

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматизації та комп'ютерно-інтегрованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації  
мехатроніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розроблення компонентів технології оптимізації комп'ютерної мережі  
CALS-системи  
(тема)

Виконав:

студент II курсу, групи КІТІВМ-21-1

Смірнов М. Є.

(прізвище, ініціали)

Спеціальність 151 Автоматизація та  
комп'ютерно-інтегровані технології  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані  
технологічні процеси і виробництва  
(повна назва освітньої програми)

Керівник проф. Невлюдов І. Ш.

(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри КІТАМ

\_\_\_\_\_

Невлюдов І. Ш.

(прізвище, ініціали)

2022 р.

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматизації та комп'ютерно-інтегрованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки

Рівень вищої освіти другий (магістерський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології  
(код і повна назва)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Смірнову Миколі Євгеновичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення компонентів технології оптимізації комп'ютерної мережі CALS-системи»

затверджена наказом університету від «07» грудня 2022 р. № 1464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії «07» грудня 2022 р.

3. Вихідні дані до роботи Об'єкт дослідження – процес оптимізації комп'ютерних мереж CALS-систем. Предмет дослідження – моделі та методи оптимізації комп'ютерних мереж CALS-систем. Предмет розробки – математичні та програмні компоненти технології оптимізації комп'ютерної мережі CALS-системи. Функція – оптимізація топологічних структур комп'ютерних. Результат – програмний продукт з віконно-графічним інтерфейсом. Технічне забезпечення: IBM-сумісний персональний комп'ютер.

4. Перелік питань, що потрібно опрацювати в роботі: Вступ. Аналіз проблеми оптимізації комп'ютерних мереж CALS-систем. Огляд та аналіз CALS-систем. Структури та технології комп'ютерних мереж. Математичні моделі та методи оптимізації комп'ютерних мереж. Постановка мети та задач дослідження. Математичне забезпечення задачі. Постановка задачі, вибір та удосконалення методу оптимізації комп'ютерних мереж. Розробка алгоритмів розв'язання задачі. Розроблення програмного забезпечення задачі. Вибір мови програмування та середовища розробки. Опис програмного забезпечення Результати експериментів та їх аналіз. Висновки. Перелік джерел посилання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри): демонстраційний матеріал, представлений у форматі презентації PowerPoint (\*.ppt) на аркушах формату А4 (12-16 сторінок).

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання, аналіз завдання, уточнення плану роботи	01.09.22	Виконано
2	Аналіз сучасного стану проблеми оптимізації комп'ютерних мереж CALS-систем	08.09.22	Виконано
3	Огляд сучасних математичних моделей і методів оптимізації комп'ютерних мереж	15.09.22	Виконано
4	Розробка математичного забезпечення задачі	29.09.22	Виконано
5	Розробка програмного забезпечення задачі	13.10.22	Виконано
6	Проведення експериментальних досліджень	27.10.22	Виконано
7	Підготовка публікацій за результатами дослідження	10.11.22	Виконано
8	Оформлення пояснювальної записки	24.11.22	Виконано
9	Подання закінченої роботи науковому керівникові	26.11.22	Виконано
10	Усунення зауважень наукового керівника	01.12.22	Виконано
11	Подання роботи на рецензування	02.12.22	Виконано
12	Підготовка презентації	03.12.22	Виконано
13	Попередній захист	05.12.22	Виконано
14	Подання роботи до екзаменаційної комісії	07.12.22	Виконано

Дата видачі завдання \_\_\_\_\_

Студент \_\_\_\_\_  
(підпис)

Смірнов М. Є. \_\_\_\_\_

Керівник роботи \_\_\_\_\_  
(підпис)

зав. каф. Невлюдов І. Ш.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 79 с., 4 табл., 29 рис., 1 дод., 31 джерело.

### КОМП'ЮТЕРНА МЕРЕЖА, МАТЕМАТИЧНА МОДЕЛЬ, ОПТИМІЗАЦІЯ, ТОПОЛОГІЧНА СТРУКТУРА, CALS-СИСТЕМА.

Об'єкт дослідження – процес оптимізації комп'ютерних мереж CALS-системи.

Предмет дослідження – моделі та методи оптимізації комп'ютерних мереж CALS-системи.

Мета кваліфікаційної роботи – підвищення ефективності функціонування комп'ютерних мереж CALS-систем за рахунок оптимізації їх топологічних структур.

Методи дослідження – математичного моделювання, покоординатного спуску, спрямованого перебору.

Розроблено математичні та програмні компоненти технології оптимізації комп'ютерної мережі CALS-системи. Запропоновано удосконалений алгоритм методу спрямованого перебору з покоординатною оптимізацією розміщення вузлів мережі на основі процедури «Мультистарт». Для методів розроблено схеми алгоритмів та здійснено їх програмну реалізацію мовою Java.

Розроблені компоненти технології оптимізації можуть бути використані на підприємствах і в організаціях, де вирішуються задачі проєктування чи реінжинірингу комп'ютерних мереж. Їхнє практичне використання дозволить за рахунок підвищення точності розв'язання задач скорочувати наведені витрати на створення й експлуатацію комп'ютерних мереж.

Результати кваліфікаційної роботи апробовані на 3-х наукових конференціях.

## **ABSTRACT**

Explanatory note: 79 p., 4 tab., 29 fig., 1 add, 31 sources.

COMPUTER NETWORK, MATHEMATICAL MODEL, OPTIMIZATION, TOPOLOGICAL STRUCTURE, CALS SYSTEM.

The object of the study is the optimization process of computer networks of the CALS system.

The subject of research is models and methods of optimization of computer networks of the CALS system.

The goal of the qualification work is to increase the efficiency of computer networks of CALS systems by optimizing their topological structures.

Research methods – mathematical modeling, coordinate descent, directional search.

The mathematical and software components of the computer network optimization technology of the CALS system have been developed. An improved algorithm of the directed search method with coordinate-based optimization of the placement of network nodes based on the "Multistart" procedure is proposed. Algorithm schemes were developed for the methods and their software implementation was carried out in the Java language.

The developed components of the optimization technology can be used at enterprises and organizations where the tasks of designing or reengineering computer networks are solved. Their practical use will make it possible to reduce the stated costs of creating and operating computer networks by increasing the accuracy of problem solving.

The results of the qualification work were tested at 3 international conferences.

## ЗМІСТ

Вступ.....	8
1 Аналіз проблематики оптимізації комп'ютерних мереж CALS-систем .....	10
1.1 Огляд та аналіз CALS-систем .....	10
1.2 Основні поняття та опис технологій комп'ютерних мереж.....	17
1.3 Методи та математичні моделі оптимізації топології комп'ютерних мереж...30	
1.4 Постановка мети та задач дослідження.....	36
1.5 Висновки до першого розділу.....	37
2 Математичне забезпечення задачі оптимізації комп'ютерних мереж CALS-систем.....	37
2.1 Постановка задачі оптимізації комп'ютерних мереж CALS-систем .....	37
2.2 Вибір методу оптимізації комп'ютерних мереж CALS-систем.....	38
2.2.1 Метод покоординатного спуску.....	38
2.3 Розробка цільової функції .....	47
2.4 Розробка алгоритму оптимізації комп'ютерних мереж.....	49
2.5 Висновки до другого розділу .....	51
3 Розроблення програмного забезпечення оптимізації комп'ютерних мереж CALS-систем.....	53
3.1 Вибір мови програмування.....	53
3.2 Вибір середовища розробки .....	55
3.3 Опис програмного забезпечення.....	60
3.4 Огляд результатів експерименту та його аналіз.....	65

3.5 Висновки на третього розділу.....	71
4 Охорона праці.....	70
4.1 Загальні положення.....	70
4.2 Вимоги до фахівця у час праці.....	70
4.3 Вимоги до приміщення.....	71
4.3 Основні вимоги безпеки перед початком робіт.....	73
4.4 Вимоги після закінчення робіт.....	73
4.5 Висновки до четвертого розділу.....	74
Висновки.....	75
Перелік джерел посилання.....	77
Додаток А Презентація.....	80

## ВСТУП

Сучасні комп'ютерно-інтегровані виробництва реалізуються на основі CALS-систем і технологій. Такі системи забезпечують повну інформаційну підтримку життєвого циклу виробу, починаючи з етапу генерації ідеї його створення та, закінчуючи його утилізацією. Принципи CALS надають можливість значно зменшити обсяги проектувальних й управлінських робіт та суттєво спростити процеси обміну інформацією щодо виробу між учасниками процесів проектування, виготовлення, супроводу й утилізації продукту.

Основу інфраструктури CALS-систем складають комп'ютерні мережі, що реалізують доступ усієї множини користувачів до головного вузла, в якому зберігається інформація щодо всіх етапів життєвих циклів поточного, перспективного та попередніх продуктів, через проміжні вузли. Проміжними вузлами можуть слугувати локальні сервера, робочі станції окремих підрозділів, точки доступу тощо. Задачі оптимізації комп'ютерних мереж CALS-систем передбачають визначення їх структури, топології, параметрів та технології збору й обміну інформацією в мережі. З урахуванням того, що множини параметрів обладнання та технологій збору й обміну інформацією в мережі є невеликими, основну складність мають комбінаторні задачі оптимізації їх структур і топології елементів.

Для того, щоб вирішити проблеми, які виникають при проектуванні топології комп'ютерних мереж CALS-систем та у процесі їх експлуатації, слід проаналізувати методи оптимізації топології комп'ютерних мереж, обрати найбільш ефективний для вирішення поточної задачі, проаналізувати його складові та покращити їх.

Метою кваліфікаційної роботи є підвищення ефективності функціонування комп'ютерних мереж CALS-систем за рахунок оптимізації їх топологічних структур

Об'єкт дослідження – процес оптимізації комп'ютерних мереж CALS-системи.

Предмет дослідження – моделі та методи оптимізації комп'ютерних мереж CALS-системи.

Для досягнення мети необхідно вирішити множину задач:

- виконати аналіз сучасного стану проблеми оптимізації комп'ютерних мереж CALS-систем;
- сформулювати постановку задачі розробки компонентів технології оптимізації комп'ютерних мереж;
- обрати метод, запропонувати його удосконалення та побудувати схеми алгоритмів розв'язання задачі;
- обрати мову програмування та середовище розробки програми;
- створити програму для розв'язання задачі;
- провести перевірку працездатності розробки;
- спланувати, провести комп'ютерні експерименти, оцінити точність отриманих результатів;
- розглянути питання безпеки життєдіяльності співробітників у відділі моделювання.

Результати кваліфікаційної роботи опубліковані у матеріалах 3-х міжнародних конференцій.

Кваліфікаційна робота оформлена згідно з вимогами ДСТУ 3008:2015 [1], а також з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти [2].

# 1 АНАЛІЗ ПРОБЛЕМАТИКИ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ CALS-СИСТЕМ

## 1.1 Огляд та аналіз CALS-систем

На поточний час, багато підприємств використовують або перейшли на використання CALS-систем для управління процесами життєвого циклу продукції [3]. Такі принципи дозволяють значно зменшити обсяг проектувальних робіт та спростити процес передачі даних серед учасників виготовлення виробу [3]. Окрім цього, використання концепції інформаційної інтеграції в процесах життєвого циклу продукту значно підвищує такі компоненти якості, як ефективність виготовлення, безпека зберігання інформації та її обмін, рівень добротності самого продукту тощо.

