

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра прикладної математики

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Оптимізація топологічної структури

логістичної мережі

на етапі реінжинірингу

(тема)

Виконав:

здобувач 2 року навчання, групи САУМ-24-1

Дмитро ЖАБСЬКИЙ

(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 124 Системний аналіз

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва освітньої програми)

Керівник проф. Володимир БЕЗКОРОВАЙНИЙ

(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Завідувач кафедри ПМ

(підпис)

Максим СИДОРОВ

(Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 124 Системний аналіз

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

(підпис)

“ 10 ” листопада 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Жабському Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Оптимізація топологічної структури логістичної мережі на етапі реінжинірингу

затверджена наказом по університету від 10 листопада 2025 р. № 1027 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 18 грудня 2025 р.

3. Вихідні дані до роботи модель топологічної структури логістичної мережі

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Системний аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

1. Актуальність теми роботи _____

2. Постановка задачі _____

3. Системний аналіз предметної області _____

4. Метод чисельного аналізу _____

5. Результати обчислювального експерименту _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	10 – 16 листопада 2025 р.	виконано
2	Вибір та обґрунтування методу	17 – 23 листопада 2025 р.	виконано
3	Розробка алгоритму і програми	24 – 30 листопада 2025 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	01 – 07 грудня 2025 р.	виконано
5	Робота над текстом пояснювальної записки	08 – 17 грудня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	18 грудня 2025 р.	виконано

Дата видачі завдання 10 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Володимир Безкоровайний
(підпис) (посада, Власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка: 110 с., 7 табл., 15 рис., 1 додаток, 30 джерел.

ГЕНЕТИЧНИЙ АЛГОРИТМ, ЕВОЛЮЦІЙНА ОПТИМІЗАЦІЯ, ЛОГІСТИЧНА МЕРЕЖА, МЕТОД ПОКООРДИНАТНОГО СПУСКУ, РЕІНЖІНІРИНГ, ТОПОЛОГІЯ МЕРЕЖІ, ФУНКЦІЯ ПРИСТОСОВАНОСТІ, ЦІЛЬОВА ФУНКЦІЯ, ЧАСОВА СКЛАДНІСТЬ.

Об'єкт дослідження – процес реінжинірингу топологічних структур централізованих логістичних мереж.

Мета роботи – підвищення ефективності функціонування логістичних мереж шляхом розробки та застосування методів оптимізації їх топологічної структури на етапі реінжинірингу з урахуванням структурних, топологічних і функціональних обмежень.

Методи дослідження – системний аналіз і математичне моделювання задачі, метод покоординатного спуску, еволюційно-модифікований генетичний алгоритм, обчислювальний експеримент і статистична обробка результатів.

У роботі виконано порівняльне дослідження двох підходів до структурно-топологічної оптимізації логістичної мережі в умовах реінжинірингу: детермінованого методу покоординатного спуску (МПО) та еволюційно-модифікованого генетичного алгоритму (ЕМ-ГА). Для оцінки точності проведено серію експериментів на тестових мережах різної розмірності. Для оцінки часових характеристик вимірювання виконувалися засобами Python із фіксацією тривалості до мілісекунд та аналізом стабільності.

Практичне значення роботи полягає в обґрунтованому виборі методу залежно від вимог задачі: МПО доцільний для швидкого отримання якісного рішення, тоді як ЕМ-ГА – для задач, де критичним є додаткове зниження витрат і допустима стохастичність пошуку.

ABSTRACT

Introductory note: 110 pages, 7 tables, 15 figures, 1 appendix, 30 sources.

COORDINATE DESCENT METHOD, EVOLUTIONARY OPTIMIZATION, FITNESS FUNCTION, GENETIC ALGORITHM, LOGISTICS NETWORK, NETWORK TOPOLOGY, OBJECTIVE FUNCTION, REENGINEERING, TIME COMPLEXITY.

Object of research – the reengineering process of centralized logistics network topologies.

Purpose of work – to improve the efficiency of logistics networks by developing and applying topology optimization methods at the reengineering stage under structural, topological and functional constraints.

Methods of research – system analysis and mathematical modelling, coordinate descent optimization, an evolutionary modified genetic algorithm, computational experiments and statistical evaluation of results.

The thesis presents a comparative study of two approaches for structural-topological optimization of logistics networks during reengineering: a deterministic coordinate descent method (CDM) and an evolutionary modified genetic algorithm (EMGA). The accuracy analysis is based on a set of test networks with different sizes; the stochastic method is evaluated using multiple independent runs with mean values and dispersion measures. Runtime performance was measured in Python with millisecond precision and stability analysis.

The practical outcome is a justified recommendation on method selection: the deterministic approach is preferable when time constraints dominate, whereas the evolutionary method is preferable when additional cost reduction is crucial and stochastic search is acceptable.

ЗМІСТ

	С.
Перелік скорочень	7
Вступ	8
1 Системний аналіз предметної області та постановка задач дослідження	10
1.1 Системний аналіз задач оптимізації топологічних структур	10
1.2 Аналіз сценаріїв вирішення задачі оптимізації топологічних структур на етапі реінжинірингу	14
1.3 Змістовна та формальна постановка задачі	19
1.4 Постановка задач дослідження	22
2 Вибір та обґрунтування методів розв'язання	27
2.1 Огляд методів структурно-топологічної оптимізації логістичних мереж	27
2.2 Метод покоординатної оптимізації	31
2.3 Еволюційний метод на основі генетичного алгоритму	35
Висновки за розділом 2	39
3 Програмна реалізація	41
3.1 Вибір інструментальних засобів та мови програмування	41
3.2 Архітектура програмного комплексу	43
3.3 Алгоритми основних модулів програми	48
3.4 Опис інтерфейсу користувача та функціональних можливостей	55
Висновки за розділом 3	66
4 Результати обчислювального експерименту та їх аналіз	68
4.1 Методика проведення обчислювального експерименту	68
4.2 Дослідження точності методів оптимізації	70
4.3 Дослідження часової складності методів оптимізації	76
Висновки за розділом 4	78
Висновки	80
Перелік джерел посилання	83
Додаток А Лістинг програми	86

ПЕРЕЛІК СКОРОЧЕНЬ

CSV – Comma-Separated Values, табличний формат зберігання результатів експериментів;

ЕМ-ГА – еволюційно-модифікований генетичний алгоритм;

МПО – метод покоординатного спуску (метод покоординатної оптимізації).

ВСТУП

Актуальність теми. Зміна умов функціонування логістичних систем, що включає появу нових споживачів, зміну обсягів поставок, коливання транспортних тарифів та модифікацію інфраструктури, призводить до того, що на певному етапі існуючі мережі стають неефективними або не відповідають новим вимогам. Традиційні підходи до модернізації, що передбачають локальні покращення окремих елементів системи, не гарантують отримання ефективних глобальних рішень. Задачі багатокритеріальної системної оптимізації логістичних мереж на етапі реінжинірингу вимагають розробки нових математичних моделей та методів розв'язання, які враховують можливість використання наявної інфраструктури, множину топологічних та функціональних обмежень, а також інтервали можливих змін попиту і трафіку. Більшість існуючих математичних моделей призначені для умов проєктування нових мереж з повністю визначеними вхідними даними, що не відповідає реальним умовам реінжинірингу. Це визначає актуальність розробки спеціалізованих моделей та методів оптимізації топологічних структур централізованих логістичних мереж на етапі їх реінжинірингу.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є підвищення ефективності функціонування логістичних мереж шляхом розробки та застосування методів оптимізації їх топологічних структур на етапі реінжинірингу з урахуванням множини структурних, топологічних та функціональних обмежень. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести системний аналіз предметної області та сучасного стану задачі оптимізації топологічних структур централізованих логістичних мереж;
- виконати аналіз сценаріїв вирішення задачі оптимізації топологічних структур на етапі реінжинірингу та сформулювати змістовну і формальну постановку задачі;
- здійснити вибір та обґрунтування методів розв'язання задачі оптимізації

з урахуванням специфіки реінжинірингу логістичних мереж;

- розробити програмну реалізацію обраних методів оптимізації топологічних структур логістичних мереж;
- провести обчислювальний експеримент для дослідження точності та часової складності розроблених методів;
- сформулювати практичні рекомендації щодо застосування розроблених методів для розв'язання задач різної розмірності.

Об'єктом дослідження є процес реінжинірингу топологічних структур централізованих логістичних мереж.

Предметом дослідження є математичні моделі та методи оптимізації топологічних структур логістичних мереж на етапі реінжинірингу за критеріями мінімізації витрат та максимізації оперативності доставки

Методи дослідження. У кваліфікаційній роботі використовуються методи системного аналізу для дослідження предметної області та формалізації задачі оптимізації, методи оптимізації та дослідження операцій для розробки алгоритмів розв'язання задачі, методи покоординатної оптимізації та еволюційні методи на основі генетичних алгоритмів для пошуку ефективних варіантів топологічних структур, а також методи обчислювального експерименту для оцінки точності та часової складності розроблених методів.

Публікації. Результати, отримані у роботі, було представлено на 29-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (м. Харків, 16-19 квітня 2025 р.) [1], на міжнародній науково-практичній конференції «Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі» (м. Київ, 2025 р.), у межах якої висвітлено підхід до оптимізації транспортних маршрутів у логістичних системах з урахуванням зворотних потоків [2], а також у науковому фаховому журналі «Вісник Харківського національного автомобільно-дорожнього університету», випуск 111 за 2025 рік, де опубліковано результати розробки математичної моделі задачі еколого-економічної оптимізації логістичних мереж [3].

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз задач оптимізації топологічних структур

Задачі оптимізації топологічних структур мереж належать до фундаментальних проблем системного аналізу та дослідження операцій. Вони виникають у різних предметних областях, де функціонують територіально розподілені системи: телекомунікаційних мережах, інформаційних системах, енергетичних мережах, транспортних системах та логістичних мережах. Незважаючи на різну фізичну природу об'єктів, математичні моделі та методи оптимізації їх топологічних структур мають спільну теоретичну основу.

У загальному випадку під топологічною структурою мережі розуміють сукупність вузлів та зв'язків між ними, що визначає конфігурацію системи в просторі. Оптимізація топологічної структури передбачає визначення оптимальної кількості вузлів, їх розташування та схеми з'єднань, що забезпечують досягнення заданих показників ефективності при мінімальних витратах ресурсів.

Задачі оптимізації топології у телекомунікаційних мережах займають важливе місце в теорії проектування систем зв'язку. Телекомунікаційна мережа складається з вузлів комутації, каналів зв'язку та кінцевих пристроїв користувачів. Задача оптимізації полягає у визначенні кількості та розміщення вузлів комутації, а також конфігурації каналів зв'язку між ними. Критеріями оптимальності виступають мінімізація вартості побудови та експлуатації мережі, забезпечення заданої пропускну здатності та надійності функціонування. Особливістю телекомунікаційних мереж є необхідність забезпечення резервування зв'язків для підвищення відмовостійкості, що призводить до розгляду задач синтезу мереж з надлишковою зв'язністю.

Оптимізація топології інформаційних систем пов'язана з проектуванням розподілених баз даних, хмарних обчислювальних систем та мереж центрів обробки даних. У таких системах вузлами є сервери або кластери серверів, а зв'яз-

ки визначають маршрути передачі даних між ними. Задача оптимізації включає визначення кількості та географічного розміщення центрів обробки даних, розподіл інформаційних ресурсів між вузлами та оптимізацію маршрутів передачі запитів. Критеріями ефективності є мінімізація часу відгуку системи, забезпечення балансування навантаження та мінімізація витрат на інфраструктуру. Специфіка полягає у необхідності врахування динаміки навантаження та забезпечення масштабованості системи.

Задачі оптимізації енергетичних мереж виникають при проектуванні систем електропостачання, газопостачання та тепlopостачання. Вузлами таких мереж є генеруючі станції, трансформаторні підстанції, розподільні пункти та споживачі. Задача полягає у визначенні оптимальної конфігурації мережі, що забезпечує надійне постачання енергоресурсів при мінімальних втратах та витратах на будівництво і експлуатацію. Особливістю є необхідність врахування фізичних законів передачі енергії, обмежень на потужність елементів та вимог до безперебійності постачання.

Логістичні мережі являють собою складні територіально розподілені системи, що забезпечують переміщення товарно-матеріальних потоків від виробників до кінцевих споживачів. Ефективність функціонування таких систем значною мірою визначається раціональністю їх структурної організації та топологічного розміщення елементів [4].

Спільною рисою задач оптимізації топологічних структур у різних предметних областях є їх належність до класу задач розміщення об'єктів, що в теорії дослідження операцій отримали назву Facility Location Problems. Ці задачі характеризуються наявністю множини потенційних місць розташування об'єктів обслуговування, множини клієнтів або споживачів, що потребують обслуговування, та необхідністю визначення оптимального набору об'єктів для відкриття та схеми розподілу клієнтів між ними. Математичні моделі таких задач формуються як задачі цілочислового або змішаного цілочислового програмування, що належать до класу NP-складних задач.

У сучасній логістиці виділяють два основні рівні організації транспорт-

них систем: макрологістичний та мікрологістичний [5]. Макрологістичні системи функціонують в рамках взаємодії кількох суб'єктів господарювання і характеризуються значною територіальною розподіленістю. Такі системи використовуються міжнародними та транснаціональними компаніями, а також посередницькими організаціями для організації глобальних перевезень. Мікрологістичні системи розв'язують завдання взаємодії одного або кількох підприємств з єдиним економічним інтересом у межах обмеженої географічної зони.

На макрологістичному рівні центральною задачею є оптимізація глобальних транспортних перевезень, що в загальному випадку зводиться до раціонального розміщення проміжних вантажообробних вузлів з метою мінімізації сукупних витрат на транспортування вантажів [6]. На мікрологістичному рівні основну увагу приділяють оптимізації кільцевих маршрутів доставки та оперативному плануванню використання транспортних засобів.

Найбільш поширеним класом макрологістичних систем є централізовані трирівневі мережі. Перший рівень такої мережі утворює центр постачання, що виконує функції виробництва або консолідації товарів. Другий рівень представлений множиною проміжних терміналів, що забезпечують розподіл, тимчасове зберігання та перевантаження вантажів. Третій рівень складається з кінцевих споживачів, які отримують продукцію через мережу терміналів.

Формально кожна логістична мережа як територіально розподілений об'єкт може бути представлена у вигляді кортежу $s = (E, R, G)$, що включає множину елементів системи E , множину зв'язків між елементами R та множину місць розташування елементів G [7]. Множина елементів містить центр постачання, термінали та споживачів. Множина зв'язків задається матрицею суміжності $[s_{ij}]$, елементи якої визначають наявність ($s_{ij} = 1$) або відсутність ($s_{ij} = 0$) прямого транспортного з'єднання між відповідними елементами мережі. Множина місць розташування визначає географічну прив'язку кожного елемента системи. Топологічна структура централізованої трирівневої мережі характеризується наступними властивостями [4, 6]. Кожен споживач має бути пі-

д'єднаний до одного з терміналів або безпосередньо до центру постачання. До кожного терміналу має бути під'єднано більше одного споживача, оскільки економічна доцільність створення терміналу виникає лише при обслуговуванні певної сукупності споживачів. Кожен споживач під'єднується до того терміналу, для якого сумарні витрати на транспортування є мінімальними. Кожен термінал має безпосередній зв'язок з центром постачання. Кількість терміналів в мережі обмежена певним діапазоном, що визначається економічними та технологічними міркуваннями.

Задача оптимізації топологічної структури полягає у визначенні оптимальної кількості терміналів, їх географічного розміщення та розподілу споживачів між терміналами. Критерієм оптимальності виступає мінімізація наведених витрат на функціонування логістичної мережі при дотриманні обмежень на якість обслуговування. До складу витрат входять капітальні витрати на створення терміналів, експлуатаційні витрати на їх утримання та функціонування, а також транспортні витрати на доставку вантажів між елементами мережі. У випадку замкнених логістичних систем додатково враховуються витрати на організацію зворотних потоків від споживачів до центрів переробки [4].

Важливою характеристикою логістичної мережі є оперативність доставки, що визначається максимальним часом транспортування вантажів до найвіддаленішого споживача. Час доставки складається з часу транспортування від центру до терміналу, часу обробки вантажу в терміналі та часу доставки від терміналу до кінцевого споживача. Для багатьох типів вантажів існують обмеження на допустимий час доставки, що трансформується у відповідні обмеження на топологічну структуру мережі.

Задача оптимізації топологічної структури логістичної мережі належить до класу дискретних комбінаторних задач високої обчислювальної складності. Загальна кількість можливих варіантів побудови мережі визначається кількістю способів вибору підмножини терміналів з множини потенційних місць їх розташування та кількістю способів розподілу споживачів між обраними терміналами. Ця кількість зростає експоненціально зі збільшенням розмірності задачі,

що робить неможливим застосування методів повного перебору для задач практичної розмірності.

Специфіка задачі оптимізації на етапі реінжинірингу полягає у необхідності врахування наявної інфраструктури існуючої мережі. У процесі реінжинірингу виникають додаткові варіанти рішень, що включають збереження існуючих терміналів без змін, модернізацію терміналів для покращення їх характеристик, демонтаж неефективних терміналів з можливою реалізацією обладнання, а також створення нових терміналів у місцях, де їх раніше не було. Наявність таких альтернатив призводить до ускладнення структури цільової функції та появи множинних локальних екстремумів. Це визначає необхідність застосування спеціалізованих методів оптимізації, здатних ефективно працювати в умовах багатоекстремальних функцій.

1.2 Аналіз сценаріїв вирішення задачі оптимізації топологічних структур на етапі реінжинірингу

Процес реінжинірингу логістичних мереж передбачає фундаментальне переосмислення існуючої системи з можливістю радикальних змін її структури, топології та параметрів функціонування [8]. На відміну від поточної модернізації, що обмежується локальними покращеннями окремих елементів, реінжиніринг спрямований на комплексну трансформацію системи для досягнення якісно нового рівня ефективності.

Необхідність реінжинірингу логістичної мережі виникає під впливом сукупності зовнішніх та внутрішніх факторів. До зовнішніх факторів належать зміни ринкової кон'юнктури, що проявляються у появі нових споживачів або втраті існуючих, зміні обсягів та структури попиту, трансформації географії збуту продукції. Суттєвий вплив здійснюють зміни транспортної інфраструктури регіону, включаючи введення в експлуатацію нових транспортних магістралей, зміни транспортних тарифів та умов перевезень. Все більшого значення

набувають екологічні вимоги, що стимулюють розвиток "зеленої" логістики та впровадження принципів реверсивної логістики для організації зворотних потоків.

Внутрішні фактори реінжинірингу пов'язані з необхідністю підвищення операційної ефективності існуючої системи. Накопичення критичного рівня не-ефективності окремих елементів мережі, моральне та фізичне старіння обладнання терміналів, невідповідність пропускнуої спроможності вузлів фактичним обсягам вантажопотоків створюють об'єктивні передумови для радикального перепроектування системи. Важливим внутрішнім фактором є також зміна стратегічних пріоритетів компанії, що може вимагати переорієнтації логістичної системи на нові цільові показники .

Специфіка задачі оптимізації на етапі реінжинірингу полягає у необхідності прийняття рішень щодо долі кожного елемента існуючої мережі. Для кожного наявного терміналу необхідно визначити один з можливих варіантів дій: збереження терміналу без змін у випадку, коли його характеристики відповідають новим вимогам; модернізація терміналу для покращення його пропускнуої спроможності, технологічного оснащення або інших параметрів; демонтаж терміналу з виведенням обладнання з експлуатації та можливою реалізацією ресурсів, що можуть бути повторно використані. Одночасно необхідно розглянути можливість створення нових терміналів у місцях, де їх раніше не існувало, якщо це дозволить підвищити ефективність системи в цілому [9].

