

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «АЛГОРИТМ ОПТИМІЗАЦІЇ ОБЧИСЛЮВАЛЬНИХ ТА
КОМУНІКАЦІЙНИХ РЕСУРСІВ МЕС-МЕРЕЖ З
ВИКОРИСТАННЯМ БПЛА»

на здобуття освітнього ступеня магістра
зі спеціальності 126 – Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми: Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис)

Марина ПАВЛЕНКО
Ім'я, ПРІЗВИЩЕ здобувача

Виконала: здобувачка вищої освіти гр. ІСДМ-61

Марина ПАВЛЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник:
науковий ступінь,
вчене звання

Оксана ТКАЛЕНКО
Ім'я, ПРІЗВИЩЕ
к.т.н., доцент

Рецензент:
науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИЙ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій**

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти – Магістр

Спеціальність підготовки 126 – Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних систем та технологій

_____ Каміла СТОРЧАК

« ____ » _____ 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Павленко Марині Юріївні

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Алгоритм оптимізації обчислювальних та комунікаційних ресурсів МЕС-мереж з використанням БПЛА»

керівник роботи Ткаленко Оксана Миколаївна, кандидат технічних наук, доцент кафедри ІСТ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 року № 320.

2. Строк подання кваліфікаційної роботи: «27» грудня 2024 року.

3. Вихідні дані до кваліфікаційної роботи:

Genetic Algorithm, Particle Swarm Optimization.

Технології мобільного зв'язку: МЕС, 5G NR, FDD/TDD моделі.

Інструменти моделювання: MATLAB Simulink, Python (NumPy, SciPy, NetworkX).

Науково-технічна література з питань, пов'язаних з застосуванням технологій.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Концептуальні основи взаємодії БПЛА та систем МЕС.

2. Розвантаження мобільних периферійних обчислень.

3. Мобільні периферійні обчислення з підтримкою кількох безпілотників із балансуванням навантаження для мереж IoT.

4. Ефективне використання ресурсів у МЕС із підтримкою БПЛА.

5. Перелік ілюстративного матеріалу: *презентація*

1. Актуальність роботи.

2. Огляд проблематики.

3. Результати проведеної роботи.

4. Варіант схеми доступу для систем U-МЕС.

5. Результати моделювання.

6. Аналіз ефективності алгоритмів та параметрів системи.

7. Практична значущість.

8. Висновки, апробації.

6. Дата видачі завдання «15» жовтня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	15.10 – 22.10.24	Виконано
2	Вивчення основ взаємодії БПЛА і МЕС	23.10 – 02.11.24	Виконано
3	Розвантаження мобільних обчислень	03.11 – 17.11.24	Виконано
4	Дослідження особливостей балансування навантаження у мережах	18.11 – 28.11.24	Виконано
5	Вступ, висновки, реферат	01.12 – 04.12.24	Виконано
6	Розробка обов'язкових демонстраційних креслень	15.10 – 22.10.24	Виконано
7	Попередній захист роботи	25.12.2024	Виконано

Здобувачка вищої освіти

_____ (підпис)

Марина ПАВЛЕНКО

Ім'я, ПРИЗВИЩЕ

Керівник

кваліфікаційної роботи

_____ (підпис)

Оксана ТКАЛЕНКО

Ім'я, ПРИЗВИЩЕ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 73 стор., 22 рис., 2 табл., 30 джерел.

Мета роботи – розробка та оптимізація архітектури мобільних периферійних обчислень (МЕС) із підтримкою безпілотних літальних апаратів (БПЛА) для забезпечення ефективного управління ресурсами, зниження затримки та енерговитрат у мережах IoT.

Об'єкт дослідження – система мобільних периферійних обчислень із підтримкою БПЛА.

Предмет дослідження – методи та алгоритми оптимізації управління ресурсами і мінімізації затримки в МЕС-системах із підтримкою БПЛА.

Короткий зміст роботи:

Робота присвячена розробці та аналізу ефективних методів інтеграції БПЛА у системи МЕС. Дослідження включає вивчення архітектур систем, комунікаційних і обчислювальних моделей, а також завдань, пов'язаних із мінімізацією затримки та енергоспоживання.

У роботі розглянуто підходи до управління ресурсами, оптимізації розміщення завдань і вибору мережі, а також балансування навантаження. Запропоновано інноваційні алгоритми планування завдань із використанням навчання з підкріпленням і методів диференціальної еволюції. Проведене моделювання демонструє підвищення ефективності роботи системи.

Отримані результати можуть бути застосовані для вдосконалення МЕС-систем, розширення їх функціональності в умовах використання IoT-пристроїв та забезпечення їх стабільної роботи у сучасних бездротових мережах.

КЛЮЧОВІ СЛОВА: МОБІЛЬНІ ПЕРИФЕРІЙНІ ОБЧИСЛЕННЯ, БЕЗПІЛОТНІ ЛІТАЛЬНІ АПАРАТИ, ОПТИМІЗАЦІЯ РОБОТИ, IoT, ЕНЕРГОЕФЕКТИВНІСТЬ, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, U-МЕС.

ABSTRACT

Text part the master`s qualification work:

73 pages, 22 figures, 2 tables, 30 sources.

The purpose of the work is to develop and optimize the architecture of mobile peripheral computing (MEC) with the support of unmanned aerial vehicles (UAVs) to ensure effective resource management, reduce latency and energy consumption in IoT networks.

Object of research – is a mobile peripheral computing system with the support of UAVs.

Subject of research – is methods and algorithms for optimizing resource management and minimizing latency in MEC systems with the support of UAVs.

Summary of the work:

The work is devoted to the development and analysis of effective methods for integrating UAVs into MES systems. The study includes the study of system architectures, communication and computational models, as well as tasks related to minimizing latency and energy consumption.

The paper considers approaches to resource management, optimization of task placement and network selection, as well as load balancing. Innovative task scheduling algorithms using reinforcement learning and differential evolution methods are proposed. The simulation demonstrates an increase in the efficiency of the system.

The results obtained can be used to improve MEC systems, expand their functionality in the conditions of using IoT devices and ensure their stable operation in modern wireless networks.

KEYWORDS: MOBILE EDGE COMPUTING, UNMANNED AIRCRAFT, OPTIMIZATION OF WORK, IoT, ENERGY EFFICIENCY, REINFORCEMENT LEARNING, U-MEC.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	10
ВСТУП.....	11
РОЗДІЛ 1 КОНЦЕПТУАЛЬНІ ОСНОВИ ВЗАЄМОДІЇ БПЛА ТА СИСТЕМ	
МЕС	14
1.1 Безпілотні літальні апарати	14
1.2 Мобільні периферійні обчислення	16
1.3 Співпраця БПЛА та МЕС	17
1.4 Система МЕС з підтримкою БПЛА.....	18
1.5 Можливі архітектури в системі U-МЕС	20
1.6 МЕС: завдання, програми і майбутні напрямки	22
1.6.1 Пов'язані концепції та технології.....	22
1.7 Структура та архітектура МЕС.....	27
РОЗДІЛ 2 РОЗВАНТАЖЕННЯ МОБІЛЬНИХ ПЕРИФЕРІЙНИХ ОБЧИСЛЕНЬ	32
2.1 Процес розвантаження обчислень.....	32
2.2 Режими роботи для розвантаження обчислень	34
2.3 Схеми доступу до розвантаження обчислень.....	36
2.4 Підходи до оптимізації	38
2.4.1 Оптимізація розміщення послуг	38
2.4.2 Оптимізація для вибору мережі.....	40
2.4.3 Оптимізації затримки в МЕС-системах	42
РОЗДІЛ 3 МОБІЛЬНІ ПЕРИФЕРІЙНІ ОБЧИСЛЕННЯ З ПІДТРИМКОЮ	
КІЛЬКОХ БЕЗПІЛОТНИКІВ ІЗ БАЛАНСУВАННЯМ НАВАНТАЖЕННЯ ДЛЯ	
МЕРЕЖ ІоТ	44
3.1 Модель мережі.....	44
3.2. Комунікаційна модель	46
3.3 Обчислювальна модель	48
3.4 Формулювання проблеми.....	48
3.5 Спільне розгортання БПЛА та оптимізація планування завдань.....	51
3.6 Баланс навантаження для БПЛА	52
3.7 Призначення вузла на основі GAP	54
3.8 Планування завдань із підкріпленням за допомогою навчання.....	58

3.9 Розгортання кількох БПЛА на основі диференціальної еволюції	61
3.10 Результати моделювання	63
РОЗДІЛ 4 ЕФЕКТИВНЕ ВИКОРИСТАННЯ РЕСУРСІВ У МЕС З ПІДТРИМКОЮ БПЛА	70
4.1 Управління ресурсами	70
4.2 Енергоефективність	72
4.3 Обчислювальна ефективність	79
4.4 Мінімізація затримки	80
4.5 Балансування навантаження та забезпечення секретності	82
4.6 Майбутні напрямки досліджень	85
ВИСНОВКИ.....	87
ПЕРЕЛІК ПОСИЛАНЬ	88
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	92

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

BS	<i>Base Station</i> — базова станція
CC	<i>Cloud Computing</i> — хмарні обчислення
FC	<i>Fog Computing</i> — туманні обчислення
FW-UAVs	<i>Fixed-Wing Unmanned Aerial Vehicles</i>
GBS	<i>Ground Base Station</i> — наземна базова станція
HAP	<i>High-Altitude Platform</i> — платформа високої висоти
IoT	<i>Internet of Things</i> — Інтернет речей
LAN	<i>Local Area Network</i> — локальна мережа
LAP	<i>Low-Altitude Platform</i> — платформа низької висоти
LoS	<i>Line of Sight</i> - "пряма видимість"
MAEC	<i>Mobile and Edge Computing</i> — мобільні й крайові обчислення
MCC	<i>Mobile Cloud Computing</i> — мобільні хмарні обчислення
MD	<i>Mobile Device</i> — мобільний пристрій
MEH	<i>Mobile Edge Hosts</i> — хости мобільного краю
MEC	<i>Mobile Edge Computing / Multi-access Edge Computing</i>
MU	<i>Mobile User</i> — мобільний користувач
NFV	<i>Network Functions Virtualization</i> — віртуалізація мережевих функцій
QoE	<i>Quality of Experience</i> — якість досвіду
QoS	<i>Quality of Service</i> — якість обслуговування
RW-UAVs	<i>Rotary-Wing Unmanned Aerial Vehicles</i>
SSL	<i>Secure Sockets Layer</i> — протокол захищеного з'єднання
SWaP	<i>Size, Weight, and Power</i> — розмір, вага та енергоспоживання
uRLLC	<i>Ultra-Reliable Low Latency Communication</i>
VIM	<i>Virtualized Infrastructure Manager</i>
VNF	<i>Virtual Network Function</i> — віртуальна мережна функція
VTOL	<i>Vertical Take-Off and Landing</i> — вертикальний зліт і посадка

ВСТУП

Сучасний розвиток інформаційно-комунікаційних технологій обумовлює зростаючу потребу в забезпеченні ефективних обчислювальних ресурсів і високої продуктивності мереж. Впровадження мобільних периферійних обчислень (*Mobile Edge Computing, MEC*) у поєднанні з використанням безпілотних літальних апаратів (*БПЛА*) відкриває нові перспективи для розв'язання завдань у сферах IoT, розумних міст, логістики та надзвичайних ситуацій.

Актуальність теми. Можливість інтеграції БПЛА в MEC-системи забезпечує розширення зон покриття, підвищення адаптивності та зниження затримки в обробці даних. Разом із тим, постають нові виклики, зокрема оптимізація розміщення ресурсів, управління затримками, балансування навантаження і забезпечення енергоефективності. У контексті розвитку мереж 5G і перспективних технологій 6G, дослідження цих питань набувають особливої актуальності.

Робота спрямована на дослідження концептуальних основ інтеграції БПЛА та MEC, а також розробку підходів до підвищення ефективності таких систем.

Аналіз останніх досліджень і публікацій. Проблематику оптимізації MEC-систем і впровадження БПЛА активно досліджують у наукових роботах видатних авторів, які запропонували інноваційні підходи до зменшення затримки та енерговитрат у MEC. Значний внесок зроблено у сфері планування завдань і розподілу ресурсів, де використано алгоритми машинного навчання. Однак, незважаючи на досягнення, залишаються невирішеними питання ефективного розвантаження обчислень, динамічної оптимізації затримки і адаптації до умов реального часу в мережах MEC з підтримкою БПЛА.

Мета дослідження — розробка та оптимізація архітектури мобільних периферійних обчислень із підтримкою БПЛА для забезпечення ефективного управління ресурсами, зниження затримки та енерговитрат у мережах IoT.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Дослідити особливості архітектури MEC та інтеграції БПЛА.

2. Розробити підходи до оптимізації розміщення ресурсів і балансування навантаження.
3. Вивчити можливі схеми доступу для систем U-MEC.
4. Провести моделювання та аналіз запропонованих методів.

Об'єктом дослідження є система мобільних периферійних обчислень із підтримкою БПЛА.

Предметом дослідження є методи та алгоритми оптимізації управління ресурсами і мінімізації затримки в таких системах.

Методи дослідження. У роботі використано методи математичного моделювання, оптимізації, алгоритми машинного навчання, а також методи імітаційного моделювання для оцінки ефективності розроблених підходів.

Наукова новизна. У роботі вперше запропоновано комплексну архітектуру MEC із підтримкою БПЛА, що забезпечує підвищення енергоефективності та зниження затримки в IoT-мережах. Розроблено нові методи спільного планування завдань і розгортання БПЛА, які базуються на алгоритмах навчання з підкріпленням та диференціальної еволюції. Також удосконалено підходи до балансування навантаження в системах MEC.

Практична значущість. Запропоновані методи та підходи можуть бути використані для оптимізації роботи реальних IoT-мереж, забезпечуючи покращення якості обслуговування (QoS) і користувацького досвіду (QoE). Результати дослідження можуть бути використані для вдосконалення сучасних мереж зв'язку, а також у сфері автоматизації логістики, моніторингу та управління надзвичайними ситуаціями.

Апробація результатів та публікації:

Павленко М. Ю. «Інтеграція Mobile edge computing та IoT для покращення функціонування “розумних будинків” і “розумних міст”». Тези доповіді на II Всеукраїнську науково-технічну конференцію «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ, 18 листопада 2024 р.

Павленко М. Ю. «Вплив мобільних периферійних обчислень на еволюцію інформаційних систем у 5G-мережах». Тези доповіді на VII Всеукраїнську науково-технічну конференцію «Комп'ютерні технології: інновації, проблеми, рішення». – Житомир, 2 грудня 2024 р.

Теоретична значущість роботи полягає у розробці нових методів оптимізації ресурсів і затримок у МЕС-системах із підтримкою БПЛА. Запропоновані моделі можуть бути використані для подальших досліджень у галузі мереж IoT і технологій 6G.

Методична значущість полягає у формуванні чітких алгоритмів та підходів, які можна адаптувати для інших типів обчислювальних систем із мобільними вузлами.

Практична значущість забезпечується можливістю безпосереднього використання розроблених підходів для вдосконалення існуючих мережевих систем.

1 КОНЦЕПТУАЛЬНІ ОСНОВИ ВЗАЄМОДІЇ БПЛА ТА СИСТЕМ МЕС

У сучасному світі стрімкий розвиток безпілотних літальних апаратів (БПЛА) і мобільних периферійних обчислень (МЕС) сприяє створенню нових підходів до розподілу обчислювальних і комунікаційних ресурсів. Взаємодія цих технологій відкриває перспективи підвищення продуктивності системи, скорочення затримок і оптимізації використання енергоресурсів у різних сценаріях.

Цей розділ присвячений аналізу ключових компонентів, які лежать в основі інтеграції БПЛА в системи МЕС. Зокрема, розглядаються характеристики безпілотних літальних апаратів, особливості мобільних периферійних обчислень, а також можливі архітектури, що забезпечують їх ефективну взаємодію. Особливу увагу приділено системам U-МЕС, які поєднують можливості БПЛА для підтримки функціонування периферійних мереж у динамічному середовищі.

Розуміння концептуальних засад такої взаємодії необхідне для створення оптимізованих рішень, що відповідають сучасним вимогам до надійності, енергоспоживання та масштабованості обчислювальних систем.

1.1 Безпілотні літальні апарати

БПЛА, широко відомі як дрони/дистанційно керовані літальні апарати, використовуються у багатьох військових і цивільних цілях, наприклад, для доставки контенту, інтелектуального спостереження, моніторингу руху, телекомунікації, відстеження, дистанційного зондування тощо [1].

БПЛА використовуються в бездротовому зв'язку завдяки лінії зв'язку LoS (*Line of Sight*) із наземними користувачами, що зменшує ефект затінення і блокування та уникає перешкод, завдяки адаптивній висоті, високій мобільності та гнучкості. Невеликі безпілотники встановлюють надійне з'єднання з низькою потужністю передачі, забезпечуючи рентабельні та енергоефективні рішення для мобільних наземних користувачів, розташованих на великій географічній території.

Бездротовий зв'язок на основі дронів зазвичай розгортається в надзвичайних ситуаціях, у важкодоступних місцях, де неможливо розгорнути дротову інфраструктуру.

Короткий огляд функцій, що можуть використовуватись:

- Повітряні базові станції на базі дронів, також звані бездротовими системами з підтримкою об'єктів, в яких БПЛА служить літаючою базовою станцією, забезпечуючи надійну пропускну здатність мережі, зв'язку по висхідній та низхідній лініях з наземними користувачами, наприклад, у надзвичайних ситуаціях [2].

- Повітряні користувачі на базі БПЛА (які також називаються БПЛА з підключенням до стільникового зв'язку) у яких дрон виконує роль літаючого мобільного користувача в стільниковій мережі, забезпечуючи надійний зв'язок із малою затримкою, наприклад, у потоковому відео в реальному часі.

- Бездротові ретранслятори на основі БПЛА покращують покриття мобільних користувачів [3].

За апаратним забезпеченням можна переважно розділити на дві основні групи:

- 1) БПЛА з нерухомим крилом (*FW-UAVs*) — невеликі літальні апарати з нерухомими крилами. *FW-UAV* мають аеродинамічну конструкцію, схожу на звичні для нас літаки, що дозволяє їм літати на далекі відстані зі збереженням високої швидкості і енергоефективності. Завдяки своїй конструкції зазвичай використовуються для довготривалих місій, таких як розвідка, моніторинг територій або пошуково-рятувальні операції.

- 2) Гвинтокрилий БПЛА (*RW-UAVs*) — це безпілотні літальні апарати з поворотними гвинтами, як, наприклад, квадрокоптери або гелікоптери. Вони мають здатність вертикального зльоту та посадки (*VTOL*), а також можуть зависати на місці, що робить їх ідеальними для виконання завдань у тісних просторах, точних маневрів та операцій на малих висотах. *RW-UAVs* часто використовуються для спостереження, огляду об'єктів, доставки вантажів або проведення рятувальних операцій.

За висотою польоту БПЛА можна класифікувати на два типи: високо-висотні платформи (*HAP*) і низько-висотні платформи (*LAP*).

- HAP (*High-Altitude Platform*) є квазістаціонарними і можуть літати до 17 кілометрів над поверхнею землі. HAP можуть забезпечити бездротове покриття великих географічних територій за допомогою польотів до кількох місяців і розроблені для довгострокових застосувань, але потребують багато часу для розгортання та є дорожчими порівняно з LAP.
- LAP (*Low-Altitude Platform*) можуть літати над землею на висоті до кількох кілометрів. LAP мають високу мобільність і гнучкіші, ніж HAP. Завдяки низькій вартості та легкому розгортанню використовуються в екстрених та непередбачуваних ситуаціях. За потреби LAP можна перезаряджати або замінювати під час польоту. Встановлення зв'язку LoS на короткій відстані з наземними користувачами робить їх більш ефективними та значущими.

Залежно від застосувань і цілей, різні типи БПЛА використовуються в різних сценаріях відповідно до їх можливостей і функцій.

1.2 Мобільні периферійні обчислення

MEC — це відносно нова концепція, яка з'явилася в 2014 році Групою галузевих специфікацій (*ISG*) Європейського інституту стандартів телекомунікацій (*ETSI*) для мереж п'ятого покоління (*5G*) [5]. У вивчених мною джерелах інформації це визначено було так: «Мобільні граничні обчислення надають середовище ІТ-послуг і можливості хмарних обчислень у мережі радіодоступу (RAN) у безпосередній близькості до мобільного пристрою». Пізніше це визначення було дещо розширено у більш сучасних розробках і проектах [6]: «Граничні обчислення стосуються широкого набору методів, призначених для переміщення обчислень і зберігання даних за межі віддаленої хмари (загальнодоступної чи приватної) і ближче до джерела даних». Згідно з цією концепцією, обчислювальні ресурси виводяться на межу мережі поблизу мобільних пристроїв.

Через обмеження потужності та часу автономної роботи мобільних пристроїв *Mobile Edge computing/Multi-access edge computing* (MEC) є багатообіцяючим і найкращим рішенням цієї проблеми, яке може вирішити конфлікт між пристроями з

обмеженими ресурсами та ресурсоемними завданнями, розміщення обчислювальних серверів на периферійних вузлах, таких як базові станції (*BS*), або пристрої користувача з високою обчислювальною здатністю, ближче до мобільних пристроїв.

МЕС — це розподілений підхід і складається з мобільних користувачів і сервера МЕС. Мобільні пристрої передають часткові або повні обчислювальні завдання в тісно пов'язану потужну периферійну обчислювальну хмару (межовий сервер/сервер МЕС) для обчислень, відомих як розвантаження обчислень, що економить енергію, зменшує затримку та підвищує швидкість обробки пристроєм для відповідності QoS (якість обслуговування) і QoE (якість досвіду).

1.3 Співпраця БПЛА та МЕС

Розташування наземного сервера МЕС зазвичай є фіксованим і не може бути змінено або переміщено відповідно до потреб мобільних користувачів, що обмежує можливості сервера [7]. Наземні сервери МЕС не працюють з обмеженою доступною інфраструктурою, такою як поля битв, аварійні та рятувальні операції, пустельні території тощо.

Завдяки високій гнучкості і мобільності, легкому розгортанню, низькій вартості та невеликому розміру БПЛА - його можна розгорнути як вузол периферійних серверів, створюючи вдосконалений БПЛА периферійний/інтелектуальний сервер МЕС, який надає мобільні периферійні послуги та зв'язок на вимогу і обчислювальних ресурсів для користувачів, якщо фіксовані наземні мережі МЕС недоступні або легко встановлені, або де стихійні лиха руйнують системи.

Наведено такі основні переваги системи МЕС з БПЛА:

- ✓ Система U-МЕС використовує встановлення зв'язку LoS для забезпечення ширшого спектру застосувань, що робить систему гнучкою та ефективною.
- ✓ Система U-МЕС покращує обчислювальні послуги, надаючи користувачам високу пропускну здатність (велике покриття) при цьому покращуючи пропускну здатність системи.

- ✓ Система U-MEC забезпечує розвантаження обчислень енергозберігаючим способом з низькою затримкою. Це здійснюється для підвищення продуктивності системи, незважаючи на різні застосування та переваги.

Обмежена батарея, потужність, низька затримка, пришвидшення високих з'єднань, взаємодія кількох користувачів і обмеження SWaP БПЛА створюють нові виклики для систем U-MEC. Тому енергоефективна модель є ключовою. Також враховується енергоспоживання рушійної сили, необхідне БПЛА, щоб залишатися в повітрі.

1.4 Система MEC з підтримкою БПЛА

У цьому підрозділі представлені мережі MEC з підтримкою БПЛА та наведені три можливі архітектури в системах U-MEC. Модель системи наведена на рисунку 1.1.