### 1.1.1 Суть концепції інформаційної інтеграції в процесах життєвого циклу продукту

У своєму базовому розумінні, принципи Continuous Acquisition and Life cycle Support (CALS) являють собою концепцію інформаційної підтримки життєвого циклу виробу на всіх його етапів. Така безперервна підтримка життєвого циклу відбувається з використанням єдиного інформаційного простору, що дає змогу вносити нову інформацію щодо виробу або корегувати її всім учасникам життєвого циклу продукту. Головна ідея концепції CALS – це створення єдиної інтегрованої моделі виробу, яка супроводжує продукт впродовж всього його життєвого циклу (ЖЦ). На кожному етапі ЖЦ, така модель буде доповнюватись додатковими властивостями виробу, які впроваджуються в ході проходження певного етапу його ЖЦ. Також розроблена інтегрована модель виробу є дуже корисною для створення послідовних аналогічних виробів.

До концепції CALS відносяться такі базові принципи, як без паперове введення документації щодо етапів життєвого циклу виробу та обмін даними з використанням електронного підпису, аналіз, реінжиніринг та паралельний інжиніринг бізнес- процесів, інтегрована логістична підтримка тощо.

Зазвичай інформація про виріб зберігається в базах даних на мережевих серверів. Доступ до такої системи надається всім користувачам CALS-системи поточного продукту. Учасники життєвого циклу виробу мають змогу вносити нові дані стосовно виробу. Використання такої системи значно скорочує час та зменшують обсяг робіт витрачених на проектування за рахунок того, що в базах даних мережевих серверів знаходиться наступна інформація:

- опис робіт по проектуванню та складових частин минулих виробів;
- проблеми, які були виникненні в ході розробки минулого продукту та шляхи їх розв'язку;
- документація минулих виробів;
- всі дані кожного етапу життєвого циклу виробу з першого до поточного;
- детальний опис та результати функціонування поточних складових продукту.

За рахунок того, що вся документація кожного етапу життєвого циклу виробу введеться за допомогою комп'ютерного середовища, стає не потрібним використання паперового документообороту. Завдяки цьому, значно підвищується швидкість розроблення, експлуатації та утилізації продукції. Також, враховуючи, що інформація стосовно виробу зберігається у базі даних мережевої системи замість паперової документації, виключається людський фактор загублення важливих даних. Також процедура передачі інформації про виріб іншим учасникам, як займаються проектуванням продукту стає набагато зручніше. Слід врахувати, що обмін даними з використанням такої системи є набагато безпечнішим та конфіденційним тому, що не передбачається авторизація третіх осіб, які не беруть участі у виготовленні продукту та вся документація зберігається у електронному

виді замість фізичного.

Використання CALS-системи значно зменшує часові та ресурсні витрати на провадження та підтримку інформаційної інтеграції між декількома підприємствами. За допомогою CALS принципів, процес обміну інформації між підприємствами стає більш зручнішим та безпечнішим, ніж використовуючи паперовий документооборот. Також це значено спрощує документооборот між підприємством та його замовниками. Дуже важливо розуміти, що кожне підприємство може створювати свої стандарти інтеграції даних, на основі базових принципів Continuous Acquisition and Life cycle Support.

Також слід визначити підвищення таких складових, як ефективність бізнес-процесів, керованість бізнес-процесів, конкурентоспроможності виробів, кооперація з іншими підприємствами тощо. Окрім цього, слід звернути увагу на наступні вагомні скорочення:

- витрат та трудомісткості, які були витрачені на технічну підготовку та освоєння процесів виробництва;
- частки браку та витрат, які пов'язані зі змінами у конструкторських робіт;
- витрат на експлуатацію, обслуговування та ремонт виробів.

### 1.1.2 Опис складових компонентів CALS-систем

У своєму базовому розумінні за допомогою CALS забезпечується інформаційна підтримка всього життєвого циклу продукту. Життєвий цикл виробу – це сукупність взаємозв'язаних процесів зміни стану виробу [4]. Тобто життєвий цикл продукту являє собою всі стадії його розвитку, починаючи з моменту прийняття рішення про необхідність у створенні виробу та закінчуючи повної утилізації.

У стандартному виді, життєвий цикл виробу складається з наступних етапів [4]:

- науково-дослідні та дослідно-конструкторські роботи (НДДКР);
- проектування продукту;
- технологічна підготовка виробництва;
- саме виробництва;
- обслуговування та експлуатація продукту;
- утилізація виробу.

На стадії науково-дослідних та дослідно-конструкторських робіт зароджується базове уявлення щодо продукту та команда розробників повинна відповісти на такі питання: «Для чого необхідно створювати продукт?», «Які є аналоги?», «Чим краще наш виріб за аналоги?», «Які дослідження необхідно провести?» тощо. Також на цьому етапі проводиться пошук рішень основних комерційних проблем, аналіз поточного стану ринку, прогнозування попиту на виріб та розвиток його технічних характеристик.

Після формулювання основної ідеї продукту та проведення всіх необхідних науково-дослідних робіт починається стадія проектування виробу. На поточному етапу, перед командою розробників стає задача у реалізації таких проектних процедур, як формулювання проектного рішення, розробка геометричних форм продукту та його креслень, проведення розрахунків стосовно функціонування продукту, формулювання математичної моделі виробу (на основі математичної моделі моделюються процеси функціонування виробу).

Як тільки команда закінчила проектування виробу, вся інформація з цього етапу заноситься до відповідної бази даних CALS-системи, яка знаходиться на мережевих серверів та після цього настає наступний етап життєвого циклу виробу, а саме – технологічна підготовка продукту (ТПП). На стадії ТПП команда розробляє маршрутну та операційну технологію виготовлення складових виробу, технологію складання та монтажу, технологію контролю якості виробництва та випробувань.

Після завершення підготовки до виробництва та занесення відповідних даних до баз даних на мережевих серверів, розпочинається саме виробництва виробу. На

цьому етапі відбуваються наступні дії:

- формування календарних термінів;
- оперативне планування;
- аналіз ринку матеріалів, проведення нормо контролю над необхідними матеріалами, придбання матеріалів та комплектуючих для виготовлення виробу;
- придбання необхідних матеріалів та комплектуючих для виготовлення продукту;
- проведення необхідних видів обробки таких, як механообробна;
- контроль якості над результатами обробки;
- збирання виробу;
- приведення необхідних експериментів та іспитів над виробом;
- ґрунтуючись на результати, які були отриманні в ході проведення випробувань над виробом, встановлення останнього контролю якості.

Після вдалого виготовлення виробу, його необхідно запустити до продажу. На етапі обслуговування та експлуатація продукту необхідно правильно продати та обслуговувати виріб. Команда, яка відповідальна за сектор продаж, повинна знайти шлях для успішної презентації продукту. Презентація виробу має дуже важливе значення для успішного його продажу. Необхідно донести до споживача, чому йому слід купляти та використовувати саме цей продукт. Зазвичай, клієнт звертає увагу на такі аспекти, як ціна, присутність новітніх технологій, відповідність до сучасним вимогам щодо дизайну та форми, дотримання всіх вимог якості функціонування, тривалість термінів обслуговування виробу та якість наданого обслуговування. У свою чергу, компанії, яка займається виробництвом виробу, слід надавати клієнту, як можна довший та якісний термін обслуговування. Окрім цього, слід впровадити успішні логістичні рішення. Логістичні витрати, становлять велику частину загальних витрат. Слід продумати кожен деталь стосовно перевезення виробу та змодельовати цей процес для отримання результатів та усунення можливих проблем. Також, виходячи зі специфіки продукту, клієнту необхідно надати роботи стосовно

монтажу.

Розглянемо промислові автоматизовані системи. Виробництво високотехнологічних виробів, реалізація всіх поставлених цілей щодо нього та керування життєвим циклом виробу є неможливим без використання автоматизованих систем (АС).

CAD (Computer-aided Design) – це програмний пакет, який застосовується для створення конструкторських креслень, технологічної документації та 3D моделі виробу. Зазвичай CAD використовується разом з CAE. Отримані результати заносяться до CAM. За допомогою CAD команда конструкторів продумує та надає виробу форму та інші конструкторські характеристики.

CAE (Computer-aided Engineering) – це програмний засіб, який використовується для проведення інженерних розрахунків та аналізу фізичних процесів продовж функціонування моделі виробу. За допомогою цього програмного пакету чітко встановлюється, яку поведінку буде мати виріб у час реальної експлуатації. Зазвичай, під час проведення інженерних розрахунків, CAE використовує такі методи, як метод кінцевих елементів, метод кінцевих об'ємів, метод різниць тощо.

CAM (Computer-aided Manufacturing) – це використання розробленого програмного засобу та приладів, які контролюються комп'ютером для автоматизації виробництва. Прилади виготовлюють виріб чітко по написаній програмі. Тобто для функціонування CAM системи необхідно написати програмне забезпечення, мати прилади, які будуть перетворювати базові матеріали на кінцевий продукт та компілятор, який буде перетворювати написаний код програмного забезпечення на зрозумілу для приладів мову.

EDM (Enterprise Data Management) – це процес інвентаризації та керування бізнес даними, а також внесення організації в ці процеси. Зазвичай менеджерами кооперативних даних являються менеджери проекту, адміністратори бази даних або IT адміністратори. Їх основна задача керувати усім життєвим циклом бізнес даних,

починаючи від внесення інформації до бази даних та закінчуючи видаленням не потрібної. Керуючи даними, вони менш вразливі до порушень, неправильного аналізу або юридичних ускладнень. Під юридичними ускладненнями розуміється можливий витік персональних даних, які були занесені до системи.

PDM (Product Data Management) – це програмний засіб, який надає можливість управлінню проектними даними виробу. До основного функціоналу PDM системи відноситься управління зберіганням даних і документів, управління процесами та потоками робіт, управління структурою продукту, автоматизація генерації вибірок і звітів та механізм авторизації. PDM-системи є невід’ємною частиною PLM-системи тому, що зберігаються дані не тільки корисні для етапу виробництва, а ще й для експлуатації та утилізації виробу [4]. Окрім цього PDM-системи надають необхідні дані для коректної роботи MRP (Material Requirement Planning) та CRP (Capacity Requirement Planning) систем.

PLM (Product Lifecycle Management) надає можливість управління даними всього життєвого циклу виробу. Впродовж життєвого циклу виробу, на кожному етапі використовуються свої системи автоматизації. PLM-система виступає як єдиний інформаційний простір для кожної системи автоматизації і об’єднує у собі різні методики та засоби інформаційної підтримки життєвого циклу виробу. Окрім цього, PLM-система надає можливість забезпечити не тільки взаємодію автоматизованих систем в рамках одного підприємства, а також впровадити інтеграцію автоматизованих систем різних підприємств.