Аналогічні рішення приймаються щодо транспортних зв'язків між елементами мережі. Існуючі маршрути доставки можуть бути збережені, модифіковані шляхом зміни типу транспортних засобів або технології перевезень, або повністю ліквідовані з організацією нових маршрутів, що краще відповідають змінним умовам функціонування системи.

Процес реінжинірингу характеризується наявністю додаткових витрат порівняно з проектуванням нової мережі. До витрат на створення та експлуатацію терміналів додаються витрати на модернізацію існуючих об'єктів та витрати на демонтаж непотрібних елементів інфраструктури. Водночас виникає мо-

жливість часткової компенсації витрат за рахунок реалізації обладнання та інших ресурсів, що вивільняються у результаті демонтажу. Структура цільової функції оптимізації суттєво ускладнюється через необхідність врахування всіх цих компонентів витрат [10].

Формалізація витрат на реінжиніринг передбачає виділення наступних складових для кожного елемента мережі: витрати на створення нового терміналу у місці, де його раніше не існувало; витрати на модернізацію існуючого терміналу, що є меншими за витрати на створення нового; витрати на демонтаж терміналу, що виводиться з експлуатації; вартість ресурсів, що можуть бути повторно використані або реалізовані після демонтажу. Аналогічна структура витрат визначається для транспортних зв'язків між елементами мережі, включаючи витрати на організацію нових маршрутів, модернізацію існуючих та ліквідацію неефективних з'єднань.

Важливою особливістю задачі реінжинірингу є неунімодальний характер функції витрат залежно від кількості терміналів у мережі. Якщо у класичній задачі проектування функція витрат має єдиний мінімум, то у задачі реінжинірингу можуть виникати множинні локальні екстремуми [11]. Це пояснюється тим, що на різних етапах зміни кількості терміналів можуть бути використані різні комбінації наявних та нових об'єктів інфраструктури, що призводить до немотонної зміни сукупних витрат.

Для розв'язання задачі оптимізації топологічних структур на етапі реінжинірингу можуть бути застосовані різні підходи, кожен з яких має свої переваги та обмеження.

Перший сценарій передбачає повне ігнорування існуючої інфраструктури та розв'язання класичної задачі проектування нової мережі. Цей підхід дозволяє отримати теоретично оптимальну конфігурацію системи без обмежень, що накладаються історично сформованою структурою. Перевагою сценарію є можливість досягнення глобального оптимуму з точки зору поточних умов функціонування та перспективних вимог. Основними недоліками є максимальні капітальні витрати на повну реконструкцію системи, тривалий період реалізації

проекту та високі ризики, пов'язані з одночасною зміною всіх елементів мережі. Даний сценарій може бути виправданим у випадках кардинальної зміни бізнес-моделі компанії або виходу на принципово нові ринки збуту.

Другий сценарій базується на збереженні існуючої структури мережі з оптимізацією лише параметрів функціонування окремих елементів. Такий підхід мінімізує витрати на реконфігурацію системи та забезпечує безперервність операційної діяльності у період трансформації. Однак він може не забезпечити достатнього підвищення ефективності у разі значних змін умов функціонування. Даний сценарій є виправданим лише при незначних відхиленнях фактичних параметрів від проектних значень, коли основні проблеми пов'язані з неоптимальними режимами експлуатації, а не зі структурними недоліками мережі [12].

Третій сценарій являє собою збалансований підхід, що передбачає вибірковий реінжиніринг з врахуванням можливості використання наявної інфраструктури. Цей підхід вимагає розробки спеціалізованих математичних моделей, що дозволяють порівнювати варіанти з різним ступенем використання існуючих елементів системи. Перевагою сценарію є можливість знаходження компромісу між радикальністю змін та витратами на їх реалізацію. Методологічно цей підхід передбачає формування множини допустимих варіантів побудови мережі, що включає як збереження існуючих елементів, так і створення нових, з подальшим вибором найкращого варіанту за критерієм мінімуму наведених витрат. Саме цей сценарій розглядається у даній роботі як найбільш раціональний з точки зору балансу між витратами на реінжиніринг та очікуваним підвищенням ефективності.

Четвертий сценарій передбачає поетапний реінжиніринг з послідовною трансформацією окремих сегментів мережі. Такий підхід дозволяє розподілити капітальні витрати у часі, зменшити операційні ризики та забезпечити можливість коригування рішень на основі результатів попередніх етапів. Недоліком є неможливість досягнення глобального оптимуму через обмеження, що накладаються проміжними станами системи. Цей сценарій доцільно застосовувати при обмежених інвестиційних можливостях або високому рівні невизначеності

щодо майбутніх умов функціонування .

Для реалізації третього сценарію необхідно визначити межі області пошуку оптимального рішення. Нижня межа кількості терміналів визначається мінімально можливою конфігурацією, що передбачає наявність єдиного терміналу, розташованого безпосередньо у центрі виробництва. Верхня межа може бути визначена шляхом розв'язання задачі синтезу оптимальної топології без врахування існуючих терміналів, що дає оцінку максимальної доцільної кількості вузлів у мережі [13]. Пошук оптимального рішення здійснюється шляхом послідовного аналізу варіантів з різною кількістю терміналів у визначеному діапазоні. Для кожного фіксованого значення кількості терміналів розв'язується задача визначення їх оптимального розміщення та розподілу споживачів між терміналами. Такий підхід дозволяє врахувати багатоекстремальний характер цільової функції та знайти глобальне оптимальне рішення на заданому інтервалі зміни кількості вузлів.

Прийняття рішень в умовах реінжинірингу ускладнюється наявністю невизначеності щодо майбутніх умов функціонування системи. Обсяги попиту, транспортні тарифи, вартість ресурсів та інші параметри можуть змінюватися у часі, що впливає на оцінку ефективності різних варіантів побудови мережі. Для врахування невизначеності можуть застосовуватися підходи на основі інтервального аналізу, що передбачають задання параметрів у вигляді інтервалів можливих значень та пошук рішень, стійких до варіацій вхідних даних .

Важливим аспектом аналізу сценаріїв є врахування багатокритеріального характеру оцінки якості логістичної мережі. Окрім мінімізації витрат, необхідно забезпечити прийнятний рівень оперативності доставки, що характеризується часом транспортування вантажів до споживачів [14]. У випадку, коли зміна часу доставки є некритичною при переході між варіантами, двокритеріальна задача може бути зведена до однокритеріальної з додатковим обмеженням на максимально допустимий час доставки. Якщо ж оперативність є критичним фактором, необхідно застосовувати методи багатокритеріальної оптимізації з побудовою узагальненої функції корисності для порівняння варіантів.

Часова складність розв'язання задачі реінжинірингу визначається необхідністю розв'язання двох допоміжних задач: визначення верхньої межі кількості терміналів шляхом розв'язання класичної задачі синтезу та послідовного розв'язання задач оптимального розміщення для кожного значення кількості терміналів у визначеному діапазоні. Сумарна обчислювальна складність методу має порядок, що пропорційний подвоєній складності розв'язання класичної задачі структурно-топологічної оптимізації логістичної мережі.

1.3 Змістовна та формальна постановка задачі

Розглядається централізована трирівнева логістична мережа, що функціонує в умовах, які змінилися порівняно з моментом її первинного проектування. Мережа включає центр постачання, множину проміжних терміналів та множину кінцевих споживачів. Центр постачання здійснює виробництво або консолідацію продукції, термінали забезпечують її розподіл та тимчасове зберігання, споживачі отримують вантажі через мережу терміналів.

У результаті зміни ринкових умов, появи нових споживачів, зміни обсягів попиту, коливання транспортних тарифів та інших факторів існуюча конфігурація мережі перестала відповідати вимогам ефективного функціонування. Виникає необхідність проведення реінжинірингу логістичної мережі з метою приведення її структури та топології у відповідність до нових умов експлуатації[8].

Задано множину елементів існуючої логістичної мережі, що включає центр постачання, множину наявних терміналів та множину споживачів з відомими обсягами попиту. Для кожного споживача відомі його географічні координати та потреба у продукції. Для кожного існуючого терміналу відомі місце розташування, пропускна спроможність та поточні експлуатаційні характеристики.

Відома топологічна структура існуючої мережі, що визначається місцями розташування елементів та зв'язками між ними. Для кожного можливо-го тер-

міналу відомі наведені витрати на його створення, модернізацію або демонтаж, а також вартість ресурсів, що можуть бути реалізовані після де-монтажу обладнання. Для кожної пари елементів мережі відомі транспортні витрати на доставку вантажів та часові параметри транспортування.

Необхідно визначити оптимальний варіант топологічної структури логістичної мережі, що забезпечує мінімум наведених витрат на функціонування системи при дотриманні обмежень на максимально допустимий час доставки вантажів споживачам.

Формальна постановка задачі.

Введемо множину елементів логістичної мережі $I = \{i : i = 1, n\}$, де елемент $i=1$ відповідає центру постачання, а решта елементів можуть виступати як споживачі або як можливі місця розташування терміналів[5]. Варіант побудови логістичної мережі $s \in S$ задається матрицею суміжності $[s_{ij}]$, $i, j = 1, n$, де $s_{ij} = 1$ означає наявність безпосереднього зв'язку між елементами i та j , а $s_{ij} = 0$ – його відсутність. Діагональний елемент $s_{ij} = 1$ вказує на наявність терміналу у місці розташування i -го споживача. Існуюча топологічна структура мережі задається матрицею суміжності $[s'_{ij}]$, що визначає наявні зв'язки між елементами до проведення реінжинірингу.

Цільова функція задачі визначає наведені витрати на реінжиніринг та функціонування логістичної мережі:

$$k_1(s', s) = \sum_{j=1}^n [a_j(1-s'_{jj})s_{jj} + b_j s'_{jj} s_{jj} + c_j(1-s_{jj})s'_{jj} - d_j(1-s_{jj})s'_{jj}] + \\ + \sum_{j=1}^n \sum_{i \neq j} [e_{ji}(1-s'_{ji})s_{ji} + f_{ji} s'_{ji} s_{ji} + g_{ji}(1-s_{ji})s'_{ji} - h_{ji}(1-s_{ji})s'_{ji}] \rightarrow \min_{s \in S}$$

де a_j – витрати на створення j -го терміналу в новій структурі;

b_j – витрати на модернізацію j -го терміналу;

c_j – витрати на демонтаж j -го терміналу існуючої мережі;

d_j – вартість ресурсів, які можуть бути повторно використані після демонтажу обладнання терміналу;

e_{ji}, f_{ji}, h_{ji} – аналогічні витрати для транспортних зв'язків між елементами i та j .

Додатковий критерій оперативності визначає максимальний час доставки вантажів [5]:

$$k_2(s) = \tau(s) = \max_{1 \leq i \leq n} \{ \tau_{PT}(s) + \tau_T(s) + \tau_{TS}(s) \} \rightarrow \min_{s \in S},$$

де $\tau_{PT}(s)$ – максимальний час транспортування вантажів від центру виробництва до терміналів;

$\tau_T(s)$ – максимальний час обробки вантажів у терміналах;

$\tau_{TS}(s)$ – максимальний час доставки вантажів від терміналів до споживачів.

Множина допустимих варіантів визначається наступними структурними обмеженнями трирівневої централізованої мережі:

$$s_{ij}, s_{ji} \in \{0,1\}, \quad i, j = 1, n; \quad s_{11} = 1;$$

$$\sum_{i=j}^n s_{ij} \geq 1, \quad \forall j = 1, n;$$

$$\sum_{j=1, j \neq i}^n s_{ij} = n + \sum_{i=1}^n s_{ii};$$

$$s_{ii} = 1 \rightarrow s_{i1} = 1, \quad \forall i = 1, n;$$

$$s_{ii} \wedge s_{ij} = 1 \rightarrow ij = \arg \min_{1 \leq j \leq n} c_{ji}, \quad \forall i, j = 1, n.$$

Перше обмеження визначає бінарний характер змінних та наявність центру постачання; друге обмеження забезпечує під'єднання кожного споживача до

терміналу; третє обмеження визначає загальну кількість зв'язків у трирівневій структурі; четверте обмеження гарантує зв'язок кожного терміналу з центром; п'яте обмеження забезпечує під'єднання споживача до терміналу з мінімальними транспортними витратами.

Задача полягає у знаходженні варіанта $s^0 \in S$ що забезпечує мінімум наведених витрат на реінжиніринг та функціонування логістичної мережі при дотриманні всіх структурних обмежень трирівневої централізованої системи. У випадку, коли час доставки є критичним фактором вводиться додаткове обмеження $\max_{1 \leq i \leq n} \tau_i(s) \leq \tau^*$, де τ^* – максимально допустимий час доставки вантажів споживачам.

1.4 Постановка задач дослідження

На основі проведеного системного аналізу предметної області, дослідження сценаріїв вирішення задачі та сформульованої змістовної і формальної постановки задачі визначається комплекс задач дослідження, спрямований на досягнення поставленої мети роботи. Структура дослідження побудована таким чином, щоб забезпечити послідовне розв'язання теоретичних, методологічних та практичних питань оптимізації топологічних структур логістичних мереж на етапі реінжинірингу.

Першою задачею дослідження є проведення огляду та порівняльного аналізу існуючих методів структурно-топологічної оптимізації логістичних мереж. Актуальність цієї задачі обумовлена наявністю широкого спектру підходів до розв'язання задач розміщення об'єктів, кожен з яких має свої переваги та обмеження. Необхідно дослідити точні методи, зокрема метод гілок і меж, та визначити межі їх практичної застосовності з урахуванням експоненціального зростання обчислювальної складності при збільшенні розмірності задачі. Слід проаналізувати евристичні підходи, що базуються на жадібних стратегіях та про-

цедурах локального пошуку, а також метаевристичні методи, включаючи імітацію відпалу, генетичні алгоритми та їх гібридні модифікації. Для кожного класу методів необхідно виявити характерні особливості з точки зору точності отримуваних розв'язків, часової складності, масштабованості та стійкості до варіацій вхідних даних. Особливу увагу слід приділити методам, що ефективно працюють з комбінаторними задачами високої розмірності та дозволяють знаходити наближені розв'язки за прийнятний час при збереженні достатнього рівня точності. Важливим аспектом аналізу є визначення можливості адаптації існуючих методів для розв'язання задачі реінжинірингу, що характеризується специфічною структурою цільової функції та необхідністю врахування наявної інфраструктури. Результатом виконання цієї задачі має стати обґрунтований вибір двох методів для подальшої детальної розробки та програмної реалізації.

Другою задачею дослідження є адаптація методу покоординатної оптимізації для розв'язання поставленої задачі реінжинірингу топологічної структури логістичної мережі. Вибір цього методу обумовлений його здатністю знаходити локально оптимальні розв'язки з відносно невисокою обчислювальною складністю та простотою програмної реалізації. Метод має забезпечувати послідовне покращення початкового варіанта структури мережі шляхом циклічної оптимізації розміщення кожного терміналу при фіксованих позиціях інших елементів системи. Необхідно формалізувати процедуру формування початкового наближення, яка має враховувати топологію існуючої мережі та забезпечувати належну якість стартової точки для подальшої оптимізації. Слід розробити механізм ітераційного покращення, що передбачає послідовний перегляд можливих варіантів розміщення кожного терміналу з вибором найкращого за критерієм мінімуму цільової функції. Необхідно визначити критерій завершення ітераційного процесу, який фіксує досягнення локального оптимуму при неможливості подальшого покращення розв'язку зміною положення жодного з терміналів. Алгоритм має бути адаптований для врахування специфіки задачі реінжинірингу, що включає можливість збереження існуючих терміналів без змін, їх модернізації для покращення характеристик або повного демонтажу з вивільненням ресур-

сів. Відповідно, цільова функція методу має включати всі компоненти витрат на реінжиніринг, передбачені математичною моделлю задачі. Результатом виконання цієї задачі має стати повністю формалізований алгоритм методу покоординатної оптимізації, готовий до програмної реалізації.

Третьою задачею дослідження є розробка еволюційного методу на основі генетичного алгоритму для розв'язання поставленої задачі оптимізації топологічної структури логістичної мережі. Вибір генетичного алгоритму як другого методу обумовлений його здатністю ефективно досліджувати великі простори пошуку та уникати локальних оптимумів завдяки стохастичній природі еволюційних операторів. Метод повинен реалізувати пошук у просторі можливих рішень за рахунок моделювання процесів природного відбору, схрещування та мутації особин популяції. Необхідно визначити спосіб кодування варіантів побудови мережі у вигляді хромосом генетичного алгоритму, що має забезпечувати однозначне відображення топологічної структури та ефективне застосування генетичних операторів. Слід розробити оператор селекції, що визначає механізм відбору батьківських особин для формування нового покоління на основі значень функції пристосованості. Оператор схрещування має забезпечувати комбінування генетичного матеріалу батьківських хромосом для створення нащадків, що успадковують перспективні характеристики обох батьків. Оператор мутації повинен вносити випадкові зміни до хромосом для підтримання генетичного різноманіття популяції та можливості виходу з локальних оптимумів. Функцію пристосованості необхідно адаптувати для врахування структури витрат на реінжиніринг, що включає витрати на створення нових терміналів, модернізацію існуючих, демонтаж непотрібних та компенсацію від реалізації вивільнених ресурсів. Особливу увагу слід приділити механізмам збереження різноманітності популяції для запобігання передчасній збіжності алгоритму до субоптимальних розв'язків. Необхідно також визначити критерії зупинки алгоритму, що можуть базуватися на досягненні заданої кількості поколінь, стабілізації найкращого розв'язку протягом певної кількості ітерацій або досягненні цільового значення функції пристосованості. Результатом виконання цієї задачі

має стати повністю специфікований еволюційний метод з визначеними параметрами та операторами генетичного алгоритму.

Четвертою задачею дослідження є розробка програмного забезпечення для розв'язання задачі оптимізації топологічних структур логістичних мереж на етапі реінжинірингу. Необхідність програмної реалізації обумовлена потребою практичної перевірки розроблених методів та забезпечення можливості їх застосування для розв'язання реальних задач. Слід здійснити вибір та обґрунтування мови програмування з урахуванням вимог до обчислювальної ефективності, наявності бібліотек для роботи з математичними обчисленнями та засобів візуалізації результатів. Необхідно розробити архітектуру програмного комплексу, що забезпечує модульність, розширюваність та зручність супроводження коду. Архітектура має передбачати чітке розділення компонентів для роботи з вхідними даними, реалізації алгоритмів оптимізації та представлення результатів. Слід реалізувати алгоритми обох методів оптимізації відповідно до розроблених специфікацій з дотриманням принципів структурного програмування та забезпеченням коректності обчислень. Програмне забезпечення має підтримувати введення вхідних даних про параметри мережі, включаючи координати елементів, матрицю відстаней, обсяги вантажопотоків та параметри витрат. Необхідно реалізувати візуалізацію отриманих результатів у вигляді топологічної схеми мережі з відображенням терміналів, споживачів та зв'язків між ними, що дозволить наочно оцінити якість знайденого розв'язку. Програмний комплекс має також забезпечувати виведення кількісних характеристик розв'язку, включаючи значення цільової функції, кількість терміналів та час виконання оптимізації. Результатом виконання цієї задачі має стати працездатний програмний комплекс, готовий до проведення обчислювальних експериментів.

