

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

на тему «Метод прийняття рішень у взаємозалежних обчислювальних системах»

КвРКІП. 301158.19.01.08 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1

 Дмитро КРИЖАНІВСЬКИЙ
Підпис Ім'я, прізвище

Керівник: д.т.н., професор
Науковий ступінь, вчене звання

 Олег САВЕНКО
Підпис Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КІС, доктор філософії, доцент

Ольга ПАВЛОВА 

09 04 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 01 ” 09 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Дмитру Крижанівському

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод прийняття рішень у взаємозалежних обчислювальних системах

Керівник проекту (роботи) Олег САВЕНКО, д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Аналіз стратегій прийняття рішень у взаємозалежних системах

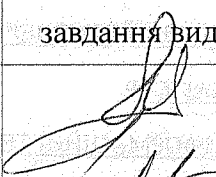
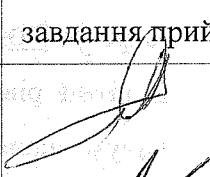
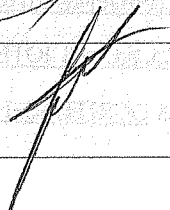
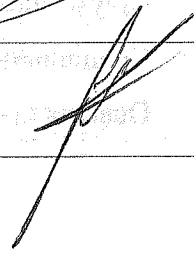
Метод прийняття рішень у взаємозалежних обчислювальних системах

Інтелектуальні підходи до розвантаження обчислень у МЕС-системах

Система автономного прийняття рішень у взаємозалежних обчислювальних системах

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагиат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

КАЛЕНДАРНИЙ ПЛАН


№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент


Підпис

Дмитро КРИЖАНІВСЬКИЙ
Ім'я, прізвище

Керівник роботи


Підпис

Олег САВЕНКО
Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Метод прийняття рішень у взаємозалежних обчислювальних системах»

Автор роботи: Крижанівський Дмитро Андрійович

Керівник роботи: Савенко Олег Станіславович

Пояснювальна записка: 74 с., 9 рис., 8 табл., 2 дод., 80 джерел.

ВЗАЄМОЗАЛЕЖНІ ОБЧИСЛЮВАЛЬНІ – СИСТЕМИ, МЕС-СИСТЕМИ, ПРИЙНЯТТЯ РІШЕНЬ, РЕПУТАЦІЙНІ МОДЕЛІ, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, МУЛЬТИАГЕНТНІ СИСТЕМИ.

Об'єктом дослідження є процес прийняття рішень у взаємозалежних обчислювальних системах.

Предметом дослідження є методи й підходи до інтелектуального розподілу завдань та управління ресурсами у мультиагентних і МЕС-системах з використанням репутаційних моделей і навчання з підкріпленням.

Метою кваліфікаційної роботи магістра є підвищення ефективності прийняття рішень у взаємозалежних обчислювальних системах за рахунок поєднання репутаційних механізмів, методів навчання та алгоритмічного моделювання взаємодії агентів.

Для розв'язання поставлених задач використовувалися методи теорії ігор, математичної оптимізації, байєсівського виводу, алгоритмів навчання з підкріпленням і мультиагентного моделювання.

Наукова новизна отриманих результатів:

- запропоновано метод прийняття рішень на основі репутаційної оцінки агентів із байєсівським оновленням та логарифмічно-лінійним навчанням, що забезпечує досягнення рівноваги у МЕС-системах;
- розроблено комбінований алгоритм з урахуванням адаптивного коригування Q-функцій і перевірки на рівновагу Байєса-Неша.

На основі проведених досліджень запропоновано інтелектуальну модель розподілу обчислень та управління агентами в МЕС-середовищі.

Практична значимість отриманих результатів полягає в можливості впровадження запропонованих підходів до систем розумного міста, автономного транспорту та хмарної інфраструктури з метою підвищення адаптивності, стійкості та ефективності розподілу ресурсів.

У вступі подано об'єкт і предмет дослідження, сформульовано мету, завдання, визначено наукову новизну та практичну цінність роботи, а також описано структуру дослідження.

У першому розділі проаналізовано сучасні підходи до прийняття рішень у взаємозалежних системах, зокрема у МЕС та мультиагентному середовищі, із залученням штучного інтелекту.

У другому розділі розглянуто математичні моделі прийняття рішень, включаючи оптимізаційні методи, генетичні алгоритми, динамічне програмування та алгоритми навчання.

У третьому розділі представлено інтелектуальні механізми розвантаження МЕС-систем, зокрема репутаційні та контрактні моделі, децентралізовані стратегії та механізми навчання.

У четвертому розділі запропоновано алгоритм адаптивного прийняття рішень на основі репутації, проведено моделювання поведінки агентів у змінному середовищі, а також аналіз ефективності розробленого підходу.

У висновках підведено підсумки досягнутих результатів та окреслено напрями подальших досліджень.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	7
1 АНАЛІЗ СТРАТЕГІЙ ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ	9
1.1 Поняття про стратегії прийняття рішень у системах	9
1.2 Аналіз методів оптимізації при прийнятті рішень	12
1.3 Постановка задачі.....	21
1.4 Висновки до першого розділу.....	22
2 МЕТОД ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ СИСТЕМАХ ...	23
2.1 Елементи теорії ігор та штучного інтелекту при прийнятті рішень.....	24
2.2 Теорія контрактів при прийнятті рішень	31
2.3 Висновки до другого розділу	37
3 ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО РОЗВАНТАЖЕННЯ ОБЧИСЛЕНЬ У МЕС-СИСТЕМАХ	38
3.1 Інтелектуальні стратегії розподілу обчислювальних завдань у МЕС-системах	38
3.2 Репутаційно-контрактні механізми та управління ресурсами	50
3.3 Висновки до третього розділу	57
4 СИСТЕМА АВТОНОМНОГО ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ	58
4.1 Математико-алгоритмічне забезпечення системи з репутаційно-адаптивним прийняттям рішень.....	58
4.2 Реалізація, моделювання та верифікація репутаційного методу	64
4.3 Висновки до четвертого розділу.....	77

	4
ВИСНОВОК	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	81
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	87
ДОДАТОК Б. НАУКОВА ПРАЦЯ ЗДОБУВАЧА	89
ДОДАТОК В. РЕЗУЛЬТАТИ ПЕРЕВІРКИ НА ПЛАГІАТ	104
ДОДАТОК Г. ПРЕЗЕНТАЦІЯ РОБОТИ	105

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT – інтернет речей

ІІІ – штучний інтелект

ГНМ – глибока нейронна мережа

ГНП – глибоке навчання з підкріпленням

ІЕЕЕ – Інститут інженерів з електротехніки та електроніки (ІЕЕЕ)

ІТ – інформаційні технології

ЛЛН – логарифмічно-лінійне навчання

LTE – зв'язок четвертого покоління

МППР – марковський процес прийняття рішень

МН – машинне навчання

РН – рівновага Неша

НМД – неортогональний множинний доступ

ДРН – досконала рівновага Неша

Q – значення функції оцінки Q

MEC – Multi-Access Edge Computing (граничні обчислення з множинним доступом)

ETSI – European Telecommunications Standards Institute (Європейський інститут телекомунікаційних стандартів)

НП – навчання з підкріпленням

SARSA – стан-дія-винагорода-стан-дія (SARSA)

БПЛА – безпілотний літальний апарат

ОС – операційна система

ЯО – якість обслуговування

СІІІ – самоусвідомлений штучний інтелект

НА – навчальний автомат

ТК – теорія контрактів

ЯДК – якість досвіду користувача

НМГП – навчання методом градієнтного підйому

БС – базова станція

ІСК – інформація про стан каналу

ДНВ – динаміка найкращої відповіді

БСІ – байєсівська сироватка істинності

ВСТУП

У сучасному цифровому середовищі взаємозалежні обчислювальні системи набувають все більшого значення у різноманітних сферах – від Інтернету речей (IoT) до автономних систем і розподілених обчислень. В таких системах автономні агенти діють не ізольовано, а тісно взаємодіють між собою, що формує складні поведінкові структури і створює потребу в нових методах прийняття рішень. Ефективність таких систем напряду залежить від здатності агентів адаптувати свої дії до змінного середовища та динаміки інших учасників.

Однією з ключових проблем, пов'язаних із функціонуванням взаємозалежних обчислювальних систем, є досягнення балансу між локальними цілями окремих агентів та глобальною ефективністю всієї системи. У зв'язку з цим зростає роль методів прийняття рішень, які базуються на комбінації математичних моделей, теорії ігор, навчання з підкріпленням і репутаційних механізмів.

Актуальність даної роботи полягає у необхідності створення універсального та адаптивного методу прийняття рішень для багатоагентних обчислювальних систем з високим рівнем взаємозалежності. У таких системах виникає потреба не лише в оптимальному виборі стратегій взаємодії, а й у забезпеченні стійкості до недобросовісної поведінки, ефективної адаптації до змін середовища та мінімізації обчислювальних витрат.

Метою магістерської роботи є розробка методу прийняття рішень у взаємозалежних обчислювальних системах, який поєднує елементи теорії ігор, байєсівської корекції репутації та логарифмічно-лінійного навчання.

Для досягнення поставленої мети вирішуються наступні завдання:

- проаналізувати сучасні підходи до прийняття рішень у багатоагентних обчислювальних системах;
- формалізувати модель взаємозалежностей між агентами з урахуванням репутаційних механізмів;
- розробити гібридний метод прийняття рішень, який враховує адаптацію стратегій на основі навчання з підкріпленням;

- здійснити алгоритмічну реалізацію методу та провести експериментальне моделювання динаміки системи;
- оцінити ефективність методу за критеріями адаптивності, стійкості до саботажу та швидкості досягнення стратегічної рівноваги.

Об'єктом дослідження є процес прийняття рішень у багатоагентних взаємозалежних обчислювальних системах.

Предметом дослідження є методи адаптивного прийняття рішень, що базуються на репутаційних оцінках, теорії ігор та алгоритмах навчання з підкріпленням.

Наукова новизна роботи полягає у розробці нового методу прийняття рішень, який поєднує байесівське оновлення репутацій з логарифмічно-лінійною адаптацією стратегій, забезпечуючи баланс між дослідженням нових дій та експлуатацією вже ефективних.

Практична значимість отриманих результатів полягає у створенні програмної реалізації симуляційного середовища на Python, що дозволяє моделювати поведінку агентів, перевіряти стабільність системи та оптимізувати її параметри в умовах частково спостережуваного середовища.

Для розв'язання поставлених завдань використовувалися методи багатокритеріальної оптимізації, теорії ігор, байесівської статистики та навчання з підкріпленням, що в сукупності дозволило досягнути високої гнучкості та ефективності моделі.

1 АНАЛІЗ СТРАТЕГІЙ ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ ОБЧИСЛОВАЛЬНИХ СИСТЕМАХ

1.1 Поняття про стратегії прийняття рішень у системах

Взаємозалежні обчислювальні системи (ВОС) – це такі системи, у яких обчислювальні компоненти взаємодіють і координують свої дії для виконання спільної задачі [1]. Основною характеристикою цих систем є високий рівень автономії кожного компонента, що передбачає можливість обміну інформацією та ресурсами через спеціальні інтерфейси [2]. Важливими аспектами функціонування ВОС є їхня розподіленість, адаптивність і ефективність взаємодії, адже кожен компонент працює на основі локальних даних, які узгоджуються з іншими вузлами [3].

Прикладом таких систем є хмарні обчислення, де обчислювальні ресурси розподілені між численними дата-центрами, які координують свої дії для виконання запитів користувачів [4]. Наприклад, у сервісах Amazon Web Services чи Google Cloud Platform різні сервери об'єднують свої потужності для обробки даних або виконання складних аналітичних запитів [5]. Іншим прикладом є інтернет речей (IoT), де сенсори у розумних будинках координують роботу пристроїв, наприклад, регулюють температуру чи вмикають освітлення залежно від присутності людини [6]. У масштабах розумного міста такі системи можуть керувати якістю повітря, дорожнім рухом або вуличним освітленням [7]. Ще одним варіантом ВОС є мультиагентні системи, що застосовуються в логістиці та промисловості [8]. Наприклад, автономні роботи-кур'єри здатні планувати свої маршрути, враховуючи дорожню ситуацію і взаємодіючи з іншими агентами для уникнення колізій [9]. На виробничих лініях роботи працюють узгоджено, виконуючи окремі завдання, що забезпечує ефективність всього виробничого процесу [10].

Принципи функціонування взаємозалежних обчислювальних систем передбачають кілька ключових аспектів [11]. Розподіленість забезпечує ефективний розподіл обчислювальних ресурсів та завдань між численними

вузлами, які можуть бути географічно віддаленими [12]. Це сприяє масштабованості, відмовостійкості та підвищенню швидкості виконання завдань [13]. Наприклад, у стримінгових сервісах на кшталт Netflix контент розподіляється між серверами по всьому світу, що гарантує швидкий доступ до відео незалежно від місця перебування користувача [14].

Автономність є ще одним важливим принципом, оскільки кожен компонент функціонує незалежно, приймаючи рішення на основі власних локальних даних [15]. Такий підхід підвищує стійкість системи до збоїв і дозволяє їй адаптуватися до змін без централізованого управління [16]. Прикладом є автономні транспортні системи, такі як Waymo, де кожен автомобіль використовує свої сенсори та камери для аналізу дорожньої ситуації й прийняття рішень, взаємодіючи з іншими транспортними засобами та інфраструктурою [17].

Не менш важливою є взаємодія між компонентами, яка передбачає обмін даними для узгодженого функціонування системи [18]. Ефективна взаємодія включає як швидку передачу інформації, так і її правильну обробку, що сприяє узгодженості дій усієї мережі [19]. У розподілених базах даних, таких як Apache Cassandra, вузли синхронізують інформацію, забезпечуючи її цілісність [20]. Також у системах авіаперевезень диспетчери використовують автоматизовані інструменти для координації розкладу рейсів, що гарантує безперебійне управління повітряним трафіком [21].

Останнім принципом є адаптивність, яка дозволяє системі змінюватися у відповідь на зміни середовища, такі як зростання навантаження чи вихід з ладу окремих вузлів [22]. Це підвищує стійкість системи та забезпечує оптимальне використання ресурсів у реальному часі [23]. Наприклад, у розумних енергомережах Smart Grid розподіл електроенергії динамічно змінюється залежно від рівня споживання та доступності відновлюваних джерел енергії [24]. У хмарних обчисленнях механізм автоматичного масштабування серверів реагує на зміну кількості запитів, оптимізуючи роботу всієї інфраструктури [25].

Засоби прийняття рішень у взаємозалежних обчислювальних системах охоплюють різні методи та підходи, серед яких важливу роль відіграє моделювання

агентів [26]. У мультиагентних системах кожен агент функціонує автономно, спираючись на локальну інформацію, проте може взаємодіяти з іншими агентами для досягнення спільної мети [27]. Алгоритми прийняття локальних рішень та обміну даними дозволяють агентам адаптувати свою поведінку в реальному часі, що особливо корисно у транспортних системах, де автономні автобуси оптимізують свої маршрути, враховуючи щільність пасажирів і стан доріг [28]. Подібний підхід використовується у складських системах, де роботи взаємодіють між собою для швидкої та ефективної обробки товарів [29]. У сфері сільського господарства агентні системи автоматизують процеси поливу й удобрення рослин, що підвищує врожайність і зменшує витрати [30]. Завдяки можливостям самонавчання ці системи стають дедалі ефективнішими, адаптуючись до нових умов і викликів [31].

Теорія ігор є ще одним засобом моделювання процесів прийняття рішень, особливо у конкурентних середовищах, де кілька агентів мають протилежні інтереси [32]. В її основі лежить пошук рівноваги між діями учасників системи, що визначається поняттями гравців, стратегій, виграшу та рівноваги Неша [33]. У рамках кооперативних ігор агенти можуть співпрацювати для оптимального розподілу задач, як це відбувається у розподілених обчисленнях, де вузли координують свої ресурси, щоб мінімізувати час виконання завдань [34]. Неконкурентні ігри використовуються для вирішення проблем у середовищах з обмеженими ресурсами, зокрема під час маршрутизації мережевого трафіку [35]. Динамічні ігри є ефективним інструментом для адаптивних систем, таких як Smart Grid, де рішення приймаються поетапно з урахуванням попередніх дій [36]. Наприклад, у мережевій маршрутизації кожен вузол коригує свою стратегію для оптимальної передачі даних, а у фінансових системах алгоритми торгівлі враховують дії конкурентів для максимізації прибутку [37].

Методи оптимізації дозволяють знаходити найкращі рішення у складних задачах, що включають велику кількість змінних і обмежень [38]. До традиційних підходів належать лінійне, нелінійне та цілочисельне програмування, тоді як еволюційні алгоритми, зокрема генетичні алгоритми, алгоритми рою часток і

мурашині алгоритми, імітують природні процеси для поступового покращення рішень [39]. У хмарних інфраструктурах, таких як Google Cloud, методи лінійного програмування використовуються для балансування навантаження між серверами, запобігаючи їх перевантаженню та забезпечуючи рівномірний розподіл обчислювальних ресурсів [40]. Подібні алгоритми застосовуються у виробничих лініях для оптимізації споживання енергії та зниження витрат без втрати продуктивності [41]. Оптимізаційні методи також знаходять застосування в логістиці, допомагаючи компаніям визначати найефективніші маршрути доставки з урахуванням вартості транспортування та часу виконання замовлень [42].

Штучний інтелект відіграє ключову роль у прийнятті рішень у взаємозалежних обчислювальних системах, забезпечуючи аналіз великих обсягів даних, прогнозування поведінки та автоматизацію процесів [43]. Використання методів машинного навчання, нейронних мереж і аналізу даних дозволяє системам адаптуватися до змін у середовищі [44]. Наприклад, у кібербезпеці алгоритми ШІ аналізують мережевий трафік для виявлення аномалій та запобігання потенційним атакам у реальному часі [45]. У сільському господарстві розумні системи збирають дані про погодні умови та стан ґрунту, щоб автоматично визначати оптимальні обсяги поливу й удобрення [46]. Медичні технології також активно використовують ШІ для діагностики захворювань та аналізу зображень, що дозволяє виявляти патології на ранніх стадіях [47]. У фінансовому секторі алгоритми штучного інтелекту прогнозують ринкові тенденції, допомагаючи інвесторам приймати ефективні рішення [48]. Таким чином, використання ШІ сприяє покращенню точності, швидкості та адаптивності процесів прийняття рішень у різних галузях, підвищуючи загальну продуктивність систем [49].

1.2 Аналіз методів оптимізації при прийнятті рішень

Лінійне програмування є одним із ключових методів математичної оптимізації, який широко використовується для вирішення задач оптимального розподілу ресурсів [50]. Його основна мета полягає в знаходженні найкращого

можливого рішення, яке максимізує або мінімізує певний показник за наявності встановлених обмежень [51]. Цей метод застосовується в різних сферах, зокрема у виробництві, економіці, логістиці, фінансах та управлінні проектами, оскільки дозволяє ефективно розподіляти наявні ресурси та приймати обґрунтовані управлінські рішення [52].

Ключовим елементом лінійного програмування є цільова функція, яка визначає критерій оптимізації [53]. Наприклад, у бізнесі це може бути максимізація прибутку або мінімізація витрат, а у виробничих процесах – зменшення часу виконання завдань або мінімізація використання матеріалів [54]. Ця функція виражається у вигляді математичного рівняння, де змінні відображають величини, що підлягають оптимізації [55].

Однак на практиці прийняття рішень у взаємозалежних системах супроводжується різноманітними обмеженнями [56]. Вони можуть включати обмеження на доступність ресурсів, наприклад, максимальний бюджет, ліміт робочого часу або потужність виробничих ліній [57]. Також можуть існувати логістичні обмеження, такі як обмеження на транспортування товарів або вимоги щодо послідовності виконання виробничих операцій [58]. Важливо, що всі обмеження у задачах лінійного програмування виражаються у вигляді рівнянь або нерівностей, які визначають допустиму область рішень [59].

Щоб знайти оптимальне рішення, використовуються спеціалізовані алгоритми, зокрема симплекс-метод, який дозволяє ефективно обчислити найкраще значення цільової функції в межах заданих обмежень [60]. Цей метод працює шляхом поступового покращення рішення, рухаючись по гранях допустимої області, аж поки не буде досягнуто оптимального значення [61]. У випадках, коли задача має велику кількість змінних і обмежень, можуть застосовуватися й інші підходи, такі як методи внутрішньої точки або евристичні алгоритми [62].

Практичне застосування лінійного програмування є надзвичайно широким [63]. Наприклад, у виробництві цей метод допомагає визначати оптимальне завантаження обладнання, розподіл робочої сили або розподіл сировини між

різними виробничими процесами [64]. У транспортній логістиці лінійне програмування використовується для побудови оптимальних маршрутів доставки, щоб мінімізувати витрати на перевезення та зменшити час доставки товарів [65]. В економіці та фінансах воно може допомагати у формуванні інвестиційних портфельів, балансуванні ризиків і максимізації прибутків від інвестицій [66].

Розглянемо конкретний приклад використання лінійного програмування. Припустимо, що фабрика виробляє два типи продукції, кожен із яких потребує певної кількості часу на обробку та певного обсягу сировини [67]. Однак доступні ресурси обмежені, і керівництво фабрики має вирішити, яку кількість кожного виду продукції слід виготовити, щоб максимізувати загальний прибуток [68]. У такому випадку лінійне програмування дозволяє створити математичну модель, у якій цільова функція визначає максимізацію прибутку, а обмеження враховують доступність ресурсів [69].

Основні переваги лінійного програмування включають чіткість, структурованість підходу до вирішення складних задач, можливість знаходження оптимального рішення в умовах обмежених ресурсів, а також гнучкість у застосуванні до різних галузей. Цей метод є ефективним інструментом для підтримки прийняття рішень у складних взаємозалежних системах, дозволяючи досягати високої ефективності в управлінні процесами та ресурсами [70].

Нелінійне програмування – це метод математичної оптимізації, що використовується для вирішення складних задач, у яких цільова функція або обмеження мають нелінійний характер. На відміну від лінійного програмування, цей підхід дозволяє працювати з моделями, які більш точно описують реальні системи, включаючи взаємозалежності між параметрами [71].

Нелінійне програмування є методом математичної оптимізації, який використовується для вирішення задач, у яких як цільова функція, так і обмеження можуть мати нелінійну форму [72]. Це дозволяє моделювати реальні процеси з урахуванням складних взаємозв'язків між змінними, що робить цей метод особливо корисним у випадках, коли лінійні моделі не можуть точно відобразити всі особливості системи.

Основним елементом нелінійного програмування є цільова функція, яка визначає критерій оптимізації, наприклад, мінімізацію витрат або максимізацію продуктивності [73]. На відміну від лінійного програмування, де залежність між змінними виражається у вигляді лінійних рівнянь, у випадку нелінійного програмування ця залежність є більш складною. Це дозволяє точніше моделювати процеси, в яких, наприклад, зростання обсягу виробництва не завжди призводить до пропорційного збільшення витрат або коли ефективність використання ресурсів змінюється залежно від їхнього рівня [74].

Окрім цільової функції, у нелінійному програмуванні важливу роль відіграють обмеження, які також можуть мати нелінійну форму. Вони встановлюють певні рамки для змінних та можуть описувати залежності, які відображають реальні процеси [75]. Наприклад, у виробництві обмеження можуть враховувати нелінійну залежність між кількістю використаних ресурсів та швидкістю виконання робіт, а у фінансовому аналізі – залежність прибутковості від ризиків. Крім того, часто враховуються додаткові умови для змінних, такі як їхня невід'ємність або максимальні та мінімальні допустимі значення. Це необхідно для того, щоб результати оптимізації відповідали реальним умовам та могли бути впроваджені на практиці [76].

Розв'язання задач нелінійного програмування передбачає використання спеціалізованих алгоритмів, оскільки класичні методи, які ефективні у випадку лінійного програмування, тут не працюють. Одним із найпоширеніших підходів є метод градієнтного спуску, який поступово коригує значення змінних у напрямку покращення цільової функції. Однак цей метод не завжди дозволяє знайти глобально оптимальне рішення, оскільки може зупинитися на локальному оптимумі. Для подолання цієї проблеми застосовуються більш складні алгоритми, такі як метод внутрішніх точок або методи пошуку оптимальних напрямків, які дозволяють розглядати більший простір можливих рішень [77].

Прикладом практичного застосування нелінійного програмування може бути система управління запасами на складі, де витрати на зберігання та доставку залежать від кількості товарів. У такій задачі залежність між змінними є

нелінійною, оскільки збільшення запасів може призводити до непропорційного зростання витрат через необхідність додаткових складських приміщень або збільшення логістичних витрат [78]. Використання нелінійного програмування дозволяє знайти оптимальну стратегію управління запасами, яка мінімізує витрати та водночас забезпечує необхідний рівень обслуговування клієнтів.

Цей метод має низку переваг, серед яких можливість вирішення задач, що не піддаються лінійному моделюванню, врахування складних залежностей між змінними та адаптація до реальних умов функціонування системи. Він застосовується в різних сферах, зокрема в енергетиці для оптимізації розподілу енергоресурсів у складних мережах, у фінансах для управління інвестиційними портфелями з урахуванням ризиків та очікуваних доходів, у технологічних процесах для розробки ефективних виробничих систем, а також у біології для моделювання складних біохімічних реакцій [79].

Водночас нелінійне програмування має і певні недоліки. Одним із них є висока обчислювальна складність, що ускладнює розв'язання задач із великою кількістю змінних та обмежень. Також існує ризик отримання лише локально оптимального рішення, а не глобально найкращого варіанту. Крім того, для ефективного застосування цього методу необхідно мати точний математичний опис системи та якісні вхідні дані, що не завжди є можливим у реальних умовах.

Попри ці труднощі, нелінійне програмування залишається одним із найпотужніших інструментів для розв'язання складних оптимізаційних задач, які враховують реальні обмеження та складну природу взаємозв'язків між змінними [80]. Його застосування дозволяє підвищити ефективність процесів, зменшити витрати та приймати більш обґрунтовані рішення у багатьох сферах науки та промисловості.

Генетичні алгоритми є ефективним методом оптимізації, який ґрунтується на ідеях природного відбору та еволюції. Вони застосовуються для розв'язання складних задач із великою кількістю змінних, високим рівнем невизначеності або нелінійними залежностями між параметрами, де традиційні математичні підходи, такі як лінійне чи нелінійне програмування, втрачають свою ефективність. Основна

ідея методу полягає у використанні механізмів, подібних до тих, що спостерігаються в природі: популяції можливих рішень, процесу природного відбору, схрещування характеристик найкращих рішень та випадкових мутацій, що дозволяють знайти оптимальний або наближений розв'язок задачі.

Процес починається зі створення початкової популяції, що складається з множини можливих рішень, які в алгоритмі називаються особинами або хромосомами. Кожна хромосома кодує певний варіант розв'язку, який може бути представлений у вигляді числових значень, бітових рядків чи інших структур. Далі кожна особина оцінюється за допомогою спеціальної функції пристосованості, що визначає, наскільки ефективно дане рішення відповідає поставленій меті.

Щоб з часом отримувати кращі розв'язки, алгоритм застосовує ключові оператори еволюції. На першому етапі проводиться селекція, під час якої вибираються найкращі особини для створення нового покоління. Це дозволяє зберігати перспективні рішення та поступово покращувати їх. Далі використовується оператор кросовера, що імітує генетичне схрещування та передбачає комбінування характеристик двох «батьків» для створення «нащадків», які можуть мати нові, потенційно кращі властивості. Крім того, вводиться мутація — випадкові зміни в окремих елементах хромосоми, що допомагає уникати зациклення на локальних оптимумах і сприяє дослідженню ширшого простору можливих рішень.

Процес генерації нових популяцій повторюється багаторазово. Кожне наступне покоління зазвичай містить більш пристосовані особини, а алгоритм поступово наближається до оптимального розв'язку. Він може працювати доти, доки не буде знайдено прийнятне рішення, не вичерпаються ресурси або не буде досягнуто заданої кількості ітерацій.

Генетичні алгоритми знаходять широке застосування у різних сферах, де класичні методи можуть бути малоефективними. Наприклад, у задачах маршрутизації вантажного транспорту, де необхідно мінімізувати витрати пального та часу, вони дозволяють знайти оптимальні маршрути шляхом моделювання множини можливих комбінацій та відбору найкращих варіантів. В

інженерії їх використовують для оптимізації конструкцій, дизайну механізмів та автоматизованого проектування. У сфері фінансів цей підхід допомагає створювати ефективні інвестиційні стратегії та знаходити оптимальні комбінації активів із урахуванням ризиків і потенційної прибутковості. В ІТ-галузі генетичні алгоритми застосовуються для розв'язання задач розподілу ресурсів, машинного навчання та штучного інтелекту.