CRP (Capacity Requirement Planning). Використовуючи CRP бізнес процеси, компанія може чітко визначити скільки необхідно робочих, трудових та машинних ресурсів для реалізації поставлених цілей щодо виробу. Також це надає можливість компанії прорахувати, скільки продуктів може бути виготовлено [4]. Для планування вимог завантаженості, повинні бути так вхідні дані, як завантаженість кожного робочого центру, терміни виконання, замовлення кожного цеху, маршрути та заплановані випуски замовлень. Всі ці дані беруться з інших систем автоматизації

PLM середовища. Кожна компанія розробляє свої CRP процеси. Для цього, спочатку необхідно визначити робочу завантаженість (види робіт, характеристика кожного робочого, робочі процеси тощо), оцінити необхідно ресурси для виготовлення продукту, порахувати кількість зроблених робіт [4]. Після цього, слід визначити поточну завантаженість. Для цього треба знайти кількісну оцінку кожного компонента, зазначеного в угоді про рівень надання послуг, спів ставити цю оцінку з цілями та визначити встановлення використання ресурс. Тепер, на основі отриманих даних, необхідно оцінити часові терміни щодо виготовлення продукції [5]. Також слід розуміти, що повторна переоцінка має дуже важливе значення.

MRP (Material Requirement Planning). MRP (Material Requirement Planning) надає можливість компаніям мати уявлення про вимоги до запасів для задоволення попиту, допомагаючи компанії оптимізувати рівень необхідних запасів та виробничі терміни [5]. MRP розбиваються на чотири етапи, а саме визначення вимог для задоволення попиту, перевірка запасів та визначення ресурсів (використання MRP для перевірки та зміну статусу деталі та його місце знаходження), створення термінів виробництва (система визначає необхідні часові рамки та кількість робіт для того, щоб виробництво проходило без затримок та згідно з часовими термінами), виявлення проблем та формулювання рекомендацій для їх рішення [6].

## 1.2 Основні поняття та опис технологій комп'ютерних мереж

### 1.2.1 Загальна характеристика компонентів комп'ютерних мереж

У базовому своєму розумінні комп'ютерна мережа представляє собою сукупність серверів, процес обміну інформації між якими відбувається за рахунок каналом передачі даних [6]. Передача даних може відбуватися використовуючи кабельне або бездротове середовище.

До основних можливостей комп'ютерних мереж необхідно віднести:

- швидка передача даних серед компонентів комп'ютерної мережі, які знаходяться на великій відстані від кожного;
- обмін інформації у режимі «offline»;
- обмін інформації у режимі реального часу;
- віддалене керування комп'ютерами;
- спільний доступ до периферійних пристроїв;
- спільне використання ресурсів.

Обладнання, яке отримує або передає аналоговий чи бездротовий сигнал через комп'ютерну мережу називається DCE (Digital Communication Equipment) [7]. В якості DCE зазвичай виступає мережевий концентратор, комутатор або модем. Основна ціль DCE це обробити отриманий сигнал та передати його або до DTE (Data Terminal Equipment), або до іншого DCE іншої комп'ютерної мережі. В якості Data Terminal Equipment (DTE) виступає будь який апарат, який функціонує, як джерело інформації або як адресат для цифрових даних у двійковому вигляді. Data Terminal Equipment може бути персональним комп'ютером, факсом, принтером або будь якою іншою машиною, яка формулює або отримує цифрові дані. DTE формулює сигнал у цифровому виді та передає його до DCE. В свою чергу DCE оброблює сигнал та конвертує його у прийнятну форму середовища, для передачі до DCE іншої мережі. DCE іншої мережі, отримує цей сигнал, конвертує його у прийнятну для зачитування форму DTE цієї мережі та передає йому [7].

На рисунку 1.1 можна побачити типову схему процесу передачі даних у комп'ютерних мережах використовуючи кабельне середовище.

На рисунку 1.1 об'єкти «Комп'ютер 1» та «Комп'ютер 2» представляють собою Data Terminal Equipment, а «Модем 1» та «Модем 2» – Digital Communication Equipment.

Мережеві пристрої використовують спеціальний набір системних правил для здійснення передачі даних між собою через дротові або бездротові технології [8]. Такі правила називаються протоколами передачі даних. Також протокол передачі

даних можна описати, як «мова спілкування» для приладів комп'ютерної мережі.

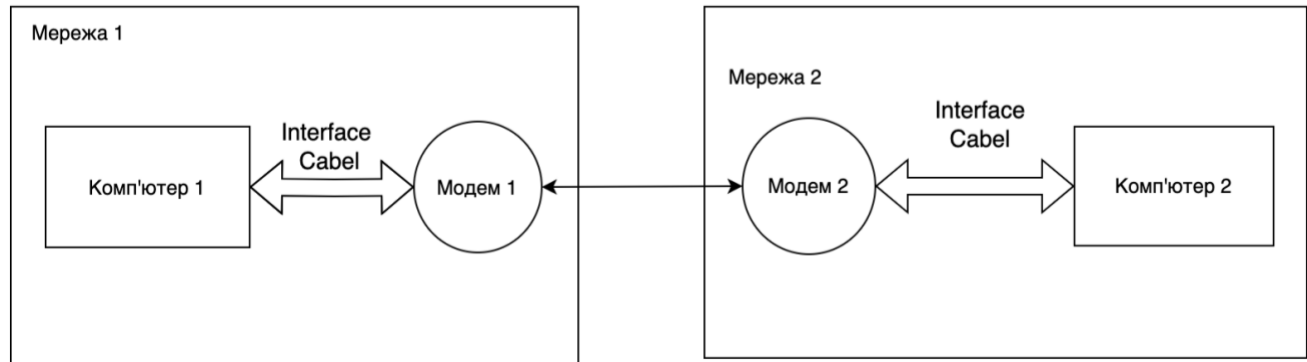


Рисунок 1.1 – Процес передачі даних у комп'ютерних мережах з використанням кабельного середовища

Існує декілька видів протоколу передачі даних, а саме:

- фізичний рівень;
- канальний рівень;
- мережевий рівень;
- транспортний рівень;
- сесійний рівень;
- репрезентативний рівень;
- прикладний рівень.

Фізичний рівень – це самий низький рівень, який відповідає за фізичне підключення між приладами комп'ютерної мережі, передачі кожного біта з одного вузла до іншого. При передачі даних, цей рівень отримує сигнал, конвертує його у біти («0» та «1») та відсилає його до канального рівня. Фізичний рівень виконує такі функції, як синхронізація бітів, контроль бітрейту, визначення топології розташування приладів/вузлів та режиму передачі даних у мережі [8].

Канальний рівень відповідає за забезпечення безпомилкової передачі пакету даних по фізичному рівні до відповідного хосту використовуючи його MAC адресу. Пакет з мережевого рівня ділиться на певну кількість кадрів, залежно від розміру

кадру NIC (Network Interface Card). У кожного кадру є заголовок, в якому міститься MAC адреси відправника та отримувача [8]. В свою чергу, каналний рівень ділиться на такі підрівні, як керування логічного зв'язку та доступом до середовища. Рівень керування логічного зв'язку (Logical Link Control) забезпечую керування передачею даних та забезпечую перевірку та правильність передачі інформації. Рівень керування доступом до середовища (Media Access Control) виступає у якості інтерфейсу між фізичним рівнем та рівнем керування логічного зв'язку та забезпечує механізм керуванням доступу до мережі. Канальний рівень виконує такі функції, як створення кадрів, додавання MAC адресів отримувача та відправника у заголовок кадру, контролювання помилок, керування швидкості передачі кадра та доступом до мережі [8].

Мережевий рівень відповідає за передачу інформації від одного хоста до іншого хоста, які знаходяться у різних комп'ютерних мережах та забезпечує найкоротший маршрут передачі пакету серед можливих. Коли пакет транспортується з однієї мережі до іншої, мережевий рівень записує до заголовку пакета IP адресу відправника та отримувача. Основні функції, які виконує мережевий рівень, це використовуючи мережевий протокол визначити, який шлях передачі даних між різними мережами від джерела до отримувача буде більш підходящим та забезпечення логікою передачу даних, тобто визначається мережева схема.

Транспортний рівень відповідає за весь шлях доставки повідомлення, починаючи від етапу його відправки з однієї мережі та закінчуючи доставкою повідомлення до приладу іншої мережі [9]. У разі, якщо виникла помилка при процесі транспортування даних, транспортний рівень відповідальний за повторну відправку цього повідомлення. У разі якщо передача пройшла успішно, цей рівень зобов'язаний підтвердити це. Зі сторони відправника, транспортний рівень отримує необхідні дані, конвертує їх до сегменту та реалізує контроль за потоком та помилками забезпечуючи коректну передачу інформації. Окрім цього, до заголовку

сегмента додається номер порту джерела та отримувача та сегмент відправляється до мережевого рівня. Зі сторони отримувача, транспортний рівень зчитує номер порту у заголовку та відправляє дані до відповідної програми. Основні функції цього рівня – це сегментація, повторне складання повідомлень та додавання до сегменту адреси порту. Також транспортний рівень надає два сервіси, а саме сервіс орієнтований на підключення та сервіс без підключення. Перший сервіс має три фази – оцінка підключення, передача даних та відключення. Як тільки пакет дійде до отримувача, отримувач посилає підтвердження назад до джерела, з якого прийшли дані. Використовуючи сервіс без підключення, отримувач не сповіщає джерело про отримання пакету з даними. Такий спосіб є більш швидшим та має тільки одну фазу проте менш безпечним.

Рівень сесії відповідає за оцінку з'єднання та авторизацію, підтримує та керує сесією, забезпечуючи безпеку. Основний функціонал – це встановлення сесії, підтримка та припинення, синхронізація та контроль діалогу між декількома пристроями [9]. Для того щоб забезпечити синхронізацію, рівень сесії встановлює точки контролю синхронізації. Таким чином, ці синхронізаційні точки допомагають ідентифікувати помилки, та у разі їх наявності дані пере синхронізуються, що виключає втрату даних. Керування діалогом надає можливість двом системам взаємодіяти використовуючи напівдуплексний або повнодуплексний режими [10].

На транспортному рівні, дані рівня витягуються з прикладного рівня та оброблюються у певному форматі для передачі по всій мережі. Цей рівень забезпечує такий функціонал, як переклад даних, кодування та декодування, компресія даних [10]. Під перекладом даних розуміється перетворення даних з однієї системи до іншої, наприклад з ASCII до EBCDIC. За допомогою кодування, дані перетворюються до іншого формату або коду. Розшифрування переводить зашифровані дані у звичайний формат.

Прикладний рівень являється самим верхнім рівнем та він реалізується мережевими програмами. На цьому рівні виробляються дані, які будуть

передаватися та оброблятися по всій мережі. Прикладний рівнем виступає як «вікно» до доступу служб програми та відображення відповідних даних користувачу [10].

### 1.2.2 Класифікація комп'ютерних мереж

Комп'ютерні мережі можуть бути поділені в залежності від географічної області, яку вони покривають, функціональні можливості та засоби комунікації.

Класифікація комп'ютерних мереж за географічними можливостями, має наступний вигляд:

- персональні мережі (Personal Area Network – PAN);
- міські мережі (Metropolitan Area Network);
- локальні мережі (Local Area Network);
- глобальні мережі (Global Area Network).

Персональні мережі охоплюють персональні комп'ютери та можуть включати у себе різноманітні периферійні прилади. Розмір персональної мережі починається від кількох сантиметрів та закінчується кількома метрами [11]. Прикладом персональної мережі може бути Bluetooth з'єднання між телефоном та навушниками. Персональна мережа, може бути провідні та безпроводні. Для створення провідного типу з'єднання використовується USB або FireWire. Для безпроводного – Bluetooth, Wifi, IrDa тощо. Оскільки PAN мережі не містять маршрутизаторів, вони не можуть на пряму бути підключенням до мережі. Персональні мережі можуть бути підключеним до локальної мережі, яка у свою чергу на пряму підключається до Інтернету. Прикладом може бути використання комп'ютера з клавіатурою, безпроводними навушниками та мишею. Всі ці периферійні пристрої підключені один до одного, проте тільки комп'ютер має доступ до Інтернету [11].