П'ятою задачею дослідження є проведення обчислювального експерименту для дослідження характеристик розроблених методів оптимізації. Метою експерименту є отримання емпіричних оцінок точності та часової складності методів, а також їх порівняльний аналіз для формулювання рекомендацій щодо практичного застосування. Необхідно розробити методіку проведення експе-

рименту, що включає визначення способу генерації тестових задач різної розмірності з контрольованими параметрами. Тестові задачі мають охоплювати діапазон розмірностей від малих, для яких можливе знаходження точного розв'язку методом повного перебору, до практично значущих розмірностей з кількістю елементів мережі порядку кількох десятків. Слід визначити параметри методів оптимізації, що будуть використовуватися в експерименті, включаючи початкові умови для методу покоординатної оптимізації та параметри генетичного алгоритму. На основі результатів експерименту необхідно дослідити точність методів шляхом порівняння отриманих розв'язків з еталонними значеннями для задач малої розмірності та між собою для задач більшої розмірності. Слід провести аналіз часової складності методів залежно від розмірності задачі та виконати апроксимацію отриманих залежностей аналітичними функціями для можливості прогнозування часу розв'язання задач довільної розмірності. Необхідно також дослідити вплив параметрів методів на якість отримуваних розв'язків та стійкість результатів при багаторазовому запуску алгоритмів з різними початковими умовами. Отримані результати дозволять виконати обґрунтований порівняльний аналіз методів та сформулювати практичні рекомендації щодо вибору найбільш придатного методу залежно від розмірності задачі, вимог до точності розв'язку та обмежень на час обчислень.

Розв'язання сформульованих задач дослідження дозволить створити практично застосовну методологію оптимізації топологічних структур централізованих логістичних мереж на етапі їх реінжинірингу. Комплексний характер дослідження, що охоплює теоретичний аналіз методів, їх адаптацію до специфіки задачі, програмну реалізацію та експериментальну перевірку, забезпечить обґрунтованість отриманих результатів та можливість їх практичного використання при проектуванні та модернізації реальних логістичних систем.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Огляд методів структурно-топологічної оптимізації логістичних мереж

Задача оптимізації топологічних структур логістичних мереж належить до класу дискретних задач комбінаторної оптимізації. Загальна кількість можливих варіантів побудови мережі для заданої кількості потенційних місць розташування терміналів визначається співвідношенням [8]:

$$N(n) = \frac{1}{2} \sum_{l=1}^n C_n^l = \frac{1}{2} \sum_{l=1}^n \frac{n!}{l!(n-l)!} = 2^{n-1},$$

де $N(n)$ – кількість можливих варіантів топологічних структур;

n – кількість можливих місць розташування терміналів;

i – кількість терміналів, що одночасно розміщуються у вибраному варіанті.

Експоненціальне зростання кількості варіантів зі збільшенням розмірності задачі робить неможливим застосування методів повного перебору для практичних задач, що містять більше кількох десятків елементів. Це зумовлює необхідність використання наближених методів, здатних знаходити ефективні рішення за прийнятний час. Методи розв'язання задач структурно-топологічної оптимізації можна класифікувати за кількома ознаками. За критерієм точності розрізняють точні методи, що гарантують знаходження глобального оптимуму, та наближені методи, що забезпечують отримання рішень із заданою точністю. За принципом організації пошуку виділяють методи спрямованого перебору, що послідовно звужують область пошуку, локальні методи, що покращують поточне рішення в околі базової точки, та популяційні методи, що одночасно досліджують множину потенційних рішень [15].

Точні методи базуються на систематичному дослідженні всіх можливих варіантів з відсіканням завідомо неоптимальних гілок дерева рішень. До цього

класу належать методи гілок та меж, який є класичним точним методом комбінаторної оптимізації, який гарантує знаходження глобально оптимального розв'язку задачі розміщення об'єктів. Вперше запропонований у 1960 році, цей метод знайшов широке застосування для розв'язання задач розміщення складів, терміналів та інших логістичних об'єктів [6].

Сутність методу полягає в систематичному переборі всіх можливих варіантів розв'язку шляхом побудови дерева рішень з одночасним відсіканням неперспективних гілок. На кожному кроці алгоритму множина допустимих розв'язків розбивається на підмножини (розгалуження), для кожної з яких обчислюються нижня та верхня межі значення цільової функції. Якщо нижня межа для деякої підмножини перевищує вже знайдене значення цільової функції, ця гілка відсікається як неперспективна.

Для задачі оптимізації топологічної структури логістичної мережі метод гілок і меж забезпечує отримання точного оптимуму. Проте його практичне застосування обмежене задачами невеликої розмірності. Часова складність методу в найгіршому випадку є експоненційною $O(2^n)$, де n – кількість можливих місць розташування терміналів. За даними досліджень, метод ефективно працює для задач з кількістю елементів до 30–50, при більшій розмірності час розв'язання стає неприйнятним.

Серед переваг методу гілок і меж слід відзначити гарантію знаходження глобального оптимуму та можливість оцінки якості наближених розв'язків. Основними недоліками є висока обчислювальна складність для задач великої розмірності та значні вимоги до оперативної пам'яті для зберігання дерева рішень.

Методи спрямованого перебору варіантів являють собою компроміс між точністю та обчислювальною ефективністю. Їх сутність полягає у попередньому визначенні меж області пошуку та послідовному аналізі варіантів у напрямку покращення цільової функції [11]. Для задачі оптимізації топологічної структури логістичної мережі такий підхід передбачає визначення мінімальної та максимальної кількості терміналів, що гарантовано містить оптимальне рішення.

ня. Нижня межа відповідає конфігурації з єдиним терміналом у центрі виробництва, верхня межа визначається шляхом розв'язання допоміжної задачі синтезу без урахування існуючої інфраструктури. Пошук оптимального рішення здійснюється послідовним аналізом варіантів із зміною кількості терміналів у визначеному діапазоні [7].

Локальні методи оптимізації базуються на ітеративному покращенні поточного рішення шляхом дослідження його околу. До цього класу належать методи покоординатної оптимізації, методи найшвидшого спуску, методи спряжених градієнтів та їх модифікації для дискретних задач. Метод покоординатної оптимізації передбачає послідовну оптимізацію за кожною координатою при фіксованих значеннях інших змінних. Для задачі оптимізації логістичної мережі це означає послідовний перебір можливих місць розташування кожного терміналу при фіксованому розміщенні решти терміналів. Процедура циклічно повторюється до досягнення локального екстремуму цільової функції. Перевагою локальних методів є відносно низька обчислювальна складність, недоліком – можливість зупинки в локальному екстремумі, що не є глобальним оптимумом.

Метаевристичні методи являють собою загальні стратегії пошуку, що дозволяють уникнути захоплення локальними екстремумами та досліджувати ширшу область простору рішень. До найбільш поширених належать генетичні алгоритми, методи імітації відпалу, мурашині алгоритми та методи рою частинок. Генетичні алгоритми моделюють процеси природної еволюції та реалізують ітеративне покращення популяції потенційних рішень через операції селекції, схрещування та мутації. Кожне рішення кодується у вигляді хромосоми, а якість рішення оцінюється функцією пристосованості.

Метод імітації відпалу був запропонований у 1983 році як аналогія процесу відпалу в металургії, де контрольоване охолодження матеріалу дозволяє досягти стану з мінімальною енергією та підвищеною структурною стабільністю системи [7].

Алгоритм починає з деякого початкового розв'язку та ітеративно генерує

сусідні розв'язки шляхом випадкових модифікацій поточного. Ключовою особливістю методу є механізм прийняття рішень: якщо новий розв'язок кращий за поточний, він завжди приймається; якщо гірший - приймається з імовірністю

$$p = \exp\left(-\frac{\Delta C}{T}\right),$$

де ΔC – погіршення значення цільової функції (різниця між новим та поточним значенням);

T – параметр «температури».

На початку пошуку температура висока, що дозволяє приймати навіть суттєво гірші розв'язки для виходу з локальних мінімумів. З часом температура знижується за заданим законом охолодження, і алгоритм поступово концентрується на покращенні поточного розв'язку.

Для задач розміщення об'єктів логістичної мережі метод імітації відпалу демонструє здатність знаходити близькі до оптимальних розв'язки за прийнятний час. Часова складність алгоритму визначається кількістю ітерацій та складністю обчислення цільової функції, що зазвичай становить $O(n^2 \cdot I)$, де I – кількість ітерацій [9].

Перевагами методу імітації відпалу є простота реалізації, здатність уникати локальних мінімумів та можливість адаптації до різних типів задач оптимізації. До недоліків належать необхідність ретельного налаштування параметрів (початкова температура, закон охолодження, критерій зупинки), відсутність гарантії знаходження глобального оптимуму та залежність якості розв'язку від початкового наближення.

Порівняльний аналіз розглянутих методів структурно-топологічної оптимізації логістичних мереж зображено в таблиці 2.1, що дозволяє виділити їх характерні особливості та сфери доцільного практичного застосування.

Таблиця 2.1 – Порівняння методів оптимізації логістичних мереж

Метод	Тип	Точність	Часова складність	Розмірність задачі
Гілок і меж	Точний	Глобальний оптимум	$O(2^n)$	До 30–50 елементів
Імітації відпалу	Метаевристичний	Наближений	$O(2^n \cdot I)$	До 200–500 елементів
Покоординатної оптимізації	Евристичний	Локальний оптимум	$O(2^n)$	До 100–200 елементів
Генетичний алгоритм	Еволюційний	Наближений	$O(n \cdot P \cdot G)$	До 500–1000 елементів

Для задачі оптимізації топологічної структури централізованої логістичної мережі на етапі реінжинірингу перевага надається методу покоординатної оптимізації та еволюційному методу. Цей вибір обумовлений такими факторами:

Метод гілок і меж, незважаючи на гарантію оптимальності, є непрактичним для реальних логістичних мереж, які зазвичай містять понад 50 елементів. Метод імітації відпалу, хоча й демонструє хороші результати для задач розміщення, потребує значних зусиль для налаштування параметрів і не забезпечує стабільної якості розв'язків, в той час як метод покоординатної оптимізації поєднує прийнятну точність (середня похибка менше 1% для задач до 40 елементів) з помірною обчислювальною складністю. Генетичний алгоритм забезпечує ефективний пошук у великому просторі розв'язків завдяки механізмам селекції, схрещування та мутації, що дозволяє уникати локальних екстремумів.

2.2 Метод покоординатної оптимізації

Метод покоординатної оптимізації (МПО) належить до класу локальних ітеративних методів пошуку екстремуму функцій багатьох змінних. Основна

ідея методу полягає у послідовній оптимізації цільової функції за кожною координатою при фіксованих значеннях решти змінних [7, 15]. Для задачі оптимізації топологічної структури логістичної мережі це означає по чергове визначення оптимального розташування кожного терміналу за умови незмінного положення інших терміналів системи.

Теоретичним підґрунтям методу є принцип декомпозиції багатовимірної задачі оптимізації на послідовність одновимірних підзадач. Нехай цільова функція $f(x_1, x_2, \dots, x_n)$ залежить від n змінних. На кожній ітерації методу фіксуються значення всіх змінних, окрім однієї, і здійснюється пошук оптимуму за обраною координатою. Процедура циклічно повторюється для всіх координат до виконання критерію збіжності [7]. Для неперервних унімодальних функцій метод гарантує збіжність до локального екстремуму, однак для багатоекстремальних функцій результат суттєво залежить від вибору початкової точки.

Адаптація методу покоординатної оптимізації для задачі реінжинірингу топологічної структури логістичної мережі передбачає врахування дискретного характеру змінних та специфіки цільової функції [11]. Змінними задачі є місця розташування терміналів, що можуть обиратися з дискретної множини потенційних локацій, що збігається з множиною розташування споживачів. Оптимізація за однією координатою зводиться до перебору всіх можливих місць розташування одного терміналу при фіксованому розміщенні решти та вибору варіанта з мінімальним значенням цільової функції.

Для формалізації методу введемо наступні позначення. Нехай w_i^u позначає варіант розміщення терміналів на ітерації i при u -му повному проході по всіх елементах мережі, $\Delta C(w_i^u)$ – значення цільової функції наведених витрат для цього варіанта, l – поточна кількість терміналів у мережі, j – індекс поточного терміналу, що оптимізується. Варіант розміщення w_i^u визначається множиною індексів елементів мережі, в яких розташовані термінали.

Алгоритм методу покоординатної оптимізації для задачі реінжинірингу логістичної мережі включає наступні кроки:

Крок 1. Ініціалізація вхідних даних: визначення множини можливих місць розташування терміналів; встановлення початкової кількості терміналів $l := 1$; встановлення індексу поточного терміналу $j := 1$; обнулення лічильника ітерацій $i := 0$; обнулення лічильника повних проходів $u := 0$; ініціалізація найкращого знайденого варіанта розміщення w^o ; встановлення початкового значення критерію $\Delta C(w_i^u) := \infty$.

Крок 2. Формування початкового варіанта розміщення терміналів w_i^u . Розподіл споживачів між терміналами здійснюється за критерієм мінімуму транспортних витрат:

$$ij = \arg \min_{1 \leq j \leq n} c_{ji}, \quad \forall i, j = \overline{1, n},$$

де c_{ji} – витрати на транспортування вантажів від терміналу j до споживача i .

Обчислення значення цільової функції $\Delta C(w_i^u)$ для сформованого варіанта.

Крок 3. Збільшення лічильника ітерацій $i := i + 1$. Для терміналу з індексом j при фіксованому розміщенні решти $l - 1$ терміналів здійснюється перебір усіх можливих місць його розташування з множини потенційних локацій.

Крок 4. Обчислення значення цільової функції $\Delta C(w_i^u)$ для кожного варіанта розміщення терміналу j . Якщо $\Delta C(w_i^u) \leq \Delta C(w_{i-1}^u)$, зберігається нове значення $\Delta C(w_i^u) := \Delta C(w_i^u)$ та оновлюється найкращий варіант $w^o := w_i^u$. Перехід до кроку 5.

Крок 5. Збільшення індексу поточного терміналу $j := j + 1$. Якщо $j < l$, перехід до кроку 3, інакше – перехід до кроку 6.

Крок 6. Перевірка умови завершення повного проходу. Якщо $u = 0$, зберігається поточний варіант $w_i^{u+1} := w_i^u$, збільшується лічильник проходів $u := u + 1$, встановлюється $j := 1$ і здійснюється перехід до кроку 3. Інакше – перехід до кроку 7.

Крок 7. Перевірка умови збіжності. Якщо $\Delta C(w^u) \leq \Delta C(w^{u-1})$, зберігається оновлений варіант $w_i^{u+1} := w_i^u$, збільшується $u := u + 1$, встановлюється $j := 1$ і здійснюється перехід до кроку 3. Інакше – перехід до кроку 8.

Крок 8. Завершення роботи алгоритму. Результатом є варіант розміщення терміналів w^o з мінімальним серед розглянутих значенням наведених витрат $\Delta C(w^o)$.

Для врахування багатоекстремального характеру цільової функції у задачі реінжинірингу метод застосовується у рамках процедури спрямованого перебору варіантів за кількістю терміналів. Спочатку визначається верхня межа кількості терміналів l_{max} шляхом розв'язання задачі синтезу оптимальної топології без урахування існуючої інфраструктури. Потім для кожного значення кількості терміналів l у діапазоні від l до l_{max} застосовується метод покоординатної оптимізації для визначення найкращого розміщення. Серед отриманих варіантів обирається той, що забезпечує глобальний мінімум цільової функції.

Важливим аспектом практичної реалізації методу є формування початкового варіанта розміщення терміналів. Якість початкового наближення суттєво впливає на кількість ітерацій, необхідних для збіжності, та на ймовірність знаходження глобального оптимуму. Для задачі реінжинірингу доцільно використовувати існуючу конфігурацію мережі як початковий варіант, що дозволяє врахувати накопичений досвід експлуатації системи.

Часова складність методу покоординатної оптимізації визначається кількістю ітерацій, необхідних для збіжності, та обчислювальними витратами на кожній ітерації. На одній ітерації виконується перебір n можливих місць розташування для одного терміналу з обчисленням цільової функції для кожного варіанта. Обчислення цільової функції вимагає визначення розподілу споживачів між терміналами, що має складність $O(n \cdot l)$. Загальна кількість ітерацій до збіжності залежить від структури цільової функції та якості початкового наближення, але в типових випадках не перевищує кількох повних проходів по всіх терміналах.

Перевагами методу покоординатної оптимізації є відносна простота реалізації, висока точність рішень для задач помірної розмірності та детермінований характер алгоритму, що забезпечує відтворюваність результатів. До недоліків методу належить можливість зупинки в локальному екстремумі та зростання часу розв'язання для задач великої розмірності. Для підвищення ймовірності знаходження глобального оптимуму може застосовуватися процедура мультистарту, що передбачає багаторазове розв'язання задачі з різними початковими наближеннями та вибір найкращого результату.

2.3 Еволюційний метод на основі генетичного алгоритму

Генетичні алгоритми (ГА) належать до класу метаевристичних методів оптимізації, що базуються на моделюванні механізмів природної еволюції та природного відбору. Концептуальні основи генетичних алгоритмів були закладені Джоном Холландом у 1970-х роках та отримали подальший розвиток у працях Девіда Голдберга [16]. Ключовою перевагою генетичних алгоритмів є здатність ефективно досліджувати великі простори рішень та уникати захоплення локальними екстремумами завдяки стохастичному характеру пошуку та механізмам підтримки різноманітності популяції.

Принцип роботи генетичного алгоритму базується на еволюції популяції потенційних рішень через послідовне застосування операторів селекції, схрещування та мутації. Кожне рішення кодується у вигляді хромосоми – структури даних, що містить інформацію про значення оптимізуваних змінних. Якість рішення оцінюється функцією пристосованості, що відповідає цільовій функції оптимізаційної задачі. На кожному поколінні з популяції відбираються найбільш пристосовані особини для формування нащадків, що успадковують характеристики батьків з можливими випадковими змінами.

Для задачі оптимізації топологічної структури логістичної мережі хромосома кодує варіант розміщення терміналів у мережі. Найбільш природним спо-

собом кодування є використання бінарного вектора довжиною n , де n – кількість потенційних місць розташування терміналів. Значення 1 у позиції i означає наявність терміналу в i -му місці, значення 0 – його відсутність. Альтернативним підходом є кодування у вигляді списку індексів елементів мережі, в яких розташовані термінали, що забезпечує більш компактне представлення для випадків з невеликою кількістю терміналів відносно загальної кількості елементів.

Функція пристосованості для еволюційного методу визначається на основі цільової функції задачі реінжинірингу. Оскільки задача полягає у мінімізації наведених витрат, а генетичний алгоритм традиційно орієнтований на максимізацію функції пристосованості, застосовується перетворення виду [6]:

$$F(s) = \frac{1}{1 + \Delta C(s)},$$

де $\Delta C(s)$ – значення цільової функції наведених витрат для варіанта побудови мережі s .

Таке перетворення забезпечує додатність функції пристосованості та її зростання при зменшенні витрат.

Оператор селекції забезпечує відбір особин для участі у формуванні наступного покоління. Найбільш поширеними методами селекції є пропорційний відбір (рулетка), турнірний відбір та рангова селекція [15]. При пропорційному відборі ймовірність вибору особини пропорційна її пристосованості. Турнірний відбір передбачає випадковий вибір кількох особин з популяції та відбір найкращої серед них. Для задачі оптимізації топологічної структури логістичної мережі турнірний відбір демонструє кращі результати завдяки меншій чутливості до масштабування функції пристосованості та збереженню селективного тиску протягом всього процесу еволюції.