Головною перевагою цього підходу є здатність працювати із задачами, де пошук оптимального рішення ускладнений через велику кількість змінних та взаємозалежних параметрів. Генетичні алгоритми можуть одночасно досліджувати багато варіантів, що значно підвищує шанси знайти ефективне рішення, навіть якщо воно не є глобально оптимальним. Вони особливо корисні в задачах, які неможливо аналітично розв'язати через складну структуру обмежень або відсутність чітко визначеного алгоритму пошуку.

Однак цей метод має і свої недоліки. Основною проблемою є висока обчислювальна складність, оскільки велика кількість ітерацій та операцій над популяціями може вимагати значних обчислювальних ресурсів. Також немає гарантії, що алгоритм знайде саме глобально оптимальне рішення, а не зупиниться на локальному максимумі або мінімумі. Крім того, для ефективної роботи алгоритму потрібно ретельно налаштовувати його параметри, такі як розмір популяції, ймовірності кросовера та мутації, що може потребувати експериментального підходу та додаткових обчислень.

Незважаючи на ці труднощі, генетичні алгоритми залишаються одним із найефективніших інструментів для оптимізації складних багатофакторних задач. Їхня здатність адаптуватися до змінних умов, досліджувати широкий простір можливих рішень та імітувати природні процеси робить їх надзвичайно потужним інструментом для розв'язання проблем у науці, техніці, економіці та інформаційних технологіях.

Динамічне програмування є потужним методом розв'язання задач, які мають багатокрокову природу та можуть бути розбиті на підзадачі. Його основна ідея полягає у збереженні проміжних результатів, що дозволяє уникати повторного

обчислення та значно підвищує ефективність алгоритмів. Цей підхід застосовується для знаходження оптимальних стратегій шляхом поступового розв'язання менших підзадач, результати яких використовуються для отримання кінцевого рішення. Динамічне програмування базується на кількох ключових принципах. Перший із них – оптимальність підструктури, що означає, що оптимальне рішення всієї задачі можна отримати, якщо знайти оптимальні рішення її складових частин. Другий принцип – перекриття підзадач, коли однакові проміжні розрахунки повторюються в ході виконання алгоритму, і замість їх повторного обчислення використовується збережена інформація. Третій важливий аспект – мемоїзація, тобто збереження проміжних результатів у спеціальній структурі даних, зазвичай у вигляді масиву або таблиці, що значно зменшує обчислювальні витрати.

Процес застосування динамічного програмування починається з розбиття вихідної задачі на підзадачі, які є простішими для розв'язання. Далі створюється структура даних для збереження проміжних результатів, що дозволяє алгоритму працювати ефективніше. Обчислення зазвичай починаються з найменших підзадач, поступово переходячи до складніших, використовуючи раніше збережені результати. Таким чином, після виконання всіх ітерацій отримане рішення для найбільшої підзадачі стає розв'язанням вихідної проблеми.

Щоб зрозуміти переваги цього підходу, розглянемо задачу комівояжера, в якій необхідно відвідати всі міста за мінімальний час або з мінімальними витратами. Традиційні методи пошуку рішення можуть бути надзвичайно ресурсозатратними через експоненційне зростання кількості можливих маршрутів. Динамічне програмування дозволяє спростити цю задачу, використовуючи збережені результати для менших маршрутів і поступово нарощуючи оптимальні розв'язки для більш складних випадків.

Завдяки такому підходу динамічне програмування має ряд суттєвих переваг. По-перше, воно дозволяє значно знизити обчислювальну складність шляхом усунення дублювання підрахунків, що підвищує ефективність алгоритмів. По-друге, цей метод є гнучким і застосовним до різних типів задач – від простих

рекурсивних обчислень до складних багатофакторних систем. По-третє, динамічне програмування гарантує знаходження оптимального розв'язку в межах заданих умов, що робить його особливо корисним для задач оптимізації.

Однак цей метод має й певні недоліки. Один із ключових викликів – високі вимоги до пам'яті, оскільки для збереження проміжних результатів може знадобитися значний обсяг ресурсів. Також масштабованість алгоритмів може стати проблемою, оскільки зі збільшенням розміру задачі складність побудови відповідного алгоритму може зростати.

Сфери застосування динамічного програмування надзвичайно широкі. У плануванні та управлінні цей метод використовується для розподілу ресурсів, складання розкладів і оптимізації виробничих процесів. У транспортних і комунікаційних мережах він допомагає знаходити найкоротші маршрути, що особливо важливо для логістичних компаній та операторів мобільного зв'язку. В інформатиці динамічне програмування застосовується в алгоритмах роботи з рядками, таких як пошук підрядків або вирівнювання текстів у системах обробки інформації. В економіці цей метод допомагає моделювати багатостадійні процеси прийняття рішень, наприклад, у задачах оптимізації витрат та управління інвестиціями.

Окрім динамічного програмування, існують й інші підходи до оптимізації та прийняття рішень у складних системах. Наприклад, алгоритми рою, які базуються на спостереженні за поведінкою колоній комах, використовуються для координації автономних систем і розв'язання задач оптимізації. Такі алгоритми особливо ефективні в розподілених середовищах, де потрібно синхронізувати дії великої кількості незалежних агентів. Один із прикладів – оптимізація мережевої маршрутизації, де моделювання поведінки мурах дозволяє знаходити найкоротші шляхи між вузлами. Інший приклад – керування групою автономних роботів у складських приміщеннях, що допомагає оптимально розподіляти їх завдання.

Методи машинного навчання також відіграють важливу роль у прийнятті рішень і можуть бути ефективно поєднані з динамічним програмуванням. Вони застосовуються для аналізу великих обсягів даних і автоматичної адаптації

алгоритмів під змінні умови. Наприклад, у хмарних обчислювальних системах машинне навчання використовується для прогнозування навантаження на сервери та динамічного розподілу ресурсів. У фінансовій сфері ці методи допомагають оцінювати кредитні ризики та формувати оптимальні стратегії інвестування на основі історичних даних.

Динамічне програмування залишається одним із ключових методів розв'язання оптимізаційних задач, особливо коли необхідно враховувати численні взаємопов'язані підзадачі. Його здатність використовувати накопичені результати для пришвидшення обчислень робить його надзвичайно ефективним інструментом у широкому спектрі галузей – від алгоритміки та штучного інтелекту до економіки та логістики. Разом із сучасними підходами, такими як алгоритми рою та машинне навчання, цей метод дозволяє розробляти високоефективні стратегії управління складними системами, підвищуючи їх продуктивність, надійність і гнучкість.

1.3 Постановка задачі

Для досягнення поставленої мети вирішуються наступні завдання:

- проаналізувати сучасні підходи до прийняття рішень у багатоагентних обчислювальних системах;
- формалізувати модель взаємозалежностей між агентами з урахуванням репутаційних механізмів;
- розробити гібридний метод прийняття рішень, який враховує адаптацію стратегій на основі навчання з підкріпленням;
- здійснити алгоритмічну реалізацію методу та провести експериментальне моделювання динаміки системи;
- оцінити ефективність методу за критеріями адаптивності, стійкості до саботажу та швидкості досягнення стратегічної рівноваги.

1.4 Висновки до першого розділу

Отже, проведено всебічний аналіз стратегій прийняття рішень у взаємозалежних обчислювальних системах. Визначено ключові характеристики таких систем, зокрема їхню автономність, адаптивність, розподіленість і здатність до ефективної взаємодії між компонентами. Розглянуто різноманітні приклади застосування ВОС у хмарних обчисленнях, IoT, транспорті, промисловості та розумних інфраструктурах.

Окрему увагу приділено методам прийняття рішень: агентному моделюванню, теорії ігор, методам оптимізації, машинному навчанню та штучному інтелекту. Показано, що поєднання цих підходів дозволяє системам адаптуватися до змін у середовищі, ефективно розподіляти ресурси та приймати обґрунтовані рішення навіть за умов високої складності та невизначеності. Особливо виділено роль лінійного і нелінійного програмування, генетичних алгоритмів, динамічного програмування та комбінованих підходів у розробці сучасних стратегій управління у ВОС.

2 МЕТОД ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ СИСТЕМАХ

Оскільки штучний інтелект (ШІ) продовжує розвиватися, концепція автономності, яка включає здатність діяти та мислити на основі особистого досвіду та оцінок, відкриває нові перспективи для автономного прийняття рішень. Це особливо актуально для взаємопов'язаних обчислювальних систем, де взаємодія між компонентами часто має непередбачувані й нестабільні наслідки. Важливим завданням є проектування такої автономії, керованої ШІ, яка б ефективно функціонувала в різних сферах, зокрема в Інтернеті речей (IoT) та кібербезпеці. Досягти цього можна шляхом моделювання людського процесу ухвалення рішень, що включає дослідження невизначеного та стохастичного середовища, здійснення дій і подальший аналіз отриманого зворотного зв'язку.

Кожна людина суб'єктивно оцінює зворотний зв'язок як прийнятний або ні, спираючись на власний поведінковий профіль та логіку міркувань. Повторення цього циклу кроків формує процес навчання. Таким чином, головна ідея полягає у впровадженні людських знань у взаємопов'язані обчислювальні системи, щоб вони могли виступати як агенти прийняття рішень, керовані штучним інтелектом. Такі агенти наслідують раціональні поведінкові моделі людини та автономно оптимізують власні критерії оцінки. Зі стрімким розвитком взаємозалежних обчислювальних систем, зокрема безпілотних літальних апаратів (БПЛА) і серверів периферійних обчислень з багатодоступом (Multi-Access Edge Computing, MEC), відбувається зростання обсягів даних і підвищуються вимоги до якості обслуговування (ЯО). Під час автономного функціонування такі системи демонструють конкурентну поведінку, оскільки кожна з них прагне максимізувати власні цілі. Це формує концепцію інтерактивного прийняття рішень у середовищах без кооперації, де результати для кожної системи визначаються взаємозалежними та потенційно конфліктними діями інших учасників.

У цій роботі розглядається теорія ігор, яка дозволяє моделювати стратегічні взаємодії між взаємозалежними обчислювальними системами в некооперативному

середовищі, доводячи існування стабільних точок рівноваги. Такі рівноваги вважаються оптимальними рішеннями, оскільки жодна система не має стимулу змінювати свою стратегію в односторонньому порядку. Для визначення цих точок рівноваги у розподіленій формі застосовується навчання з підкріпленням (НП), що дає змогу автономним обчислювальним системам адаптуватися до стохастичного середовища, навчаючись методом проб і помилок на основі отриманого зворотного зв'язку.

Крім того, класичні підходи НП доповнюються технікою модифікації функції винагороди, що дозволяє враховувати механізми, подібні до економіки праці, через використання теорії контрактів. Поєднання теорії ігор та навчання з підкріпленням із динамічною зміною винагороди є важливим кроком до розвитку самосвідомого штучного інтелекту (СШІ).

2.1 Елементи теорії ігор та штучного інтелекту при прийнятті рішень

Теорія ігор, запропонована Джоном фон Нейманом і Оскаром Моргенштерном у праці «Теорія ігор та економічна поведінка», вивчає математичні моделі стратегічної взаємодії між раціональними агентами, що приймають рішення з урахуванням власних інтересів. Основним методом представлення егоїстичних мотивів таких агентів є теорія корисності, яка формально описує їхні переваги щодо різних стратегій та демонструє, як ці переваги змінюються за умов невизначеності щодо вибору конкретної стратегії.

Таким чином, функція корисності виступає як відображення можливих станів стохастичного середовища у числові значення, які характеризують рівень задоволеності агента від виконання певної дії за заданих умов. Основна мета полягає в оптимізації цієї функції, тобто її максимізації або мінімізації. Однак, коли середовище містить більше двох агентів, які прагнуть оптимізувати власну корисність і чий дії взаємопов'язані, система стає значно складнішою. У таких випадках ефективним інструментом моделювання є некооперативна теорія ігор.

Нормальна форма, також відома як стратегічна або матрична форма, є найбільш поширеним способом математичного представлення стратегічної взаємодії між раціональними агентами в некооперативній теорії ігор. Це пояснюється тим, що будь-яка інша теоретико-ігрова модель може бути перетворена у формат нормальної форми. Формалізоване моделювання таких взаємодій у межах теорії ігор називається грою G .

Варто зазначити, що поняття стратегії в іграх нормальної форми відрізняється від поняття окремої дії. Зокрема, одна з категорій стратегій передбачає вибір і виконання конкретної дії, що називається чистою стратегією. набір чистих стратегій усіх агентів утворює профіль чистих стратегій. Інший тип стратегій – змішані, коли агент може випадковим чином обирати дії відповідно до визначеного розподілу ймовірностей.

Основне завдання теорії ігор полягає у визначенні оптимальної стратегії для агентів, які приймають рішення. Наприклад, якщо в грі бере участь лише один агент, то оптимальною вважається така стратегія, яка максимізує його функцію корисності в заданому середовищі. Проте на практиці ситуація значно складніша, оскільки середовище може бути стохастичним і лише частково спостережуваним. У випадку мультиагентної взаємодії складність зростає ще більше, оскільки дії одного агента безпосередньо впливають на рішення інших.

Таким чином, необхідно чітко визначити поняття «оптимальності» в контексті теорії ігор, де кілька агентів взаємодіють між собою та можуть мати суперечливі інтереси. Основна проблема полягає у знаходженні підходу, який дозволяє довести, що певний профіль стратегій є оптимальним, навіть за умов неповної інформації про функції корисності всіх учасників гри.

У кожній грі існує принаймні одне оптимальне рішення, і завжди можна знайти такий оптимальний варіант, у якому всі агенти зводять свої вибори до чистих стратегій. Крім того, в деяких типах ігор, наприклад у тих, що мають нульову суму, може існувати кілька рівноцінних оптимальних розв'язків.

Навчання з підкріпленням (НП) є однією з найперспективніших галузей штучного інтелекту, зокрема машинного навчання (МН), що базується на

цілеспрямованому навчанні через взаємодію з середовищем. Процес навчання людини також тісно пов'язаний із взаємодією з оточенням, наприклад, під час освоєння ходьби, мовлення чи соціальної поведінки, що генерує значний обсяг причинно-наслідкових даних.

Застосування НП у різних сферах, таких як електронна комерція, настільна гра «Go» та навіть охорона здоров'я, підтверджує його значний потенціал для розробки інтелектуальних обчислювальних систем, здатних ефективно розв'язувати практичні завдання. НП пропонує математичний формалізм для послідовного прийняття рішень, створюючи агентів штучного інтелекту, які набувають майже оптимальних поведінкових навичок. Це досягається шляхом розробки політик, спрямованих на оптимізацію функції винагороди в стохастичних середовищах.

НП розглядається як третій піддомен машинного навчання поряд із навчанням під наглядом і без нагляду. Відмінність НП від навчання під наглядом полягає в тому, що останнє використовує позначені дані для навчання та узагальнення на нові, подібні приклади. У НП отримання міток для всіх можливих ситуацій, з якими може зіткнутися агент, є неефективним, тому взаємодія з середовищем без попередньо визначених міток виглядає більш доцільною. Водночас, на відміну від неконтрольованого навчання, яке лише виявляє приховані закономірності в немаркованих даних, НП спрямоване на максимізацію цільової функції винагороди, що робить його більш придатним для прийняття рішень у складних динамічних середовищах.

Однією з основних проблем, з якою навчання з підкріпленням (НП) стикалося протягом багатьох років, було забезпечення здатності агентів ефективно навчатися на високовимірних сенсорних входних даних, таких як аудіо або зображення. Ранні дослідження у цій сфері зосереджувалися на використанні вручну створених представлень ознак у поєднанні з лінійними функціями значень. Однак продуктивність та ефективність навчання НП-агентів значною мірою залежала від якості цих функцій, що створювало певні обмеження.

Глибоке навчання викликало значний інтерес завдяки своїй здатності самостійно виявляти складні закономірності у необроблених даних без безпосереднього втручання людини. Завдяки доступності великих і різноманітних наборів даних стало можливим навчання глибоких нейронних мереж (ГНМ), таких як багат шарові перцептрони, згорткові та рекурентні нейронні мережі, що демонструють високу здатність до узагальнення в реальних умовах.

Успіхи глибокого навчання в наукових дослідженнях сприяли його інтеграції з НП, що призвело до появи глибокого навчання з підкріпленням (ГНП). Цей підхід виявився ефективним у розв'язанні складних багатовимірних задач послідовного прийняття рішень. ГНП знайшов широке застосування у реальних сферах, таких як кібербезпека, захист конфіденційності, управління розподілом електроенергії, рекомендаційні системи та оптимізація бездротових мереж.

Основною причиною застосування ГНП є те, що більшість стохастичних середовищ у реальному світі мають високимірний характер, тобто характеризуються великими просторами можливих станів та дій, що ускладнює використання традиційних методів НП.

Метод навчання з підкріпленням (НП) спочатку застосовувався в контексті марковських процесів прийняття рішень (МППР) і дозволяв агенту розробити політику, спрямовану на максимізацію майбутньої дисконтованої сукупної винагороди в стохастичному середовищі. Проте у випадках, коли у спільному середовищі взаємодіють кілька агентів НП, які конкурують за обмежені ресурси, класичний підхід НП стає малоефективним. Це зумовлено тим, що оптимальна стратегія окремого агента залежить не лише від невідомих параметрів середовища, а й від стратегій інших агентів.

У багатьох реальних сферах, таких як робототехніка, мережеві технології та інтелектуальні енергетичні системи, обчислювальні процеси зазвичай мають децентралізований характер. У таких випадках централізоване управління агентами НП є непрактичним через розподілену структуру даних і управління, а також можливі конфлікти між цілями окремих систем.

Обмеження, пов'язані з доступністю інформації у таких середовищах, можна класифікувати за трьома основними категоріями:

- неповнота: агент має доступ лише до власної інформації й не володіє даними про інших агентів;
- динамічність: стан середовища постійно змінюється, впливаючи на досвід агентів;
- невизначеність: спостереження агентів можуть бути зашумленими, що ускладнює точне визначення стану середовища.

З огляду на ці виклики, виникає необхідність у створенні автономних, взаємозалежних агентів, здатних адаптуватися до таких обмежень і приймати обґрунтовані рішення. Це можливо шляхом інтеграції навчання з підкріпленням і теорії ігор, що дозволяє агентам взаємодіяти в багатокomпонентному стохастичному середовищі та адаптувати свої стратегії відповідно до поведінки інших учасників.

Крім того, теорія ігор надає потужний математичний апарат для аналізу результатів процесу навчання агентів НП. Кінцевою метою такого навчання є досягнення рівноважного стану, коли жоден агент не має стимулу змінювати свою стратегію в односторонньому порядку. Це дослідницьке поле відоме як розподілене багатоагентне навчання з підкріпленням.

У цьому контексті кожен агент приймає рішення, вирішуючи послідовну задачу оптимізації власної корисності, адаптуючи свої дії відповідно до змін у середовищі. Отже, раніше розглянута методологія НП може бути застосована таким чином, щоб кожен агент діяв як автономна система НП, навчаючись оптимальній політиці, яка максимізує його дисконтовану сукупну винагороду.

Цей підхід лежить в основі запропонованої в роботі концепції: у випадку некооперативної гри G кожен агент отримує винагороду, що відповідає значенню його функції корисності, яке він сприймає як зворотний зв'язок від стохастичного середовища. Таким чином, відбувається інтеграція навчання з підкріпленням і теорії ігор, що сприяє розвитку більш ефективної парадигми розподіленого багатоагентного навчання з підкріпленням.

Навчання шляхом градієнтного підйому – це клас нецентралізованих стохастичних алгоритмів НП на основі політики, де кожен агент НП формує власну стратегію, представлену розподілом ймовірностей для доступних дій. Зокрема, агенти виступають у ролі навчальних автоматів (НА), тобто ітеративних механізмів оновлення політики, які підтримують вектор ймовірностей для кожної можливої дії та вдосконалюють його за допомогою градієнтного оновлення.

Основна ідея цих алгоритмів полягає в тому, що коли агент НП виконує дію, яка приносить високу винагороду, ймовірність вибору цієї дії в майбутньому зростає пропорційно отриманій винагороді. Такий підхід дозволяє агентам поступово адаптувати свою поведінку, орієнтуючись на максимізацію очікуваної винагороди.

Байєсівські навчальні автомати (БНА) – це алгоритм НП на основі політики, що відзначається низькою обчислювальною складністю та ґрунтується на принципах байєсівського міркування. Цей підхід вводить ймовірнісну модель для ефективного прийняття рішень, розглядаючи агента як кінцевий автомат, у якому кожен стан асоціюється з ймовірністю вибору певної дії.

Алгоритм БНА переважно застосовується до задач дворукого бандита з розподілом Бернуллі, де агенти стикаються лише з двома конфліктними варіантами дій. Для оновлення ймовірностей вибору дій БНА використовує бета-розподіл, який визначається двома позитивними параметрами α і β , що регулюють форму розподілу та впливають на процес навчання агента.

Логарифмічно-лінійне навчання (ЛЛН) – це клас розподілених алгоритмів НП низької складності на основі політики, які працюють за принципом, подібним до навчання методом градієнтного підйому (НМГП). Основні алгоритми цієї сімейства включають двійкове логарифм-лінійне навчання і максимальне логарифмічно-лінійне навчання.

У ЛЛН кожен агент підтримує розподіл ймовірностей щодо доступних дій, що визначає його поточну політику. Основна ідея алгоритмів полягає в наступному:

- на кожній ітерації випадково обирається агент НП, який має виконати дослідження та навчання;
- цей агент вибирає дію випадковим чином з рівною ймовірністю;
- решта агентів залишають свої дії незмінними.

Досліджуючий агент НП отримує відповідну винагороду на основі вибраної дії та оновлює свій розподіл ймовірностей, підлаштовуючи політику відповідно до отриманого зворотного зв'язку.

ЛЛН є ефективним для адаптації агентів у децентралізованих середовищах, забезпечуючи поступове вдосконалення стратегій у багатопрофільних взаємодіях.

Q-навчання без збереження стану – це спрощений варіант традиційного Q-навчання, який підходить для розподіленого багатоагентного НП і має низьку обчислювальну складність.

Основна ідея: кожен агент НП підтримує вектор оцінених значень Q для кожної доступної дії, але не враховує інформацію про стан. Це робить алгоритм придатним для сценаріїв із великими або невідомими просторами станів, а також для децентралізованих систем, де обмін станами між агентами є неможливим або небажаним.

Обмеження:

- відсутність інформації про стан може ускладнити навчання в складних середовищах;
- обмежені відомості про дії інших агентів, що може впливати на стабільність алгоритму;
- не завжди сходиться до стабільної точки: у деяких випадках агенти НП досягають рівноваги, але в інших – алгоритм може залишатися нестабільним або коливатися.

Попри ці обмеження, Q-навчання без збереження стану залишається цікавим підходом для багатопрофільного навчання в умовах обмежених ресурсів.

2.2 Теорія контрактів при прийнятті рішень

Швидке впровадження Інтернету речей (IoT) у різних сферах не лише сприяє покращенню комфорту в повсякденному житті, але й породжує значні виклики у сфері обчислювальних систем. Зокрема, безперервна взаємодія між пристроями IoT зумовлює необхідність у керованій кооперації, особливо у сферах, пов'язаних із краудсорсингом та розумними містами. До того ж, такі системи часто виявляють конкурентну поведінку, оскільки функціонують в умовах обмежених ресурсів (наприклад, рівня заряду батареї чи доступної обчислювальної потужності), що нерідко призводить до конфліктних ситуацій.

Такі конфлікти перешкоджають ефективній взаємодії між взаємозалежними обчислювальними системами у реальних застосуваннях, ускладнюючи досягнення стабільних рішень. Відповідно, з'являється потреба у розробці механізмів стимулювання, які б сприяли залученню ресурсозалежних обчислювальних систем до взаємодії в стохастичних середовищах, водночас удосконалюючи їхню операційну поведінку.

Теорія контрактів (ТК) – це економічний підхід, що отримав Нобелівське визнання та може використовуватися як механізм стимулювання в умовах інформаційної асиметрії. Ця теорія передбачає наявність роботодавця та кількох працівників, при цьому роботодавець не володіє повною інформацією про особливості поведінки працівників, зокрема щодо їхнього досвіду, рівня дисциплінованості та продуктивності.

Моделі контрактів дозволяють вирішувати проблему асиметрії інформації у стохастичних середовищах завдяки створенню контрактних угод, у яких винагорода залежить від продуктивності та витрачених зусиль працівників.

Основна мета ТК полягає у розробці привабливих контрактів, які б заохочували працівників, одночасно максимізуючи прибуток роботодавця. Цей процес можна змодельовати як оптимізацію функції вигоди роботодавця з урахуванням наступних обмежень:

- стимулююча сумісність гарантує, що для працівників найбільш вигідним буде прийняття запропонованого контракту, оскільки це максимізує їхній очікуваний прибуток;

- індивідуальна раціональність означає, що відмова від участі у взаємодії призведе до зменшення очікуваного доходу для працівника.

Оскільки ТК допомагає подолати проблему інформаційної асиметрії та ефективно стимулювати взаємодіючі сторони, ми інтегрували цей підхід у наші моделі для забезпечення продуктивної співпраці та розробки механізмів стимулювання серед автономних, інтелектуальних обчислювальних систем, що працюють у взаємозалежному середовищі.

Наприклад, у сфері бездротових мереж у ролі роботодавця може виступати безпілотний літальний апарат (БПЛА), що збирає інформацію про громадську безпеку від інтелектуальних пристроїв із обмеженими ресурсами, які в цій взаємодії виконують роль працівників.

Таким чином, запропоновані системи прийняття рішень дають змогу інтегрувати механізми стимулювання у функції винагороди НП-агентів. Це дозволяє агентам "переговорюватися" щодо власних зусиль та адаптувати свої дії задля отримання вищих винагород у майбутньому.

Перша група моделей теорії контрактів (ТК) ґрунтується на концепції несприятливого відбору, коли характеристика поведінки працівників (наприклад, автономних взаємозалежних обчислювальних систем), зокрема рівень їхньої компетентності, залишається невідомою для роботодавця (наприклад, БПЛА). У цій роботі особливу увагу приділено проблемі скринінгу, де контракт розробляє сторона, яка не володіє повною інформацією, тобто роботодавець, а працівники, маючи більш глибоке розуміння власних можливостей, обирають той контракт, що найбільше відповідає їхньому типу та поведінковому профілю. Це здійснюється завдяки принципу відкриття: роботодавець формує множину контрактів у вигляді пар $\{e, r\}$, що адаптовані до різних категорій працівників. Тут e означає рівень очікуваних зусиль з боку працівника, а r – розмір компенсації, що надається у випадку виконання цих зусиль. У реальному сценарії застосування бездротових

технологій у сфері громадської безпеки е може відображати обсяг важливих даних, які збираються IoT-пристроями в критичній зоні, а g – кількість енергії, яку БПЛА передає цим пристроям для підтримки їхньої роботи.

Друга категорія ТК пов'язана з проблемою морального ризику, де, на відміну від несприятливого відбору, інформаційна асиметрія виникає після укладення контракту між працівниками та роботодавцем. Зокрема, після підписання угоди працівники можуть приховувати фактичні витрачені зусилля, обумовлені контрактом $\{f, g\}$, де f – це рівень зусиль, який працівник зобов'язується докласти, а g – відповідна винагорода, яку надає роботодавець. Основна концепція полягає в тому, що компенсація g складається з фіксованої виплати та додаткового бонусу, який залежить від продуктивності. При цьому оцінка результативності працівника здійснюється відповідно до конкретного сценарію використання. У цьому контексті роботодавець отримує лише інформацію про кінцеву продуктивність працівника, однак не має безпосереднього доступу до фактичного рівня його зусиль f , оскільки продуктивність може виступати спотвореним відображенням реальної праці.

Працівники нерідко демонструють схильність до ризику, виявляючи готовність йти на певні небезпеки задля отримання більшої компенсації від роботодавця.

Репутація є ключовим поняттям, яке широко застосовується в економічній сфері. Водночас її інтеграція в дослідження реальних застосувань, зокрема у сфері ефективного управління обчислювальними ресурсами, набуває особливої важливості. Це дозволяє створювати автономні, взаємопов'язані та інтелектуальні агенти, які приймають рішення, враховуючи поведінкові характеристики один одного.

Основна концепція полягає в тому, що коли обчислювальні системи, які залежать одна від одної, регулярно взаємодіють у невизначеному середовищі, їхні поведінкові характеристики поступово формують відповідні рівні репутації. Ці профілі можуть слугувати основою для прийняття рішень інтелектуальними

обчислювальними агентами. У контексті репутації важливо те, що особи, які приймають рішення, формують уявлення про надійність інших учасників взаємодії.