U-MEC використовує три основні моделі:

- 1) Модель локального обчислення, яка виконує завдання локально на мобільних пристроях.
- 2) Модель розвантаження обчислювальних завдань, у якій користувач розвантажує завдання, яке має виконати сервер MEC на БПЛА.
- 3) Модель зависання/польоту БПЛА, в якій завдання завантажуються у фіксоване місце на деякий час.

У стаціонарних наземних серверах MEC мобільні користувачі локально виконують свої завдання на вбудованих мікропроцесорах, споживаючи при цьому велику кількість енергії [8].

Однак в U-MEC мобільний пристрій вирішує або обчислювати важкі завдання локально, або перевантажувати такі важкі завдання на сервер MEC на базі БПЛА, розташований у дроні (граничні обчислення), ближче до користувача, заощаджуючи його енергію та трафік та навантаження на стаціонарні сервери.

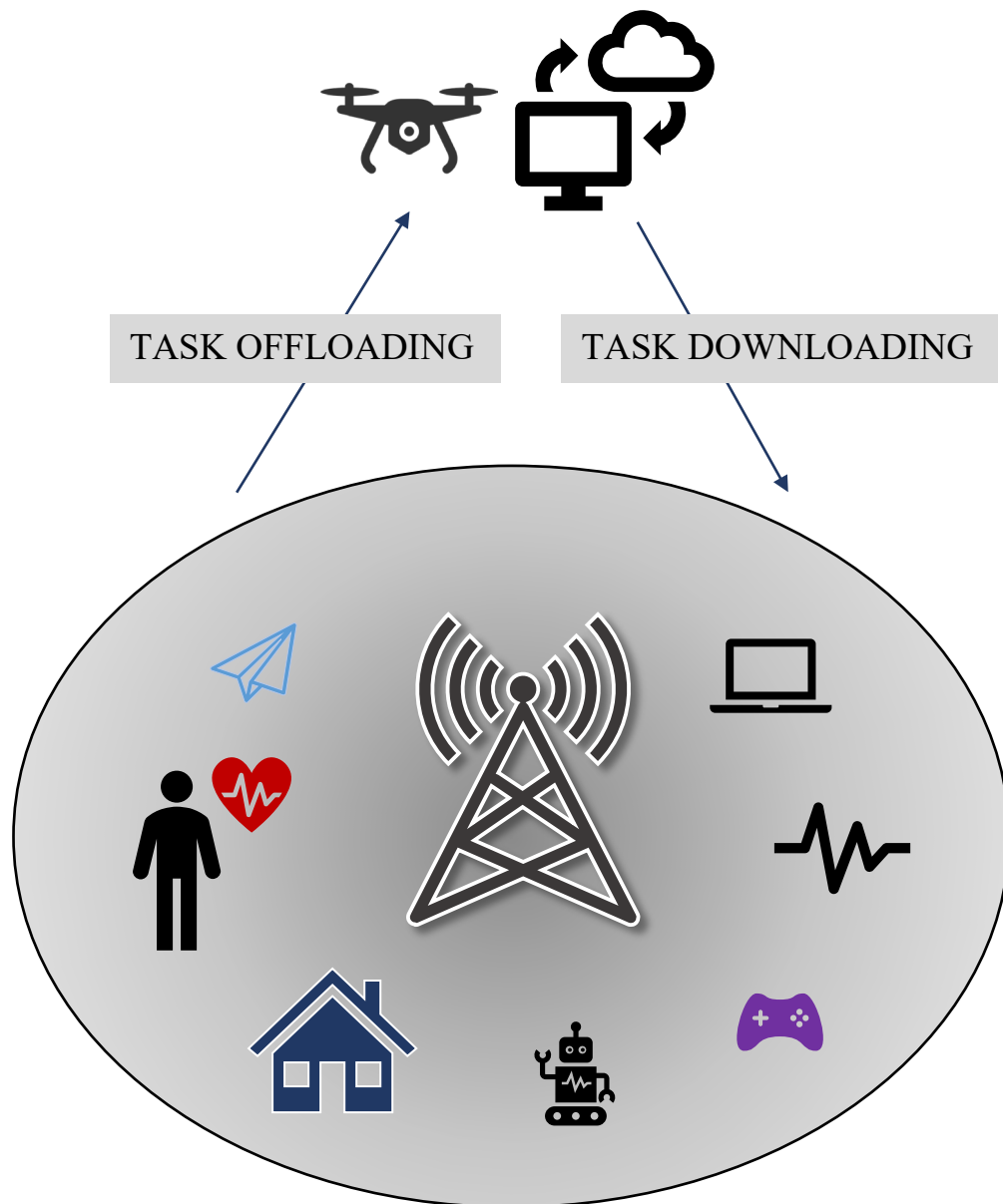


Рис. 1.1 Сервер MEC з підтримкою БПЛА

Потім БПЛА виконує завдання як сервер MEC від свого імені та передає обчислення назад на мобільний пристрій. Нарешті, користувач мобільного пристрою завантажить результати.

Кожен мобільний пристрій пов'язаний з U-MEC, що має достатньо ресурсів і акумулятора. Завдяки зв'язку LoS між дроном та наземними мобільними користувачами можна збільшити потужність розвантаження та завантаження, а також розширити охоплення в системі MEC, що підтримує БПЛА.

1.5 Можливі архітектури в системі U-MEC

Архітектури U-MEC мають три можливі сценарії, засновані на ролі, яку відіграють БПЛА. У кожному сценарії двома основними компонентами є мобільні пристрої/кінцеві користувачі та сервери MEC, які є невеликими центрами обробки даних, які встановлюють оператори зв'язку ближче до користувачів. Користувачі та сервери з'єднані один з одним через бездротове з'єднання за допомогою бездротових мереж і технологій зв'язку. Сервер пов'язаний із хмарними дата-центрами через Інтернет, шлюз [9].

Допоміжний U-MEC: перша архітектура показана на рисунку 1.2.

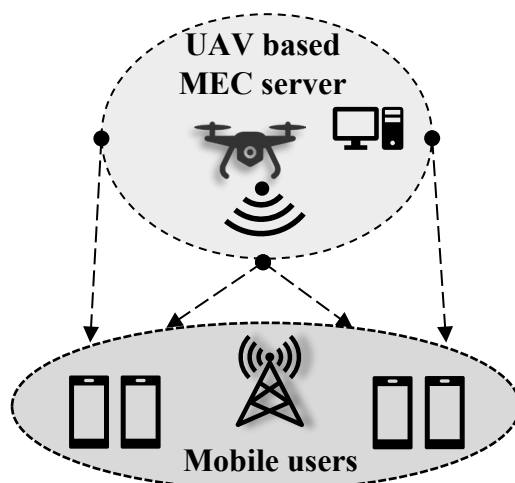


Рис. 1.2 БПЛА оснащений MEC сервером

У цьому сценарії БПЛА діє як повітряна базова станція з підтримкою серверу MEC, яка допомагає наземним користувачам, на яку користувачі передають свої інтенсивні обчислювальні завдання для виконання. один або декілька БПЛА. Крім того, він застосовний у регіонах, де наземна інфраструктура обмежена або відсутня, а наземні базові станції не можуть надавати послуги через несподівані події або реагування на стихійні лиха. Така архітектура зазвичай використовується для мінімізації загального споживання енергії пристроями, що відповідають вимогам QoS.

U-MEC із стільниковим зв'язком: друга архітектура показана на рисунку 1.3. У цьому сценарії БПЛА діє як повітряний користувач, який має важкі обчислювальні завдання, наприклад, оптимізацію траєкторії, щоб розвантажити наземні базові станції (*GBS*), обладнані Сервером MEC (наземний фіксований сервер MEC/хост MEC) для віддаленого обчислення.

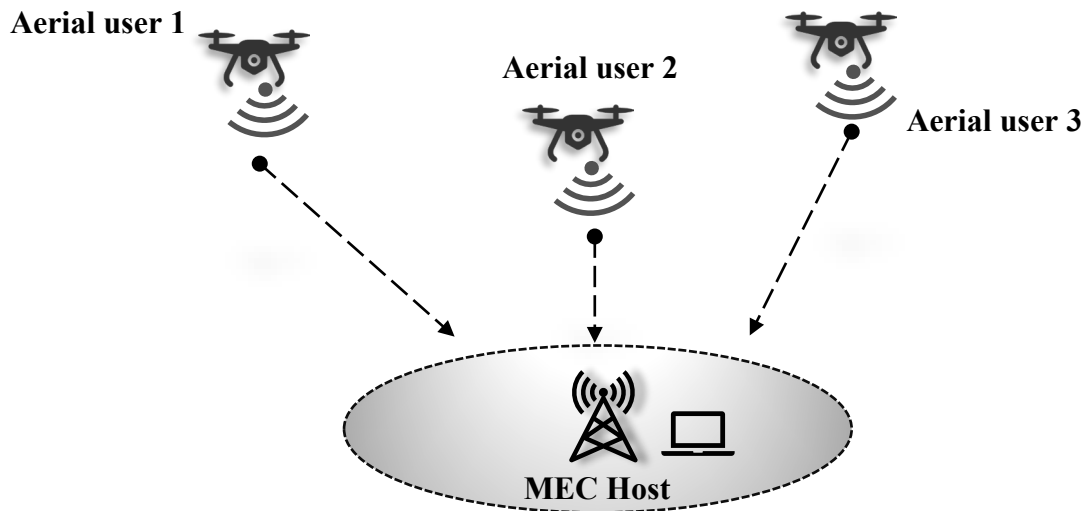


Рис. 1.3 GBS оснащений сервером MEC

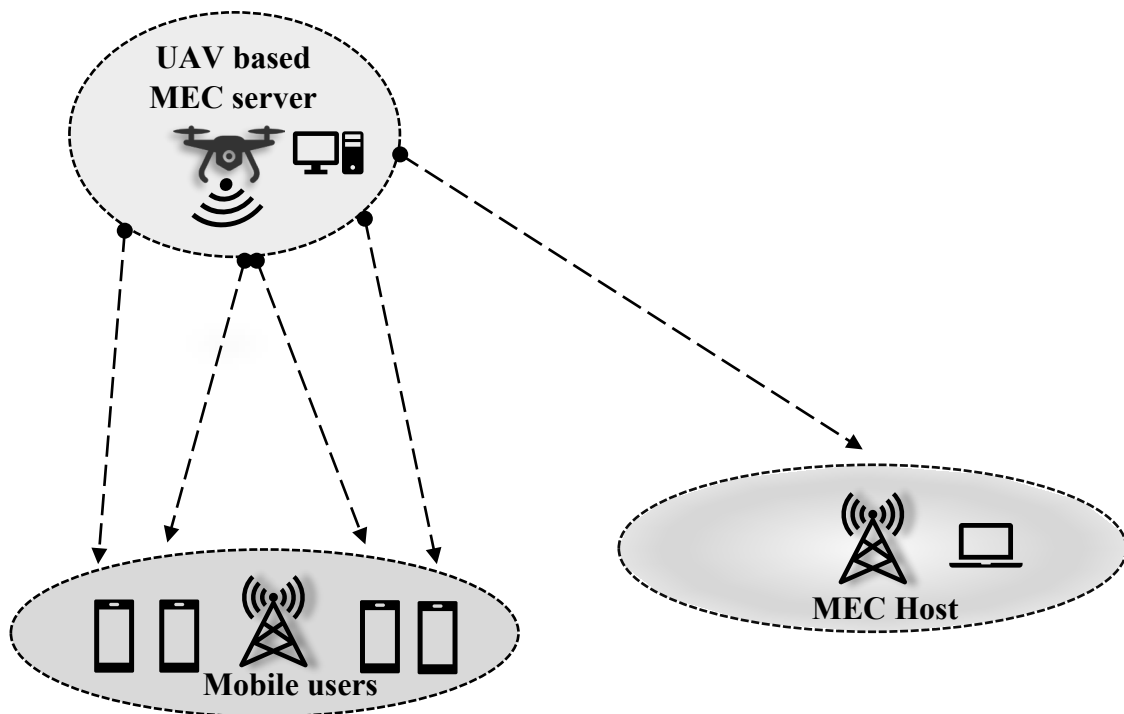


Рис. 1.4 БПЛА в якості ретранслятора

Подібна архітектура як представлено на рисунку 1.3 застосовується в областях, де БПЛА має обмежену ємність бортової батареї та обчислювальних можливостей, але він повинен виконувати завдання, що потребує інтенсивних обчислень.

Ретрансляційний U-MEC: третя архітектура показана на рисунку 1.4. У цьому сценарії БПЛА працює як ретранслятор, який допомагає користувачам перекласти важкі обчислювальні завдання на GBS, інтегровані з сервером MEC [10].

Ця архітектура застосовна там, де зв'язок між користувачами та наземним сервером MEC є недосконалим. Також БПЛА не оснащений сервером MEC.

1.6 MEC: завдання, програми і майбутні напрямки

У сучасну епоху технологій використання планшетів і смартфонів відіграє важливу роль у будь-якій ситуації. Оскільки кількість користувачів мобільного зв'язку зростає, якість обслуговування (QoS) і якість досвіду (QoE) стикаються з більшими проблемами. Таким чином, це може значно зменшити затримку та оптимізувати споживання електроенергії завданнями, що виконуються локально.

Більшість робіт зосереджені лише на оптимізації якості в динамічних сервісних макетах. Однак вони проігнорували значний вплив точного вибору мережі доступу та ідеального розміщення послуг. У зв'язку з цим опитування класифікує підходи на основі розміщення послуг, вибору мережі, параметрів QoS і QoE, а також таких ресурсів, як затримка, енергія, пропускна здатність, пам'ять, сховище та обробка.

1.6.1 Пов'язані концепції та технології

На рисунку 1.5 представлено детальний аналіз різних периферійних парадигм із впровадженням інфраструктури від CORE до EDGE. У таблиці 1.1 представлено детальне порівняння MEC, FC, MCC і Cloudlet. З порівняння видно, що MEC забезпечує кращу продуктивність QoS порівняно зі звичайними парадигмами периферійних обчислень. Нарешті, відкриті дослідницькі проблеми та виклики MEC

ефективно аналізуються та використовуються для розробки нових тенденцій досліджень, які відсутні у відповідній роботі.

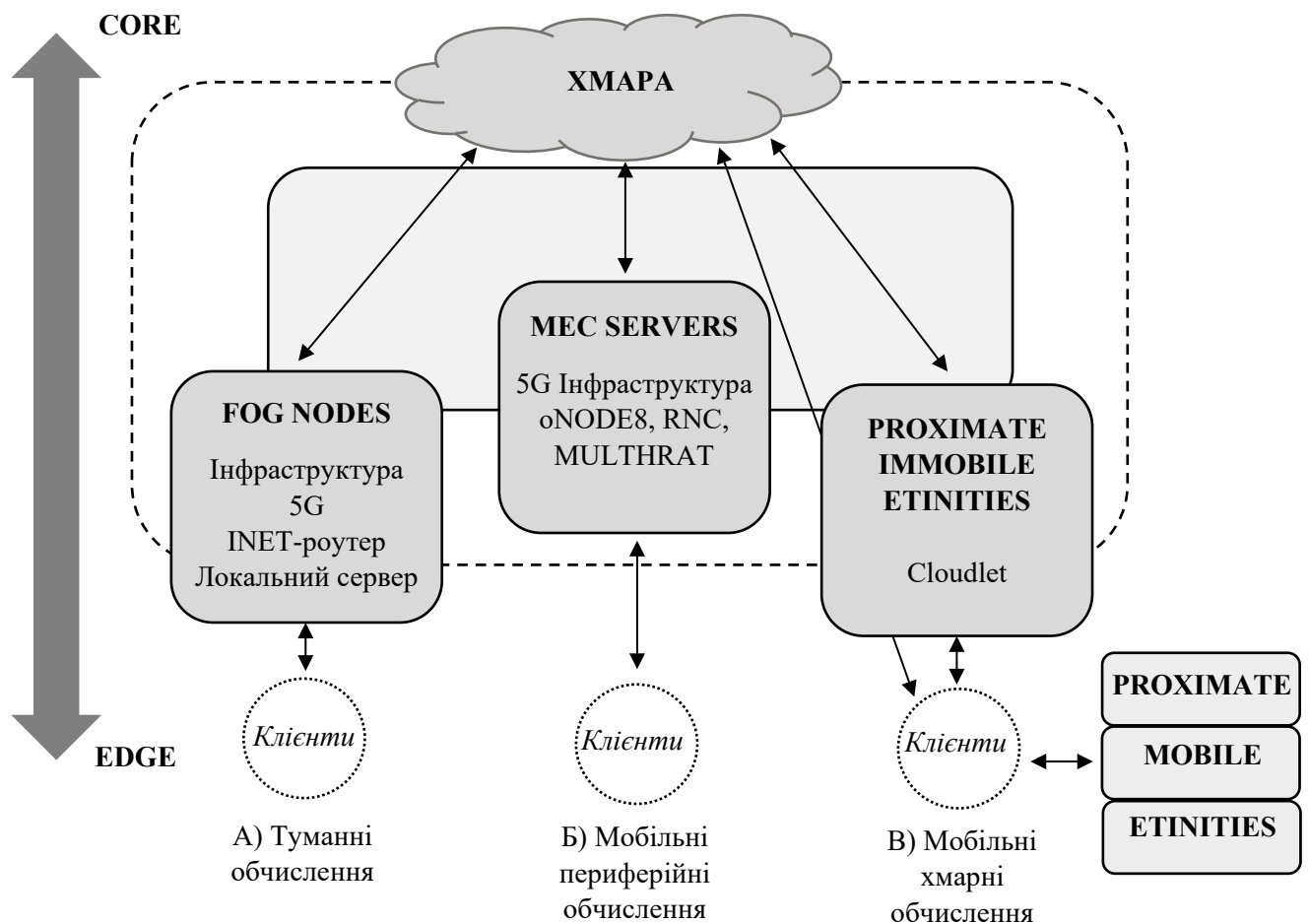


Рис. 1.5 Спрощений огляд основних парадигм Edge

MCC – *Mobile cloud computing*: мобільні хмарні обчислення. Поєднують у собі мобільні мережі та хмарні обчислення (*cloud computing, CC*). Методи мобільних і хмарних обчислень об'єднані та сформовані як модель MCC. Ця модель забезпечує достатні енергетичні ресурси, обчислювальні та сховища для мобільних користувачів через централізовану хмару.

Згідно з MCC, хмарний сервер демонструє необмежені ресурси, і він знаходиться далеко від мобільного користувача (*MU*). Таким чином, відстань між MU та хмарним сервером збільшує затримку доступу. Основна мережа MCC має

зменшену обчислювальну потужність і ресурси. Запити на доступ, створені з МУ, ніколи безпосередньо не виконуються базовою мережею.

Таблиця 1.1

Порівняння характеристик парадигм Edge

Критерій	Cloudlet	MCC	FC	MEC
Масштабованість	середній	високий		
Доступність	високий			
Місцева обізнаність	-	так		
Затримка, джиттер	середній	низький		
Мобільність	-	так		
Архітектура мережі	централізована	розподілена, централізована		
Сервіс	віртуалізація	віртуалізація, інші	віртуалізація	
Обладнання	сервери	сервери, UD	гетерогенні сервери	
Розгортання	ядро мережі	край мережі, пристрої	близький край, край	
Право власності	приватні особи	приватні особи, фізичні особи		телек. комп.

FC – *Fog computing*: туманні обчислення. MCC широко використовується в мережах IoT нижчого рівня, але він все ще створює численні проблеми, такі як визначення місця розташування, відсутність підтримки мобільності та ненадійні затримки в мережах IoT. Туманні обчислення використовуються мережами IoT для вирішення вищезазначених проблем. FC ефективно розподіляє гнучкі послуги та ресурси для МУ, підтримує як підключені, так і непідключені пристрої. Спочатку FC був розроблений компанією *Cisco Systems* для надання необмежених послуг корпоративній мережі, підключеній до периферії. Туманні обчислення пропонують меншу затримку порівняно зі звичайними підходами хмарних обчислень. У будь-якому випадку метод страждає від більш надійного безперервного бездротового з'єднання. Невелике переривання з'єднань призводить до збою процесу. Таким чином, складні операції не виконуються точно через FC, і він підтримує інтелект лише в локальній мережі. Таким чином, більша кількість атак і загроз впливає на

продуктивність і конфіденційну інформацію користувачів. Таким чином, МЕС ефективно використовується для підтримки безпеки як на рівні IoT, так і для зв'язку між мобільними телефонами. Крім того, також підтримує різноманітні операції в мережах IoT зі скороченими обчисленнями.

Cloudlets: *Cloudlet* — це невелике джерело хмарної мережі, сформованої з окремих користувачів та груп із кількох. Ці хмарлети використовуються переважно в офісних будівлях, торгових центрах і громадських місцях, таких як транспортні зони та лікарні з невеликими серверами даних. Основна мета Cloudlet полягає в тому, щоб з'єднати різні хмарні технології та надати максимальну кількість ресурсів кінцевим користувачам із скороченою затримкою. В основному залежать від технологій Wi-Fi, WMAN і WLAN. Таким чином, якщо в цих технологіях генеруються будь-які переривання, то з'єднання з одним і кількома переходами між користувачами, підключеними до Інтернету, закриваються. Cloudlet також страждає від проблем конфіденційності та безпеки, пов'язаних із проблемами конфіденційності на основі доступу. Таким чином, МЕС є хорошим способом вирішення цих проблем, і він працює з усіма стандартами комунікаційних технологій.

МЕС— Mobile edge computing: мобільні периферійні обчислення. МЕС відіграє важливу роль у розвиненому зв'язку 4G і 5G, оскільки він долає проблеми, створені системами Cloudlets, FC і MCC відповідно. Воно складається з багатофункціональних MCC для виконання всіх типів розподілу ресурсів для кінцевих користувачів мережі радіодоступу (*RAN*). Але ці методи не змогли забезпечити максимальну ефективність у високих обчислювальних середовищах і призвели до зниження коефіцієнта розвантаження даних. Таким чином, МЕС оснащений орієнтованим на споживача блоком керування послугами розвантаження, блоком керування послугами на основі третьої сторони оператора та блоком керування послугами продуктивності мережі та покращення QoE, як показано на рисунку 1.6.

Основна мета МЕС полягає в тому, щоб маршрутизувати пакети між різними програмами на основі їхніх служб. Маршрутизація виконується на основі незалежного режиму, режиму відведення, внутрішнього режиму та режиму прориву відповідно. Захищене сеансове з'єднання буде переоцінене між віддаленим сервером і

швидкість розвантаження з покращеною смугою пропускання та зменшеною затримкою мережі завдяки ідеальному виконанню різних режимів операцій.

Однак технології MEC страждають від різноманітних проблем у сценаріях реального часу [15]. Були проведені різні дослідження, але вони все ще обмежені. Як наслідок, ретельний аналіз різних звичайних технологій MEC може забезпечити значні рішення для всіх проблем, які виникають у периферійній мережі. Опитування також розкриває неохоплену сферу потенційних досліджень і визначає основні напрямки дослідження.

1.7 Структура та архітектура MEC

Технологія MEC, нещодавно представлена групою галузевих специфікацій ETSI (*ISG*) з мобільних периферійних обчислень (*MEC*) (тобто *ETSI MEC ISG*), пропонує можливості хмарних обчислень у межах RAN та середовища IT-сервісів на межі мобільної мережі. , близько до абонентів мобільного зв'язку. Це середовище характеризується наднизькою затримкою, високою пропускнуою здатністю та доступом у режимі реального часу до інформації про радіомережі, яка може бути використана програмами та платформами оптимізації QoE.

«Край» може стосуватися як самих базових станцій (наприклад, eNodeB, контролер радіомережі тощо), так і центрів обробки даних поблизу радіомережі (наприклад, розташованих у «точках агрегації»). Оператори можуть відкривати свій край RAN для авторизованих третіх сторін, що дозволяє їм гнучко та швидко розгортати інноваційні програми та послуги для мобільних абонентів, підприємств і вертикальних сегментів.