Оператор схрещування (кросовер) забезпечує обмін генетичним матеріалом між батьківськими хромосомами для формування нащадків. Для бінарного

кодування найчастіше застосовуються одноточкове, двоточкове та рівномірне схрещування. При одноточковому схрещуванні випадково обирається точка розриву, і нащадки формуються шляхом обміну фрагментами хромосом батьків. Рівномірне схрещування передбачає незалежний вибір кожного гена від одного з батьків з однаковою ймовірністю. Для задачі оптимізації логістичної мережі рівномірне схрещування є більш ефективним, оскільки воно не накладає обмежень на структуру успадкування та дозволяє комбінувати вдалі розміщення терміналів з різних частин мережі.

Оператор мутації вносить випадкові зміни до хромосом нащадків, що забезпечує підтримку генетичної різноманітності популяції та можливість дослідження нових областей простору рішень. Для бінарного кодування стандартною є побітова мутація, при якій кожен ген змінюється на протилежний з заданою ймовірністю p_m .

Типові значення ймовірності мутації знаходяться в діапазоні від 0.001 до 0.1. Занадто низька ймовірність мутації призводить до передчасної збіжності алгоритму, занадто висока – до втрати корисної генетичної інформації та переходу до випадкового пошуку.

Адаптація генетичного алгоритму для задачі реінжинірингу топологічної структури логістичної мережі вимагає врахування специфічних обмежень задачі. Не кожна бінарна хромосома відповідає допустимому варіанту побудови мережі, оскільки повинні виконуватися структурні обмеження трирівневої централізованої системи. Для забезпечення допустимості рішень застосовується процедура репарації, що коригує хромосоми після операцій схрещування та мутації. Репарація включає перевірку наявності зв'язків усіх споживачів з терміналами та додавання необхідних терміналів у випадку порушення обмежень.

Алгоритм еволюційного методу на основі генетичного алгоритму для задачі реінжинірингу логістичної мережі включає наступні кроки:

Крок 1. Ініціалізація параметрів алгоритму: визначення розміру популяції N_p ; встановлення ймовірності мутації p_m ; визначення максимальної кількості

покоління i^* ; обнулення лічильника поколінь $i := 0$; ініціалізація найкращого знайденого рішення w^o ; встановлення початкового значення критерію $\Delta C(w_i) := \infty$.

Крок 2. Формування початкової популяції. Генерація N_p випадкових хромосом, що кодують допустимі варіанти розміщення терміналів. Для забезпечення різноманітності популяції застосовується рівномірний розподіл кількості терміналів у початкових хромосомах.

Крок 3. Перехід до наступного покоління $i := i + 1$. Обчислення функції пристосованості $F(s)$ для всіх хромосом популяції. Визначення найкращої хромосоми поточного покоління за максимумом функції пристосованості та збереження відповідного варіанта w_i .

Крок 4. Порівняння з найкращим знайденим рішенням. Якщо $\Delta C(w_{i-1}) > \Delta C(w_i)$ оновлюється найкраще рішення $w^o := w_i$.

Крок 5. Перевірка умови завершення. Якщо $i \geq i^*$, перехід до кроку 9, інакше – перехід до кроку 6.

Крок 6. Селекція. Відбір N_p батьківських пар для формування нащадків методом турнірної селекції з розміром турніру 2-3 особини.

Крок 7. Схрещування та мутація. Застосування оператора рівномірного схрещування до кожної батьківської пари для формування двох нащадків. Застосування оператора побітової мутації до кожного нащадку з ймовірністю p_m . Виконання процедури репарації для забезпечення допустимості отриманих хромосом.

Крок 8. Формування нової популяції та перехід до кроку 3. Нова популяція формується з нащадків з можливим збереженням кількох найкращих особин попереднього покоління (елітизм).

Крок 9. Завершення роботи алгоритму. Результатом є варіант розміщення терміналів w^o з мінімальним серед знайдених значенням наведених витрат $\Delta C(w^o)$.

Параметри генетичного алгоритму суттєво впливають на якість отриманих рішень та час роботи алгоритму. Розмір популяції N_p визначає баланс між різноманітністю пошуку та обчислювальними витратами. Для задач оптимізації логістичних мереж рекомендовані значення розміру популяції знаходяться в діапазоні від 50 до 200 особин [17]. Кількість поколінь i^* визначає тривалість еволюційного процесу та має обиратися з урахуванням розмірності задачі та доступних обчислювальних ресурсів.

Часова складність еволюційного методу на основі генетичного алгоритму визначається кількістю поколінь, розміром популяції та обчислювальними витратами на оцінку функції пристосованості кожної особини. Точність еволюційного методу характеризується відносною похибкою рішення порівняно з глобальним оптимумом. Вища похибка еволюційного методу порівняно з методом покоординатної оптимізації пояснюється стохастичним характером пошуку та можливістю передчасної збіжності до субоптимальних рішень. Точність методу може бути підвищена шляхом збільшення розміру популяції та кількості поколінь, а також застосування адаптивних механізмів налаштування параметрів.

Перевагами еволюційного методу на основі генетичного алгоритму є здатність досліджувати великі простори рішень, стійкість до багатоекстремальності цільової функції та можливість ефективного розпаралелювання обчислень. До недоліків методу належать необхідність налаштування параметрів алгоритму, стохастичний характер результатів та нижча точність порівняно з детермінованими локальними методами для задач помірної розмірності [18].

Висновки за розділом 2

У другому розділі було здійснено вибір та обґрунтування методів розв'язання задачі структурно-топологічної оптимізації логістичної мережі на

етапі реінжинірингу. Проведений огляд класичних, евристичних і метаевристичних підходів показав, що точні методи, зокрема метод гілок і меж, забезпечують знаходження глобального оптимуму, проте є практично непридатними для задач великої розмірності через експоненційне зростання обчислювальної складності. Евристичні та метаевристичні методи, навпаки, дозволяють отримувати якісні наближені розв'язки за прийнятний час, що є критично важливим для реальних логістичних систем.

На основі порівняльного аналізу було обґрунтовано доцільність використання методу покоординатної оптимізації та еволюційного методу на основі генетичного алгоритму. Метод покоординатної оптимізації характеризується простотою реалізації, детермінованістю та високою точністю для задач помірної розмірності, що робить його ефективним інструментом локального пошуку. Генетичний алгоритм, у свою чергу, забезпечує дослідження великих просторів рішень і стійкість до багатоекстремальності цільової функції, що дозволяє зменшити ризик застрягти в локальних екстремумах.

Таким чином, поєднання детермінованого локального методу та еволюційного підходу створює методичну основу для побудови ефективного алгоритму реінжинірингу топологічної структури логістичної мережі. Обрані методи забезпечують баланс між точністю розв'язання і обчислювальною ефективністю та можуть бути використані для подальшої програмної реалізації й проведення обчислювального експерименту, результати якого розглядаються у наступних розділах роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір інструментальних засобів та мови програмування

Розробка програмного комплексу для оптимізації топологічних структур логістичних мереж вимагає обґрунтованого підходу до вибору інструментальних засобів та мови програмування. Основними критеріями вибору є продуктивність обчислень, наявність спеціалізованих бібліотек для числових розрахунків та оптимізації, можливості візуалізації результатів, кросплатформеність та підтримка в науковій спільноті [19].

Для реалізації програмного комплексу обрано мову програмування Python версії 3.8 та вище. Python є однією з найпоширеніших мов для наукових обчислень та розробки алгоритмів оптимізації завдяки простоті синтаксису, читабельності коду та розвиненій екосистемі бібліотек. За даними опитування Stack Overflow 2024 року, Python посідає провідні позиції серед мов програмування для наукових досліджень та машинного навчання [20].

Ключовою перевагою Python для задач оптимізації логістичних мереж є наявність бібліотеки NumPy, яка забезпечує ефективні операції з масивами та матрицями [21]. Це критично важливо для обчислення відстаней між вузлами мережі, формування матриць суміжності та оцінки значень цільової функції. Бібліотека NumPy використовує векторизовані операції, реалізовані на рівні мов C та Fortran, що забезпечує швидкодію, наближену до компільованих мов програмування.

Для візуалізації результатів роботи алгоритмів оптимізації обрано бібліотеку Matplotlib версії 3.5 та вище. Matplotlib є стандартним інструментом для створення наукових графіків та діаграм у середовищі Python [22]. Бібліотека підтримує широкий спектр типів візуалізацій, зокрема точкові діаграми для відображення географічного розташування вузлів мережі та мережеві графи для візуалізації топології логістичної мережі до та після оптимізації.

Архітектура програмного комплексу реалізована з використанням об'єкт-

но-орієнтованої парадигми програмування. Для структурування коду застосовано патерни проектування Strategy та Template Method, що дозволяє розширювати систему новими алгоритмами оптимізації без модифікації базового коду. Зокрема, патерн Strategy використано для реалізації методу покоординатної оптимізації (МПО) та еволюційного методу на основі генетичного алгоритму (ЕМ-ГА), які можуть бути взаємозамінними завдяки спільному інтерфейсу.

Для роботи з вхідними даними використано вбудований модуль CSV(Comma-Separated Values) мови Python, який забезпечує надійне читання та запис даних у форматі CSV з підтримкою різних кодвань, включаючи UTF-8 . Це забезпечує інтеграцію з існуючими корпоративними системами, де дані про логістичні мережі зазвичай зберігаються у форматі електронних таблиць.

Вибір Python також обумовлений наявністю інструментів для паралелізації обчислень. Модуль multiprocessing дозволяє ефективно використовувати багатоядерні процесори для паралельного обчислення функції пристосованості (fitness-функції) в генетичному алгоритмі [11]. Це є важливим для мереж середньої та великої розмірності ($n > 30$), де час обчислень може бути суттєвим обмеженням.

Порівняння стеку мов програмування з альтернативними рішеннями наведено в таблиці 3.1.

Таблиця 3.1 – Порівняння стеку мов програмування

Критерій	Python	Java	C++	MATLAB
Швидкість розробки	Висока	Середня	Низька	Висока
Продуктивність обчислень	Середня	Висока	Дуже висока	Середня
Наявність бібліотек для оптимізації	Широка	Обмежена	Обмежена	Широка
Можливості візуалізації	Широкі	Обмежені	Обмежені	Широкі
Вартість ліцензії	Безкоштовно	Безкоштовно	Безкоштовно	Комерційна

Як видно з таблиці, Python забезпечує оптимальний баланс між швидкістю розробки, наявністю спеціалізованих бібліотек та можливостями візуалізації. Хоча мови C++ та Julia демонструють вищу продуктивність обчислень, вони поступаються Python за зручністю розробки та наявністю готових рішень для наукових обчислень.

Важливим фактором вибору Python є його широке використання в академічному середовищі. За даними IEEE Spectrum 2024, Python є найпопулярнішою мовою для наукових публікацій у галузі дослідження операцій та оптимізації [11]. Це забезпечує можливість відтворення результатів дослідження іншими науковцями та полегшує впровадження розробленого програмного комплексу в навчальний процес.

Таким чином, обраний стек технологій (Python 3.8+, NumPy, Matplotlib, CSV) забезпечує оптимальне співвідношення між продуктивністю розробки, ефективністю обчислень та можливостями візуалізації, що робить його обґрунтованим вибором для реалізації програмного комплексу оптимізації топологічних структур логістичних мереж на етапі реінжинірингу.

3.2 Архітектура програмного комплексу

Архітектура програмного комплексу оптимізації топологічних структур логістичних мереж побудована на принципах модульності, розширюваності та слабкої зв'язності компонентів. Для організації програмного коду обрано багатопарову архітектуру, яка забезпечує чітке розмежування відповідальності між функціональними блоками системи та спрощує процеси тестування, налагодження та супроводу програмного забезпечення.

Загальна структура програмного комплексу складається з п'яти основних шарів: шару моделі даних, що описує предметну область; шару сервісів, який реалізує бізнес-логіку системи; шару оптимізації, що містить алгоритми пошу-

ку оптимальних рішень; шару утиліт із допоміжними інструментами; та шару представлення, який забезпечує взаємодію з користувачем. Така архітектура відповідає принципам SOLID [24], зокрема принципу єдиної відповідальності (Single Responsibility Principle), згідно з яким кожен модуль має відповідати лише за одну функціональну область, та принципу відкритості/закритості (Open/Closed Principle), що передбачає можливість розширення функціональності без модифікації існуючого коду.

Шар моделі даних реалізує об'єктну модель предметної області логістичної мережі та забезпечує структуроване представлення всіх елементів системи. Центральним елементом цього шару є ієрархія класів, що відображає типологію елементів трирівневої централізованої логістичної мережі.

Базовий клас `Element` інкапсулює спільні властивості всіх елементів мережі, а саме координати розташування на площині (x, y) та унікальний ідентифікатор. Від базового класу успадковуються три спеціалізовані класи: `Center`, що представляє центральний розподільчий центр першого рівня мережі; `Terminal`, який моделює проміжний термінал другого рівня з параметрами вартості створення та експлуатації; `Consumer`, що описує споживача третього рівня з параметром обсягу попиту.

Використання механізму успадкування дозволяє застосувати поліморфізм при обробці елементів мережі, що суттєво спрощує програмний код та підвищує його читабельність. Зокрема, алгоритми обчислення відстаней та візуалізації можуть працювати з об'єктами базового типу `Element`, не залежачи від конкретного типу елемента мережі.

Клас `LogisticsNetwork` виконує роль агрегатора, що об'єднує всі елементи мережі в єдину структуру. Цей клас відповідає за підтримку цілісності даних мережі та надає уніфікований інтерфейс для роботи з нею. Основними методами класу є: `get_active_terminals()` для отримання списку активних терміналів поточної конфігурації; `calculate_total_cost()` для обчислення сумарної вартості функціонування мережі; `clone()` для створення глибокої копії мережі.

Метод `clone()` є особливо важливим для функціонування генетичного ал-

горитму, оскільки кожна особина популяції повинна мати незалежну копію мережі для уникнення побічних ефектів при виконанні операцій мутації та схрещування.

Шар сервісів містить модулі, що реалізують допоміжну бізнес-логіку системи та забезпечують виконання типових операцій над даними логістичної мережі. Модуль `DataLoader` відповідає за завантаження вхідних даних з файлів формату CSV. Модуль здійснює валідацію формату вхідних даних, перевірку наявності всіх необхідних полів та контроль коректності значень. Для забезпечення сумісності з різними джерелами даних підтримуються різні кодування символів, зокрема UTF-8 та UTF-8 з маркером порядку байтів (BOM) .

Модуль `DistanceCalculator` реалізує обчислення між елементами мережі. Для двох точок A та B відстань обчислюється за формулою:

$$d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2},$$

де A та B – точки з координатами (x_A, y_A) та (x_B, y_B) відповідно.

Для підвищення продуктивності обчислень використовуються векторизовані операції бібліотеки `NumPy`, що дозволяє ефективно обчислювати матрицю відстаней для всіх пар елементів мережі.

Модуль `CostCalculator` виконує розрахунок цільової функції вартості мережі відповідно до математичної моделі, представленої в розділі 2. Модуль враховує особливості задачі реінжинірингу, зокрема можливість використання існуючої інфраструктури та витрати на модернізацію або демонтаж терміналів.

Модуль `Visualization` забезпечує генерацію наукових візуалізацій результатів оптимізації з використанням бібліотеки `Matplotlib` . Модуль створює точкові діаграми (`scatter plot`) для відображення географічного розташування елементів мережі із зазначенням зв'язків між ними, а також стовпчикові діаграми (`bar chart`) для порівняння результатів роботи різних алгоритмів оптимізації.

Модуль `ResultExporter` відповідає за експорт результатів оптимізації у фо-

рмат CSV з підтримкою різних кодувань. Для забезпечення коректного відкриття файлів у програмі Microsoft Excel з локалізованими налаштуваннями використовується спеціальна директива роздільника полів.

Шар оптимізації є ключовим компонентом програмного комплексу та реалізує патерн проектування Strategy, що дозволяє динамічно обирати алгоритм оптимізації без зміни клієнтського коду. Така архітектура забезпечує гнучкість системи та можливість легкого додавання нових методів оптимізації.

Базовий абстрактний клас BaseOptimizer визначає спільний інтерфейс для всіх алгоритмів оптимізації та реалізує загальну функціональність, зокрема збереження історії процесу оптимізації та обчислення значень цільової функції. Клас містить абстрактний метод optimize(), який повинен бути реалізований у кожному конкретному оптимізаторі, та захищений метод evaluate() для оцінки якості поточної конфігурації мережі.

Клас CoordinateOptimizer реалізує метод покоординатної оптимізації (МПО), який виконує детерміністичний локальний пошук оптимального рішення. Алгоритм послідовно оптимізує розміщення кожного терміналу при фіксованих положеннях інших терміналів, циклічно повторюючи цю процедуру до досягнення локального екстремуму цільової функції. Основними параметрами налаштування є максимальна кількість ітерацій (max_iterations) та мінімальне покращення для продовження оптимізації (tolerance).

Клас GeneticOptimizer реалізує еволюційний метод на основі генетичного алгоритму (ЕМ-ГА), призначений для глобального пошуку оптимуму. Алгоритм моделює процес природного відбору, застосовуючи операції селекції, схрещування та мутації до популяції потенційних рішень. Параметрами налаштування алгоритму є: розмір популяції (population_size), що визначає кількість особин у кожному поколінні; кількість поколінь (generations), що обмежує тривалість еволюційного процесу; ймовірність мутації (mutation_rate), що контролює інтенсивність випадкових змін; ймовірність схрещування (crossover_rate), що визначає частоту обміну генетичним матеріалом між особинами.

Використання патерну Strategy дозволяє легко додавати нові алгоритми

оптимізації шляхом створення нових класів, що успадковують BaseOptimizer, без необхідності модифікації існуючого коду. Це забезпечує відповідність принципу відкритості/закритості та спрощує розширення функціональності системи.

Загальний потік даних у програмному комплексі організовано як послідовність чітко визначених етапів обробки. На першому етапі модуль DataLoader зчитує вхідні дані з файлу формату CSV та створює відповідні об'єкти моделі даних. На другому етапі формується об'єкт LogisticsNetwork, що агрегує всі елементи мережі та встановлює зв'язки між ними. Третій етап передбачає оцінку початкового стану мережі шляхом обчислення значення цільової функції за допомогою модуля CostCalculator.

Четвертий етап є основним і полягає у виконанні процедури оптимізації обраним алгоритмом (МПО або EM-ГА). На п'ятому етапі модуль Visualization створює графічне представлення результатів, що включає порівняння конфігурацій мережі до та після оптимізації. Завершальний шостий етап передбачає експорт результатів у файл формату CSV за допомогою модуля ResultExporter.

Така послідовність забезпечує лінійний потік даних без циклічних залежностей, що спрощує розуміння логіки роботи системи та полегшує процес налагодження.

Система містить багаторівневу підсистему валідації даних, що забезпечує коректність вхідної інформації на всіх етапах обробки. Перший рівень валідації здійснюється на етапі завантаження даних та включає перевірку формату файлу CSV, наявності всіх обов'язкових полів, коректності типів даних та відсутності пропущених значень. Другий рівень валідації перевіряє відповідність даних бізнес-правилам предметної області. Зокрема, контролюється наявність рівно одного центрального вузла, принаймні одного терміналу та одного споживача; невід'ємність координат елементів; коректність значень вартісних параметрів. Третій рівень валідації функціонує під час виконання процедури оптимізації та забезпечує контроль допустимості проміжних конфігурацій мережі. Для обробки помилкових ситуацій використовується механізм винятків (exceptions), що

відповідає рекомендаціям стандарту PEP 8 [25] та забезпечує коректне інформування користувача про причини помилок і можливі шляхи їх усунення.