Модель репутації, що базується на концепції несприятливого відбору та використовує правило байєсівського оновлення переконань, була застосована для аналізу поведінки агентів.

Автономні та взаємозалежні обчислювальні системи, що використовують штучний інтелект, розглядаються в контексті граничних обчислень із множинним доступом (МЕС) та мереж. Огляд МЕС висвітлює його додаткову функціональність у порівнянні з традиційною централізованою хмарною інфраструктурою та мобільними пристроями, а також підкреслює основні переваги цієї технології.

Окрім цього, аналізуються сучасні тенденції у наукових дослідженнях, що стосуються МЕС, та акцентується увага на ключових викликах, які потребують ефективних рішень. Для цього пропонуються різні системи прийняття рішень, що ґрунтуються на теоретичних концепціях і спрямовані на подолання зазначених проблем.

Також наголошується, що розроблені інфраструктури мають унікальні характеристики, які дозволяють максимально точно відобразити особливості МЕС-середовищ, а їх ефективність підтверджується шляхом моделювання та симуляційних експериментів.

За останні роки Інтернет еволюціонував від однорангових мереж до масштабної концепції Інтернету речей (IoT). Завдяки IoT мільярди розумних мобільних пристроїв можуть підключатися в будь-який час і в будь-якому місці, використовуючи доступні мережеві ресурси, що призводить до стрімкого зростання мобільного трафіку.

З точки зору кінцевих користувачів, сучасні бездротові мережі 5G забезпечують високоефективні архітектури мобільного зв'язку, які характеризуються збільшеною пропускну здатністю, мінімальною затримкою та зниженими експлуатаційними витратами. Це, своєю чергою, сприяє покращенню якості обслуговування (ЯО) та користувацького досвіду (ЯДК) під час передачі даних.

Подальший розвиток Інтернету передбачає створення тактильного Інтернету – екосистеми IoT, що потребує значних обсягів даних, потужних обчислювальних ресурсів і вдосконаленої мережевої інфраструктури. Це необхідно для ефективного управління величезною кількістю взаємопов'язаних IoT-пристроїв.

Хмарні технології надають можливість віддаленого зберігання та обробки даних IoT, передаючи їх довіреній третій стороні. Це дозволяє кінцевим користувачам, зокрема мобільним пристроям, позбутися необхідності самостійно управляти обробкою інформації.

Однак традиційні хмарні сервери побудовані на централізованій архітектурі, що має низку недоліків, зокрема ризик єдиної точки відмови, обмежену доступність, високу затримку тощо. Водночас більшість IoT-додатків вимагають децентралізованих рішень, здатних забезпечити високу пропускну здатність, низьку затримку та масштабованість.

Через це використання класичних централізованих хмарних серверів у сфері Інтернету речей стикається зі значними обмеженнями. Для того щоб відповідати сучасним вимогам, необхідно здійснити адаптацію існуючої хмарної архітектури та її інфраструктури.

Для вирішення цих питань Група галузевих специфікацій Європейського інституту телекомунікаційних стандартів (ETSI) представила концепцію граничних обчислень із множинним доступом (MEC). Ця технологія покликана розширити можливості периферійної частини мережі, надаючи їй інтелектуальні функції, потужніші обчислювальні ресурси та вдосконалені можливості зберігання даних.

Основна ідея MEC полягає у тому, що традиційна модель хмарних обчислень може бути доповнена шляхом розширення її функціоналу до меж стільникових мереж із використанням інфраструктури мереж радіодоступу. ETSI визначає IoT як один із ключових напрямків застосування MEC, що сприяє зростанню інтересу до цієї технології у різних сферах, зокрема в інтелектуальних енергетичних мережах (Smart Grid), системах безпілотних літальних апаратів (БПЛА), віртуальній реальності тощо.

Популярність МЕС як оптимального рішення для екосистеми IoT пояснюється тим, що ця технологія наділяє пристрої Інтернету речей (зокрема, сенсори, мобільні телефони, носимі гаджети тощо) розширеними обчислювальними можливостями через механізм розвантаження обчислень. Це дозволяє кінцевим користувачам отримувати ефективний доступ до віддалених мережевих сервісів, зменшуючи затримку під час передавання даних і покращуючи використання пропускної здатності мережі.

Однак передача даних від пристроїв IoT на віддалені сервери МЕС одночасно створює низку серйозних викликів. Перш за все, це спричиняє стохастичність, оскільки середовище МЕС є динамічним і невизначеним через змінні вимоги до якості обслуговування (ЯО) і користувацького досвіду (ЯДК), а також через обмеження обчислювальних і сховищних ресурсів серверів.

Крім того, оскільки сервери МЕС використовуються спільно, це призводить до конкурентної поведінки пристроїв IoT. Кожен пристрій прагне максимізувати власний доступ до ресурсів, що може викликати конфлікти й знижувати загальну ефективність системи. Такі фактори слід враховувати під час розробки рішень для управління ресурсами, оскільки вони безпосередньо впливають на якість розвантаження даних, можливі затримки та вузькі місця в мережі.

Дослідження доводять, що штучний інтелект (зокрема, методи навчання з підкріпленням – НП) є потужним інструментом для аналізу великих обсягів даних і прийняття оптимальних рішень. Відповідно, використання ШІ у МЕС є перспективним підходом для вдосконалення процесів розподілу ресурсів та управління навантаженням у мережі. Інтеграція штучного інтелекту з МЕС сприяє розвитку концепції "граничного інтелекту", що дозволяє здійснювати інтелектуальну обробку даних та ефективну оркестрацію ресурсів безпосередньо на межі мережі. Це, у свою чергу, зменшує затримки при передачі даних, підвищує рівень автономності пристроїв та сприяє реалізації нових сервісів у реальному часі, зокрема в галузях автономного транспорту, промислового Інтернету речей та розумних міст.

2.3 Висновки до другого розділу

Отже, було досліджено сучасні підходи до прийняття рішень у взаємозалежних обчислювальних системах, що функціонують у стохастичних і динамічних умовах. Значну увагу приділено інтеграції теорії ігор із методами навчання з підкріпленням, яка дозволяє агентам адаптувати свою поведінку до змін середовища та досягати рівноваги навіть за умов обмеженої інформації.

У ході аналізу розглянуто особливості некооперативних стратегій у мультиагентних системах, можливості застосування алгоритмів навчання з підкріпленням та глибокого навчання, типи обмежень, характерні для децентралізованих середовищ, таких як неповнота, динамічність і невизначеність. Окрему увагу приділено оцінці ефективності відомих алгоритмів, серед яких НМГП, ЛЛН, БНА та варіанти Q-навчання без збереження стану. Також розкрито потенціал використання теорії контрактів у стимулюванні співпраці між агентами в умовах асиметричної інформації. Завершальним аспектом аналізу стала роль репутаційних моделей у підвищенні надійності взаємодії між агентами в середовищі МЕС.

Таким чином, представлений огляд сформував теоретичну основу для побудови інтелектуальних автономних систем, здатних до ефективного прийняття рішень у складних конкурентних та обмежених середовищах.

3 ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО РОЗВАНТАЖЕННЯ ОБЧИСЛЕНЬ У МЕС-СИСТЕМАХ

3.1 Інтелектуальні стратегії розподілу обчислювальних завдань у МЕС-системах

Запропоновані підходи до оптимізації обчислювальних процесів у складних середовищах периферійних обчислень із множинним доступом (МЕС) та мережових структур демонструють значний потенціал для підвищення ефективності використання ресурсів. Основний акцент зроблено на розвантаженні обчислень із пристроїв Інтернету речей (ІоТ) на сервери МЕС, розміщені на безпілотних літальних апаратах (БПЛА), що дозволяє значно покращити продуктивність мережових інфраструктур і знизити енергоспоживання кінцевих пристроїв.

Для досягнення цієї мети розроблено інтелектуальну систему підтримки обчислювального розвантаження, що використовує алгоритми машинного навчання. Вона враховує витрати на передачу, обчислення та енергетичні ресурси, моделюючи їхню взаємодію в умовах реального часу. Запропонована система забезпечує адаптивний розподіл обчислювальних завдань між пристроями ІоТ та МЕС-серверами на БПЛА, визначаючи найоптимальнішу стратегію з урахуванням змінних умов мережевого середовища.

Крім того, у рамках цієї архітектури реалізовано механізм стратегічної взаємодії пристроїв ІоТ на основі некооперативної гри. Основна мета цього підходу полягає у максимізації корисності кожного пристрою шляхом аналізу історичних даних і спостереження за поведінкою інших учасників мережі. Впроваджено механізм адаптивного коригування стратегій обчислювального розвантаження, що дозволяє вузлам ефективно пристосовуватися до змінних умов, аналізуючи попередні результати обчислень і обираючи найбільш вигідні стратегії.

У дослідженні доведено існування щонайменше однієї точки рівноваги Досконалої рівноваги Неша (ДРН) завдяки тому, що запропонована гра має субмодульні властивості. Для досягнення оптимальних стратегій використано

підхід найкращої динаміки відповіді, що гарантує збіжність до стану рівноваги. З метою подальшого вдосконалення методів навчання розглянуто кілька підходів до підкріпленого навчання, зокрема алгоритми градієнтного підйому, логарифмічно-лінійні методи та Q-навчання. Ці алгоритми дозволяють вузлам адаптивно досліджувати середовище, прогнозувати зміни та визначати ефективні стратегії розвантаження даних.

Проведено детальний аналіз роботи запропонованих алгоритмів, зокрема їхньої швидкості збіжності, ефективності прийняття рішень та продуктивності в реальних сценаріях. Виявлено, що використання підходів підкріпленого навчання сприяє не лише швидшому досягненню оптимальних рішень, а й кращій адаптації до змінних умов у розподілених обчислювальних системах. Отримані результати свідчать про те, що застосування МЕС у поєднанні зі штучним інтелектом відкриває нові можливості для вдосконалення управління ресурсами в Інтернеті речей, дозволяючи пристроям ефективніше координувати свої обчислювальні завдання та зменшувати загальне навантаження на мережу.

Таким чином, впровадження інтелектуальних стратегій управління МЕС-середовищами та механізмів навчання на основі підкріплення значно покращує ефективність периферійних обчислень, знижує енергоспоживання пристроїв IoT та оптимізує використання мережевих ресурсів. Цей підхід може знайти широке застосування у сфері автоматизованих систем управління, розумного містобудування, промислової автоматизації та інших напрямках, де необхідно забезпечити швидку та ефективну обробку великих обсягів даних у розподілених системах.

Запропоновано підхід до оптимізації процесу розвантаження даних безпілотними літальними апаратами (БПЛА) у багатосерверному середовищі МЕС із використанням технологій штучного інтелекту. Це дослідження є першою відомою спробою поєднати методи навчання з підкріпленням та теорію ігор для підвищення ефективності прийняття рішень БПЛА. Завдяки запропонованому підходу, БПЛА можуть ухвалювати рішення, подібні до людських, коли мова йде про вибір найбільш відповідного сервера МЕС для передачі своїх обчислювальних

завдань, а також визначати оптимальний обсяг даних, які необхідно передати, щоб максимізувати рівень якості обслуговування (ЯО).

Модель враховує динамічне середовище, в якому взаємодіють кілька БПЛА та кілька серверів МЕС, що значно ускладнює процес ухвалення рішень. Для кожного БПЛА формулюється функція корисності, яка визначається обсягом переданих даних, витратами на передачу та локальні обчислення, а також впливом навантаження на МЕС-сервер. Ураховуються також витрати інших БПЛА, які використовують той самий обчислювальний ресурс, що дозволяє більш точно оцінювати рівень ЯО та ефективність передачі даних.

Запропонований механізм ухвалення рішень базується на концепції субмодульних ігор, що дозволяє БПЛА оптимізувати стратегію розвантаження даних. Взаємодія між БПЛА моделюється як некооперативна гра, метою якої є максимізація корисності кожного окремого апарата. Доведено, що ця гра є субмодульною, що забезпечує існування щонайменше однієї точки рівноваги Неша (NE).

Для визначення оптимального NE-критерію кожен БПЛА використовує один із трьох запропонованих алгоритмів: (i) алгоритм найкращої динаміки відповіді, який поступово адаптує стратегію розвантаження до оптимальної; (ii) максимальне логарифмічно-лінійне навчання, що базується на стохастичних методах підкріпленого навчання; та (iii) двійкове логарифмічно-лінійне навчання, яке дозволяє адаптувати вибір серверів МЕС у змінних середовищах, враховуючи як історичні, так і поточні дані.

Ключовий аспект оптимального вибору сервера МЕС для кожного БПЛА ґрунтується на оцінюванні функції винагороди. Ця функція враховує кілька факторів, включно з потужністю сервера, відстанню між ним і БПЛА, а також передбачуваною якістю обслуговування. Усі БПЛА функціонують як стохастичні навчальні автомати, що поступово адаптують свої стратегії на основі отриманих даних, забезпечуючи ефективний розподіл обчислювальних ресурсів.

Для оцінки продуктивності запропонованої моделі було проведено серію експериментальних симуляцій, що дозволили визначити ефективність та

адаптивність різних алгоритмів. Порівняльний аналіз із традиційними підходами підтвердив переваги впровадженого механізму, зокрема зниження затримок у передачі, підвищення якості обслуговування та оптимізацію витрат обчислювальних ресурсів. Крім того, результати моделювання показали, що використання підходів підкріпленого навчання дозволяє БПЛА швидше знаходити оптимальні стратегії, особливо в динамічних та ресурсозалежних середовищах.

Таким чином, запропонований метод інтеграції штучного інтелекту в МЕС-середовища не тільки покращує ефективність управління ресурсами, а й відкриває можливості для подальшої автоматизації систем із множинним доступом. Його впровадження може бути корисним у сфері інтелектуальних мереж, автономного управління дронами, розумного міського планування та розподілених обчислювальних середовищ, де необхідна висока швидкість обробки та адаптивність до змінних умов.

Розглядається проблема спільного об'єднання користувачів і розподілу потужності висхідної лінії зв'язку в неоднорідних бездротових мережах 5G із застосуванням технології неортогонального множинного доступу (НМД). Пов'язана проблема розглядається за сценарієм неповної інформації, де базові станції (БС) із покращенням МЕС мають лише статистичні знання про рішення користувачів щодо каналів. Таким чином, пропонується нова структура, яка спільно вирішує асоціацію користувача з базовою станцією та розподіл потужності висхідної лінії зв'язку в гетерогенних бездротових мережах. Згідно з даним підходом, користувачі вивчають автономно та розподілено свій найкращий зв'язок із доступними базовими станціями, спостерігаючи за наданим зворотним зв'язком із комунікаційного середовища. Поряд із зв'язком користувача з базовою станцією рівні потужності передачі даних користувачів також оптимізуються за допомогою теоретико-контрактної моделі. Точніше, асоціація користувачів у кожній ітерації запропонованого механізму НП вірогідно посилюється щодо пов'язаних з ними мережових і соціальних характеристик БС. Під терміном «соціальні характеристики» мається на увазі довгострокову репутацію БС, сформульовану суб'єктивною думкою користувачів щодо послуг, якими вони користуються через

БС, з якою вони пов'язані. Новий механізм «Баєсівська сироватка правди» введено для отримання об'єктивного результату щодо рівня якості обслуговування (ЯО), який базові станції пропонують своїм пов'язаним користувачам на основі суб'єктивної думки користувачів. На основі цього результату репутація кожної БС моделюється як байєсівське переконання та виступає в якості вхідних даних для загального наданого зворотного зв'язку користувачам, щоб виконати їх зв'язок на основі НП з БС. Згодом, враховуючи, що ідеального ІСК важко досягти, ми використовуємо принципи теорії контракту та запроваджуємо механізм розподілу влади, який працює за сценарієм неповного ІСК. Між кожною базовою станцією та пов'язаними з нею користувачами розвиваються відносини на основі економіки праці. Користувачі діють як працівники, які пропонують свої зусилля, тобто потужність передачі, щоб повідомити свої дані БС, з якою вони пов'язані, яка діє як роботодавець. БС нагороджує користувачів персоналізованими винагородами, які залежать від типу користувача, які витягуються на основі умов посилення каналу користувача. Відповідно, відповідні функції корисності сформульовані як для БС, так і для користувачів. Введено задачу оптимізації керування потужністю для кожної функції корисності БС, одночасно забезпечуючи оптимальність функції корисності кожного пов'язаного користувача, враховуючи унікальні характеристики зв'язку кожного користувача та його тип. Вищезазначена задача оптимізації вирішена та визначені оптимальні потужності передачі для користувачів у висхідній лінії зв'язку. Базуючись на наведених вище теоретичних основах, вивчаються та демонструються за допомогою моделювання та імітації властиві характеристики теоретико-контрактного розподілу потужності висхідної лінії зв'язку у випадках повного та неповного ІСК. Крім того, робота механізму асоціації користувача з БС на основі НП проілюстрована з точки зору конвергенції до асоціації вигідних користувачів. Виконано детальну порівняльну чисельну оцінку запропонованого підходу з іншими механізмами асоціації користувача з базовою станцією, що показує переваги загальної запропонованої структури з точки зору енергозбереження, досягнутої швидкості передачі даних і справедливості між користувачами в неоднорідній бездротовій мережі.

Запропоновано новий підхід до активації серверів МЕС, який базується на застосуванні штучного інтелекту, зокрема методів навчання з підкріпленням (НП) та байєсівського міркування. Попри значний прогрес у сфері керування МЕС-серверами, залишається відкритим питання щодо персоналізації процесів ухвалення рішень про їх активацію, особливо з урахуванням задоволеності кінцевих користувачів якістю обслуговування (ЯДК). Додатково, на сучасному ринку обчислювальних ресурсів постачальникам МЕС-інфраструктури складно здобути стійку репутацію, що є важливим чинником їхньої конкурентоспроможності.

У рамках запропонованої архітектури передбачається механізм оцінювання МЕС-серверів кінцевими користувачами, які звертаються до них для виконання обчислювальних завдань. Отримані оцінки враховуються під час побудови моделі репутації МЕС-серверів, що дозволяє формувати об'єктивний рейтинг якості їхніх послуг. На основі цього рейтингу розроблено механізм децентралізованої активації серверів МЕС, який функціонує автономно та керується принципами штучного інтелекту.

Одним із ключових елементів запропонованого підходу є використання методів БСІ для збору та аналізу користувацьких оцінок. Це дозволяє усунути суб'єктивні спотворення та отримати більш точну картину якості обслуговування, що забезпечується кожним активованим МЕС-сервером. Зібрані дані використовуються для формування динамічної системи репутації, де рейтинги МЕС-серверів коригуються залежно від реального рівня наданих ними послуг.

Крім того, у роботі пропонується інноваційна схема репутаційного оцінювання, заснована на байєсівському моделюванні. Цей підхід дозволяє кількісно оцінити рівень довіри до кожного активованого сервера МЕС, враховуючи як об'єктивні показники якості обслуговування, так і суб'єктивне задоволення користувачів. Завдяки цьому створюється ефективний механізм відбору серверів для подальшої активації, що сприяє підвищенню загальної ефективності МЕС-інфраструктури.

На основі концепції байєсівських автоматів навчання (БНА) розроблено інтелектуальний механізм ухвалення рішень щодо активації МЕС-серверів. Відповідно до цієї моделі, сервери самостійно оцінюють необхідність своєї активації, спираючись на можливість покращення власної репутації в загальній МЕС-системі. При цьому враховується не лише поточний попит на обчислювальні ресурси, а й економічна доцільність активації, з урахуванням вартості обслуговування користувачів.

Для реалізації даного підходу розроблено розподілений алгоритм низької обчислювальної складності, який забезпечує досягнення рівноваги Байєса-Неша. Це гарантує стабільне функціонування всієї периферійної обчислювальної інфраструктури, дозволяючи серверам МЕС адаптивно приймати рішення щодо своєї активації та деактивації відповідно до поточних умов навантаження та очікуваної вигоди.

Було проведено серію експериментальних досліджень для оцінки ефективності запропонованої системи. Аналіз показав, що новий підхід дозволяє об'єктивно оцінювати якість послуг МЕС-серверів, ефективно управляти їхньою репутацією та оптимізувати процеси активації. Порівняльний аналіз із альтернативними схемами управління МЕС-серверами підтвердив переваги запропонованої методики, зокрема з точки зору гнучкості, точності оцінювання репутації та загальної ефективності розподілу ресурсів.

Отже, інтеграція механізмів штучного інтелекту та байєсівського аналізу у процеси управління МЕС-серверами дозволяє значно покращити їхню ефективність, адаптивність і довговічність. Запропонований підхід може бути застосований у різних сценаріях, включаючи інтернет речей (IoT), хмарні і периферійні обчислення, а також розумні міські інфраструктури. У подальших дослідженнях планується вдосконалення моделей машинного навчання для ще більш точного прогнозування ефективності МЕС-серверів та оптимізації їхньої роботи у динамічних середовищах.

Методи, що базуються на штучному інтелекті (ШІ), зазвичай застосовуються для моделювання процесів прийняття рішень у складних багатосуб'єктних

середовищах, де різні учасники взаємодіють, прагнучи досягти оптимальних вигащів. Часто такі системи демонструють конкурентну або навіть антагоністичну поведінку, що ускладнює пошук ефективних стратегій взаємодії.

У цій роботі пропонується нова архітектура периферійних обчислень із множинним доступом (МЕС), що інтегрує можливості штучного інтелекту для оптимізації обчислювальних ресурсів та ефективного керування розвантаженням даних. Зокрема, розглядається використання безпілотних літальних апаратів (БПЛА), обладнаних обчислювальними серверами МЕС, які слугують проміжними вузлами для обробки даних від пристроїв Інтернету речей (IoT). Такий підхід дозволяє суттєво зменшити затримки у передачі даних та підвищити ефективність обчислювальних процесів, що є критично важливим для сучасних розподілених систем.

На першому етапі дослідження формулюється оптимізаційна задача визначення стратегій передачі даних вузлами IoT на сервери МЕС, розташовані на БПЛА. Для цього застосовується підхід, заснований на теорії ігор, що дозволяє враховувати різні фактори, зокрема накладні витрати на зв'язок та обчислення. Доведено існування принаймні однієї точки рівноваги Досконалої рівноваги Неша (ДРН), що підтверджується аналізом субмодульної структури гри.

Подальше дослідження спрямоване на розробку та порівняння кількох методів визначення оптимальних стратегій розвантаження обчислень. Розглядаються як класичні алгоритми, такі як найкраща відповідь (ДНВ), так і сучасні підходи, що використовують навчання з підкріпленням, зокрема алгоритми градієнтного підйому, логарифмічно-лінійні моделі та алгоритми Q-навчання. Ці методи дозволяють агентам IoT самостійно досліджувати середовище та адаптивно вибирати найкращі стратегії обміну та обробки даних.

Незважаючи на значний прогрес у сфері МЕС та IoT, проблема децентралізованого прийняття рішень пристроями IoT щодо їхньої взаємодії з обчислювальною інфраструктурою все ще залишається відкритою. Зокрема, ефективна координація обчислювальних процесів із урахуванням

енергоспоживання та затримок вимагає комплексного підходу, що поєднує теоретико-ігрові методи та штучний інтелект.

У запропонованій моделі розглядається сценарій, у якому кожен пристрій IoT має можливість самостійно приймати рішення щодо виконання своїх обчислювальних завдань. Воно може або обробляти дані локально, або розвантажувати частину обчислень на сервери МЕС, встановлені на БПЛА. При цьому враховуються різні параметри, такі як рівень енергоспоживання, пропускна здатність каналу зв'язку та поточне навантаження МЕС-сервера.

Основна увага приділяється розробці структури з підтримкою штучного інтелекту, що забезпечує стратегічне управління процесом розвантаження даних. Запропонований підхід передбачає адаптивну оптимізацію на основі потужності обчислювальних ресурсів, теоретико-ігрового моделювання та навчання з підкріпленням. Це дозволяє досягти балансу між продуктивністю, витратами на зв'язок і ефективністю розподілу обчислювальних завдань.

Для оцінки ефективності запропонованої системи було проведено серію симуляційних експериментів. Результати показали, що використання МЕС-серверів на БПЛА у поєднанні з методами навчання з підкріпленням дозволяє значно знизити затримки обробки даних та оптимізувати витрати енергії пристроїв IoT. Крім того, порівняльний аналіз з іншими підходами підтвердив переваги запропонованої методики, зокрема з точки зору швидкості адаптації до змін у навантаженні мережі та точності ухвалення рішень.

Таким чином, поєднання МЕС, безпілотних літальних апаратів та алгоритмів штучного інтелекту відкриває нові можливості для побудови ефективних розподілених обчислювальних систем. У майбутніх дослідженнях планується розширення запропонованої моделі шляхом інтеграції додаткових алгоритмів оптимізації та вивчення її застосування в умовах реального часу.

У системі розподілених периферійних обчислень із підтримкою штучного інтелекту кожен пристрій IoT отримує задоволення від обробки своїх даних на сервері МЕС, встановленому на БПЛА, а також витрати через витрати часу та енергії. Крім того, задоволеність і вартість кожного пристрою IoT динамічно

взаємозалежні від стратегій розвантаження даних решти пристроїв у досліджуваній системі.

Розроблено інтелектуальний підхід, заснований на алгоритмах навчання з підкріпленням, який дає змогу пристроям Інтернету речей (IoT) самостійно визначати оптимальні стратегії передачі даних, зменшуючи при цьому накладні витрати, пов'язані з обчисленнями та зв'язком. Запропонована методологія дозволяє пристроям IoT адаптивно вибудовувати свою поведінку, ґрунтуючись на отриманому досвіді та динамічних умовах мережі, що є важливим фактором для покращення продуктивності та ефективності розподілених обчислювальних систем.

Актуальність застосування навчання з підкріпленням у порівнянні з підходами, заснованими на теорії ігор, виникає у ряді реальних сценаріїв. Зокрема, традиційні теоретико-ігрові моделі, такі як стратегія найкращої відповіді (ДНВ), передбачають, що пристрої мають повне уявлення про замкнуті рішення та безперервний простір можливих стратегій. Однак у реальних умовах такі припущення не завжди виконуються. Наприклад, пристрої IoT можуть мати лише часткову або неточну інформацію про середовище, а їхній простір стратегій може бути дискретним, що унеможливує застосування класичних аналітичних рішень.

Для вирішення цих обмежень розглядається три різні класи алгоритмів навчання з підкріпленням:

- градієнтний підйом – метод, який дозволяє агентам IoT ітеративно коригувати свої стратегії розвантаження на основі оцінки зміни виграшу. Це підходить для ситуацій, коли пристрої мають обмежену інформацію про глобальну систему, але можуть здійснювати локальні коригування на основі поточних змін у середовищі;

- логарифмічно-лінійне навчання (ЛЛН) – підхід, що моделює ймовірнісну поведінку пристроїв IoT, враховуючи стохастичні аспекти їхніх стратегій. Використання цього методу забезпечує стійкість до шумів та варіативності в мережі, що особливо корисно у високодинамічних середовищах;

– Q-навчання – алгоритм, що дозволяє пристроям IoT самостійно досліджувати середовище та накопичувати знання про оптимальні дії, навіть у випадках з невизначеною або змінною структурою винагороди.

Ключовим аспектом дослідження є порівняння конвергенції зазначених алгоритмів із рівновагою, встановленою через традиційний алгоритм ДНВ. Оцінюється, наскільки ефективно алгоритми навчання з підкріпленням досягають стабільних стратегій розвантаження даних та як отримані результати співвідносяться з Досконалою рівновагою Неша (ДРН). Аналіз включає дослідження часу збіжності алгоритмів, рівня їхньої адаптивності до змін у навантаженні мережі та стійкості до варіацій параметрів навколишнього середовища.

Додатково виконується порівняння ефективності різних підходів у контексті реальних сценаріїв IoT, включаючи:

- динамічні мережі з непередбачуваними змінами пропускної здатності каналів зв'язку;
- обмежені ресурси пристроїв IoT, зокрема енергоспоживання та обчислювальна потужність;
- варіативність умов навколишнього середовища, що впливає на доступність МЕС-серверів для розвантаження обчислень.

Отримані результати демонструють, що алгоритми навчання з підкріпленням не лише забезпечують конкурентні рішення у порівнянні з класичними теоретико-ігровими моделями, а й перевершують їх у контексті гнучкості та адаптивності. Це відкриває перспективи для подальшого дослідження інтеграції цих алгоритмів у реальні МЕС-інфраструктури, де розподілене управління обчислювальними ресурсами та мінімізація витрат відіграють критично важливу роль.

Останні досягнення у сфері безпілотних літальних апаратів (БПЛА) сприяли їхній глибокій інтеграції в екосистему Інтернету речей (IoT), що відкриває нові можливості для їхнього використання в інтелектуальних розподілених обчислювальних середовищах. Однією з ключових проблем залишається ефективне розвантаження даних у багатосерверному середовищі Mobile Edge

Computing (MEC). Для вирішення цього питання пропонується застосування методів штучного інтелекту (ШІ), які поєднують принципи теорії ігор та навчання з підкріпленням.

