3. Сучасний стан досліджень у галузі безпілотної доставки

Процес аналізу розпочинається з виявлення та ідентифікації всіх релевантних наукових та технічних досліджень, що проводились у сфері безпілотної доставки. Сучасний стан досліджень цієї галузі дає можливість

визначити поточні тенденції, напрацювання та можливі недоліки наявних рішень, які стануть основою для розробки нових методів та засобів оптимального розподілу завдань в системах безпілотної доставки.

На сьогоднішній день велика увага зосереджена на дослідженнях безпілотних систем доставки та методам розподілу завдань між ними. У роботі [11] проведено систематичний огляд літератури, присвяченої методам доставки на основі дронів, основним викликам та майбутнім перспективам. Дослідження спрямоване на виявлення поточних тенденцій, аналіз математичних моделей, що використовуються для оптимізації процесів доставки, та визначення ключових напрямів подальших досліджень у цій галузі. Дослідження зосереджено на систематизації наукових робіт, класифікуючи їх за такими аспектами, як задачі комівояжера (Traveling Salesman Problem), проблему маршрутизації транспортних засобів (Vehicle Routing Problem), планування маршрутів доставки дронами, оптимізація дронів та проблеми «останньої милі» в урбанізованих середовищах.

Одним із основних елементів дослідження є аналіз математичних моделей, таких як цілочисельне програмування та різноманітні евристичні методи, зокрема генетичні алгоритми, адаптивні пошукові алгоритми та методи гібридної оптимізації. Значна частина досліджень даної тематики присвячена аналізам проблем, які виникають під час практичного впровадження даних систем. Існують рекомендації, щодо подолання цих труднощів [11], зокрема інтеграція штучного інтелекту, розробка стійких алгоритмів та покращення енергоефективності.

У роботі [12] детально проаналізовано алгоритми розподілу завдань у системах з великою кількістю дронів орієнтованих на операції порятунку. Основна увага зосереджена на двох підходах розподілу завдань, а саме статичному та динамічному. Проведено їхні порівняння за ключовими параметрами, а саме гнучкості, масштабованості і обчислювальної складності. Дослідження підкреслює, що статичний розподіл завдань базується на попередньо визначених сценаріях і даних, зібраних на початку місії. Такий підхід ефективний в передбачуваних умовах, але має низьку адаптацію до змін, таких як поява нових

завдань. Натомість, динамічний розподіл завдань, дає змогу переорганізувати завдання в режимі реального часу, реагуючи на зміни в середовищі чи в параметрах завдань. Порівняння характеристик цих двох підходів представлено у табл. 1.4.

Таблиця 1.4

Різниця між статичним та динамічним розподілом завдань [12]

Параметри	Статичний розподіл завдань	Динамічний розподіл завдань
Розподіл завдань	Визначене заздалегідь	Адаптивне
Гнучкість	Низька: завдання не можуть перепризначатися	Висока: завдання можуть перепризначатися
Масштабованість	Низька	Висока
Обчислювальна складність	Низька	Висока
Придатність	Передбачувані робочі навантаження	Непередбачувані робочі навантаження
Залежність від даних	В основному на даних перед виконанням	Залежить від даних у реальному часі
Вимоги до комунікації	Помірні	Високі
Можливості БПЛА	Базова обчислювальна потужність	Розширена обчислювальна потужність
Використання ресурсів	Можливе неповне використання потенціалу	Кращий баланс навантаження

В дослідженні узагальнено, що статичні методи характеризуються низькою гнучкістю, простотою реалізації та обмеженою здатністю адаптуватись до змін. У той час, як динамічні методи відзначаються високою адаптивністю, здатністю до переорганізації завдань та рівномірним розподілом ресурсів.

У роботі [13] досліджено алгоритми динамічного розподілу завдань в системі безпілотних літальних пристроїв, орієнтованих на наземні цілі. В основному увага приділяється характеристикам і застосуванню алгоритмів, які вирішують складні завдання в умовах динамічних змін (рис. 1.7). Аналіз охоплює створення моделей динамічного розподілу завдань та підходи до їхнього вирішення.

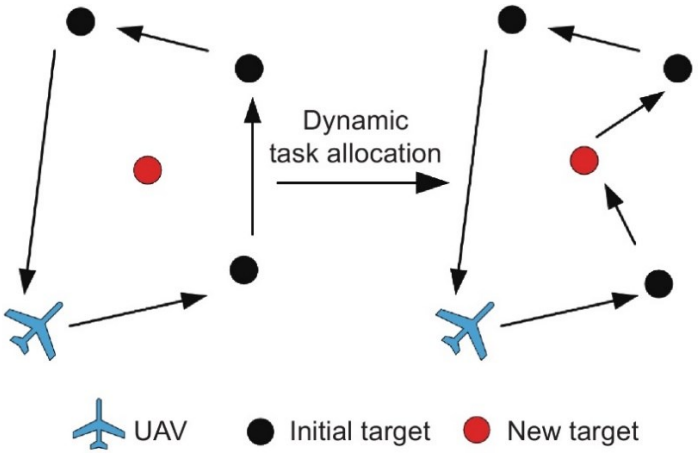

Рис. 1.7. Проблема динамічного призначення завдань [13]

Зазначається, що динамічний розподіл завдань є NP-складною задачею, яка вимагає обробки великих обсягів даних у режимі реального часу. Особливу увагу приділено порівнянню популярних динамічних алгоритмів (табл. 1.5), таким як аукціонні алгоритми, методи на основі ринкових механізмів, генетичні алгоритми, алгоритми оптимізації рою (PSO) та алгоритми кластеризації.

Таблиця 1.5

Порівняння поширених алгоритмів динамічного розподілу завдань [13]

Алгоритм	Динамічність	Стійкість	Толерантність до збоїв	Точність	Масштаб застосування	Швидкість
Аукціонний алгоритм	С	С	С	П	В	Ш

Продовж. табл 1.5

Алгоритм	Динамічність	Стійкість	Толерантність до збоїв	Точність	Масштаб застосування	Швидкість
Смарт-контракт алгоритм (CNP)	С	С	С	П	П	Ш
Генетичний алгоритм (GA)	С	С	С	П	П	Ш
Метод рою частинок (PSO)	С	С	П	П	В	Ш
Кластерний алгоритм	С	Сл	П	Сл	П	Ш
Примітка: С- сильний; П- помірний; Сл-слабкий; В- великий; Ш- швидкий.						

Таблиця 1.5 демонструє різницю в їхній динамічності, стійкості до відмов, точності та швидкості обчислень. Наприклад, кластерні алгоритми характеризуються високою швидкістю, але обмеженою точністю, тоді як генетичний алгоритм забезпечує високі показники точності, стійкості та швидкості.

У статті [14] спроектовано комплексну систему автономної навігації для дронів, орієнтовану на забезпечення ефективної та точної доставки вантажів, особливо в умовах складного міського середовища. Основну проблему, яку вирішує дане дослідження є досягнення безпечного й точного приземлення дронів, де традиційні методи, такі як GPS виявляються не достатньо точними через вплив завад і обмеженість сигналу. Структурна схема системи зображена на рис. 1.8.

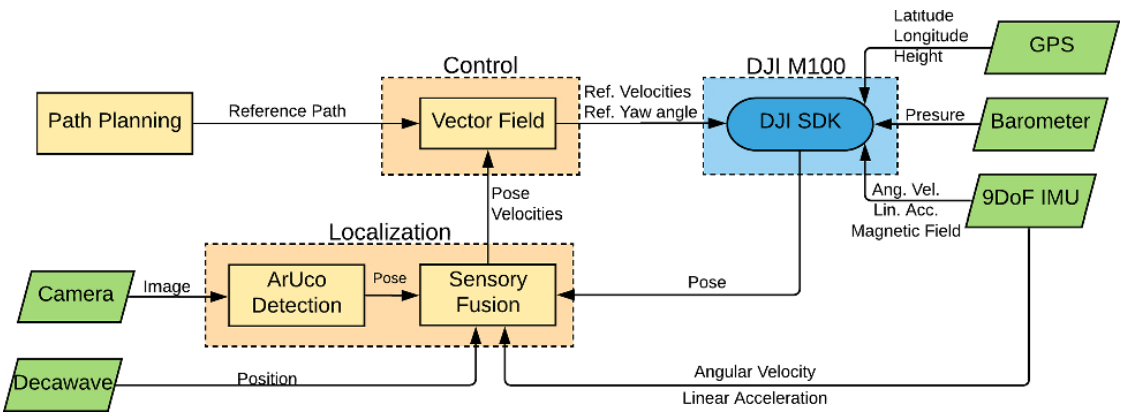

Рис. 1.8. Структурна схема взаємодії між модулями КС [14]

Система навігації базується на поєднанні даних з сенсорів GPS, інерційного вимірювального блоку (IMU), барометра та камери. Схема взаємодії всіх компонентів дрона зображена на рис. 1.9.

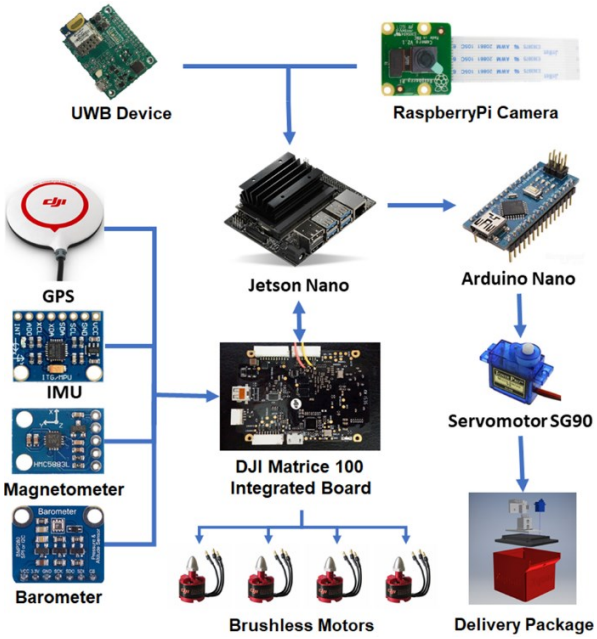

Рис. 1.9. Схема взаємодії компонентів дрона [14]

Для підвищення точності локалізації використовується розширений фільтр Калмана (EKF), що інтегрує дані від сенсорів і дозволяє зменшити похибку приземлення до 12 см, тоді як використання лише GPS дає похибку понад 1 метр.

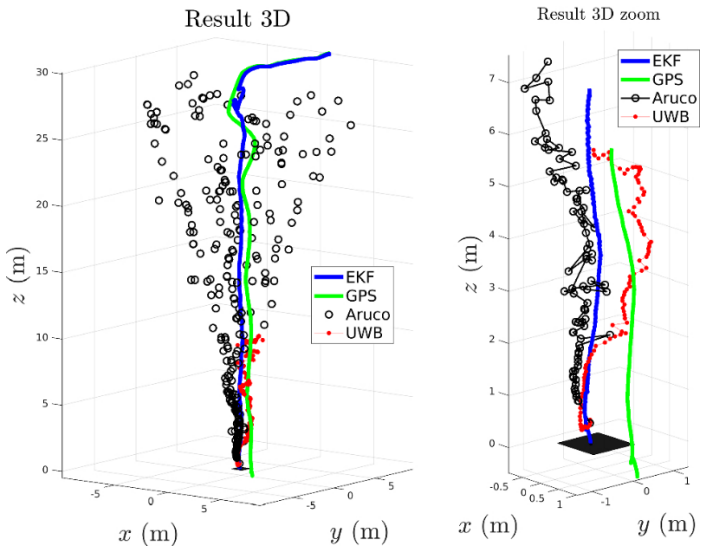

Рис. 1.10. Графік 3D позиціонування дрона під час приземлення [14]

Управління маршрутом реалізовано за допомогою методу векторного поля, що забезпечує плавність руху дрона і його точне слідування запланованому маршруту (див. рис. 1.10).

1.4. Висновки до розділу

У першому розділі кваліфікаційної роботи було проведено аналіз сучасних комп’ютеризованих систем безпілотної доставки. Розглянуто архітектуру комп’ютеризованих систем та методи взаємодії з безпілотними пристроями.

Здійснено аналіз існуючих засобів для безпілотної доставки. Проаналізовано їхні характеристики та оцінено спроможність для інтеграції з іншими комп’ютеризованими системами. Описано сучасний стан досліджень у галузі безпілотної доставки та методів розподілу завдань між безпілотниками.

Як показує аналіз, системи безпілотної доставки потребують вдосконалення в умовах динамічних змін середовища з врахуванням типів та технічних характеристик пристроїв.

РОЗДІЛ 2
МЕТОДИ ДИНАМІЧНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНИХ СИСТЕМАХ БЕЗПІЛОТНОЇ ДОСТАВКИ

2.1. Алгоритми динамічного розподілу завдань

Основою роботи комп'ютеризованої системи безпілотної доставки є алгоритми розподілу завдань. Залежно від архітектури системи, вони поділяються на централізовані та децентралізовані.

Централізовані алгоритми це ті, у яких усі обчислення виконує центральний сервер, який передає дронам інформацію про їхні завдання. Даний підхід передбачає постійний обмін даними між основним сервером і безпілотниками.

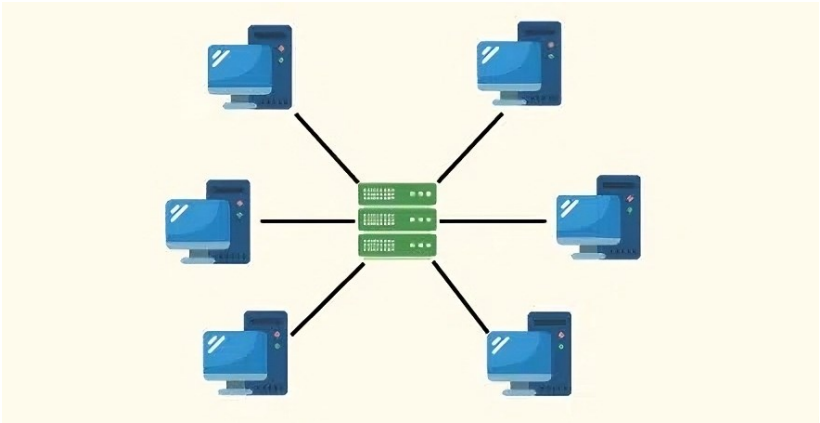

Рис. 2.1. Принцип централізованого підходу

Однією з основних проблем централізованих алгоритмів є їхня залежність від центральної обчислювальної системи. У разі збою центрального сервера процес розподілу завдань повністю припиняється, що вимагає впровадження механізмів резервування для зниження ризику відмови. Проте, зі збільшенням складності та кількості обмежень, таких як моделювання енергоспоживання чи доставок із часовими рамками, виникає необхідність у використанні апроксимацій для досягнення прийнятних результатів [6].

Попри недоліки централізовані алгоритми мають значну кількість переваг.

Ці алгоритми забезпечують створення глобального оптимального плану розподілу завдань, оскільки всі дані про безпілотники, завдання та обмеження збираються в єдиному місці. Це дозволяє максимально ефективно враховувати такі фактори, як енергоспоживання, географічне розташування цілей та робочі зони засобів.

Наявність центрального вузла спрощує процес управління флотом дронів, оскільки всі рішення приймаються в одному місці. Це дозволяє швидко адаптуватися до змін у системі, наприклад, якщо з'являються нові завдання або змінюються умови виконання поточних.

Перевагою є також простота реалізації у порівняні з децентралізованими алгоритмами. Централізований підхід потребує менш складної взаємодії між безпілотниками, та не потребує додаткового втручання в ПЗ наявних засобів, що зменшує ризик комунікаційних та системних помилок і спрощує розробку.

Отже, централізована структура дозволяє легше відстежувати процес прийняття рішень і аналізувати ефективність алгоритму, що є важливим для тестування та вдосконалення системи.

Децентралізовані алгоритми розподіляють обчислення між усіма учасниками групи дронів. Кожен засіб бере участь у пошуку оптимального розподілу завдань. Даний підхід усуває необхідність в централізованому вузлі управління.