Розглянемо локальні мережі. Локальні мережі – це мережа, яка складається з певної кількості комп'ютерів чи інших приладів, об'єднаних у форму мережі в межах чітко визначеної області. Прикладом може бути університет, робочий офіс,

школа, лікарня тощо. Всі користувачі такої комп'ютерної мережі, можуть спільно користуватися, як і апаратними так і програмними ресурсами. Під апаратними ресурсами мається на увазі принтер, факс, модем тощо. Майже всі локальні мережі використовують одне середовище зв'язку, оскільки воно обмежене розміром території. За допомогою використання NOS (Network Operation System) контролюються всі ресурси та діяльності із керування.

Міська мережа – це мережа, яка має більш охоплену територію ніж локальна мережа. Вона здатна охоплювати прилади, які знаходяться в межах території від десяти до ста кілометрів. Швидкість діапазону передачі даних в межах міської мережі вимірюється у Mbps. Така мережа підходить замовникам, які потребують високошвидкісний зв'язок. Зазвичай такі мережі служать у кості ISP (Internet Service Provider). Проте таку мережу досить важко створити та підтримувати.

Глобальна мережа (GAN) – це така мережа, що складається з різних взаємопов'язаних мереж, які охоплюють необмежену географічну область. Перевага глобальної мережі в порівнянні з іншими складається в тому, що вона можна охоплювати значно більші географічні території. Оскільки глобальні мережі використовуються для того, щоб підтримати мобільний зв'язок серед декількох локальних мереж, ключовою проблемою для глобальної мережі є передача даних користувача с одного LAN до іншого. Найбільш розповсюдженим типом GAN є ширококутова глобальна мережа. Ширококутова GAN – це глобальна супутникова Інтернет мережа, яка використовує портативні термінали для забезпечення телефонного зв'язку. Така мережа підключає портативні комп'ютери, які знаходяться у віддалених місцях до ширококутової мережі.

### 1.2.3 Аналіз мережевої архітектури комп'ютерних мереж

Зазвичай мережева архітектура складається з наступних важливих елементів:

- топологія локальної мережі;

- лінійно-кабельна інфраструктура;
- активне мережеве обладнання.

Локальна мережа може бути створена, використовуючи Ethernet LAN (підключення за допомогою кабелів типу Ethernet) або WLAN (бездротове). Ethernet LAN використовує кабелі для підключення приладів до мережі. У типу підключення Ethernet, установка мережі є кабельною, комутатори та маршрутизатори налаштовуються використовуючи параметри мережі [12]. Використовується стандартний Інтернет протокол, який дозволяє працювати з Інтернетом на різних приладах комп'ютерної мережі. На рисунку 1.2 зображена схема локальної мережі Ethernet.



Рисунок 1.2 – Локальна мережа Ethernet

Існує три головних типи підключення LAN топології:

- зіркова топологія;
- кільцева топологія;
- топологія шини.

У зірковій топології всі вузли підключені до центрального вузла. Прилади не підключені один до одного та передають повідомлення до центрального вузла.

Центральний вузол у свою чергу відповідальний за передачу повідомлення до необхідного адресату. Така топологія є найбільш широкою для LAN типу підключення [12]. На рисунку 1.3 наведений приклад підключення використовуючи зіркову топологію.

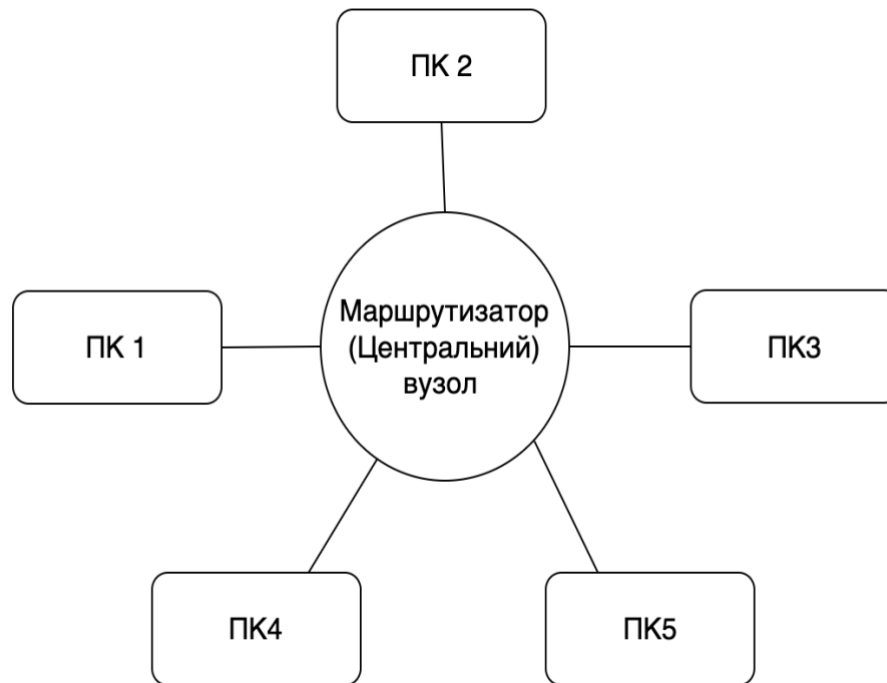


Рисунок 1.3 – Зіркова топологія підключення

При використанні кільцевої топології, підключення комп'ютерів створює кільцевий шлях передачі даних. Кожен прилад такої мережі підключається до двох сусідніх, створюючи круг. Пакети з даними відсилаються від одного приладу до іншого, поки не дійдуть до потрібного. Існує односпрямований та двонаправлений типи передачі даних, які використовуються у кільцевої топології. Односпрямований дає змогу рухатися пакетам з даними тільки по одному напрямку. Двонаправлений – в обидві сторони [12].

На рисунку 1.4 наведений приклад комп'ютерної мережі яка має зіркову топологію.

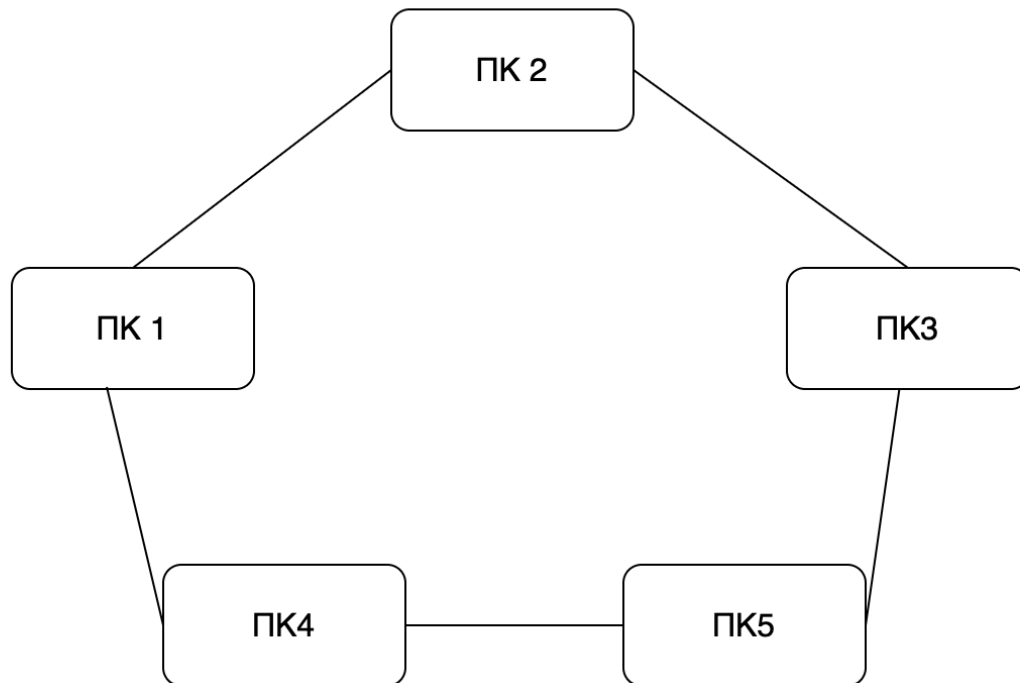


Рисунок 1.4 – Кільцева топологія підключення

Топологія шини – це така топологія, в якій прилади мережі підключені через один центральний мережевий кабель. Через цей кабель відбувається обмін пакетів з даними між всіма комп'ютерами, підключених до мережі. Зазвичай, у якості центрального кабелю, використовується один з двох типів кабелів – RJ-45 мережевий кабель або коаксіальний кабель. Існує два типи топології шини – лінійна та розподілена. При використанні лінійного типу, всі прилади мережі підключені до одного загального кабелю з двома кінцевими точками. При розподіленому типу, є один загальний кабель, який розгалужується на декілька гілок та має вже більше ніж дві кінцевих точок. Комп'ютери, які належать до розподіленого типу топології шини, можуть бути підключені до будь-якої з гілок загального кабелю [11].

На рисунку 1.5 наведений приклад шинної топології лінійного типу. За рахунок того, що кабелі не потрібні для створення інфраструктури зв'язку між комп'ютерами мережі, у бездротовому підключенні (WLAN) використовуються радіохвилі, як засіб зв'язку між приладами мережі. Також комп'ютери такої мережі

мають більшу мобільність, використовуючи бездротовий доступ до Інтернету, якщо вони знаходяться у межі дії маршрутизаторів.

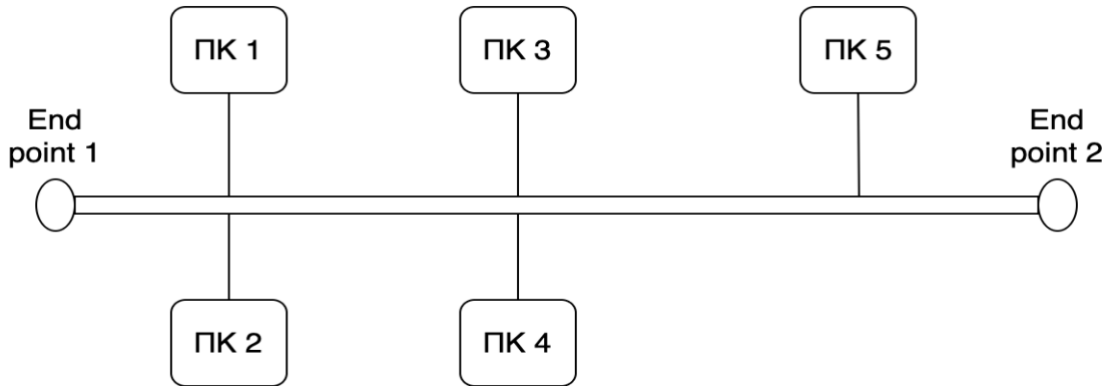


Рисунок 1.5 – Топологія шини лінійного типу

На рисунку 1.6 наведений приклад топології шини розподіленого типу.