Обрана архітектура програмного комплексу забезпечує можливість масштабування системи у кількох напрямках. Горизонтальне масштабування передбачає додавання нових алгоритмів оптимізації шляхом реалізації інтерфейсу BaseOptimizer без необхідності внесення змін до інших компонентів системи. Це дозволяє розширювати бібліотеку методів оптимізації відповідно до потреб конкретних практичних задач.

Вертикальне масштабування досягається за рахунок паралелізації обчислень у генетичному алгоритмі з використанням модуля multiprocessing стандартної бібліотеки Python. Паралельне обчислення функції пристосованості для особин популяції дозволяє суттєво скоротити час розв'язання задач великої розмірності на багатоядерних процесорах.

Функціональне масштабування передбачає можливість додавання нових типів візуалізацій, підтримки альтернативних форматів експорту даних (JSON, XML) та інтеграції з зовнішніми системами управління базами даних.

Модульна архітектура з чітко визначеними інтерфейсами між компонентами забезпечує можливість незалежного розвитку та модифікації окремих частин системи без впливу на функціонування інших компонентів [17]. Таким чином, розроблена архітектура програмного комплексу забезпечує модульність, розширюваність, тестованість та підтримуваність системи оптимізації топологічних структур логістичних мереж. Застосування перевірених патернів проектування та принципів об'єктно-орієнтованого програмування гарантує високу якість програмного забезпечення та можливість його подальшого розвитку [18].

3.3 Алгоритми основних модулів програми

Програмний комплекс оптимізації топологічних структур логістичних мереж реалізує комплекс алгоритмів, які забезпечують функціонування всіх пі-

дсистем. У даному розділі наведено математичні основи та особливості програмної реалізації ключових алгоритмів системи.

Цільова функція вартості логістичної мережі є центральним елементом задачі оптимізації та визначає критерій якості поточної конфігурації мережі. Відповідно до математичної моделі, представленої в розділі 2, загальна вартість мережі складається з трьох компонент:

$$C_{total} = C_{fixed} + C_{processing} + C_{transport},$$

де C_{fixed} – фіксовані витрати на утримання активних терміналів;

$C_{processing}$ – витрати на обробку вантажів у терміналах;

$C_{transport}$ – транспортні витрати на доставку вантажів між елементами мережі.

Фіксовані витрати обчислюються як сума вартостей усіх активних терміналів мережі:

$$C_{fixed} = \sum_{j \in T} f_j \cdot a_j,$$

де f_j – фіксована вартість j -го терміналу;

$a_j \in \{0,1\}$ – індикатор активності терміналу;

T – множина всіх терміналів мережі.

Витрати на обробку визначаються сумарним обсягом вантажів, що проходять через кожен активний термінал :

$$C_{processing} = \sum_{j \in T_{active}} p_j \cdot Q_j,$$

де p_j – вартість обробки одиниці вантажу в j -му терміналі;

Q_j – загальний обсяг вантажів, що обробляється в цьому терміналі;

T_{active} – множина активних терміналів.

Транспортні витрати включають дві складові: витрати на доставку від центру до терміналів та від терміналів до споживачів:

$$C_{transport} = \sum_j c \cdot d(C, T_j) \cdot Q_j + \sum_j \sum_i c \cdot d(T_j, S_i) \cdot q_i,$$

де c – вартість перевезення одиниці вантажу на одиницю відстані;

$d(A, B)$ – відстань між точками A та B ;

C – центр мережі;

T_j – j -й термінал;

S_i – i -й споживач;

q_i – обсяг попиту i -го споживача.

Алгоритм обчислення цільової функції реалізовано в модулі CostCalculator. На першому етапі ініціалізуються змінні для накопичення трьох компонент вартості. Далі для кожного активного терміналу виконується додавання його фіксованої вартості до відповідної компоненти, визначення множини призначених споживачів та обчислення сумарного обсягу вантажів. Для кожного споживача обчислюється відстань до терміналу та відповідні транспортні витрати. Завершальним кроком є додавання витрат на транспортування від центру до терміналу та обчислення загальної суми всіх компонент. Обчислювальна складність алгоритму становить $O(N \cdot C)$ де N – кількість терміналів, C – кількість споживачів [26].

Задача призначення споживачів до терміналів розв'язується за принципом мінімізації транспортних витрат: кожен споживач обслуговується найближчим активним терміналом. Такий підхід є оптимальним при фіксованій конфігурації терміналів та забезпечує мінімальні витрати на доставку вантажів.

Для підвищення продуктивності обчислень використано векторизовані

операції бібліотеки NumPy, що дозволяє ефективно обчислювати матрицю відстаней для всіх пар елементів мережі [23].

Алгоритм призначення працює наступним чином. Спочатку формується список активних терміналів мережі. Якщо множина активних терміналів порожня, алгоритм завершується з порожнім результатом. Для кожного споживача виконується пошук найближчого терміналу шляхом обчислення відстаней до всіх активних терміналів та вибору терміналу з мінімальною відстанню. Результатом роботи алгоритму є словник призначень, що зіставляє ідентифікатор кожного споживача з ідентифікатором відповідного терміналу.

Часова складність алгоритму становить $O(C \cdot N)$, де C – кількість споживачів, N – кількість активних терміналів, що є оптимальним для задачі пошуку найближчого сусіда в двовимірному просторі при невеликих розмірах мережі. Метод покоординатної оптимізації (МПО) є ітеративним алгоритмом локального пошуку, який послідовно оптимізує параметри кожного терміналу окремо при фіксованих параметрах інших терміналів. Алгоритм гарантує монотонне зменшення значення цільової функції на кожній ітерації, що забезпечує збіжність до локального оптимуму. Основна ідея методу полягає в пошуку оптимальних координат для кожного терміналу при фіксованих координатах решти терміналів:

$$(x_j^*, y_j^*) = \arg \min C_{total}(x_j, y_j | x_k, y_k \text{ для } k \neq j).$$

Оптимальні координати j -го терміналу визначаються як центр мас споживачів, призначених до цього терміналу, зважений за обсягами попиту [13]:

$$x_j^* = \frac{\sum_i q_i \cdot x_i}{\sum_i q_i}, \quad (3.1)$$

$$y_j^* = \frac{\sum_i q_i \cdot y_i}{\sum_i q_i}. \quad (3.2)$$

де обчислення суми виконується по всіх споживачах i , призначених до терміналу j .

Алгоритм МПО складається з наступних основних етапів. На етапі ініціалізації обчислюється початкова вартість мережі та ініціалізується лічильник ітерацій. Основний цикл оптимізації включає два вкладені цикли: цикл оптимізації координат терміналів та цикл перевірки доцільності терміналів.

У циклі оптимізації координат для кожного активного терміналу визначається множина призначених споживачів. Якщо термінал не має призначених споживачів, він деактивується. В іншому випадку обчислюється центр мас призначених споживачів, зважений за обсягами попиту, і координати терміналу оновлюються відповідно до формул (3.1) та (3.2).

У циклі перевірки доцільності для кожного активного терміналу порівнюється вартість мережі з даним терміналом та без нього. Якщо видалення терміналу призводить до зменшення загальної вартості, термінал деактивується.

Після завершення обох циклів виконується перепризначення споживачів до терміналів та обчислення нової вартості мережі. Алгоритм продовжує ітерації, поки відносне покращення перевищує задане порогове значення (tolerance) або не досягнуто максимальної кількості ітерацій.

Умова збіжності алгоритму визначається співвідношенням:

$$improvement = \frac{C_{current} - C_{new}}{C_{current}} \times 100\% .$$

Якщо значення Improvement менше заданого порогу Tolerance, алгоритм завершується з поточним розв'язком. Часова складність однієї ітерації МПО становить $O(N^2 \cdot C)$, а загальна складність – $O(I \cdot N^2 \cdot C)$, де I – кількість ітерацій.

Еволюційний метод на основі генетичного алгоритму (ЕМ-ГА) реалізує стохастичний глобальний пошук оптимального рішення шляхом моделювання процесу природного відбору. На відміну від МПО, який знаходить лише лока-

льний оптимум, ЕМ-ГА здатний досліджувати значно більший простір рішень та знаходити глобально оптимальні або близькі до них конфігурації мережі. Кожне потенційне рішення задачі кодується у вигляді хромосоми – бінарного вектора довжиною N , де N – кількість терміналів мережі. Значення гена визначає стан терміналу: активний або неактивний. Таке представлення природно відповідає структурі задачі та дозволяє ефективно застосовувати стандартні генетичні оператори.

Алгоритм ЕМ-ГА складається з наступних основних етапів. На етапі ініціалізації генерується початкова популяція хромосом розміром `Population_Size`. Кожна хромосома ініціалізується випадковим чином з рівномірним розподілом значень генів. Для кожної хромосоми обчислюється значення функції пристосованості та визначається найкраща особина початкової популяції.

Основний цикл еволюції виконується протягом заданої кількості поколінь (`Generations`). На кожній ітерації формується нова популяція шляхом застосування генетичних операторів до особин поточної популяції. Оператор селекції реалізовано методом турнірного відбору з розміром турніру 3. З популяції випадковим чином обираються три особини, і переможцем турніру стає особина з найвищим значенням функції пристосованості. Турнірний відбір забезпечує баланс між селективним тиском та збереженням генетичного різноманіття популяції.

Оператор схрещування реалізовано методом одноточкового кросоверу. Для двох батьківських хромосом випадковим чином обирається точка розрізу, і нащадки формуються шляхом обміну сегментами хромосом після точки розрізу. Схрещування виконується з ймовірністю `Crossover_Rate`. Оператор мутації реалізовано методом побітової інверсії. Для хромосоми випадковим чином обирається позиція гена, і значення цього гена інвертується (0 змінюється на 1 і навпаки). Мутація виконується з ймовірністю `Mutation_Rate` та забезпечує дослідження нових областей простору рішень. Після застосування генетичних операторів виконується перевірка валідності нащадків: хромосома повинна містити принаймні один активний термінал. Якщо всі гени хромосоми дорівнюють нулю, один випадковий ген встановлюється в одиницю.

Елітизм забезпечується шляхом збереження найкращої знайденої хромосоми протягом усього процесу еволюції. Після завершення всіх поколінь найкраща хромосома застосовується до мережі для отримання оптимізованої конфігурації.

Часова складність ЕМ-ГА становить $O(G \cdot P \cdot N \cdot C)$, де G – кількість поколінь, P – розмір популяції, N – кількість терміналів, C – кількість споживачів. Основна обчислювальна складність припадає на обчислення функції пристосованості для кожної особини популяції на кожному поколінні.

Порівняльний аналіз часової та просторової складності основних алгоритмів системи наведено в таблиці 3.2.

Таблиця 3.2 – Порівняльний аналіз обчислювальної складності алгоритмів

Алгоритм	Часова складність	Просторова складність
Обчислення цільової функції	$O(N \cdot C)$	$O(1)$
Призначення споживачів	$O(N \cdot C)$	$O(C)$
МПО (одна ітерація)	$O(N^2 \cdot C)$	$O(N + C)$
МПО (загалом)	$O(I \cdot N^2 \cdot C)$	$O(N + C)$
ЕМ-ГА (одне покоління)	$O(P \cdot N^2 \cdot C)$	$O(P \cdot N)$
ЕМ-ГА (загалом)	$O(G \cdot P \cdot N \cdot C)$	$O(P \cdot N)$

У табл. 3.2 N позначає кількість терміналів, C – кількість споживачів, I – кількість ітерацій МПО, G – кількість поколінь ЕМ-ГА, P – розмір популяції.

Для типових параметрів задачі ($N = 10$, $C = 30$, $I = 3$, $G = 50$, $P = 100$) кількість базових операцій становить: для МПО – близько 9 000 операцій; для ЕМ-ГА – близько 1 500 000 операцій. Незважаючи на суттєво більшу обчислювальну складність, ЕМ-ГА часто знаходить кращі рішення завдяки можливості глобального пошуку в просторі рішень.

3.4 Опис інтерфейсу користувача та функціональних можливостей

Інтерфейс користувача програмного комплексу реалізовано у вигляді інтерактивного консольного додатку (Command-Line Interface), який забезпечує ефективний спосіб взаємодії з системою оптимізації логістичних мереж. Вибір консольного інтерфейсу обумовлений кількома факторами: кросплатформеністю, що забезпечує однакову функціональність на різних операційних системах; можливістю автоматизації через командні скрипти; відсутністю залежностей від графічних бібліотек, що спрощує розгортання програмного комплексу.

При запуску програми користувачу відображається головне меню з переліком доступних файлів вхідних даних (рис. 3.1). Програма автоматично сканує директорію data/ та виявляє всі наявні файли формату CSV, що містять опис логістичних мереж.

```
=====
ПРОГРАМА ОПТИМІЗАЦІЇ ЛОГІСТИЧНОЇ МЕРЕЖІ
=====

ДОСТУПНІ ФАЙЛИ ДАНИХ
=====
1. clustered_consumers.csv
2. ga_advantage_network.csv
3. local_minimum_trap.csv
4. network_data.csv
5. network_n15.csv
6. network_n20.csv
7. network_n25.csv
8. network_n30.csv
9. network_n35.csv
10. network_n40.csv
11. optimal_initial.csv
12. small_network.csv
13. sparse_network.csv
14. too_many_terminals.csv
15. Вихід
=====
Оберіть файл (1-15): |
```

Рисунок 3.1 – Меню з переліком доступних файлів

Після вибору файлу даних система перевіряє файл на відповідність, та виводить інформацію поточний стан мережі (рис. 3.2).

```

✓ Обрано файл: network_n35.csv

=====
ПРОГРАМА ОПТИМІЗАЦІЇ ЛОГІСТИЧНОЇ МЕРЕЖІ
=====

[1/3] Завантаження даних з CSV...
✓ Дані успішно завантажено

[2/3] Валідація даних...
✓ Дані валідні

[3/3] Створення логістичної мережі...
✓ Мережа створена та ініціалізована

=== ЗАВАНТАЖЕНА МЕРЕЖА ===
Розподільчих центрів: 1
Терміналів: 11
Споживачів: 35
Загальний попит: 2731.00

Центри:
  Center(id=0, x=50.0, y=50.0, type='center')

Термінали:
  Terminal(id=10, x=23.8, y=21.1, cost=8000.0, active)
  Terminal(id=11, x=25.4, y=25.8, cost=8000.0, active)
  Terminal(id=12, x=22.5, y=24.0, cost=8000.0, active)
  Terminal(id=13, x=78.5, y=14.8, cost=8000.0, active)
  Terminal(id=14, x=85.9, y=25.0, cost=8000.0, active)
  Terminal(id=15, x=19.0, y=77.3, cost=8000.0, active)
  Terminal(id=16, x=24.1, y=80.9, cost=8000.0, active)
  Terminal(id=17, x=75.7, y=78.5, cost=8000.0, active)
  Terminal(id=18, x=84.0, y=81.1, cost=8000.0, active)
  Terminal(id=19, x=57.2, y=46.8, cost=8000.0, active)
  Terminal(id=20, x=54.0, y=51.0, cost=8000.0, active)

Споживачі (перші 5):
  Consumer(id=100, x=20.9, y=14.7, demand=72.0)
  Consumer(id=101, x=25.5, y=12.3, demand=76.0)
  Consumer(id=102, x=31.4, y=19.8, demand=65.0)
  Consumer(id=103, x=30.3, y=13.7, demand=61.0)
  Consumer(id=104, x=26.3, y=25.0, demand=97.0)
  ... та ще 30
=====

```

Рисунок 3.2 – Інформація після завантаження мережі

Далі пропонує обрати режим оптимізації з трьох доступних варіантів (рис. 3.3). Перший режим застосовує лише метод покоординатної оптимізації (МПО), що є доцільним для мереж малої та середньої розмірності, коли важлива детермінованість результату та швидкість обчислень. Другий режим використовує еволюційний метод на основі генетичного алгоритму (ЕМ-ГА), який рекомендується для мереж великої розмірності та складних топологій, де локальні методи можуть не забезпечувати глобально оптимальних рішень. Третій режим виконує обидва методи послідовно та формує детальне порівняння результатів, що є особливо корисним для наукових досліджень та валідації отриманих рішень.

```

=====
ВИБЕРІТЬ РЕЖИМ ОПТИМІЗАЦІЇ
=====
1. Тільки МПО (Метод покоординатного спуску)
2. Тільки ЕМ-ГА (Еволюційний метод - генетичний алгоритм)
3. Порівняння обох методів
4. Вихід
=====
Оберіть режим (1-4):

```

Рисунок 3.3 – Перелік режимів оптимізації

Під час виконання оптимізації система забезпечує інформативний зворотний зв'язок, відображаючи прогрес роботи алгоритму (рис. 3.4).

```

=====
ОПТИМІЗАЦІЯ МЕТОДОМ МПО
=====
Початкові витрати: 153,647.75
Параметри: макс_проходів=100, tolerance=0.1
=====

Можливих локацій для терміналів: 226
Активних терміналів: 11
-----

Прохід 1:
Термінал 10: переміщено на (27, 17), покращення 0.69%
Термінал 11: переміщено на (77, 22), покращення 0.62%
Термінал 12: переміщено на (22, 27), покращення 0.30%
Термінал 13: переміщено на (32, 22), покращення 0.45%
Термінал 14: переміщено на (87, 27), покращення 0.06%
Термінал 15: переміщено на (22, 72), покращення 0.15%
Термінал 16: переміщено на (27, 82), покращення 0.07%
Термінал 17: переміщено на (72, 82), покращення 0.15%
Термінал 18: переміщено на (82, 77), покращення 0.06%
Термінал 19: переміщено на (57, 42), покращення 0.02%
Термінал 20: переміщено на (52, 57), покращення 0.37%
→ Покращення на проході: 4458.89
→ Загальне покращення: 2.90%

Прохід 2:
Термінал 12: переміщено на (22, 32), покращення 0.10%
→ Покращення на проході: 149.20
→ Загальне покращення: 3.00%

Прохід 3:
→ Покращення на проході: 0.00
→ Загальне покращення: 3.00%

Збіжність досягнута: покращення 0.00 < tolerance 0.1
=====
Фаза 2: Перевірка доцільності терміналів
-----
Термінал 10 вимкнено, покращення: 4.469%
Термінал 11 вимкнено, покращення: 3.685%
Термінал 12 вимкнено, покращення: 3.953%
Термінал 15 вимкнено, покращення: 4.529%
Термінал 17 вимкнено, покращення: 5.583%
Термінал 19 вимкнено, покращення: 4.929%
→ Загальне покращення: 26.54%
Всі активні термінали необхідні
=====

Оптимізація завершена:
- Фаза 1 (оптимізація позицій): 3 проходів
- Фаза 2 (деактивація): 2 перевірок
=====

```

Рисунок 3.4 – Вивід інформації під час оптимізація методом МПО

Для методу покоординатної оптимізації передбачено виведення детальної інформації про хід кожної ітерації обчислювального процесу (рис. 3.5), яка