MEC принесе значні переваги не лише операторам, але й третім сторонам і OTT (*Over-The-Top*) компаніям, які матимуть можливість запускати свої програми на межі мобільної мережі, поблизу мобільних абонентів (з пов'язаними новими бізнес-можливостями). Нарешті, звичайний споживач також отримає можливість відчути покращену продуктивність і нові послуги, які пропонує система MEC.

ETSI MEC ISG було запущено в грудні 2014 року, і йому було поставлено завдання розробити набір специфікацій для забезпечення розміщення додатків третіх сторін у середовищі багатьох постачальників.

З тих пір ISG працює над специфікаціями, починаючи з варіантів використання для визначення вимог, а потім – визначення архітектури. Роботу було поставлено на 2 роки. ISG MEC вже опублікувала базові специфікації щодо технічних вимог і еталонної архітектури.

ISG перейшов до наступного етапу, щоб визначити платформу підтримки додатків, мобільні периферійні служби та необхідні API, які мобільні периферійні програми можуть використовувати для надання цілого нового спектру послуг із розширеним досвідом кінцевого користувача.

Структура Mobile Edge Computing, представлена на рисунку 1.7, показує задіяні функціональні об'єкти високого рівня. Об'єкти далі групуються на системному рівні, рівні хоста та об'єктах мережевого рівня.

Рівень мобільного граничного хосту складається з мобільного граничного хосту та відповідного об'єкта керування рівня.

Хост ME додатково розділений на мобільну периферійну платформу, мобільні периферійні програми та інфраструктуру віртуалізації. Рівень мереж складається з пов'язаних зовнішніх об'єктів, якими є стільникова мережа 3GPP, локальні мережі та зовнішні мережі. Цей рівень забезпечує підключення до локальних мереж, стільникових мереж і зовнішніх мереж, таких як Інтернет.

Крім усього іншого, це управління на рівні мобільної периферійної системи, яке за визначенням має загальну видимість усієї мобільної периферійної системи.

Мобільна периферійна система складається з хостів ME та мобільного периферійного керування, необхідних для запуску мобільних периферійних додатків у мережі оператора або її підгрупі.

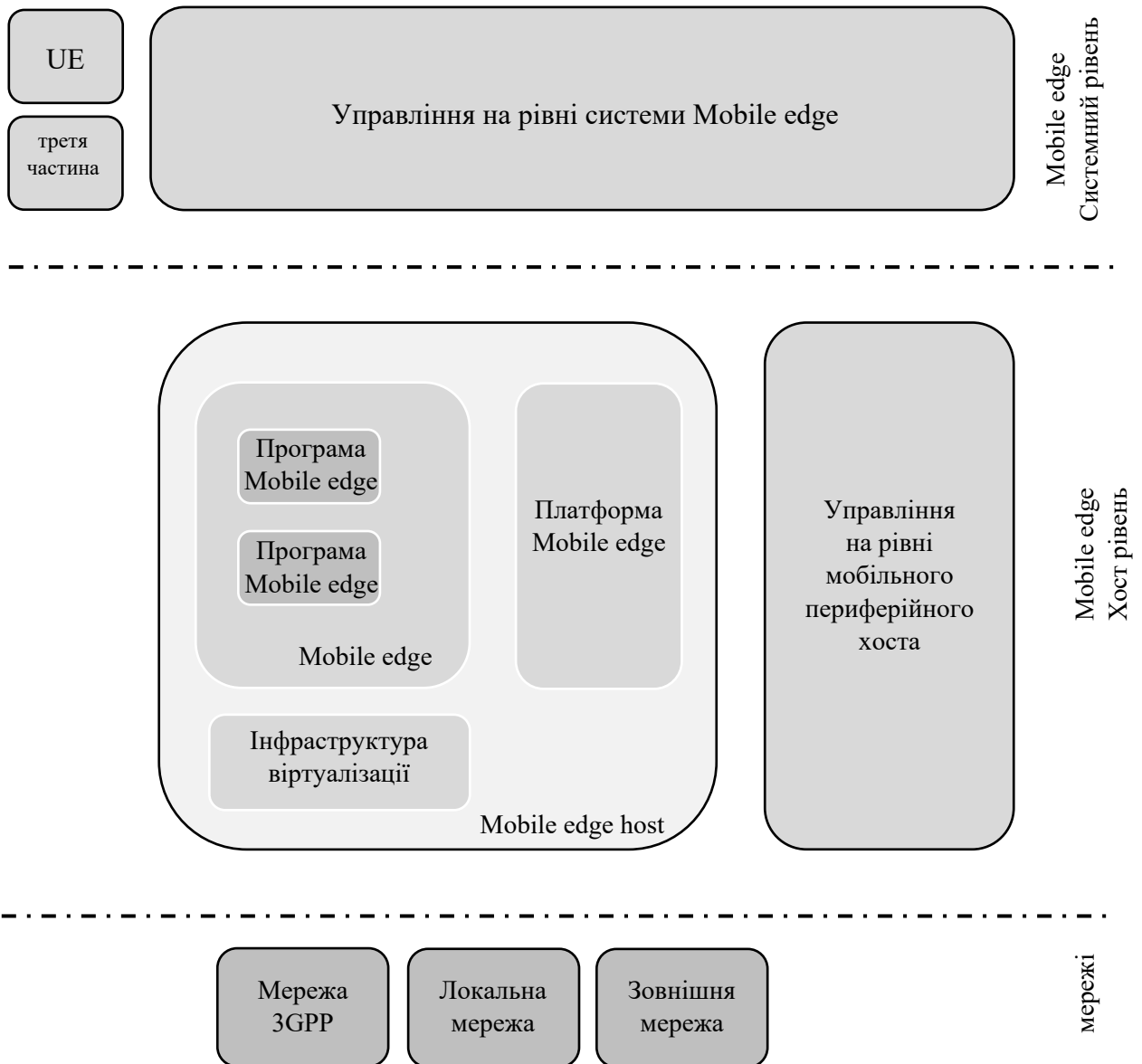


Рис. 1.7 Інфраструктура мобільних периферійних обчислень

Більш повне розуміння мобільної периферійної системи можна отримати за допомогою еталонної архітектури (зображеної на рисунку 1.8 нижче), яка більш детально визначає функціональні об'єкти та їхні зв'язки один з одним.

Еталонна архітектура має такий самий поділ на системний рівень і рівень хоста, що й фреймворк. Мережевий рівень не відображається в еталонній архітектурі, оскільки для доступу до цих об'єктів не потрібні специфічні контрольні точки Mobile Edge Computing.

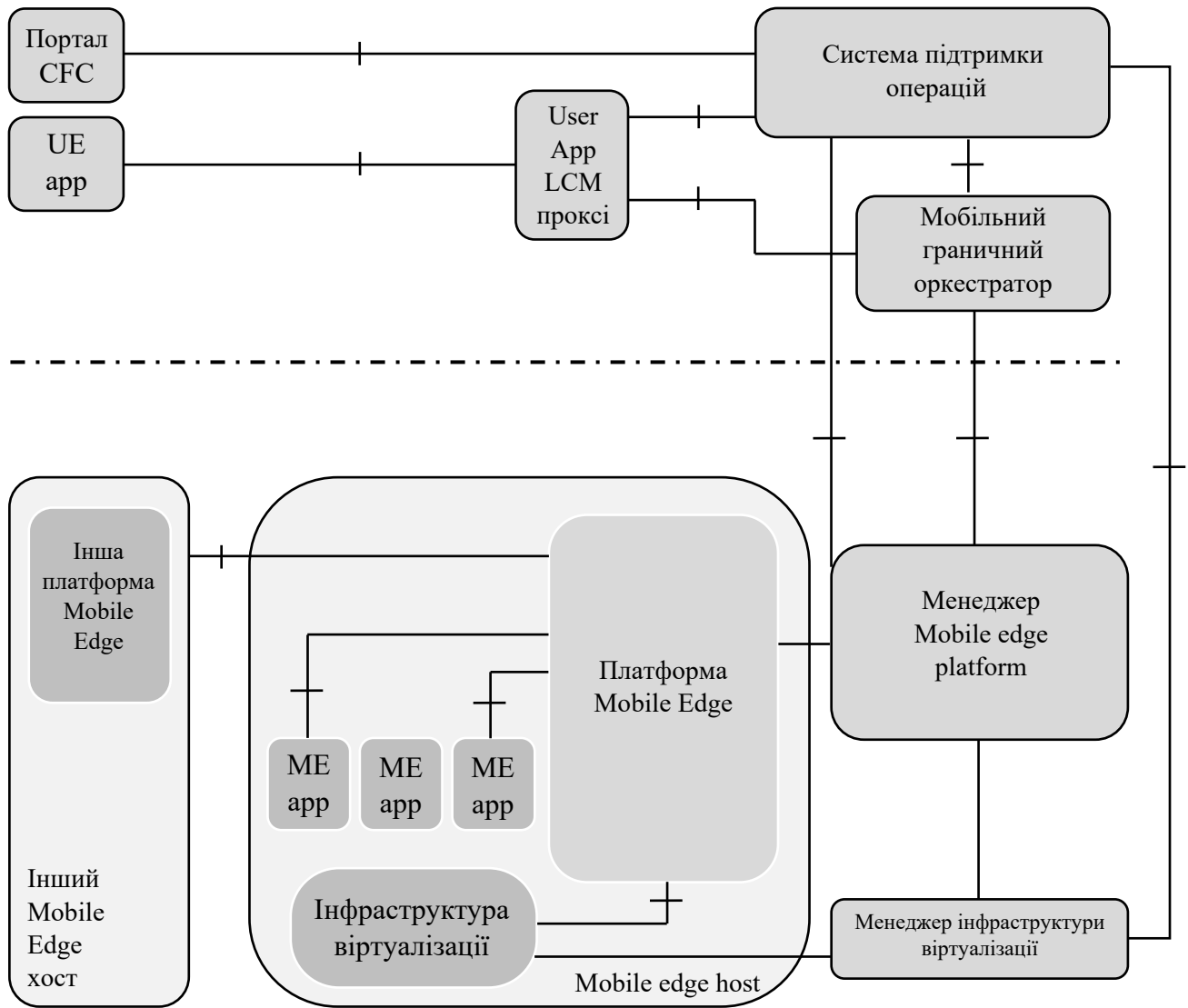


Рис. 1.8 Еталонна архітектура Mobile Edge Computing

Мобільний граничний хост (*ME-хост*) — це об'єкт, що складається з мобільної граничної платформи та інфраструктури віртуалізації, яка надає обчислювальні ресурси, сховище та мережеві ресурси для мобільних граничних додатків. Крім того, хост ME може забезпечити постійне зберігання та інформацію про час доби для програм. Інфраструктура віртуалізації включає площину даних, яка виконує правила пересилання, отримані мобільною периферійною платформою, і направляє трафік між програмами, службами та мережами.

Мобільна периферійна платформа (*платформа МЕ*) представляє набір базових функціональних можливостей, необхідних для запуску програм на певному хості МЕ та для того, щоб мобільні периферійні програми могли виявляти, рекламувати пропозиції та використовувати послуги МЕ. Мобільні граничні послуги можуть надаватися платформою та додатками, і платформа, і додатки можуть використовувати послуги МЕ.

Основні базові функції платформи МЕ необхідні для керування трафіком між програмами, службами та мережами. Платформа МЕ отримує правила пересилання трафіку від менеджера мобільної крайової платформи, мобільних крайових додатків і служб МЕ, і на основі них, а також на основі політик, надає інструкції площині пересилання. Крім того, платформа МЕ підтримує налаштування локального DNS-проксі/сервера, який можна використовувати для спрямування трафіку користувачів до бажаних мобільних додатків.

Платформа МЕ може спілкуватися з іншими платформами МЕ через опорну точку MP3, яка призначена для процедур літака управління. За допомогою цього інтерфейсу кілька платформ можна згрупувати разом і сформувати комунікаційну мережу. Мобільні периферійні програми (МЕ-програми) працюють як віртуальні машини поверх інфраструктури віртуалізації, наданої хостом МЕ. Додатки взаємодіють із платформою МЕ через опорну точку Mr1, щоб використовувати послуги, які пропонує платформа. Програми також можуть пропонувати надання послуг платформі, яка надалі може надавати їх іншим програмам.

Еталонна точка Mr1 також використовується для додаткових процедур підтримки, наприклад для вказівки на доступність програми або для підготовки до переміщення стану програми для користувача у разі подій передачі. Додатки МЕ можуть вказувати вимоги щодо необхідних ресурсів або послуг, а також вказувати свої обмеження щодо максимально дозволених затримок.

2 РОЗВАНТАЖЕННЯ МОБІЛЬНИХ ПЕРИФЕРІЙНИХ ОБЧИЛЕНЬ

У цьому розділі підсумовуються методи, які використовуються в обчислювальному процесі наземних мобільних користувачів, наприклад, у локальному обчисленні. Крім того, зосереджено увагу на процесі розвантаження обчислень, різних режимах роботи для розвантаження завдань на БПЛА та рішеннях, які приймаються під час цього процесу.

Обчислювальне розвантаження — це процес передачі обчислювальних завдань від мобільних пристроїв (MD) до зовнішніх джерел для виконання через бездротовий доступ, наприклад крайовий сервер, який має достатні обчислювальні ресурси для відповідних завдань. Граничний сервер, у свою чергу, надсилає результати назад на мобільний пристрій.

MECO (від англ. *MEC-Oriented*, або *Mobile Edge Computing-Oriented*) — це термін, який використовується для опису підходів, рішень або архітектур, спрямованих на впровадження та оптимізацію мобільних периферійних обчислень.

MECO корисний, якщо пристрій не в змозі виконувати важкі завдання. Наприклад, кодування відео – енерговитратна задача. Як рішення, служби кодування розвантажуються на периферійні сервери для виконання, що заощадить енергію та зменшить затримку пристрою, забезпечуючи хорошу якість відео.

MECO є ключовою парадигмою в МЕС, оскільки вона має кілька переваг:

- Термін служби батареї MD можна продовжити, уникаючи локального виконання.
- Перенесення завдань на сервер МЕС зменшить споживання енергії за рахунок збільшення продуктивності системи та прискорення обчислень.
- Мінімізує загальний час виконання, а також покращує взаємодію з користувачем і якість послуг.

2.1 Процес розвантаження обчислень

Сервер МЕС виконує наступні кроки перед виконанням розвантаження обчислень, як показано на рисунку 2.1.

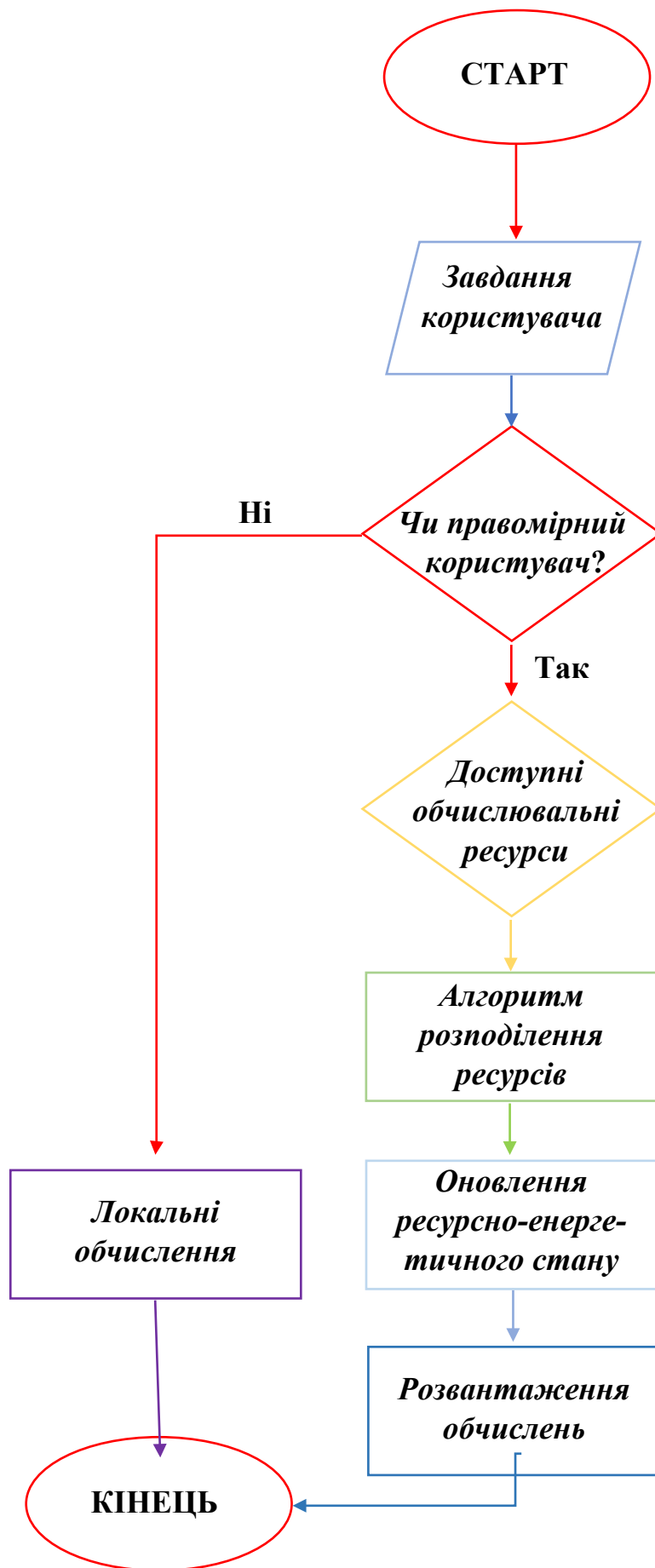


Рис. 2.1 Загальна блок-схема для розвантаження обчислень

- В першу чергу сервер МЕС перевіряє достовірність звернення користувача за послугою розвантаження обчислень.
- Якщо користувач є перевіреним, обчислювальні ресурси призначаються користувачеві, зберігаючи обсяг даних про завдання та використання ресурсів за допомогою алгоритму розподілу ресурсів. Водночас обчислювальні ресурси та енергетичні умови миттєво оновлюються. Тепер користувач може виконувати розвантаження обчислень на сервер МЕС.
- Здебільшого мета алгоритму розвантаження обчислень полягає в тому, щоб мінімізувати енергію, споживану MD, шляхом переміщення завдань на сервер МЕС, одночасно зменшуючи затримку виконання відповідно. Можна досягти 90% економії енергії та зменшити затримку на 98%. Отже, компроміс між обома можна оптимізувати за допомогою розвантаження обчислень [16].

2.2 Режими роботи для розвантаження обчислень

Розвантаження обчислень з мобільних пристроїв на U-МЕС може здійснюватися в трьох режимах роботи:

1) Локальний обчислювальний процес, у якому всі завдання виконуються локально. Більшість пристроїв обробляють дані без доступу до сервера МЕС. Швидкість читання даних і продуктивність центрального процесора впливають на ефективність локального обчислювального процесу. Це корисно, якщо сервер МЕС недоступний, зв'язок між користувачем і сервером поганий.

Система U-МЕС має два методи локальних обчислень [17].

- *Постійна частота:* для фіксованої частоти процесора обчислювальної схеми пристрою локальні обчислення здійснюються з фіксованою постійною швидкістю.
- *Динамічна частота:* для динамічної частоти ЦП мобільного пристрою локальні обчислення виконуються за допомогою методів динамічного масштабування напруги та частоти, у яких напруга або частота ЦП регулюються/знижуються відповідно до обчислювальних завдань, які мають бути виконані,

щоб заощадити енергію та затримку в складних обчислювальних схемах. Цей метод є більш потужним, ніж фіксований, щоб зменшити споживання енергії всією системою.

2) Двійковий/повний процес розвантаження, у якому завдання не можна розділити на під завдання, тобто всі завдання виконуються локально або завершуються, вивантажуються на сервер МЕС у цілому для обчислення, а кінцеві результати надсилаються назад на мобільні пристрої.

Ефективність процесу повного розвантаження залежить від обчислювальної потужності сервера U-МЕС, пропускної здатності каналу та стану каналу. Гнучкість схем розподілу ресурсів може бути обмежена, оскільки БПЛА не може виконувати локальні обчислення та розвантажувати завдання одночасно.

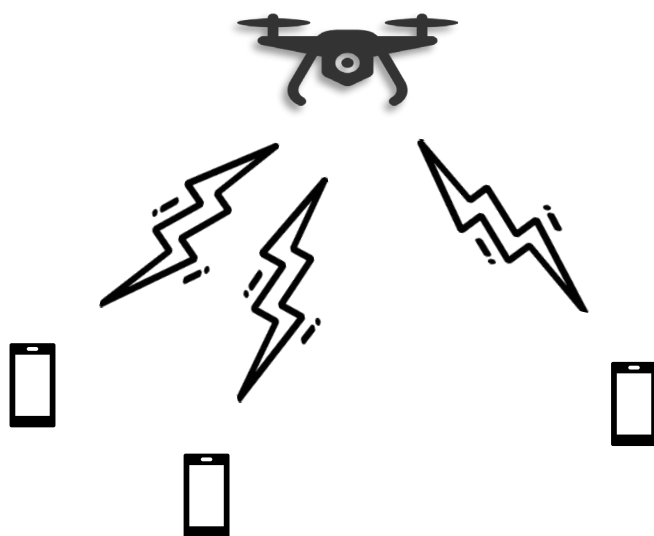


Рис. 2.2 Режими розвантаження обчислень

3) Процес часткового розвантаження, у якому завдання можуть бути розділені на під завдання, тобто деякі обробляються локально, а деякі з них, які споживають більше енергії або є складними, розвантажуються на сервер U-МЕС. Для обчислень використовуються як сервер U-МЕС, так і мобільні пристрої, а остаточні

результати отримують після поєднання обох окремих обчислень. Такі фактори, як передача даних, споживання енергії, обробка даних завдань і розподіл ресурсів, впливають на ефективність цього процесу, також впливають один на одного. Він більш ефективний, ніж двійковий режим, оскільки БПЛА може динамічно розподіляти обчислювальні ресурси для локальних обчислювальних і комунікаційних процесів. Але цей режим є більш складним, ніж інші, оскільки він потребує поєднання кількох факторів.

Вибір одного з режимів залежить від програми, яку використовує користувач, особливостей обчислювальної задачі та структури БПЛА. На рисунку 2.2 показано процес розвантаження режимів у мережах U-MEC.

2.3 Схеми доступу до розвантаження обчислень

Розвантаження обчислень забезпечується зв'язком по висхідній і низхідній лінії зв'язку між мобільними користувачами та БПЛА з використанням різних методів/схем, які є більш ефективними, ніж наземні мережі MEC, завдяки використанню БПЛА в мережі U-MEC. Мною досліджено і розглянуто найбільш часто використовувані схеми розвантаження обчислень у системі U-MEC.

Схеми множинного доступу

Схеми множинного доступу дозволяють кільком мобільним користувачам спільно використовувати та отримувати доступ до одного каналу та обмежених ресурсів, щоб уникнути перешкод. Він класифікується на дві групи:

І) Схема ортогонального множинного доступу (ОМА): кожен наземний користувач може використовувати ортогональні ресурси з точки зору смуги частот, що називається схемою множинного доступу з ортогональним частотним поділом (OFDMA), і часовим інтервалом, відомим як схема множинного доступу з тимчасовим поділом каналів (TDMA) [18], щоб пом'якшити багаторазовий доступ. Періодичний/циклічний TDMA також використовується для зв'язку між мобільними користувачами та БПЛА, коли БПЛА знаходяться ближче до наземних користувачів, що підвищує ефективність розвантаження. В ОМА загальна пропускна

здатність розділена на різні підканали, один користувач займає кожен підканал, а для кожного користувача виділяються однакові часові ресурси. ОМА забезпечує недостатню та невідповідну продуктивність системи. Це менш складно, використовуючи прості приймачі в системі, і в ідеалі немає взаємних перешкод між кількома користувачами.