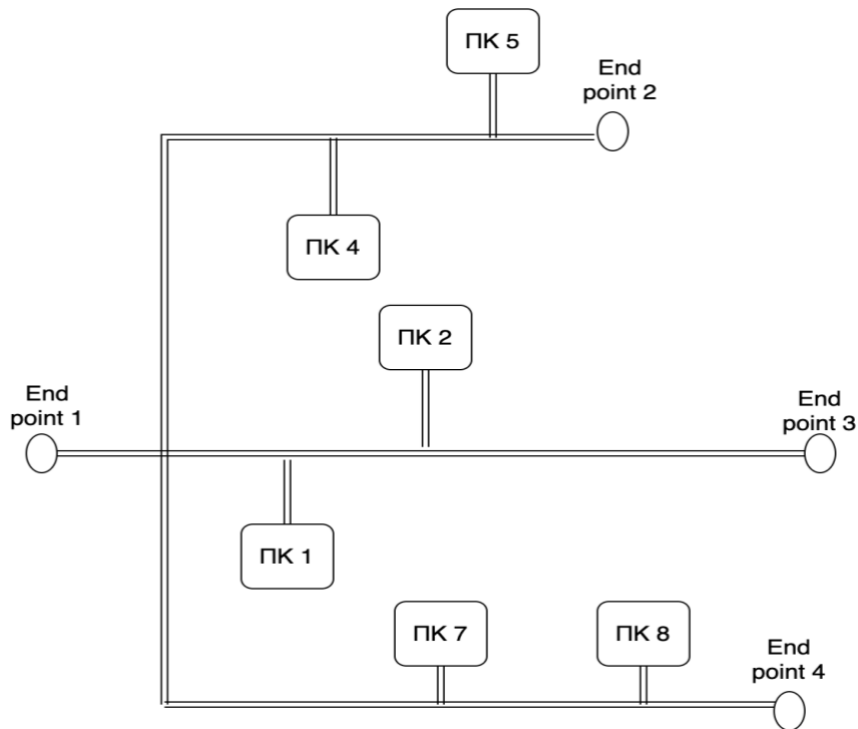


Рисунок 1.6 – Типологія шини розподіленого типу

За допомогою WLAN підключення, всі комп'ютери мережі, які знаходяться в області дії маршрутизатора, можуть одночасно бути підключеними до Інтернету. Існує декілька видів WLAN архітектури – точка доступу, станція, маршрутизатор та міст.

Архітектура точка доступу використовується для з'єднання дротової мережі з бездротової. Таким чином, прилад точка доступу підключається до маршрутизатора дротової системи та передає по сигнал іншим бездротовим учасникам мережі. На рисунку 1.7 наведений приклад використання архітектури точка доступу.

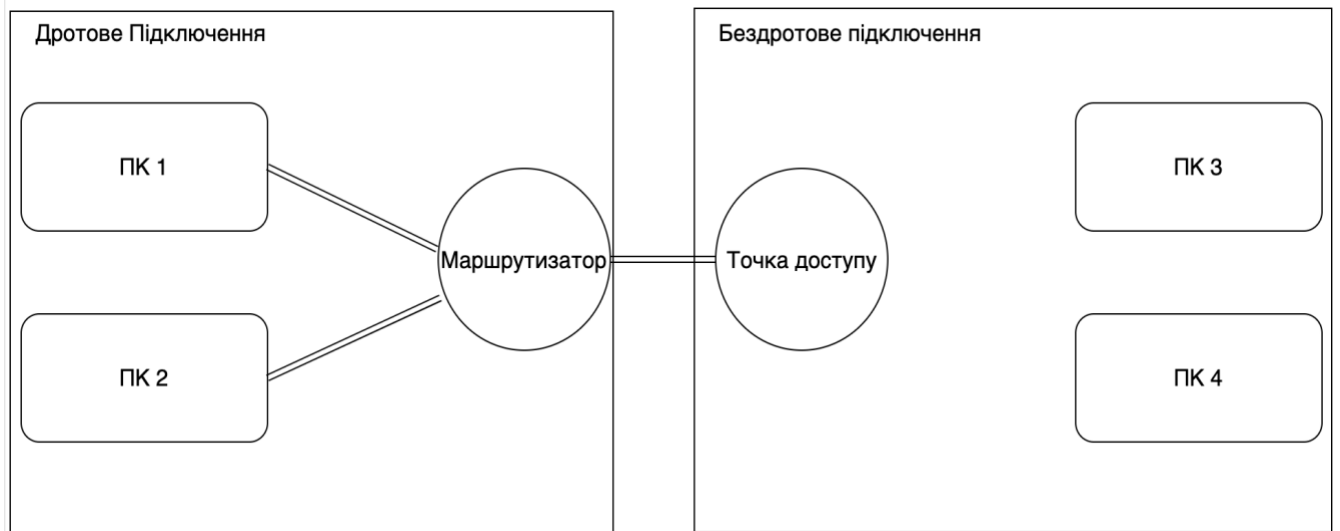


Рисунок 1.7 – Бездротова мережа з архітектурою точки доступу

Станція – це такий прилад, який має доступ до бездротової мережі та дозволяє передачу або отримання даних. Обидва кінці спільного використання даних називаються станціями. Станція це той прилад, звідки дані відправляються або який їх отримую.

На рисунку 1.8 наведений приклад мережі з використанням станцій.

Маршрутизатор працює на мережевому рівні. Він забезпечує передачу пакету з даними між джерелом та отримувачем. Маршрутизатор зчитує адресу отримувача, яка знаходиться у заголовку пакету з даними, та відправляє пакет з даними до

отримувача [13]. Також він визначає найліпший маршрут для передачі даних на основі таблиці маршрутизації.

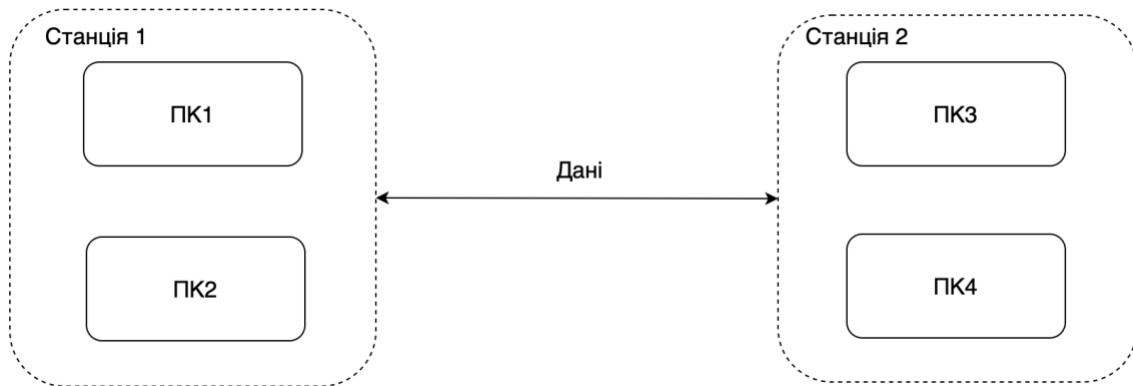


Рисунок 1.8 – Мережа з використанням станцій

На рисунку 1.9 наведений приклад мережі з використанням маршрутизатора.

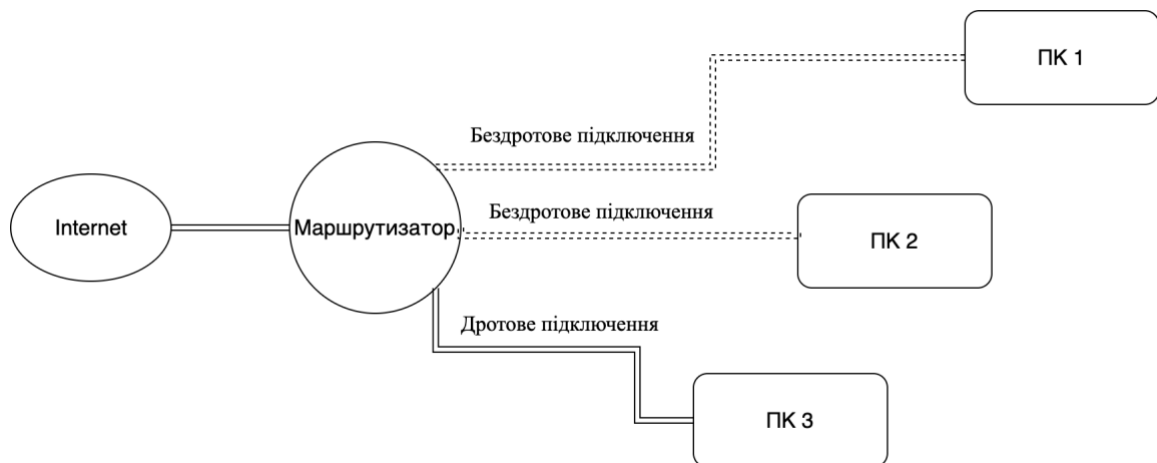


Рисунок 1.9 – Мережа з використанням маршрутизатора

Міст працює на каналному рівні мережі. Він забезпечує передачу даних від одного вузла до іншого. Також виступає в якості центрального вузла, який з'єднує дві різні мережі в одну. На рисунку 1.10 наведена мережа з використанням моста.

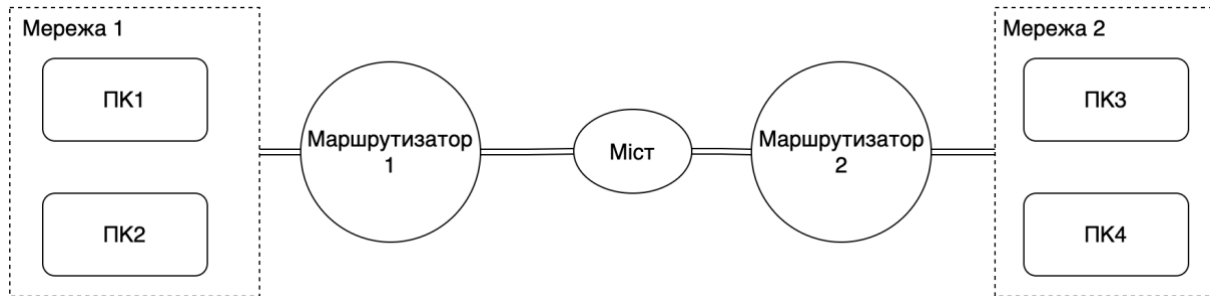


Рисунок 1.10 – Мережа з використанням мосту

### 1.3 Методи та математичні моделі оптимізації топології комп'ютерних мереж

#### 1.3.1 Методи оптимізації топології комп'ютерних мереж

Метод перебору ґрунтується на рішенні задачі шляхом перебору всіх ймовірних варіантів з послідовним вибором оптимального рішення. Оптимальне рішення залежить від критерію оцінювання. Метод прямого перебору – це різновид методу перебору, який передбачає апроксимацію чи розбиття задачі оптимізації на декілька під задач, кожна з яких може бути представлена і лінійному вигляді. Найбільш ефективними методами оптимізації є метод покоординатного спуску та Tabu Search з використанням методу перебору.

Метод покоординатного спуску – це ітераційний метод, у якому кожна ітерація отримується шляхом встановлення фіксуючого значення більшості координат вектору  $w$  та оптимізується одна з них за певним критерієм.