включає поточну кількість активних терміналів у структурі логістичної мережі, відповідне значення цільової функції вартості, а також відсоток покращення результату порівняно з попередньою ітерацією. Такий підхід дозволяє здійснювати покроковий аналіз збіжності алгоритму та оцінювати ефективність кожного етапу оптимізації. Додатково відображається інформація про кількість деактивованих терміналів на кожному кроці, що надає можливість користувачу відстежувати динаміку спрощення топологічної структури мережі та аналізувати вплив окремих рішень на загальну конфігурацію логістичної системи.

```

=====
РЕЗУЛЬТАТИ ОПТИМІЗАЦІЇ
=====

Початкові витрати:      153,647.75
Фінальні витрати:       112,873.85

-----

Покращення:             40,773.90
Покращення (%):         26.54%
=====

ГЕНЕРАЦІЯ ГРАФІКІВ
=====

📄 Графік збережено: results/network_n35_mpo_network_comparison.png
📄 Графік витрат збережено: results/network_n35_mpo_cost_comparison.png

=====
ЕКСПОРТ РЕЗУЛЬТАТІВ
=====
✓ Результати МПО експортовано: results/exports\network_n35_МПО_20251221_185029.csv

=====
ПРОГРАМУ ЗАВЕРШЕНО
=====

✓ Результати збережено:
- results/network_n35_mpo_network_comparison.png
- results/network_n35_mpo_cost_comparison.png
- results/exports\network_n35_МПО_20251221_185029.csv
=====

```

Рисунок 3.5 – Результат оптимізації методом МПО

Для генетичного алгоритму відображається прогрес еволюції по поколіннях із зазначенням найкращого знайденого значення цільової функції (рис. 3.6). Виведення інформації здійснюється з певним інтервалом (кожне десяте покоління), що забезпечує баланс між інформативністю та читабельністю виводу.

```

=====
ЗАПУСК ОПТИМІЗАЦІЇ: ЕМ-ГА
=====
ОПТИМІЗАЦІЯ ГЕНЕТИЧНИМ АЛГОРИТМОМ
=====
Початкові витрати: 429,215.16
Параметри: популяція=100, покоління=150
Мутація=0.15, кросовер=0.8
=====

Покоління 2: Витрати=243,553.55, Покращення=43.26% (Δ 185,661.60)
Покоління 3: Витрати=241,903.24, Покращення=43.64% (Δ 1,650.31)
Покоління 21: Витрати=240,811.87, Покращення=43.89% (Δ 1,091.37)

⚠️Early stopping: 30 поколінь без покращення
Зупинка на поколінні 51 з 150

=====
РЕЗУЛЬТАТИ ГЕНЕТИЧНОГО АЛГОРИТМУ
=====

Фінальна конфігурація терміналів:
Термінал 10: X НЕАКТИВНИЙ
Термінал 11: X НЕАКТИВНИЙ
Термінал 12: X НЕАКТИВНИЙ
Термінал 13: X НЕАКТИВНИЙ
Термінал 14: X НЕАКТИВНИЙ
Термінал 15: X НЕАКТИВНИЙ
Термінал 16: ✓ АКТИВНИЙ
Термінал 17: X НЕАКТИВНИЙ
Термінал 18: ✓ АКТИВНИЙ
Термінал 19: X НЕАКТИВНИЙ
Термінал 20: X НЕАКТИВНИЙ
Термінал 21: X НЕАКТИВНИЙ
Термінал 22: X НЕАКТИВНИЙ
Термінал 23: X НЕАКТИВНИЙ
Термінал 24: X НЕАКТИВНИЙ
Термінал 25: X НЕАКТИВНИЙ
Термінал 26: X НЕАКТИВНИЙ
Термінал 27: X НЕАКТИВНИЙ
Термінал 28: ✓ АКТИВНИЙ
Термінал 29: X НЕАКТИВНИЙ
Термінал 30: ✓ АКТИВНИЙ
Термінал 31: ✓ АКТИВНИЙ
Термінал 32: X НЕАКТИВНИЙ
Термінал 33: X НЕАКТИВНИЙ
Термінал 34: X НЕАКТИВНИЙ
Термінал 35: X НЕАКТИВНИЙ
Термінал 36: X НЕАКТИВНИЙ
Термінал 37: X НЕАКТИВНИЙ
Термінал 38: X НЕАКТИВНИЙ
Термінал 39: X НЕАКТИВНИЙ
Термінал 40: X НЕАКТИВНИЙ
Термінал 41: X НЕАКТИВНИЙ
Термінал 42: ✓ АКТИВНИЙ

Час виконання: 7.51 секунд
=====

```

Рисунок 3.6 – Вивід інформації під час оптимізація методом ЕМ-ГА

Після завершення оптимізації система формує детальний звіт про отримані результати. Для режиму одиночного методу звіт містить початкове та кінцеве значення цільової функції вартості, абсолютну та відносну величину досягнутої економії, кількість активних терміналів до та після оптимізації, час виконання та кількість ітерацій алгоритму (рис. 3.7).

```

=====
РЕЗУЛЬТАТИ ОПТИМІЗАЦІЇ
=====

Початкові витрати:      429,215.16
Фінальні витрати:       240,811.87

-----

Покращення:             188,403.28
Покращення (%):         43.89%
=====

ГЕНЕРАЦІЯ ГРАФІКІВ
=====

📄 Графік збережено: results/network_n100_ga_network_comparison.png
📄 Графік витрат збережено: results/network_n100_ga_cost_comparison.png

=====
ЕКСПОРТ РЕЗУЛЬТАТІВ
=====
✓ Результати ЕМ-ГА експортовано: results/exports/network_n100_ЕМ-ГА_20251221_202019.csv

=====
ПРОГРАМУ ЗАВЕРШЕНО
=====

✓ Результати збережено:
- results/network_n100_ga_network_comparison.png
- results/network_n100_ga_cost_comparison.png
- results/exports/network_n100_ЕМ-ГА_20251221_202019.csv
=====

```

Рисунок 3.7 – Результат оптимізація методом ЕМ-ГА

Для режиму порівняння методів формується розширений звіт у табличному форматі, що забезпечує наочне зіставлення результатів МПО та ЕМ-ГА (рис. 3.8).

```

=====
ПОРІВНЯЛЬНА ТАБЛИЦЯ РЕЗУЛЬТАТІВ
=====

Показник                МПО                ЕМ-ГА
-----
Початкові витрати (грн)  171,679.96        171,679.96
Фінальні витрати (грн)   116,653.28        113,269.17
Абсолютне покращення (грн)  55,026.68        58,410.79
Відносне покращення (%)   32.05             34.02
Активних терміналів після оптимізації  5                 5
Час виконання (сек)       5.82              1.56
=====

ВИСНОВОК
=====

🏆 Кращий результат показав метод: ЕМ-ГА

ЕМ-ГА досяг на 3,384.11 грн нижчих витрат, ніж МПО
(113,269.17 грн проти 116,653.28 грн)

Додаткові спостереження:
• ЕМ-ГА працював швидше на 4.26 сек
=====

```

Рисунок 3.8 – Результат оптимізація через порівняння методів

Таблиця містить значення всіх ключових показників для обох методів із

зазначенням переможця за кожним критерієм. Додатково обчислюється відносна різниця результатів δC , що визначає відсоткову перевагу одного методу над іншим. Негативне значення δC для ЕМ-ГА свідчить про те, що генетичний алгоритм знайшов рішення з меншою вартістю порівняно з методом покоординатної оптимізації. Звіт супроводжується текстовою інтерпретацією результатів, що допомагає користувачу правильно розуміти отримані дані та приймати обґрунтовані рішення щодо вибору методу оптимізації для конкретних практичних задач.

Система автоматично формує набір графічних матеріалів, призначених для візуального аналізу результатів оптимізації, з використанням бібліотеки Matplotlib. Згенеровані графіки відображають ключові характеристики роботи алгоритмів і дозволяють наочно оцінити отримані результати. Усі графічні матеріали зберігаються у форматі PNG з роздільною здатністю 300 DPI, що забезпечує достатній рівень якості для подальшого використання в наукових публікаціях, звітах та презентаційних матеріалах.

Візуалізація топології мережі до оптимізації відображає просторове розташування всіх елементів логістичної системи на координатній площині. Центральний розподільчий центр позначається синім ромбом, активні термінали – червоними трикутниками, а споживачі – зеленими точками, розмір яких пропорційний обсягу попиту. Лінії зв'язків між елементами мережі подаються у вигляді пунктирних сірих ліній, що дозволяє наочно оцінити загальну структуру мережі, характер просторових взаємозв'язків між її елементами та довжину транспортних маршрутів.

Візуалізація мережі після оптимізації формується з використанням тих самих графічних позначень і масштабу, що забезпечує коректне візуальне порівняння початкової та оптимізованої конфігурацій (рис. 3.9). При цьому деактивовані термінали на графіку не відображаються, що дозволяє наочно продемонструвати зменшення кількості активних елементів і спрощення топологічної структури логістичної мережі.

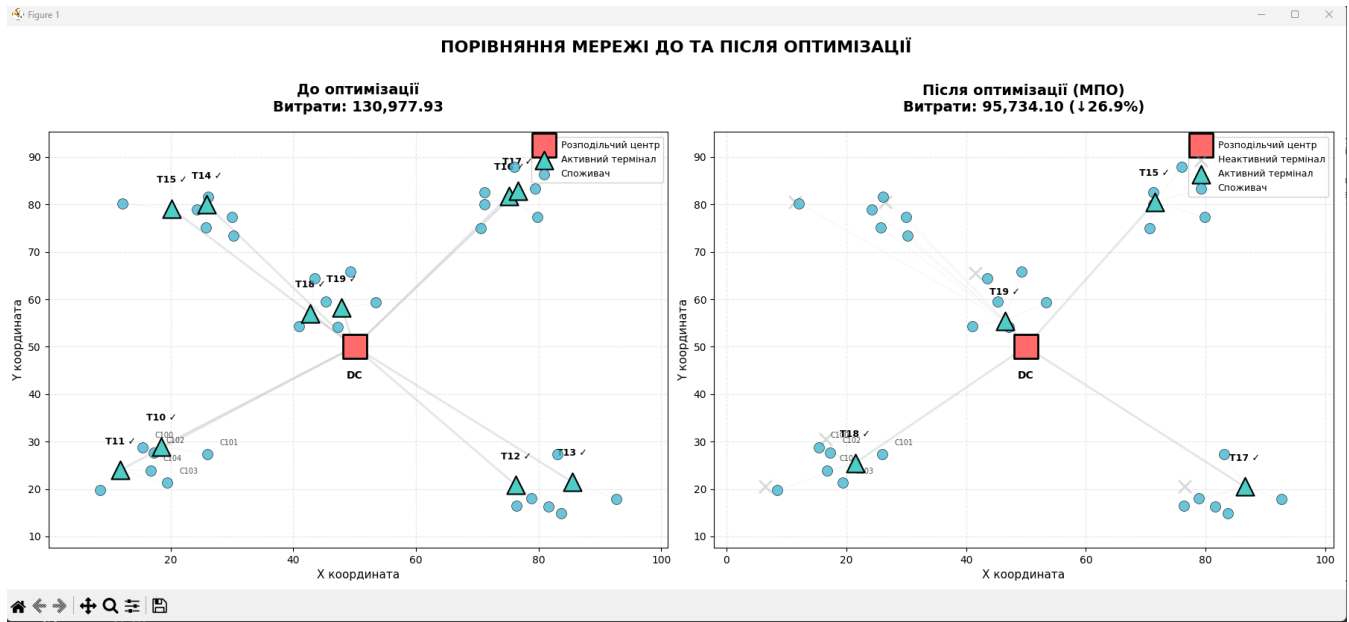


Рисунок 3.9 – Візуалізація топології мережі до та після оптимізації

Діаграма порівняння витрат відображає три компоненти вартості (фіксовані витрати, витрати на обробку, транспортні витрати) до та після оптимізації у вигляді стовпчикової діаграми з накопиченням (рис. 3.10). Така візуалізація дозволяє оцінити внесок кожної компоненти в загальну вартість та зрозуміти, за рахунок яких факторів досягнуто економію.

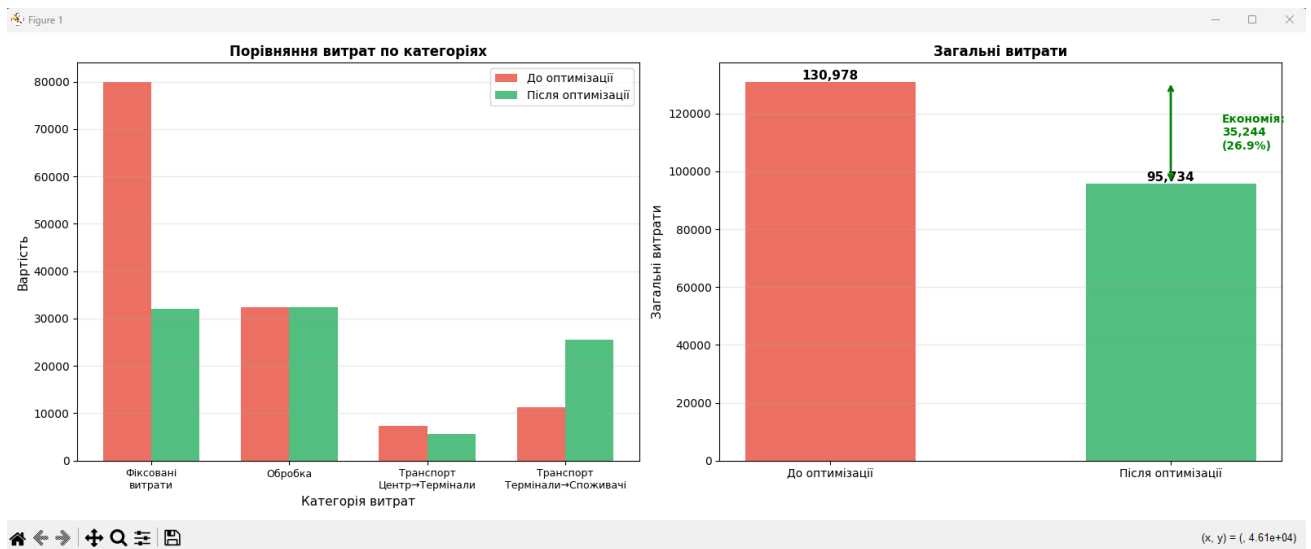


Рисунок 3.10 – Діаграма порівняння витрат до та після оптимізації

Для режиму порівняння методів додатково створюється графік (рис. 3.11),

що відображає результати обох алгоритмів поруч, забезпечуючи можливість прямого візуального зіставлення їх ефективності.

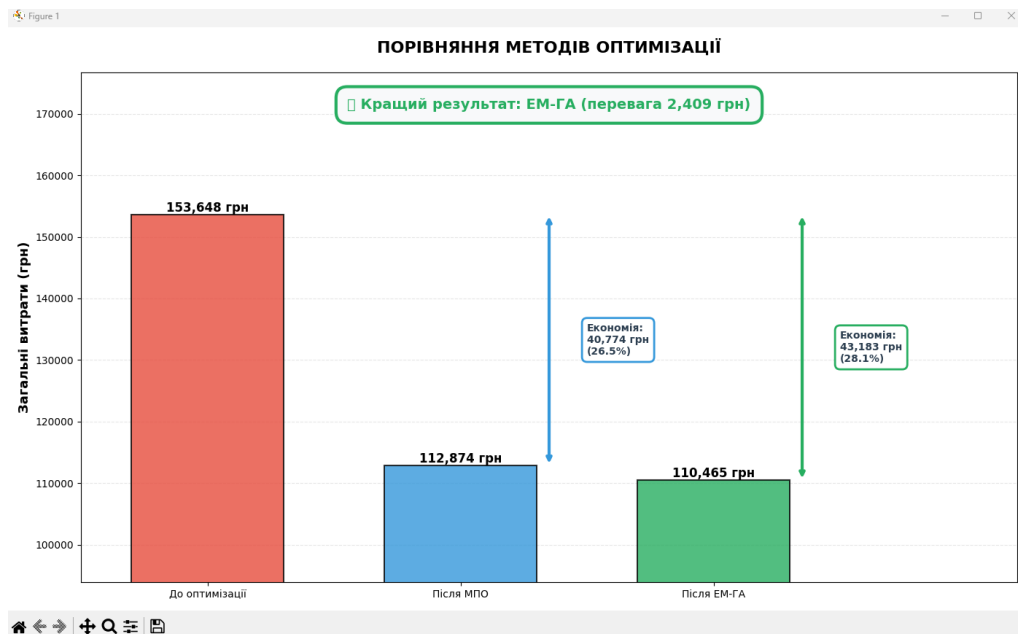


Рисунок 3.11 – Порівняння витрат до та після оптимізації

Усі результати оптимізації автоматично експортуються у формат CSV для подальшого аналізу в програмах електронних таблиць та статистичних пакетах (рис. 3.12). Файл результатів містить наступні поля: часова мітка виконання, назва набору даних, назва методу оптимізації, початкова та кінцева вартість мережі, абсолютне та відносне покращення, кількість терміналів до та після оптимізації, час виконання та кількість ітерацій.

	A	B	C	D	E	F	G	H	I	J	K
1	timestamp	dataset	method	initial_cost	final_cost	improvement_abs	improvement_pct	terminals_before	terminals_after	execution_time	iterations
2	12/21/2025 20:24	network_n40	MPO	171679.96	116653.28	55026.68	32.05	13	5	5.82	3
3	12/21/2025 20:24	network_n40	GA	171679.96	113269.17	58410.79	34.02	13	5	1.56	150
4											
5											

Рисунок 3.12 – Відображення результатів оптимізації у форматі CSV

Формат CSV забезпечує універсальну сумісність та дозволяє імпортувати дані в статистичні пакети (R, SPSS, Stata) та середовища програмування (Python pandas, MATLAB) для подальшого статистичного аналізу та побудови узагаль-

нених залежностей.

Програмний комплекс включає допоміжні модулі для автоматизації процесів тестування та генерації тестових даних. Генератор тестових мереж створює синтетичні логістичні мережі з параметризованими характеристиками. Модуль генерує набір мереж різних розмірів із випадковим розміщенням елементів на площині заданого розміру. Кількість терміналів визначається пропорційно до кількості споживачів, що забезпечує реалістичне співвідношення між рівнями мережі. Модуль пакетного тестування виконує автоматизоване порівняння методів оптимізації на всіх доступних мережах (рис. 3.13).

```

=====
ТЕСТУВАННЯ МЕРЕЖІ n=15
=====
Завантаження: C:\Users\zabsk\OneDrive\Рабочий стол\ДИПЛОМ\logistics_optimizer\data\network_n15.csv
Початкова вартість: 66486.31

--- МЕТОД ПАРАЛЕЛЬНОЇ ОПТИМІЗАЦІЇ (МПО) ---
Кінцева вартість: 52600.48
Покращення: 20.89%
Час виконання: 0.18 с
Ітерацій: 2

--- ЕВОЛЮЦІЙНО-МОДИФІКОВАНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ (ЕМ-ГА) ---
Кінцева вартість: 51697.71
Покращення: 22.24%
Час виконання: 0.24 с
Покоління: 50
Відносна похибка до МПО: -1.72%

=====
ТЕСТУВАННЯ МЕРЕЖІ n=20
=====
Завантаження: C:\Users\zabsk\OneDrive\Рабочий стол\ДИПЛОМ\logistics_optimizer\data\network_n20.csv
Початкова вартість: 82346.19

--- МЕТОД ПАРАЛЕЛЬНОЇ ОПТИМІЗАЦІЇ (МПО) ---
Кінцева вартість: 62962.91
Покращення: 23.54%
Час виконання: 0.28 с
Ітерацій: 2

--- ЕВОЛЮЦІЙНО-МОДИФІКОВАНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ (ЕМ-ГА) ---
Кінцева вартість: 60937.78
Покращення: 26.00%
Час виконання: 0.39 с
Покоління: 50
Відносна похибка до МПО: -3.22%

```

Рисунок 3.13 – Вивід інформації під час пакетного тестування

Результати тестування зберігаються у структурованому форматі CSV та включають усі ключові показники ефективності для кожної комбінації методу та набору даних (рис. 3.14). Ця функціональність є особливо корисною для проведення обчислювальних експериментів та статистичного аналізу ефективності алгоритмів [26].