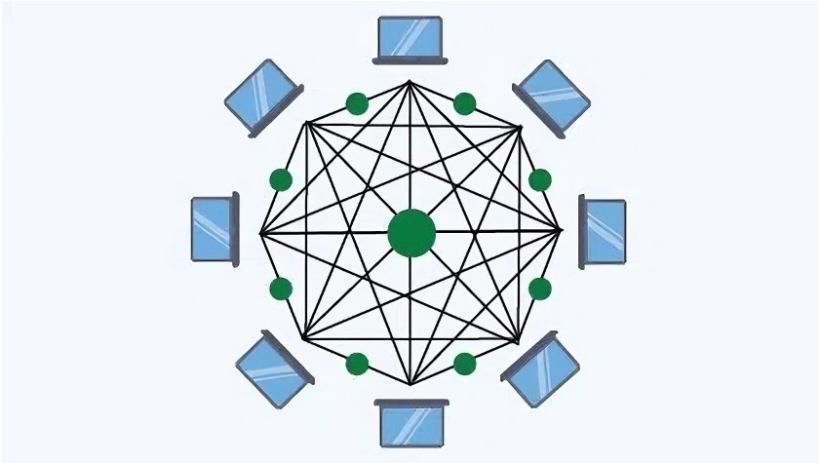

Рис. 2.2. Принцип децентралізованого підходу

Він є більш стійким до відмов, оскільки кожен пристрій працює незалежно і в разі поломки одного з них вся система й надалі буде працювати без будь-яких порушень. Несправний дрон просто припиняє виконувати завдання до моменту свого ремонту. Найчастіше в децентралізованих системах використовуються алгоритми на основі ринкових механізмів де кожен пристрій робить ставку, для того аби отримати завдання, яке найкраще відповідає його параметрам. Усі розрахунки проводяться на борту дрона, що значно зменшує необхідність у передачі даних порівняно з централізованими алгоритмами.

Варто відзначити що додатковою перевагою розподілених алгоритмів є їхня масштабованість. Зростання кількості дронів у групі не призводить до суттєвого зростання часу обчислень.

Попри всі переваги дані алгоритми мають низку недоліків:

- вони вимагають більше обчислювальної потужності на борту кожного безпілотника, що матиме значний ефект на споживанні електроенергії;

- для реалізації даного підходу необхідно доповнити відповідним програмним забезпеченням всі пристрої в групі. Це є досить складним завданням оскільки виробники безпілотних засобів мають власні ПЗ для навігації пристроїв, та логіки їхньої роботи. Будь-яке втручання зі сторони може призвести до збоїв у роботі всього пристрою.

Аукціонний алгоритм (Auction algorithm) – це алгоритм для знаходження найкоротшого шляху в орієнтованому графі. Він працює на основі імітації процесу торгів, де завдання розподіляються між учасниками через процес ставок. Кожен учасник аналізує завдання, яке може бути ним виконане та оголошує свою ставку, що відображає оцінку енергетичних витрат, необхідних для виконання цього завдання. Усі ставки передаються до аукціонера, який їх порівнює та обирає найвищу з них. В результаті учасник, який зробив найвищу ставку отримує завдання. В даному процесі можуть виникати затримки в передачі даних, оскільки є необхідність в постійній комунікації з аукціонером. Децентралізованість

алгоритму забезпечує стійкість до збоїв, оскільки процес виконання завдань не припиняється в разі виходу з ладу одного із учасників.

Основним недоліком аукціонного алгоритму є високі вимоги до обчислювальних ресурсів безпілотних засобів, оскільки всі операції розрахунків відбуваються в ньому. Застосування даного алгоритму в системі безпілотної доставки призведе до значного зниження запасу ходу пристроїв, що не є ефективним рішенням.

Метод рою частинок (PSO) — це метаевристичний алгоритм, який моделює поведінку рою для знаходження оптимальних рішень у задачах з багатьма параметрами. Кожна частинка “агент” у системі намагається знайти найкраще рішення взаємодіючи з іншими. В залежності від свого положення у просторі можливих рішень, кожна частинка коригує свій шлях у напрямку оптимального або близького до оптимального розв’язку. Даний алгоритм є простим у реалізації та вимагає меншої кількості параметрів для налаштування. Він здатний обробляти великі простори рішень та адаптуватись до змін, що робить його ефективним в задачах динамічного розподілу завдань.

До недоліків даного алгоритму входить обмежена здатність до глобальної оптимізації при складних задачах та неможливість враховувати велику кількість обмежень. Дані недоліки є критичними для систем з різними типами та характеристиками безпілотних засобів.

Генетичний алгоритм (GA) – це метод оптимізації який імітує процес еволюції для знаходження оптимальних рішень у складних задачах. Принцип роботи даного алгоритму полягає в послідовному застосуванні процесів природного відбору, а саме операцій схрещування та мутацій. Даний алгоритм є ефективним у пошуку глобального оптимального рішення та здатним уникати локальних мінімумів. GA може вирішувати завдання з великою кількістю обмежень і параметрів. А також адаптуватись до змін у середовищі.

До недоліків генетичного алгоритму варто віднести обчислювальну складність, оскільки для вирішення великих задач потребує великої кількості обчислень та ітерацій.

Даний недолік вирішується перенесенням усіх обчислень на центральний сервер, що забезпечить ефективний розподіл завдань та мінімальне навантаження на безпілотники.

2.2. Методи визначення енергетичних витрат безпілотних пристроїв

Для оцінки енергетичних витрат безпілотного пристрою необхідно визначити чинники які впливають на енерговитрати пристрою. У роботі [15] запропоновано модель підрахунку енерговитрат для безпілотних літальних пристроїв у міській логістиці, на основі рівноваги Ньютона в умовах стаціонарного польоту (формула 2.1).

$$\min E_{t_j}^i = \frac{1}{\eta}\left(c_d^i \rho A_d^i \left(L_{1_j}^i + L_{2_j}^i\right) {v_j^i}^2 + \frac{L_{1_j}^i \sqrt{\left(\left(m^i + m_{p_j}\right)g\right)^3}}{v_j^i \sqrt{2\rho A_r^i}} + \frac{L_{2_j}^i \sqrt{\left(\left(m^i + m_{p_j}\right)g\right)^3}}{v_j^i \sqrt{2\rho A_r^i}} \right) \quad (2.1)$$

де $E_{t_j}^i$ – мінімальна енергія використана i-им безпілотником для виконання j-го завдання з поточного місця розташування пристрою;

η – коефіцієнт ефективності обчислення енергії;

c_d^i – коефіцієнт лобового опору безпілотника;

ρ - густина повітря;

A_d^i – поперечний переріз безпілотника відносно його руху;

$L_{1_j}^i$ – мінімально-ризикована дистанція від поточних координат дрона до місця підбору посилки;

$L_{2_j}^i$ – мінімально-ризикована дистанція від точки підбору до місця призначення;

m^i – вага i-го безпілотника;

m_{p_j} – вага j посилки;

v_j^i – швидкість руху безпілотника;

A_r^i – загальна площа ділянки пропелера ротора;

g – прискорення вільного падіння.

Дана математична модель не враховує енергетичні витрати при злеті чи приземленні та відображає мінімальні витрати енергії за ідеальних обставин. Формула 2.2 розраховує необхідну енергію для втримання дрона в повітрі [16]:

$$P = \sqrt{\frac{\left(\left(m^i + m_{p_j}\right) g\right)^3}{2\rho A_r^i}}, \tag{2.2}$$

Цих даних буде цілком достатньо аби оцінити енергоефективність роботи системи. Оскільки при реальних умовах дані споживання енергії будуть відправлятись безпосередньо кожним пристроєм, що забезпечить точніший аналіз та кращий розподіл завдань.

2.3. Аналіз існуючих методів динамічного розподілу завдань

Динамічний розподіл завдань є однією з ключових задач у комп’ютеризованій системі безпілотної доставки. Існує низка методів та підходів для вирішення цієї задачі, спрямованих на забезпечення динамічності та ефективності у розподілі завдань між безпілотниками в умовах обмежених ресурсів та змінних сценаріїв. Одним із сучасних підходів, що демонструє високу ефективність у вирішенні таких задач є генетичний алгоритм.

У роботі [18] розглянута проблема оптимального розподілу завдань розвідки для групи безпілотних літальних апаратів (БПЛА). Дослідження розглядає генетичний алгоритм на основі протилежних хромосом з подвійним кодуванням та множинних операторів мутації (OGA-DEMMO). Даний алгоритм було використано для знаходження ефективного розподілу завдань, який мінімізує витрати часту та ресурсів БПЛА. Обраний підхід комбінує переваги генетичних алгоритмів, які здатні опрацьовувати великі простори рішень. Застосування методу протилежних хромосом дозволяє прискорити процес оптимізації та уникнути локальні оптимуми. Результати експериментів показують, що даний підхід дозволяє зменшити час виконання завдань і підвищити ефективність використання ресурсів БПЛА.

Дослідження [6] спрямоване на вирішення задачі оптимального розподілу завдань для групи дронів у міських умовах. Яке полягає у врахуванні обмежень, таких як енергетична ємність батареї, вантажопідйомність та часові рамки для виконання завдань. Для мінімізації загальних енергетичних витрат групи дронів застосовано генетичний алгоритм OGA-DEMMO. Оскільки така задача є NP-складною, яку можна розв’язати за недетермінований поліноміальний час. У роботі запропоновано підхід, який дозволяє врахувати вагові обмеження та коефіцієнт попадання у часові рамки під час розподілу завдань. Також алгоритм передбачає можливість додавання завдань для підзарядки дронів у випадку, якщо в них недостатньо ємності для виконання наступного завдання (рис. 2.3.).

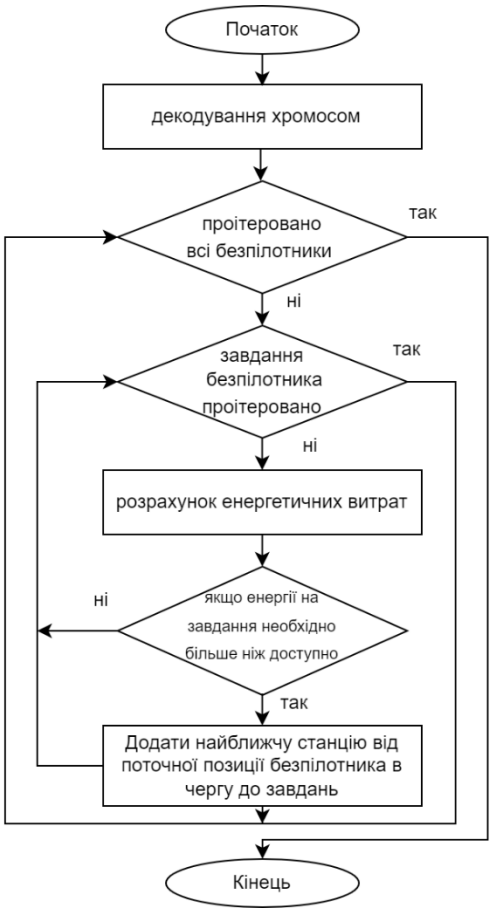

Рис. 2.3. Блок-схема роботи алгоритму додавання завдань для підзарядки

Оскільки черга виконання завдань відома заздалегідь, даний алгоритм перевіряє чи достатньо ємності батареї кожного дрона для завершення всіх завдань. В разі, якщо ємності буде недостатньо для виконання наступного завдання, алгоритм знайде найближчу зарядну станцію та додасть її в чергу перед тим завданням. Зазначено, що даний алгоритм розрахований на сценарії де зарядка відбувається заміною акумуляторної батареї безпілотного пристрою.

Результати симуляцій показали скорочення енергетичних витрат, особливо коли завдання для підзарядки додаються після операції розподілу [6].

Наявні дослідження демонструють ефективність роботи генетичних алгоритмів у задачах оптимального розподілу з врахуванням різноманітних обмежень. А також вказують на потреби для їхнього вдосконалення, а саме необхідності враховувати різні типи і характеристики безпілотних засобів та їхні робочі зони.

2.4. Застосування генетичного алгоритму в системах безпілотної доставки

Згідно аналізу наявних алгоритмів для запропонованої комп'ютеризованої системи розподілу завдань, найкращим є генетичний алгоритм. Для розрахунку оптимального розподілу завдань між безпілотними пристроями в сценарії, де N_U – кількість безпілотників та N_T - кількість завдань, застосовується генетичний алгоритм на основі протилежних хромосом з подвійним кодування та множинних операторів мутації (OGA-DEMMO). В даному алгоритмі генерується покоління хромосом, одна частина якої описує розподіл завдання між безпілотниками, а інша – оптимальні маршрути виконання цих завдань. Графічне представлення принципу подвійного кодування хромосом зображене на рис. 2.4.

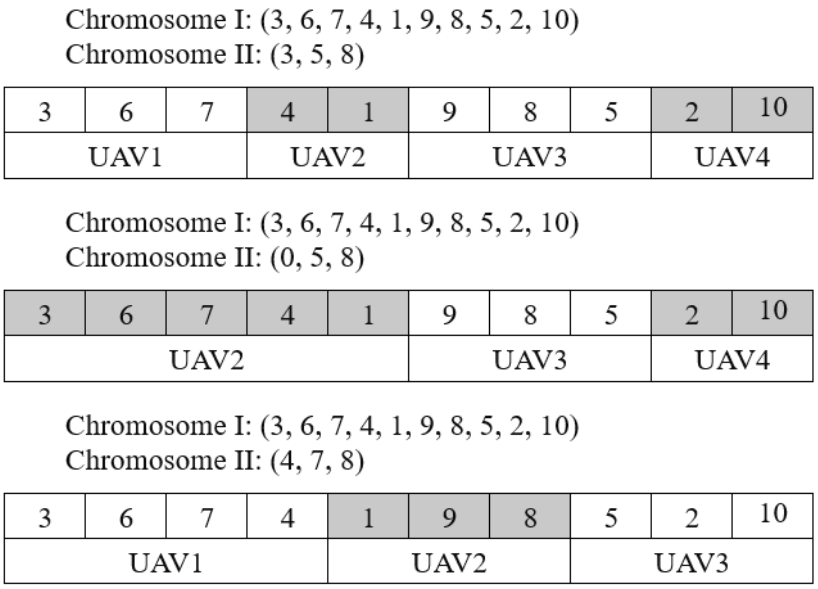

Рис. 2.4. Принцип подвійного кодування хромосоми

На рисунку зображені три приклади розподілу завдань між безпілотними пристроями. На першому прикладі хромосома I має перелік завдань (3, 6, 7, 4, 1, 9, 8, 5, 2, 10), друга хромосома в свою чергу описує межі розподілу першої хромосоми (3, 5, 8). Це означає що перша хромосома матиме три лінії поділу: перша між завданням 7 і 4, друга між 1 і 9 та третя між 5 і 2. Це створює наступну групу завдань (3, 6, 7), (4, 1), (9, 8, 5), (2, 10). Після цього кожна група завдань

присвоюється відповідному безпілотному пристрою який розраховує енергетичні витрати на кожне із завдань в тому порядку якому він його отримав.

Блок-схема OGA-DEMMO алгоритму зображена на рис. 2.5.