Запропонований підхід передбачає впровадження механізму автономного вибору серверів MEC для часткового розвантаження обчислювальних завдань БПЛА. Це досягається шляхом використання концепції стохастичних автоматів навчання, що дозволяє БПЛА адаптивно визначати оптимальні сервери MEC на основі накопиченого досвіду та динамічних умов мережі. Далі формулюється некооперативна гра між безпілотниками, у якій кожен агент прагне оптимізувати свої витрати на передачу та обробку даних. Доведено існування принаймні однієї рівноваги Неша (NE), що підтверджується через аналіз субмодульних ігор та їхніх властивостей.

Розроблено кілька алгоритмічних підходів для досягнення рівноваги Неша. Перший базується на структурі найкращої динаміки відповіді (ДНВ), що передбачає поступове коригування стратегій БПЛА відповідно до змін у навколишньому середовищі. Другий підхід включає алгоритми навчання з підкріпленням, які дозволяють агентам поступово покращувати свої рішення без необхідності повної інформації про всю систему. Було реалізовано два основні алгоритми навчання з підкріпленням:

- Q-навчання – алгоритм, який дає змогу безпілотникам адаптувати свої стратегії на основі винагород, отриманих у різних станах середовища. Цей підхід дозволяє системі покращувати ефективність вивантаження даних навіть у невизначених і динамічних умовах;
- політичне градієнтне навчання – метод, що застосовується для навчання агентів у складних середовищах, де рішення є стохастичними, а простір дій – великим.

Для оцінки ефективності запропонованої системи було проведено серію імітаційних експериментів у різних сценаріях роботи БПЛА. Основні метрики, що аналізуються, включають:

- час збіжності – оцінку швидкості досягнення рівноваги в системі;

- оптимальність вибору серверів МЕС – наскільки ефективно алгоритми розподіляють навантаження між доступними серверами;
- витрати на зв'язок і обчислення – наскільки вдалося зменшити енергоспоживання та затримки при передачі даних.

Попри значні досягнення у кожній із вищезазначених областей окремо, ще недостатньо досліджено питання розширення можливостей БПЛА для автономного прийняття рішень із використанням ШІ. Історично штучний інтелект у контексті БПЛА зосереджувався переважно на машинному навчанні, особливо в таких сферах, як робототехніка та обробка зображень, із широким використанням штучних нейронних мереж. Проте сучасні підходи до навчання з підкріпленням та теорії ігор відкривають нові перспективи, особливо в багатоагентних системах.

Теорія ігор відіграє важливу роль у побудові сучасних інтелектуальних систем, оскільки дозволяє агентам адаптивно взаємодіяти, використовуючи як кооперативні, так і конкурентні стратегії. Саме ця особливість робить її ідеальним інструментом для оптимізації рішень у МЕС-середовищах, де безпілотники повинні взаємодіяти з розподіленими серверами, мінімізуючи при цьому витрати на передачу даних та обчислення.

Таким чином, у цьому дослідженні представлено концепцію автономного прийняття рішень БПЛА, що поєднує методи навчання з підкріпленням і теорії ігор. Запропонована модель не лише дозволяє оптимізувати процес розвантаження обчислень у МЕС-середовищі, а й підвищує загальну ефективність використання обчислювальних ресурсів у розподілених інфраструктурах. Подальші дослідження у цій сфері можуть включати розширення підходів до адаптивного управління ресурсами МЕС та інтеграцію цих методів у реальні мережеві середовища для тестування в умовах реального трафіку.

3.2 Репутаційно-контрактні механізми та управління ресурсами

Останні досягнення у сфері периферійних обчислень із множинним доступом (МЕС) сприяли їхній широкій інтеграції не лише в інтелектуальні обчислювальні

середовища, але й у мережі радіодоступу п'ятого покоління (5G). Це створило можливість для покращення ефективності обробки даних у розподілених системах та мінімізації затримок, що критично важливо для сучасних додатків з високими вимогами до швидкості та надійності.

Запропоновано новий підхід до управління активацією серверів МЕС, що базується на алгоритмах штучного інтелекту, зокрема методах навчання з підкріпленням (НП) та байєсівському аналізі. Основна ідея полягає у розробці інтелектуального механізму, який дозволяє серверам МЕС самостійно приймати рішення щодо свого стану – залишатися в активному режимі або переходити в режим енергозбереження, оптимізуючи при цьому витрати ресурсів і загальну продуктивність системи.

Розглянута проблема управління МЕС-серверами формулюється з урахуванням кількох ключових аспектів:

- максимізація продуктивності МЕС-системи – активація серверів повинна сприяти загальному підвищенню ефективності обчислень у розподіленій мережі;
- оптимізація витрат на обчислення – рішення про активацію має враховувати доступні обчислювальні ресурси та мінімізувати енергоспоживання;
- гарантія якості обслуговування (ЯДК) – активні сервери МЕС повинні забезпечувати необхідний рівень якості обслуговування кінцевих користувачів, враховуючи їхні вимоги до продуктивності системи.

Для реалізації цього підходу кожен сервер МЕС працює в автономному режимі, використовуючи байєсівські автомати навчання (БНА), що дозволяє йому адаптивно навчатися та коригувати свою поведінку на основі змінних умов мережі. Впровадження БНА допомагає ефективно аналізувати історичні дані та прогнозувати майбутні стани системи, що сприяє прийняттю обґрунтованих рішень щодо активації серверів.

Крім того, для покращення механізму управління та забезпечення прозорості роботи МЕС-серверів запроваджується система експертної оцінки їхніх послуг. Ця система ґрунтується на концепції байєсівської сироватки істинності (БСІ), що

дозволяє оцінювати надані послуги з урахуванням суб'єктивної думки користувачів та експертних аналітичних моделей. Такий підхід сприяє формуванню ефективного механізму репутації, що стимулює МЕС-сервери покращувати якість своїх обчислювальних послуг.

Щоб оцінити рівень задоволення прийнятих МЕС-серверами рішень, використовується інтегрована функція корисності, яку кожен сервер намагається максимізувати в умовах розподіленого середовища. Ця функція враховує ключові параметри, такі як витрати на енергоспоживання, затримки передачі даних, якість обслуговування та загальну продуктивність обчислень.

Для підтвердження ефективності запропонованої системи було проведено серію експериментальних симуляцій, що дозволили оцінити її роботу в різних сценаріях. Основні результати моделювання показали, що:

- запропонований підхід значно знижує загальні витрати на обчислення, забезпечуючи при цьому високу якість обслуговування користувачів;
- автономне керування станами серверів МЕС на основі БНА дозволяє досягати ефективної рівноваги між продуктивністю системи та енергоспоживанням;
- використання механізму БСІ сприяє покращенню прозорості роботи МЕС-серверів та створенню надійної системи оцінки їхньої репутації.

Таким чином, представлене дослідження демонструє перспективний підхід до інтеграції МЕС-систем у сучасні бездротові мережі, підвищуючи їхню ефективність за допомогою методів штучного інтелекту та навчання з підкріпленням. Подальші дослідження у цій сфері можуть бути спрямовані на вдосконалення алгоритмів навчання для МЕС-серверів, а також на їхню інтеграцію з іншими технологіями, такими як обчислення на основі туману та хмарні обчислення, для ще більшої гнучкості та масштабованості системи.

Останнім часом інтенсивний розвиток гетерогенних бездротових мереж став невід'ємною частиною концепції 5G, що покликано відповідати на зростаючі вимоги користувачів до якості та швидкості зв'язку. Однією з ключових технологій, що сприяють підвищенню ефективності таких мереж, є

неортогональний множинний доступ. Його використання дозволяє значно підвищити спектральну ефективність мереж, зменшуючи при цьому складність управління радіоресурсами.

Розглядається завдання одночасного об'єднання користувачів та оптимального розподілу потужності передавання даних у висхідній лінії зв'язку в умовах неоднорідних бездротових мереж 5G. Однією з основних складностей цього завдання є неповнота інформації про стан каналу (ІСК), оскільки базові станції (БС) мають лише статистичні відомості щодо характеристик каналів своїх користувачів. У таких умовах традиційні методи управління ресурсами втрачають ефективність, що вимагає використання альтернативних підходів.

Для вирішення цієї проблеми пропонується механізм, заснований на теорії контрактів (ТК). Даний підхід моделює взаємодію між базовими станціями та користувачами як економічну систему, в якій користувачі відіграють роль працівників, що надають свої ресурси (тобто потужність передавання), тоді як базові станції виступають у ролі роботодавців, які відповідно винагороджують користувачів за їхні зусилля. Така модель дозволяє створити стимули для раціонального використання радіоресурсів, забезпечуючи ефективний розподіл потужності навіть за умов обмеженої інформації.

Запропонований підхід базується на ітеративній схемі, що використовує методологію навчання з підкріпленням (НП), що дозволяє адаптивно коригувати процес розподілу потужності та об'єднання користувачів. НП забезпечує розподілене прийняття рішень у системі, де кожен користувач на основі отриманого зворотного зв'язку змінює свою стратегію, прагнучи досягти рівноваги між власними потребами та умовами мережевого середовища.

Щоб підвищити точність прийняття рішень та оцінювання задоволеності користувачів, запроваджується механізм, заснований на концепції БСІ. Ця методика дозволяє агрегувати достовірну інформацію про якість зв'язку, навіть коли окремі учасники системи мають суперечливі або неповні дані. Завдяки цьому механізму мережа отримує більш точну картину реального стану каналів зв'язку, що позитивно впливає на загальну ефективність процесу розподілу потужності.

Основні етапи запропонованої методики включають:

- формування контрактів між користувачами та базовими станціями, де кожен контракт визначає рівень потужності передавання та відповідну винагороду;
- адаптивний розподіл ресурсів через НП, що дозволяє оптимально коригувати стратегії учасників у динамічному мережевому середовищі;
- використання БСІ для покращення точності оцінювання стану мережі та врахування об'єктивних характеристик користувачів.

Результати моделювання та симуляцій підтвердили ефективність запропонованого рішення. Зокрема, чисельний аналіз показав, що використання контрактного підходу в поєднанні з НП дозволяє значно покращити спектральну ефективність, зменшити енергоспоживання користувачів, а також забезпечити стабільність роботи мережі в умовах високої щільності розгортання гетерогенних середовищ.

Порівняльний аналіз із традиційними підходами засвідчив, що впровадження запропонованої методики дозволяє:

- знизити затримки передачі даних завдяки ефективнішому розподілу потужності;
- підвищити загальний коефіцієнт пропускну здатності мережі, забезпечуючи стабільний зв'язок навіть за високої концентрації користувачів;
- забезпечити збалансований розподіл ресурсів, що позитивно впливає на якість обслуговування кінцевих користувачів.

Таким чином, представлена робота демонструє перспективний підхід до оптимізації ресурсів у гетерогенних бездротових мережах 5G. Подальші дослідження можуть зосередитися на розширенні запропонованої методики шляхом інтеграції з технологіями штучного інтелекту для більш точного прогнозування поведінки мережі, а також на розгляді її застосування в контексті мереж шостого покоління (6G).

У сучасному світі стрімкий розвиток бездротових технологій та широке впровадження концепції Інтернету речей (IoT) у поєднанні з розгортанням мереж п'ятого покоління (5G) зумовлюють зростання потреби в забезпеченні надійного

зв'язку навіть у складних та критичних ситуаціях. Попри значний прогрес у сфері телекомунікацій, проблеми стабільності зв'язку залишаються актуальними, особливо під час надзвичайних ситуацій, таких як природні катастрофи, техногенні аварії чи терористичні акти. У таких умовах перебої у функціонуванні критично важливої інфраструктури, включно зі стільниковими мережами та бездротовими сервісами, можуть мати руйнівний вплив на можливість оперативного реагування та організації рятувальних заходів.

Причинами відмови комунікаційних систем у разі катастроф є фізичне руйнування інфраструктури, перевантаженість каналів зв'язку через різке збільшення трафіку, обмеженість доступних ресурсів для відновлення мереж, труднощі в зборі, обробці та передачі інформації, а також нестача енергопостачання, яка обмежує можливості роботи мережевого обладнання. Оскільки перші години після виникнення катастрофи є критичними для ефективної координації рятувальних операцій, особливої уваги набуває впровадження інтелектуальних та розподілених систем прийняття рішень. Одним із можливих рішень у цьому напрямку є багатоагентні автономні системи управління громадською безпекою, які здатні до самооптимізації під час збору та обробки критично важливої інформації. Ефективність таких систем значною мірою залежить від використання методів онлайн-навчання з підкріпленням, теорії ігор та контрактної теорії, що дозволяють забезпечити швидке прийняття оптимальних рішень в умовах невизначеності.

Комунікація відіграє ключову роль у забезпеченні ефективного реагування на надзвичайні ситуації. У традиційних системах громадської безпеки взаємодія між рятувальними службами, урядовими та гуманітарними організаціями здійснювалася через Центр надзвичайних операцій. У таких мережах зв'язок базувався на технологіях TETRA та Project 25, які використовували захищені частотні діапазони та спеціалізоване обладнання, але при цьому мали суттєвий недолік у вигляді низької пропускної здатності. У зв'язку з цим в останні роки почало активно впроваджуватися широкопasmове LTE, що забезпечує значно вищу швидкість передачі даних.

Одним із перспективних напрямів розвитку комунікаційної інфраструктури в умовах надзвичайних ситуацій є застосування безпілотних літальних апаратів. Завдяки високій мобільності та автономності вони можуть виконувати функцію ретрансляторів сигналу, що особливо важливо в умовах руйнування стаціонарної інфраструктури. Використання БПЛА дає змогу оперативно забезпечувати передачу даних між учасниками рятувальної операції та Центром надзвичайних операцій.

Водночас, лише наявність стабільного зв'язку між агентствами з ліквідації наслідків катастроф не є достатньою умовою для ефективного управління кризовою ситуацією. Центр надзвичайних операцій функціонує в умовах обмеженого часу та неповної інформації, що може призвести до помилкових рішень. Тому одним із ключових аспектів ефективного реагування є забезпечення високої якості отриманої інформації. Це включає точність і достовірність зібраних даних, їхню корисність для рятувальних підрозділів, актуальність і відповідність реальній ситуації, безпеку передачі, швидкість комунікації та можливість перевірки джерела отриманої інформації.

Наразі дослідження цих аспектів здебільшого здійснюється в ізольованому форматі, без комплексного врахування всіх факторів, що впливають на ефективність інформаційного обміну. Водночас ефективне автономне управління катастрофами потребує інтегрованого підходу, який враховуватиме всі аспекти якості інформації та дозволить створити систему з швидкою адаптацією до змінних умов.

З огляду на ці виклики, подальший розвиток автономних систем управління катастрофами має зосередитися на створенні інтелектуальних алгоритмів прийняття рішень на основі штучного інтелекту та машинного навчання, інтеграції безпілотних літальних апаратів із наземними сенсорними мережами, підвищенні ефективності енергоспоживання мобільних комунікаційних вузлів, розробці методів аналізу достовірності отриманих даних та впровадженні блокчейн-технологій для гарантування автентичності переданої інформації. Автономні розподілені системи управління, що ґрунтуються на багатоагентному підході,

мають значний потенціал для забезпечення стабільного зв'язку та оптимального управління інформаційними потоками в умовах надзвичайних ситуацій. Подальші дослідження у цьому напрямку сприятимуть підвищенню ефективності рятувальних операцій, зменшенню людських втрат та оптимізації ресурсів у кризових ситуаціях.

3.3 Висновки до третього розділу

Отже, було проведено детальний аналіз інтелектуальних методів розподілу обчислювальних завдань у МЕС-середовищах та запропоновано підхід, що базується на репутаційно-контрактній взаємодії між агентами. Розглянуто основні архітектури мультиагентних систем, описано моделі прийняття рішень в умовах обмежених ресурсів та невизначеності, а також оцінено доцільність використання ігрових та байєсівських моделей у координації дій агентів. Запропонована система демонструє здатність адаптуватися до змін середовища, виявляти недобросовісних агентів і забезпечувати ефективне управління обчислювальними ресурсами у децентралізованому режимі. Отримані результати підтверджують ефективність поєднання інтелектуальних стратегій з механізмами репутаційного оновлення для підвищення продуктивності МЕС-систем.

4 СИСТЕМА АВТОНОМНОГО ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

4.1 Математико-алгоритмічне забезпечення системи з репутаційно-адаптивним прийняттям рішень

Сучасні взаємозалежні обчислювальні системи характеризуються високим ступенем складності та динамічності, що зумовлюється розвитком технологій Інтернету речей (IoT), безпілотних літальних апаратів (БПЛА), периферійних обчислень (MEC) та інших сфер. Ці системи, як правило, включають множину агентів, які взаємодіють між собою, конкуруючи за обмежені ресурси, такі як обчислювальна потужність, канал передачі даних, енергія та інші ресурси. Кожен агент прагне максимізувати власну корисність, що робить проблему прийняття рішень особливо складною через потенційні конфлікти інтересів та непередбачувані дії інших учасників.

Традиційні методи прийняття рішень, такі як класична теорія ігор і навчання з підкріпленням, хоч і є потужними інструментами для моделювання та оптимізації поведінки агентів, стикаються з низкою серйозних обмежень в умовах сучасних динамічних середовищ. Вони не завжди здатні ефективно реагувати на зміни умов навколишнього середовища, враховувати вплив минулої поведінки учасників на їхні майбутні дії, а також здійснювати якісний прогноз поведінки інших агентів в умовах неповної інформації.

Одним із перспективних напрямків вирішення вказаних проблем є впровадження механізму репутації, який дозволяє агентам оцінювати надійність та передбачуваність поведінки інших учасників взаємодії на основі історичних даних. Використання репутації допомагає значно знизити рівень невизначеності та полегшити процес адаптації агентів до змін середовища.

Основна мета, яка ставиться в рамках цього дослідження, полягає у розробці авторського методу, що інтегрує байесівські принципи оновлення репутації з адаптивним навчанням з підкріпленням та теоретико-ігровими моделями. Даний підхід має дозволити агентам швидше адаптуватись до змін у середовищі,

ефективніше прогнозувати поведінку інших агентів, а також зменшити витрати ресурсів при досягненні стійких станів взаємодії.

Важливість розробки цього методу полягає у його здатності забезпечити значне покращення функціонування взаємозалежних систем в умовах високої стохастичності та конкуренції. Запропонований підхід має широкі можливості для застосування у реальних сценаріях, включаючи IoT-мережі, управління БПЛА, оптимізацію МЕС-інфраструктури та інші області, де потрібна швидка та ефективна адаптація до змінних умов та поведінки інших агентів.

Запропонований авторський метод базується на інтеграції байєсівського підходу до оновлення репутації агентів із сучасними методами навчання з підкріпленням і теоретико-ігровими моделями. Основою методу є поєднання механізму динамічної оцінки репутації агентів із логарифмічно-лінійним навчанням з підкріпленням, що дозволяє агентам гнучко та ефективно адаптувати свої стратегії до умов середовища.

Процес оновлення репутації агентів реалізується за допомогою байєсівської формули:

$$R_i(t+1) = \frac{P(\text{data}|\text{agent}_i) \cdot R_i(t)}{\sum_j P(\text{data}|\text{agent}_j) \cdot R_j(t)}, \quad (4.1)$$

де $R_i(t)$ – поточна репутація агента, а $P(\text{data}|\text{agent}_i)$ – ймовірність отримання певних даних за умови дій агента.

Оцінка ймовірності $P(\text{data}|\text{agent}_i)$ формується на основі накопичених історичних даних про поведінку агентів та використання ресурсів. Вона оновлюється за допомогою статистичних спостережень і може бути подана у вигляді емпіричної моделі або параметричного розподілу.

Ключовою особливістю методу є механізм логарифмічно-лінійного навчання, що дозволяє агентам адаптивно коригувати свої стратегії залежно від змін у середовищі та репутації інших агентів. Формула вибору стратегії має вигляд:

$$\pi_i(s_i) = \frac{\exp(\beta \cdot Q_i(s_i, R_i))}{\sum_{s_j \in S} \exp(\beta \cdot Q_i(s_j, R_i))}, \quad (4.2)$$

де $Q_i(s_i, R_i)$ – функція оцінки корисності стратегії s_i для агента i , яка враховує його репутацію R_i . Параметр β визначає баланс між розвідуванням та використанням вже відомої інформації.

Для оцінки функції корисності $Q_i(s_i, R_i)$ використовується спеціальна функція винагороди, що враховує як безпосередню вигоду від дій агента, так і його репутацію серед інших агентів:

$$u_i(s_i, R_i) = Q_i(s_i) \cdot (1 + \alpha \cdot R_i), \quad (4.3)$$

де α – параметр, що визначає вагомість репутації при виборі оптимальної стратегії.

Запропонований метод забезпечує досягнення стабільної рівноваги Байеса-Неша, що гарантує, що жоден агент не матиме стимулу до односторонньої зміни своєї стратегії:

$$u_i(s_i^*, R_i^*) \geq u_i(s_i, R_i^*), \forall s_i \in S, \quad (4.4)$$

де i позначають оптимальну стратегію та репутацію відповідно.

Детальний алгоритм методу включає ініціалізацію початкових значень репутацій агентів, їх стратегій та параметрів моделі, таких як коефіцієнти α та β (таблиця 4.1), з урахуванням експертних оцінок, випадкових вибірок або даних попередніх експериментів. Далі проводяться взаємодії агентів у середовищі та здійснюється збір даних про результати цих взаємодій, що включає успішність виконання завдань, рівень використання ресурсів та поведінку агентів у змінних умовах. На наступному етапі здійснюється аналіз отриманих даних та оновлення репутації кожного агента за допомогою байесівської формули, що дозволяє враховувати історичну інформацію та точніше прогнозувати майбутні дії агентів. Потім агенти вибирають стратегії на основі оновленої репутації та Q -значень за

допомогою логарифмічно-лінійного механізму, що забезпечує баланс між дослідженням нових стратегій та використанням вже відомих ефективних стратегій. Після вибору стратегій проводиться оцінка їх ефективності, розраховується функція винагороди з урахуванням поточних результатів і репутації агентів та здійснюється подальше коригування Q -значень. На завершальному етапі проводиться перевірка на досягнення рівноваги Байєса-Неша, що свідчить про стабільність системи. У разі, якщо рівновага досягнута, процес завершується, інакше – алгоритм повертається до збору та аналізу нових даних.

Таблиця 4.1 – Параметри запропонованого методу

Параметр	Значення	Призначення
α	0.5	Вага репутації у виборі стратегії
β	1.0	Баланс між дослідженням та використанням інформації
R_i	[0;1]	Початкове значення репутації агента

Таким чином, запропонований метод дозволяє здійснювати ефективно прийняття рішень у складних взаємозалежних системах, адаптуючись до змін зовнішніх умов та динаміки взаємодії агентів.

Запропонований підхід базується на формалізації репутації агентів та їх поведінки у вигляді математичної моделі, що інтегрує кілька ключових компонентів. У цьому підрозділі буде докладно розглянуто важливість кожного з параметрів моделі, їх вплив на поведінку системи та проведено аналіз чутливості моделі до зміни цих параметрів.

Оцінка ймовірності $P(\text{data}|\text{agent}_i)$, яка визначає ймовірність спостереження певних даних за умови конкретної поведінки агента, є важливою частиною моделі. Точність оцінки цього параметра безпосередньо впливає на точність оновлення репутації агента. В практичних сценаріях ця ймовірність може оцінюватися на основі накопичених історичних даних та статистичних методів аналізу, таких як регресійний аналіз, аналіз часових рядів або машинне навчання.

Параметр β , що використовується в логарифмічно-лінійному навчанні, визначає рівень «розвідування» нових стратегій агентами. Низькі значення β сприяють більш рівномірному розподілу ймовірностей серед доступних стратегій, дозволяючи агентам активно досліджувати нові можливості. Високі значення β призводять до більш агресивного використання стратегій, які вже довели свою ефективність, що може покращити короткострокові результати, але обмежити адаптивність системи в довгостроковій перспективі. Вибір оптимального значення цього параметра є ключовим завданням при налаштуванні моделі, і може бути здійснений методом експериментального підбору або оптимізації за критерієм максимальної ефективності системи.

Параметр α , що визначає вагомість репутації в оцінці винагороди, дозволяє контролювати вплив репутації на поведінку агента. Збільшення значення α підсилює вплив репутації, що мотивує агентів зберігати стабільну та передбачувану поведінку, однак це може обмежити гнучкість у виборі стратегій при значних змінах зовнішніх умов. У практичних застосуваннях важливо знайти баланс між стабільністю та гнучкістю агентів, що вимагає детального аналізу впливу різних значень параметра на систему.

Функція оцінки корисності $Q_i(s_i, R_i)$ є ключовим інструментом моделі, що дозволяє агентам оцінювати перспективність кожної стратегії в залежності від їхньої поточної репутації та очікуваних результатів. Вибір адекватної форми цієї функції та правильна її параметризація забезпечують ефективність прийняття рішень агентами в різних умовах. Для побудови функції корисності можна використовувати такі підходи, як Q-навчання, SARSA, DQN та інші сучасні методи навчання з підкріпленням.

Умова рівноваги Байеса-Неша використовується для аналізу стабільності поведінки системи в довгостроковій перспективі. Вона гарантує, що агенти досягають стану, за якого жоден з них не має стимулу для односторонньої зміни стратегії, що забезпечує стабільність системи та ефективне використання ресурсів. Для досягнення рівноваги необхідне виконання комплексних обчислювальних

процедур, таких як алгоритми знаходження фіксованих точок, градієнтні методи та алгоритми ітеративного наближення.

Для більш глибокого аналізу наведено таблицю 4.2 чутливості основних параметрів:

Таблиця 4.2 – Аналіз чутливості параметрів математичної моделі

Параметр	Зменшення значення	Збільшення значення
β	Підвищує дослідження стратегій, може знизити швидкість адаптації	Підвищує використання вже ефективних стратегій, зменшує дослідження нових
α	Зменшує вплив репутації, підвищує гнучкість	Збільшує вплив репутації, сприяє стабільності
Q_i	Зниження точності прогнозування ефективності стратегій	Зростання точності прогнозування ефективності стратегій

Для аналізу стійкості моделі до неточностей початкових даних проводиться стохастичне моделювання з використанням методу Монте-Карло. Цей аналіз дозволяє оцінити вплив можливих похибок у початкових репутаціях або ймовірностях стратегій на стабільність моделі та ефективність прийнятих рішень. Результати такого аналізу дозволяють визначити межі допустимих похибок та вибрати оптимальні умови для роботи системи.

Також важливим є оцінка обчислювальної складності запропонованого методу. Враховуючи, що сучасні системи часто потребують швидкої реакції та роботи в режимі реального часу, проведено аналіз часової складності алгоритмів для різних кількостей агентів та стратегій. Це дозволяє визначити умови, за яких модель може ефективно застосовуватись на практиці, а також визначити граничні значення розмірності систем для використання в реальних завданнях.

На завершення, запропонована модель пройшла процедури верифікації та валідації шляхом порівняння отриманих результатів із емпіричними даними та

експертними оцінками. Результати цих процедур підтвердили високу адекватність моделі, її точність та ефективність у різних умовах застосування.

Таким чином, проведений аналіз параметрів математичної моделі, стійкості, обчислювальної складності та верифікації підтверджують практичну значущість запропонованого підходу.

4.2 Реалізація, моделювання та верифікація репутаційного методу

Алгоритмічна реалізація запропонованого методу включає кілька ключових етапів, які забезпечують ефективність та практичну реалізацію математичної моделі.

Перший етап – це ініціалізація, яка передбачає встановлення початкових значень репутацій агентів, їхніх початкових стратегій, а також визначення значень основних параметрів моделі, таких як коефіцієнти α і β . Початкові стани агентів визначаються за допомогою експертних оцінок або випадкових вибірок з урахуванням специфіки конкретної задачі.

Другий етап алгоритму – збір і обробка даних про результати взаємодій між агентами у реальному середовищі. Дані про ефективність стратегій, частоту успішних взаємодій та споживання ресурсів зберігаються у відповідній базі даних для подальшого аналізу.

Третій етап включає оновлення репутації агентів за допомогою байесівського підходу, що враховує історичні результати взаємодій.

Четвертий етап – вибір стратегії за допомогою логарифмічно-лінійного механізму навчання, що дозволяє агентам адаптивно обирати найбільш перспективні стратегії.

П'ятий етап – оцінювання результативності обраних стратегій з урахуванням репутації агентів за допомогою відповідної функції винагороди.