II) Схема неортогонального множинного доступу (NOMA): кожен користувач мобільного зв'язку може одночасно використовувати неортогональні ресурси для розвантаження завдань у висхідній лінії зв'язку, а БПЛА надсилає остаточний результат у низхідній лінії зв'язку. У NOMA весь діапазон частот або часові інтервали спільно використовуються між кількома користувачами мобільного зв'язку. Він використовує методи послідовного придушення перешкод (SIC) на стороні приймача, що ускладнює його. Незважаючи на це, він більш ефективний, ніж ОМА, оскільки забезпечує високу спектральну ефективність і меншу затримку. NOMA широко використовується в мережах U-MEC через високе використання спектра порівняно з ОМА [19].

Схема доступу «один за одним»

У цій схемі БПЛА може підключитися лише до одного мобільного користувача в будь-який час. Він кращий за ОМА, тому що щонайбільше один мобільний пристрій може спілкуватися з пов'язаним з ним БПЛА протягом кожного часового інтервалу. Мною розглянуто застосування механізму планування доступу один за одним, що робить систему більш ефективною та простішою, ніж ОМА [20].

Дуплексні схеми

Прямий канал зв'язку: канал зв'язку від U-MEC до мобільного користувача називається низхідним каналом зв'язку (DL або D/L).

Зворотний канал зв'язку: канал зв'язку від мобільного користувача до U-MEC називається висхідним каналом зв'язку (UL або U/L).

Повний дуплекс або просто дуплекс — це двонаправлена схема, в якій передача може відбуватися в обох напрямках (UL та DL), тобто в U-MEC процес розвантаження та завантаження результату відбувається одночасно. Крім того,

перешкоди між процесом розвантаження та завантаження ефективно контролюються. Дуплексні схеми класифікуються на два типи: FDD і TDD [21].

Дуплекс із частотним поділом даних (FDD): у FDD розвантаження та завантаження здійснюються одночасно за допомогою двох каналів зв'язку для UL та DL. Він використовує великий спектр, тобто вдвічі, ніж TDD, і має більше перешкод. Зазвичай він використовується у сценаріях міжміських зв'язків між U-MEC та мобільними користувачами.

Дуплекс із тимчасовим поділом (TDD): у TDD канали висхідної та низхідної лінії зв'язку використовують один і той самий частотний діапазон, розподіляючи чергування часових інтервалів для розвантажування та завантаження. Він швидший, ніж FDD, і використовує один канал спектру з меншими перешкодами. Він використовується в сценаріях на короткі відстані [22].

Обидві схеми мають свої переваги та недоліки. Залежно від застосування та використання, вони застосовуються в різних сферах, щоб отримати найбільшу користь.

2.4 Підходи до оптимізації

Мною були досліджені і розглянуті різні методи оптимізації для забезпечення ефективних параметрів QoS і ресурсів MEC для кінцевих користувачів. Ці параметри надають різні ресурси на основі середовища MEC.

2.4.1 Оптимізація розміщення послуг

У середовищах MEC найважливіше, про що слід подумати, – це розміщення служб, оскільки краще розміщення служб означає меншу затримку, кращу продуктивність мережі та менше втраченої енергії. Дослідники зосередилися на головній проблемі MEC, яка полягає у збільшенні обмежень QoS. Для виконання поставленої задачі автор скоротив час виконання та застосував циклічний алгоритм планування.

Результат підходу забезпечив високу швидкість виконання робіт в обумовлений проміжок часу. Вони вирішували питання МЕС у хмарному середовищі. Щоб подолати цю проблему, автори запропонували покращений алгоритм клонального відбору, який використовується для вибору найкращого ресурсу та розподілу ресурсів на запити користувачів з мінімальним часом. Однак цей алгоритм збільшує робочий діапазон, час відгуку та вартість.

Обговорення селективних алгоритмів для розподілу розвантаження даних для архітектури МЕС, де розвантаження даних розподіляється на вимогу, мінімально-мінімально, мінімально-максимально та евристичні алгоритми інтегровані у вибіркові алгоритми. Усі три з трьох наведених вище алгоритмів забезпечують мінімізований проміжок часу на розподіл, і це добре працює лише для незалежних завдань. Загальна продуктивність системи покращується, а використання системи збільшується.

Пізніше розробляється та реалізується ієрархічно структурований МЕС з використанням методів управління знаннями, які використовуються для зменшення вартості порушень узгодження рівня затримки, покращення продуктивності, зменшення споживання енергії та розподілу ресурсів. Реконфігурація віртуальної машини застосовується кожного разу, коли МЕС, і це займає більше часу та зменшує доступність ресурсів.

Також дослідники запропонували алгоритм планування, який є найбільш ефективним для МЕС у промисловому Інтернеті речей (*IIoT*). Процес планування зосереджується на розподілі часу відповідно до розміру роботи та зменшує споживання енергії. Вони в основному зосереджені на QoS з точки зору енергоефективності та ефективного МЕС. Головним недоліком цієї роботи є те, що компроміс між енергоефективністю та затримкою не досягається ідеально. Удосконалений алгоритм, який залежить від програмно-визначеної мережі (*SDN*), був представлений для підтримки зв'язку між різними показниками QoS МЕС.

Автори проаналізували переваги периферійних обчислень із множинним доступом (*MAEC*) над звичайними МЕС, а потім найкращим чином використали *MAEC* для вирішення проблеми балансування навантаження, що виникає в

хмарних обчисленнях, розробивши модель балансування навантаження після ресурсу віртуалізації. управління. МАЕС є різновидом модифікованого гібрида МЕЦ; виконання демонструє значно кращий час конвергенції та результати оптимізації порівняно зі звичайним МЕС, але не змогло забезпечити кращу затримку в Mobile Edge Hosts (MEH). Щоб вирішити цю проблему представили віртуалізацію мережевих функцій (NFV) для середовища МЕС. Оскільки гібридні хмари використовуються для все більшої кількості великомасштабних програм, служба керування робочим процесом повинна мати можливість ефективно планувати та керувати виконанням програм. Ця робота має продовжувати зосереджуватися на вартості, розмірі діапазону, справедливості, ефективності на системному рівні та інших подібних аспектах, пов'язаних з алгоритмами.

Усі ці параметри розглядаються в матеріалах, де реалізовано спільний підхід до балансування навантаження та де автори, перш за все, розробили проблему планування ресурсів. Після цього вводиться політика планування ресурсів, щоб досягти часу виконання завдання, оптимального використання ресурсів, середнього споживання енергії та середньої вартості. Але цей SDN страждає від атак розподіленої відмови в обслуговуванні (DDoS). Нещодавно для вирішення цих проблем безпеки представили стабільну та безпечну схему балансування навантаження для МЕС.

2.4.2 Оптимізація для вибору мережі

Правильний вибір мережі є важливим чинником для забезпечення найкращого розміщення користувачів, зменшення кількості віртуальних шляхів у мережі та покращення охоплення кінцевих користувачів. Для вирішення цих завдань було запропоновано низку підходів і алгоритмів, які спрямовані на вдосконалення процесів розподілу ресурсів, планування та обслуговування у мережах МЕС.

Один із підходів – концепція динамічного нарізання мережі, яка поєднує управління хмарними завданнями та постачальниками послуг для ефективного розподілу ресурсів. Динамічний контролер ресурсів аналізує трафік і виконує завдання

оптимізації управління. Для розв'язання задачі нарізання ресурсів також були запропоновані мережі мікро-операторів, які спрощують процес інтеграції МЕС у хмарну інфраструктуру.

У МЕС-системах також реалізується ймовірнісний підхід для вибору мережі, що забезпечує ефективний розподіл завдань і зменшення затримок. В іншій роботі запропоновано схему управління ресурсами з урахуванням QoS, яка сприяє підвищенню використання серверів та пропускної здатності мережі, забезпечуючи якість обслуговування для кожної програми.

Додатково розроблені методології керування робочим навантаженням із використанням архітектури малих комірок. У цьому підході завдання розподіляються між найближчими комірками з урахуванням QoS, що дає змогу покращити управління хмарними ресурсами та мінімізувати затримку. Проте недоліком цього методу є відсутність врахування помилок виконання завдань, що може вплинути на точність обробки.

Для підвищення продуктивності було запропоновано алгоритм безперервного навчання підтримки (*SSL*), який виконує спільний вибір мережі з оптимізацією затримки. Цей підхід забезпечує швидку та стійку конвергенцію, а також задоволення обмежень QoS.

Інший механізм фокусується на динамічному розвантаженні завдань і плануванні в оптимальних мережах. Він використовує гнучке середовище, що дає змогу динамічно враховувати споживання енергії, рівні навантаження на мережу, час виконання завдань тощо.

Дослідники також приділили увагу введенню живучих функцій віртуальних мереж (*VNF*), зменшуючи витрати на операції через оптимізацію розташування хмарних вузлів і зіставлення віртуальних з'єднань. Для енергоефективної кластеризації у *VNF* було запропоновано схему, яка покращує обробку ресурсів і розподіл завдань між серверами. У разі невдачі алгоритм активує додаткові сервери, оптимізуючи навантаження та враховуючи залишкову пропускну здатність.

Нарешті, для прогнозування навантаження у МЕС-системах запропоновані нові підходи, що дозволяють враховувати динаміку запитів. Це забезпечує

ефективне управління ресурсами, навіть за умов значної варіативності шаблонів запитів, підвищуючи загальну продуктивність мережі.

Ці підходи сприяють вирішенню проблем оптимізації в МЕС-системах, забезпечуючи баланс між якістю обслуговування, енергоспоживанням і продуктивністю.

2.4.3 Оптимізації затримки в МЕС-системах

Затримка є однією з ключових проблем в МЕС, оскільки зі збільшенням кількості віртуальних мережевих функцій вона пропорційно зростає. Для її мінімізації розроблено кілька ефективних підходів.

Одним із таких підходів є алгоритм низької затримки (*uRLLC*), спрямований на зменшення загальної затримки виконання завдань при дотриманні обмежень. Він базується на адаптивному постачанні ресурсів і пропонує хмарні сервіси, що підлаштовуються під змінні умови обслуговування. Проте метод стикається із проблемою динамічного розподілу ресурсів, що може спричиняти збільшення затримки.

Для оптимізації динамічних ресурсів запропоновано механізм, який дозволяє розподіляти ресурси між користувачами в чергах відповідно до принципів справедливості. Це сприяє кращому використанню ресурсів, але водночас значно ускладнює обчислення. Щоб вирішити цю проблему, розроблено МЕС на основі NFVO з низькою затримкою, де планування запитів здійснюється через відповідні ресурси, що дозволяє зменшити затримки в обслуговуванні.

Ще один підхід базується на використанні менеджера віртуальної інфраструктури (*VIM*), який оцінює ефективність алгоритмів розміщення VNF. Ця система враховує особливості попиту на ресурси та кількість доступних вузлів, але стикається з обмеженнями в оптимізації розміщення VNF із одночасним зменшенням затримок. Для цього була розроблена модель, яка дозволяє виконувати спільну оптимізацію розташування і пом'якшення затримки.

Для покращення продуктивності мережі було представлено техніку динамічного надання ресурсів, яка створює коаліції віртуальних шляхів у хмарі. Це допомагає зменшити перекриття віртуальних шляхів, що виникає через зміни в мережі, та оптимізувати їх. Додатково було запропоновано генетичний алгоритм, який обирає найкращі шляхи та уникає їхнього дублювання, що зменшує затримку, пов'язану з переповненням даних у VNF. Однак цей метод має обмеження, пов'язані з недостатньою масштабістю через невеликий розмір популяції.

Для управління ресурсами з низькою затримкою розроблено механізми, що враховують різноманітні потреби користувачів. Вони дозволяють справедливо розподіляти ресурси, запобігаючи нерівномірному використанню, та підтримувати високу якість обслуговування.

Також представлено алгоритми оптимізації затримки, що використовують планування завдань із урахуванням QoS. Ці підходи дозволяють швидко та масштабовано реагувати на запити, ефективно підтримуючи баланс між QoE та QoS, що забезпечує стабільну продуктивність і мінімізацію затримки.

3 МОБІЛЬНІ ПЕРИФЕРІЙНІ ОБЧИСЛЕННЯ З ПІДТРИМКОЮ КІЛЬКОХ БЕЗПІЛОТНИКІВ ІЗ БАЛАНСУВАННЯМ НАВАНТАЖЕННЯ ДЛЯ МЕРЕЖ IoT

Безпілотні літальні апарати (БПЛА) широко використовуються для забезпечення розширеного інформаційного покриття, а також послуг ретрансляції для наземних мереж Інтернету речей (IoT). Враховуючи суттєво обмежені можливості обробки, пристрої IoT можуть не впоратися з важкими обчислювальними завданнями.

У цьому розділі показано систему периферійних мобільних обчислень (МЕС) із підтримкою кількох БПЛА, де дрони діють як вузли МЕС, щоб надавати послуги розвантаження обчислень для наземних вузлів Інтернету речей, які мають обмежені локальні обчислювальні можливості. Заради балансування навантаження для БПЛА запропоновано механізм розгортання на основі диференціальної еволюції (*DE*) [1], де ми моделюємо проблему доступу як узагальнену задачу призначення (*GAP*), яка потім розв'язується майже оптимальним алгоритмом рішення. Виходячи з цього, ми можемо досягти балансу навантаження цих дронів, гарантуючи обмеження покриття та задовільну якість обслуговування (*QoS*) вузлів IoT [23].

Розроблений алгоритм глибокого підкріпленого навчання (*DRL*) для планування завдань у певному БПЛА, що підвищує ефективність виконання завдань у кожному дроні.

Результати моделювання показують здійсненність і перевагу запропонованої схеми розгортання БПЛА, орієнтованої на балансування навантаження, а також алгоритму планування завдань.

3.1 Модель мережі

Модель системи МЕС проілюстрована на рисунку 3.1, складається з K вузлів IoT і N БПЛА. Тим часом вузли IoT на землі можуть перекласти свої завдання на БПЛА.

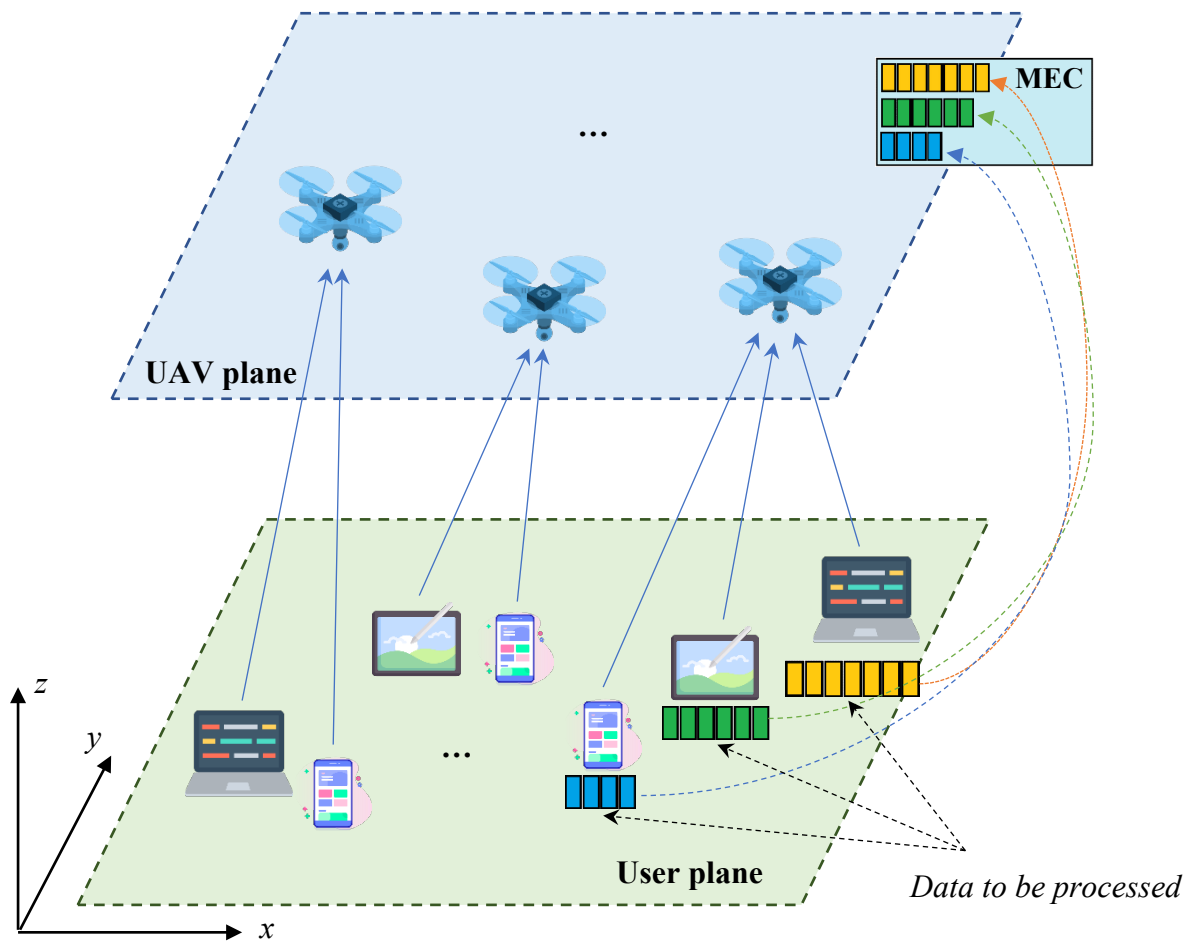


Рис. 3.1 Архітектура мережі

Нехай $\psi = \{m_1, m_2, \dots, m_k\}$ позначають набір вузлів IoT і $\eta = \{u_1, u_2, \dots, u_N\}$ буде набором БПЛА. У даній системі вузли IoT складаються з різних типів пристроїв з різними рівнями розвантаження. Припускаємо, що існує Z типів вузлів IoT, представлених $e = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_Z\}$, де відповідний рівень розвантаження позначається $\tilde{e} = \{\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \dots, \tilde{\varepsilon}_Z\}$.

Без втрати загальності ми припускаємо, що всі вузли IoT закріплені на землі, а всі БПЛА літають на фіксованій висоті $H (H > 0)$. Положення k -го вузла IoT позначається як $m_k = [x_k^m, y_k^m, 0]$, а положення n -го БПЛА позначається як $u_n = [x_n^u, y_n^u, H]$.

Нехай $\alpha_{n,k}$ вказує на зв'язок між БПЛА та вузлом IoT, де $\alpha_{n,k} = 1$ означає, що k -й вузол IoT підключений до n -го БПЛА. Враховуючи, що кожен вузол IoT може підключатися лише до одного БПЛА одночасно, ми маємо:

$$\sum_{n=1}^N \alpha_{n,k} = 1, \forall k = 1, 2, \dots, K, \quad (3.1)$$

Відстань між u_n та m_k буде обчислюватись за формулою $d(u_n, m_k) = [H^2 + (x_n^u - x_k^m)^2 + (y_n^u - y_k^m)^2]^{\frac{1}{2}}$. Для розвантаженого завдання $F_z(u_n, m_k)$, $\forall z = 1, 2, \dots, Z$, між m_k і u_n , враховуючи необхідний обчислювальний $c_z(u_n, m_k)$, необхідний час виконання $t_z(u_n, m_k)$ і необхідний трафік зв'язку $h_z(u_n, m_k)$, визначимо $F_z(u_n, m_k) = [c_z(u_n, m_k), t_z(u_n, m_k), h_z(u_n, m_k)]$.

3.2. Комунікаційна модель

На рисунку 3.2 показано сценарій IoT з підтримкою кількох БПЛА. Кожен БПЛА має фіксоване покриття з радіусом R_c . Крім того, наземні вузли IoT можуть отримати доступ до БПЛА через мультиплексний доступ з ортогональним частотним поділом каналів, отже, перешкоди між зв'язками вузла БПЛА нехтуються.

Припускаємо, що в каналах зв'язку від БПЛА до вузлів IoT домінує канал LOS, і, отже, посилення каналу між m_k і u_n можна отримати як:

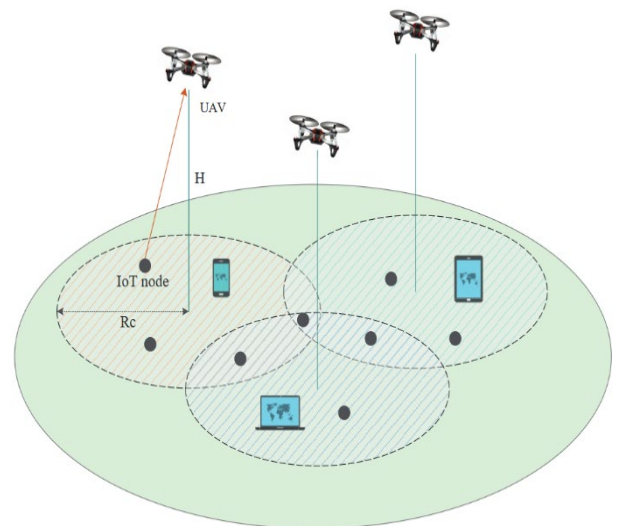


Рис. 3.2 IoT з підтримкою мульти-БПЛА

$$g(u_n, m_k) = \beta_0 d(u_n, m_k)^{-2} = \frac{\beta_0}{H^2 + (x_n^u - x_k^m)^2 + (y_n^u - y_k^m)^2}, \quad (3.2)$$

де β_0 позначає посилення каналу на еталонній відстані 1 метр. Нехай $\tilde{p}(u_n)$ представляє потужність передачі u_n . Тоді швидкість передачі даних висхідної лінії зв'язку між u_n та m_k може бути сформульована за допомогою

$$\begin{aligned} \tilde{v}(u_n, m_k) &= B \log_2 \left(1 + \frac{\tilde{p}(u_n) g(u_n, m_k)}{\sigma^2} \right) \\ &= B \log_2 \left(1 + \frac{\beta_0 \tilde{p}(u_n)}{\sigma^2 (H^2 + (x_n^u - x_k^m)^2 + (y_n^u - y_k^m)^2)} \right), \end{aligned} \quad (3.3)$$

де σ^2 позначає дисперсію білого Гаусового шуму, а B позначає смугу каналу.

Затримку передачі між m_k та u_n для завдання $F_z(u_n, m_k)$ можна показати як

$$t_z^T(u_n, m_k) = \frac{h_z(u_n, m_k)}{\tilde{v}(u_n, m_k)}, \quad (3.4)$$

Нехай $\phi(u_n)$ представляє набір вузлів IoT, підключених до u_n :

$$\bigcup_{n=1}^N \phi(u_n) = \psi, \quad (3.5)$$

Тоді загальний рівень розвантаження u_n можна позначити як:

$$\Omega_n = \sum_{k=1}^K \tilde{\varepsilon}(m_k) \alpha_{n,k}, \forall n = 1, 2, \dots, N, \quad (3.6)$$

де $\tilde{\varepsilon}(m_k)$ представляє рівень розвантаження m_k та $\tilde{\varepsilon}(m_k) \in \tilde{\varepsilon}$.

3.3 Обчислювальна модель

У системі йдеться про P типів ресурсів у БПЛА, таких як ЦП, пам'ять, жорсткий диск тощо. На рисунку 3.3 показано приклад планування завдань. Для завдання $F_z(u_n, m_k)$, ідеальний час завершення є $t_z(u_n, m_k)$, тому ми обчислюємо загальний час виконання завдання $F_z(u_n, m_k)$ відповідно до

$$t_z^A(u_n, m_k) = t_z(u_n, m_k) + t_z^{\text{delay}}(u_n, m_k), \quad (3.7)$$

де $t_z^{\text{delay}}(u_n, m_k)$ позначає час затримки завдання $F_z(u_n, m_k)$. Отже, $t_z^{\text{delay}}(u_n, m_k)$ дорівнює 0, коли завдання $F_z(u_n, m_k)$ виконується негайно, або більше 0, коли поточна черга обробки БПЛА заповнена і завдання має почекати.