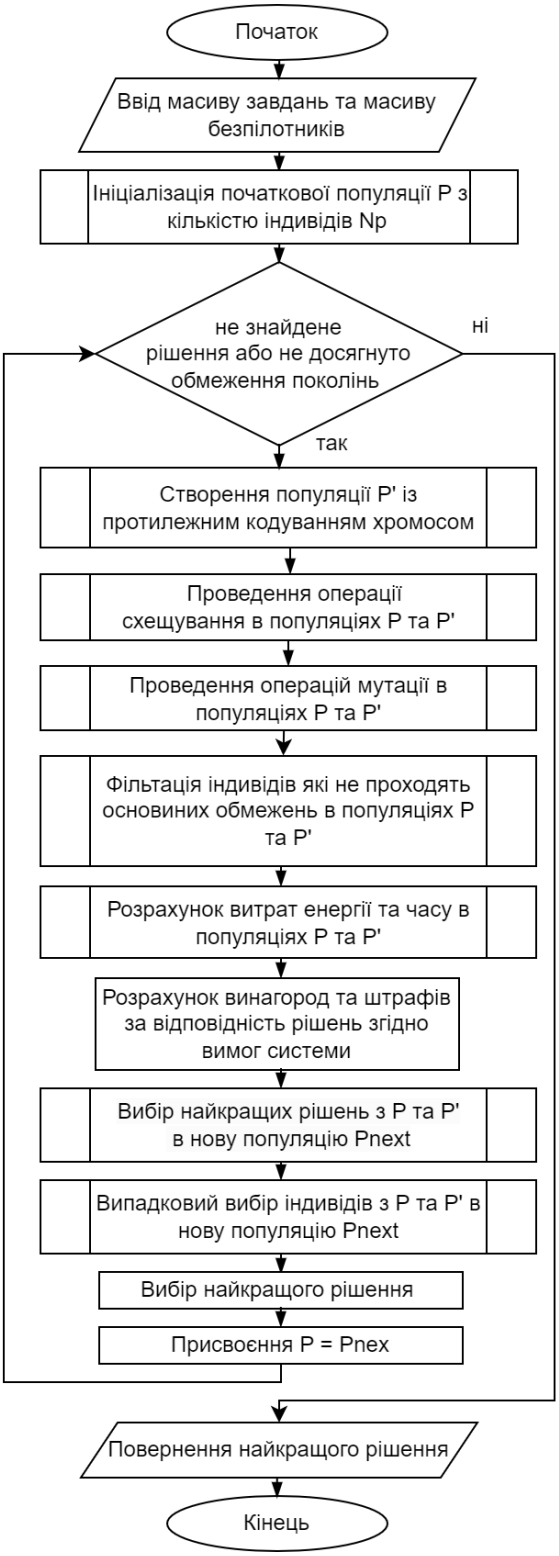

Рис. 2.5. Блок-схема OGA-DEMMO алгоритму

Варто зазначити що порядок генів в другій хромосомі має бути відсортований в зростаючому порядку, а значення перетину повинні бути в межах N_T. Кількість генів в другій хромосомі рівна $N_U - 1$, для забезпечення повноцінного розподілу між усіма безпілотниками N_U. Також для забезпечення коректних розрахунків алгоритму перша хромосома повинна містити тільки унікальні значення генів, оскільки вони відповідають індексу завдання, а загальна кількість генів не повинна перевищувати N_T.

2.4.1. Додавання протилежної популяції. Основною метою створення протилежної популяції (opposition-based learning) є створення протилежної за кодуванням ген індивіда, щоб підвищити ймовірність знаходження кращих рішень. Дана методика основана на реальних спостереженнях світу, де протилежністю слабкого індивіда є сильний.

Логіка роботи в даному алгоритмі, базується на визначенні протилежного коду для кожного гена першої хромосоми. Для кожного z гена в інтервалі від $[a, b]$, протилежний код z' визначається наступною формулою:

$$z' = a + b - z, \qquad z' \in [a, b], \tag{2.13}$$

де z' - протилежний код нового гена

Для першої хромосоми [1, 7, 9, 2, 5, 3, 6, 4, 8], найменше та найбільші значення яким належать індекси завдань це 1 і 9 відповідно. Тоді хромосома із протилежним кодуванням матиме вигляд [9, 2, 1, 7, 4, 6, 3, 5, 2]. Друга хромосома в даній операції не змінюється.

2.4.2. Операції схрещування. Однією із найважливіших етапів в генетичному алгоритмі є операція схрещування. Оскільки вона відповідає за обмін інформації між хромосомами при створенні нових популяцій. Дана операція відіграє важливу роль у продуктивності алгоритму. Ефективний генетичний алгоритм потребує

детального вибору операції схрещування, для досягнення оптимального результати за відносно короткий час.

У роботі [17], досліджено проблему оптимального розподілу завдань між командами агентів. Особлива увага приділяється операціям схрещування в генетичному алгоритмі. В досліджені проаналізовано велику кількість операцій схрещування та запропоновано власні рішення.

В ході експериментальних досліджень найефективнішою виявилась операція схрещування на основі позиції зі списком перестановок (position-based shuffled list crossover operator). Результати демонструють, що даний метод значно перевищує інші операції схрещування в якості рішень та здатності уникати локальних оптимумів. Приклад роботи даної операції зображено на рис. 2.6.

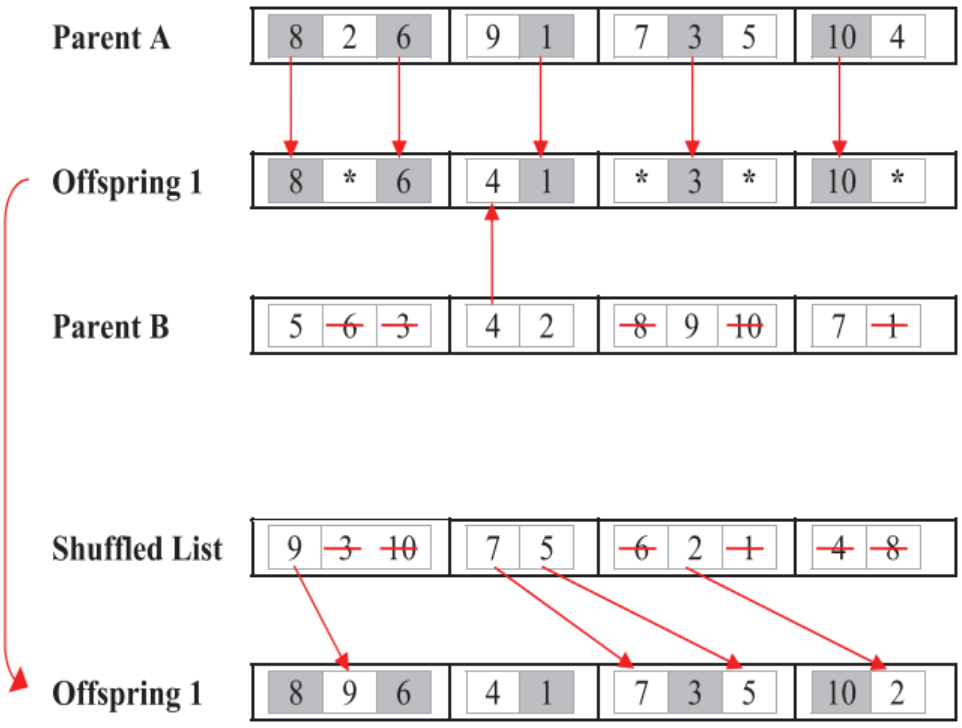

Рис. 2.6. Принцип роботи операції схрещування на основі позиції зі списком перестановок [17]

В даному механізмі схрещування, у випадковому порядку (з ймовірністю 0.5) з хромосоми Parent A обираються позиції генів. Обрані гени копіюються в

хромосому нащадок Offspring 1. Значення генів які були обрані з хромосоми Parent A, видаляються з хромосоми Parent B, а решта генів копіюються у відповідні гени нащадка Offspring 1. У незаповнені гени нащадка (позначені *), змішаним порядком копіюються незадіяні гени з хромосоми Parent B.

Автори також зазначають, що для забезпечення найкращих результатів ймовірність вибору генів від одного з батьків повинна бути налаштована в діапазоні 70-80% для досягнення оптимального балансу між дослідженням нових рішень та покращенням поточних.

2.4.3. Операції кількісної мутації. Операція мутації хромосоми є надзвичайно важливим механізмом в генетичному алгоритмі, оскільки вона запобігає передчасному застряганню на локальних оптимумах. Для підвищення різноманітності популяції в OGA-DEMMO алгоритмі застосовуються декілька операцій мутації. Варто зазначити що даний алгоритм містить дві хромосоми тому вони мають різні механізми мутації.

Для першої хромосоми, яка відповідає за послідовність виконання завдань застосовуються операції перевернутого ряду генів (flip), заміни генів місцями (swap) та ковзання ряду генів (slide)

Мутація типу flip в випадковому порядку вибирає дві позиції в першій хромосомі. Тоді робить інверсію вибраної ділянки. Приклад принципу flip мутації зображено на рис. 2.7.

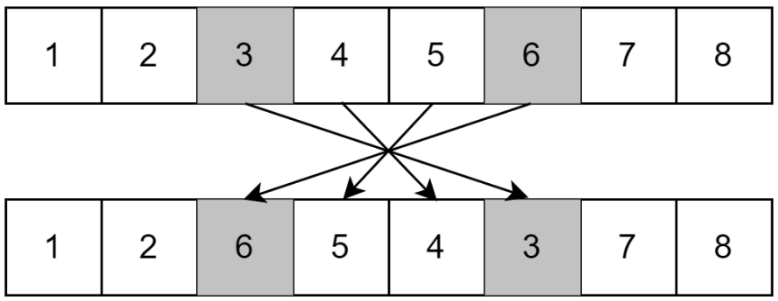

Рис.2.7. Приклад операції flip

Мутація типу slide також в випадковому порядку вибирає дві позиції в першій хромосомі. Наступним кроком переміщує перший елемент на місце останнього та зміщує ділянку в сторону першого елементу. Приклад принципу slide мутації зображено на рис. 2.8.

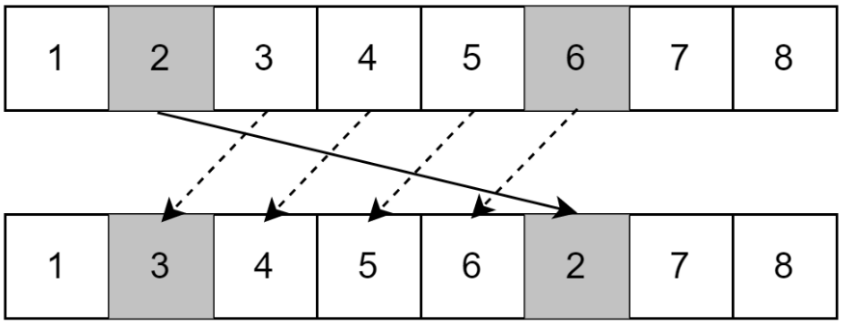

Рис. 2.8. Приклад операції slide

Swap мутація є найпростішою, де в випадковому порядку обираються дві позиції після чого вони обмінюються своїми значеннями. Приклад принципу swap мутації зображено на рис. 2.9

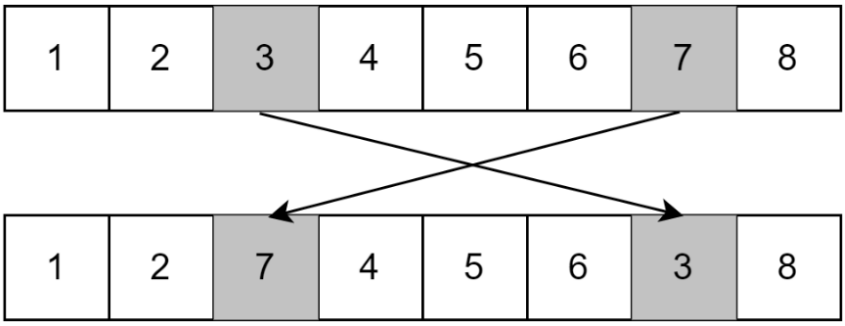

Рис. 2.9. Приклад операції swap

Для другої хромосоми відбувається механізм регенерації. Даний механізм в випадковому порядку регенерує нові значення для кожного гена. Варто зазначити що тут також необхідно врахувати межі, які не повинні перевищувати значення N_T.

2.5. Висновки до розділу

У другому розділі кваліфікаційної роботи були розглянуті існуючі методи динамічного розподілу завдань їхні особливості та недоліки. Було проаналізовано методи вимірювання енергоспоживання безпілотних пристроїв з врахуванням впливу додаткової ваги на споживання енергії.

Особлива увага була присвячена новим алгоритмам та підходам до розподілу завдань з врахуванням різноманітних обмежень, а саме вагових і енергетичних та підходів з додаванням можливості підзарядки.

В результаті проведеної роботи над другим розділом, було отримано важливі відомості про існуючі методи динамічного розподілу завдань в комп’ютеризованій системі безпілотних пристроїв та потенціал застосування генетичних алгоритмів. Ці дані стануть основою для розробки та реалізації методу динамічного розподілу завдань з врахуванням кожного типу пристрою, його характеристик та робочої зони.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ОПТИМАЛЬНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНІЙ СИСТЕМІ

3.1. Врахування типу безпілотників та зони їхньої досяжності

Зважаючи на тенденцію появи нових безпілотних засобів, їхні технічні характеристики можуть відрізняються один від одного. Більшість із засобів мають безпечні робочі зони в яких вони здатні виконувати поставленні завдання. Для ефективного розподілу завдань в такій системі безпілотної доставки, зменшення витрат енергії та підвищення ефективності запропоновано враховувати тип і характеристики пристроїв та їхню робочу зону у цьому процесі.

В даному досліджені враховано, що кожен безпілотний пристрій доставки має свою робочу зону, яка обмежується досяжністю пристрою та зарядними станціями. Такий сценарій дозволить ефективніше розподіляти засоби в зонах з великою кількістю завдань. Та забезпечити пристрій від розряду або поломки в непередбачуваних для нього ділянках.

Для вирішення даної проблеми запропоновано використати граф зв'язків на основі алгоритму Дейкстри (рис. 3.1.). Даний граф зберігатиме усі локальні робочі зони безпілотників, а алгоритм дозволить швидко знайти зони, які пересікаються. Це дасть можливість ефективніше локалізувати робочі ділянки безпілотників, що забезпечить ефективнішу роботу системи оптимального розподілу завдань.

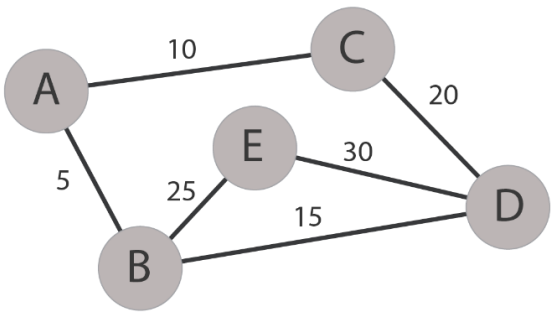

Рис. 3.1. Вигляд графу зв'язків

Після кожного додавання чи видалення робочої зони граф зв’язків перебудовується. Це здійснюється методами сутності StationGraph (див. рис. 3.3).

3.2. Архітектура комп’ютеризованої системи безпілотної доставки

Для ефективної роботи, система оптимального розподілу завдань повинна враховувати типи пристроїв, їхні характеристики, робочі зони, вагові обмеження, поточні координати та стан у якому перебуває кожен з них. Дана система розрахована на взаємодію із центрами керування безпілотних пристроїв. Запропонована архітектура такої взаємодії зображена на рис. 3.2.

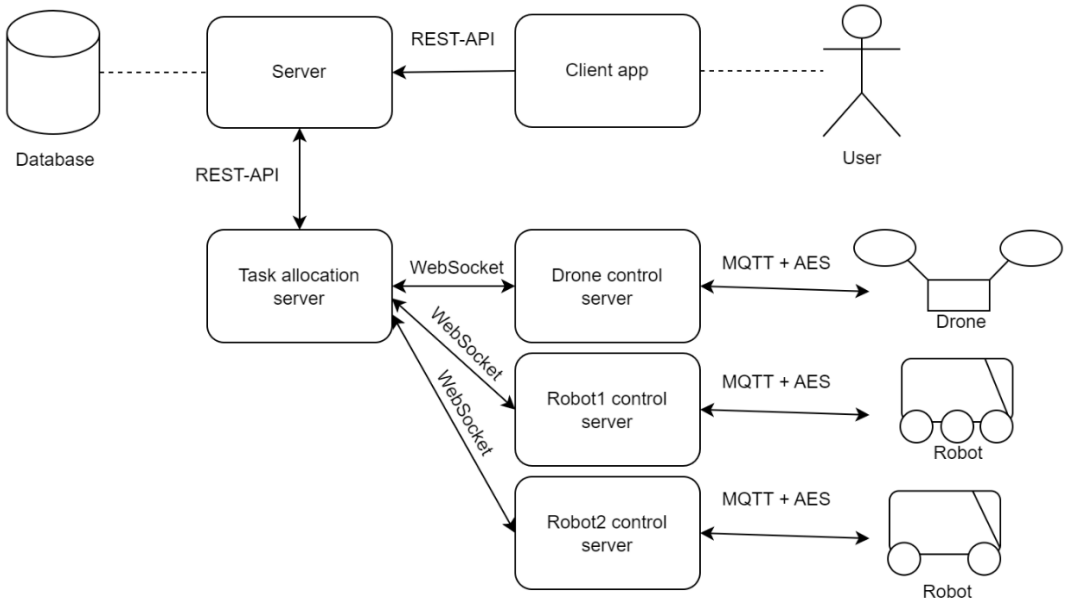

Рис. 3.2. Архітектура комп’ютеризованої системи безпілотної доставки

Запропонована архітектура передбачає, що сервер оптимального розподілу завдань буде взаємодіяти безпосередньо із керуючими серверами кожного з типів безпілотників. Така взаємодія є досить реалістичною, оскільки розглянута частина існуючих засобів безпілотної доставки передбачає можливості інтеграції з їхніми сервісами. Такий підхід також є важливим з міркувань безпеки, оскільки кожен керуючий сервер має сертифіковані методи координації та визначення оптимального шляху для безпілотного засобу.