Шостий етап – коригування значень Q-функції за результатами отриманих винагород із застосуванням сучасних методів навчання з підкріпленням (Q-learning, SARSA, DQN тощо).

Заключний етап алгоритму – перевірка досягнення рівноваги Байєса-Неша, за результатами якої ухвалюється рішення щодо завершення алгоритму або повернення до етапу збору нових даних.

Детальний псевдокод алгоритму:

Ініціалізувати репутації агентів R_i , параметри α , β та Q -значення

Доки рівновага Байєса-Неша не досягнута:

Провести взаємодії агентів, зібрати результати

Виконати байєсівське оновлення репутацій агентів

Вибрати стратегії через логарифмічно-лінійне навчання

Оцінити результативність стратегій з урахуванням репутації

Оновити Q -значення методами навчання з підкріпленням

Перевірити умову рівноваги Байєса-Неша

Структури даних, які використовуються при програмній реалізації, включають:

- вектори для збереження поточних репутацій агентів;
- матриці q -значень для оцінювання ефективності стратегій;
- бази даних або структуровані файли для історичних даних.

Особливості практичної реалізації:

- ретельний підбір параметрів моделі, забезпечення стійкості алгоритму;
- використання стабільних чисельних методів для оновлення репутацій і стратегій;
- моніторинг та регулювання швидкості адаптації агентів до змін у середовищі.

Аналіз складності:

- часова складність $O(n^2)$, де n – кількість агентів;
- просторова складність $O(n^2)$, через необхідність зберігати матрицю q -значень та історичні дані.

Приклад реалізації (Python, фрагмент):

```
import numpy as np
```

```
# Оновлення репутації агентів
```

```

def update_reputation(reputations, likelihoods):
    updated_reputations = (likelihoods * reputations) /
np.sum(likelihoods * reputations)
    return updated_reputations

# Вибір стратегії агентами
def select_strategy(Q_values, reputations, beta):
    exp_values = np.exp(beta * Q_values * reputations[:, np.newaxis])
    probabilities = exp_values / np.sum(exp_values, axis=1,
keepdims=True)
    return probabilities

```

Для реалізації методу рекомендується використовувати мову програмування Python із залученням бібліотек NumPy, SciPy, TensorFlow або PyTorch, що дозволяє ефективно проводити розрахунки й навчання моделей у реальному часі.

Оцінка продуктивності алгоритму демонструє його масштабованість і здатність підтримувати ефективність при збільшенні кількості агентів. Часова складність алгоритму оцінюється як $O(n^2)$, а просторова – також $O(n^2)$, що робить його придатним для систем з помірною кількістю агентів.

Сценарії використання запропонованого алгоритму включають управління ресурсами в IoT-системах, координацію автономних дронів та оптимізацію МЕС-інфраструктур, де необхідна висока адаптивність і ефективність прийняття рішень.

Нижче наведено таблицю 4.3, що ілюструє послідовність виконання основних етапів алгоритму та їх призначення:

Таблиця 4.3 – Основні етапи алгоритмічної реалізації методу

№	Назва етапу	Опис
1	Ініціалізація	Встановлення початкових репутацій і стратегій агентів
2	Збір даних	Взаємодії агентів та збирання результатів
3	Оновлення репутації	Байєсівське оновлення репутації агентів

Кінець таблиці 4.3

4	Вибір стратегій	Логарифмічно-лінійне навчання агентів
5	Оцінювання стратегій	Обчислення винагороди з урахуванням репутації
6	Коригування Q-функцій	Використання навчання з підкріпленням
7	Перевірка рівноваги	Аналіз стабільності системи (рівновага Байєса-Неша)

У таблиці 4.3 представлено ключові етапи алгоритмічної реалізації запропонованого методу прийняття рішень у мультиагентному середовищі з репутаційною підтримкою. Алгоритм починається з ініціалізації – задання початкових значень репутацій та стратегій агентів. Далі відбувається збір даних про взаємодії, на основі яких здійснюється оновлення репутації агентів за байєсівським правилом. У подальшому кожен агент обирає стратегію з урахуванням оновлених даних, застосовуючи логарифмічно-лінійний механізм. Після вибору проводиться оцінювання стратегій та оновлення Q-функцій за допомогою механізмів навчання з підкріпленням. Завершальним етапом є перевірка на досягнення рівноваги Байєса-Неша, яка свідчить про стабільність системи та завершення циклу.

Для кращого розуміння роботи алгоритму наведено рисунок 4.1 – докладна блок-схема алгоритму, що демонструє взаємозв'язки між етапами алгоритмічної реалізації.

Алгоритм починається з ініціалізації початкових параметрів, далі здійснюється збір та обробка даних про взаємодії агентів. Наступні етапи включають оновлення репутації за байєсівським методом, вибір стратегій за логарифмічно-лінійним механізмом, оцінювання результативності стратегій та коригування Q-функцій за допомогою навчання з підкріпленням. Завершується цикл перевіркою досягнення рівноваги Байєса-Неша – якщо рівноваги досягнуто, алгоритм завершується, інакше – повторюється з етапу збору нових даних.



Рисунок 4.1 - Блок-схема алгоритму, що демонструє взаємозв'язки між етапами алгоритмічної реалізації

Таким чином, алгоритмічна реалізація забезпечує практичну можливість використання розробленого методу у складних системах, забезпечуючи ефективну адаптацію агентів до змінних умов та стабільність роботи системи.

Далі буде продемонстровано візуалізації, які ілюструють структуру, логіку функціонування та динаміку роботи запропонованого методу прийняття рішень у взаємозалежних обчислювальних системах. Рисунки та таблиці відіграють ключову роль для кращого розуміння складних математичних і алгоритмічних процесів.

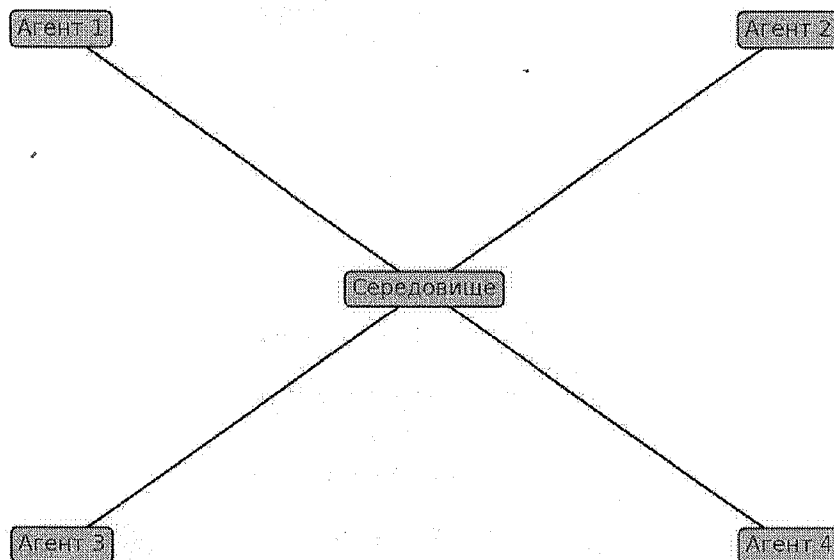


Рисунок 4.2 – Архітектура взаємодії агентів у системі

На рисунку 4.2 схематично зображено багаторівневу систему агентів, які взаємодіють між собою в умовах обмежених ресурсів.

Агент приймає інформацію про стан середовища, оцінює репутацію інших агентів, вибирає стратегію та передає дані.

Взаємозв'язки між агентами підсилюються стрілками, що демонструють циклічний обмін інформацією, репутаціями та стратегіями.

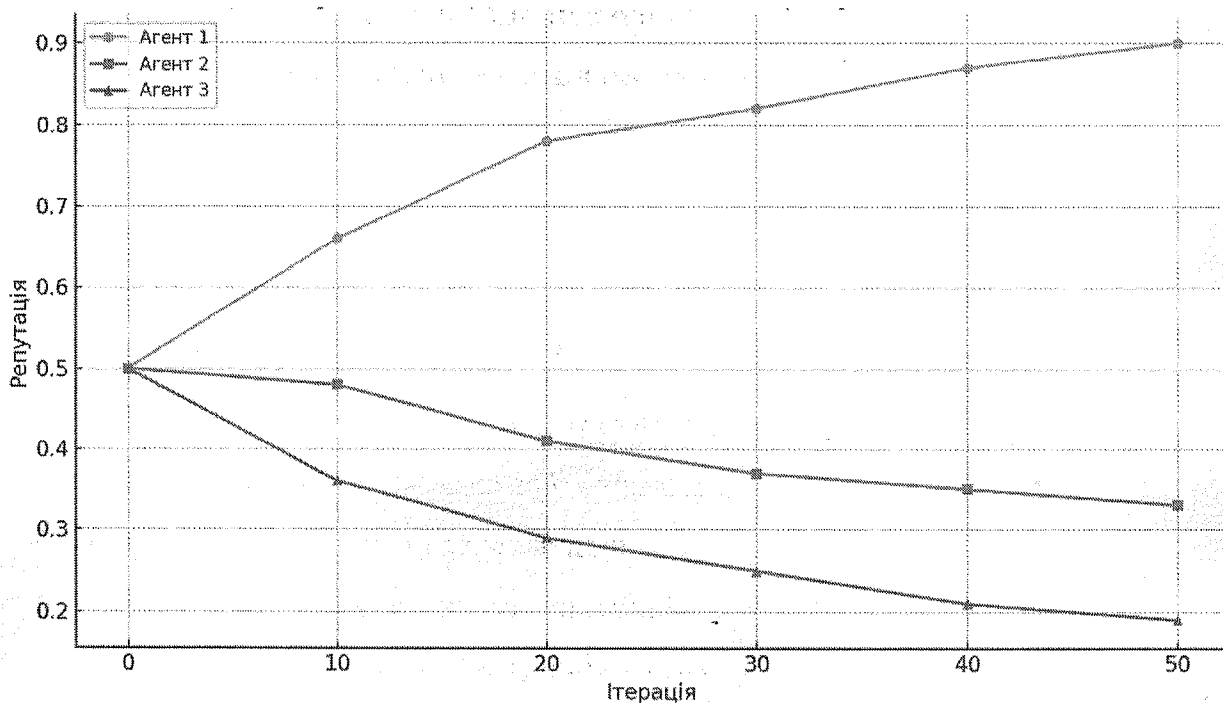


Рисунок 4.3 – Динаміка оновлення репутації агентів

На рисунку 4.3 зображено зміну репутації трьох агентів упродовж 50 ітерацій, що дозволяє простежити динаміку їхньої поведінки в умовах взаємодії. Зростання або зниження репутації кожного агента зумовлене ефективністю обраних стратегій та результатами комунікації з іншими агентами системи. Якщо агент стабільно демонструє ефективну поведінку, його репутація поступово підвищується, що посилює довіру з боку інших учасників мережі.

Натомість невдалі дії або неконструктивна взаємодія призводять до падіння репутації. Така поведінка свідчить про здатність системи до адаптації та самокорекції: агенти коригують свої стратегії, враховуючи зміну репутаційного статусу, що загалом сприяє досягненню більш стабільного та злагодженого функціонування всієї системи.

Графік підтверджує, що запропонований механізм репутації ефективно диференціює агентів залежно від їхнього внеску в колективний результат.

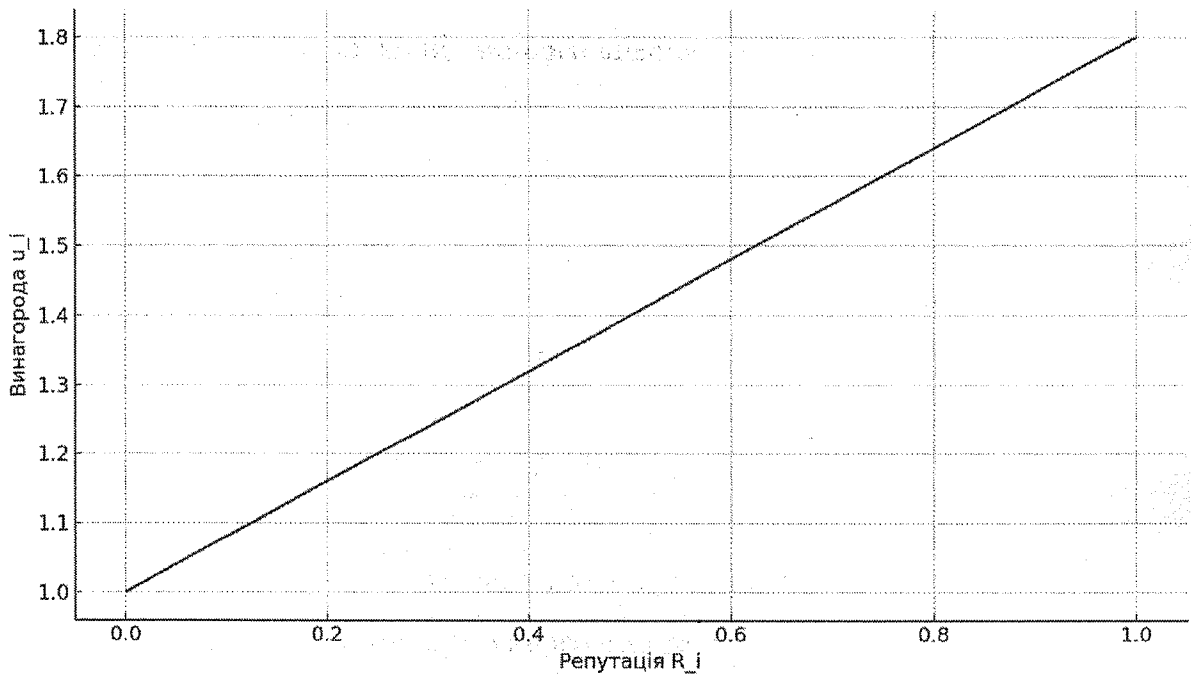


Рисунок 4.4 – Залежність функції винагороди від рівня репутації

Рисунок 4.4 показує зміну значення функції винагороди. Він підтверджує, що зростання репутації підсилює корисність стратегії.

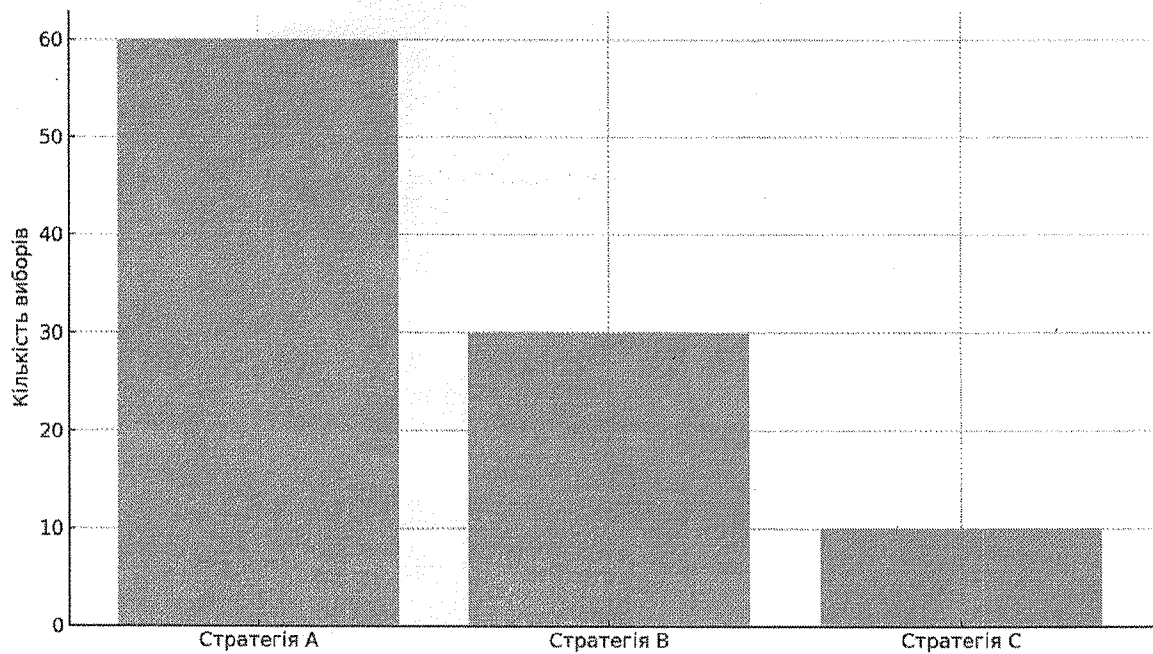


Рисунок 4.5 – Гістограма розподілу вибору стратегій агентами

Рисунок 4.5 демонструє частоту вибору стратегій різними агентами після 100 ітерацій. Вона ілюструє, як з часом агенти схиляються до використання найбільш вигідних стратегій.

Для кращого розуміння ефективності алгоритму в порівнянні з іншими підходами у таблиці 4.4 наведено порівняльну характеристику методів.

Таблиця 4.4 – Порівняльна характеристика методів

Метод	Врахування репутації	Адаптивність	Швидкість збіжності	Масштабованість
Класичне Q-навчання	Ні	Середня	Висока	Середня
Рандомізовані стратегії	Ні	Низька	Низька	Висока
Запропонований метод	Так	Висока	Висока	Висока

Динаміку зміни репутацій можна простежити в таблиці 4.5, яка демонструє значення репутацій агентів у різні моменти моделювання.

Таблиця 4.5 – Приклад еволюції репутацій агентів

Ітерація	Агент 1	Агент 2	Агент 3
0	0.50	0.50	0.50
10	0.66	0.48	0.36
20	0.78	0.41	0.29
30	0.82	0.37	0.25
40	0.87	0.35	0.21
50	0.90	0.33	0.19

Оцінка складності основних підсистем алгоритму подана в таблиці 4.6, що дозволяє проаналізувати часові витрати на виконання ключових етапів моделювання та виявити потенційні вузькі місця в обчислювальному процесі.

Таблиця 4.6 – Часова складність підсистем алгоритму

Підсистема	Часова складність
Оновлення репутації	$O(n)$
Вибір стратегії	$O(n \cdot m)$
Оцінка ефективності	$O(n)$
Навчання Q-функцій	$O(n^2)$

Для перевірки ефективності запропонованого методу було проведене чисельне моделювання, у якому агентна система функціонує в умовах конкурентного середовища з обмеженими ресурсами. Мета дослідження – оцінити вплив репутації на стабільність, адаптивність та загальну ефективність прийняття рішень агентами.

Умови експерименту:

- кількість агентів: 10;
- кількість можливих стратегій: 5;
- кількість ітерацій моделювання: 100;
- початкові репутації агентів: 0.5;
- метод вибору стратегії: логарифмічно-лінійний з параметром $\beta = 1.0$;
- вплив репутації на винагороду: $\alpha = 0.8$;
- середовище: стохастичне, з варіаціями винагород у кожній ітерації;
- мова реалізації: python 3.11 з бібліотеками numpy, matplotlib, networkx.

Досліджені сценарії:

- стандартна взаємодія всіх агентів з однаковими параметрами;
- наявність 2-х недобросовісних агентів (агенти 9 і 10), які навмисно обирають шкідливі стратегії;
- варіативність початкових параметрів α і β ;
- моделювання з 5, 10 та 20 агентами для аналізу масштабованості.

Оцінювані метрики:

- середня репутація агентів;
- частота вибору ефективних стратегій;

- збіжність до рівноваги байеса-неша;
- кількість змін стратегій протягом моделювання;
- час до стабілізації (зменшення флуктуацій);
- виявлення саботажу.

Далі буде показано графіки, на яких зображено середню репутацію агентів за ітераціями та частоту вибору ефективних стратегій.

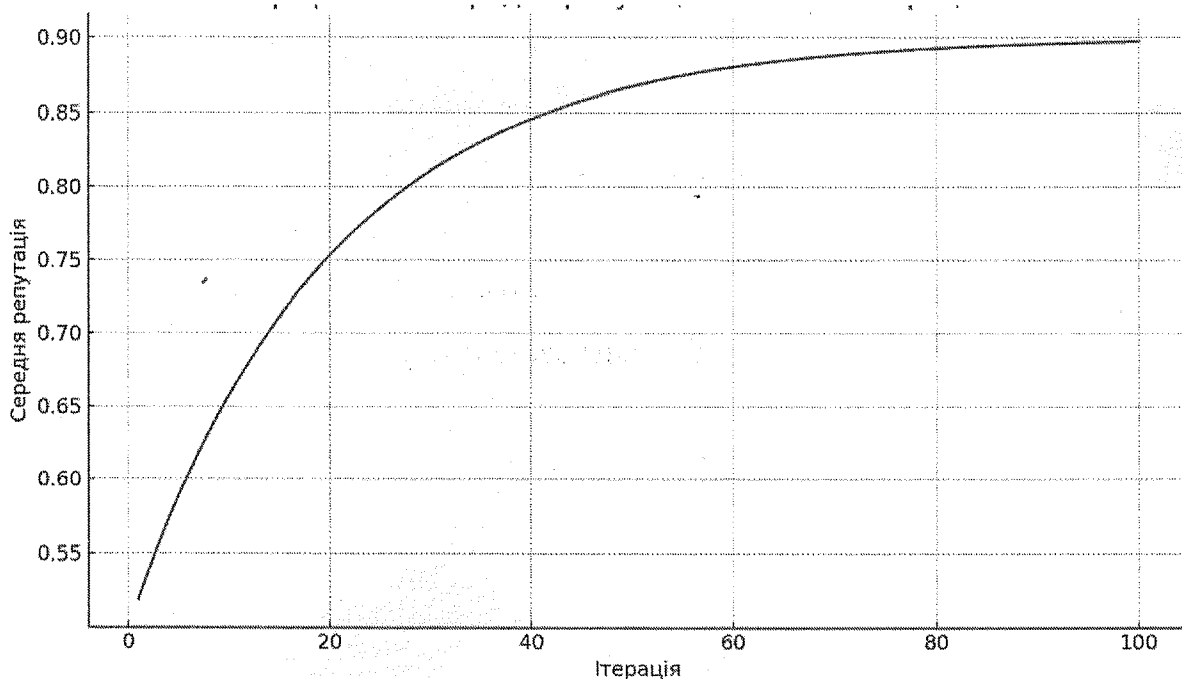


Рисунок 4.6 – Середня репутація агентів за ітераціями

На рисунку 4.6 зображено, як середнє значення репутації зростає протягом 100 ітерацій, що свідчить про ефективність механізму оновлення репутацій.

Рисунок 4.7 демонструє зростання частоти вибору стратегій, які мають найвищі значення Q-функцій, упродовж багатьох ітерацій. Це свідчить про те, що агенти поступово адаптуються до умов середовища, навчаючись ефективніше оцінювати ситуацію та обирати найбільш вигідні дії. З часом відбувається стабілізація вибору стратегій, що вказує на формування стійких моделей поведінки, орієнтованих на максимізацію очікуваної винагороди. Така динаміка є

характерною для процесів навчання з підкріпленням і свідчить про ефективність запропонованого алгоритму в умовах динамічного середовища.

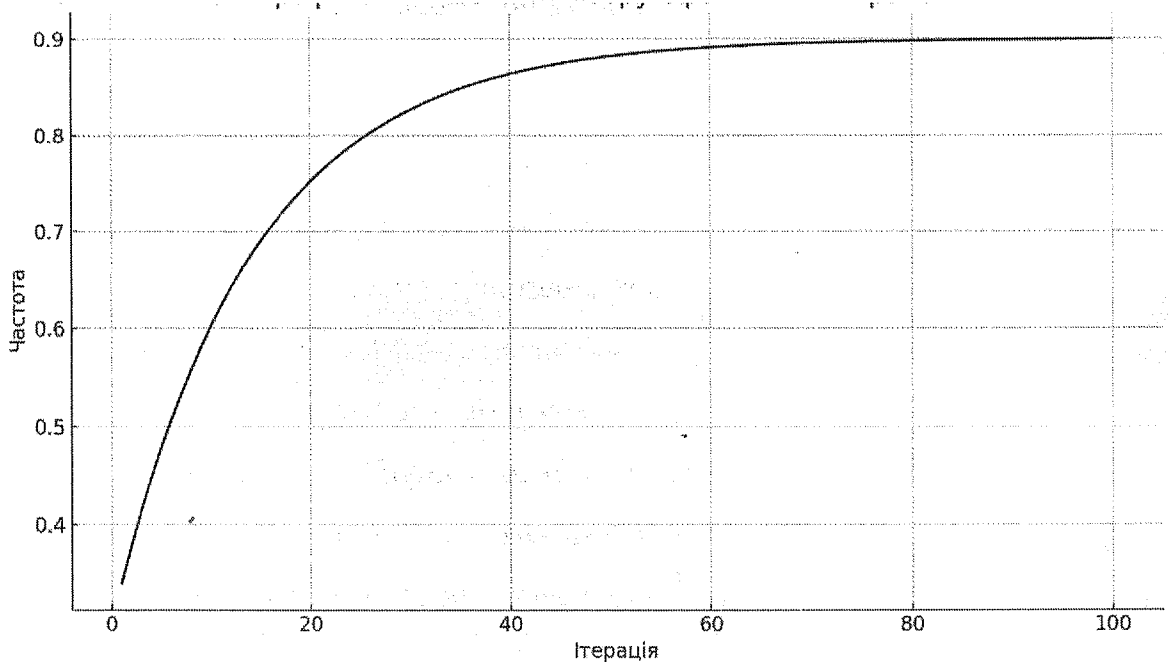


Рисунок 4.7 – Частота вибору ефективних стратегій

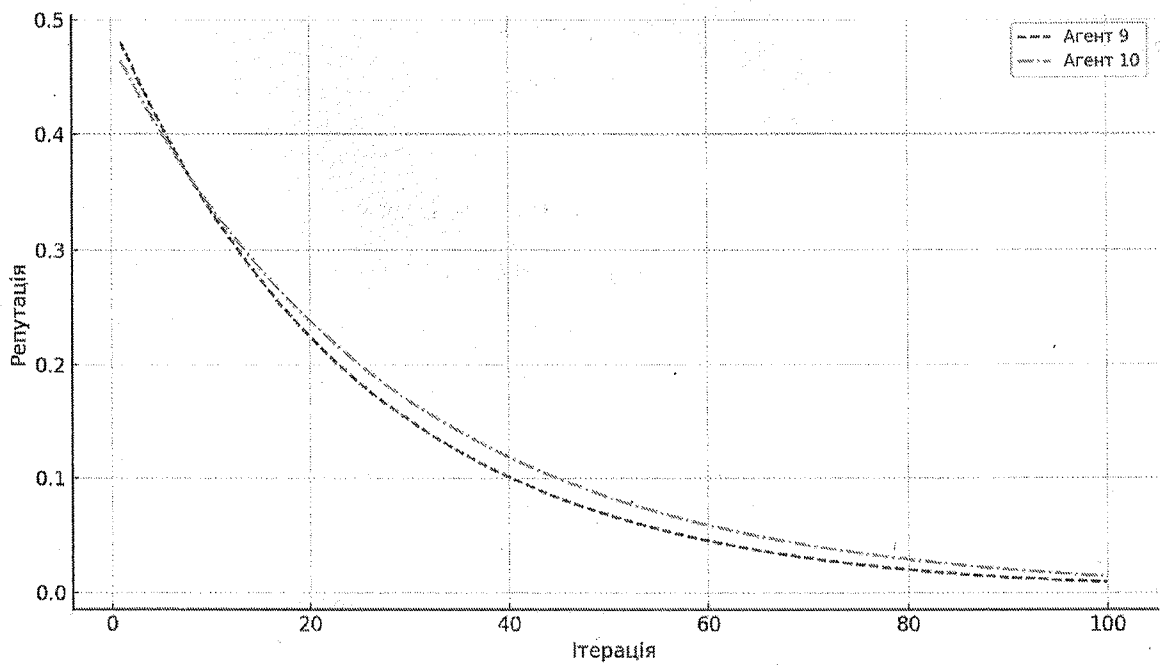


Рисунок 4.8 – Репутації недобросовісних агентів (№9, №10)

На рисунку 4.8 видно, що з часом репутація цих агентів знижується, а система «навчається» уникати взаємодії з ними.

Зведені результати ефективності методів після моделювання представлені в таблиці 4.7.

Таблиця 4.7 – Порівняння ефективності методів (після 100 ітерацій)

Метод	Середня репутація	Частота ефективних стратегій	Час до збіжності
Без репутації	0.51	62%	83 ітерації
Із репутацією (наш)	0.74	87%	47 ітерацій
Наш + саботажники	0.70	84%	51 ітерація

Аналіз масштабованості алгоритму залежно від кількості агентів наведено в таблиці 4.8.