3.4 Формулювання проблеми

У своїй роботі я вивчаю оптимізацію середнього сповільнення для розвантажених завдань, де баланс навантаження для БПЛА гарантований. Оскільки розподіл вузлів IoT загалом нерівномірний, стратегія розгортання БПЛА матиме суттєвий вплив на продуктивність системи МЕС.

Як правило, БПЛА повинні бути якомога ближче до вузлів IoT, щоб зменшити вартість передачі. Крім того, надмірне навантаження на БПЛА призведе до збільшення кількості завдань у черзі обробки, що збільшить загальний час обробки завдань і ще більше вплине на продуктивність системи.

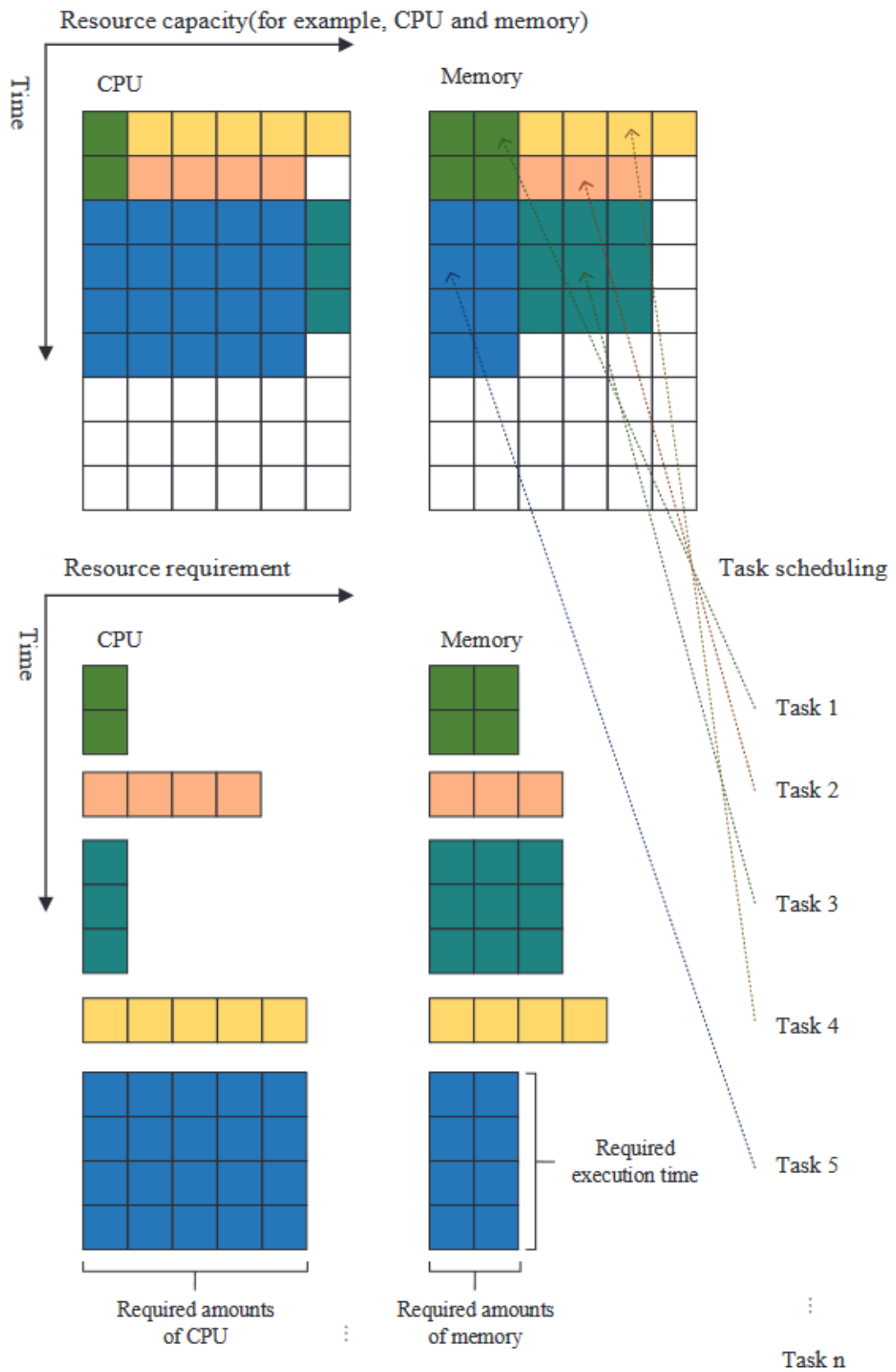


Рис. 3.3 Приклад іграшки для пояснення схеми планування завдань

Таким чином, для системи МЕС з декількома БПЛА необхідно досягти балансу навантаження, щоб уникнути загального зниження продуктивності системи, коли деякі БПЛА виконують занадто багато завдань, а інші не виконують завдань.

Зокрема, проблема стосується двох аспектів, а саме розгортання кількох БПЛА з балансуванням навантаження та планування завдань на БПЛА з урахуванням затримки. Що стосується розгортання кількох БПЛА з балансуванням навантаження, то мета полягає в досягненні навантаження для всіх БПЛА, де ступінь балансу навантаження можна виразити як

$$S_o(\Omega) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Omega_i - \bar{\Omega})^2}, \quad (3.8)$$

де

$$\bar{\Omega} = \frac{1}{N} \sum_{i=1}^N (\Omega_i), \quad (3.9)$$

Зверніть увагу, що $S_o(\Omega) \geq 0$. Отже, чим більше $S_o(\Omega)$, тим більш нерівномірним є розподіл завдань між БПЛА. У той же час враховується середня вартість передачі між БПЛА та вузлами Інтернету речей, і прагнення мінімізувати середню вартість передачі відповідно до формули (3.4), яке можна розрахувати за допомогою:

$$t_c(u_n, m_k) = \frac{1}{K} \sum_{k=1}^K \sum_{n=1}^N t_z^T(u_n, m_k) \alpha_{n,k}, \quad (3.10)$$

Що стосується планування завдань з урахуванням затримки, відповідно до формули (3.7), ми можемо отримати уповільнення завдання, яке можна розрахувати:

$$t_d(u_n, m_k) = \frac{t_z^A(u_n, m_k)}{t_z(u_n, m_k)}, \quad (3.11)$$

де $t_d(u_n, m_k) \geq 1$.

Тоді, згідно з формулами (3.8), (3.10) і (3.11), розгортання кількох БПЛА з балансуванням навантаження, а також задачу оптимізації планування завдань з урахуванням затримки можна сформулювати як

$$\begin{aligned} P1: \min_{(x_n^*, y_n^*)} & \frac{1}{K} \sum_{k=1}^K \sum_{n=1}^N t_d(u_n, m_k) \alpha_{n,k} \\ & + \ell_1 [S_O(\Omega)] + \ell_2 [t_c(u_n, m_k)] \\ \text{s.t. } & d(u_n, m_k) \leq R_c, \\ & m_k \in \psi, \forall k = 1, 2, \dots, K, \\ & u_n \in \eta, \forall n = 1, 2, \dots, N, \\ & \alpha_{n,k} \in \{0, 1\}, \forall k = 1, 2, \dots, K, n = 1, 2, \dots, N, \\ & \sum_{n=1}^N \alpha_{n,k} = 1, \forall k = 1, 2, \dots, K, \end{aligned} \quad (3.12)$$

де ℓ_1 and ℓ_2 позначають вагові коефіцієнти для вимог балансування навантаження та середньої вартості передачі відповідно.

Зокрема, $\ell_1 = 0$ означає, що вплив балансу навантаження для БПЛА не враховується, а $\ell_2 = 0$ означає, що вплив середньої вартості передачі не враховується.

3.5 Спільне розгортання БПЛА та оптимізація планування завдань

Мною було детально вивчено і оптимізовано алгоритм розгортання кількох БПЛА та схему планування завдань. Проблема розгортання кількох БПЛА є доволі складною. Щоб знайти задовільний розв'язок, розглянемо алгоритм розгортання кількох БПЛА з підтримкою диференціальної еволюції для вирішення цієї проблеми [24]. Спочатку всі БПЛА розташовуються випадковим чином, а потім ми

призначаємо максимальне навантаження для кожного БПЛА на основі його розташування та вузлів Інтернету речей. Таким чином проблему доступу між дронами та вузлами IoT можна змодельовати в узагальнену проблему призначення. Після цього використовуємо алгоритм наближення, щоб визначити всі зв'язки. У той же час для планування завдань для вхідних завдань використовується алгоритм глибокого підкріплення. Нарешті, майже оптимальні положення БПЛА отримують за допомогою ітерацій алгоритму диференціальної еволюції.\

3.6 Баланс навантаження для БПЛА

Оскільки існує загальна Z типів вузлів IoT, представлених $e = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_Z\}$, а відповідний рівень розвантаження становить $\tilde{e} = \{\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \dots, \tilde{\varepsilon}_Z\}$, загальне розвантаження рівень E_t можна сформулювати як

$$E_t = \sum_{k=1}^K \tilde{\varepsilon}(m_k), \quad (3.13)$$

де $\tilde{\varepsilon}(m_k)$ представляє рівень розвантаження m_k .

В ідеалі кожен БПЛА несе навантаження $E_{avg} = E_t/N$ для досягнення балансу навантаження. Однак, оскільки розповсюдження IoT вузлів, як правило, нерівномірне - балансування навантаження для всіх БПЛА призведе до надмірного споживання передачі даних, коли деякі вузли знаходяться далеко від підключеного БПЛА, таким чином не вдасться досягти мети оптимізації формули (3.12). Отже, відповідно до розташування всіх БПЛА та вузлів IoT, нехай $e_{lb}(u_n)$ буде нижньою межею набору вузлів IoT, у якому вузли IoT можуть бути підключені лише до u_n через відстань. Загальний рівень розвантаження $e_{lb}(u_n)$ можна сформулювати як

$$\tilde{e}_{lb}(u_n) = \sum_{m_k \in e_{lb}(u_n)} \tilde{\varepsilon}(m_k), \forall k = 1, 2, \dots, K, \quad (3.14)$$

Подібним чином, нехай $e_{ub}(u_n)$ буде верхньою межею набору вузлів IoT, які може охопити u_n . Загальний рівень розвантаження $e_{ub}(u_n)$ можна сформулювати як

$$\tilde{e}_{ub}(u_n) = \sum_{m_k \in e_{ub}(u_n)} \tilde{\varepsilon}(m_k), \forall k = 1, 2, \dots, K, \quad (3.15)$$

Тоді нехай опорне навантаження u_n буде $\tilde{E}(u_n)$ ми можемо досягти:

$$\tilde{e}_{lb}(u_n) \leq \tilde{E}(u_n) \leq \tilde{e}_{ub}(u_n), \quad (3.16)$$

Враховуючи E_{avg} , $\tilde{e}_{lb}(u_n)$ і $\tilde{e}_{ub}(u_n)$, якщо $E_{avg} \geq \tilde{e}_{ub}(u_n)$, найкращим параметром для $\tilde{E}(u_n) \in \tilde{E}(u_n) = \tilde{e}_{ub}(u_n)$ для задоволення вимоги щодо загального навантаження u_n . Подібним чином, якщо $E_{avg} \leq \tilde{e}_{lb}(u_n)$, найкраще налаштування для $\tilde{E}(u_n) \in \tilde{E}(u_n) = \tilde{e}_{lb}(u_n)$. У випадку $\tilde{e}_{lb}(u_n) \leq E_{avg} \leq \tilde{e}_{ub}(u_n)$, u_n має більше ексклюзивних вузлів IoT, коли E_{avg} близьке до $\tilde{e}_{lb}$, де $\tilde{E}(u_n)$ потрібно відповідно збільшення. Коли E_{avg} наближається до $\tilde{e}_{ub}$, більше вузлів IoT, охоплених u_n , також будуть охоплені іншими БПЛА, де $\tilde{E}(u_n)$ потрібно відповідно зменшити. Отже, ми можемо отримати:

$$\begin{aligned} \tilde{E}(u_n) &= E_{avg} + \lambda_1(\tilde{e}_{ub}(u_n) - E_{avg}) \\ &\quad - \lambda_2(E_{avg} - \tilde{e}_{lb}(u_n)) \\ &= (1 - \lambda_1 - \lambda_2)E_{avg} + \lambda_1\tilde{e}_{ub}(u_n) + \lambda_2\tilde{e}_{lb}(u_n) \end{aligned}, \quad (3.17)$$

де λ_1 і λ_2 вказують на фактори впливу $\tilde{e}_{ub}(u_n)$ та $\tilde{e}_{lb}(u_n)$ відповідно. Тоді опорне навантаження $\tilde{E}(u_n)$ можна виразити за допомогою

$$\tilde{E}(u_n) = \begin{cases} \tilde{e}_{lb}, & E_{avg} \leq \tilde{e}_{lb}(u_n), \\ \tilde{e}_{ub}, & E_{avg} \geq \tilde{e}_{ub}(u_n), \\ (1 - \lambda_1 - \lambda_2)E_{avg} + \lambda_1\tilde{e}_{ub}(u_n) + \lambda_2\tilde{e}_{lb}(u_n), & \end{cases} \quad (3.18)$$

Тоді суму всього базового навантаження можна виразити як $\tilde{E}_t = \sum_{i=1}^N \tilde{E}(u_n)$. Іноді може бути, що між $\tilde{E}_t$ і E_t існує проміжок у випадку $\tilde{E}_t < E_t$, який потрібно заповнити. Отже, у випадку $\tilde{E}_{lb}(u_n) \leq E_{avg} \leq \tilde{E}_{ub}(u_n)$, ми заповнюємо прогалину, додаючи $\tilde{E}_{gap}$ до $\tilde{E}(u_n)$, а в іншому випадку опорне навантаження залишається незмінним. Тоді ми маємо:

$$\tilde{E}_{gap} = \frac{E_t - \tilde{E}_t}{n^*}, \quad (3.19)$$

де n^* – це кількість БПЛА, що у випадку $\tilde{E}_{lb}(u_n) < E_{avg} < \tilde{E}_{ub}(u_n)$. Нарешті, ми переписуємо формулу (3.18), і отримуємо кінцеве еталонне навантаження всіх БПЛА:

$$\tilde{E}(u_n) = \begin{cases} \tilde{e}_{lb}, & E_{avg} \leq \tilde{e}_{lb}(u_n), \\ \tilde{e}_{ub}, & E_{avg} \geq \tilde{e}_{ub}(u_n), \\ (1 - \lambda_1 - \lambda_2)E_{avg} + \lambda_1\tilde{e}_{ub}(u_n) + \lambda_2\tilde{e}_{lb}(u_n) + \tilde{E}_{gap}, & \end{cases} \quad (3.20)$$

3.7 Призначення вузла на основі GAP

У цьому підрозділі було розглянуто оптимізацію розподілу вузлів між БПЛА відповідно до контрольного навантаження в попередньому підрозділі. Враховуючи відоме положення всіх БПЛА, проблему призначення вузлів можна сформулювати як різновид узагальненої задачі призначення. У GAP як завдання, так і агенти мають розміри, де розмір кожного завдання відрізняється для різних агентів [25].

У цій роботі кожен БПЛА можна розглядати як агента, а кожен вузол IoT має певні завдання. Визначити $w_{n,k}$ як вартість завдання, вивантаженого з k -го вузла IoT на n -й БПЛА, де $w_{n,k}$ визначається лише типом завдання. У результаті $w_{n,k} = \tilde{\epsilon}(m_k)$. Крім того, нехай $f_{n,k}$ буде прибутком від завдання, на який головним чином впливає відстань між u_n і m_k і необхідний трафік зв'язку $h_z(u_n, m_k)$. Відповідно до формули (3.4), ми можемо отримати

$$f_{n,k} = \begin{cases} \varphi t_z^T(u_n, m_k), & d(u_n, m_k) \leq R_c, \\ -\infty, & d(u_n, m_k) > R_c, \end{cases}, \quad (3.21)$$

де $\varphi > 0$ позначає фактор впливу. Загалом, чим ближча відстань між u_n і m_k , тим більшим є $f_{n,k}$. Коли відстань між u_n і m_k перевищує радіус обслуговування R_c , $f_{n,k}$ наближається до $-\infty$.

Для того, щоб максимізувати загальний прибуток, сформулюємо цю задачу розподілу як задачу цілочисельного програмування (IP), яку можна записати як:

$$\begin{aligned} P2: \min_{x_{n,k}} & \sum_{n=1}^N \sum_{k=1}^K -f_{n,k} x_{n,k} \\ \text{s.t.} & \sum_{k=1}^K w_{n,k} x_{n,k} \leq \tilde{E}(u_n), \forall n = 1, 2, \dots, N, \\ & \sum_{n=1}^N x_{n,k} = 1, \forall k = 1, 2, \dots, K, \\ & x_{n,k} \in \{0, 1\}, \forall n = 1, 2, \dots, N, k = 1, 2, \dots, K, \end{aligned} \quad (3.22)$$

Оскільки задача P2 є GAP, то проблема P2 є складною. Тут пропонується наближений алгоритм для знаходження майже оптимального рішення, яке описується у три етапи.

- *Етап 1: релаксація лінійного програмування*

Перетворіть задачу цілочисельного програмування в наведену нижче релаксацію лінійного програмування (LP).

$$\begin{aligned}
 P3: \min_{x_{n,k}} & \sum_{n=1}^N \sum_{k=1}^K -f_{n,k} x_{n,k} \\
 \text{s.t.} & \sum_{k=1}^K w_{n,k} x_{n,k} \leq \tilde{E}(u_n), \forall n = 1, 2, \dots, N, \\
 & \sum_{n=1}^N x_{n,k} = 1, \forall k = 1, 2, \dots, K \\
 & x_{n,k} \geq 0, \forall n = 1, 2, \dots, N, k = 1, 2, \dots, K.
 \end{aligned} \tag{3.23}$$

Тоді, розв'язуючи цю формулу, ми отримуємо дробове рішення.

➤ *Етап 2: Побудова дводольного графа*

Потрібно перетворити дробове рішення на дводольний граф із дробовими розмірами ребер, де одна сторона представляє вузли IoT, а інша — слоти в БПЛА.

Визначаємо дводольний граф $C = (\hat{K}, \hat{B})$, де $\hat{K} = [K]$ і $\hat{B} = \{(n', s') \mid n' \in [N], s' \in [s'_n]\}$. Ми проілюструємо ребра у дводольному графі лише на випадку $n' = 1$. Для $n' = 1$ розглянемо лише вузол IoT m_k так, що $x_{1,k} > 0$, і припустимо, що це $[K]$ без втрати загальності. Припустимо, що $f_{1,1} \geq f_{1,2} \geq \dots f_{1,K}$ і відсортуємо їх, якщо необхідно. Для кожного $k' \in [K]$ ми відстежуємо часткову суму $\sum_{k=1}^{k'} x_{1,k}$. Якщо $\sum_{k=1}^{k'} x_{1,k} \leq s'$ і $\sum_{k=1}^{k'-1} x_{1,k} \geq s' - 1$, потім додаємо ребро $(k', (1, s'))$ з значенням $x_{1,k'}$. В іншому випадку, якщо $\sum_{k=1}^{k'} x_{1,k} > s'$, і $\sum_{k=1}^{k'-1} x_{1,k} < s'$, потім додаємо ребро $(k', (1, s' - 1))$ із значенням $s' - \sum_{k=1}^{k'-1} x_{1,k}$ і ребро $(k', (1, s'))$ зі значенням $\sum_{k=1}^{k'} x_{1,k} - s'$.

➤ *Етап 3: детерміноване округлення*

Зауважте, що дводольний графік — це дробове ідеальне збіг базової функції витрат $c(k', (n', s')) = -f_{n,k}$. Можна показати, що всі крайні точки багатогранника є цілими. Шляхом знаходження циклу або шляху збільшення в попередньому

дводольному графі ми можемо перетворити дробове рішення в інтегральне за поліноміальний час, не збільшуючи цільове значення. Усі ребра з вагою в діапазоні $(0,1)$ називають *ненасиченими* та $\{0,1\}$ *насиченими*. Ми представляємо ітераційний метод таким чином, що після одного циклу кількість *ненасичених* ребер зменшується щонайменше на 1, а сума вагових коефіцієнтів, прикріплених до лівої вершини, зберігається як 1. Оскільки в кінці не залишилося *ненасиченого* ребра, а кожна ліва вершина приєднана до ребер із ваговими коефіцієнтами, що дорівнюють 1, кожна ліва вершина відображається на праву через унікальне ребро з вагою 1. Незабаром ми побачимо, що ця карта також є ін'єктивною, доводячи, що вона є ідеальною відповідністю з лівої сторони. Розглянемо наступні два випадки:

Випадок 1: якщо є цикл *ненасичених* країв, розфарбуємо їх по черзі синім і червоним кольором. Збільшуючи ваги всіх червоних ребер на невелику величину δ і зменшуючи ваги синіх на δ , жодні обмеження LP не порушуються за умови, що δ є достатньо малим. Таким чином, ми можемо переміщати δ у цьому напрямку, доки одне з червоних країв не дійде до 1 або одне з синіх країв не дійде до 0. У будь-якому випадку зменшили кількість *ненасичених* ребер принаймні на 1.

Випадок 2: Називаємо вершину *одиначною*, якщо вона приєднана лише до *ненасиченого* ребра, тоді вага на вершині легко вважати рівною вазі ребра. Зверніть увагу, що жодна ліва вершина не може бути *одиначною*, оскільки всі вони мають вагу рівно 1. Припустимо, що тепер не залишилося жодного циклу *ненасиченого* ребра, тоді кожен максимальний шлях *ненасиченого* ребра повинен починатися і закінчуватися в одній вершині. Обираємо максимальний шлях і розфарбовуємо край поперемінно синім і червоним кольором. Подібно до випадку 1, зміна ваги всіх синіх і червоних ребер на невелику величину δ не порушує жодних обмежень LP, і ми можемо вільно переміщати δ , доки одне червоне ребро не досягне 1 або одне синє ребро не досягне 0.

Слід зазначити, що після детермінованого округлення фактичний рівень розвантаження кількох БПЛА може бути трохи вищим за еталонне навантаження, але це не спричинить негативних наслідків, оскільки не перевищить максимальну обробну здатність БПЛА.

3.8 Планування завдань із підкріпленням за допомогою навчання

У цьому підрозділі показано схему планування завдань для вхідних завдань на основі алгоритму глибокого навчання з підкріпленням. DRL поєднує глибоке навчання та навчання з підкріпленням, щоб досягти наскрізного навчання від «сприйняття» до «дії». Агент спочатку відчуває стан середовища, а потім виконує дію. Середовище з часом може переходити в інший стан [26]. У той же час агент може отримати зворотний зв'язок за функцією винагороди. Можна побачити, що DRL є моделлю взаємодії людини та середовища. Алгоритм Q-навчання є класичним алгоритмом навчання з підкріпленням, і він може знайти майже оптимальну дію, взаємодіючи з середовищем. Крім того, для кроку t стани можна представити як S_t , а простір політики можна визначити як π , тоді накопичені винагороди $V^\pi(S_t)$ можна обчислити як

$$V^\pi(S_t) = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots, \quad (3.24)$$

де R_t — миттєва винагорода на кроці t , а $\gamma (0 < \gamma < 1)$ — коефіцієнт дисконтування, який відображає вплив майбутньої винагороди. В алгоритмі Q-навчання Q-значення є функцією станів S_t і дії A_t , які можна сформулювати як

$$Q_t(S_t, A_t) = R_t + \gamma V^\pi(S_{t+1}), \quad (3.25)$$

Q-навчання – це свого роду алгоритм з ітерацією значення. Оскільки ми можемо отримати лише обмежену серію зразків і лише обмежені зразки можуть бути використані для роботи, ми оновлюємо значення Q відповідно до

$$Q_{t+1}(S_t, A_t) = (1 - \vartheta)Q_t(S_t, A_t) + \vartheta[R_t + \gamma \max_{A_{t+1}} Q_t(S_{t+1}, A_{t+1})] \quad (3.26)$$

де швидкість навчання ϑ ($0 < \vartheta < 1$) визначає швидкість навчання.