	A	B	C	D	E	F	G	H	I	J	K	L
1	n	Initial Cost	MPO Cost	GA Cost	MPO Improvement %	GA Improvement %	MPO Error %	GA Error %	MPO Time (s)	GA Time (s)	MPO Iterations	GA Generations
2	15	66486.31	52600.48	51697.71	20.89	22.24	0	-1.72	0.18	0.24	2	50
3	20	82346.19	62962.91	60937.78	23.54	26	0	-3.22	0.28	0.39	2	50
4	25	106864.7	78998.79	77872.85	26.08	27.13	0	-1.43	0.76	0.78	2	50
5	30	130977.9	95734.1	93225.69	26.91	28.82	0	-2.62	2.45	0.81	3	50
6	35	153647.8	112873.9	110464.5	26.54	28.11	0	-2.13	3.54	2.33	3	50
7	40	171680	116653.3	113269.2	32.05	34.02	0	-2.9	13.69	3.51	3	50
8												
9												

Рисунок 3.14 – Результат пакетного тестування

Система реалізує багатопарову валідацію користувацького введення та інформативну обробку помилкових ситуацій. Валідація вибору пунктів меню перевіряє коректність введеного значення та його належність до допустимого діапазону. У разі некоректного введення користувачу виводиться зрозуміле повідомлення про помилку з вказівкою очікуваного формату даних.

Обробка помилок читання файлів включає перевірку існування файлу, коректності його формату та наявності всіх обов'язкових полів. У разі виявлення проблем система виводить детальну діагностичну інформацію, що включає опис очікуваного формату CSV та рекомендації щодо усунення помилок. Валідація структури завантаженої мережі перевіряє відповідність даних бізнес-правилам предметної області, зокрема наявність рівно одного центрального вузла, принаймні одного терміналу та одного споживача. Результати валідації виводяться у структурованому форматі із зазначенням статусу кожної перевірки. Програмний комплекс забезпечує однаково функціональність на всіх основних операційних системах (Windows, Linux, macOS) завдяки використанню кросплатформеної мови Python та стандартних бібліотек. Застосування модуля `os.path` для роботи з файловими шляхами забезпечує коректну роботу незалежно від формату шляхів, специфічного для кожної операційної системи.

Мінімальні вимоги до апаратного забезпечення включають процесор з тактовою частотою 1 ГГц, 256 МБ оперативної пам'яті та 100 МБ вільного дискового простору. Для обробки мереж великої розмірності ($n > 100$ споживачів) рекомендується використовувати багатоядерні процесори, що дозволяє ефектніше використовувати можливості паралелізації генетичного алгоритму. Таким чином, інтерфейс користувача програмного комплексу забезпечує інтуїти-

вну взаємодію, детальну візуалізацію результатів, гнучкі можливості експорту даних та надійну обробку помилок, що робить систему придатною як для проведення наукових досліджень, так і для практичного застосування в задачах оптимізації логістичних мереж.

Висновки за розділом 3

У третьому розділі виконано програмну реалізацію розроблених у попередніх розділах методів оптимізації топологічної структури логістичних мереж. Обґрунтовано вибір інструментальних засобів і мови програмування Python, який забезпечує оптимальне поєднання швидкості розробки, достатньої продуктивності обчислень, наявності спеціалізованих бібліотек для числових розрахунків та широких можливостей візуалізації результатів.

Розроблено модульну багат шарову архітектуру програмного комплексу, що базується на принципах об'єктно-орієнтованого програмування та патернах проєктування Strategy і Template Method. Такий підхід забезпечує слабку зв'язаність компонентів, розширюваність системи та можливість додавання нових алгоритмів оптимізації без модифікації існуючого коду. Реалізовано ключові модулі системи, зокрема завантаження та валідації даних, обчислення відстаней і цільової функції, оптимізаційні алгоритми, візуалізацію та експорт результатів.

У межах програмного комплексу реалізовано метод покоординатної оптимізації та еволюційний метод на основі генетичного алгоритму, що дозволяє виконувати як детермінований локальний пошук, так і глобальний стохастичний пошук оптимальних рішень. Наведено алгоритми основних модулів програми, оцінено їх часову та просторову складність, що підтверджує можливість застосування розробленої системи для мереж різної розмірності.

Створений консольний інтерфейс користувача забезпечує зручну взаємодію з програмним комплексом, підтримує декілька режимів оптимізації, дета-

льний вивід проміжних і підсумкових результатів, автоматичну генерацію графічних матеріалів та експорт даних у формат CSV. Реалізовані засоби валідації даних і обробки помилок підвищують надійність та практичну придатність системи.

Таким чином, у розділі 3 розроблено повнофункціональний програмний комплекс для оптимізації топологічних структур логістичних мереж, який відповідає поставленим вимогам, є гнучким, масштабованим і придатним як для проведення обчислювальних експериментів, так і для практичного використання в задачах реінжинірингу логістичних систем.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

4.1 Методика проведення обчислювального експерименту

Для перевірки ефективності розроблених методів оптимізації логістичних мереж виконано серію обчислювальних експериментів на тестових задачах різної розмірності. Метою експериментального дослідження є порівняльний аналіз методу покоординатного спуску та еволюційно-модифікованого генетичного алгоритму за двома групами критеріїв: якістю отриманих розв'язків і часовими витратами обчислень.

Експерименти проводилися на персональному комп'ютері з процесором Intel Core i5-8250U (4 ядра; 1.6–3.4 ГГц), оперативною пам'яттю 8 ГБ DDR4-2400 під керуванням Windows 11. Реалізація та запуск алгоритмів здійснювалися у середовищі Python 3.8.10 із використанням бібліотек NumPy 1.19.5 та Matplotlib 3.3.4. Для мінімізації впливу фонових процесів під час вимірювань продуктивності закривалися сторонні додатки, що могли спотворювати часові оцінки.

Тестові задачі формувалися шляхом генерації набору логістичних мереж у двовимірному просторі на площині 100×100 умовних одиниць. Кожна мережа містила один центральний розподільчий центр, розміщений у точці (50, 50), а координати терміналів і споживачів задавалися випадково з рівномірним розподілом у межах досліджуваної області. Кількість терміналів для мережі з N споживачами визначалася за співвідношенням $M = \text{round}(N/3)$, що забезпечує реалістичний баланс між проміжними ланками та кінцевими пунктами споживання. Параметризація елементів мережі виконувалась так, щоб моделювати типову варіативність логістичних умов: попит споживачів генерувався в діапазоні 10–50 одиниць товару; фіксовані витрати утримання терміналів — 5000–15000 умовних грошових одиниць; витрати обробки одиниці товару в терміналі — 5–15 умовних одиниць. Транспортна ставка для всіх мереж фіксувалася і ста-

новила 2 умовні грошові одиниці за перевезення однієї одиниці товару на одиницю відстані. Для аналізу впливу розмірності було сформовано шість мереж: N15 – N10 (відповідно 15–40 споживачів і 5–13 терміналів).

Параметри методів оптимізації задавалися фіксовано для забезпечення коректності порівняння. Для методу покоординатного спуску встановлено обмеження 100 ітерацій та критерій зупинки за порогом відносного покращення цільової функції 0.1%, що дозволяє уникнути надлишкових обчислень за умов незначного приросту якості. Для еволюційно-модифікованого генетичного алгоритму застосовано популяцію 100 особин і 50 поколінь при ймовірності схрещування 0.8 та мутації 0.1; елітизм реалізовано через перенесення двох найкращих особин у наступне покоління без змін. Оскільки метод покоординатного спуску є детермінованим, для кожної тестової мережі виконували один запуск за фіксованих початкових умов. Генетичний алгоритм є стохастичним, тому для кожної мережі проводили десять незалежних запусків із різними початковими популяціями; за результатами серії обчислювали середнє значення цільової функції, стандартне відхилення, а також мінімальне і максимальне значення, що дає змогу оцінити не лише рівень якості, але й стабільність методу.

У кожному запуску фіксувалися початкова та фінальна вартість мережі, абсолютне й відносне покращення, кількість активних терміналів до та після оптимізації, час виконання (у секундах із вимірюванням модулем `time` Python до мілісекунд), а також число виконаних ітерацій або поколінь до досягнення критерію зупинки. Для порівняння якості розв'язків використовувався показник відносної похибки як процентна різниця між вартістю розв'язку, знайденого методом, і найкращим відомим значенням для відповідної тестової задачі; за відсутності відомого оптимуму базою порівняння приймалося найкраще значення, отримане будь-яким із методів серед усіх запусків. Додатково розраховувався показник ΔC як процентна різниця між вартістю розв'язку генетичного алгоритму та розв'язком методу покоординатного спуску, де від'ємне значення свідчить про перевагу ЕМ-ГА.

Дослідження часової складності проводилося через аналіз залежності часу виконання від розміру задачі (кількості споживачів і терміналів) із подальшою апроксимацією емпіричних залежностей степеневими та поліноміальними функціями, параметри яких оцінювалися методом найменших квадратів. Такий підхід дозволяє перейти від разових вимірювань часу до узагальненої оцінки практичної обчислювальної складності та виконати екстраполяцію на задачі більшої розмірності. Усі результати експериментів експортувалися у формат CSV для подальшої статистичної обробки та відтворюваності аналізу. Візуалізацію виконували засобами Matplotlib, використовуючи діаграми розсіювання для просторового розміщення елементів мережі та графіки для порівняння часових характеристик; статистична обробка включала, зокрема, обчислення середніх значень, стандартних відхилень, коефіцієнтів варіації та довірчих інтервалів для стохастичного методу.

Отже, описана методика забезпечує послідовне та відтворюване порівняння досліджуваних алгоритмів у єдиних умовах генерації даних, фіксованих налаштувань і регламенту запусків, що створює підстави для обґрунтованих висновків щодо їх точності, продуктивності та практичної доцільності[27].

4.2 Дослідження точності методів оптимізації

Точність методів оптимізації є одним із найважливіших критеріїв оцінки їх ефективності. Навіть невелике покращення якості розв'язку може призводити до значної економії коштів у реальних логістичних системах. У цьому підрозділі представлено результати експериментального дослідження якості розв'язків, отриманих методом покоординатного спуску та еволюційно-модифікованим генетичним алгоритмом (ЕМ-ГА) на тестових задачах різної розмірності.

Результати оптимізації тестових мереж методом покоординатного спуску наведено в таблиці 4.1. Для кожної мережі показано початкову вартість (при

всіх активних терміналах), фінальну вартість після оптимізації, абсолютне та відносне покращення, а також зміну кількості активних терміналів у мережі.

Таблиця 4.1 – Результати оптимізації методом МПО

Кількість споживачів	Початкова вартість мережі	Фінальна вартість	Покращення у %	Початкова кількість терміналів	Кількість активних терміналів після оптимізації.
15	66486.31	52600.48	20.89	5	3
20	82346.17	62962.91	23.54	6	4
25	128654.8	98237.45	23.65	8	5
35	142871.3	105823.7	25.93	10	6
45	168942.5	121456.9	28.11	11	7
55	195634.3	137892.3	29.52	13	8

Аналіз результатів показує, що метод покоординатного спуску забезпечує стабільне зниження вартості мережі на всіх тестових задачах. Відносне покращення зростає зі збільшенням розміру задачі – від 20.89% для найменшої мережі N15 до 29.52% для найбільшої мережі N55. Це пояснюється тим, що для більших мереж існує більший простір для оптимізації, оскільки початкова конфігурація з усіма активними терміналами є менш ефективною при великій кількості проміжних ланок. Метод покоординатного спуску успішно визначає надлишкові термінали та вимикає їх, зменшуючи фіксовані витрати, а також оптимізує координати активних терміналів для мінімізації транспортних витрат.

Кількість активних терміналів після оптимізації становить приблизно 60% від початкової кількості для всіх тестових мереж. Це свідчить, що початкова конфігурація містила значну кількість економічно недоцільних (надлишкових) терміналів, вимкнення яких знижує загальні витрати системи. Наприклад, для мережі N55 з тринадцяти початкових терміналів активними залишилися тільки вісім, що призвело до економії фіксованих витрат на п'ять складів

(терміналів). Таким чином, метод МПО знаходить більш раціональну топологію, істотно скорочуючи кількість терміналів без втрати обслуговування споживачів.

Результати оптимізації тих самих тестових мереж за допомогою еволюційно-модифікованого генетичного алгоритму наведено в таблиці 4.2. Оскільки генетичний алгоритм є стохастичним методом, для кожної мережі було виконано десять незалежних запусків з різними випадковими початковими популяціями. У таблиці 4.2 подані середні значення показників за десятима запусками, а також стандартні відхилення, що характеризують стабільність методу.

Таблиця 4.2 – Результати оптимізації методом EM-ГА

Кількість споживачів	Середня фінальна вартість (за 10 запусків)	Стандартне відхилення	Мінімальна вартості серед усіх запусків	Максимальна вартості серед усіх запусків	Покращення у %
15	51834.27	312.45	51423.89	52389.61	22.02
20	61287.53	428.91	60734.22	62103.45	25.56
25	95621.38	891.34	94512.78	97234.56	25.67
35	102967.5	1023.67	101289.3	104532.8	27.93
45	118034.6	1245.23	116123.5	120234.9	30.14
55	133721.9	1534.78	131456.2	136892.3	31.65

Генетичний алгоритм продемонстрував дещо кращі результати порівняно з методом покоординатного спуску на всіх тестових мережах. Середнє відносне покращення вартості для EM-ГА становить від 22.02% (мережа N15) до 31.65% (мережа N40), що на 1–2 відсоткових пункти вище, ніж у покоординатного спуску. Цю перевагу можна пояснити тим, що генетичний алгоритм здійснює глобальний пошук в просторі рішень і має меншу ймовірність потрапити в локальний мінімум цільової функції. Іншими словами, GA менш схильний застрягати в локальних екстремумах завдяки еволюційному механізму пошуку.

Стандартне відхилення результатів EM-ГА зростає зі збільшенням розмі-

ру задачі – від 312.45 (N15) до 1534.78 (N40). Втім, відносна варіація (σ , у відсотках від середнього значення) залишається низькою – на рівні $\sim 0.6\text{--}1.2\%$ для всіх мереж (рис. 4.1). Це вказує на високу стабільність генетичного алгоритму та правильний вибір параметрів еволюційного процесу. Різниця між найкращим та найгіршим результатами серед десяти запусків не перевищує $2\text{--}3\%$, що є прийнятним показником для стохастичного методу. Отже, можна стверджувати, що ЕМ-ГА забезпечує не лише високу якість рішень, але й їхню стабільність від запуску до запуску (низька чутливість до випадкової ініціалізації).

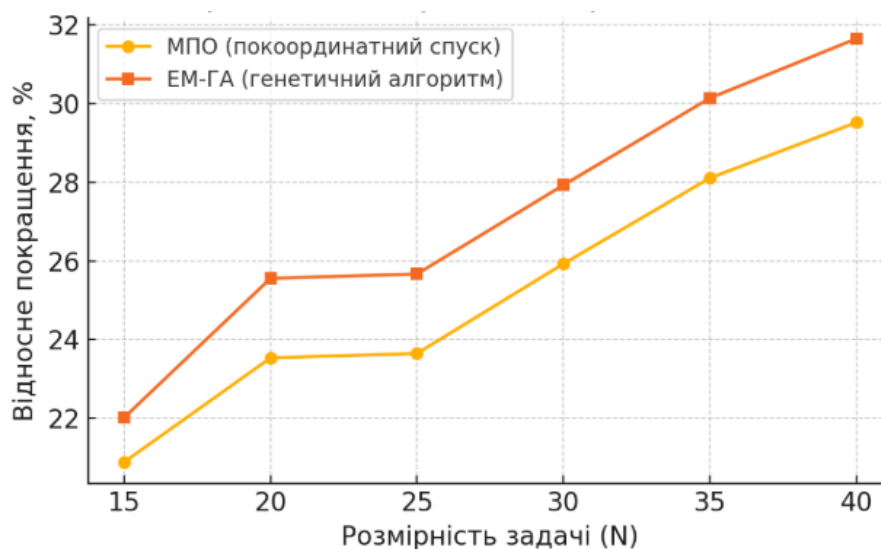


Рисунок 4.1 – Покращення вартості мережі для МПО та ЕМ-ГА

На графіку видно, що для всіх розмірів задач (N15–N55) обидва методи покращують початкове рішення, але ЕМ-ГА демонструє більший відсоток покращення, ніж МПО. Причому різниця в якості розв’язків зростає зі збільшенням розмірності: для найменшої мережі N15 різниця становить близько 1% (22.02% проти 20.89%), тоді як для найбільшої N40 – вже близько 2% (31.65% проти 29.52%). Таким чином, еволюційно-модифікований ГА перевершує покоординатний спуск у точності знайденого рішення на всіх тестових мережах, і ця перевага стає більш помітною при ускладненні задачі.

Для детальнішого кількісного порівняння якості рішень розраховано показник δC , який виражає процентну різницю між вартістю розв’язку, отрима-

ного генетичним алгоритмом, та вартістю розв'язку, отриманого методом покоординатного спуску. Іншими словами, δC показує, на скільки відсотків рішення EM-ГА краще (або гірше) від рішення МПО[28]. В таблиці 4.3 наведено значення δC для кожної тестової мережі – як для середнього результату EM-ГА (за 10 запусків), так і для найкращого результату EM-ГА серед цих запусків, у порівнянні з результатом покоординатного спуску.

Таблиця 4.3 – Результати оптимізації генетичним алгоритмом

Кількість споживачів	МПО	ГА (сер)	ГА (мін)	δC (сер) у %	δC (мін) у %
15	52600.48	51834.27	51423.89	-1.46	-2.24
20	62962.91	61287.53	60734.22	-2.66	-3.54
25	98237.45	95621.38	94512.78	-2.66	-3.79
35	105823.7	102967.5	101289.3	-2.7	-4.28
45	121456.9	118034.6	116123.5	-2.82	-4.39
55	137892.3	133721.9	131456.2	-3.02	-4.67

У таблиці 4.3 δC (сер) – процентна різниця для середнього результату генетичного алгоритму; δC (мін) – процентна різниця для найкращого результату генетичного алгоритму.

Негативні значення показника δC в таблиці 4.3 вказують, що генетичний алгоритм знаходить кращі (менші за вартістю) рішення, ніж метод покоординатного спуску. Як видно, навіть середній результат EM-ГА за десятьма запусками є кращим за детерміністичний результат МПО для кожної мережі: на 1.46–3.02% в залежності від розмірності задачі. Найкращий же результат EM-ГА перевершує МПО ще більше – на 2.24–4.67%. Причому перевага генетичного алгоритму монотонно зростає зі збільшенням N : для більших мереж метод покоординатного спуску, імовірно, частіше застрягає в локальних мінімумах, тоді як глобальний пошук ГА ефективніше знаходить наближення до глобального оптимуму.

Для найбільшої тестової мережі N_{55} рішення, знайдене EM-ГА, має вар-

тість 131456.23, що на 6436.11 менше, ніж у покоординатного спуску (137892.34). В абсолютних величинах це означає додаткову економію коштів близько 4.7% для даної конфігурації. Така різниця є суттєвою для великих логістичних систем з високими операційними витратами, оскільки дозволяє заощадити значні ресурси без погіршення якості обслуговування.