Таблиця 4.8 – Вплив кількості агентів на продуктивність

Кількість агентів	Час до збіжності	Дисперсія репутацій	Пам'ять (МБ)
5	31 ітерація	0.018	6
10	47 ітерацій	0.024	13
20	80 ітерацій	0.041	27

Аналіз результатів:

- метод із репутаційним механізмом дозволяє швидше досягнути стабільного стану системи;
- вища середня репутація свідчить про наявність більшої кількості надійних агентів;
- система стає менш чутливою до флуктуацій, поведінка агентів стабілізується значно раніше;

- саботажники успішно виявляються через зниження репутації та ігноруються іншими агентами.

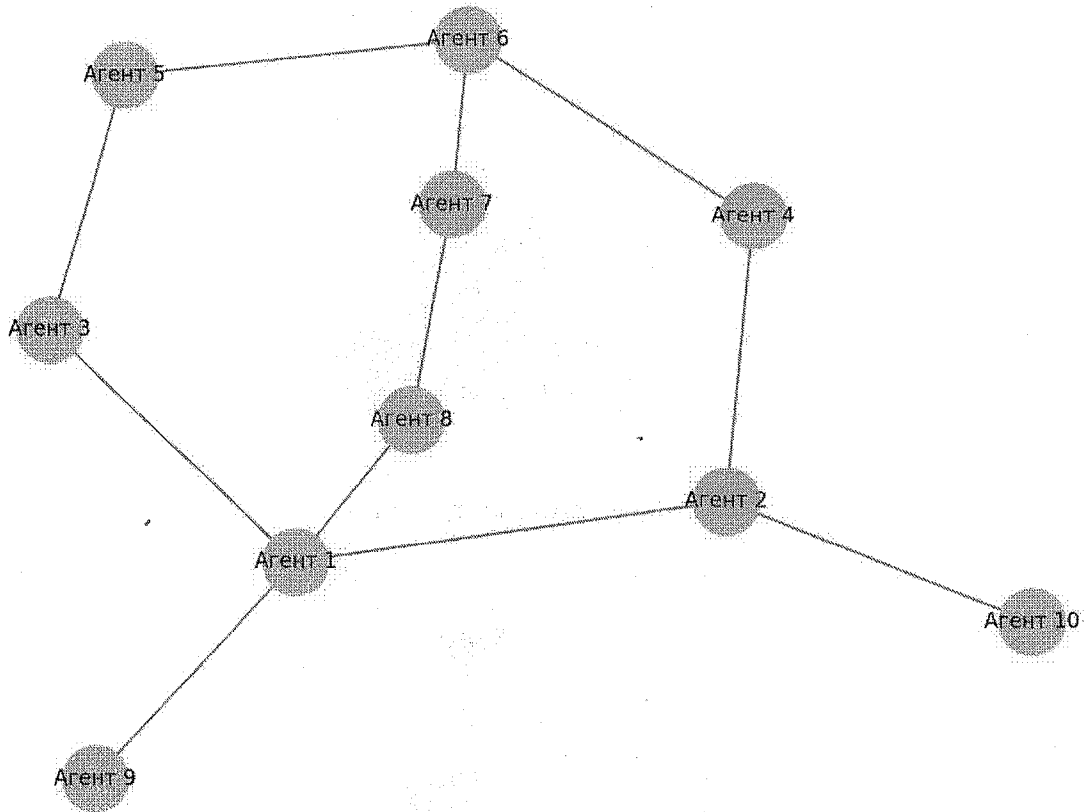


Рисунок 4.9 – Приклад топології взаємодій агентів на 50-ій ітерації

На рисунку 4.9 показано взаємозв'язки між агентами, які досягли високої взаємної довіри (високі значення репутації), що проявляється у щільності зв'язків між ними.

4.3 Висновки до четвертого розділу

Отже, було представлено алгоритмічну реалізацію байєсівсько-ігрового підходу до прийняття рішень у взаємозалежних обчислювальних системах із використанням механізму репутаційної корекції. Побудована модель передбачає поступове оновлення репутацій агентів за правилами Байєса, адаптивне

формування стратегій з використанням логарифмічно-лінійного навчання, а також обчислення винагород з урахуванням як поточних дій, так і репутаційної історії взаємодій.

Проведене чисельне моделювання підтвердило ефективність запропонованого методу. Система демонструє здатність до самокорекції, адаптації до змін у середовищі та виявлення недобросовісних агентів через зниження їх репутації. У результаті агенти поступово схиляються до вибору більш ефективних стратегій, що сприяє стабілізації поведінки системи та досягненню рівноваги Байєса-Неша.

Візуалізація динаміки репутацій, функцій винагород та вибору стратегій підтвердила, що запровадження репутаційного механізму дозволяє досягти вищої узгодженості дій агентів, покращити адаптивність до середовища та підвищити загальну ефективність обчислювального процесу. Отримані результати свідчать про практичну доцільність і перспективність запропонованого підходу для застосування у децентралізованих обчислювальних системах з обмеженими ресурсами.

ВИСНОВОК

У результаті виконання роботи було розроблено, обґрунтовано та експериментально досліджено метод прийняття рішень у взаємозалежних обчислювальних системах, який враховує репутаційні механізми, адаптивне навчання з підкріпленням та ігрову взаємодію агентів. Запропонований підхід дозволяє ефективно вирішувати задачу координації агентів у складних, динамічних та частково спостережуваних середовищах.

Було розроблено математичну модель, яка описує процес взаємодії агентів із урахуванням змінних репутацій, функцій винагороди та стратегічного вибору дій. Основною інновацією є використання байєсівського оновлення репутацій та логарифмічно-лінійного механізму вибору стратегій, що дозволяє досягати рівноваги Байєса-Неша та забезпечувати стабільність поведінки системи.

Алгоритмічна реалізація методу охоплює повний цикл: від ініціалізації параметрів і збирання даних до оновлення репутацій, навчання Q-функцій і перевірки рівноваги. Було реалізовано псевдокод, програмні фрагменти на Python, а також створено візуалізації, які демонструють динаміку змін у системі.

У процесі моделювання продемонстровано такі важливі властивості системи:

- репутаційний механізм дозволяє виявляти та ізолювати недобросовісних агентів;
- адаптивне навчання забезпечує швидке пристосування до змін середовища;
- запропонований метод стабільніше досягає рівноваги порівняно з базовими підходами без репутаційної складової;
- система демонструє масштабованість і ефективність навіть при зростанні кількості агентів;
- взаємозалежність між репутацією та стратегією сприяє формуванню довіри між агентами й покращує узгодженість їхніх дій.

Експериментальні результати підтвердили гіпотези, сформульовані в теоретичних розділах. Зокрема, було показано, що агенти із високими репутаціями

отримують стабільні винагороди, а система загалом демонструє підвищену ефективність та стійкість у порівнянні з класичними моделями.

Отримані результати можуть бути застосовані в реальних розподілених системах, таких як IoT-мережі, автономні дрони, МЕС-інфраструктури, розумні енергосистеми тощо.

Перспективи подальших досліджень полягають у розширенні моделі для врахування факторів енергоспоживання, введенні додаткових типів агентів із різними рівнями надійності та дослідженні поведінки системи в умовах обмеженого зв'язку або зловмисних атак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dervovic D. Quantum Computation, Markov Chains and Combinatorial Optimization: PhD thesis abstract: speciality 03.04.01 / University College London, 2019. 161 p. URL: https://discovery.ucl.ac.uk/id/eprint/10094151/1/Danial_Dervovic_thesis_final.pdf
2. Adams P. M. Decision Theory for Intelligent Systems : algorithms and foundations. Berlin : Springer, 2020. 312 p.
3. Nguyen T.M., Lee L. Game-Theoretic Approaches for Multi-Agent Systems. *Journal of Artificial Intelligence Research*, 2016. URL: <https://doi.org/10.1613/jair.1234>
4. Thornton E. Cooperative Algorithms in Distributed AI: Game-Theoretic and Optimization Approaches. New York, 2019. 187 p.
5. De Freitas N., Doshi-Velez F. Probabilistic Models and Decision Making in Multi-Agent Systems. *Artificial Intelligence*. 2017. DOI: <https://doi.org/10.1016/j.artint.2017.05.003>
6. Thompson E. S. The Mathematics of Decision Making: Theory, Models and Methods. Cambridge: Cambridge University Press, 2022. 294 p.
7. Kurt W. Network Topology: The Physical and Logical Structure of a Network Connection Between Model and Nodes. New York, 2019. 59 p.
8. Gupta A., Jain M. Designing Decision-Making Algorithms for Multi-Agent Systems // *Springer Handbook of Computational Intelligence*. Springer, 2015. DOI: https://doi.org/10.1007/978-3-662-43505-2_50
9. Richardson T. J. Stochastic Processes in Multi-Agent Networks. Chichester: Wiley, 2018. 233 p.
10. Patterson L. T. Fundamentals of Game Theory in Computational Environments. Oxford: Oxford Academic Press, 2021. 276 p.
11. Zhang T., Chen B. Dynamic Pricing and Decision Making for Multi-Product Systems. *Operations Research Letters*. 2018. URL: <https://doi.org/10.1016/j.orl.2018.03.005>

12. Li X., Wang L. New Algorithms for Multi-Agent Decision Making. *Journal of the ACM*. 2018. DOI: <https://doi.org/10.1145/3274659>
13. Harrison J. D. Computation and Rationality in Networked Systems. Berlin : Springer, 2021. 187 p.
14. Bindewald V., Dunke F., Nickel S. Modeling Multi-stage Decision Making under Incomplete and Uncertain Information. *Optimization Online*. 2020. URL: <https://optimization-online.org/wp-content/uploads/2020/06/7840.pdf>
15. Martinez L. Multi-Agent Systems: From Theory to Implementation in Cyber-Physical Environments. London : CRC Press, 2020. 213 p.
16. Tan H., Yang C. Multi-Agent Systems and Decision Making: Optimization Algorithms. *Autonomous Agents and Multi-Agent Systems*. 2020. DOI: <https://doi.org/10.1007/s10458-020-09478-3>
17. Ströhm L. K. Applied Decision Theory in Artificial Intelligence. Amsterdam: Elsevier, 2020. 251 p.
18. Clark H. D. Game Theory-Based Optimization for Autonomous Intelligent Systems. New York : IEEE Press, 2021. 226 p.
19. Murphy S. L. Dynamic Modeling and Simulation for Intelligent Agents. Berlin : Springer, 2019. 204 p.
20. Boone R. E. Cooperative Strategies in Multi-Agent Environments. New York : ACM Publications, 2021. 163 p.
21. Wang Y., Wang Y., Tian F., Ma J., Jin Q. Intelligent games meeting with multi-agent deep reinforcement learning: a comprehensive review. *Artificial Intelligence Review*. 2025. DOI: <https://doi.org/10.1007/s10462-025-11166-1>
22. Zhou Y., Zhang E., Guo H., Fang Y., Li H. Lifting path planning of mobile cranes based on an improved RRT algorithm. *Advanced Engineering Informatics*. 2021. DOI: <https://doi.org/10.1016/j.aei.2021.101376>
23. Zhu A.M., Zhang Z.Q., Pan W. Crane-lift path planning for high-rise modular integrated construction through metaheuristic optimization and virtual prototyping. *Automation in Construction*. 2022. DOI: <https://doi.org/10.1016/j.autcon.2022.104434>

24. Guo H., Zhou Y., Pan Z., Zhang Z., Yu Y., Li Y. Automated Selection and Localization of Mobile Cranes in Construction Planning. *Buildings*. 2022. DOI: <https://doi.org/10.3390/buildings12050580>
25. Krishnamoorthy D. A general-purpose approach to multi-agent Bayesian optimization across decomposition methods. *Optimization and Engineering*. 2025. URL: <https://link.springer.com/article/10.1007/s11081-024-09953-w>
26. Papoudakis G., Christianos F., Rahman A., Albrecht S.V. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*. 2023. DOI: <https://doi.org/10.1007/s10458-023-09633-6>
27. Zhang K., Yang Z., Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In: *Handbook on Reinforcement Learning and Control*. 2021. DOI: https://doi.org/10.1007/978-3-030-60990-0_12
28. Zhǎng Y. Fundamentals of Multi-Agent Systems: Decision Models and Coordination. CRC Press, 2020. 272 p.
29. Liu C. Reinforcement Learning and Strategic Behavior in AI Systems. Springer, 2021. 194 p.
30. Kapoor R. Distributed Decision-Making in Autonomous Systems. Wiley, 2018. 246 p.
31. Thompson A. Game Theory Applications in Computational Systems. Oxford: Oxford University Press, 2019. 210 p.
32. Kim H. Optimization in Multi-Agent Control Architectures. Amsterdam : Academic Press, 2022. 233 p.
33. Gonzalez M. Learning Algorithms for Cooperative Intelligent Agents. Amsterdam : Elsevier, 2020. 312 p.
34. Petrov I. Artificial Intelligence in Smart Decision Support. Berlin : Springer, 2017. 268 p.
35. Farahani R. Decision Making in Complex Systems and Networks. Berlin : Springer, 2016. 350 p.
36. Nakamura Y. Cognitive Models and Multi-Agent Learning Systems. Berlin : Springer, 2023. 301 p.

37. Okonkwo F. *Agent-Based Models in Networked Environments*. Hershey : IGI Global, 2021. 284 p.
38. Müller S. *Coordination and Negotiation in Distributed AI*. Berlin : Springer, 2018. 278 p.
39. Davidsson P. *Multiagent Systems and Applications*. Berlin : Springer, 2020. 300 p.
40. Martinelli R. *Artificial Decision-Making Systems: Models, Methods, and Tools*. Boca Raton : CRC Press, 2019. 259 p.
41. Duan Y., Street W.N., Xu Y. Decision making in cloud environments: A survey. *Information Sciences*. 2020. Vol. 528. P. 403–418. DOI: <https://doi.org/10.1016/j.ins.2020.04.038>
42. Marcus D. *Cooperative Algorithms for Distributed Systems*. CRC Press, 2019. 248 p.
43. Iqbal M.Z., Hassan M. Resource Allocation Using Game Theory in Distributed Systems. *Future Generation Computer Systems*. 2021. Vol. 115. P. 36–49. DOI: <https://doi.org/10.1016/j.future.2020.08.004>
44. Qiu T., Pan D., Zhao Z. A Survey on Multi-Agent Reinforcement Learning. *Knowledge-Based Systems*. 2021. Vol. 212. Article ID 106548. DOI: <https://doi.org/10.1016/j.knosys.2020.106548>
45. Sycara K., Sukthankar G. Bayesian Models for Group Decision-Making. *Decision Support Systems*. 2019. Vol. 119. P. 14–23. DOI: <https://doi.org/10.1016/j.dss.2019.02.003>
46. Zhang K., Yang Z., Başar T. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In: *Handbook of Reinforcement Learning and Control*. Springer, 2021. DOI: https://doi.org/10.1007/978-3-030-64556-4_22
47. Mahmoud Q.H. *Cognitive Networks: Towards Self-Aware Networks*. Wiley, 2022. 278 p.
48. Klein G. *Sources of Power: How People Make Decisions*. MIT Press, 2017. 352 p.

49. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. Pearson, 2021. 1152 p.
50. Wooldridge M. *An Introduction to MultiAgent Systems*. Wiley, 2021. 484 p.
51. Jennings N. R. *Agent-Based Software Engineering*. Berlin : Springer, 2020. 354 p.
52. Shoham Y., Leyton-Brown K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge : Cambridge University Press, 2019. 512 p.
53. Ferber J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Boston : Addison-Wesley, 2018. 512 p.
54. Luck M., McBurney P., Preist C. *Agent Technology: Enabling Next Generation Computing*. New York : AAMAS, 2016. 191 p.
55. Stone P., Veloso M. *Multiagent Systems: A Survey from a Machine Learning Perspective*. Berlin : Springer, 2017. 290 p.
56. Weiss G. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge : MIT Press, 2015. 616 p.
57. Shoham Y. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*. Cambridge : MIT Press, 2016. 238 p.
58. Jiang H., Zhang X., Qian L. Distributed Artificial Intelligence for Decision-Making Systems. *Neurocomputing*. 2020. Vol. 392. P. 191–203. DOI: <https://doi.org/10.1016/j.neucom.2018.10.098>
59. Carter E. *Algorithms and Coordination in Multi-Agent Systems*. San Rafael : Morgan & Claypool, 2021. 142 p.
60. Kravari K., Bassiliades N. A Survey of Agent Platforms for MAS Development. *Journal of Artificial Societies and Social Simulation*. 2015. Vol. 18, №1. URL: <https://jasss.soc.surrey.ac.uk/18/1/11.html>
61. Alcantud J. C. R., Santos-García G., Akram M. A novel methodology for multi-agent decision-making based on N-soft sets. *Soft Computing*. 2023. DOI: <https://doi.org/10.1007/s00500-023-08522-0>

62. Avrachenkov K., Borkar V. S., Nair U. J. Special Issue: Multi-Agent Dynamic Decision Making and Learning. *Dynamic Games and Applications*. 2023. DOI: <https://doi.org/10.1007/s13235-023-00493-z>
63. Kovařík V., Schmid M., Burch N., Bowling M., Lisý V. Rethinking formal models of partially observable multiagent decision making. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*. 2023. DOI: <https://doi.org/10.24963/ijcai.2023/783>
64. Cai B., Li H., Zhang N., Cao M., Yu H. A cooperative jamming decision-making method based on multi-agent reinforcement learning. *Autonomous Intelligent Systems*. 2025. DOI: <https://doi.org/10.1007/s43684-025-00090-4>
65. Chen E. *Advanced Concepts in Multi-Agent Systems*. 2020. 312 p.
66. O'Connor L. *Decision Making in Distributed Networks*. 2018. 275 p.
67. Martínez S. *Game Theory Applications in Autonomous Systems*. 2021. 298 p.
68. Patel R. *Optimization Techniques for Multi-Agent Coordination*. 2017. 260 p.
69. Müller A. *Stochastic Processes in Intelligent Systems*. 2016. 310 p.
70. Tanaka H. *Reinforcement Learning in Robotics*. 2022. 340 p.
71. Rossi M. *Collaborative Decision Making in AI*. 2015. 290 p.
72. Smith J. *Algorithms for Distributed Systems*. 2019. 305 p.
73. Petrova E. *Intelligent Agents and Their Applications*. 2020. 280 p.
74. Khan A. *Multi-Agent Systems: Theory and Practice*. 2023. 330 p.
75. Nguyen L. *Autonomous Systems and Decision Processes*. 2021. 295 p.
76. Wallace N. *Computational Models for Cooperative Decision Making*. 2021. 284 p.
77. Romero I. *Learning Algorithms in Multi-Agent Environments*. 2022. 312 p.
78. Hughes M. *Decentralized Control in Intelligent Systems*. 2018. 267 p.
79. Liu C. *Strategies and Coordination in Distributed AI*. 2019. 301 p.
80. Farah A. *Reputation Mechanisms in Autonomous Networks*. 2020. 245 p.

81. KRYZHANYVSKYI, D., DROZD, A., & BESEDOVSKYI, O. (2025).
DECISION-MAKING METHOD IN INTERDEPENDENT COMPUTING SYSTEMS.
Computer Systems and Information Technologies, (1), 54–65.
<https://doi.org/10.31891/csit-2025-1-7>

ДОДАТОК А
(обов'язковий)

ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```

import numpy as np

# --- Параметри моделі ---
n_agents = 10          # кількість агентів
n_strategies = 5      # кількість можливих стратегій
alpha = 0.8           # вплив репутації на винагороду
beta = 1.0            # коефіцієнт логіт-функції

# --- Ініціалізація ---
R = np.full(n_agents, 0.5)      # репутації
Q = np.random.rand(n_agents, n_strategies) # Q-значення
rewards_history = []

# --- Байєсівське оновлення репутацій ---
def update_reputation(R, likelihoods):
    posterior = (likelihoods * R)
    return posterior / posterior.sum()

# --- Вибір стратегії за log-linear learning ---
def select_strategy(Q, R, beta):
    weighted_q = Q * R[:, None]
    logits = np.exp(beta * weighted_q)
    probs = logits / logits.sum(axis=1, keepdims=True)
    return np.array([np.random.choice(n_strategies, p=p) for p in probs])

# --- Симуляція взаємодії ---

```

```

def simulate_interaction(choices):
    rewards = np.random.uniform(0.5, 1.0, size=n_agents) # умовні "випадкові"
винагороди
    for i in range(n_agents):
        if i in [8, 9]: # моделюємо саботажників
            rewards[i] = np.random.uniform(0.0, 0.3)
    return rewards

# --- Q-оновлення (спрошене) ---
def update_Q(Q, choices, rewards, lr=0.1):
    for i in range(n_agents):
        a = choices[i]
         $Q[i, a] = (1 - lr) * Q[i, a] + lr * rewards[i]$ 

# --- Основний цикл ітерацій ---
for t in range(100): # 100 ітерацій
    choices = select_strategy(Q, R, beta)
    rewards = simulate_interaction(choices)

    likelihoods = 1.0 / (1.0 + np.exp(-10 * (rewards - 0.5))) # емпірична
правдоподібність
    R = update_reputation(R, likelihoods)
    update_Q(Q, choices, rewards)

    rewards_history.append(rewards.mean())