Глибока Q-мережа (DQN) поєднує згорткову нейронну мережу (CNN) із Q-навчанням, яке використовує цільову функцію Q-навчання для створення тегів. Крім того, DQN використовує механізм повторного відтворення досвіду для вирішення проблем кореляції даних. У той же час CNN (MainNet) використовується для генерації поточного Q-значення, а інший CNN (Target) використовується для генерації цільового Q-значення. Функція втрат DQN може бути задана за допомогою

$$L(\theta) = E \left[(Q^T - Q(S_t, A_t; \theta))^2 \right], \quad (3.27)$$

де цільова функція Q^T може бути сформульована як

$$Q^T = R_t + \gamma \max Q(S_{t+1}, A_{t+1}; \theta), \quad (3.28)$$

де θ – параметр нейронної мережі.

Алгоритм 1 *Планування завдань на основі глибокого навчання з підкріпленням*

Input: states, S ; maximum number of iteration, t_{max} ;

Output: action-value function, $Q(S, A)$;

- 1: initialize action-value function Q with random weights;
 - 2: **for** $t=1$; $t < t_{max}$; $t++$ **do**:
 - 3: divide step t into b_t sub-steps;
 - 4: select a random sub-action in each sub-step, and all sub-actions make up action a_t ;
 - 5: execute action a_t , then obtain the reward r_t , and arrive at the next state s_{t+1} ;
 - 6: calculate target Q-value according to Eq. (3.28);
 - 7: update the deep Q-network by minimizing the loss according to Eq. (3.27);
 - 8: **end for**
-

У своїй роботі я використовую алгоритм DRL, що показаний як Алгоритм 1, для вирішення проблеми планування завдань заради мінімізації середнього уповільнення виконання завдань у БПЛА. Враховуючи, що набір $\phi(u_n)$ вузлів IoT отримав доступ до n -го БПЛА згідно з рівнянням (3.5) три ключові чинники глибокої Q-мережі подано наступним чином:

- *Простір станів*: стани ресурсів БПЛА включають поточні виділені обчислювальні ресурси та необхідні обчислювальні ресурси для можливих запланованих завдань, як показано на рисунку 3.3.

Припускаємо, що потреба в ресурсах завдання задано $F_z(u_n, m_k) = [c_z(u_n, m_k), t_z(u_n, m_k), h_z(u_n, m_k)]$,

де $c_z(u_n, m_k) = [c_{z,1}(u_n, m_k), c_{z,2}(u_n, m_k), \dots, c_{z,p}(u_n, m_k)]$ і $c_{z,p}(u_n, m_k), p = 1, 2, \dots, P$ вказує на p -й тип ресурсів.

Крім того, поточні виділені обчислювальні ресурси n -му БПЛА дорівнюють $G_{n,z}$. Отже, стани системи із загальною кількістю N БПЛА можна подати як $G_{n,z} = [S_1^S, S_2^S, \dots, S_n^S, n = 1, 2, \dots, N]$, де $S_n^S = \{\cup_{k=1}^K F_z(u_n, m_k) \alpha_{n,k}, G_{n,z}, z = 1, 2, \dots, Z\}$.

- *Простір дій*: якщо припустити, що на t -му кроці кількість $M_{n,t}$ завдань поставлено в чергу в n -му БПЛА. Тоді розмір простору дії в n -му БПЛА може бути представлений як $2^{M_{n,t}}$, що робить завдання на DRL.

Щоб спростити простір дій, ми визначаємо як $A_{n,t} = \{\emptyset, 1, 2, \dots, M_{n,t}\}$ і ділимо кожен крок на $b_{n,t}$, де $b_{n,t}$ позначає загальну кількість піддій, які будуть заплановані на t -му кроці. Тоді на кожному під-кроку буде заплановано лише одну під-дію розмір простору дії зменшено до $M + 1$.

Оскільки завдання заплановано в кластері, як показано на рисунку 3.3, це завдання буде видалено з черги очікування. Потім агент спостерігає перехід стану, коли заплановане завдання переміщується у відповідну позицію. Коли агент вибирає $\emptyset$ або недійсну під-дію, це вказує на те, що агент не бажає планувати подальші завдання на поточному кроці, і час фактично триває, іншими словами, перейдіть до наступного кроку $t + 1$.

- *Винагорода*: функція винагорода призначена для того, щоб наказати агенту мінімізувати середнє уповільнення БПЛА. Зокрема, ми встановлюємо винагороду на кожному часовому етапі як $\sum_{M_n} \frac{-1}{t_z(u_n, m_k)}$, де M_n — це набір завдань, які зараз виконує n -й БПЛА.

Що стосується складності запропонованого алгоритму планування завдань за допомогою DRL, то в процесі навчання часова складність становить $O(E * D/B * T)$, яка в основному залежить від складності навчання CNN DQN, де E представляє епохи, D позначає розмір набору даних, B позначає розмір партії, а T представляє часову складність однієї ітерації. Більше того, складність часу виконання алгоритму становить $O(n)$, оскільки це в основному матрична операція.

3.9 Розгортання кількох БПЛА на основі диференціальної еволюції

Диференціальна еволюція - це випадкова модель, яка імітує біологічну еволюцію. Через багаторазові ітерації зберігаються популяції, які адаптовані до середовища. У даній роботі розмір кожної популяції становить $2N$, що вдвічі більше, ніж кількість БПЛА.

Нехай $X_d(i)$, $d = 1, 2, \dots, D$ вказує на місце розміщення БПЛА i -го покоління, де D – загальна кількість популяцій.

На початку алгоритму ми випадковим чином визначаємо позицію розгортання БПЛА у всіх сукупностях як $X_d(1)$, $d = 1, 2, \dots, D$. У g -й ітерації кожна популяція $X_d(g)$ мутує та генерує вектор мутації $H_d(g)$ шляхом випадкового вибору трьох інших популяцій із генерації, наприклад, $X_{da}(g)$, $X_{db}(g)$, $X_{dc}(g)$, тоді як $da \neq db \neq dc \neq d$. Тоді маємо

$$H_d(g) = X_{da}(g) + \xi(X_{db}(g) - X_{dc}(g)), \quad (3.29)$$

де ξ вказує на коефіцієнт, що використовується для контролю впливу різних векторів. Тоді кросоверний вектор $V_d(g)$ може бути обчислений за допомогою

$$I_d(g) = \begin{cases} H_d(g), \text{rand}(0,1) < P_{cr}, \\ X_d(g), \text{else} \end{cases}, \quad (3.30)$$

де P_{cr} — ймовірність кросинговеру. Нарешті, відповідно до значення функції відповідності, ми вибираємо найкращу популяцію між кросоверним вектором $V_d(g)$ і вихідним вектором $X_d(g)$ як наступне покоління, яке можна сформулювати як

$$X_d(g+1) = \begin{cases} I_d(g), f(I_d(g)) > f(X_d(g)), \\ X_d(g), \text{else} \end{cases}, \quad (3.31)$$

де функція $f(*)$ є цільовою функцією формули (3.12).

Відповідно до характеристик DE, після певної кількості ітерацій ми можемо отримати конвергентні популяції, з яких можна отримати майже оптимальне розгортання всіх БПЛА.

Щодо складності запропонованого диференціалу еволюційного алгоритму розгортання кількох БПЛА, у кожному етапі прийняття рішень ми маємо оновлювати генерацію кожної сукупності відповідно до формули. (3.31). Отже, складність кожного етапу прийняття рішень має порядок $O(n)$.

Потік запропонованого алгоритму розгортання кількох БПЛА на основі DE наведено в алгоритмі 2.

Алгоритм 2 Розгортання кількох БПЛА на основі DE

Input: the set of (IoT nodes), φ ; objective function; the number of UAVs, N ; maximum number of iteration, l_{max} ;

Output: deployment of UAVs, $[x_n^u, y_n^u]$, $n=1, 2, \dots, N$; best IoT node assignment, $\varphi(u_n)$;

- 1: randomly initialize populations $X_d(1)$;
- 2: **for** $i=1$; $i < l_{max}$; $i++$ **do**:
- 3: obtain mutation populations through differential strategy according to Eq. (3.29);
- 4: obtain crossover populations according to Eq. (3.30);
- 5: determine reference load for multiple UAVs according to Eq. (3.20);
- 6: get the fractional solution of node assignment problem in Eq. (3.23);
- 7: construct the bipartite graph based on the fractional solution;
- 8: deterministic round the fractional solution;
- 9: find the optimal task scheduling scheme according to Algorithm 1;
- 10: calculate the fitness for all populations according to the objective function of Eq. (3.12);
- 11: get the next generation of DE according to Eq. (3.31);
- 12: **end for**

3.10 Результати моделювання

У моделюванні розглядається прямокутна область розміром 400×400 m, де випадково розподілені наземні вузли IoT.

Смуга пропускання каналу становить $B = 1\text{MHz}$, а потужність передачі — $\tilde{p} = 0.5$ W. Дисперсія білого гауссового шуму становить $\sigma^2 = 5 * 10^{-15}$.

Усі БПЛА літають на фіксованій висоті $H = 100$ m з максимальним покриттям $R_c = 100$ m.

Підсилення каналу встановлюється як $\beta_0 = \eta_{\text{los}} \left(\frac{4\pi f}{c} \right)^2$, де $\eta_{\text{los}} = 1$ — коефіцієнт ослаблення, що відповідає LOS, несуча частота — $f = 2.4\text{GHz}$, а c — швидкість світла. Загалом є 100 вузлів IoT і 5 БПЛА.

На рисунку 3.4 показаний приклад майже оптимального результату для розгортання мульти-БПЛА. Існують різні кольори БПЛА, і менші покажчики вказують на вузли IoT, тоді як кожен вузол IoT підключено до БПЛА того ж кольору.

Явно, враховуючи загальну кількість $N=5$ БПЛА та $K=100$ вузлів IoT, отримуємо майже оптимальну стратегію розгортання кількох БПЛА, пов'язану з координатами $[171.41, 126.49]$, $[149.50, 245.87]$, $[270.68, 253.43]$, $[284.26, 152.94]$ та $[227.91, 181.27]$ для кожного дрона.

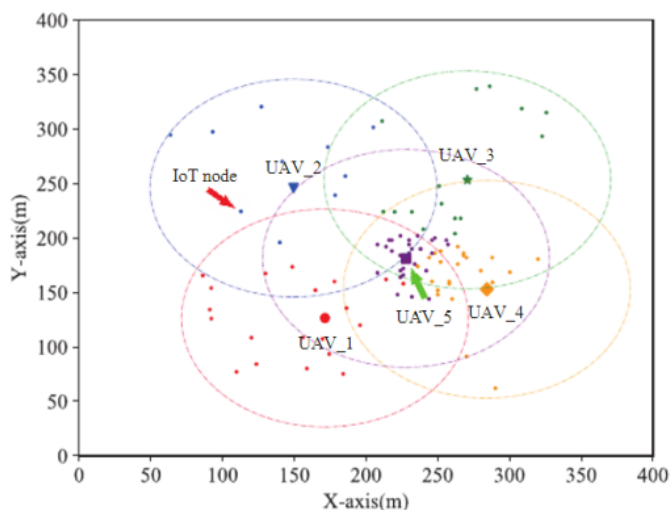


Рис. 3.4 Результат моделювання розгортання кількох БПЛА в контексті $N = 5$ і $K = 100$

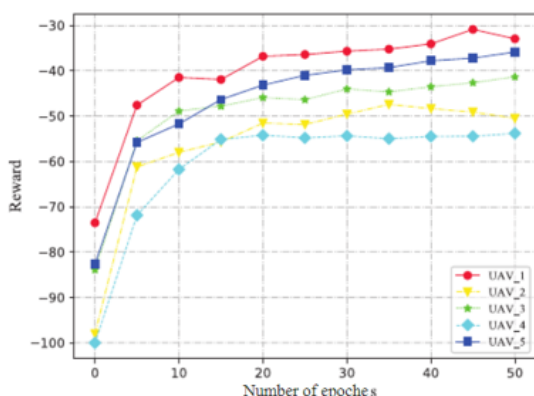


Рис. 3.5 Нагорода за планування завдань мережі БПЛА

Що стосується планування завдань, використовується нейронна мережу для побудови мережі глибокого Q-навчання зі швидкістю навчання 0,0025. Генерація завдань вузлами IoT відповідає розподілу Пуассона, а швидкість надходження дорівнює λ . Усього в системі існує три типи завдань і два типи ресурсів, які потрібні для кожного завдання. Між тим, ідеальний час виконання та вимоги до ресурсів для кожного завдання випадково генеруються з $[1, 15]$ і $[1, 10]$

На рисунку 3.5 і на рисунку 3.6 зображено винагороду та середнє уповільнення відповідно на кожній ітерації. Як і очікувалося, обидва значення зростають із покращенням ітерацій. На рисунку 3.5 чітко показано, що зі збільшенням кількості ітерацій загальна винагорода від DRL поступово зростатиме та наблизитиметься до ідеального значення. І рисунок 3.6 показує, що зі збільшенням ітерацій БПЛА навчилися краще планувати вхідні завдання, тому середнє уповільнення поступово наблизитиметься до мінімального значення.

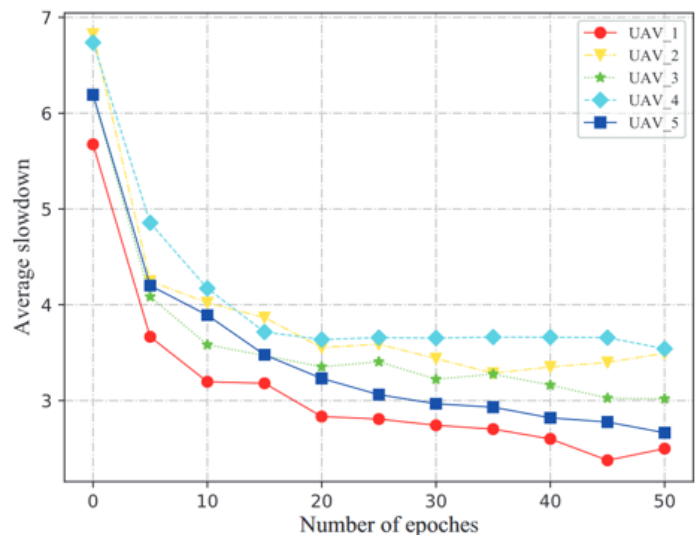


Рис. 3.6 Середнє уповільнення розвантажених завдань БПЛА

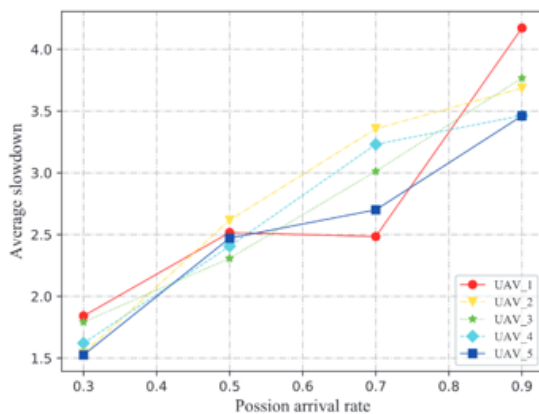


Рис. 3.7 Середнє уповільнення порівняно з темпами прибуття $\lambda = (0,3, 0,5, 0,7, 0,9)$

На рисунку 3.7 і 3.8 показано вплив кількості завдань на продуктивність планування БПЛА. Оскільки генерація завдань вузлами IoT відповідає розподілу Пуассона, ми можемо отримати, що зі збільшенням швидкості надходження завдань середнє уповільнення збільшується, а винагорода зменшується через обмежені можливості обробки кожного БПЛА.

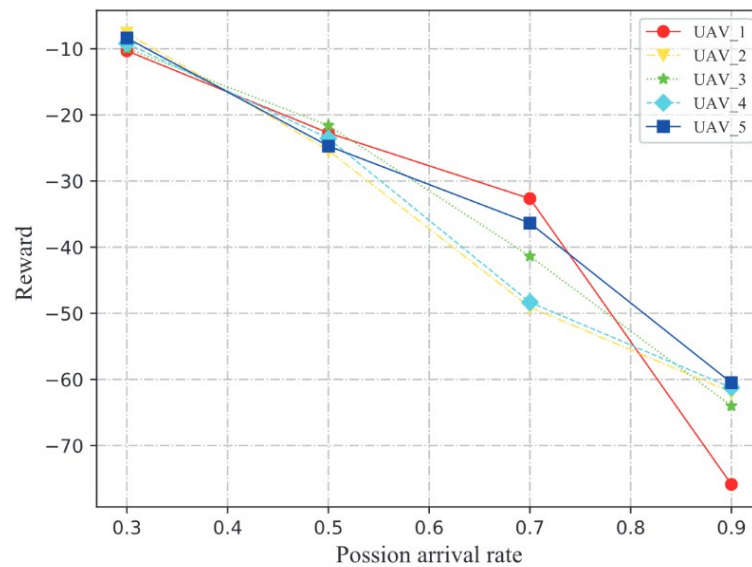


Рис. 3.8 Винагорода порівняно з темпами прибуття $\lambda = (0,3, 0,5, 0,7, 0,9)$

Порівнюємо запропонований алгоритм планування завдань із підтримкою DRL із трьома класичними алгоритмами планування завдань, тобто алгоритмом «першим прийшов, першим обслуговується» (First Come First Serve — FCFS), алгоритмом «найкоротша робота — спочатку» (Shortest Job First — SJF) і алгоритмом Round Robin (RR), як показано на рисунку 3.9.

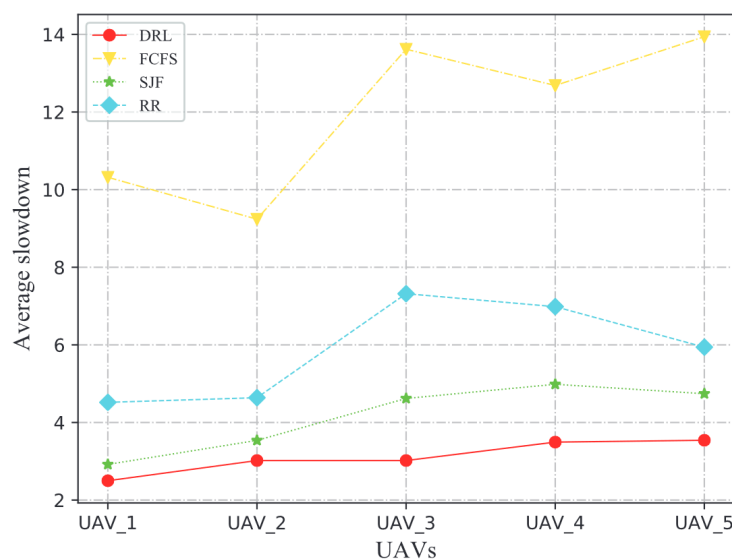


Рис. 3.9 Порівняння середнього уповільнення DRL, FCFS, SJF і RR

Зокрема, SJF спочатку планує менші завдання порівняно з FCFS і RR. Продуктивність алгоритму RR значною мірою залежить від вибору довжини часового кванта. Якщо часовий проміжок довший, він, як правило, демонструє таку саму поведінку, як FCFS. Однак, якщо часовий проміжок менший, ніж потрібно, більша кількість перемикань завдань може знизити ефективність планування завдань. Навпаки, алгоритм планування завдань із підтримкою DRL може вчитися на досвіді та резервувати мережеві ресурси. Таким чином, невеликі завдання, які нещодавно надійшли, можна вирішити одразу, а середнє уповільнення зведено до мінімуму.

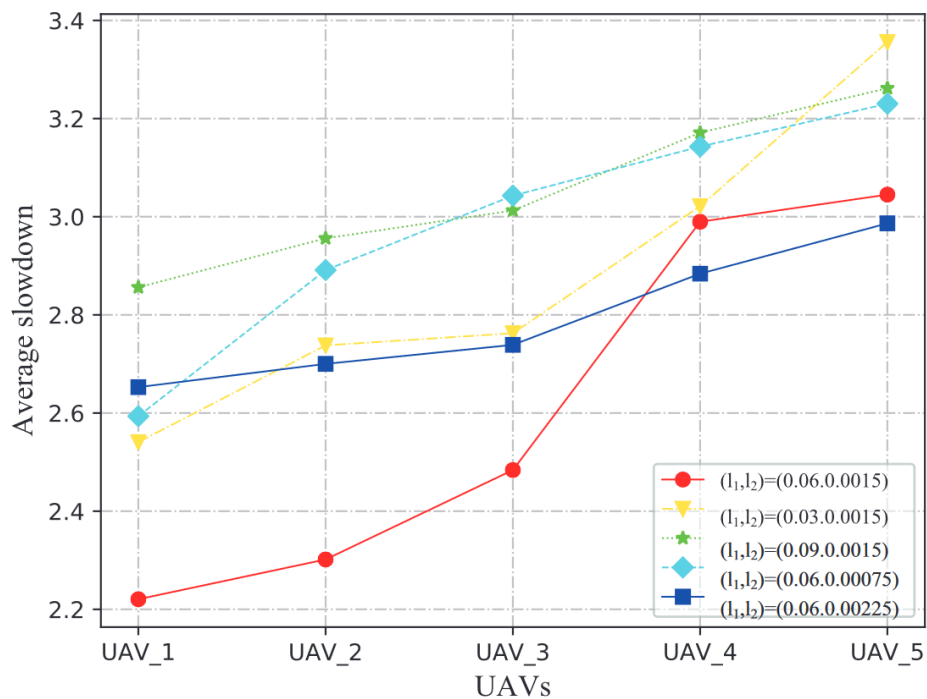


Рис. 3.10 Середнє уповільнення від вагових коефіцієнтів (ℓ_1, ℓ_2) для алгоритму DE

На рисунку 3.10 і рис. 3.11 показано вплив різних вагових коефіцієнтів ℓ_1 і ℓ_2 відповідно до рівняння (3.12). Оскільки ℓ_1 і ℓ_2 позначають вагові коефіцієнти для вимог балансування навантаження та середньої вартості передачі відповідно, відповідні ℓ_1 і ℓ_2 дадуть системі МЕС кращу продуктивність. На рисунку 3.10 і рисунку 3.11 видно, що, коли $\ell_1 = 0.06$ і $\ell_2 = 0.0015$, він отримає найкращу загальну

продуктивність у нашій системі. Крім того, незалежно від того, чи ℓ_1 і ℓ_2 занадто малі чи занадто великі, середнє уповільнення збільшиться, а винагорода зменшиться. Це тому, що недооцінка ℓ_1 призведе до незбалансованого навантаження між БПЛА, тоді як іноді деякі вузли МЕС не можуть відповідати вимогам якості обслуговування, а також завищення ℓ_1 призведе до збільшення вартості передачі БПЛА. Таким же чином недооцінка ℓ_2 призведе до високої вартості передачі, а завищення ℓ_2 призведе до незбалансованого навантаження.