3.3. Структура системи оптимального розподілу завдань

Система оптимального розподілу завдань в комп'ютеризованій системі безпілотної доставки спроектована таким чином, аби забезпечити можливість її масштабування для ефективнішого розподілу ресурсів. Схематичне представлення предметної області розробленої системи, зображене на рис. 3.3.

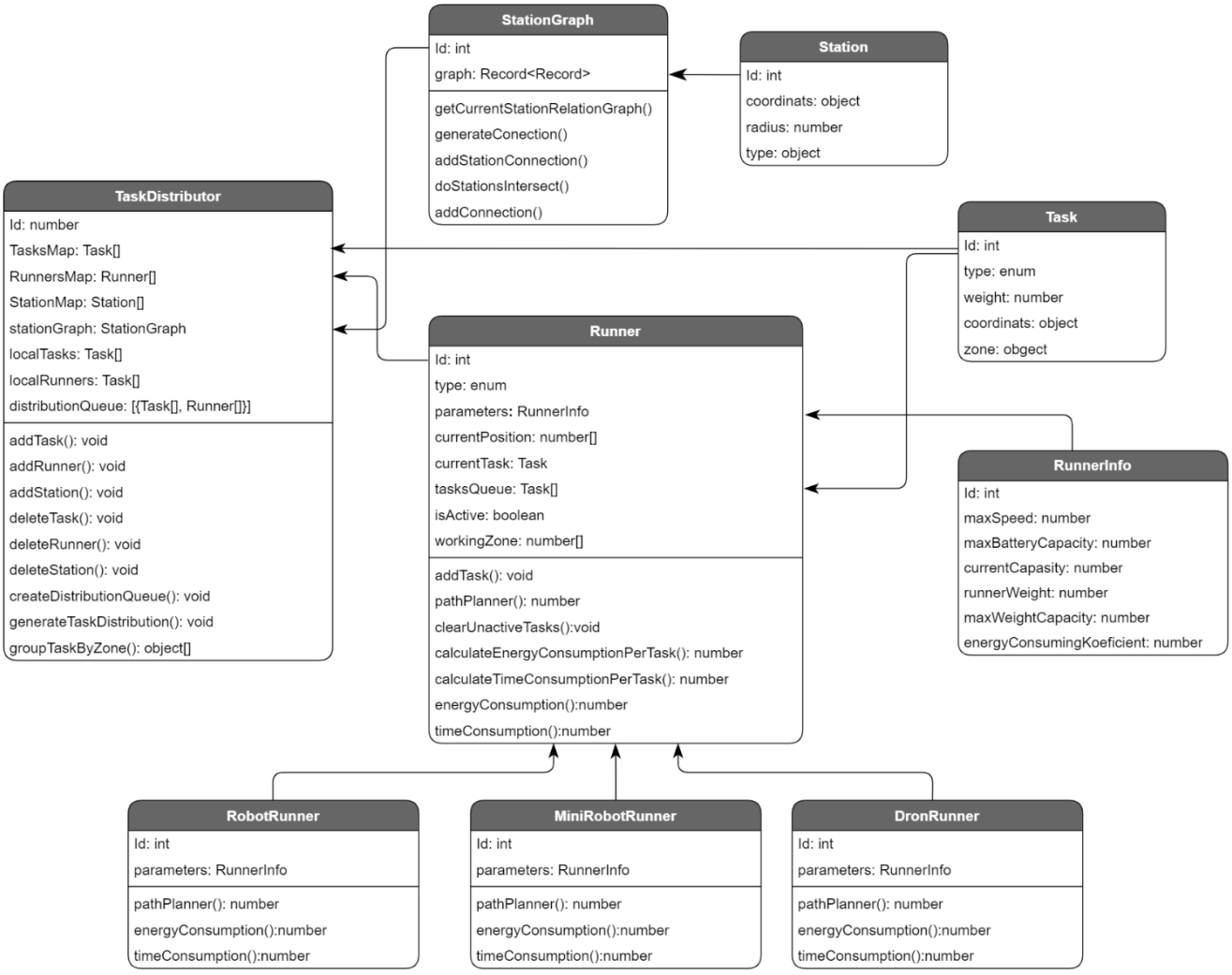

Рис. 3.3. Предметна область системи оптимального розподілу завдань

Розроблена система оптимального розподілу завдань складається з наступних основних компонентів:

– Цент розподілу завдань (TaskDistributor);

– Безпілотник (Runner);

– Граф робочих зон (StationGraph);

– Робоча зона (Station);

– Завдання (Task);

– Інформація безпілотника (RunnerInfo);

– Робот-кур’єр 1(RobotRunner);

– Робот-кур’єр 2 (MiniRobotRunner);

– Дрон (DronRunner).

Система оптимального розподілу завдань була спроектована таким чином, щоб вона могла працювати з різними типами безпілотних пристроїв з мінімальною кількістю змін.

Сутність TaskDistributor відповідає за зберігання основної інформації про безпілотні засоби, їхні робочі зони, статуси та завдання. Дана сутність виконує функціонал додавання нових пристроїв, завдань, робочих ділянок, виконує локалізацію завдань згідно робочих зон та вибір засобів для розподілу завдань.

Сутність Runner відповідає за безпілотний засіб для доставки. Вона містить усі необхідні характеристики пристрою, а саме його тип, стан, поточні координати, завдання яке виконується (за його наявності), перелік завдань які стоять в черзі на виконання та робочу область до якої належить пристрій. Функціонал даної сутності включає:

– побудову оптимального шляху для доставки;

– розрахунок енергетичних витрат для виконання завдання;

– розрахунок часових витрат для виконання завдання.

Runner виступає в ролі батьківської сутності та може мати ряд похідних екземплярів. Це забезпечує єдину структуру у взаємодії з усіма екземплярами батьківського компонента. Оскільки кожен пристрій має свої фізичні та функціональні характеристики, похідні екземпляри можуть перевизначити всі необхідні параметри.

Сутності RobotRunner, MiniRobotRunner, DronRunner, представляють собою вищезгадані похідні екземпляри. Вони наслідують усі базові характеристики з класу Runner та перевизначають ключові параметри, а саме:

– основні характеристики;

– функцію побудови оптимального шляху для доставки;

– функцію розрахунку енергетичних витрат для виконання завдання;

– функцію розрахунку часових витрат для виконання завдання.

Основні характеристики похідних екземплярів Runner зберігаються в сутності RunnerInfo, а також за необхідності можуть бути доповненими в разі необхідності. До основних атрибутів RunnerInfo входять:

– швидкість переміщення (maxSpeed);

– ємність батареї (maxBatteryCapacity);

– поточна ємність батареї (currentCapasity);

– вага безпілотника (runnerWeight);

– максимальна вага навантаження (maxWeightCapacity);

– коєфіцієнт енергетичних витрат (energyConsumingKoeficient).

Сутність StationGraph відповідає за локалізацію робочих зон кожного із безпілотних засобів. При додаванні робочої зони того чи іншого типу безпілотників, створюється відповідний зв'язок в графі (graph). Даний граф зберігає усі робочі ділянки та їхні зв’язки, якщо ділянки перетинаються. Дана сутність дозволяє швидко отримати групи робочих зон для швидкої локалізації вхідних завдань.

Сутність Station зберігає характеристики робочих ділянок, а саме координати, радіус дії, та тип безпілотника на який вона поширюється.

Остання сутність Task містить інформацію про завдання з доставки. Дана сутність також описує дані про посилку, а саме координати місця підбору і доставки, її вагу та робочу зону якій вона належить. В разі, якщо координати знаходяться за межами любої з доступних робочих зон, дане завдання не братиметься до обрахунків.

3.4. Розробка алгоритмічного та програмного забезпечення

3.4.1. Алгоритмічне забезпечення системи. Створення алгоритму є ключовим етапом під час розробки ПЗ. На даному етапі визначається логіка послідовності виконання операцій, для досягнення поставленої мети. Алгоритм роботи системи оптимального розподілу завдань зображений на рис. 3.4.

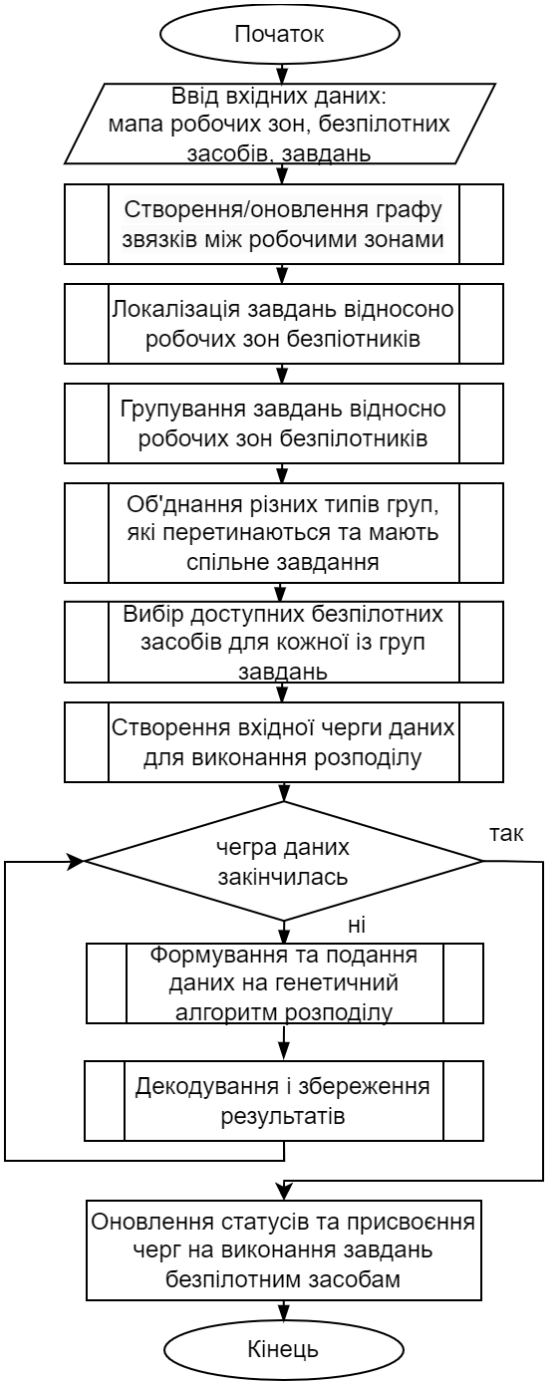

Рис. 3.4. Блок-схема алгоритму роботи системи оптимального розподілу

Процес розподілу розпочинається із отримання даних про безпілотні засоби, їхні робочі зони та наявні завдання. Вхідні дані зберігаються у відповідних мапах, які дозволяють швидко отримати необхідні дані за їхнім ідентифікатором. Під час додавання нових робочих зон, формується граф їхніх зв’язків, в даному етапі застосовується алгоритм Дейкстри для швидкого знаходження інформації про те чи мають між собою зв'язок дві робочі зони чи ні. Після цього відбувається локалізація та групування вхідних завдань відносно робочих зон. Якщо завдання належить декільком робочим зонам, то для пошуку оптимальних рішень, дані групи об’єднуються в одну.

В результаті локалізації завдань вихідним отримується масив об’єктів, який містить зону або групу зон, та масив завдань які належать даній зоні. Для кожного з них відбувається вибір доступних безпілотників. Після знаходження безпілотних засобів формується черга на виконання операції з обчислення оптимального розподілу. Дана черга, являє собою масив об’єктів, який містить масив завдань та масив ідентифікаторів безпілотних засобів.

Варто зауважити, що на цьому етапі також відбувається присвоєння ідентифікаторів робочих зон, яким належать завдання та безпілотні засоби. В об’єктах завдань зберігаються зони, яким належать координати їхнього підбору та координати місця куди необхідно їх доставити. Для безпілотних засобів зберігається масив ідентифікаторів робочих зон, яким вони належать. Це важливий етап оскільки він дозволяє швидко ідентифікувати завдання, які знаходяться поза межами робочої зони безпілотників та не витрачати час на додаткові обчислення їхньої локалізації. Ці дані необхідні під час обрахунку оптимального розподілу завдань між безпілотниками, робочі зони яких пересікаються.

Сформована черга на виконання операцій розподілу передається в цикл, який здійснює обрахунки оптимального розподілу для кожного об’єкту черги. Після обрахунків отримані результати декодуються та зберігаються. Після чого відбувається операція оновлення статусів та присвоєння черг завдань безпілотним пристроям.

Варто зауважити, що на етапі локалізації нових черг завдань, безпілотні засоби можуть мати вже закріплені за ними завдання. Це необхідно врахувати перед формуванням нової черги на виконання. Алгоритм вибору доступних безпілотників зображений на рис. 3.5.

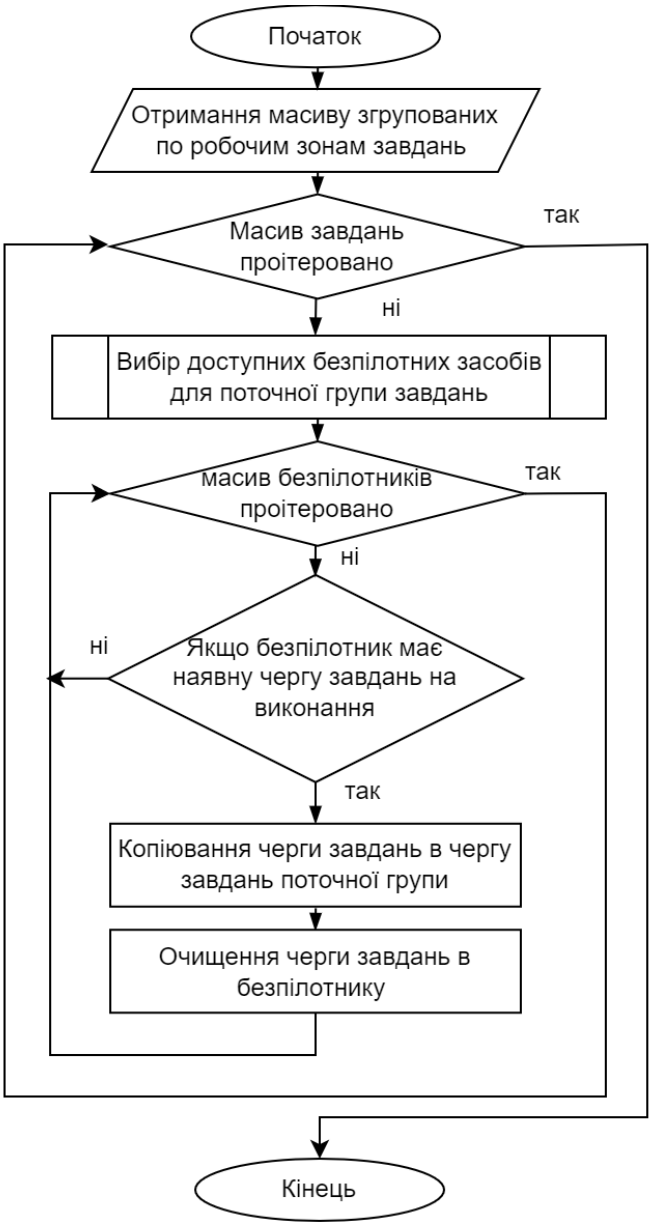

Рис. 3.5. Блок-схема алгоритму вибору доступних безпілотників

Даний алгоритм циклічно проходить через увесь масив згрупованих завдань та знаходить наявну кількість безпілотників, які належіть робочим зонам поточної групи завдань. Наступним кроком перевіряється кожен безпілотник на наявність

уже закріплених за ним завдань. В разі їхньої наявності, дані завдання копіюються в масив завдань поточної групи. Після цього відбувається очищення наявної черги завдань в екземплярі безпілотника. Варто зазначити, що стираються тільки ті завдання, які в стані “assigned”, тобто заплановані. Завдання стану “active” залишається, оскільки це означає, що безпілотник в процесі виконання даного завдання.