Мета методу покоординатного спуску – мінімізувати функцію  $\frac{\min}{w} f(w)$ , де  $f(w) = f(w_0, w_1, \dots, w_n)$  для всіх значень компонентів  $w_j$ , які збігаються з основною метою градієнтного спуску. Спочатку, необхідно мінімізувати один компонент з вектору  $w$ , причому всі інші повинні мати фіксуюче значення. Наприклад, якщо ми маємо два компонента  $w_0$  та  $w_1$ , які лежать на векторі  $W$ , тобто  $W = (w_0, w_1)$ ,

спочатку нам необхідно за визначеним критерієм мінімізувати значення  $w_0$ , причому значення  $w_1$  залишається не змінним. Після того, як мінімізували значення  $w_0$ , необхідно мінімізувати значення  $w_1$ , значення  $w_0$  залишається фіксуючим.

Представимо процес по координатного спуску у вигляді контурної діаграми (рисунок 1.11).

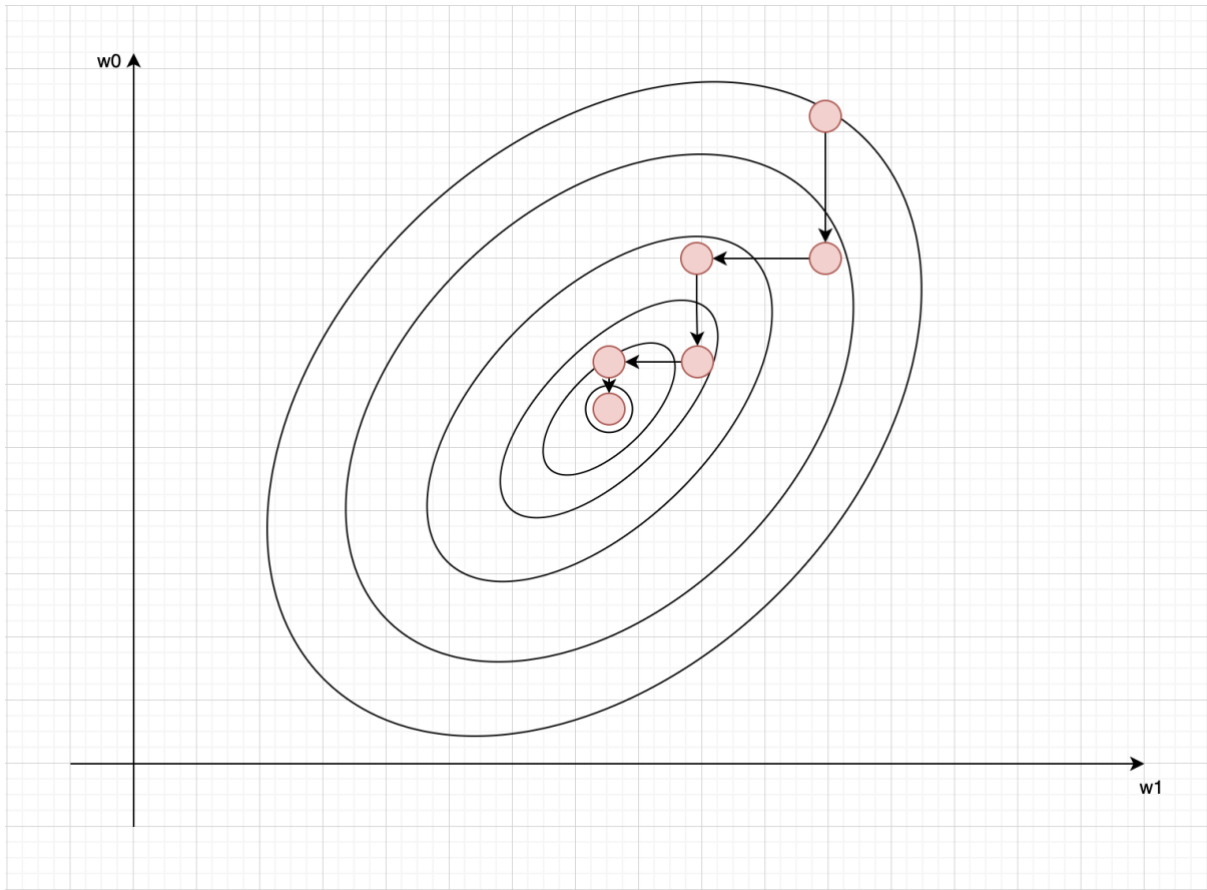


Рисунок 1.11 – Контурна діаграма процесу по координатного спуску

З рисунку 1.11 можна побачити шлях до оптимізації за певним критерієм у вигляді сходів. Це тому, що відбувається оптимізація тільки однієї координати  $w$  за один крок. У центрі знаходиться найбільш оптимальне рішення, яке задовольняє обраний критерій.

Як можна побачити з рисунку 1.11, спочатку була проведена оптимізація за координатою  $w_0$ , а  $w_1$  мала фіксуюче значення. Потім на другому кроці, було

оптимізовано координату  $w_1$  та  $w_0$  мала фіксує значення. Такі дії відбуваються до тих пір, поки не буде знайдено оптимальне рішення. На рисунку 2.11 оптимальне рішення – це найменша область, яка знаходиться у середині всіх областей та не містить в собі інші області.

Використання у базовому методі MDR розміщення вузлів шляхом по координатного спуску MCD, дає евристичний метод структурно-топологічного синтезу MDRCД. Оцінка точності метода MDRCД відбувалось шляхом порівняння рішень 106 задач синтезу трирівневих топологічних структур, включавши у себе від 10 до 40 елементів, з точними рішеннями, отриманих по методу MDCRD. Середнє значення відносної погрішності склало  $\underline{\varepsilon} = 0,0026$ , а максимальне –  $\underline{\varepsilon} = 0,0477$ .

Tabu Search метод – це мета-евристичний метод, який використовується для оптимізації параметрів моделі. Мета-евристика у своєму розумінні – це загальна стратегія, яка використовується для управління і контролю фактичної евристики. Основна ідея Tabu Search методу полягає в тому, щоб «штрафувати» рішення, які ведуть до раніше розглянутої пошукової області. Цей метод запобігає застряганню у локальному мінімуму. Tabu Search має два типи пам'яті: короткочасну та довгочасну. Короткочасна пам'ять запобігає алгоритм пошуку повторно розглядувати рішення та може бути використано для повернення до попередніх рішень, щоб локалізувати або інтенсифікувати пошук.

До основного недоліку метода Tabu Search, можна віднести зупинку при досягненні локального оптимуму при пошуку глобального. Така риса зумовлена тим, що глобальний оптимум є також і локальним, тому для успішного пошуку множини рішень виникає потреба переходу від одного локального оптимуму до іншого.

Для того, щоб подолати вищевказаний недолік, вводиться Tabu List. Такий список зберігає у собі фіксовану кількість нещодавно зроблених дій. До прикладів таких дій, можна віднести:

- поміняти місцями вузли у графі;

- видаляти або додавати ребра у графі;
- змінювати місцезнаходження вузлів комп'ютерної мережі.

Окрім ходів, список Tabu List може містити у собі певну множину рішень. Важливо розуміти, що при виборі нового рішення забороняється вибирати рішення, що містяться в списку заборон Tabu List [15].

Впровадження списку заборон Tabu List вирішує проблему застрягання у точці локального мінімум. Виходячи з цього, множина пошуку рішень значно розширюється, дозволяючи алгоритму Tabu Search знаходити більш оптимальні рішення [15].

Також для оптимізації комп'ютерних мереж CALS-систем може використовуватися метод кластеризації  $k$ -means. Такий метод є найбільш простим у порівнянні з іншими вищерозглянутими. Проте також його результати мають найменшу точність. У базовому своєму розумінні кластеризація – це розбиття певної множини вхідних векторів на кластери (групи) за рівнем схожості. Таким чином, основною метою методу кластеризації  $k$ -means є розбиття певної множини елементів векторного простору на заздалегідь відоме число кластерів  $k$  [16].

Алгоритм такого метода прагне мінімізувати середньоквадратичне відхилення на точках кожного кластера [16]. На кожній ітерації перераховується центр маси кожного кластера, який був отриманий на попередньому етапі. Після перерахування, вектори знов розбиваються на кластери відносно того, який з нових центрів знаходиться ближче до обраної метрики. Ітерації такого алгоритму закінчаться тільки тоді, коли не цій ітерації не будуть впроваджуватися зміни кластерів.

Можна виділити наступні недоліки методу кластеризації  $k$ -means:

- необхідно заздалегідь знати кількість кластерів;
- алгоритм дуже чутливий до вибору початкових центрів кластерів;
- алгоритм не підходить для ситуацій, коли об'єкт належить до різних кластерів рівною мірою.

Алгоритм методу кластеризації  $k$ -means, зводиться до пошуку кластера, який

знаходиться у мінімальній відстані до центра (формула 1.1) [17]

$$\arg \min_S \sum_{i=1}^k \sum_{\chi \in S_i} \rho(\chi, \mu_i)^2, \quad (1.1)$$

де  $\mu_i$  – це центр кластерів;

$\chi$  – це множина рішень;

$\rho(\chi, \mu_i)^2$  – функція відстані між  $\chi$  та  $\mu_i$ .

### 1.3.2 Математичні моделі оптимізації топології комп'ютерних мереж

Типова топологія комп'ютерних мереж CALS-систем має радіально-вузлову структуру. Кожен елемент такої мережі зв'язаний з центром через відповідні вузли. Під елементами такої мережі можна розуміти користувачів [18]. Ці елементи спілкуються з головним вузлом, де зберігається вся інформація життєвого циклу виробу через вузли.

При проектуванні топології комп'ютерної мережі CALS-системи накладаються наступні обмеження [18]:

- заборона створення зв'язку між елементами та між вузлами;
- до вузла повинен бути підключений, як мінімум один елемент;
- всі елементи мережі повинні підключатися до вузлів;
- до вузла повинен битв підключений, як мінімум один елемент;
- всі вузли системи в свою чергу повинні бути підключеними до центрального вузла;

- елемент не може підключатися до центрального вузла не напряму.

У багатьох випадків вважається, що центр та вузли комп'ютерних мереж CALS-систем можуть розташовуватися на основі її елементів.

Для оптимізації комп'ютерних мереж CALS-систем можна використовувати критерій ефективності, який поданий формулою (1.2) [18]:

$$C = C_C + C_U + C_{UC} + C_E + C_{EU}, \quad (1.2)$$

де  $C_C$  – витрати на центр;

$C_U$  – витрати на вузли;

$C_{UC}$  – витрати на зв'язок між центром;

$C_E$ -витрати на елементи;

$C_{EU}$  – витрати на створення зв'язку між елементами та вузлами.

Головною метою відповідно до (1.2) вважається мінімізація витрат на створення та експлуатацію топології комп'ютерних мереж CALS-систем [19].

Виходячи з цього, певна функція  $C(n_U)$  завжди прагне до мінімуму  $C(n_U) \rightarrow \min$ .

#### 1.4 Постановка мети та задач дослідження

Проведений огляд і аналіз сучасного стану питання впровадження CALS-систем і технологій дозволив встановити актуальність завдань оптимізації їх комп'ютерних мереж. Оптимізація мереж дозволяє з меншими наведеними витратами реалізувати швидку та надійну взаємодію між користувачами таких систем.