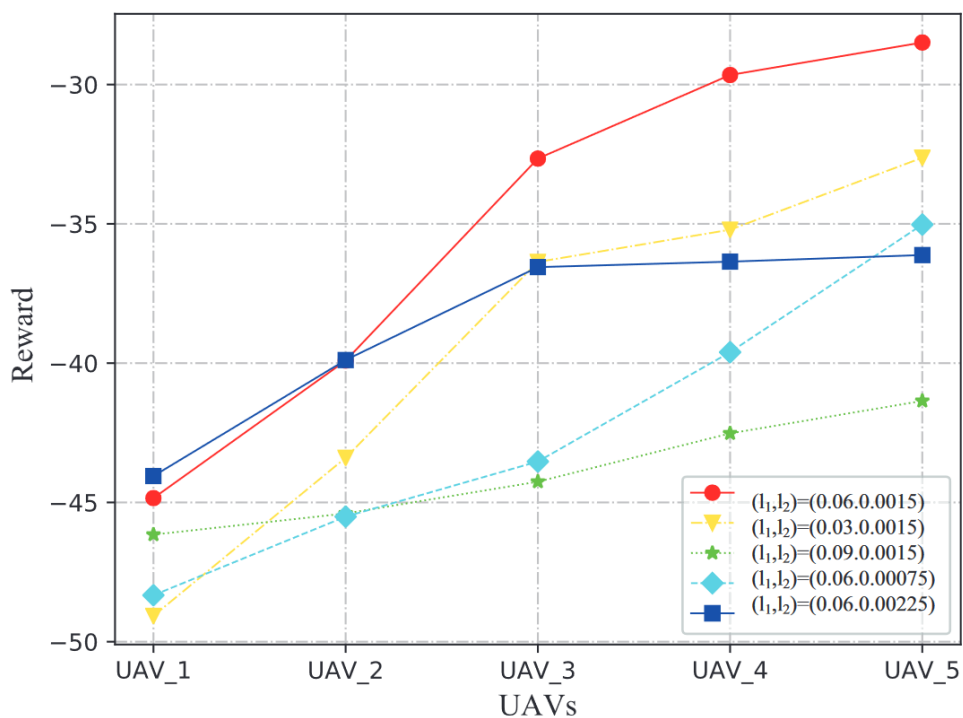


Рис. 3.11 Коефіцієнти винагороди та вагових коефіцієнтів (ℓ_1, ℓ_2) для алгоритму DE

На рисунку 3.12 показано порівняння з класичним генетичним алгоритмом (GA) та оптимізацією рою частинок (PSO) в алгоритмі 2. В DE розмір популяції дорівнює 10, а ймовірність кросинговеру, а також ймовірність мутації DE встановлені як 0,7 і 0,2 відповідно. У PSO число частинок дорівнює 10, що збігається з DE.

Крім того, вагові коефіцієнти швидкості, найкращого положення та глобально оптимального положення становлять 0,2, 0,5 та 0,5 відповідно. А в GA розмір популяції становить 64, і одну популяцію вилучено за покоління, де ймовірність кросинговеру, а також ймовірність мутації такі ж, як і DE. Очевидно, що зі збільшенням ітерацій алгоритм DE має найкращу придатність порівняно з двома іншими алгоритмами.

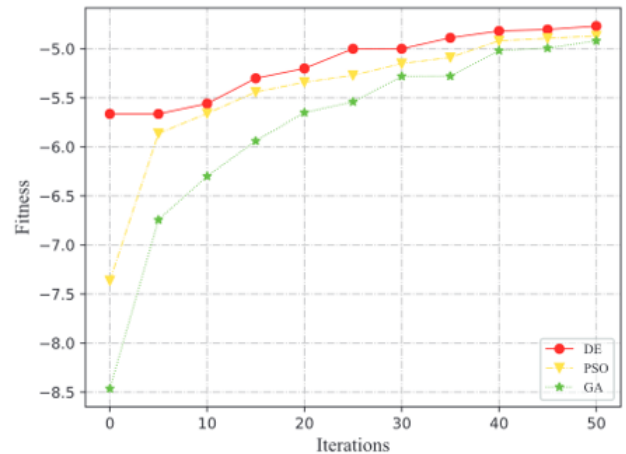


Рис. 3.12 Порівняння придатності DE, PSO та GA

Розвиток технології БПЛА надає нове рішення для розвантаження пристроїв Інтернету речей. Порівняно зі стаціонарними наземними базовими станціями, БПЛА можуть ефективно підходити до наземних пристроїв IoT завдяки легкому розгортанню, низькій вартості та доступній мобільності. У цьому дослідженні показано як розробити систему МЕС з кількома БПЛА з метою глобального балансу навантаження та мінімізації сповільнення. Продемонстровано алгоритм диференціальної еволюції для пошуку майже оптимальних місць розташування БПЛА. А також алгоритм апроксимації за допомогою GAP призначений для визначення кожного з'єднання між БПЛА та вузлами IoT. Крім того, для ефективного планування розвантажених завдань використовується схема планування завдань на основі DRL.

Результати моделювання підтверджують ефективність і перевагу запропонованого алгоритму.

4 ЕФЕКТИВНЕ ВИКОРИСТАННЯ РЕСУРСІВ У МЕС З ПІДТРИМКОЮ БПЛА

4.1 Управління ресурсами

У цьому підрозділі коротко пояснюється енергоефективне управління ресурсами в мережах МЕС з підтримкою БПЛА. Управління ресурсами є фактором, необхідним для завершення діяльності з метою досягнення бажаної цілі, наприклад пропускної здатності або максимальної потужності, які зазвичай обмежені в бездротових комунікаціях.

Управління ресурсами — це процес керування, планування або розподілу ресурсів від одного користувача/системи до іншого за допомогою різних схем для забезпечення загальної ефективної роботи системи. Для покращення продуктивності розвантажувальних служб динамічна доступність ресурсів відіграє вирішальну роль. Різні схеми розподілу приймаються, щоб гарантувати найкраще застосування ресурсів для досягнення продуктивності мережі U-МЕС і задоволення зростаючого попиту на програми, що потребують інтенсивних обчислень, і пристрої з обмеженими ресурсами.

Розподіл ресурсів (комунікація та обчислення/зберігання) має вирішальне значення для того, щоб відповідати вимогам QoS та QoE. У мережах U-МЕС розподіл ресурсів є ключовим викликом через обмеження акумулятора та траєкторії БПЛА. Ресурс може включати потужність передачі обчислювальної швидкості MU, обчислювальний ресурс на серверах МЕС, смугу пропускання та час, призначений користувачам, що займаються розвантаженням.

У моделі локальних обчислень частота центрального процесора (CPU) і час обчислень є ресурсами, які потрібно оптимізувати. У моделі розвантаження обчислень ресурсами, які потрібно оптимізувати, є такі ресурси зв'язку, як смуга пропускання, потужність розвантаження завдань, призначення завдань і час розвантаження. У моделі польоту БПЛА траєкторія, максимальний час польоту, вага та

швидкість БПЛА є ресурсами, які необхідно оптимізувати через їх обмежені обчислювальні ресурси через обмеження SWaP.

Розробка управління ресурсами в системах U-MEC досягає різних цілей:

- мінімізація енергоспоживання;
- час завершення (більше значення серед часу локального обчислення та часу обчислення розвантаження);
- мінімізація витрат;
- максимізація обчислювальних бітів;
- ефективність обчислень (обчислювальні біти на Джоуль енергії).

Як згадувалося вище, через компактні розміри та обмежений час автономної роботи пристроїв Інтернету речей вони мають обмежені запаси енергії та ресурси. Перезарядка або заміна батареї пристроїв IoT часто є досить неефективною. Таким чином, архітектура U-MEC використовується для зменшення споживання енергії цими пристроями, що призводить до подовження часу роботи батареї. Це досягається шляхом перенесення важких обчислювальних завдань користувачів на сусідній сервер MEC із багатими ресурсами, що в кінцевому підсумку підвищує енергоефективність усієї системи. Завдяки своїй маневреності, легкому розгортанню, гнучкості та мобільності БПЛА оснащені серверами MEC, що забезпечує додаткове покриття. Мережі U-MEC зазвичай використовуються в районах, де традиційна наземна інфраструктура недоступна. Енергія БПЛА споживається під час зависання, мобільності, руху, зв'язку та обчислень.

Як правило, БПЛА мають обмежений бортовий акумулятор, і їх потрібно часто заряджати. Енергія, яка споживається в мережах U-MEC для локального обчислювального процесу, складається з частоти процесора, часу обчислення та швидкості читання даних.

Споживання енергії в процесі розвантаження включає потужність передачі, енергію, пов'язану з обчисленнями, і час розвантаження завдання. Енергія, яка споживається в процесі польоту, включає енергію рушія, енергію зв'язку, швидкість та прискорену швидкість БПЛА.

4.2 Енергоефективність

Проблема: Система МЕС із підтримкою БПЛА стикається з трьома основними викликами:

- Великомасштабний простір пошуку, що ускладнює ефективне знаходження оптимальних рішень.
- Змішані змінні рішення, які вимагають інтегрованого підходу до розподілу ресурсів та планування завдань.
- Ігнорування кореляції між розгортанням БПЛА та плануванням завдань, що впливає на загальну ефективність системи.

Мета. Методом системи МЕС із підтримкою BPLA є зменшення споживання енергії всією системою шляхом оптимізації використання ресурсів та забезпечення відповідності обмеженням затримки.

Вирішення. Для вирішення зазначених проблем використовується дворівневий метод оптимізації:

- ✓ Верхній рівень: Використовується механізм кодування, що визначає розташування кожного BPLA, представляючи всю популяцію як повне розгортання BPLA.
- ✓ Нижній рівень: Рішення про розподіл ресурсів, планування завдань та розвантаження. Для оптимізації використовує алгоритм ToDeTaS (Two-Level Deployment and Task Scheduling), який виконує евристичний пошук на основі популяції, не потребуючи градієнтної інформації.

Проблема:

1. Енергоспоживання в МЕС-системах із підтримкою БПЛА:

- Задача мінімізації енергії, яка торгується на обчислення, передачу даних і механічний рух (зависання, політ), є складною через різноманітні параметри, які необхідно використовувати одночасно.

2. Висока складність розв'язання задачі:

- Не випуклість завдань через комплексні кореляції між розподілом обчислювальних потужностей, положенням БПЛА, плануванням завдань і асоціацією користувачів.
- Великий простір пошуку, який ускладнює використання традиційних методів оптимізації [27].

3. Неузгодженість рішення:

- Ігнорування взаємозалежності між плануванням завдань і розташуванням БПЛА результат до зниження ефективності.
- Складність у налаштуваннях оптимальної кількості БПЛА, які мінімізують енергоспоживання системи.

Мета. Глобальна оптимізація систем енергоспоживання, що включає:

- Енергетичні витрати на зв'язок, обчислення та траєкторію БПЛА.
- Врахування QoS для користувачів і обмеження затримки.
- Забезпечення балансу між ефективним виконанням завдань і споживанням ресурсів шляхом координації рішень на всіх рівнях систем.

Вирішення.

1. Оптимізація розташування та кількість БПЛА:

- ✓ Алгоритм диференціальної еволюції (DE) із використанням оператора виключення:
 - Забезпечує пошук оптимального розташування дронів.
 - Поступово зменшує кількість БПЛА до мінімально необхідного рівня після виконання завдань.
- ✓ Механізм кодування:
 - Оптимізує розташування дронів у просторі, зменшуючи складність пошуку.

2. Спільна оптимізація обчислень, комунікацій і траєкторій:

- ✓ Метод SCA (послідовні опукле наближення):
 - Перетворює не випуклу задачу на низку розв'язних опуклих задач.
- ✓ Координата блокового спуску (BCD):

- Використовується для поетапного розв'язання підзадач, таких як розподіл бітів, планування часових інтервалів, проектування траєкторій БПЛА.
- ✓ Лагранжева подвійність:
 - Ви можете розділити задачу на дві частини: оптимізацію обчислювальних потужностей і проектування траєкторій.
- 3. Нижній рівень оптимізації:
 - ✓ Жадібний алгоритм:
 - Мінімізує час обчислення для завдань бінарного програмування, забезпечуючи ефективність рішень щодо розвантаження завдань.
 - ✓ 1D-метод пошуку:
 - Оптимізує 3D-планування розташування та ширину зміни для БПЛА, скорочуючи енергетичні витрати.
- 4. Інтеграція механізмів збереження енергії:
 - ✓ Оптимізація траєкторії БПЛА:
 - Враховує енергетичні обмеження, причинність даних і вимоги до швидкості дронів.
 - ✓ Спільна оптимізація розвантаження завдань та обчислювальних ресурсів:
 - Забезпечує мінімальне споживання енергії при передачі, обчисленнях і завантаженні даних.
 - ✓ Декомпозиція задачі:
 - Розробка завдань на підзадачі забезпечує кращу керованість і швидкість пошуку неоптимальних рішень.
- 5. Переваги ToDeTaS:
 - ✓ Ефективність у скороченні простору пошуку:
 - Завдяки кодуванню та алгоритму виключення оператора швидко знаходяться оптимальні рішення.
 - ✓ Врахування кореляції між завданнями та розташуванням:
 - Забезпечує кращу інтеграцію рішень верхнього та нижнього рівнів.
 - ✓ Зменшення енергоспоживання:

- Досягається через точний розподіл ресурсів і узгодженість між завданнями.

Результат:

- Мінімізоване загальне споживання електроенергії системи МЕС із підтримкою БПЛА.
- Підвищена ефективність виконання завдань за допомогою інтегрованого підходу до оптимізації.
- Забезпечено виконання QoS для наземних користувачів при мінімальних витратах на ресурси.

У таблиці 4.1 представлено ключові методи, підходи та стратегії, які використовуються для вирішення завдань енергоефективності та оптимізації ресурсів у МЕС-системах із БПЛА. Це охоплює аспекти планування траєкторій, розподілу обчислювальних завдань, управління енергоспоживанням та інтеграції сучасних алгоритмів прогнозування і оптимізації.

Таблиця 4.1

Ефективні стратегії та методи оптимізації енергоспоживання в
МЕС-системах із використанням БПЛА

Категорія	Ключова інформація
<i>Мінімізація енергоспоживання БПЛА</i>	Задача оптимізації формулюється з урахуванням обмежень на кількість обчислювальних бітів, частоти процесора (ЦП) користувачів, причинності збору енергії та траєкторії БПЛА. Для вирішення застосовується метод послідовного опуклого наближення (SCA), який дозволяє поетапно наближатися до оптимального рішення. Розглянуто введення допоміжних змінних для трансформації неопуклої задачі в опуклу.

Продовження таблиці 4.1

Категорія	Ключова інформація
<i>Ефективне використання енергії</i>	Запропоновано трирівневу стратегію розвантаження обчислень: 1) оптимізація обсягів розвантажувальних обчислень; 2) частота процесорів дронів; 3) корекція траєкторії польоту. Алгоритм дозволяє зменшити енергоспоживання, враховуючи обмеження ресурсів БПЛА.
<i>Алгоритм прогнозування завдань</i>	Використання LSTM-алгоритму для прогнозування обчислювальних завдань, що передаються користувачами на БПЛА. Прогнозування сприяє підвищенню ефективності завдань та мінімізації енергоспоживання всієї системи.
<i>Оптимізація траєкторії БПЛА</i>	Траєкторія дронів адаптується для охоплення всіх користувачів, забезпечуючи покращену якість передачі даних та зменшення затримки. Використовуються алгоритми SCA, DAI та методика IPM для врахування енергетичних та обчислювальних обмежень.
<i>Модель кооперативної системи МЕС</i>	Формулюється задача спільної оптимізації, яка включає: частоту ЦП, кількість бітів для розвантаження, траєкторію БПЛА, потужність передачі. Задача враховує: 1) обмеження затримки користувачів; 2) причинність збору енергії; 3) обмеження обчислювальних ресурсів; 4) траєкторію дронів. Проблему розв'язують ітераційно, застосовуючи алгоритми SCA, декомпозиції та ітерації (DAI) для зменшення складності розрахунків.

Продовження таблиці 4.1

Категорія	Ключова інформація
<i>Траєкторія виявлення БПЛА</i>	Підхід DTPUS (<i>Detection Trajectory Planning for Multi-Sorties and UAV Scheduling</i>) включає: 1) групування вітрових турбін (WT); 2) планування траєкторій для кожного польоту; 3) оптимізацію розвантаження обчислень (IOTCO). Метод подвійності Лагранжа використовується для одночасної оптимізації частоти обчислень, часу розвантаження та потужності. Враховується вплив вітрових умов для підвищення надійності.
<i>Спільна оптимізація</i>	Розроблено підхід для спільної оптимізації: 1) розподілу потужності у висхідній лінії зв'язку; 2) обчислювальної потужності; 3) розподілу каналів; 4) 3D-розташування БПЛА. Враховуються обмеження на затримку, пропускну здатність і обчислювальні ресурси. Для вирішення задачі застосовуються методи BCD, СССР та умови ККТ.
<i>Енергоефективні стратегії</i>	1) Екстремальний випадок: Послідовне пробудження користувачів для зв'язку з БПЛА, що знижує складність системи. Використання подвійного методу Лагранжа для оптимізації енергоспоживання. 2) Практичний випадок: Спільна оптимізація ресурсів для користувачів та дронів, мінімізація енергоспоживання через кооперативний підхід. Проблеми розв'язуються шляхом декомпозиції задачі на під-проблеми.

Продовження таблиці 4.1

Категорія	Ключова інформація
<i>Моделі</i> <i>TDMA для MEC</i>	Модель передбачає розподіл ресурсів, траєкторій БПЛА та часу зависання для підвищення енергоефективності. Формулюється задача мінімізації енергії з обмеженням на асоціацію користувачів. Оптимізація виконується через подвійний метод Лагранжа та методику планування потоку.
<i>Системи MEC</i> <i>на основі NOMA</i>	Формулюється задача мінімізації максимального споживання енергії серед користувачів (задача min-max). Вирішується проблема спільної оптимізації розподілу обчислювальних ресурсів, траєкторії БПЛА та розподілу, завдань. Завдяки цьому зменшуються перешкоди між користувачами та забезпечується стійкість траєкторії дронів.
<i>Динамічне програмування</i>	Використовується для оптимізації енергоспоживання транспортних засобів через соціальний Інтернет транспортних засобів (SIoV). Розподіл потужності та траєкторії БПЛА адаптується відповідно до закону енергоспоживання кожного автомобіля.
<i>Режими розвантаження (бінарний/частковий)</i>	Формулюється задача оптимізації: кількість обчислювальних бітів, потужність передачі, частота ЦП, траєкторія БПЛА. Використовуються методи SCA та ADMM. Завдання користувачів виконуються локально або передаються на БПЛА залежно від компромісу між енергоспоживанням та швидкістю виконання.

Продовження таблиці 4.1

Категорія	Ключова інформація
<i>Енергоефективність у МЕС з БПЛА</i>	Використовуються методи для мінімізації споживання енергії: SCA, алгоритм Дінкельбаха, ADMM. У випадку невідомої мобільності користувачів застосовуються техніки просторового розподілу для прогнозування розташування та оптимального використання ресурсів.

4.3 Обчислювальна ефективність

Різні підходи до підвищення обчислювальної ефективності в МЕС-системах із БПЛА охоплюють аспекти оптимізації траєкторій, ресурсів, енергоспоживання та асоціації користувачів.

1. Максимізація ефективності обчислень через спільну оптимізацію ресурсів.

Розглядається задача оптимізації ефективності обчислень шляхом врахування частот центрального процесора (ЦП), часу розвантаження завдань, енергетичних обмежень, положення та мобільності БПЛА. Метод спільно оптимізує:

- потужність передачі користувачів;
- частоту ЦП;
- траєкторії БПЛА;
- час розвантаження.

Для цього застосовується метод подвійності Лагранжа для оптимізації потужності передачі та частоти ЦП, тоді як метод послідовного опуклого наближення (SCA) використовується для вирішення задачі оптимізації траєкторій БПЛА.

2. Оптимізація асоціації користувачів, траєкторій і ресурсів.

Пропонується спільна оптимізація асоціації користувачів, планування траєкторій та розподілу обчислювальних ресурсів для максимізації обчислювальної ефективності. Враховуються:

- локальний розподіл частот ЦП;
- потужність передачі;

- розподіл спектральних ресурсів;
- траєкторії БПЛА.

Для розв'язання задачі оптимізації застосовується параметрична модель, вирішення якої проводиться ітераційно через структуру подвійного циклу:

- Зовнішній цикл: використовує метод Дінкельбаха для оновлення ефективності обчислень.
- Внутрішній цикл: застосовується спільний алгоритм оптимізації для асоціації користувачів, траєкторій і розподілу ресурсів.

3. Максимізація кількості вивантажених даних при мінімізації енергоспоживання.

Описано підхід до максимізації ефективності обчислень, де кількість вивантажених даних збільшується, а споживання енергії БПЛА мінімізується.

Основна мета — забезпечення якісного обслуговування користувачів (QoE) через спільну оптимізацію:

- планування користувачів;
- траєкторій БПЛА;
- потужності передачі;
- розподілу пропускної здатності.

Проблема формулюється як неопукла, а її вирішення здійснюється за допомогою багатоетапного алгоритму оптимізації, що забезпечує високу ефективність обчислень навіть за обмежених ресурсів.

Ці підходи демонструють, що досягнення балансу між ефективністю обчислень та мінімізацією енергоспоживання в МЕС-системах із БПЛА є можливим завдяки інноваційним алгоритмам та методам оптимізації.

4.4 Мінімізація затримки

Мінімізація затримки є ключовим завданням у системах *Mobile Edge Computing* (МЕС) із використанням безпілотних літальних апаратів (БПЛА), оскільки вона безпосередньо впливає на якість обслуговування користувачів (QoE) [28].

У дослідженнях пропонуються різні методи, що дозволяють зменшити затримку завдяки оптимізації ресурсів, маршрутів і процесів розвантаження.

1. Оптимізація min-max затримки серед користувачів.

Основна мета полягає в мінімізації максимальної затримки серед усіх користувачів у кожному часовому інтервалі.

Для цього реалізується спільна оптимізація таких параметрів:

Бінарні змінні планування користувача – визначають, які користувачі передають дані в конкретний часовий проміжок.

Коефіцієнт розвантаження завдань – регулює частку завдань, що виконується на МЕС-серверах, зменшуючи навантаження на користувацькі пристрої.

Траєкторії БПЛА – налаштовуються для покращення передачі даних і охоплення користувачів.

Задача формулюється із врахуванням двійкових і дискретних обмежень. Для її вирішення застосовано підхід подвійної композиції штрафів (PDD), що складається з двох основних етапів:

(I) Внутрішній цикл: змінні оптимізуються за допомогою алгоритму увігнуто-опуклої процедури (СССР), який ефективно вирішує змішані завдання.

(II) Зовнішній цикл: оновлюються множники розширеного Лагранжа (AL) та коефіцієнти штрафів, щоб забезпечити збіжність алгоритму до оптимального рішення.

Для зменшення обчислювальної складності було розроблено алгоритм, заснований на l_0 -нормі, який значно скорочує кількість необхідних обчислень, зберігаючи при цьому точність.

2. Мінімізація затримки в багаторівневій МЕС-архітектурі

Дослідження зосереджене на трирівневій архітектурі обробки даних, що включає:

- Наземні користувачі – пристрої IoT, смартфони тощо.
- Периферійні вузли (UAV-BS) – БПЛА, які функціонують як базові станції та периферійні обчислювальні вузли.

- Хмарний центр – має практично необмежені ресурси та обслуговує складні обчислювальні запити.

Мета полягає у зменшенні затримки обробки даних на рівні БПЛА через оптимізацію:

- маршрутів для кількох БПЛА з урахуванням покриття території та навантаження;
- розподілу обчислювальних ресурсів між користувачами та периферійними вузлами;
- ефективного використання спектра.

Ключові аспекти:

Обмеження: враховуються енергетичні можливості БПЛА, обчислювальна здатність МЕС-вузлів і споживання енергії.

Для вирішення задачі використовується алгоритм на основі глибокого навчання з підкріпленням (DRL). Він дозволяє в реальному часі створювати маршрути для БПЛА з урахуванням покращення покриття та мінімізації затримок.