3.4.2. Програмне забезпечення системи. Система оптимального розподілу завдань була реалізована мовою програмування TypeScript. Проект було поділено на основні частини, а саме генетичний алгоритм і його механізми та систему розподілення завдань. Генетичний алгоритм на основі протилежних хромосом з подвійним кодування та множинних операторів мутації лістинг. Даний алгоритм реалізований в функції geneticAagorithm() вхідні параметри, якої зображені на рис. 3.6.

```
export function geneticAlgorithm(
  tasksMap: Map<number, Task>,
  runnersMap: Map<number, Runner>,
  tasks: number[],
  runners: number[],
  populationSize: number,
  maxGenerations: number,
  maxIndividualsPerGeneration: number,
  solutionDifference: number,
  timeDistributionCoef: number,
  runnerDistributionCoef: number
): TEnergyIndividual
```

Рис. 3.6. Лістинг параметрів функції генетичного алгоритму geneticAagorithm()

На вхід даної функції надходить мапа завдань (tasksMap), мапа безпілотних засобів (runnersMap), масив завдань (tasks) та безпілотників (runners) для проведення обрахунків. А також параметри процесу, такі як розмір популяції (populationSize), максимальна допустима кількість генерацій (maxGenerations), максимальна кількість індивідів в поколінні (maxIndividualsPerGeneration), умова завершення обрахунків (solutionDifference), коефіцієнт часового розподілу

(timeDistributionCoef) та коефіцієнт використання безпілотників (runnerDistributionCoef). Лістинг коду функції генетичного алгоритму представлений в додатку Б.

Після проведення розрахунків алгоритм повертає на вихід індивіда з найкращим кодуванням двох хромосом, які відображають розподіл завдань між безпілотниками. Для подальших операцій присвоєння завдань між пристроями дану хромосому необхідно декодувати. Лістинг функції декодування хромосом зображений на рис. 3.7.

```
export function decodeChromosomes(
  chromosomeI: number[],
  chromosomeII: number[]
): number[][] {
  const tasksPerDrone: number[][] = [];
  let partiotionPointA = 0;

  for (const partitionPointB of chromosomeII) {
      let  taskFordrone  =  chromosomeI.slice(partiotionPointA,
partitionPointB);

    tasksPerDrone.push(taskFordrone);

    partiotionPointA = partitionPointB;
  }

  if (partiotionPointA < chromosomeI.length) {
    tasksPerDrone.push(chromosomeI.slice(partiotionPointA));
  }

  return tasksPerDrone;
}
```

Рис. 3.7. Лістинг програми декодування хромосом

Функція декодування хромосом на виході повертає масив індексів завдань для кожного індексу безпілотника. Після чого робиться пошук відповідних безпілотників та завдань в мапі де вони зберігаються.

Програмне забезпечення системи розподілу завдань включає в себе методи локалізації вхідних завдань відповідно до робочих зон та пошук безпілотних засобів, які належать кожній із них. Ці методи реалізовані у сутності TaskDistributor. Дана сутність виступає в ролі центрального механізму, який

виконує основні операції системи. До них входять операції додавання і видалення нових завдань, безпілотників та робочих зон. А також операції локалізації вхідних завдань, їхнє групування, та створення черги вхідних даних для обрахунку оптимального розподілу.

Процес локалізації відбувається за допомогою сутності StationGraph, яка формує граф зв'язків між робочими зонами (див. рис. 3.8).

```
class StationGraph {
  private static instance: StationGraph;
  graph: Record<string, Record<string, number>> = {};
  constructor() {
    if (!StationGraph.instance) {
      StationGraph.instance = this;
    }
    return StationGraph.instance;}
  getCurrentStationRelationGraph() {
    return this.graph;}
   generateConection(newStation: Station, stations: Map<number,
Station>) {
    stations.forEach((existingStation) => {
      if (newStation.id !== existingStation.id &&
        newStation.type === existingStation.type
      ) {
        this.addStationConnection(newStation, existingStation);
      } else {
        if (!this.graph[newStation.id]) {
          this.graph[newStation.id] = {};
        }
      }
    });}
  addStationConnection(station1: Station, station2: Station) {
    const distance = calculateDistance(
      station1.coordinats,
      station2.coordinats
    );
    if (this.doStationsIntersect(station1, station2)) {
      this.addConnection(station1.id, station2.id, distance);
      this.addConnection(station2.id, station1.id, distance);
    }}
   doStationsIntersect(station1: Station, station2: Station):
boolean {
    const distance = calculateDistance(
      station1.coordinats,
      station2.coordinats
    );
    return distance <= station1.radius + station2.radius;}
  addConnection(from: number, to: number, distance: number) {
    if (!this.graph[from]) {this.graph[from] = {};}
    this.graph[from][to] = distance;
}}
```

Рис. 3.8. Лістинг коду сутності StationGraph

3.5. Аналіз результатів оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки

Для тестування розробленого методу оптимального розподілу завдань, було розроблено веб-застосунок, з використанням бібліотеки React та підключено всі необхідні компоненти для проведення моделювань. Даний застосунок містить головну панель для створення та видалення робочих зон, безпілотних засобів та завдань. Окремо було додано можливість випадкової генерації завдань з різними параметрами ваги. А також додано інтерактивну взаємодію, яка дозволяє додавати необхідні компоненти в обрані ділянки мапи за допомогою мишки. Що значно покращує процес моделювання різних сценаріїв. Вигляд застосунку зображений на рис. 3.9.

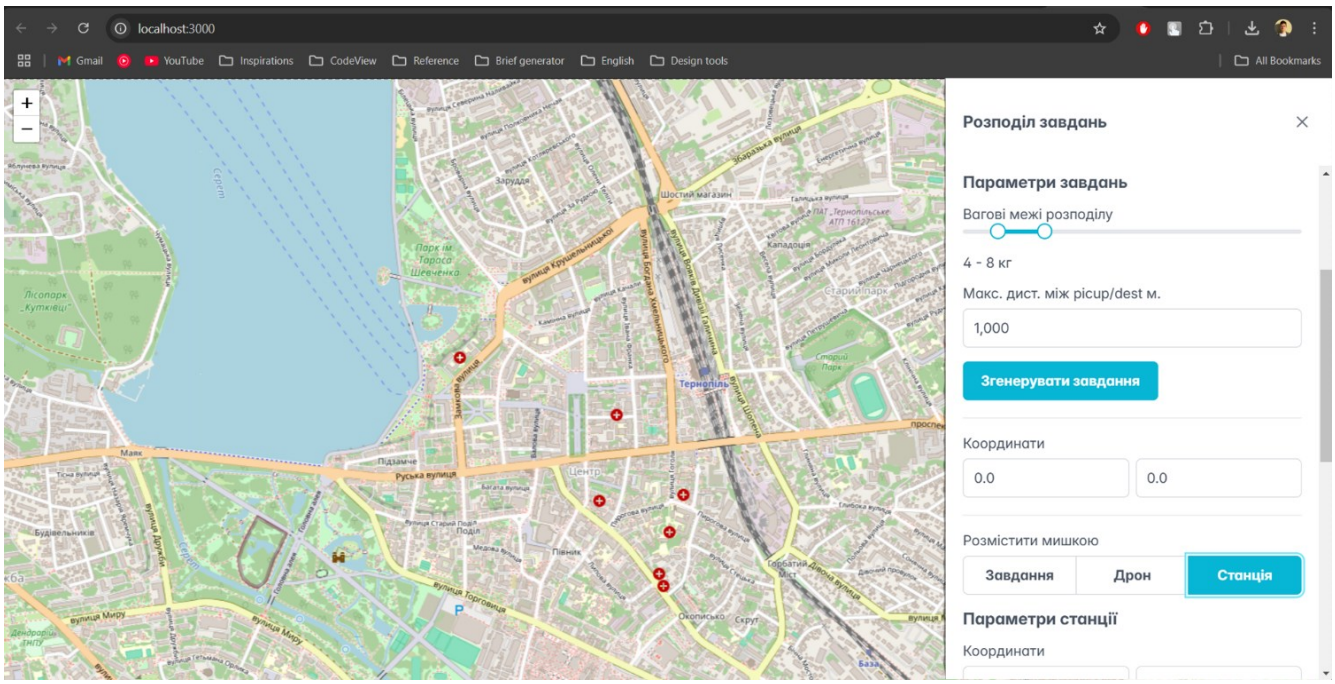

Рис. 3.9. Веб-застосунок для тестування системи оптимального розподілу

Для аналізу роботи системи оптимального розподілу було змодельовано два сценарії. Конфігурації вхідних параметрів для генетичного алгоритму подано в табл. 3.1. Дані налаштування дозволяють алгоритму краще уникати локальних оптимумів.

Таблиця 3.1.

Параметри конфігурацій генетичного алгоритму

Коефіцієнт мутації (MUTATION_RATE)	0.3
Коефіцієнт схрещування (CROSSOVER_RATE)	0.7
Розмір початкової популяції (PPULATION_SIZE)	50
Максимальна кількість ітерації (MAX_GENERATIONS)	200
Максимальна кількість індивідів в популяції (MAX_INDIVIDUALS_PER_GENERATIONS)	400
Умова зупинки обрахунків (SOLUTION_DIFFERENCE)	5 Вт
Коефіцієнт часового розподілу завдань (timeDistributionCoef)	0.5
Коефіцієнт використання безпілотних засобів (runnerDistributionCoef)	0.5

Перший сценарій полягає у розподілі 13 завдань серед 6 безпілотних засобів, які знаходяться у своїх робочих зонах, що не пересікаються. Спершу було створено 4 робочі зони для різного типу пристроїв (табл. 3.2).

Таблиця 3.2.

Параметри робочих зон

№ роб. зони	Ідентифікатор роб. зони	Тип пристрою	Радіус поширення роб. зони, м
1	P1	Robot	500
2	P2	MiniRobot	300
3	P3	MiniRobot	300
4	P4	Drone	1000

В межах кожної із зон були створені завдання у випадковому порядку. Згенеровані завдання представлені у табл. 3.3.

Таблиця 3.3.

Параметри вхідних завдань

№ завдання	Ідентифікатор завдання	Зона створення	Вага посилки, кг
1	П1	Р1	4
2	П2	Р1	5
3	П3	Р1	6
4	П4	Р1	3
5	П5	Р1	7
6	П6	Р1	8
7	П7	Р2	3
8	П8	Р3	4
9	П9	Р3	5
10	П10	Р2	6
11	П11	Р4	2
12	П12	Р4	1.5
13	П13	Р4	2
13	П14	Р4	1.5

На останньому етапі підготовки були згенеровані безпілотні засоби у відповідні робочі зони (табл. 3.4).

Таблиця 3.4.

Характеристики безпілотних засобів

№ завдання	Іденти-фікатор	Тип пристрою	Зона створення	Швидкість, м/с	Макс. вага. посилки, кг
1	Б1	Robot	Р1	1.7	10
2	Б2	Robot	Р1	1.7	10
3	Б3	MiniRobot	Р2	1.4	8
4	Б4	MiniRobot	Р3	1.4	8
5	Б5	Drone	Р4	27	4
6	Б6	Drone	Р4	27	4

Перед початком розрахунку розподілу спершу генерується черга вхідних даних, яка містить в собі розгруповані завдання відповідно до робочих зон безпілотників (див. рис. 3.10)

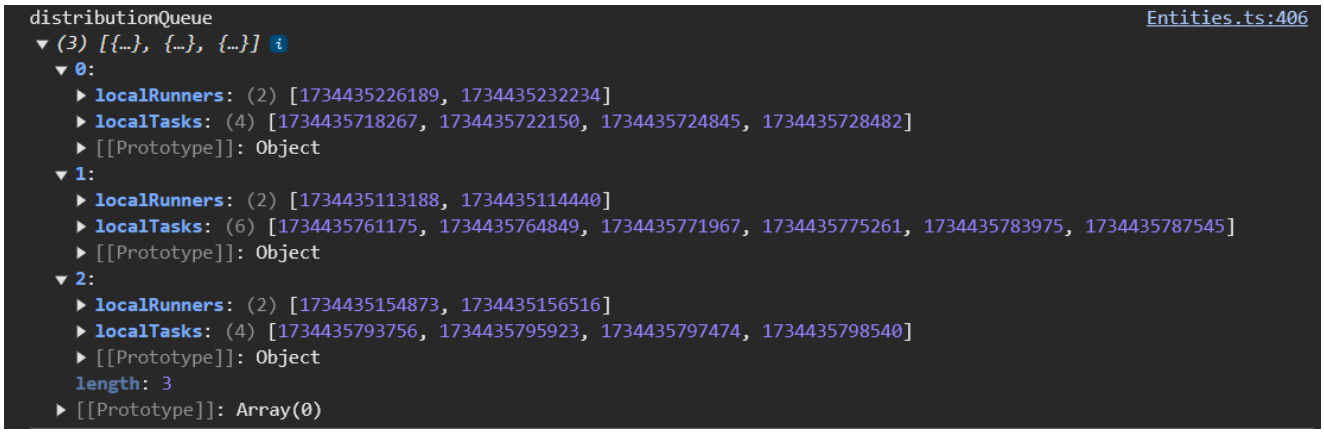

Рис. 3.10. Черга вхідних даних для обчислень оптимального розподілу

Після проведення обрахунків для кожної з груп безпілотних засобів було отримано наступний розподіл завдань (див. табл. 3.5).

Таблиця 3.5.

Результат розподілу завдань між групами

Група роб. зон	Сумарне спож. Енергії, Вт	Розподіл між безпілотниками	к-сть ітерацій
Р1	580	Б1 [П3, П1, П4], Б2[П5, П6, П2]	8
Р2, Р3	561	Б3 [П10, П7], Б4[П9, П8]	6
Р4	621	Б5 [П13, П12], Б6[П14, П11]	5

Графічне представлення результату розподілу завдань для першого сценарію з розподілом 13-ти завдань між 6-ма безпілотними засобами зображено на рис. 3.11.

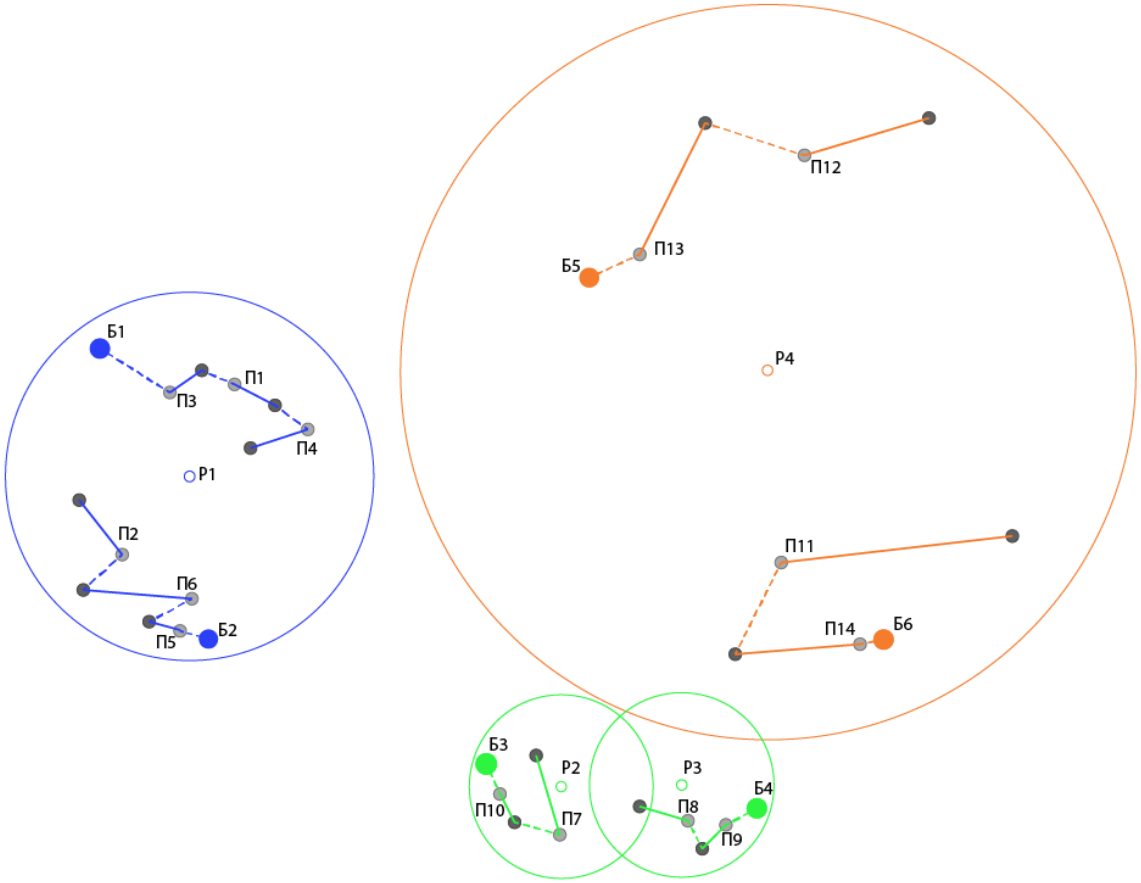

Рис.3.11. Графічне представлення розподілу завдань першого сценарію

Процес знаходження оптимального розподілу завдань кожної ітерації генетичного алгоритму зображено на рис. 3.12.