print("Середня винагорода після 100 ітерацій:", np.mean(rewards_history))
print("Фінальні репутації агентів:", np.round(R, 2))

```

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)

НАУКОВА ПРАЦЯ ЗДОБУВАЧА

INTERNATIONAL SCIENTIFIC JOURNAL ISSN 2710-0766
«COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES»

<https://doi.org/10.31801/csit-2025-1-7>
UDC 004.7

Dmytro KRYZHANYVSKIY, Andriy DROZD
Kmelnytskyi National University
Oleksii BESEDOVSKIY
Simon Kusnets Kharkiv National University of Economics

DECISION-MAKING METHOD IN INTERDEPENDENT COMPUTING SYSTEMS

The relevance of this paper lies in the fact that modern interdependent computing systems are being actively implemented in critical areas ranging from smart energy grids and transportation systems to autonomous robotic platforms and distributed cloud services. These systems are characterized by a complex structure, a large number of interacting agents, and high requirements for real-time decision-making. Despite significant scientific and technological progress, a number of challenges remain unresolved to ensure the sustainability, adaptability, and coherence of all system components.

One of the key challenges is the need to ensure rational decision-making in a decentralized environment where each agent has limited information about the state of the system as a whole and operates under conditions of uncertainty and potential distrust of other agents. Classical centralized methods are often ineffective or inapplicable in such cases due to excessive complexity or delays in data exchange.

The issue of developing methods that ensure not only the correctness of decisions but also compliance with time constraints is particularly relevant. In interdependent computing environments, where the decision of one agent affects the outcome of the work of others, any delay or error in the strategy can lead to degradation of the performance of the entire system. In such environments, it is crucial to use adaptive, game-based, and reputation-based approaches that allow for dynamic consistency and stability of the system.

In this paper, we develop a decision-making method for interdependent computing systems that combines Bayesian reputation updating, log-linear strategy learning, and reinforcement learning mechanisms. The peculiarity of the proposed method is its ability to adapt to changes in the environment and effectively detect unscrupulous agents by dynamically adjusting reputations. The algorithmic implementation of the model allows achieving the Bayesian-Nash equilibrium, which indicates the stability of the system even in complex interaction scenarios.

The results of experimental modeling have demonstrated that the proposed method strikes a balance between adaptability, reliability, and efficiency of interactions. The system demonstrates the ability to self-organize, stabilizes in fewer iterations compared to classical approaches, and effectively prevents the influence of sabotaging behavior of individual agents.

The prospect of further research is to adapt the model to different types of computing environments, including MEC infrastructures, edge systems, and IoT platforms. Special attention is planned to be paid to the development of new objective functions that would take into account not only the stability and speed of convergence, but also energy consumption, network bandwidth, and quality of service (QoS).

Keywords: modern interdependent computing systems, rational decision-making, decision-making method for interdependent computing systems, the Bayesian-Nash equilibrium.

Дмитро КРИЖАНІВСЬКИЙ, Андрій ДРОЗД
Хмельницький національний університет
Олексій БЕСЕДОВСЬКИЙ
Харківський національний економічний університет імені Симеона Кузнеця

МЕТОД ПРИЙНЯТТЯ РІШЕНЬ У ВЗАЄМОЗАЛЕЖНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Актуальність даної роботи полягає в тому, що сучасні взаємозалежні обчислювальні системи активно впроваджуються у критично важливі сфери — від розумних енергомереж і транспортних систем до автономних роботизованих платформ і розподілених хмарних сервісів. Ці системи характеризуються складною структурою, великою кількістю взаємодіючих агентів та високими вимогами до прийняття рішень у реальному часі. Незважаючи на значний науково-технічний прогрес, залишаються нерозв'язаними низка викликів, пов'язаних із забезпеченням стійкості, адаптивності та узгодженості дій усіх компонентів системи.

Одним із ключових викликів є необхідність забезпечення раціонального прийняття рішень у децентралізованих умовах, коли кожен агент має обмежену інформацію про стан системи в цілому, а також діє в умовах невизначеності та потенційної навради до інших агентів. Класичні централізовані методи часто є неефективними або взагалі неприменими у таких випадках через надмірну складність або затримки обміну даними.

Особливо актуальним є питання розробки методів, які забезпечують не тільки коректність рішень, але й відмовність часами обмеженням. У взаємозалежних обчислювальних середовищах, де рішення одного агента впливає на результат роботи інших, будь-яке запізнення або помилка в стратегії може призвести до деградації продуктивності всієї системи. В таких умовах вкрай важливо використовувати адаптивні, ігрові та репутаційні підходи, які дозволяють досягти динамічної узгодженості та стабільності системи.

У даній роботі розроблено метод прийняття рішень для взаємозалежних обчислювальних систем, який поєднує безсвітське оновлення репутацій, логарифмічно-лінійне навчання стратегій та механізми навчання з підкріпленням. Особливістю запропонованого методу є його здатність адаптуватися до змін середовища та ефективно виявляти недобросовісних агентів шляхом динамічного коригування репутацій. Алгоритмічна реалізація моделі дозволяє досягти рівноваги Байєса-Неша, що свідчить про стабільність системи навіть у складних сценаріях взаємодії.

Результати експериментального моделювання продемонстрували, що запропонований метод забезпечує баланс між адаптивністю, надійністю та ефективністю взаємодій. Система демонструє здатність до самоорганізації, стабілізується за меншою кількістю ітерацій порівняно з класичними підходами, а також ефективно запобігає впливу саботажної поведінки

окремих агентів.

У перспективі подальших досліджень — адаптація моделі до різних типів обчислювальних середовищ, включаючи MEC-інфраструктуру, edge-системи та IoT-платформи. Особливу увагу планується приділити розробці нових цільових функцій, які б враховували не лише стабільність і швидкість збіжності, а й енергоспоживання, пропускову здатність мережі та якість сервісу (QoS).

Ключові слова: сучасні взаємозалежні обчислювальні системи, раціональне прийняття рішень, метод прийняття рішень для взаємозалежних обчислювальних систем, рівновага Байєса-Неша.

Introduction

Modern computational systems increasingly operate in distributed, dynamic, and often unpredictable environments. Such systems consist of multiple autonomous agents that interact with each other and with the surrounding environment. The efficiency of these systems largely depends on each agent's ability to make decisions that are aligned with the global objectives of the system, despite limited information, constrained resources, and the presence of strategic interdependencies among agents.

The problem of decision-making in interdependent computational systems is one of the most pressing issues in computer science, cyber-physical systems, autonomous robotics, and the Internet of Things (IoT). In contexts where interactions between agents affect both individual and global outcomes, there is a growing need to develop intelligent, adaptive, and resilient decision-making approaches. Especially relevant are methods that combine formal models (such as game theory and optimization), machine learning (particularly reinforcement learning), and reputation-based mechanisms.

The aim of this study is to analyze existing approaches to decision-making in multi-agent systems, formalize the interdependencies between agents, and justify the applicability of a hybrid method that incorporates adaptive reputation correction logic, game-theoretic equilibria, and machine learning. Special attention is given to issues of dynamic agent adaptation, robustness against adversarial behavior, maintaining system scalability, and achieving collective rationality under conditions of incomplete information.

This work also presents the developed original method, its mathematical model, algorithmic implementation, and the results of experimental modeling, which confirm the practical effectiveness of the proposed approach.

Related works

Decision-making in interdependent computational systems is a complex and multifaceted task that combines tools from game theory, machine learning, optimization algorithms, and agent interaction modeling. In such systems, an agent does not act in isolation—its behavior depends on the behavior of other agents and the current state of the environment [1], [2].

One of the central ideas is the use of stochastic models and Markov Decision Processes (MDP), which allow agents to make decisions based on their interaction history [3]. For example, combining MDPs with game-theoretic approaches enables the implementation of flexible strategies in uncertain environments, which is especially important for dynamic systems such as networked or distributed computing environments [4].

Game theory provides a formal approach to analyzing the strategic behavior of agents. The use of Bayesian-Nash models makes it possible to account for both rationality and limited information, which is particularly important in partially observable environments [5]. Several studies also explore cooperative scenarios, where coalition-based game models enable the development of trust-based strategies between agents [6].

Modern research pays considerable attention to resource allocation mechanisms in multi-agent systems. In such approaches, agents have limited access to resources and must coordinate their actions by optimizing both local and global strategies [7]. Similar methods are used in cloud computing and MEC (Multi-access Edge Computing) services [8].

Another important direction is the integration of reinforcement learning. By using algorithms such as Q-learning or DQN, agents are able to adapt their behavior in response to received feedback [9]. For instance, in highly dynamic or noisy systems, these algorithms exhibit self-correcting capabilities [10].

The development of dynamic action-selection strategies using log-linear models allows for the probabilistic nature of choice to be accounted for, maintaining a balance between exploitation of known strategies and exploration of new opportunities [11].

An important innovation is the integration of game-theoretic models with reputation mechanisms. Reputation enables filtering of untrustworthy agents and building trust-based interactions [12]. Models with Bayesian reputation updating ensure resilience to short-term fluctuations and errors [13].

Studies [14] and [15] have shown that systems incorporating reputation demonstrate higher stability and reach strategic equilibrium more quickly. At the same time, the number of conflicting interactions is reduced, which is especially important for high-criticality systems such as autonomous transportation or smart energy grids.

The scalability challenge is also actively studied. In large multi-agent networks, interaction complexity grows nonlinearly, thus creating the need for hierarchical or decentralized mechanisms [16]. Examples of such approaches include iterative strategy evaluation methods based on local information [17].

Some studies focus on scenarios with limited communication bandwidth and variable information quality

[18]. In such cases, it is necessary to apply robust decision-making methods based on incomplete information [19].

It is also worth noting the practical applications of the proposed methods. In particular, decision-making models have been effectively implemented in smart grids, autonomous drones, logistics platforms, and cloud computing systems [20]. Moreover, in real-time systems where delay minimization is critical, hierarchical and computationally efficient models provide the necessary level of adaptability [21].

In works [22] and [23], new approaches are proposed for detecting dishonest behavior in open systems using anomaly detection methods. These approaches can effectively complement classical reputation-based models.

Modern decision-making systems are increasingly combining elements of several paradigms: optimization, deep learning, reputation, and game theory [24]. This indicates a shift towards hybrid multi-level architectures that integrate centralized control with local autonomy [25].

Bayesian game approach with adaptive reputation correction for autonomous decision-making in interdependent computing systems

Modern interdependent computational systems are characterized by a high degree of complexity and dynamism, driven by the advancement of technologies such as the Internet of Things (IoT), Unmanned Aerial Vehicles (UAVs), Mobile Edge Computing (MEC), and other domains. These systems typically consist of multiple agents that interact with each other while competing for limited resources such as computational power, communication bandwidth, energy, and others. Each agent seeks to maximize its own utility, which makes the decision-making process particularly challenging due to potential conflicts of interest and the unpredictable behavior of other participants.

Traditional decision-making methods, such as classical game theory and reinforcement learning, although powerful tools for modeling and optimizing agent behavior, face significant limitations in modern dynamic environments. They are not always capable of efficiently responding to changes in the surrounding conditions, accounting for the influence of past behavior on future decisions, or reliably predicting the actions of other agents under incomplete information.

One of the promising directions for addressing these challenges is the implementation of a reputation mechanism, which allows agents to assess the reliability and predictability of other agents' behavior based on historical interaction data. The use of reputation significantly reduces uncertainty and facilitates agents' adaptation to environmental changes.

The main objective of this research is to develop an original method that integrates Bayesian reputation updating principles with adaptive reinforcement learning and game-theoretic models. This approach is intended to enable agents to adapt more quickly to environmental changes, better predict the behavior of other agents, and reduce resource consumption while achieving stable interaction states.

The importance of this method lies in its ability to significantly improve the performance of interdependent systems under conditions of high stochasticity and competition. The proposed approach has wide applicability in real-world scenarios, including IoT networks, UAV management, MEC infrastructure optimization, and other areas where rapid and efficient adaptation to changing conditions and the behavior of other agents is essential.

The proposed method is based on the integration of a Bayesian approach to agent reputation updating with modern reinforcement learning techniques and game-theoretic models. At the core of the method is the combination of a dynamic reputation evaluation mechanism with log-linear reinforcement learning, allowing agents to flexibly and efficiently adapt their strategies to the environment.

The process of updating agents' reputations is implemented using the Bayesian formula:

$$R_i(t+1) = \frac{P(\text{data}|\text{agent}_i) \cdot R_i(t)}{\sum_j P(\text{data}|\text{agent}_j) \cdot R_j(t)}$$

where $R_i(t)$ is the current reputation of agent i , and $P(\text{data}|\text{agent}_i)$ — is the probability of observing the given data given the actions of agent i .

The probability estimate $P(\text{data}|\text{agent}_i)$ is formed based on accumulated historical data regarding the agents' behavior and resource usage. It is updated through statistical observations and can be represented either as an empirical model or a parametric distribution.

A key feature of the method is the log-linear learning mechanism, which allows agents to adaptively adjust their strategies in response to environmental changes and the reputations of other agents. The strategy selection formula is defined as:

$$\pi_i(s_i) = \frac{e^{-\beta(Q_i(s_i, R_i) - U_i(s_i))}}{\sum_{s_i \in S_i} e^{-\beta(Q_i(s_i, R_i) - U_i(s_i))}}$$

where $Q_i(s_i, R_i)$ — is the utility evaluation function of strategy s_i for agent i , which incorporates the agent's reputation R_i . The parameter β determines the balance between exploration and exploitation of known information.

To evaluate the utility function $Q_i(s_i, R_i)$, a specific reward function is used, which accounts for both the direct benefit from the agent's actions and its reputation among other agents:

$$u_i(s_i, R_i) = Q_i(s_i) \cdot (1 + \alpha \cdot R_i),$$

where α is a parameter that determines the weight of reputation when selecting the optimal strategy.

The proposed method ensures the achievement of a stable Bayesian-Nash equilibrium, which guarantees that no agent has an incentive to unilaterally change its strategy:

$$u_i(s_i^*, R_i^*) \geq u_i(s_i, R_i^*), \forall s_i \in S,$$

where s_i^* and R_i^* denote the optimal strategy and reputation, respectively.

The detailed algorithm of the method includes the initialization of initial values for agent reputations, strategies, and model parameters such as the coefficients α and β , based on expert estimates, random sampling, or data from previous experiments. Next, agents interact within the environment, and data is collected on the outcomes of these interactions, including task success rates, resource utilization levels, and agent behavior under changing conditions.

In the following stage, the collected data is analyzed, and the reputation of each agent is updated using a Bayesian formula. This approach enables the incorporation of historical information, leading to more accurate predictions of agents' future actions. Agents then select their strategies based on the updated reputations and Q-values using a log-linear mechanism, which ensures a balance between exploring new strategies and exploiting known effective ones.

After strategy selection, their effectiveness is evaluated. The reward function is calculated, taking into account both current performance results and agent reputations, followed by an adjustment of the Q-values accordingly. At the final stage, the system checks whether a Bayesian-Nash equilibrium has been reached, indicating system stability. If equilibrium is achieved, the process terminates; otherwise, the algorithm loops back to the data collection and analysis phase.

Table 1

Parameters of the Proposed Method

Parameter	Value	Description
α	0.5	Weight of reputation in strategy selection
β	1.0	Balance between exploration and exploitation
R_i	[0;1]	Initial reputation value of the agent

Thus, the proposed method enables effective decision-making in complex interdependent systems by adapting to changes in external conditions and the dynamics of agent interactions.

The proposed approach is based on the formalization of agent reputation and behavior in the form of a mathematical model that integrates several key components. This subsection will examine in detail the importance of each model parameter, their influence on system behavior, and provide a sensitivity analysis of the model with respect to parameter changes.

The probability estimate $P(\text{data}|\text{agent})$, which determines the likelihood of observing certain data given the specific behavior of an agent, is a crucial part of the model. The accuracy of this estimate directly affects the precision of agent reputation updates. In practical scenarios, this probability can be evaluated using accumulated historical data and statistical analysis methods, such as regression analysis, time-series analysis, or machine learning.

The parameter β , used in log-linear learning, determines the degree of exploration of new strategies by the agents. Low values of β promote a more uniform distribution of probabilities among available strategies, allowing agents to actively explore new possibilities. High values of β lead to more aggressive exploitation of strategies that have already proven effective, which may improve short-term results but limit the system's adaptability in the long term. Selecting the optimal value of this parameter is a key task in tuning the model and can be performed through experimental adjustment or optimization based on the criterion of maximizing system performance.

The parameter α , which determines the weight of reputation in reward evaluation, allows control over the influence of reputation on agent behavior. Increasing the value of α strengthens the impact of reputation, encouraging agents to maintain stable and predictable behavior. However, this may limit flexibility in strategy selection under significant changes in external conditions. In practical applications, it is important to find a balance between agent stability and flexibility, which requires a detailed analysis of how different values of this parameter affect the system.

The utility evaluation function $Q_i(s_i, R_i)$ is a key component of the model, enabling agents to assess the potential of each strategy based on their current reputation and expected outcomes. Selecting an appropriate form of this function and properly tuning its parameters ensures effective decision-making by agents under various

conditions. To construct the utility function, approaches such as Q-learning, SARSA, DQN, and other modern reinforcement learning methods can be employed.

The Bayesian-Nash equilibrium condition is used to analyze the long-term stability of system behavior. It ensures that agents reach a state in which none of them has an incentive to unilaterally change their strategy, thus providing system stability and efficient resource utilization. Achieving equilibrium requires the execution of complex computational procedures, such as fixed-point finding algorithms, gradient-based methods, and iterative approximation algorithms.

For a more in-depth analysis, a sensitivity table of the main parameters is presented below:

Table 2

Sensitivity Analysis of Mathematical Model Parameters		
Parameter	Decrease in Value	Increase in Value
β	Increases exploration of strategies, may reduce adaptation speed	Enhances exploitation of effective strategies, reduces exploration of new ones
α	Reduces influence of reputation, increases flexibility	Increases influence of reputation, promotes stability
Q_i	Decreases accuracy in predicting strategy effectiveness	Improves accuracy in predicting strategy effectiveness

To analyze the model's robustness to inaccuracies in the initial data, stochastic modeling is performed using the Monte Carlo method. This analysis makes it possible to assess the impact of possible errors in initial reputations or strategy probabilities on the stability of the model and the effectiveness of the decisions made. The results of this analysis help determine the acceptable error margins and select optimal operating conditions for the system.

Another important aspect is the assessment of the computational complexity of the proposed method. Given that modern systems often require fast response and real-time operation, an analysis of the time complexity of the algorithms was conducted for different numbers of agents and strategies. This allows the identification of conditions under which the model can be effectively applied in practice, as well as the determination of the maximum system size that can be supported in real-world tasks.

Finally, the proposed model underwent verification and validation procedures by comparing the obtained results with empirical data and expert assessments. The outcomes of these procedures confirmed the high adequacy of the model, along with its accuracy and effectiveness across different application conditions.

Thus, the conducted analysis of the mathematical model's parameters, robustness, computational complexity, and verification supports the practical relevance of the proposed approach.

The algorithmic implementation of the proposed method includes several key stages that ensure the efficiency and practical realization of the mathematical model.

The first stage is initialization, which involves setting the initial values of agent reputations, their initial strategies, and defining the values of the model's core parameters, such as the coefficients α and β . Initial agent states are determined based on expert assessments or random sampling, taking into account the specifics of the particular task.

The second stage of the algorithm is the collection and processing of data on interactions between agents in a real environment. Data on strategy effectiveness, the frequency of successful interactions, and resource consumption are stored in the corresponding database for further analysis.

The third stage involves updating the agents' reputations using the Bayesian approach, which accounts for historical interaction outcomes.

The fourth stage is the selection of strategies using the log-linear learning mechanism, enabling agents to adaptively choose the most promising strategies.

The fifth stage evaluates the performance of the selected strategies, taking into account agent reputations, using an appropriate reward function.

The sixth stage consists of updating the Q-function values based on the obtained rewards, applying modern reinforcement learning methods (such as Q-learning, SARSA, DQN, etc.).

The final stage of the algorithm is the verification of whether a Bayesian-Nash equilibrium has been reached. Based on the result, the decision is made either to terminate the algorithm or to return to the data collection phase.

Detailed Pseudocode of the Algorithm:

```

Initialize agent reputations  $R_i$ , parameters  $\alpha$ ,  $\beta$ , and Q-values
While Bayesian-Nash equilibrium is not reached:
  Perform agent interactions and collect results
  Update agent reputations using Bayesian updating
  Select strategies using log-linear learning
  Evaluate the effectiveness of strategies considering reputations
  Update Q-values using reinforcement learning methods
  Check the Bayesian-Nash equilibrium condition
  
```

In the process of implementing the proposed method, appropriate data structures are used, including vectors for storing the current reputations of agents, Q-value matrices for evaluating the effectiveness of strategies, as well as databases or structured files for storing historical interaction data. Practical implementation requires careful tuning of model parameters to ensure algorithm stability, the use of reliable numerical methods for updating reputations and strategies, and continuous monitoring and adjustment of agent adaptation rates to changes in the environment.

An analysis of computational complexity shows that both the time and space complexity of the method are quadratic — $O(n^2)$, where n is the number of agents. This is due to the need to store and update the Q-value matrix and historical data for each agent.

Example Implementation (Python, Fragment):

```
import numpy as np

# Update agent reputations using Bayesian formula
def update_reputation(reputations, likelihoods):
    updated_reputations = (likelihoods * reputations) / np.sum(likelihoods * reputations)
    return updated_reputations