Політика управління ресурсами:

Застосовується оптимізація Ляпунова для стабілізації затримки та економії енергії в системі.

Для низької швидкості передачі даних зменшується частота процесора БПЛА, що дозволяє значно знизити енергоспоживання.

Для високої швидкості передавання даних розподіляється смуга пропускання, а частина даних розвантажується на МЕС-вузли.

Застосування цих підходів забезпечує ефективну мінімізацію затримки та збалансування енергоспоживання в МЕС-системах із БПЛА, що особливо важливо для масштабних розгортань.

4.5 Балансування навантаження та забезпечення секретності

Балансування навантаження та підтримка секретності в МЕС-системах із використанням безпілотних літальних апаратів є ключовими аспектами для

підвищення ефективності та безпеки системи. Різні дослідження пропонують методи, що враховують спільну оптимізацію ресурсів, планування завдань та забезпечення конфіденційності.

1. Глобальний баланс навантаження.

Розглядається проблема мінімізації глобального навантаження на БПЛА через спільну оптимізацію:

Планування завдань – визначення, які завдання та на яких пристроях мають бути виконані для рівномірного розподілу обчислювальних ресурсів.

Розгортання БПЛА – вибір оптимальних точок розташування для ефективного охоплення користувачів із врахуванням обмежень покриття.

Для вирішення цієї задачі використовується метод *Deep Reinforcement Learning* (DRL), який забезпечує:

- ефективне планування розвантажених завдань між кількома БПЛА;
- зменшення затримки передачі даних;
- підвищення QoS користувачів.

Цей підхід дозволяє в режимі реального часу знаходити оптимальні рішення для динамічних умов системи, зберігаючи баланс між навантаженням і продуктивністю.

2. Максимізація секретності.

Основна мета – максимізація мінімального рівня секретності з урахуванням наступних обмежень:

- Мінімальні вимоги до розвантаження даних і затримки.
- Загальні обмеження потужності системи.

Для досягнення цієї мети формулюється задача спільної оптимізації, що включає:

- потужність перешкод, які генеруються БПЛА для захисту від несанкціонованого доступу;
- обчислювальні ресурси та розташування БПЛА;
- потужність передачі користувачів і асоціацію завдань до МЕС-вузлів.

Ключові етапи:

Оптимізація розташування БПЛА – забезпечує підвищення секретності кожного каналу розвантаження та зменшення затримки. Для цього враховується невідоме місцезнаходження потенційних підслухувачів.

Розбиття задачі на підпроблеми – із використанням методу BCD (*Block Coordinate Descent*) вирішуються окремі підзадачі:

- Оптимізація потужності перешкод і передачі.
- Налаштування розташування БПЛА.
- Розподіл ресурсів між користувачами.

Метод SCA (*Successive Convex Approximation*) – використовується для вирішення неопуклих задач і оптимізації ключових параметрів системи.

В результаті продемонстровано, що існує компроміс між безпекою та затримкою: підвищення рівня секретності може збільшувати затримку, проте оптимальне управління ресурсами дозволяє знизити цей вплив.

3. Балансування обслуговування IoT пристроїв.

Вивчається задача збільшення кількості IoT-пристроїв, що можуть бути обслуговані МЕС-системою, за допомогою кількох БПЛА із серверами МЕС. Мета досягається через спільну оптимізацію:

Розвантаження обчислювальних задач – визначення частини завдань, які будуть оброблені на МЕС-вузлах.

Траєкторії БПЛА – налаштування маршрутів для забезпечення ефективного покриття.

Розподіл ресурсів – енергії, спектра та обчислювальних можливостей.

Оптимізація здійснюється за допомогою методу SCA, що дозволяє звести задачу до набору послідовно вирішуваних неопуклих підзадач. Це забезпечує швидке та точне знаходження рішень навіть для систем із великими обсягами даних і високими вимогами до продуктивності.

Застосування цих підходів дозволяє досягти балансу між продуктивністю, енергоспоживанням і секретністю системи, забезпечуючи високий рівень QoS та безпеки для користувачів у МЕС-мережах із БПЛА.

4.6 Майбутні напрямки досліджень

Система U-MEC розглядається як перспективна технологія для підвищення ємності, покриття, підключення, QoS і QoE користувача. Водночас вона стикається з низкою проблем та відкритих питань, які обмежують її застосування через складність і недостатню кількість проведених досліджень у контексті систем U-MEC. Нижче наведено деякі напрямки для майбутніх досліджень та відкриті питання.

- ❖ Виконання U-MEC: Оптимізація роботи системи U-MEC, зокрема з урахуванням динамічної взаємодії з користувачами, залишається складним і відкритим питанням.

- ❖ Обмежений час польоту БПЛА: Ефективні схеми управління онлайн-ресурсами в динамічному каналному середовищі є пріоритетом через невизначеність мобільності користувачів. БПЛА повинні обслуговувати кількох наземних користувачів на великих географічних територіях.

- ❖ Мобільність користувачів та БПЛА: Спільна оптимізація траєкторій БПЛА, параметрів зв'язку та розподілу обчислювальних ресурсів може значно покращити продуктивність системи для рухомих користувачів і БПЛА.

- ❖ Енергозбереження БПЛА: Оптимізація траєкторії польоту з урахуванням енергоефективності є важливим напрямком. Координація траєкторій кількох БПЛА дозволить збільшити пропускну здатність, покриття та ефективність обчислень.

- ❖ Сценарії з багатьма БПЛА та користувачами: Дослідження таких сценаріїв може сприяти максимізації ефективності обчислень та зменшенню затримки.

- ❖ Моделі зв'язку: Складні моделі зв'язку, враховуючи мобільність користувачів і фактори загасання сигналу, є перспективною темою для майбутніх робіт.

- ❖ Розвантаження завдань: У сценаріях MEC з кількома БПЛА варто враховувати багато стрибкові мережі та адаптивні механізми розподілу задач.

- ❖ Безпека: Використання кількох легалізованих БПЛА для розвантаження користувачів відкриває нові виклики у забезпеченні безпеки. Методи фізичного рівня можуть покращити захист від атак.

❖ Інтеграція DRL: Оптимізація зв'язку, обчислень і кешування в мережах U-MEC може бути реалізована через алгоритми покриття БПЛА та прогнозування ресурсів на основі глибокого навчання з підкріпленням (DRL).

❖ Реальні моделі каналів: Замість ідеальних LoS-зв'язків в A2G/G2A варто враховувати реалістичні моделі, наприклад, завмирання Ріка чи ймовірнісні моделі.

❖ Компроміс між затримкою та енергією: Аналіз компромісів між затримкою та споживанням енергії для передачі даних з одного БПЛА до кількох користувачів є перспективним напрямком досліджень.

❖ Тривалість обчислень та черга задач: Аналіз часу обробки завдань та оптимізація черг обчислень можуть суттєво вплинути на продуктивність системи.

❖ Просторовий розподіл пропускної здатності: Обмежена потужність граничних вузлів створює виклик у забезпеченні рівномірного розподілу пропускної здатності між вузлами.

❖ Автономне планування шляху: Зміни в середовищі, що виникають раптово, можуть ускладнити автономне планування маршрутів для БПЛА.

❖ Оптимізація траєкторій: Врахування обмежень на швидкість, енергію та час польоту у процесі планування траєкторій БПЛА є складним, але необхідним завданням.

❖ Аналіз продуктивності: Параметри, такі як пропускна здатність, затримка, покриття, надійність та ємність, суттєво впливають на ефективність системи U-MEC. Їхня оптимізація вимагає комплексного підходу.

ВИСНОВКИ

У рамках даної атестаційної роботи мною було розглянуто і підтверджено перспективність інтеграції безпілотних літальних апаратів (БПЛА) у системи мобільних периферійних обчислень (МЕС). Ця технологія демонструє значний потенціал у забезпеченні ефективного функціонування мереж Інтернету речей (IoT) за рахунок зменшення затримок у передачі даних і оптимізації енергоспоживання.

У роботі розроблено модель МЕС-систем із підтримкою БПЛА, яка враховує ключові аспекти комунікаційних і обчислювальних процесів. Особливу увагу приділено балансуванню навантаження між вузлами мережі та оптимізації використання ресурсів для підвищення продуктивності системи.

Запропоновані підходи до оптимізації затримок і енерговитрат, включно з використанням алгоритмів навчання з підкріпленням та методів диференціальної еволюції, дозволили досягти значного покращення в ефективності роботи МЕС-систем порівняно з існуючими рішеннями. Результати моделювання підтвердили доцільність використання таких підходів для реальних застосувань.

Отримані результати мають практичну цінність і можуть бути застосовані в інфраструктурі розумних міст, транспортній логістиці, сільському господарстві, а також у промислових IoT-системах. Використання запропонованих методик сприятиме підвищенню надійності та продуктивності мереж, одночасно забезпечуючи економію енергії.

Ця робота відкриває нові можливості для подальших досліджень у напрямку інтеграції МЕС-систем із сучасними технологіями, такими як штучний інтелект, автономні системи та квантові обчислення. Отримані результати створюють основу для вдосконалення існуючих рішень і розробки інноваційних підходів до оптимізації МЕС-систем у майбутньому.

ПЕРЕЛІК ПОСИЛАНЬ

1. Pham, Q.-V., Hwang, W.-J. Resource allocation for heterogeneous traffic in complex communication networks. *IEEE Trans. Circuits Syst. II Exp. Briefs*, 63(10), 2016. — 959-963 с.
2. Bor-Yaliniz, R. I., El-Keyi, A., Yanikomeroglu, H. Efficient 3-D placement of an aerial base station in next generation cellular networks. *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016. — 1-5 с.
3. Zeng, Y., Zhang, R., Lim, T. J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.*, 54(5), 2016. — 36-42 с.
4. Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., Debbah, M. A tutorial on UAVs for wireless networks: Applications challenges and open problems. *IEEE Commun. Surveys Tuts.*, 21(3), 2019. — 2334-2360 с.
5. Mobile-edge computing introductory technical white paper. Sophia Antipolis, France, 2014. — [Електронний ресурс]. Режим доступу: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf
6. Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., Young, V. Mobile edge computing—A key technology towards 5G. *ETSI White Paper*, 11, 2015. — 1-16 с.
7. Wang, Y., Ru, Z.-Y., Wang, K., Huang, P.-Q. Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. *IEEE Trans. Cybern.*, 50(9), 2020. — 3984-3997 с.
8. Zhang, J., Zhou, L., Tang, Q., Ngai, E. C.-H., Hu, X., Zhao, H., et al. Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing. *IEEE Internet Things J.*, 6(2), 2019. — 3688-3699 с.
9. Nguyen, V., Khanh, T. T., Nam, P. V., Thu, N. T., Hong, C. S., Huh, E.-N. Towards flying mobile edge computing. *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2020. — 723-725 с.

10. Zhou, F., Hu, R. Q., Li, Z., Wang, Y. Mobile edge computing in unmanned aerial vehicle networks. *IEEE Wireless Commun.*, 27(1), 2020. — 140-146 с.
11. Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2014-2019. Cisco White Paper, 2015. — [Электронный ресурс]. Режим доступа: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf
12. 5G Infrastructure Association. Vision White Paper. 5G PPP, 2015. — [Электронный ресурс]. Режим доступа: <http://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf>
13. ETSI. Mobile Edge Computing: A key technology towards 5G. ETSI White Paper, September 2015. — [Электронный ресурс]. Режим доступа: www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf
14. International Data Corporation (IDC). IDC Reveals Worldwide Internet of Things Predictions for 2015. Press Release, 2015. — [45 ресурс]. Режим доступа: <https://www.idc.com/getdoc.jsp?containerId=prUS25291514>
15. ETSI GS MEC 003 V1.1.1. Mobile Edge Computing (MEC); Framework and Reference Architecture. ETSI Technical Specification, March 2016.
16. Shan, X., Zhi, H., Li, P., Han, Z. A survey on computation offloading for mobile edge computing information. *Proc. IEEE 4th Int. Conf. Big Data Secur. Cloud (Bigdata Security) Int. Conf. High Perform. Smart Comput. (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2018. — 248-251 с.
17. Li, M., Wu, Q., Zhu, J., Zheng, R., Zhang, M. A computing offloading game for mobile devices and edge cloud servers. *Wireless Commun. Mobile Comput.*, 2018. — 1-10 с.
18. Wu, Q., Zeng, Y., Zhang, R. Joint trajectory and communication design for multi-UAV enabled wireless networks. *IEEE Trans. Wireless Commun.*, 17(3), 2018. — 2109-2121 с.
19. Mishra, D., Natalizio, E. A survey on cellular-connected UAVs: Design challenges enabling 5G/B5G innovations and experimental advancements. *Comput. Netw.*, 182, Dec. 2020.

20. Hua, M., Wang, Y., Li, C., Huang, Y., Yang, L. UAV-aided mobile edge computing systems with one by one access scheme. *IEEE Trans. Green Commun. Netw.*, 3(3), 2019. — 664-678 с.
21. Multiple Access Schemes for Cellular Systems. *Electronic Notes*, Oct. 2020. — [Электронный ресурс]. Режим доступа: <https://www.electronics-notes.com/articles/connectivity/cellular-mobile-phone/multiple-access-schemes-technology-techniques.php>
22. What's the Difference Between FDD and TDD. *Electronic Design*, Oct. 2020. — [Электронный ресурс]. Режим доступа: <https://www.electronicdesign.com/technologies/communications/article/21801257/whats-the-difference-between-fdd-and-tdd>
23. Gupta, L., Jain, R., Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials*, 18(2), 2015. — 1123-1152 с.
24. Motlagh, N. H., Taleb, T., Arouk, O. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet of Things Journal*, 3(6), 2016. — 899–922 с.
25. Martello, S. Knapsack problems: Algorithms and computer implementations. *Wiley-Interscience Series in Discrete Mathematics and Optimization*, 1990.
26. Yao, H., Mai, T., Jiang, C., Kuang, L., Guo, S. AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing. *IEEE Computational Intelligence Magazine*, 14(4), 2019. — 21-30 с.
27. Li, L., Wen, X., Lu, Z., Pan, Q., Hu, W. J. A. Z. Energy-efficient UAV-enabled MEC system: Bits allocation optimization and trajectory design. *Sensors*, 19(20), 2019. — 4521 с.
28. Hu, Q., Cai, Y., Yu, G., Qin, Z., Zhao, M., Li, G. Y. Joint offloading and trajectory design for UAV-enabled mobile edge computing systems. *IEEE Internet Things J.*, 6(2), 2019. — 1879-1892 с.

29. Wu, G., Miao, Y., Zhang, Y., Barnawi, A. Energy efficient for UAV-enabled mobile edge computing networks: Intelligent task prediction and offloading. *Comput. Commun.*, 150, Jan. 2020. — 556-562 c.
30. Li, M., Wu, Q., Zhu, J., Zheng, R., Zhang, M. A computing offloading game for mobile devices and edge cloud servers. *Wireless Commun. Mobile Comput.*, 2018. — 1-10 c.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ
(Презентація)

- **Мета роботи** – розробка та оптимізація архітектури мобільних периферійних обчислень (MEC) із підтримкою безпілотних літальних апаратів (БПЛА) для забезпечення ефективного управління ресурсами, зниження затримки та енерговитрат у мережах IoT.
- **Об'єкт дослідження** – система мобільних периферійних обчислень із підтримкою БПЛА.
- **Предмет дослідження** – методи та алгоритми оптимізації управління ресурсами і мінімізації затримки в MEC-системах із підтримкою БПЛА.

Для досягнення поставленої мети мною були вирішені такі завдання:

1. Дослідити особливості архітектури MEC та інтеграції БПЛА.
2. Розробити підходи до оптимізації розміщення ресурсів і балансування навантаження.
3. Вивчити можливі схеми доступу для систем U-MEC.
4. Провести моделювання та аналіз запропонованих методів.

Актуальність роботи

Сучасний розвиток інформаційно-комунікаційних технологій обумовлює зростаючу потребу в забезпеченні ефективних обчислювальних ресурсів і високої продуктивності мереж. Впровадження мобільних периферійних обчислень (*Mobile Edge Computing, MEC*) у поєднанні з використанням безпілотних літальних апаратів (*UAV/BLA*) відкриває нові перспективи для розв'язання завдань у сферах IoT, розумних міст, логістики та надзвичайних ситуацій.



Можливість інтеграції БПЛА в MEC-системи забезпечує розширення зон покриття, підвищення адаптивності та зниження затримки в обробці даних. Разом із тим, постають нові виклики, зокрема оптимізація розміщення ресурсів, управління затримками, балансування навантаження і забезпечення енергоефективності.

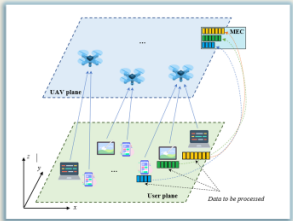
Моя робота спрямована на дослідження концептуальних основ інтеграції БПЛА та MEC, а також розробку підходів до підвищення ефективності таких систем.

Огляд проблематики

1. **Ключова проблема:**
 - Пристрої IoT мають обмежені обчислювальні можливості та енергоресурси.
 - Традиційні MEC-системи часто не здатні забезпечити достатню якість обслуговування через статичність базових станцій.
2. **Актуальність використання БПЛА:**
 - БПЛА дозволяють розширити покриття MEC і підвищити ефективність обчислювальних ресурсів.
 - Використання дронів як вузлів MEC стикається з проблемами оптимального розташування та планування завдань.
3. **Виклики:**
 - Балансування навантаження між кількома БПЛА.
 - Забезпечення покриття IoT-вузлів із урахуванням обмежень ресурсів.
 - Підвищення ефективності планування завдань для зменшення енергоспоживання та затримки.



Результати проведеної роботи



- Мною було розроблено систему із підтримкою кількох БПЛА, де дрони діють як вузли MEC, щоб забезпечити обчислювальні послуги для пристроїв IoT.

У ході тестувань та пошуку оптимальних рішень мною було запропоновано механізм розгортання БПЛА на основі диференціальної еволюції (DE), який моделює проблему доступу як узагальнену задачу призначення (GAP).

Це рішення дозволяє досягти балансування навантаження між дронами, забезпечуючи покриття мережі та якість обслуговування.

Алгоритм 1 Планування завдань на основі слабкого навчання з підкріпленням

Input: states, S_2 maximum number of iteration, t_{max}

Output: action-value function, $Q(S, A)$

- 1: initialize action-value function Q with random weights;
- 2: for $t=1, 2, \dots, t_{max}$: see do;
- 3: divide step t into k sub-steps;
- 4: select a random sub-action in each sub-step, and all sub-actions make up action a_t ;
- 5: execute action a_t , then obtain the reward r_t , and arrive at the next state s_{t+1} ;
- 6: calculate target Q -value according to Eq. (3.28);
- 7: update the deep Q -network by minimizing the loss according to Eq. (3.27);
- 8: end for;

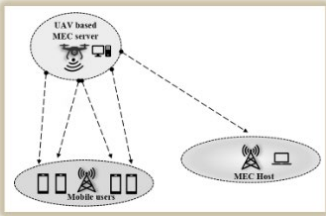
Алгоритм 2 Розподілення кількості БПЛА на основі DE

Input: the set of IoT nodes, n ; objective function; the number of nodes, N ; maximum number of iteration, t_{max}

Output: deployment of UAVs, $\{U_1, U_2, \dots, U_N\}$; best IoT node assignment, $a^*(N)$

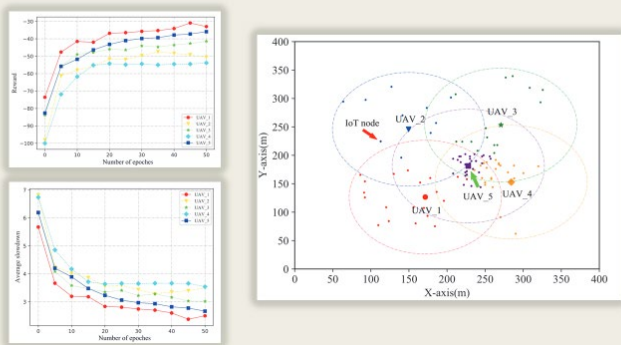
- 1: randomly initialize populations $P_0(t)$;
- 2: for $t=1, 2, \dots, t_{max}$: see do;
- 3: obtain mutation populations through differential strategy according to Eq. (3.29);
- 4: obtain crossover populations according to Eq. (3.30);
- 5: determine reference load for multiple UAVs according to Eq. (3.28);
- 6: get the fractional solution of node assignment problem in Eq. (3.28);
- 7: construct the bipartite graph based on the fractional solution;
- 8: deterministic round the fractional solution;
- 9: find the optimal task scheduling scheme according to Algorithm 1;
- 10: calculate the fitness for all population according to the objective function of Eq. (3.31);
- 11: get the next generation of St according to Eq. (3.31);
- 12: end for;

Варіант схеми доступу для систем U-MEC



Ретрансляційний U-MEC:

Результати моделювання



Аналіз ефективності алгоритмів та параметрів системи

Тема	Опис та результати
Розгортання мушкет-БПЛА	Розроблено майже оптимальну стратегію розташування БПЛА, що враховує 5 дронів і 100 вузлів IoT. Вузли IoT підключаються до найближчого дрона для забезпечення стабільності зв'язку.
Винагорода алгоритму	Алгоритм глибокого Q-навчання поступово покращується з кожною ітерацією, наближаючись до ідеального значення.
Середнє уповільнення	Уповільнення зменшується завдяки навчання БПЛА ефективно планувати входні завдання.
Швидкість надходження завдань	Зі збільшенням кількості завдань середнє уповільнення зростає, а загальна винагорода зменшується через обмежені ресурси.
Порівняння з іншими алгоритмами	Алгоритм DRL перевершує класичні методи (FCFS, SJF, RR) за продуктивністю завдань адаптивності та оптимальному використанню ресурсів.
Вплив вагових коефіцієнтів	Оптимальні значення вагових коефіцієнтів забезпечують баланс між якістю обслуговування, навантаженням і вартістю передачі даних.
Ефективність DE	Порівняно з класичними алгоритмами GA та PSO, диференціальна еволюція демонструє кращі результати завдяки оптимальним параметрам.

Практична значущість

■ Запропоновані методи та підходи можуть бути використані для оптимізації роботи реальних IoT-мереж, забезпечуючи покращення якості обслуговування (QoS) і користувацького досвіду (QoE).



■ Результати дослідження можуть бути використані для вдосконалення сучасних мереж зв'язку, а також у сфері автоматизації логістики, моніторингу та управління надзвичайними ситуаціями.

Висновки

У рамках даної роботи мною було розглянуто і підтверджено перспективність інтеграції безпілотних літальних апаратів (БПЛА) у системи мобільних периферійних обчислень (МЕС). Ця технологія демонструє значний потенціал у забезпеченні ефективного функціонування мереж Інтернету речей (IoT) за рахунок зменшення затримок у передачі даних і оптимізації енергоспоживання.

У роботі розроблено модель МЕС-систем із підтримкою БПЛА, яка враховує ключові аспекти комунікаційних і обчислювальних процесів. Особливу увагу приділено балансуванню навантаження між вузлами мережі та оптимізації використання ресурсів для підвищення продуктивності системи.

Запропоновані підходи до оптимізації затримок і енерговитрат, включно з використанням алгоритмів навчання з підкріпленням та методів диференціальної еволюції, дозволили досягти значного покращення в ефективності роботи МЕС-систем порівняно з існуючими рішеннями. Результати моделювання підтвердили доцільність використання таких підходів для реальних застосувань.

Ця робота відкриває нові можливості для подальших досліджень у напрямку інтеграції МЕС-систем із сучасними технологіями, такими як штучний інтелект, автономні системи та квантові обчислення. Отримані результати створюють основу для вдосконалення існуючих рішень і розробки інноваційних підходів до оптимізації МЕС-систем у майбутньому.