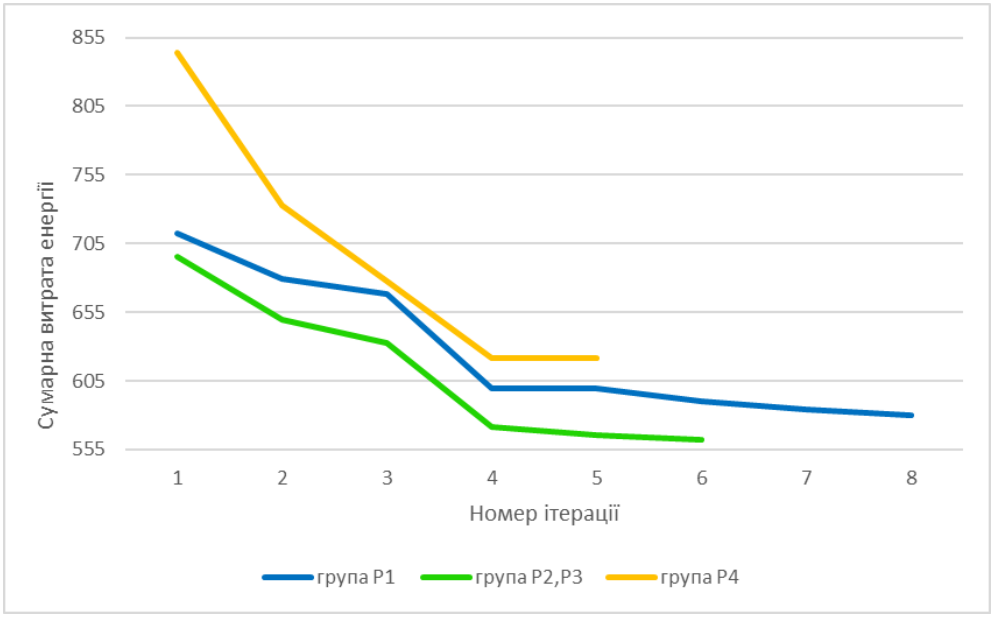

Рис. 3.12. Процес пошуку оптимального рішення для кожної з груп

Другий сценарій полягає у розподілі 12 завдань серед 4 безпілотних засобів. Але в цьому випадку робочі зони пересікаються (табл. 3.6).

Таблиця 3.6.

Характеристики робочих зон

№ роб. зони	Ідентифікатор роб. зони	Тип пристрою	Радіус поширення роб. зони, м
1	Р1	Robot	600
2	Р2	MiniRobot	300
3	Р3	MiniRobot	300

Згенеровані випадковому порядку завдання представлені у табл. 3.7.

Таблиця 3.7.

Характеристики вхідних завдань

№ завдання	Ідентифікатор завдання	Зона створення	Вага посилки, кг
1	П1	Р1, Р2	4
2	П2	Р1, Р2	5
3	П3	Р1, Р2	6
4	П4	Р1, Р2, Р3	3
5	П5	Р1, Р3	7
6	П6	Р1, Р3	8
7	П7	Р1	3
8	П8	Р1	4
9	П9	Р1	5
10	П10	Р1	6
11	П11	Р1	5
12	П12	Р1	7

Наступним кроком були згенеровані безпілотні засоби у відповідні робочі зони (табл. 3.8).

Таблиця 3.8.

Характеристики безпілотних засобів завдань

№ завдання	Іденти-фікатор	Тип пристрою	Зона створення	Швидкість, м/с	Макс. вага. посилки, кг
1	Б1	Robot	Р1	1.7	10
2	Б2	Robot	Р1	1.7	10
3	Б3	MiniRobot	Р2, Р1	1.4	8
4	Б4	MiniRobot	Р3, Р1	1.4	8

Після проведення обрахунків було отримано наступний розподіл завдань, який представлений у табл. 3.9.

Таблиця 3.9.

Результат розподілу завдань між групою Р1, Р2, Р3

Група роб. зон	Сумарне спож. енергії, Вт	Розподіл між безпілотниками	к-сть ітерацій
Р1 Р2, Р3	817	Б1 [П8, П7, П12], Б2[П9, П11, П10] Б3 [П2, П3, П4], Б3[П5, П6, П1]	45

Графічне представлення результату розподілу завдань для другого сценарію з розподілом 12-ти завдань між 4-ма безпілотними засобами зображено на рис. 3.12.

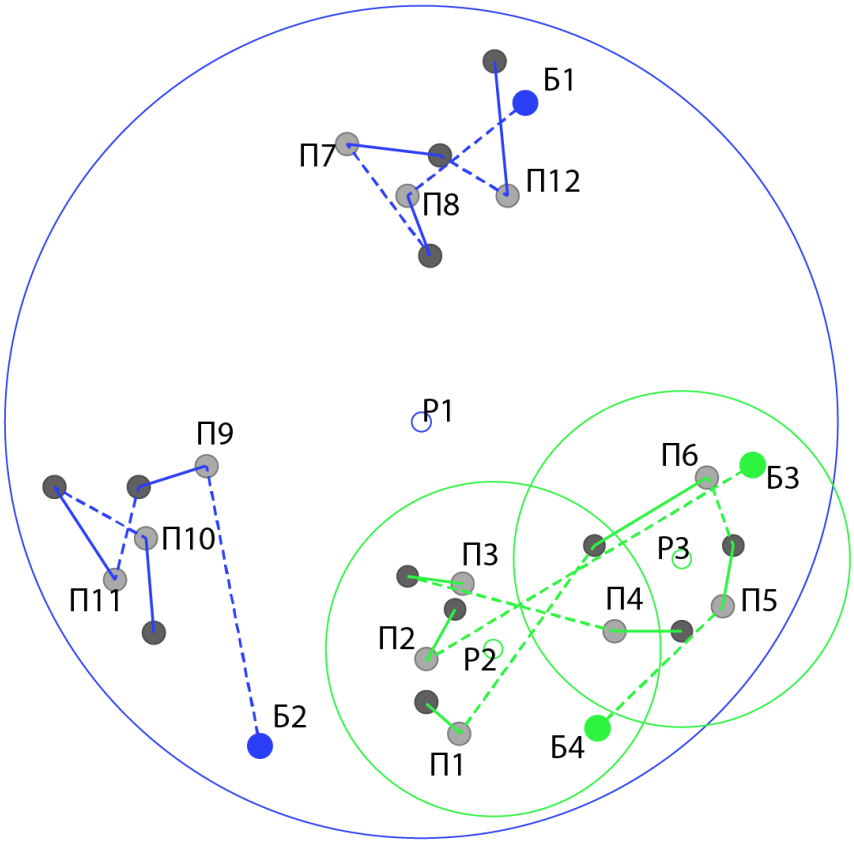

Рис. 3.13. Графічне представлення розподілу завдань другого сценарію

Процес знаходження оптимального розподілу завдань кожної ітерації генетичного алгоритму зображено на рис. 3.14.

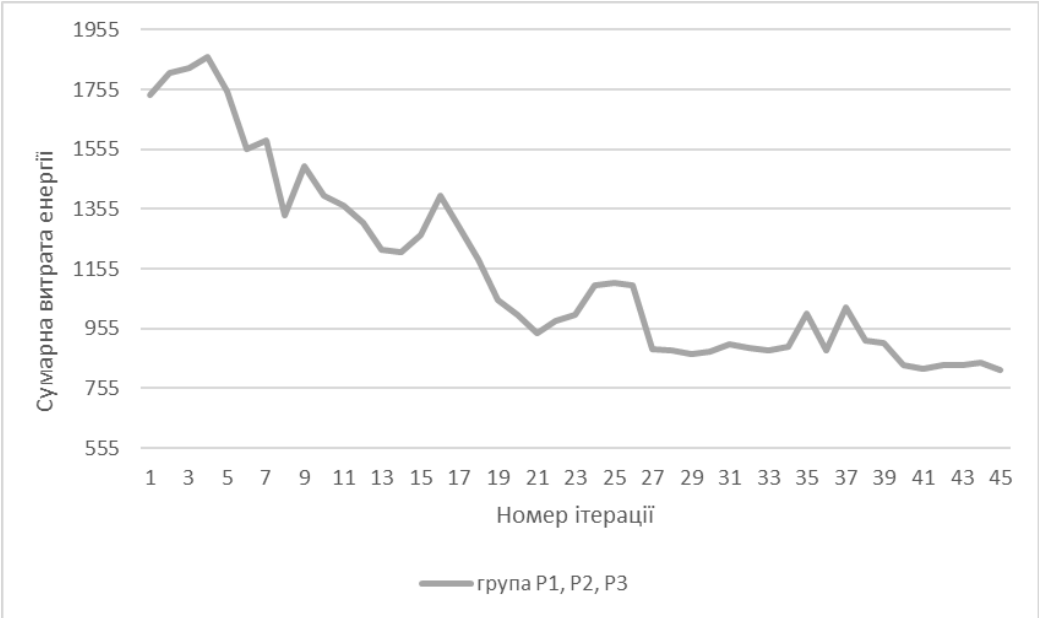

Рис. 3.14. Процес пошуку оптимального рішення кожної групи

Пошук оптимального розподілу для другого сценарію здійснив 45 ітерацій, що вказує на більшу складність завдання. На початку графіку (рис. 3.14) спостерігається швидке зниження значень енерговитрат, що вказує на ефективний пошук оптимальних рішень. Коливання, які відбуваються на середині графіку відображають широке дослідження простору рішень та уникнення локальних мінімумів. З наближенням до 45 ітерації він стабілізується, що свідчить про наближення до глобального оптимуму.

Згідно проведених тестувань роботи системи оптимального розподілу видно, що вона здатна знаходити оптимальні рішення розподілу завдань у різних сценаріях, включаючи випадки з накладанням робочих зон різних безпілотних засобів.

3.6. Висновки до розділу

У третьому розділі кваліфікаційної роботи магістра були проведені та завершені основні етапи проєкту. Представлено архітектуру комп’ютеризованої системи безпілотної доставки. Описано взаємодію між сервером розподілення задач та центрами керування безпілотних засобів доставки. Побудовано структуру даної комп’ютеризованої системи.

Було реалізовано програмне забезпечення для оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки на основі генетичного алгоритму. Воно враховує типи безпілотних засобів та їхню робочу зону. Розроблено веб-застосунок для моделювання процесу розподілу завдань та проведення аналізу роботи розробленої системи.

Представлено результати тестування та роботи оптимального розподілу завдань під час моделювання двох сценаріїв.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Мета кваліфікаційної роботи магістра полягає у розробці методів та засобів оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки. Оскільки, такі алгоритми передбачають використання комп’ютерної техніки, зокрема ПК та серверного обладнання, необхідно забезпечити дотримання вимог охорони праці, техніки безпеки та протипожежної безпеки при використанні ПК.

Основними регламентуючими нормативними документами охорони праці користувачів ПК є:

- НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями»;

- ДБН В.2.5-56:2014 «Системи протипожежного захисту»;

- Примірна інструкція з охорони праці під час експлуатації електронно-обчислювальних машин, затверджена наказом Міністерства доходів і зборів України від 5 вересня 2013 р. № 443.

Вимоги до приміщень, згідно з законом України «Про охорону праці» стаття 13 «Управління охороною праці та обов’язки роботодавця», щодо розташування робочого місця передбачають виконання наступних вимог:

- мінімальна площа, яка виділяється на одне робоче місце повинна становити мінімум 6,0 $м^2$, при об’ємі – мінімум 20,0 $м^3$;

- розташування робочих місць користувачів ПК заборонено у цокольних або підвальних приміщеннях.

Для забезпечення дотримання вимог охорони праці та техніки безпеки, роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також

забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

При організації робочих місць відповідно до НПАОП 0.00-7.15-18 передбачається наявність як природного, так і штучного освітлення. Природне освітлення зазвичай надходить у приміщення через вікна та світлові прорізи і забезпечує коефіцієнт освітленості не менше 1,5%. Оптимальна орієнтація вікон — на північ або північний схід. Штучне освітлення забезпечується за допомогою відповідних джерел, наприклад, люмінесцентних ламп.

Відповідно до ДСанПІН 3.3.2.007-98, приміщення, де використовується комп'ютерна техніка, не повинні межувати з будівлями, у яких рівень шуму або вібрації перевищує встановлені допустимі норми. Покриття підлоги повинне мати матову поверхню з коефіцієнтом відбиття у межах 0.3-0.5. Для внутрішнього оздоблення приміщень слід використовувати дифузно-відбивні матеріали з коефіцієнтом відбиття: для стелі — 0.7-0.8, для стін — 0.5-0.6.

Роботодавець несе пряму відповідальність за дотримання вимог нормативно-правових актів з охорони праці. При розробці систем оптимального розподілу завдань у комп'ютеризованій системі безпілотної доставки важливо забезпечити відповідність мікроклімату у приміщеннях вимогам ДСН 3.3.6.042-99. Категорія робіт, що відповідає цьому виду завдань, належить до легкої — Іб. Для перевірки відповідності параметрів повітря встановленим нормам необхідно кількісно оцінити кожен із них.

Оптимальні показники мікроклімату для приміщень, де експлуатуються комп'ютери в теплу пору року, повинні становити: температура повітря — +22...+24°С, відносна вологість — 40-60%, швидкість руху повітря — 0,1 м/с.

Окрім, забезпечення оптимальних показників мікроклімату, необхідно передбачити ще й ергономічну організацію робочого місця. Конструкція робочого місця повинна забезпечити підтримання оптимальної робочої пози. У відповідності до НПАОП 0.00-7.15-18, обладнання і організація робочого місця працюючих з ЕОМ мають забезпечувати відповідність конструкції всіх елементів

робочого місця та їх взаємного розташування, ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності.

Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Екран комп’ютера повинен бути розміщений на відстані 600...700 мм від очей користувача. Розташування монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 градусів до нормальної лінії погляду працівника.

Основні вимоги до пожежної безпеки вказані в НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні». Згідно з них на та під приміщеннями, в яких розміщені ЕОМ, а також у суміжних із ними приміщеннях не дозволяється розташування приміщень категорій А та Б за вибухопожежною небезпекою.

Приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (серверні), повинні бути оснащені системою автоматичної пожежної сигналізації. А також повинні бути забезпечені первинними засобами пожежогасіння, а саме вогнегасниками, що використовуються для локалізації і ліквідації пожеж у їх початковій стадії розвитку.

У кваліфікаційній роботі розроблялась система для оптимального розподілу завдань за допомогою програмного забезпечення для ПК. Через це важливо було провести аналіз основних вимог до приміщень та робочих місць з ПК, що дозволило забезпечити комфортні і безпечні умови праці інженерам комп’ютерних систем. Також проаналізовано правила електробезпеки під час роботи з ПК та вимоги до пожежної безпеки в приміщенні.

Дотримання даних вимог є необхідним при створенні методів оптимального розподілу завдань в комп’ютеризовані системі безпілотної доставки.

4.2. Заходи та засоби захисту комп'ютерних систем до впливу надзвичайних ситуацій воєнного характеру

Комп'ютерні системи є важливим елементом сучасної інфраструктури, забезпечуючи функціонування критичних галузей, таких як енергетика, фінанси, транспорт та оборона. У контексті надзвичайних ситуацій воєнного характеру ці системи стають основними цілями для атак або можуть зазнавати впливу через фізичні руйнування інфраструктури. Оскільки такі ситуації мають прямий вплив на стійкість автоматизованих комп’ютерних систем [24, 25], це підвищує актуальність розробки ефективних заходів захисту для забезпечення безперебійної роботи інформаційних систем.