Метою кваліфікаційної роботи є підвищення ефективності функціонування комп'ютерних мереж CALS-систем за рахунок оптимізації їх топологічних структур

Для досягнення мети необхідно вирішити множину таких задач:

- виконати аналіз сучасного стану проблеми оптимізації комп'ютерних мереж CALS-систем;

- сформулювати постановку задачі розробки компонентів технології оптимізації комп'ютерних мереж;

- обрати метод, запропонувати його удосконалення та побудувати схеми

алгоритмів розв'язання задачі;

- обрати мову програмування та середовище розробки програми;
- створити програму для розв'язання задачі;
- провести перевірку працездатності розробки;
- спланувати, провести комп'ютерні експерименти, оцінити точність

отриманих результатів;

- розглянути питання безпеки життєдіяльності співробітників у відділі моделювання.

### 1.5 Висновки до першого розділу

За результатами огляду й аналізу сучасного стану питання впровадження CALS-систем і технологій встановлено, що вони знаходять усе більш широке в інтегрованих виробничих компаніях. Інфраструктуру CALS-систем складають корпоративні комп'ютерні мережі, які забезпечують взаємодію між всією множиною користувачів, що забезпечують неперервний процес проєктування виробу, проєктування технологічних процесів його виготовлення, систем керування технологічними процесами тощо.

Це обумовлює актуальність завдань оптимізації комп'ютерних мереж CALS-систем, розв'язання яких дозволяє з меншими наведеними витратами реалізувати швидкий та надійний обмін інформацією між користувачами з мінімальними наведеними витратами. На цій основі сформульовано постановку мети та задач дослідження, що передбачає, зокрема, удосконалення обраного базового методу розв'язання задачі та його програмну реалізацію.

## 2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ CALS-СИСТЕМ

### 2.1 Постановка задачі оптимізації комп'ютерних мереж CALS-систем

У рамках кваліфікаційної роботи необхідно вирішити задачу оптимізації комп'ютерних мереж CALS-систем. Для цього слід проаналізувати методи оптимізації топології комп'ютерних мереж, обрати найбільш сприятливий для вирішення поточної задачі, проаналізувати його складові та покращити його.

Вирішення задачі оптимізації топології комп'ютерних мереж CALS-систем сприяє зниженню ціни на створення та експлуатацію таких мереж.

Після неформальної постановки задачі оптимізації комп'ютерних мереж CALS-систем необхідно сформулювати математичну модель задачі, визначивши множину допустимих рішень, критерії ефективності, обмеження тощо.

Результатом методу оптимізації комп'ютерної мережі слугує покращене значення критерію ефективності. Отже, для того, щоб продемонструвати результати роботи методу оптимізації комп'ютерних мереж та визначити його переваги, необхідно порівняти його з аналогами. Після порівняння, слід описати які саме переваги в оптимізації надає запропонований метод. Для реалізації механізму порівняння, слід запрограмувати запропонований метод та метод аналог, з яким буде проходити порівняння. Зіставлення методів оптимізації повинні здійснюватися на основі заданого критерію ефективності. Також, слід зазначити про необхідність наведення діаграми, які описують процеси функціонування комп'ютерної мережі з використанням обох методів. Діаграми нададуть кількісний та більш точний опис результатів оптимізації системи.

У якості вхідних даних задається кількість вузлів  $u$ ; кількість можливих місць розташування вузлів; індекс поточного вузла –  $j := 1$ ; значення поточної ітерації –

$i := 1$ ; значення лічильника проходу по всім точкам  $l := 0$ ; початкові значення для найкращого варіанту розміщення вузлів  $w^0$ ; найкраще поточне значення для критерію  $C(w^l) := \infty$  [19].

Для того, щоб запрограмувати методи оптимізації, необхідно зробити наступні дії:

- обрати середовище розробки та мову програмування;
- встановити необхідні пакети та бібліотеки для середовища розробки та мови програмування;
- розробити алгоритм запропонованого методу оптимізації комп’ютерних мереж CALS-систем;
- розробити алгоритм методу оптимізації комп’ютерних мереж, який був обраний у якості аналога;
- задати критерій ефективності;
- задати вхідні данні;
- згідно алгоритмів методів оптимізації комп’ютерних мереж, реалізувати програмно ці методи;
- відобразити відповідну діаграму та результати розрахованого критерію ефективності у форматі “user-friendly”.

## 2.2 Вибір методу оптимізації комп’ютерних мереж CALS-систем

### 2.2.1 Метод покоординатного спуску

Основною метою оптимізації комп’ютерних мереж CALS-систем вважається мінімізація наведених витрат на їх створення й експлуатацію при забезпеченні необхідних показників їхньої оперативності надійності та живучості.

Для задоволення мети, наведеної вище, було обрано метод оптимізації покоординатного спуску. Метод оптимізації покоординатного спуску

використовується для вирішення задачі розміщення вузлів.

Припустимо, є мережа, яка складається з вузлів, елементів та центрального елемента. Використання методу покоординатного спуску (MCD) може бути реалізовано різними способами. Базова схема MCD<sub>1</sub> метода передбачає покращення деякого початкового рішення, шляхом послідовного переміщення одного (обраного) вузла при тому, що вузли  $n_U - 1$ , де  $n_U$  – це кількість вузлів, залишаються фіксованими, тобто мають незмінне положення. Причому, для кожного варіанту розміщення вузлів передбачається найкраще розміщення центра. Проте використання базової схеми MCD<sub>1</sub> метода, може бути громіздкою за рахунок того, що варіантів розміщення кожного з вузлів може бути безліч. Отже чим складніше система, тим більше необхідно обчислювальної потужності машини для визначення оптимального розміщення вузлів. Складність базового метода можна зменшити, шляхом накладання обмежень на область ймовірних переміщень вузлів діаметром  $\rho$  (варіація MCD<sub>2</sub>) або шляхом зменшення кількості місць розміщення елементів які зв'язані з цим вузлом (варіація MCD<sub>3</sub>). У подальшому буде використовуватися варіація MCD<sub>3</sub> методу оптимізації покоординатного спуску комп'ютерних мереж CALS-систем.

Основною метою варіації MCD<sub>3</sub> метода покоординатного спуску є встановлення вузлів, які базуються на основі певного елемента. Кожен раз, коли вузол змінює своє місце знаходження (перебазується до іншого елемента комп'ютерної мережі CALS-системи), топологія мережі змінюється.

Використання покоординатного методу оптимізації комп'ютерних мереж CALS-систем надає можливість скоротити витрати на її побудову, експлуатацію тощо. Завдяки відносній простоті та ефективності такий метод оптимізації використовується у різних інших областях, як логістика, бази даних, при вирішенні певних математичних задач тощо.

Нехай, є певну початкова топологія комп'ютерної мережі CALS-системи (рис. 2.1). Червоним кольором зображені ті елементи, на яких базуються вузли.

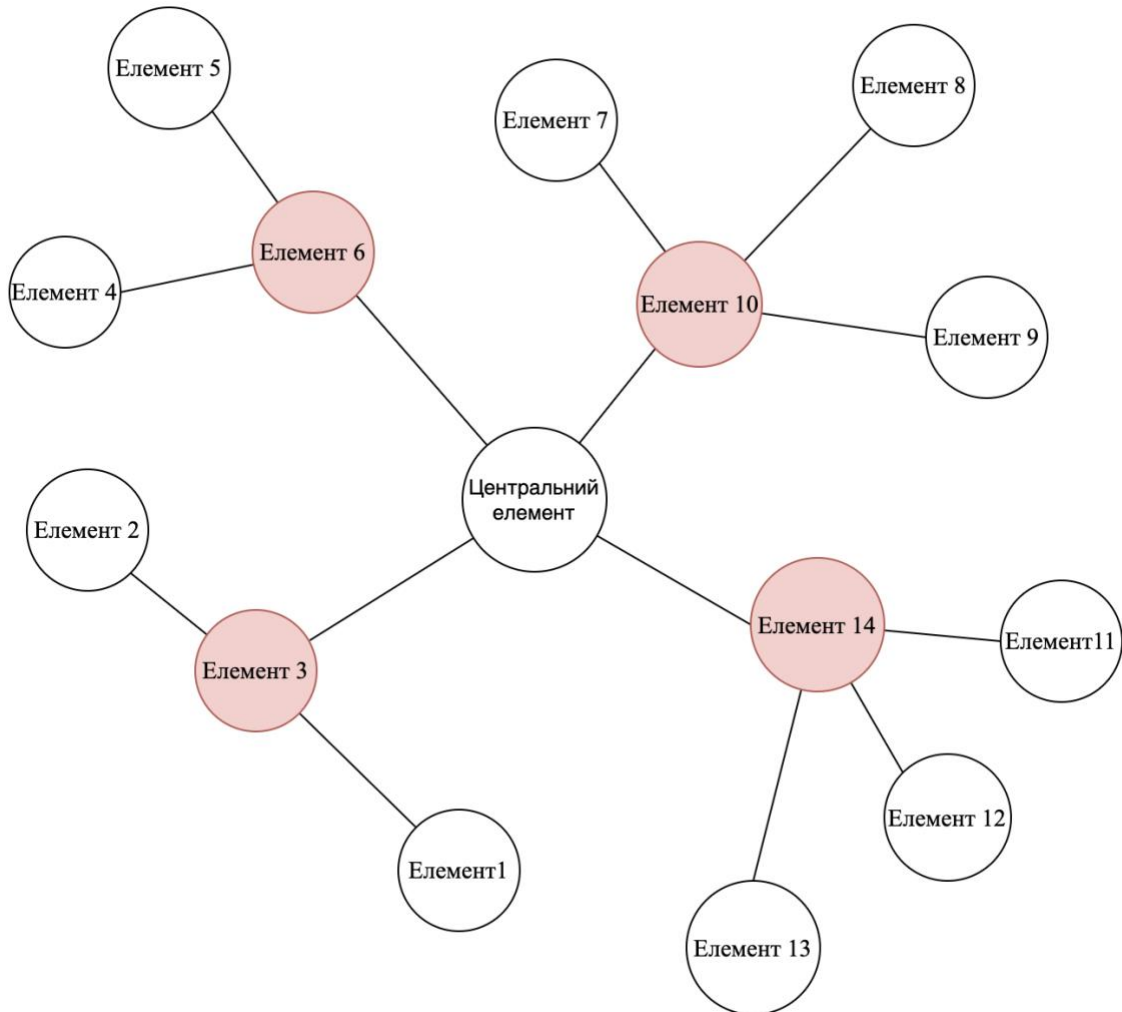


Рисунок 2.1 – Топологія комп'ютерної мережі CALS-системи без модифікацій

Звернемо увагу на елемент 3 та прилеглі до нього елементи. На рисунку 2.1 можна побачити, що вузол знаходиться на елементі 3, тобто базується на цьому елементі. Також до цього вузла підключені елементи 1 та 2. Згідно методу покоординатного спуску, необхідно розрахувати критерій ефективності для поточної топології комп'ютерної мережі CALS-системи, змінити розташування певного вузла, розрахувати цільову функцію згідно нової топології та порівняти

результати.

Припустимо, що були запроваджені зміни у топологію комп'ютерної мережі згідно методу оптимізації покоординатного спуску варіації  $MCD_3$  та тепер вузол базується на елементі 1.

На рисунку 2.2 наведена топологія комп'ютерної мережі CALS-системи після впровадженної модифікації.