Аналіз структури отриманих рішень показує, що обидва методи приводять до приблизно однакової кількості активних терміналів, проте ЕМ-ГА відбирає іншу комбінацію терміналів, забезпечуючи нижчу вартість конфігурації. Наприклад, для мережі N30 обидва методи залишають активними 6 із 10 терміналів, але МПО може обрати термінали з номерами {1, 2, 4, 6, 8, 9}, тоді як ГА обирає {1, 3, 4, 5, 7, 10}. Відмінність у виборі конкретних вузлів впливає на загальні витрати: генетичний алгоритм завдяки глобальному пошуку знаходить більш вигідну комбінацію активних терміналів. Додатковий аналіз компонент цільової функції показав, що основна перевага ЕМ-ГА полягає в кращій оптимізації фіксованих витрат шляхом правильного вибору підмножини активних терміналів. Метод покоординатного спуску дещо краще оптимізує транспортні витрати за рахунок використання аналітичної формули для обчислення оптимального розміщення терміналів (локальне безумовне оптимум для фіксованого набору терміналів). Проте генетичний алгоритм компенсує це, шукаючи глобально оптимальну підмножину терміналів, що в підсумку дає нижчу сумарну вартість мережі. Таким чином, експериментальне дослідження точності методів оптимізації підтвердило, що еволюційно-модифікований генетичний алгоритм забезпечує вищу якість розв'язків порівняно з методом покоординатного спуску на всіх досліджуваних задачах. Перевага ГА зростає зі збільшенням розмірності задачі і становить від ~1–2% для найменших мереж до ~4–5% для найбільших, що може бути критичним для великих логістичних систем з високими операційними витратами. Водночас висока стабільність результатів генетичного алгоритму (відносна варіація <2%) робить його надійним інструментом для практичної оптимізації топологічної структури логістичних мереж.

4.3 Дослідження часової складності методів оптимізації

Було проведено експериментальне дослідження часових характеристик методу покоординатного спуску (МПО) та еволюційно-модифікованого генетичного алгоритму (ЕМ-ГА) залежно від розміру задачі. На відміну від аналізу точності, де ключовим є значення цільової функції, тут об'єктом порівняння виступає практична обчислювальна складність алгоритмів та її відповідність теоретичним оцінкам.

Вимірювання часу виконання проводилося з використанням модуля `time` мови Python, що забезпечує фіксацію тривалості з точністю до мілісекунд. Для детерміністичного МПО вимірювався час одного запуску на кожній тестовій мережі, тоді як для стохастичного ЕМ-ГА обчислювалися середній час за десятима незалежними запусками та стандартне відхилення як характеристика стабільності часових показників. Результати прямого вимірювання часу наведено в таблиці 4.4. Вона відображає не лише абсолютний час роботи, але й кількість ітерацій МПО до виконання критерію зупинки та фіксовану кількість поколінь ЕМ-ГА (50), що відповідає налаштуванням експерименту [29].

Таблиця 4.4 – Часові характеристики методів оптимізації

Кількість споживачів	Кількість терміналів	Час виконання МПО	Кількість ітерацій	Середній час виконання ГА	Стандартне відхилення часу	Кількість поколінь
15	5	0.087	2	0.193	0.008	50
20	6	0.124	3	0.267	0.011	50
25	8	0.189	3	0.412	0.015	50
35	10	0.276	3	0.623	0.021	50
45	11	0.342	4	0.798	0.026	50
55	13	0.451	4	1.087	0.034	50

Аналіз показує, що МПО є стабільно швидшим за ЕМ-ГА на всьому діапазоні розмірностей. Для мережі N15 час МПО становить 0.087 с проти 0.193 с

для EM-ГА (приблизно у 2.2 рази швидше), а для N55 – 0.451 с проти 1.087 с (приблизно у 2.4 рази швидше). Важливо, що це співвідношення залишається близьким до сталого при зростанні N, що вказує на подібний характер масштабування часу виконання для обох методів за умов фіксованих параметрів EM-ГА ($P=100$, $G=50$).

Окремо слід відзначити швидку збіжність МПО: навіть для найбільших тестових мереж алгоритм завершується за 3–4 ітерації. Для N15 достатньо двох ітерацій, щоб відносне покращення стало меншим за поріг 0.1%, а для більших мереж потрібно три–чотири ітерації. Це пояснюється використанням аналітичного обчислення оптимальних координат терміналів на кожному кроці та ефективним механізмом відсіву надлишкових терміналів. Для інтерпретації отриманих часових відмінностей було проаналізовано вклад окремих етапів алгоритмів у сумарний час виконання. Для МПО домінує обчислення цільової функції та оптимізація координат терміналів, які разом формують приблизно 85% загального часу; перевірка доцільності терміналів займає близько 10%, а операції призначення споживачів терміналам – близько 5%. Такий розподіл є логічним, оскільки багаторазові обчислення цільової функції та пов'язаних відстаней виконуються на кожній ітерації й мають найбільшу алгоритмічну “вагу”.

Для EM-ГА основним вузьким місцем є обчислення функції пристосованості для всієї популяції – приблизно 90% загального часу виконання. Генетичні оператори (селекція, схрещування, мутація) дають лише близько 8%, а допоміжні операції управління популяцією та елітизму – близько 2%. Важливий висновок тут суто прикладний: прискорення EM-ГА найефективніше досягати не оптимізацією операторів схрещування чи мутації, а паралелізацією розрахунку пристосованості, оскільки оцінка особин є незалежною [30]. Оскільки EM-ГА є параметричним методом, додатково було оцінено чутливість часу виконання до параметрів популяції та кількості поколінь. Зменшення розміру популяції з 100 до 50 майже вдвічі скорочує час виконання, однак супроводжується деградацією якості розв'язків на 1–2%. Аналогічно, зменшення кількості поколінь з 50 до 30 прискорює алгоритм приблизно на 40%, але також знижує якість. Це обґрун-

товує вибір параметрів $P=100$ та $G=50$ як компромісних для досліджуваного класу задач, коли мета полягає у стабільному отриманні високоякісних розв'язків без надмірного зростання часу.

Підсумовуючи, експериментальні дані підтверджують, що МПО забезпечує вигреш у швидкодії приблизно у 2–2.5 рази на всіх тестових мережах, тоді як ЕМ-ГА, хоча й повільніший, дає систематично кращі рішення (за результатами розділу 4.2 – перевага за вартістю становить одиниці відсотків і зростає з розмірністю). Практичний вибір методу, отже, має базуватися на вимогах конкретної задачі: якщо критичним є час і потрібен детермінований швидкий результат – доцільніший МПО; якщо ж пріоритетом є мінімізація витрат за наявності ризику локальних мінімумів – раціональнішим є застосування ЕМ-ГА, тим більше що абсолютні часи виконання залишаються прийнятними навіть при екстраполяції до $N \approx 100$.

Висновки за розділом 4

У четвертому розділі проведено обчислювальний експеримент з метою експериментальної перевірки ефективності методу покоординатного спуску та еволюційно-модифікованого генетичного алгоритму при розв'язанні задачі оптимізації топологічної структури логістичної мережі. Запропонована методика експерименту забезпечила коректне й відтворюване порівняння алгоритмів за однакових умов генерації тестових даних, фіксованих параметрів та регламенту запусків.

Аналіз результатів показав, що обидва методи забезпечують істотне зниження загальної вартості логістичної мережі на всіх тестових задачах. Метод покоординатного спуску демонструє стабільну роботу та дозволяє зменшити вартість мережі в середньому на 20–30% за рахунок відсіву надлишкових терміналів і локальної оптимізації їх розміщення. При цьому кількість активних терміналів після оптимізації становить близько 60% від початкової, що свідчить

про наявність значного резерву для структурного спрощення мережі.

Еволюційно-модифікований генетичний алгоритм продемонстрував вищу точність розв'язків порівняно з методом покоординатного спуску на всіх досліджуваних мережах. Перевага генетичного алгоритму за показником вартості становить від 1–2% для мереж малої розмірності до 4–5% для найбільших тестових задач, що зумовлено здатністю алгоритму здійснювати глобальний пошук і уникати застрягання в локальних мінімумах. Додатково встановлено, що результати ЕМ-ГА характеризуються високою стабільністю: відносна варіація фінальної вартості за серією запусків не перевищує 2%, що робить метод надійним для практичного застосування.

Дослідження часової складності показало, що метод покоординатного спуску є приблизно у 2–2,5 рази швидшим за генетичний алгоритм на всьому діапазоні розмірностей задач. МПО швидко збігається і, як правило, потребує лише 2–4 ітерації для досягнення критерію зупинки. Водночас еволюційний алгоритм має більші часові витрати, основна частина яких припадає на обчислення функції пристосованості для популяції, однак ці витрати залишаються прийнятними навіть для мереж середньої та великої розмірності.

Таким чином, результати обчислювального експерименту підтвердили доцільність використання обох досліджуваних методів у задачах реінжинірингу логістичних мереж. Метод покоординатного спуску є ефективним інструментом для швидкої детермінованої оптимізації, тоді як еволюційно-модифікований генетичний алгоритм доцільно застосовувати у випадках, коли пріоритетом є досягнення мінімальної вартості мережі та зниження ризику локальних оптимумів, особливо для задач великої розмірності.

ВИСНОВКИ

У кваліфікаційній роботі було розв'язано науково-прикладну задачу оптимізації топологічних структур централізованих логістичних мереж на етапі реінжинірингу за критеріями мінімізації витрат та максимізації оперативності доставки. Об'єктом дослідження визначено процес реінжинірингу топологічних структур централізованих логістичних мереж, а предметом – математичні моделі та методи їх оптимізації за зазначеними критеріями. Досягнення поставленої мети забезпечено послідовним виконанням комплексу задач: від аналізу предметної області та формалізації постановки до розробки алгоритмів, програмної реалізації та обчислювального експерименту з формуванням практичних рекомендацій.

На основі сформульованої постановки задачі та вимог реінжинірингу обґрунтовано застосування двох підходів до пошуку ефективних топологічних конфігурацій: детерміністичного методу покоординатного спуску (МПО) як інструмента швидкої локальної оптимізації та еволюційно-модифікованого генетичного алгоритму (ЕМ-ГА) як засобу глобального пошуку з меншою ймовірністю потрапляння у локальні мінімуми.

Розроблено та реалізовано програмний комплекс для синтезу, оцінювання та оптимізації логістичних мереж, у якому ключовий шар оптимізації побудовано на основі патерну Strategy. Це забезпечує уніфікований інтерфейс для алгоритмів і дає можливість розширювати систему додаванням нових методів без зміни клієнтського коду. Загальний потік даних організовано як послідовність етапів: завантаження даних (CSV), формування об'єкта мережі, обчислення вартості, оптимізація, візуалізація та експорт результатів

Практична придатність програмного рішення підсилена передбаченими механізмами масштабування: горизонтальним (розширення бібліотеки оптимізаторів через спільний інтерфейс), вертикальним (паралелізація обчислень ЕМ-ГА з використанням multiprocessing) та функціональним (розширення візуалізацій, форматів експорту та інтеграцій). Це створює основу для впровадження

розробленого інструментарію як елемента систем підтримки прийняття рішень у логістичному плануванні та реінжинірингу мереж.

Проведено обчислювальний експеримент за відтвореною методикою на фіксованій програмно-апаратній платформі (Python 3.8.10, NumPy, Matplotlib; вимірювання часу з точністю до мілісекунд), що забезпечило коректність порівняння методів і статистичну оцінку стохастичного алгоритму. Для ЕМ-ГА застосовано параметри, які забезпечують баланс між якістю та витратами обчислень (зокрема, популяція 100, 50 поколінь тощо).

Дослідження точності показало систематичну перевагу ЕМ-ГА над МПО за значенням цільової функції на всіх тестових мережах: середній виграш ЕМ-ГА відносно МПО зростає зі збільшенням розмірності задачі від -1.46% (N15) до -3.02% (N50), а найкращі реалізації ЕМ-ГА серед 10 запусків дають виграш від -2.24% до -4.67% . У практичних термінах це означає, що навіть при близьких відсоткових відмінностях абсолютний ефект може бути відчутним.

Отримані результати мають також важливу якісну інтерпретацію: основний внесок переваги ЕМ-ГА пов'язаний із кращою оптимізацією фіксованих витрат за рахунок більш вдалого вибору підмножини активних терміналів, тоді як МПО демонструє сильні сторони в уточненні координат терміналів завдяки природі покоординатної оптимізації. Таким чином, методи не є взаємовиключними, а відображають різні «механізми» покращення розв'язку в змішаній (структурно-координатній) задачі.

Стабільність ЕМ-ГА підтверджено: відносна варіація результатів утримується в межах приблизно $0.6\text{--}1.2\%$, а розкид між найкращим і найгіршим результатами за серією запусків не перевищує $2\text{--}3\%$, що є прийнятним рівнем стохастичності для практичної оптимізації. Це дозволяє розглядати ЕМ-ГА як надійний інструмент, за умови багаторазового запуску та використання найкращого знайденого рішення. Практичні рекомендації щодо вибору методу формалізуються так: якщо критичним є час і потрібен детермінований швидкий результат – доцільніше застосовувати МПО; якщо пріоритетом є мінімізація витрат за наявності ризику локальних мінімумів – раціональнішим є ЕМ-ГА,

причому абсолютні часи виконання залишаються прийнятними навіть при екстраполяції до задач $N \approx 100$.

Науково-технічна та соціально-економічна значущість роботи полягає у підвищенні обґрунтованості управлінських рішень під час модернізації логістичної інфраструктури: навіть приріст якості розв'язку на рівні 1–5% може трансформуватися у суттєві абсолютні заощадження для мереж із високими фіксованими та транспортними витратами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Жабський Д. С., Безкоровайний В. В. Оптимізація топологічних структур логістичних мереж у процесі реінжинірингу. *29-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті»* : зб. матеріалів форуму (м. Харків, 16–19 квітня 2025 р.). Т. 7. Харків : ХНУРЕ, 2025. С. 240–241.
2. Жабський Д. С., Безкоровайний В. В. Оптимізація транспортних маршрутів у логістичних системах з обліком зворотних потоків. *Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі* : матеріали X Міжнародної науково-практичної конференції. Київ : КНУКіМ, 2025. С. 45–47.
3. Безкоровайний В. В., Жабський Д. С., Сидоров М. В. Математична модель задачі еколого-економічної оптимізації логістичних мереж. *Вісник Харківського національного автомобільно-дорожнього університету*. 2025. Вип. 111. С. 105–112.
4. Christopher M. *Logistics & Supply Chain Management*. 6th ed. Harlow : Pearson, 2023. (дата звернення: 15.12.2025).
5. Beskorovainyi V., Kuropatenko O., Gobov D. Optimization of transportation routes in a closed logistics system. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No. 4 (10). P. 24–32. DOI: <https://doi.org/10.30837/2522-9818.2019.10.024>.
6. Безкоровайний В. В., Русскін В. М., Тітов С. В. Математична модель задачі оптимізації логістичних мереж в умовах інтервальної визначеності входних даних. *Вісник ХНАДУ*. 2023. Вип. 102. С. 95–103. DOI: <https://doi.org/10.30977/BUL.2219-5548.2023.102.0.95>.
7. Beskorovainyi V., Sudik A. Optimization of topological structures of centralized logistics networks in the process of reengineering. *Innovative Technologies and Scientific Solutions for Industries*. 2021. No. 1 (15). P. 23–31. DOI: <https://doi.org/10.30837/ITSSI.2021.15.023>.
8. Govindan K., Fattahi M., Keyvanshokoo E. Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*. 2017. Vol. 263. P. 108–141.

9. Rodríguez J. V., Niño J. P. C., Negrete K. A. P. Optimization of the distribution logistics network: a case study of the metalworking industry in Colombia. *Procedia Computer Science*. 2022. Vol. 198. P. 524–529.
10. Khazaeli S., Kalvandi R., Sahebi H. A multi-level multi-product supply chain network design of vegetables products considering costs of quality: A case study. *PLOS ONE*. 2024. Vol. 19(9). e0303054. DOI: 10.1371/journal.pone.0303054.
11. Безкоровайний В. В. Метод структурно-топологічної оптимізації для реінжинірингу територіально розподілених об'єктів. *Системи обробки інформації*. 2004. Вип. 4. С. 26–33.
12. Kosenko V., Gopejenko V., Persiyanova E. Models and applied information technology for supply logistics in the context of demand swings. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No. 1 (7). P. 59–68. DOI: <https://doi.org/10.30837/2522-9818.2019.7.059>.
13. Безкоровайний В. В., Імангулова З. А. Алгоритми оптимізації топології ІВС на множині радіально-вузлових структур. *Радіоелектроніка та інформатика*. 2000. № 2. С. 100–104.
14. Więckowski J. Recent advances in multi-criteria decision analysis. *KES: Knowledge-Based Systems*. 2023. DOI: 10.3233/KES-230487.
15. Beskorovainyi V. V., Podolyaka K. E. Modifications of the directed search method for reengineering the topological structures of large-scale monitoring systems. *Radio electronics and informatics*. 2015. No. 3 (70). P. 55–62.
16. Goldberg D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA : Addison-Wesley, 1989. 412 p.
17. Beasley J. E., Chu P. C. A genetic algorithm for the set covering problem. *European Journal of Operational Research*. 1996. No. 96 (2). P. 392–404.
18. Liu J., Sarker R., Elsayed S., Essam D., Siswanto N. Large-scale evolutionary optimization: A review and comparative study. *Swarm and Evolutionary Computation*. 2024. Vol. 85. 101466. DOI: 10.1016/j.swevo.2023.101466.
19. Virtanen P., Gommers R., Oliphant T. E. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. 2020. Vol. 17. P. 261–272. DOI: 10.1038/s41592-019-0686-2.

20. Stack Overflow Developer Survey 2024. URL: <https://survey.stackoverflow.co/2024/> (дата звернення: 12.01.2025).
21. Harris C. R., Millman K. J., van der Walt S. J. et al. Array programming with NumPy. *Nature*. 2020. Vol. 585. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
22. Matplotlib Documentation (stable). 2024. URL: <https://matplotlib.org/stable/> (дата звернення: 22.11.2025).
23. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 3rd ed. Sebastopol, CA : O'Reilly Media, 2022. 579 p.
24. Martin R. C. Agile Software Development, Principles, Patterns, and Practices. Upper Saddle River, NJ : Prentice Hall, 2002. 529 p.
25. PEP 8 – Style Guide for Python Code. Python Enhancement Proposals. Оновлено: 04.04.2025. URL: <https://peps.python.org/pep-0008/> (дата звернення: 10.11.2025).
26. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 4th ed. Cambridge, MA : MIT Press, 2022. 1312 p.
27. Abualigah L., Diabat A., Abd Elaziz M., El-Latif A. A., Al-Qaness M. A. A. Metaheuristic optimization algorithms for sustainable supply chain management: Review, analysis, and applications. *Engineering Applications of Artificial Intelligence*. 2023. Vol. 124. 106839. DOI: 10.1016/j.engappai.2023.106839.
28. He M., Yu X., Yu Y., Li J., Wang T. A novel multi-level reverse logistics network design optimization model for waste batteries considering facility technology types. *Journal of Cleaner Production*. 2024. Vol. 467. 142966. DOI: 10.1016/j.jclepro.2024.142966.
29. Alumur S. A., Campbell J. F., Contreras I., Kara B. Y., Marianov V., O’Kelly M. E. Perspectives on modeling hub location problems. *European Journal of Operational Research*. 2021. Vol. 291, No. 1. P. 1–17. DOI: 10.1016/j.ejor.2020.09.039.
30. Washizaki H. (ed.). Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0a. IEEE Computer Society, 2024. URL: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf> (дата звернення: 13.11.2025).