Характер загроз для комп'ютерних систем у надзвичайних ситуаціях воєнного характеру включає:

1) фізичні загрози:

– руйнування дата-центрів та комунікаційних вузлів через обстріли;

– пошкодження інфраструктури електропостачання;

– ускладнення доступу до об’єктів через бойові дії.

2) кіберзагрози:

– цілеспрямовані кібератаки для порушення роботи систем;

– розповсюдження шкідливого програмного забезпечення;

– використання слабких місць у системах для викрадення даних чи саботажу.

3) комбіновані загрози:

– одночасний вплив фізичних та кібернетичних атак, що ускладнює швидке реагування.

Заходи захисту комп'ютерних систем до впливу надзвичайних ситуацій воєнного характеру охоплюють фізичні, організаційні, кібернетичні та інженерні аспекти.

Фізичний захист передбачає розміщення дата-центрів у підземних або захищених приміщеннях, що зменшує ризик їхнього фізичного знищення. Використання дубльованих каналів зв’язку та резервних систем енергозабезпечення дозволяє забезпечити безперебійність роботи критичних систем навіть у разі пошкодження основної інфраструктури. Додатково організовуються віддалені локації для резервного зберігання даних, що гарантує їхню доступність навіть у разі втрати основного сховища.

Організаційні заходи включають розробку планів безперервності роботи бізнесу та аварійного відновлення, які дозволяють оперативно реагувати на кризові ситуації та забезпечувати відновлення функціонування систем. Регулярне навчання персоналу щодо дій у надзвичайних ситуаціях сприяє підвищенню рівня готовності працівників. Для управління інцидентами створюються кризові центри, які координують заходи реагування та зменшують наслідки атак чи аварій [26].

Кіберзахист зосереджений на використанні сучасних засобів шифрування даних для унеможливлення несанкціонованого доступу до інформації. Системи виявлення та запобігання вторгненням (IDS/IPS) дозволяють оперативно виявляти спроби злому чи інші небажані дії. Резервне копіювання із регулярною перевіркою цілісності даних забезпечує можливість швидкого відновлення інформації у разі її пошкодження. Важливим напрямком є впровадження Zero Trust архітектури, яка мінімізує ризики, що виникають у разі компрометації одного з елементів системи [28].

Інженерні рішення включають розробку автономних модулів, які дозволяють системам функціонувати локально у разі втрати зв’язку з основною мережею. Для захисту критичних даних пропонуються захищені канали зв’язку на базі криптографії, яка забезпечує високий рівень безпеки передачі інформації. Системи автоматичного моніторингу стану інфраструктури з використанням AI-аналізу загроз допомагають ідентифікувати потенційні ризики та вживати протидійних заходів [27].

Сучасна практика доводить ефективність цих заходів, зокрема під час збройної агресії проти України у 2022 році, коли компанії масово переходили на використання хмарних технологій для захисту своїх даних. Впровадження системи Starlink забезпечило стабільний зв'язок у зонах бойових дій, а розподілене зберігання даних довело свою ефективність у захисті критичної інформації.

У надзвичайних ситуаціях воєнного характеру ключовими є комплексний підхід до захисту комп'ютерних систем, який включає фізичний, організаційний, кібернетичний захист та використання інноваційних технологій. Впровадження зазначених заходів дозволить зменшити вразливість критичної інфраструктури, забезпечити її безперервну роботу та мінімізувати наслідки атак.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістра було проведено аналіз сучасних комп’ютеризованих систем безпілотної доставки. Розглянуто їхню архітектуру та методи взаємодії з безпілотними пристроями. Проаналізовано характеристики та оцінена спроможність для інтеграції цих пристроїв з іншими комп’ютеризованими системами безпілотної доставки. Описано сучасний стан досліджень у галузі безпілотної доставки та методів розподілу завдань між ними.

Досліджено існуючі методи динамічного розподілу завдань, їхні особливості та недоліки. Проаналізовано методи вимірювання енергоспоживання безпілотних пристроїв з врахуванням впливу додаткової ваги на споживання енергії. Описано переваги застосування генетичного алгоритму для оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки.

Представлено архітектуру комп’ютеризованої системи безпілотної доставки. Описано взаємодію між сервером розподілення задач та центрами керування безпілотних засобів доставки.

Розроблено метод оптимального розподілу завдань шляхом створення універсального підходу, що дозволяє враховувати різні типи і характеристики безпілотних засобів та їхніх робочих зон для найбільш ефективної їх роботи.

Побудовано структуру та реалізовано програмне забезпечення для оптимального розподілу завдань в комп’ютеризованій системі безпілотної доставки на основі генетичного алгоритму. Програма враховує типи безпілотних засобів та їхню робочу зону. Розроблено веб-застосунок для моделювання процесу розподілу завдань та проведення аналізу роботи досліджуваної комп’ютеризованої системи. Представлено результати тестування та роботи алгоритму оптимального розподілу завдань для комп’ютеризованої системи безпілотної доставки при двох сценаріях.

Також у роботі було розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Програма та методичні рекомендації з проходження практики за тематикою кваліфікаційної роботи для студентів спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль: ТНТУ. 2024. 45 с.

2. Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль. 2024. 44 с.

3. Варавін А.В., Лещишин Ю.З., Чайковський А.В. Методичні вказівки до виконання курсового проєкту з дисципліни «Дослідження і проєктування комп'ютерних систем та мереж» для здобувачів другого (магістерського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 32 с.

4. Луцик Н. С., Олійник М. Є., Мудрий І. В. Аналіз комп'ютеризованих систем безпілотної доставки. Актуальні задачі сучасних технологій: Праці XIII наук.-техн. конф. (Тернопіль, 11-12 грудня 2024 р.), Тернопіль, 2024. С. 465.

5. Луцик Н. С., Олійник М. Є., Мудрий І. В. Методи та засоби оптимального розподілу завдань в комп'ютеризованій системі безпілотної доставки. Інформаційні моделі, системи та технології: Праці XII наук.-техн. конф. (Тернопіль, 18-19 грудня 2024 р.), Тернопіль, 2024. С. 141.

6. Cristiano B. A genetic algorithm for a task allocation problem in an urban air mobility scenario. URL: https://webthesis.biblio.polito.it/26956/1/tesi.pdf (дата звернення: 15.11.2024).

7. Zipline Fact Sheet. URL https://www.flyzipline.com/about/zipline-fact-sheet (дата звернення: 20.11.2024).

8. Starship Technologies: Autonomous robot delivery. URL:

https://www.starship.xyz (дата звернення: 20.11.2024).

9. Starship Technologies - Starship Robot. URL: https://www.wevolver.com /specs/starship-technologies-starship-robot (дата звернення: 15.11.2024).

10. Unmanned Delivery Robots Tested in Estonia. URL: https://www.iotworldtoday.com/robotics/unmanned-delivery-robots-tested-in-estonia#close-modal (дата звернення: 21.11.2024).

11. Li X., Tupayachi J., Sharmin A., Martinez Ferguson M. Drone-Aided Delivery Methods, Challenge, and the Future: A Methodological Review. Drones, 2023. 191 c. https://doi.org/10.3390/drones7030191.

12. Ghauri S.A., Sarfraz M., Qamar R.A., Sohail M.F., Khan, S.A. A Review of Multi-UAV Task Allocation Algorithms for a Search and Rescue Scenario. URL: https://doi.org/10.3390/jsan13050047 (дата звернення: 22.11.2024).

13. Peng Q., Wu H., Xue R., Review of Dynamic Task Allocation Methods for UAV Swarms Oriented to Ground Targets. URL: https://doi.org/10.23919/CSMS.2021.0022 (дата звернення: 22.11.2024).

14. Victor R. F., Adriano M. C., Thiago L., Luciano C. A., Gustavo M. Autonomous Navigation System for a Delivery Drone. URL: https://link.springer.com/article/10.1007/s40313-021-00828-4 (дата звернення: 15.11.2024).

15. Giuseppe A. Energy consumption model of aerial urban logistic infrastructures. URL: https://www.researchgate.net/publication/354759249_Energy_Consumption_Model_of_Aerial_Urban_Logistic_Infrastructures. (дата звернення: 15.11.2024).

16. Aboharba S., Mohammad E. Drone Power Consumption Model using in Delivery Task. URL: https://www.stcrs.com.ly/istj/docs/volumes/Drone%20Power%20Consumption.pdf (дата звернення: 22.11.2024).

17. Younas I., Kamrani F., Bashir M., Schubert J. Efficient genetic algorithms for optimal assignment of tasks to teams of agents. URL: https://doi.org/10.1016/j.neucom.2018.07.008 (дата звернення: 15.11.2024).

18. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. URL: https://doi.org/10.1016/j.cja.2017.09.005 (дата звернення: 15.11.2024).

19. Лупенко С. А., Пасічник В. В., Тиш Є. В. Комп'ютерна логіка. Львів: Видавництво «Магнолія - 2006». 2015. 354 с.

20. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.

21. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyy R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU, Ternopil, Ukraine, 2023. Vol. 109, No 1. P. 54–65.

22. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201-207.

23. Kharchenko A., Bodnarchuk I., Yatcysyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.

24. Стручок В.С. Безпека в надзвичайних ситуаціях: Навчальний посібник Тернопіль: Видавництво «ТНТУ», 2016. 37 ст.

25. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

26. ISO/IEC 27001:2022. Information Security Management Systems. URL: https://www.iso.org/standard/82875.html (дата звернення: 14.12.2024)

27. Досвід захисту критичної інфраструктури в умовах війни. URL: https://www.gov.ua (дата звернення: 14.12.2024).

28. Звіти CyberPeace Institute про кібератаки під час воєнних конфліктів. URL: https://cyberpeaceinstitute.org (дата звернення: 14.12.2024).

Додаток А

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

XIII Міжнародної науково-практичної конференції молодих учених та студентів

11-12 грудня 2024 року

УКРАЇНА
ТЕРНОПІЛЬ – 2024

УДК 004.9

М. Є. Олійник; І. В. Мудрий; Н. С. Луцик, Ph.D., доцент
Тернопільський національний технічний університет імені І. Пулюя, Україна

АНАЛІЗ КОП'ЮТЕРИЗОВАНИХ СИСТЕМ БЕЗПІЛОТНОЇ ДОСТАВКИ

M. Ye. Oliinyk; I. V. Mudryi; N. S. Lutsyk, Ph.D., Assoc. Prof.

ANALYSIS OF COMPUTERIZED SYSTEMS FOR UNMANNED DELIVERY

Використання дронів і наземних роботів для доставки відкриває нові можливості в логістиці, зокрема забезпечує доступ до важкодоступних місць, мінімізує людське втручання та скорочує витрати. У сучасному світі швидкість і ефективність логістики мають критичне значення для економічного розвитку. Зокрема, технології безпілотної доставки стають дедалі популярнішими серед наукових кіл та промисловості. Порівняно з традиційними методами доставки, вони забезпечують вищу гнучкість, доступність, швидкість, ефективність, підвищену безпеку, а також мають екологічні переваги [1].

Ринок безпілотних систем доставки демонструє стабільний ріст, обумовлений потребою в автоматизації логістичних процесів. Дані системи вже знаходять застосування в таких сферах, як доставка їжі, медикаментів, комерційних вантажів та гуманітарної допомоги [2]. Різноманітність безпілотних пристроїв дозволяє адаптувати їх до умов конкретних задач, що розширює напрямки їх застосування. Наприклад, дрони з нерухомими крилами є ефективними для довготривалих польотів і охоплення великих територій, тоді як квадрокоптери забезпечують високу маневреність у міських умовах. Наземні роботи, у свою чергу, добре підходять для доставки в обмежених просторах, таких як тротуари чи вузькі дороги.

Серед компаній-лідерів, які активно впроваджують безпілотні системи доставки, варто відзначити Amazon, UPS, Starship Technologies та Zipline [3]. Amazon через свою програму Prime Air тестує дрони для доставки невеликих посилок у віддалені регіони. Starship Technologies уже активно використовує автономних наземних роботів у кампусах і містах для доставки продуктів та інших товарів.

Попри очевидні переваги, існує низка невирішених питань, пов'язаних із взаємодією таких систем. Кожна компанія розробляє власні методи керування своїми пристроями, що створює труднощі у розподілі завдань між різними системами. Особливо це стосується випадків, коли в межах однієї групи використовуються пристрої з різними характеристиками. Відсутність єдиного підходу до взаємодії обмежує ефективність таких систем.

Аналіз методів і алгоритмів розподілу завдань є ключовим кроком до підвищення енергоефективності, продуктивності та доступності безпілотних систем доставки. Врахування характеристик різних типів пристроїв у цьому процесі дозволить зменшити енергоспоживання та підвищити рівень автоматизації.

Література

1. Li X, Tupayachi J, Sharmin A., Martinez Ferguson M. Drone-Aided Delivery Methods, Challenge, and the Future: A Methodological Review. *Drones*. 2023; 7(3):191. https://doi.org/10.3390/drones7030191.

2. How The Robots and Drones Delivery Could Transform Quick Commerce. URL: https://acowebs.com/robots-drone-delivery/.

3. Yevhen L., Milana V., Olena H., Svitlana V., 2024. Research of the route planning algorithms on the example of a drone delivery system software development. https://ceur-ws.org/Vol-3662/paper21.pdf.

28. **В.З. Шеремета,Р.О. Жаровський** **435**
МЕТОДИ І ІНСТРУМЕНТИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ВЕБ–ДОДАТКІВ З ВИКОРИСТАННЯМ SPRING BOOT
29. **В.І. Кравчук; М.С. Хаюк; А.Г.Микитишин** **436**
ДОСЛІДЖЕННЯ ВПЛИВУ ЗАВАД НА ПЕРЕДАВАННЯ ФАЗОМАНІПУЛЬОВАНИХ СИГНАЛІВ
30. **В.О. Бонар** **438**
ВИЯВЛЕННЯ АНОМАЛІЙ OBD2 ДАНИХ ТРАНСПОРТНИХ ЗАСОБІВ МЕТОДОМ НАЙБЛИЖЧИХ СУСІДІВ
31. **В.Р. Кошулинський** **440**
ІНТЕЛЕКТУАЛЬНІ СЕНСОРНІ СИСТЕМИ ДЛЯ ТЕЛЕКОМУНІКАЦІЙ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ (IoT)
32. **Галина Шевчук, Ростислав Трембач** **442**
ОРГАНІЗАЦІЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»
33. **Гарасівка А. В., Лупенко А. М** **444**
ВИКОРИСТАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ РЕЗЕРВНОГО КОПІЮВАННЯ ДАНИХ
34. **Д. Клочко, Ю. Лещишин, Р.Жаровський** **446**
МЕТОДИ І ЗАСОБИ МОНІТОРИНГУ СЕЙСМІЧНОЇ АКТИВНОСТІ ЗЕМНОЇ КОРИ
35. **Д.С. Матюк, М.В. Деркач** **448**
КЛАСИФІКАЦІЯ ЕМГ-СИГНАЛІВ НА ОСНОВІ МЕТОДУ ADAM
36. **Д.О. Бугай, С.Є. Тужанський** **450**
ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНА СИСТЕМА АВТОМАТИЗОВАНОГО КЛІМАТ-КОНТРОЛЮ
37. **Жаровський Р.О, Цірка І.П.** **453**
АРХІТЕКТУРА СИСТЕМИ ЗБОРУ, ПЕРЕДАЧІ ТА ЗБЕРІГАННЯ ДАНИХ ВОДОСПОЖИВАННЯ У БАГАТОКВАРТИРНИХ БУДИНКАХ
38. **Зеньо Д** **454**
МАБУТНЄ КВАНТОВИХ КОМП'ЮТЕРІВ: ПЕРЕВАГИ ТА ПЕРСПЕКТИВИ
39. **І.І. Бородій, К.І. Поліщук** **456**
ВИБІР ДЖЕРЕЛ ДАНИХ ТА ТЕХНІЧНИХ ЗАСОБІВ ДЛЯ ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ СТАНУ ЯКОСТІ АТМОСФЕРНОГО ПОВІТРЯ
40. **І.М. Климко** **458**
ОПТИМІЗАЦІЯ НАВЧАННЯ LSTM-МОДЕЛІ ДЛЯ ГЕНЕРАЦІЇ ТЕКСТУ НА РІВНІ СИМВОЛІВ
41. **І.О Мартинюк** **460**
Електронні системи на основі штучного інтелекту
42. **І.Р. Ралік** **462**
ЗАДАЧІ РОЗРОБКИ CRM-СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ТОРГІВЛІ ПІДПРИЄМСТВА
43. **Л.П. Дмитроца, О.Т. Старицький** **463**
ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ТА SEO ПРИ МАСШТАБУВАННІ ПРОЄКТІВ НА ОСНОВІ REACT І NEXT.JS
44. **М. Є. Олійник; І. В. Мудрий; Н. С. Луцик, Ph.D** **465**
АНАЛІЗ КОП'ЮТЕРИЗОВАНИХ СИСТЕМ БЕЗПІЛОТНОЇ ДОСТАВКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

М А Т Е Р І А Л И

XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

18-19 грудня 2024 року

ТЕРНОПІЛЬ
2024

УДК 004.9
М. Є. Олійник; І. В. Мудрий; Н. С. Луцик, докт. філософ. наук, доцент
Тернопільський національний технічний університет імені І. Пулюя, Україна

МЕТОДИ ТА ЗАСОБИ ОПТИМАЛЬНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНІЙ СИСТЕМІ БЕЗПІЛОТНОЇ ДОСТАВКИ

M. Ye. Oliinyk; I. V. Mudryi; N. S. Lutsyk, Ph.D., Assoc. Prof.

METHODS AND TOOLS FOR OPTIMAL TASK ALLOCATION IN A COMPUTERIZED UNMANNED DELIVERY SYSTEM

Забезпечення оптимального розподілу завдань є одним з основних етапів в роботі комп'ютеризованої системи безпілотної доставки. Особливо це актуально в умовах обмежених ресурсів, де необхідно зважати на заряд батареї пристрою та його фізичні характеристики. В комп'ютеризованій системі з різними типами пристроїв така необхідність тільки зростає. Одним із перспективних підходів для вирішення таких задач є використання генетичних алгоритмів (GA).