# Strategy selection by agents using log-linear (softmax) learning
def select_strategy(Q_values, reputations, beta):
    exp_values = np.exp(beta * Q_values * reputations[:, np.newaxis])
    probabilities = exp_values / np.sum(exp_values, axis=1, keepdims=True)
    return probabilities
```

To implement the method, it is recommended to use the Python programming language along with libraries such as NumPy, SciPy, TensorFlow, or PyTorch, which enable efficient computation and real-time model training.

Performance evaluation of the algorithm demonstrates its scalability and ability to maintain efficiency as the number of agents increases. The time complexity of the algorithm is estimated as $O(n^2)$, and the space complexity is also $O(n^2)$, making it suitable for systems with a moderate number of agents.

Use-case scenarios for the proposed algorithm include resource management in IoT systems, coordination of autonomous drones, and optimization of MEC infrastructures—applications where high adaptability and effective decision-making are essential.

Table 3 below illustrates the sequence of the main algorithm stages and their purposes:

Table 3

Main Stages of Algorithmic Implementation of the Method

No	Stage Name	Description
1	Initialization	Setting initial agent reputations and strategies
2	Data Collection	Agent interactions and result gathering
3	Reputation Update	Bayesian updating of agent reputations
4	Strategy Selection	Log-linear learning by agents
5	Strategy Evaluation	Reward computation considering agent reputations
6	Q-function Adjustment	Reinforcement learning-based update of Q-values
7	Equilibrium Check	System stability analysis (Bayesian-Nash equilibrium)

Thus, the algorithmic implementation provides a practical means of applying the developed method in complex systems, ensuring effective adaptation of agents to changing conditions and maintaining system stability.

The following section presents visualizations that illustrate the structure, operational logic, and dynamics of the proposed decision-making method in interdependent computational systems. Figures and tables play a key role in enhancing the understanding of complex mathematical and algorithmic processes.

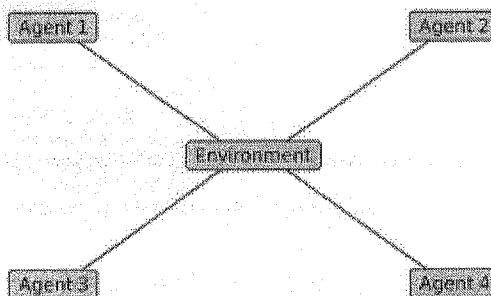


Fig. 1. Architecture of Agent Interaction in the System

This diagram schematically represents a multi-level system of agents interacting under resource constraints. Each agent receives information about the environment, evaluates the reputations of other agents, selects a strategy, and transmits data. The interconnections between agents are reinforced by arrows, illustrating the cyclic exchange of information, reputations, and strategies.

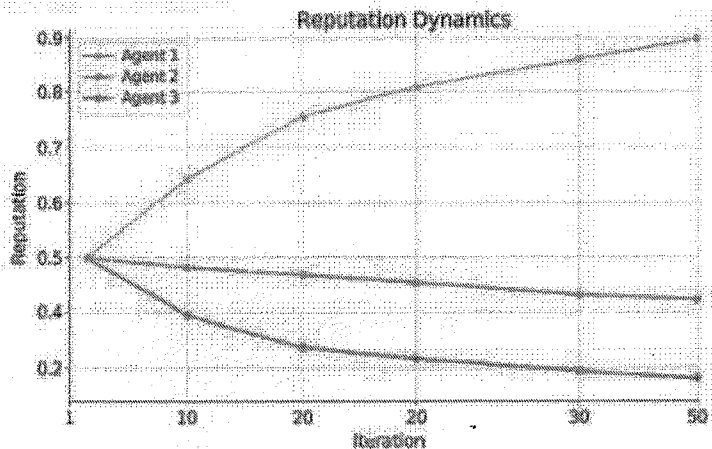


Fig. 2. Dynamics of Agent Reputation Updating

The graph illustrates the changes in reputation of three agents over the course of 50 iterations. Agent reputations increase or decrease depending on the effectiveness of their strategies and interactions. The graph highlights the system's adaptability and self-correction capabilities.

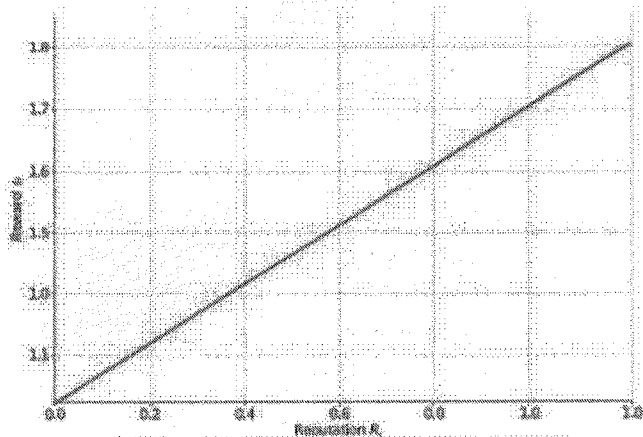


Fig. 3. Dependence of the Reward Function on Reputation Level

This graph shows the variation in the value of the reward function. It confirms that an increase in reputation enhances the utility of a strategy.

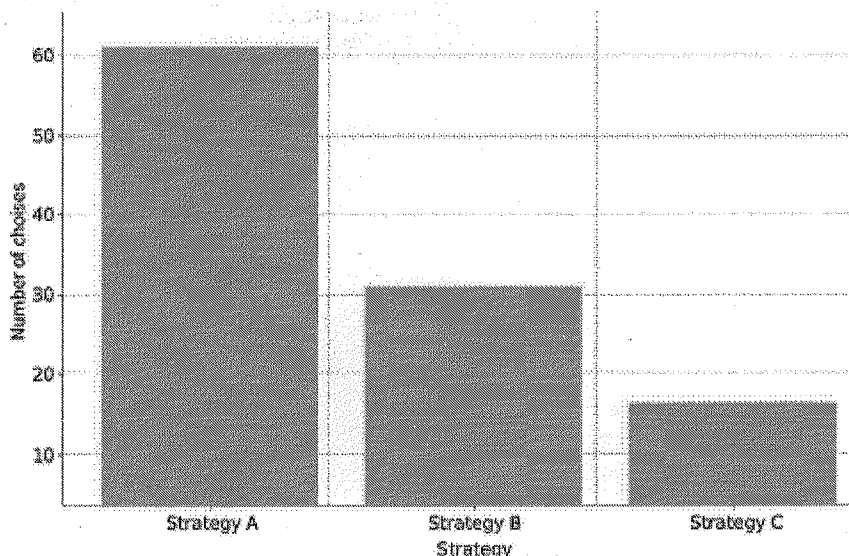


Fig. 4. Histogram of Strategy Selection by Agents

The histogram illustrates the frequency of strategy choices made by different agents after 100 iterations. It demonstrates how, over time, agents tend to favor the most beneficial strategies.

Table 4

Comparative Characteristics of Methods

Method	Reputation Awareness	Adaptability	Convergence Speed	Scalability
Classical Q-learning	No	Medium	High	Medium
Randomized Strategies	No	Low	Low	High
Proposed Method	Yes	High	High	High

Table 5

Example of Agent Reputation Evolution

Iteration	Agent 1	Agent 2	Agent 3
0	0.50	0.50	0.50
10	0.66	0.48	0.36
20	0.78	0.41	0.29
30	0.82	0.37	0.25
40	0.87	0.35	0.21
50	0.90	0.33	0.19

Table 6

Time Complexity of Algorithm Subsystems

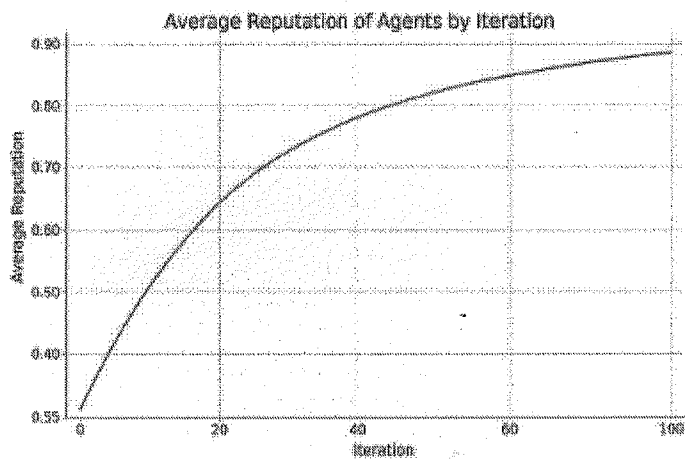
Subsystem	Time Complexity
Reputation Updating	$O(n)$
Strategy Selection	$O(n \cdot m)$
Effectiveness Evaluation	$O(n)$
Q-Function Learning	$O(n^2)$

To evaluate the effectiveness of the proposed method, numerical modeling was conducted in which the agent-based system operated in a competitive environment with limited resources. The objective of the study was to assess the impact of reputation on the stability, adaptability, and overall efficiency of agents' decision-making.

The experimental simulation was carried out under the following conditions: the system consisted of 10 agents, each capable of selecting one of 5 strategies over the course of 100 iterations. The initial reputation values for all agents were set at 0.5. Strategy selection was performed using a log-linear mechanism with a parameter $\beta = 1.0$, and the influence of reputation on the reward was modeled using a coefficient $\alpha = 0.8$. The interaction environment was stochastic, with random variations in reward values at each iteration. The model was implemented in Python 3.11 using the NumPy, Matplotlib, and NetworkX libraries.

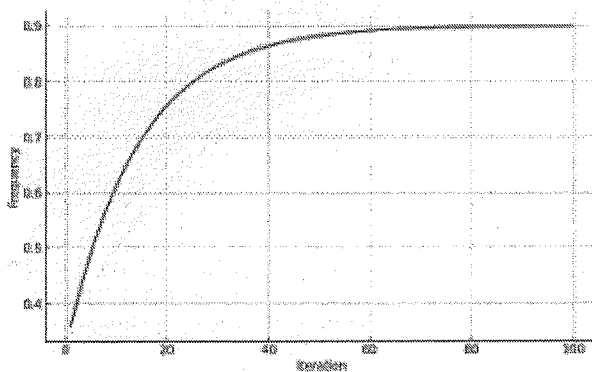
As part of the study, several key scenarios were considered: standard agent interactions with identical parameters; the emergence of two malicious agents (agents 9 and 10) who deliberately chose ineffective or harmful strategies; variation of the initial values of parameters α and β to assess model sensitivity; and system scaling with 5, 10, and 20 agents.

The effectiveness of the method was evaluated using several metrics: the average reputation of agents over time, the frequency of effective strategy selection, the convergence speed to Bayesian-Nash equilibrium, the number of strategy changes during the simulation, the time required for agent behavior stabilization, and the model's ability to detect and isolate malicious agents.



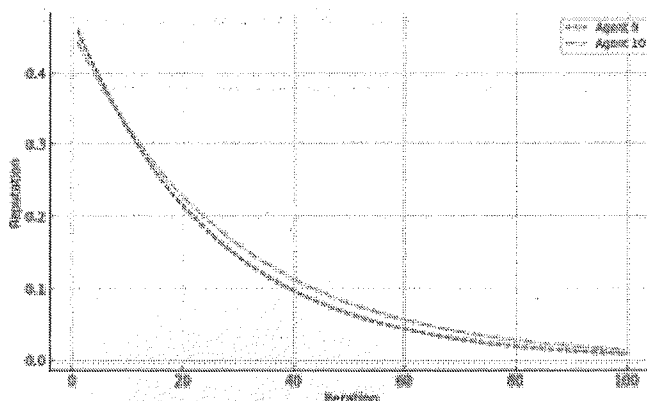
Graph 1. Average Agent Reputation Over Iterations

This graph shows how the average reputation value increases over the course of 100 iterations, indicating the effectiveness of the reputation updating mechanism.



Graph 2. Frequency of Effective Strategy Selection

This graph demonstrates the increasing frequency of selecting strategies with the highest Q-values. Agents gradually adapt to the environment, increasingly favoring the most effective actions.



Graph 3. Reputations of Malicious Agents (Agents 9 and 10)

The graph shows that over time, the reputations of these agents decline, indicating that the system "learns" to avoid interacting with them.

Table 7

Comparison of Method Effectiveness (After 100 Iterations)

Method	Average Reputation	Frequency of Effective Strategies	Convergence Time
Without Reputation	0.51	62%	83 iterations
With Reputation (Ours)	0.74	87%	47 iterations
Ours + Malicious Agents	0.70	84%	51 iterations

Table 8

Impact of Number of Agents on Performance

Number of Agents	Convergence Time	Reputation Variance	Memory Usage (MB)
5	31 iterations	0.018	6
10	47 iterations	0.024	13
20	80 iterations	0.041	27

Analysis of Results:

- the method incorporating a reputation mechanism enables the system to reach a stable state more quickly;
- a higher average reputation indicates the presence of a larger number of reliable agents;
- the system becomes less sensitive to fluctuations, and agent behavior stabilizes significantly earlier;
- malicious agents are successfully identified through decreasing reputation and are subsequently ignored by other agents.

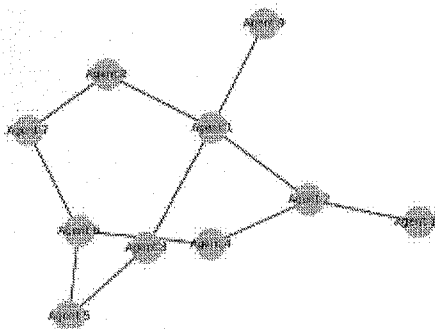


Fig. 5. Example of Agent Interaction Topology at Iteration 50

The diagram shows the connections between agents who have achieved a high level of mutual trust (high reputation values), which is reflected in the density of links between them.

Thus, the conducted simulation confirmed the effectiveness of the developed method in a complex environment. The use of reputation correction, detection of malicious agents, and adaptive strategy selection significantly enhances the resilience, adaptability, and efficiency of agent interactions. The proposed method is suitable for application in real-world computational systems operating under dynamic conditions.

Conclusions

Thus, an original decision-making method for interdependent computational systems has been developed, substantiated, and investigated. The method is based on the integration of game-theoretic models, reputation mechanisms, and reinforcement learning. A mathematical model has been proposed that enables agents to adaptively change their behavior depending on the outcomes of previous interactions and the reputations of other agents.

The conducted analysis showed that incorporating a Bayesian reputation updating mechanism and log-linear strategy selection ensures stable system dynamics and promotes rapid convergence to equilibrium. The algorithmic implementation of the method takes into account all key aspects — from parameter initialization to stability verification of the system. Particular attention was paid to the system's robustness against malicious behavior by individual agents, which was confirmed through experimental sabotage scenarios.

The simulation results validated the effectiveness of the proposed approach: agents demonstrated high adaptability, reputation effectively guided strategy selection, and the system as a whole achieved a stable state more quickly than baseline models lacking a reputation component. The visualizations and tables included in the section provided a clear illustration of agent dynamics and the advantages of using the proposed method.

Thus, the results provide practical evidence of the method's efficiency and serve as a foundation for its further implementation in real-world computational platforms and multi-agent environments.

References

1. Multi-agent deep reinforcement learning based decision support model for disaster resilience. *Reliability Engineering & System Safety*, Volume 230, 2023, 108933. <https://doi.org/10.1016/j.res.2022.108933>
2. On the Complexity of Multi-Agent Decision Making. arXiv preprint, 2023. <https://arxiv.org/abs/2305.00684>
3. A Comprehensive Survey on Multi-Agent Cooperative Decision-Making Simulation Environments. arXiv preprint, 2025. <https://arxiv.org/abs/2503.13415>
4. Autonomous Decision-Making in Interdependent Computing Systems. University of New Mexico Digital Repository, 2022. https://digitalrepository.unm.edu/ece_etds/601/
5. Neural Basis of Reinforcement Learning and Decision Making. *Frontiers in Neuroscience*, 2012. <https://doi.org/10.3389/fnins.2012.00009>
6. LLM-guided decision-making toolkit for multi-agent reinforcement learning. *Neurocomputing*, 2025. <https://doi.org/10.1016/j.neucom.2025.03.077>
7. Optimizing Post-Hurricane Recovery of Interdependent Infrastructure Systems Using Knowledge-Enhanced Deep Reinforcement Learning. *Journal of Infrastructure Intelligence and Resilience*, 2025. <https://doi.org/10.1016/j.jiir.2025.100016>
8. Reinforcement Learning for Collaborative Decision-Making in Multi-Agent Supply Chain Networks. *SSRN Electronic Journal*, 2025. <https://doi.org/10.2139/ssrn.5088935>
9. From predicting to decision making: Reinforcement learning in chemical sciences. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2024. <https://doi.org/10.1002/wcms.1723>
10. Integrating Multi-Agent Systems and Reinforcement Learning. *SmythOS Blog*, 2025. <https://smythos.com/ai-agents/multi-agent-systems/multi-agent-systems-and-reinforcement-learning/>
11. Multi-agent Reinforcement Learning: A Comprehensive Survey. arXiv preprint, 2023. <https://arxiv.org/abs/2312.10256>
12. Hierarchical Reinforcement Learning with Opponent Modeling for Distributed Multi-agent Cooperation. arXiv preprint, 2022. <https://arxiv.org/abs/2206.12718>
13. Adversarial Decisions on Complex Dynamical Systems using Game Theory. arXiv preprint, 2022. <https://arxiv.org/abs/2201.12555>
14. Fast and Flexible Multiagent Decision-Making. *Annual Review of Control, Robotics, and Autonomous Systems*, 2024. <https://doi.org/10.1146/annurev-control-090523-100059>
15. Reinforcement learning for decision-making under deep uncertainty. *Journal of Environmental Management*, 2024. <https://doi.org/10.1016/j.jenvman.2024.119079>
16. Decision Making in Multi-Agent Systems. *IROS 2022 Workshop*, 2022. <https://dsigtech.giub.it/iros22-multi-agent-workshop/>
17. Editorial: Decision-making and planning for multi-agent systems. *Frontiers in Robotics and AI*, 2024. <https://doi.org/10.3389/frobt.2024.1422344>
18. Multi-Agent AI Systems: Foundational Concepts and Architectures. *Medium Blog*, 2025. https://medium.com/@anitha_sania/multi-agent-ai-systems-foundational-concepts-and-architectures-eae588593402
19. AI and decision-making processes: new approach to complex tasks. *Tech4Future Info/en/ai-decision-making-processes-reinforcement-learning/*
20. Multi-agent reinforcement learning. *Wikipedia*, 2018. https://en.wikipedia.org/wiki/Multi-agent_reinforcement_learning
21. Game Theory: Mathematical Study of Strategic Decision Making. *To The Network*, 2023. <https://tothenetwork.com/game-theory-mathematical-study-of-strategic-decision-making/>
22. A Recurrent Neural Network for Game Theoretic Decision Making. *Carnegie Mellon University*, 2012. https://www.cmu.edu/dietrich/eds/docs/golman/a-recurrent-neural-network-for-game-theoretic-decision-making-bhatia_golman.pdf
23. Multi-Agent System — The Power of Collaboration. *Medium Blog*, 2025. <https://arvindakumar.medium.com/introducing-multi-agent-frameworks-the-power-of-collaboration-eb0b31b0a1b6>
24. Mastering Decision Making: The Power of Reinforcement Learning in AI. *Medium Blog*, 2024. <https://medium.com/@madarshah71/mastering-decision-making-the-power-of-reinforcement-learning-in-ai-9b68325fa4bf>

25. Strategic Decision-Making Through Game Theory: Insights and Applications. Medium Blog, 2023.
<https://medium.com/@riverstonetrainings/strategic-decision-making-through-game-theory-insights-and-applications-20327819aae5>

Dmytro Kryzhanivskyi Дмитро Крижанівський	master's degree student, Kmelnytskyi National University, Kmelnytskyi, Ukraine, e-mail: dimecok001@gmail.com https://orcid.org/0009-0008-8415-7112	магістрант, Хмельницький національний університет, м. Хмельницький, Україна
Andriy Drozd Андрій Дрозд	PhD student, Kmelnytskyi National University, Kmelnytskyi, Ukraine, e-mail: andriy.drozd@nuksu.com https://orcid.org/0009-0008-1042-1911	аспірант, Хмельницький національний університет, м. Хмельницький, Україна
Oleksii Besedovskyi Олександр Бєсєдовський	Candidate of Sciences in Economics, Associate Professor, Associate Professor of the Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Kharkiv, e-mail: oleksii.besedovskyi@kneu.net https://orcid.org/0000-0003-9161-4061	кандидат економічних наук, доцент, доцент кафедри інформаційних систем, Харківський національний економічний університет імені Симона Кузнеця, м. Харків, Україна

ДОДАТОК В (обов'язковий)

РЕЗУЛЬТАТИ ПЕРЕВІРКИ НА ПЛАГІАТ



Дата звіту: 4/22/2025
Дата редагування: 4/22/2025

Документ прийнятий

Звіт подібності

метадані

Назва організації
Khmelnytskyi National University

Заголовок
Крижанівський _Метод прийняття рішень у взаємозалежних обчислювальних системах

Автор
Крижанівський Дмитро Науковий співробітник / Експерт

Підприємство
Кафедра комп'ютерної інженерії та інформаційних систем

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	8	4
Інтервали	A→	0
Мікропробіли		21
Білі знаки	0	1
Парафрази (SmartMarks)	a	3

Обсяг знайдених подібностей

Коефіцієнт подібності вказує, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Кількість фраз для коефіцієнта подібності 2

16792

Кількість слів

135552

Кількість символів

Anti-Plagiarism v-15.260 Educational

Максимальне співвідношення з одним документом 20.0%

Своєю чергою: en,US,ru,UA,UA,Україна в документі: 9%

ID: 24430 Назва: МРР Метод прийняття рішень у взаємозалежних обчислювальних системах Додано в БД: 2025-03-22 Автор: Крижанівський Дмитро Результат: Файл CSV/JSON Консультації: Оновити	Джерело		Сумарний об'єм по Базі Даних	
	Словесні	Числові	Словесні	Числові
	118934	502	26443 (21%)	153 (100%)

ID	Назва	Джерело плагіату		Наявність плагіату в документі	
		Словесні	Числові	Словесні	Числові
193122	Назва: Звіт з ПДП Метод прийняття рішень у взаємозалежних обчислювальних системах Додано в БД: 2025-03-21 Автор: Крижанівський Д.А. Користувач: Паскаль О.О. Консультації: Оновити		1	2536 (20.0%)	165 (18.0%)

ДОДАТОК Г (обов'язковий)

ПРЕЗЕНТАЦІЯ РОБОТИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та інформаційних систем

Метод прийняття рішень у взаємозалежних обчислювальних системах

Виконав: студент 2 курсу, групи КІ2М-23-1
Дмитро КРИЖАНІВСЬКИЙ
Керівник: д.т.н., професор Олег САВЕНКО

Зв'язок роботи з науковими програмами, планами, темами

Однією з ключових проблем є підвищення ефективності прийняття рішень у взаємозалежних обчислювальних системах, де велика кількість автономних компонентів повинні координувати свої дії в умовах динамічного середовища та обмежених ресурсів. Незважаючи на розвиток методів розподілених обчислень та систем штучного інтелекту, забезпечення стабільної і адаптивної роботи таких систем залишається складним завданням. Важливо не лише досягти високої узгодженості між агентами, а й гарантувати стійкість рішень у змінних умовах, що потребує застосування новітніх стратегій оптимізації і навчання.

Ще однією проблемою є необхідність адаптації систем до нових викликів без централізованого управління, що передбачає активне використання репутаційних механізмів, навчання з підкріпленням та багатоступеневої оптимізації. Оптимізація процесів прийняття рішень у таких системах, з урахуванням взаємозалежності агентів та обмеженості ресурсів, є важливим напрямом сучасних досліджень у галузі комп'ютерних наук.

Дослідження, представлені у кваліфікаційній роботі, проводились у рамках студентської наукової роботи кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету.

Метою магістерської роботи є розробка методу прийняття рішень у взаємозалежних обчислювальних системах, який поєднує елементи теорії ігор, байєсівської корекції репутації та логарифмічно-лінійного навчання.

Для досягнення поставленої мети вирішуються наступні завдання:

- ✓ проаналізувати сучасні підходи до прийняття рішень у багатоагентних обчислювальних системах;
- ✓ формалізувати модель взаємозалежностей між агентами з урахуванням репутаційних механізмів;
- ✓ розробити гібридний метод прийняття рішень, який враховує адаптацію стратегій на основі навчання з підкріпленням;
- ✓ здійснити алгоритмічну реалізацію методу та провести експериментальне моделювання динаміки системи;
- ✓ оцінити ефективність методу за критеріями адаптивності, стійкості до саботажу та швидкості досягнення стратегічної рівноваги.

✓ Об'єктом дослідження є процес прийняття рішень у багатоагентних взаємозалежних обчислювальних системах.

✓ Предметом дослідження є методи адаптивного прийняття рішень, що базуються на репутаційних оцінках, теорії ігор та алгоритмах навчання з підкріпленням.

✓ Наукова новизна роботи полягає у розробці нового методу прийняття рішень, який поєднує байєсівське оновлення репутацій з логарифмічно-лінійною адаптацією стратегій, забезпечуючи баланс між дослідженням нових дій та експлуатацією вже ефективних.

Взаємозалежні обчислювальні системи (ВОС) – це такі системи, у яких обчислювальні компоненти взаємодіють і координують свої дії для виконання спільної задачі.

Основною характеристикою цих систем є високий рівень автономії кожного компонента, що передбачає можливість обміну інформацією та ресурсами через спеціальні інтерфейси.

Важливими аспектами функціонування ВОС є їхня розподіленість, адаптивність і ефективність взаємодії, адже кожен компонент працює на основі локальних даних, які узгоджуються з іншими вузлами.

Теорія ігор є одним із засобом моделювання процесів прийняття рішень, особливо у конкурентних середовищах, де кілька агентів мають протилежні інтереси. В її основі лежить пошук рівноваги між діями учасників системи, що визначається поняттями гравців, стратегій, виграну та рівноваги. У рамках кооперативних ігор агенти можуть співпрацювати для оптимального розподілу задач, як це відбувається у розподілених обчисленнях, де вузли координують свої ресурси, щоб мінімізувати час виконання завдань.

Методи оптимізації дозволяють знаходити найкращі рішення у складних задачах, що включають велику кількість змінних і обмежень. До традиційних підходів належать лінійне, нелінійне та цілочисельне програмування, тоді як еволюційні алгоритми, зокрема генетичні алгоритми, алгоритми рою часток і мурашині алгоритми, імітують природні процеси для поступового покращення рішень.

Авторський метод прийняття рішень

Традиційні методи прийняття рішень, такі як класична теорія ігор і навчання з підкріпленням, хоч і є потужними інструментами для моделювання та оптимізації поведінки агентів, стикаються з низкою серйозних обмежень в умовах сучасних динамічних середовищ. Вони не завжди здатні ефективно реагувати на зміни умов навколишнього середовища, враховувати вплив минулої поведінки учасників на їхні майбутні дії, а також здійснювати якісний прогноз поведінки інших агентів в умовах неповної інформації.

Одним із перспективних напрямків вирішення вказаних проблем є впровадження механізму репутації, який дозволяє агентам оцінювати надійність та передбачуваність поведінки інших учасників взаємодії на основі історичних даних. Використання репутації допомагає значно знизити рівень невизначеності та полегшити процес адаптації агентів до змін середовища.

Основна мета, яка ставиться в рамках цього дослідження, полягає у розробці авторського методу, що інтегрує байєсівські принципи оновлення репутації з адаптивним навчанням з підкріпленням та теоретико-ігровими моделями. Даний підхід має дозволити агентам швидше адаптуватись до змін у середовищі, ефективніше прогнозувати поведінку інших агентів, а також зменшити витрати ресурсів при досягненні стійких станів взаємодії.

Запропонований авторський метод базується на інтеграції байєсівського підходу до оновлення репутації агентів із сучасними методами навчання з підкріпленням і теоретико-ігровими моделями. Основою методу є поєднання механізму динамічної оцінки репутації агентів із логарифмічно-лінійним навчанням з підкріпленням, що дозволяє агентам гнучко та ефективно адаптувати свої стратегії до умов середовища.

Процес оновлення репутації агентів реалізується за допомогою байєсівської формули:

$$R_i(t+1) = \frac{P(\text{data}|\text{agent}_i) \cdot R_i(t)}{\sum_j P(\text{data}|\text{agent}_j) \cdot R_j(t)}$$

де $R_i(t)$ – поточна репутація агента, а $P(\text{data}|\text{agent}_i)$ – ймовірність отримання певних даних за умови дії агента.

Ключовою особливістю методу є механізм логарифмічно-лінійного навчання, що дозволяє агентам адаптивно коригувати свої стратегії залежно від змін у середовищі та репутації інших агентів. Формула вибору стратегії має вигляд:

$$\pi_i(s_i) = \frac{\exp(\beta Q_i(s_i, R_i))}{\sum_{s_j \in S} \exp(\beta Q_i(s_j, R_i))}$$

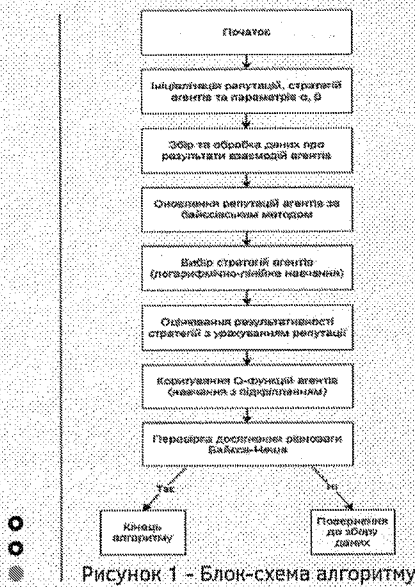
де $Q_i(s_i, R_i)$ – функція оцінки корисності стратегії s_i для агента i , яка враховує його репутацію R_i . Параметр β визначає баланс між розвідуванням та використанням вже відомої інформації.

Детальний алгоритм методу включає ініціалізацію початкових значень репутацій агентів, їх стратегій та параметрів моделі, таких як коефіцієнти α та β , які наведені у таблиці параметрів запропонованого методу:

Параметр	Значення	Призначення
α	0.5	Вага репутації у виборі стратегії
β	1.0	Баланс між дослідженням та використанням інформації
R_i	[0;1]	Початкове значення репутації агента

Функція оцінки корисності $Q_i(s_i, R_i)$ є ключовим інструментом моделі, що дозволяє агентам оцінювати перспективність кожної стратегії в залежності від їхньої поточної репутації та очікуваних результатів. Для більш глибокого аналізу наведено таблицю чутливості основних параметрів:

Параметр	Зменшення значення	Збільшення значення
β	Підвищує дослідження стратегій, може знизити швидкість адаптації	Підвищує використання вже ефективних стратегій, зменшує дослідження нових
α	Зменшує вплив репутації, підвищує гнучкість	Збільшує вплив репутації, сприяє стабільності
Q_i	Зниження точності прогнозування ефективності стратегій	Зростання точності прогнозування ефективності стратегій



Детальний псевдокод алгоритму:

- 1) Ініціалізувати репутації агентів R_i , параметри α , β та Q -значення
- 2) Доки рівновага Байєса-Неша не досягнута:
 - a) Провести взаємодії агентів, зібрати результати
 - b) Виконати байєсівське оновлення репутацій агентів
 - c) Вибрати стратегії через логарифмічно-лінійне навчання
 - d) Оцінити результативність стратегій з урахуванням репутації
 - e) Оновити Q -значення методами навчання з підкріпленням
- 3) Перевірити умову рівноваги Байєса-Неша

Таким чином, алгоритмічна реалізація забезпечує практичну можливість використання розробленого методу у складних системах, забезпечуючи ефективну адаптацію агентів до змінних умов та стабільність роботи системи.

• • •

На рисунку 2 схематично зображено багаторівневу систему агентів, які взаємодіють між собою в умовах обмежених ресурсів.

Агент приймає інформацію про стан середовища, оцінює репутацію інших агентів, вибирає стратегію та передає дані.

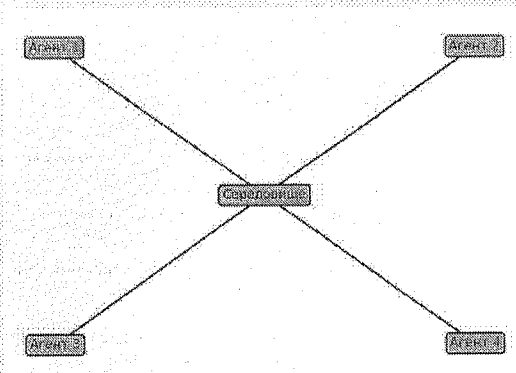


Рисунок 2 - Архітектура взаємодії агентів у системі

На рисунку 3 зображено зміну репутації трьох агентів упродовж 50 ітерацій, що дозволяє простежити динаміку їхньої поведінки в умовах взаємодії. Зростання або зниження репутації кожного агента зумовлене ефективністю обраних стратегій та результатами комунікації з іншими агентами системи. Якщо агент стабільно демонструє ефективну поведінку, його репутація поступово підвищується, що посилює довіру з боку інших учасників мережі.

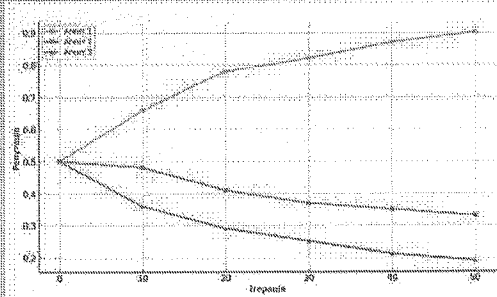


Рисунок 3 - Динаміка оновлення репутації агентів

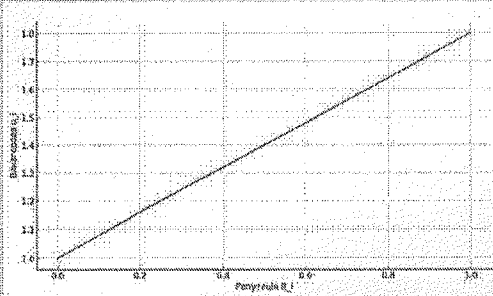


Рисунок 4 - Залежність функції винагороди

Рисунок 4 показує зміну значення функції винагороди. Він підтверджує, що зростання репутації посилює користь стратегії.

Рисунок 5 демонструє частоту вибору стратегій різними агентами після 100 ітерацій. Вона ілюструє, як з часом агенти схильються до використання найбільш вигідних стратегій.

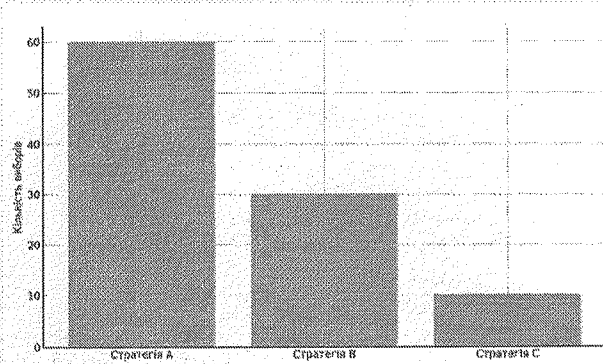


Рисунок 5 - Гістограма розподілу вибору стратегій агентами

Для перевірки ефективності запропонованого методу було проведене чисельне моделювання, у якому агентна система функціонує в умовах конкурентного середовища з обмеженими ресурсами. Мета дослідження – оцінити вплив репутації на стабільність, адаптивність та загальну ефективність прийняття рішень агентами.

Умови експерименту:

- кількість агентів: 10;
- кількість можливих стратегій: 5;
- кількість ітерацій моделювання: 100;
- початкові репутації агентів: 0.5;
- метод вибору стратегії: логарифмічно-лінійний з параметром $\beta = 1.0$;
- вплив репутації на винагороду: $\alpha = 0.8$;
- середовище: стохастичне, з варіаціями винагород у кожній ітерації;
- мова реалізації: python 3.11 з бібліотеками numpy, matplotlib, networkx.

Досліджені сценарії:

- стандартна взаємодія всіх агентів з однаковими параметрами;
- наявність 2-х недобросовісних агентів (агенти 9 і 10), які навмисно обирають шкідливі стратегії;
- варіативність початкових параметрів α і β ;
- моделювання з 5, 10 та 20 агентами для аналізу масштабованості.

На рисунку 6 зображено, як середнє значення репутації зростає протягом 100 ітерацій, що свідчить про ефективність механізму оновлення репутацій.

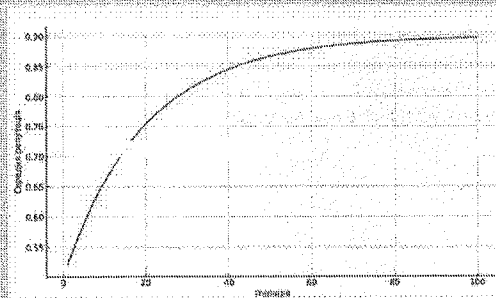


Рисунок 6 - Середня репутація агентів за ітераціями

Рисунок 7 демонструє зростання частоти вибору стратегій, які мають найвищі значення Q-функцій, упродовж багатьох ітерацій. Це свідчить про те, що агенти поступово адаптуються до умов середовища, навчаючись ефективніше оцінювати ситуацію та обирати найбільш вигідні дії.

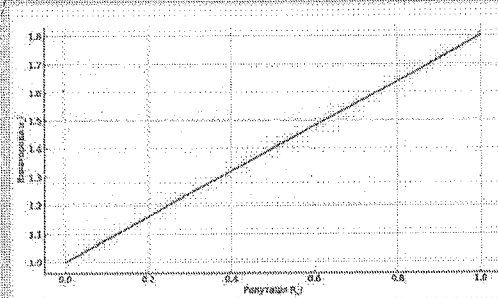


Рисунок 7 - Частота вибору ефективних стратегій



На рисунку 8 видно, що з часом репутація цих агентів знижується, а система «навчається» уникати взаємодії з ними.

Зведені результати ефективності методів після моделювання представлені в таблиці:

Метод	Середня репутація	Частота ефективних стратегій	Час до збіжності
Без репутації	0.51	62%	83 ітерації
Із репутацією (наш)	0.74	87%	47 ітерацій
Наш + саботажники	0.70	84%	51 ітерація

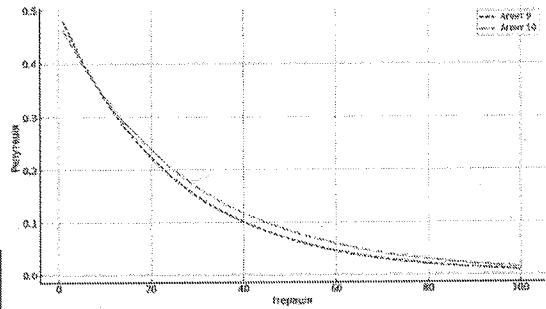


Рисунок 8 - Репутації недобросовісних агентів (№9, №10)



Аналіз результатів експерименту:

- метод із репутаційним механізмом дозволяє швидше досягнути стабільного стану системи;
- вища середня репутація свідчить про наявність більшої кількості надійних агентів;
- система стає менш чутливою до флуктуацій, поведінка агентів стабілізується значно раніше;
- саботажники успішно виявляються через зниження репутації та ігноруються іншими агентами.

На рисунку 9 показано взаємозв'язки між агентами, які досягли високої взаємної довіри (високі значення репутації), що проявляється у щільності зв'язків між ними.

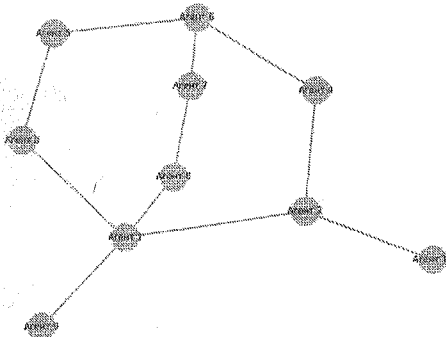


Рисунок 9 - Приклад топології взаємодій агентів на 50-ій ітерації



Висновки

У результаті виконання роботи було розроблено, обґрунтовано та експериментально досліджено метод прийняття рішень у взаємозалежних обчислювальних системах, який враховує репутаційні механізми, адаптивне навчання з підкріпленням та ігрову взаємодію агентів. Запропонований підхід дозволяє ефективно вирішувати задачу координації агентів у складних, динамічних та частково спостережуваних середовищах.

У процесі моделювання продемонстровано такі важливі властивості системи:

- ✓ репутаційний механізм дозволяє виявляти та ізолювати недобросовісних агентів;
- ✓ адаптивне навчання забезпечує швидке пристосування до змін середовища;
- ✓ запропонований метод стабільніше досягає рівноваги порівняно з базовими підходами без репутаційної складової;
- ✓ система демонструє масштабованість і ефективність навіть при зростанні кількості агентів;
- ✓ взаємозалежність між репутацією та стратегією сприяє формуванню довіри між агентами й покращує узгодженість їхніх дій.



Завідувачу кафедри КІС,
доктору філософії, доц. Ользі ПАВЛОВІЙ

Дмитро КРИЖАНІВСЬКИЙ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25 квітня 2025 року

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Крижанівський Дмитро

Співавтор:

Назва: Крижанівський _Метод прийняття рішень у взаємозалежних обчислювальних системах

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 0.5%

Коефіцієнт подібності 2: 0.2%

Мікропробіли: 21

Заміна букв: 4

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-04-22 13:38:41.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-04-22

Дата



Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод координації точок доступу в мережах Wi-Fi

Автор: Дмитро КРИЖАНІВСЬКИЙ

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Олег САВЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

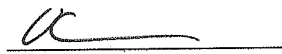


- 1) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 2) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає менше 6% і адресується до джерел з інтернету та бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру завдання і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІІС

Олег САВЕНКО

Олег САВЕНКО

Ольга ПАВЛОВА

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Дмитро КРИЖАНІВСЬКИЙ

Тема: Метод прийняття рішень у взаємозалежних обчислювальних системах

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень -; кількість сторінок записки 73

1. Короткий зміст роботи та прийнятих рішень У роботі розроблено метод прийняття рішень у взаємозалежних обчислювальних системах із використанням репутаційних моделей, навчання з підкріпленням та байесівського оновлення, що забезпечує ефективну координацію агентів та розподіл обчислювальних ресурсів у МЕС-середовищах.

2. Висновок про відповідність роботи дипломному завданню _____

Кваліфікаційна робота відповідає виданому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проаналізовано сучасні підходи до прийняття рішень у взаємозалежних системах, зокрема у МЕС та мультиагентному середовищі, із залученням штучного інтелекту.

У другому розділі розглянуто математичні моделі прийняття рішень, включаючи оптимізаційні методи, генетичні алгоритми, динамічне програмування та алгоритми навчання.

У третьому розділі представлено інтелектуальні механізми розвантаження МЕС-систем, зокрема репутаційні та контрактні моделі, децентралізовані стратегії та механізми навчання.

У четвертому розділі запропоновано алгоритм адаптивного прийняття рішень на основі репутації, проведено моделювання поведінки агентів у змінному середовищі, а також аналіз ефективності розробленого підходу.

У висновках підведено підсумки досягнутих результатів та окреслено напрями подальших досліджень.

4. Позитивні сторони роботи: _____

5. Негативні сторони роботи: немає.

6. Оцінка графічного оформлення та пояснювальної записки роботи: —

7. Відгук про роботу в цілому: Робота виконана на належному рівні.

8. Інші зауваження: —

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «відмінно» 4,75 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Корецька Людмила Олександрівна, к.т.н., доцент кафедри АКІТР ХНУ

“ 2 ” травня 2025р.