Генетичні алгоритми є ефективним підходом для розв'язання NP-складних задач завдяки їхній здатності досліджувати великі й складні простори рішень. NP-складні задачі характеризуються неможливістю їх розв'язання за поліноміальний час, і традиційні методи, такі як повний перебір або градієнтний підхід, стають непридатними через високі обчислювальні витрати при збільшенні розміру задачі [1].

Пропонований метод передбачає алгоритм на основі протилежних хромосом з подвійним кодування та множинних операторів мутації (OGA-DEMMO). В даному алгоритмі одна частина хромосоми описує розподіл завдань між безпілотниками, а інша – оптимальні маршрути виконання цих завдань.

Такий підхід дозволяє інтегрувати два рівні оптимізації в єдиний алгоритм, що підвищує точність та ефективність рішень [2]. Застосування операцій кросоверу, мутації та відбору сприяє дослідженню великих просторів рішень і запобігає передчасному застряганню на локальних оптимумах [3].

Таким чином, OGA-DEMMO алгоритм є універсальним і надійним інструментом оптимізації, який ефективно працює в задачах високої складності. Застосування генетичного алгоритму для динамічного розподілу завдань показує хороший результат в контексті енергетичних витрат, але він не враховує різні типи пристроїв та зони їхньої дії. Це обмежує сферу застосування, може призвести до некоректних результатів або збоїв у роботі даного алгоритму. Для ефективного розподілу завдань в комп'ютеризованій системі безпілотної доставки, зменшення витрат енергії та підвищення ефективності запропоновано враховувати тип і характеристики пристроїв та їхню робочу зону у цьому процесі.

Література

1. Bicchieri, C. 2023. A genetic algorithm for a task allocation problem in an urban air mobility. https://webthesis.biblio.polito.it/26956/1/tesi.pdf

2. Zhu Wang, Li Liu, Teng Long, Yonglu Wen. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. https://doi.org/10.1016/j.cja.2017.09.005.

3. Younas, I., Kamrani, F., Bashir, M., Schubert J., 2018. Efficient genetic algorithms for optimal assignment of tasks to teams of agents. https://doi.org/10.1016/j.neucom.2018.07.008.

І. О. Кухар, Г. М. Осухівська
ІНТЕГРАЦІЯ ВІДНОВЛЮВАЛЬНИХ ДЖЕРЕЛ ЕНЕРГІЇ В РОЗУМНІ ЕЛЕКТРИЧНІ МЕРЕЖІ З ВИКОРИСТАННЯМ ПЕРЕДОВИХ ТЕХНОЛОГІЙ
I. O. Kukhar, H. Osukhivska
INTEGRATION OF RENEWABLE ENERGY SOURCES INTO SMART GRIDS USING ADVANCED TECHNOLOGIES 135
Ю. Лещишин, А. Герасименко, О. Герасименко
КОМП'ЮТЕРНА СИСТЕМА МОНІТОРИНГУ РАДІОЗВ'ЯЗКУ ВІД БПЛА
Yu. Leshchyshyn, A. Herasymenko, O. Herasymenko
COMPUTER SYSTEM FOR MONITORING RADIO COMMUNICATIONS FROM UAVS 136
А. Луцків, А. Люлька
ЗАСТОСУВАННЯ АЛГОРИТМІВ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ПРОЦЕСАХ СУЧАСНИХ ОПЕРАЦІЙНИХ СИСТЕМ
A. Lutskiv, A. Liulka
APPLICATION OF LOAD BALANCING ALGORITHMS IN THE PROCESSES OF MODERN OPERATING SYSTEMS 137
Я. О. Мишаківський
МЕТОДИ РОЗПІЗНАВАННЯ СТАТІ ТА ВІКУ ЗА ЗОБРАЖЕННЯМ ОСОБИ
Ia. O. Myshakivskyi
METHODS OF GENDER AND AGE RECOGNITION FROM THE IMAGE OF A PERSON 138
Я. О. Мишаківський
МЕТОДИ ДЛЯ РЕАЛІЗАЦІЇ АЛГОРИТМУ РОЗПІЗНАВАННЯ СТАТІ ТА ВІКУ ЛЮДИНИ У ВІДЕОПОТОЦІ ДАНИХ
Ia. O. Myshakivskyi
METHODS FOR IMPLEMENTING AN ALGORITHM FOR RECOGNITION OF A PERSON'S GENDER AND AGE IN A VIDEO DATA STREAM 140
М. Є. Олійник, І. В. Мудрий, Н. С. Луцик
МЕТОДИ ТА ЗАСОБИ ОПТИМАЛЬНОГО РОЗПОДІЛУ ЗАВДАНЬ В КОМП'ЮТЕРИЗОВАНІЙ СИСТЕМІ БЕЗПІЛОТНОЇ ДОСТАВКИ
M. Y. Oliinyk, I. V. Mudryi, N. S. Lutsyk,
METHODS AND TOOLS FOR OPTIMAL TASK ALLOCATION IN A COMPUTERIZED UNMANNED DELIVERY SYSTEM 141
Н. Сороківська, В. Яцишин
ВИКОРИСТАННЯ АДАПТОВАНИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ІДЕНТИФІКАЦІЇ ПТАХІВ У ПРИРОДНИХ УМОВАХ
N. Sorokivska, V. Yatsyshyn
APPLICATION OF ADAPTED ARTIFICIAL INTELLIGENCE MODELS FOR BIRD IDENTIFICATION IN NATURAL ENVIRONMENTS 142
А. А. Станько, А. В. Гончаренко, І. В. Журик
РОЗУМНІ МІСТА: ІНТЕГРАЦІЯ ТЕХНОЛОГІЙ, ДАНИХ І СТРАТЕГІЙ СТАЛОГО РОЗВИТКУ МІСЬКОЇ ЕКОСИСТЕМИ
A. A. Stanko, A. V. Honcharenko, I. V. Zhuryk
SMART CITIES: INTEGRATING TECHNOLOGIES, DATA AND STRATEGIES FOR SUSTAINABLE URBAN ECOSYSTEM DEVELOPMENT 143
А. А. Станько, С. З. Кульчицький, Ю. В. Срібний
ФОРМУВАННЯ КЕРОВАНИХ ОЗЕР ДАНИХ: МЕТОДОЛОГІЇ, ІНСТРУМЕНТИ ТА ПРАКТИЧНІ АСПЕКТИ
A. A. Stanko, S. Z. Kulchytskyi, Y. V. Sribnyi
FORMATION OF MANAGED DATA LAKES: METHODOLOGIES, TOOLS AND PRACTICAL ASPECTS 144
А. Чуба, В. Тимощук, А. Микитишин, В. Шиманська
СУЧАСНІ ПІДХОДИ ДО МОНІТОРИНГУ ТА УПРАВЛІННЯ В DOCSIS-МЕРЕЖАХ
A. Chuba, V. Tymoshchuk, A. Mykytyshyn, V. Shymanska
MODERN APPROACHES TO MONITORING AND MANAGEMENT IN DOCSIS NETWORKS 145

Додаток Б

Лістинг коду генетичного алгоритму

```
export function geneticAlgorithm(
  tasksMap: Map<number, Task>,
  runnersMap: Map<number, Runner>,
  tasks: number[],
  runners: number[],
  populationSize: number,
  maxGenerations: number,
  maxIndividualsPerGeneration: number,
  solutionDifference: number,
  timeDistributionCoef: number,
  runnerDistributionCoef: number
): TEnergyIndividual | undefined {
  // Step 1: Initialize population

  let numberOfIteration = 0;
  let saveEnergyBest = [];
  let saveTimeBest = [];

  if (tasks.length === 0 || runners.length === 0)
    throw new Error("Input data cannot be empty");

  let population: Population = initializePopulation(
    tasks,
    runners,
    populationSize
  );

  let bestSolution: TEnergyIndividual = {
    chromosomeI: [],
    chromosomeII: [],
    energyConsuming: [],
    totalEnergyConsuming: 0,
  };

  for (let generation = 0; generation < maxGenerations;
generation++) {
    numberOfIteration++;
    // Step 2: Opposite population P_OPP creation from P
    let oppositePopulation = generateOppositePopulation(population);
    //Step 2: Crossover
    const crossoveredPopulation = partialMappedCrossover(
      population,
      CROSSOVER_RATE
    );
    const crossoveredOppositePopulation = partialMappedCrossover(
      oppositePopulation,
      CROSSOVER_RATE
    );
```

```
    // Step 3: Mutation
    crossoveredPopulation.forEach((individual, index) => {
      if (Math.random() < MUTATION_RATE) {
        crossoveredPopulation[index] = mutate(individual);
      }
    });
    crossoveredOppositePopulation.forEach((individual, index) => {
      if (Math.random() < MUTATION_RATE) {
        crossoveredOppositePopulation[index] = mutate(individual);
      }
    });
    //Step 4-5 Check stricted parametrs
    const populationRespectedLoadCapasity =
crossoveredPopulation.filter(
      (individual) =>
        isRespectedLoadCapacity(
          individual,
          runners,
          tasks,
          tasksMap,
          runnersMap
        ) &&
        isRespectWorkingZone(individual, runners, tasks, tasksMap,
runnersMap)
    );
    const crossoveredRespectedLoadCapasity =
      crossoveredOppositePopulation.filter(
        (individual) =>
          isRespectedLoadCapacity(
            individual,
            runners,
            tasks,
            tasksMap,
            runnersMap
          ) &&
          isRespectWorkingZone(individual, runners, tasks, tasksMap,
runnersMap)
      );
    //Step 6: Calaculate energy consuming
    const energyCalculatedPopulation =
populationRespectedLoadCapasity.map(
      (individual) => {
        const energyCalculatedIndividuel =
addEnergyConsumingByIndividual(
          individual,
          runners,
          tasks,
          tasksMap,
          runnersMap
        );
        const timeCalculatedIndividuel =
addTimeConsumingByIndividual(
          individual,
          runners,
```

```
          tasks,
          tasksMap,
          runnersMap
        );

        const penaltyForBadTimeDistribution =
          (timeCalculatedIndividuel.maxTimeConsumingPerRunner -
            timeCalculatedIndividuel.minTimeConsumingPerRunner) /
          timeCalculatedIndividuel.maxTimeConsumingPerRunner;

        const penaltyForMinRunnersUsed =
          energyCalculatedIndividuel.energyConsuming.length /
          energyCalculatedIndividuel.energyConsuming.reduce((total,
curr) => {
            return curr.length !== 0 ? total + 1 : total;
          }, 0);
        energyCalculatedIndividuel.totalEnergyConsuming =
          energyCalculatedIndividuel.totalEnergyConsuming +
          energyCalculatedIndividuel.totalEnergyConsuming *
            timeDistributionCoef *
            penaltyForBadTimeDistribution +
          energyCalculatedIndividuel.totalEnergyConsuming *
            runnerDistributionCoef *
            penaltyForMinRunnersUsed;
        return energyCalculatedIndividuel;
      }
    );
    // console.log(
    //   "energyCalculatedPopulation: ",
    //   JSON.stringify(energyCalculatedPopulation)
    // );
    // console.log(
    //   "populationRespectedLoadCapasity: ",
    //   JSON.stringify(populationRespectedLoadCapasity)
    // );
    const energyCalculatedOppositePopulation =
      crossoveredRespectedLoadCapasity.map((individual) => {
        const energyCalculatedIndividuel =
addEnergyConsumingByIndividual(
          individual,
          runners,
          tasks,
          tasksMap,
          runnersMap
        );
        const timeCalculatedIndividuel =
addTimeConsumingByIndividual(
          individual,
          runners,
          tasks,
          tasksMap,
          runnersMap
        );
        const penaltyForBadTimeDistribution =
```

```
          (timeCalculatedIndividuel.maxTimeConsumingPerRunner -
            timeCalculatedIndividuel.minTimeConsumingPerRunner) /
          timeCalculatedIndividuel.maxTimeConsumingPerRunner;

        const penaltyForMinRunnersUsed =
          energyCalculatedIndividuel.energyConsuming.length /
          energyCalculatedIndividuel.energyConsuming.reduce((total,
curr) => {
            return curr.length !== 0 ? total + 1 : total;
          }, 0);
        energyCalculatedIndividuel.totalEnergyConsuming =
          energyCalculatedIndividuel.totalEnergyConsuming +
          energyCalculatedIndividuel.totalEnergyConsuming *
            timeDistributionCoef *
            penaltyForBadTimeDistribution +
          energyCalculatedIndividuel.totalEnergyConsuming *
            runnerDistributionCoef *
            penaltyForMinRunnersUsed;
        return energyCalculatedIndividuel;
      });
    // Step 3: Selection
    const bestHalfFromPopulation = getHalfBestSelection(
      energyCalculatedPopulation,
      maxIndividualsPerGeneration
    );
    const bestHalfFromOpposite = getHalfBestSelection(
      energyCalculatedOppositePopulation,
      maxIndividualsPerGeneration
    );
    console.log(
      "Selection: Half Pop: ",
      bestHalfFromPopulation.length,
      " Half Opp:",
      bestHalfFromOpposite.length
    );
    // Step 6: Add charging tasks and check constraints
    // Step 7: Offspring Population (Create new population)
    let offspring = offspringPopulation(
      energyCalculatedPopulation,
      energyCalculatedOppositePopulation,
      maxIndividualsPerGeneration
    );
    const nexPopulation = [
      ...bestHalfFromPopulation,
      ...bestHalfFromOpposite,
      ...offspring,
    ];
    let bestPerEvolution = {
      ...nexPopulation.reduce((best, ind) =>
        best.totalEnergyConsuming > ind.totalEnergyConsuming ? ind :
best
      ),
    };
```

```
    saveEnergyBest.push({ ...bestPerEvolution });
    const timeCalculatedIndividuel = addTimeConsumingByIndividual(
      prepareForNexGeneration([bestPerEvolution])[0],
      runners,
      tasks,
      tasksMap,
      runnersMap
    );
    saveTimeBest.push({ ...timeCalculatedIndividuel });
    if (
      Math.abs(
        bestSolution.totalEnergyConsuming -
          bestPerEvolution.totalEnergyConsuming
      ) <= solutionDifference
    ) {
      break;
    }
    if (bestSolution.totalEnergyConsuming !== 0) {
      if (
        bestPerEvolution.totalEnergyConsuming <
        bestSolution.totalEnergyConsuming
      ) {
        bestSolution = { ...bestPerEvolution };
      }
      if (
        bestPerEvolution.totalEnergyConsuming ==
        bestSolution.totalEnergyConsuming
      ) {
        // console.log(JSON.stringify(nexPopulation));
      }
    } else {
      bestSolution = { ...bestPerEvolution };
    }
    population = [...prepareForNexGeneration(nexPopulation)];
  }
  return bestSolution;
}
```