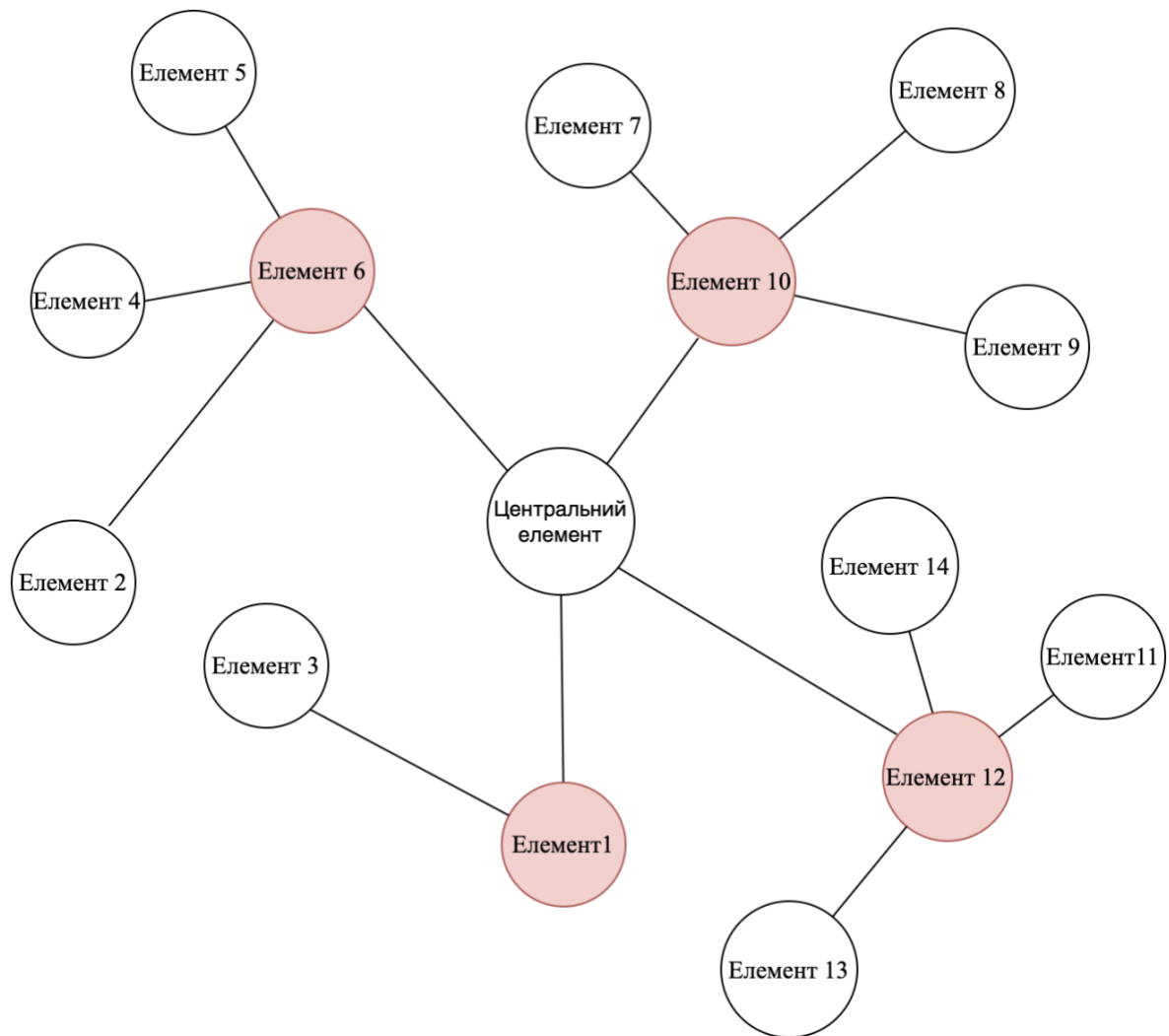


Рисунок 2.2 – Топологія комп'ютерної мережі CALS-системи після впровадження модифікації

Порівнюючи топології, які зображені на рисунках 2.1 та 2.2 помітно, що після перебазування вузла, до елемента 1, підключився елемент 2 та елемент 13. Такі перепідключення елементів до інших вузлів зумовлені тим, що після перебазування вузла, відстань від елемента до поточного вузла стала більшою та елементу вигідніше підключитися до сусіднього вузла тому, що відстань до нього менше.

Зробимо ще одну модифікацію топології комп'ютерної мережі CALS-системи. Перебазуємо вузол до елемента 12.

Нова топологія комп'ютерної мережі CALS-системи наведена на рисунку 2.3.

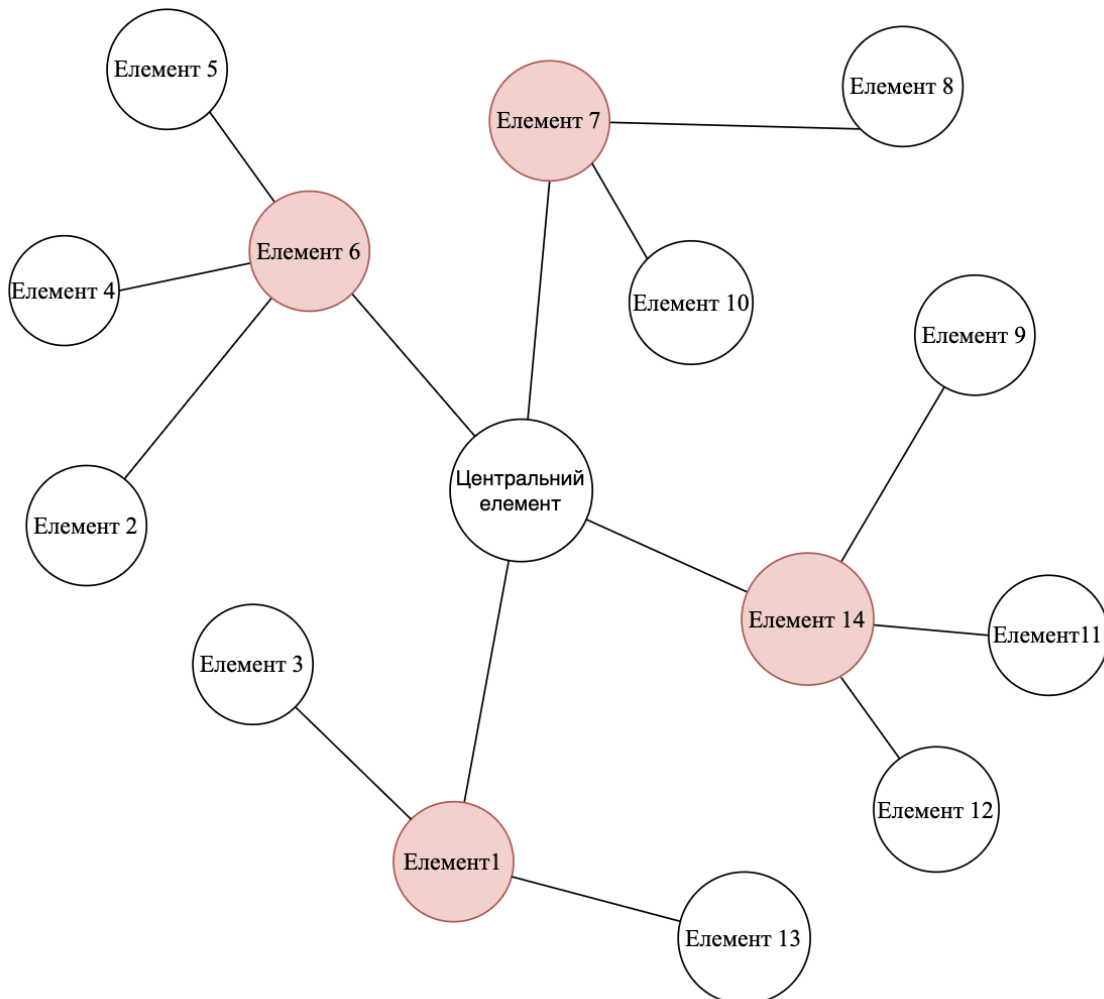


Рисунок 2.3 – Топологія комп'ютерної мережі CALS-системи після впровадження модифікації

Після того, як вузол був перебазований до іншого елемента, необхідно оцінити поточну топологію комп'ютерної мережі CALS-системи за певною цільовою функцією та порівняти її зі значенням розрахунків для попередньої топології.

Алгоритм методу по координатній оптимізації вузлів мережі [20].

Крок 1. Задати вхідні данні: кількість вузлів  $u$ ; кількість можливих місць розташування вузлів; індекс поточного вузла –  $j := 1$ ; значення поточної ітерації –  $i := 1$ ; значення лічильника проходу по всім точкам  $l := 0$ ; початкові значення для найкращого варіанту розміщення вузлів  $w^0$ ; найкраще поточне значення для критерію  $C(w^l) := \infty$ .

Крок 2. Визначити початкове розміщення вузлів  $w_i^l$  та розрахувати для нього значення критерія  $C(w_i^l)$  за формулою 1.1.

Крок 3. Збільшити значення лічильника кількості ітерацій  $i := i + 1$ ; для вузла  $j$  у розміщенні  $w_i^l$  змінити його місце розташування при фіксованих значеннях всіх інших вузлів, тобто  $u - 1$ .

Крок 4. Розрахувати значення критерія, за формулою 1.2. Якщо  $C(w^l) \leq C(w_{i-1}^l)$ , то  $C(w^l) := C(w_i^l)$ ,  $w^0 := w_i^l$  та перейти до кроку 5.

Крок 5. Збільшити індекс поточного вузла  $j := j + 1$ . Якщо  $j < u$ , тобто ще є компоненти, які на поточному кроку не були оптимізовані, перейти до кроку 3, у протилежному випадку до кроку 6.

Крок 6. Якщо  $l = 0$ , то присвоїти  $w_i^{l+1} := w_i^l$ ,  $l := l + 1$ ,  $j := 1$  та перейти до кроку 3, у протилежному випадку до кроку 7.

Крок 7. Якщо  $C(w^l) \leq C(w^{l-1})$ , то  $w_i^{l+1} := w_i^l$ ,  $l := l + 1$ ,  $j := 1$  і перейти до кроку 3, в іншому випадку – до кроку 8.

Крок 8. Закінчення алгоритму: отримане найкраще рішення щодо розташування вузлів з найменшим значенням критерія витрат  $C(w^0)$ , які були розглянуті в ході алгоритму.

На рисунку 2.4 наведена схема алгоритму покоординатної оптимізації топології комп'ютерних мереж.



Рисунок 2.4 – Схема алгоритму покоординатної оптимізації топології комп'ютерних мереж CALS-систем

### 2.2.2 Метод покоординатного спуску з функцією «Multi Start»

Основна ідея поточної модифікації полягає в тому, щоб запускати метод покоординатного спуску необхідно задану кількість разів, причому перед кожним запуском, змінювати початкове розташування вузлів. Вузли можуть розташовуватися на місцях всіх елементів, які входять до складу комп'ютерної мережі CALS-системи.

Оптимізуємо топологію комп'ютерної мережі CALS-системи, яка наведена на рисунку 2.1, використовуючи процедуру «Мультистарт». Оптимізація мережі повинна проходитися на основі певної цільової функції.

На рисунку 2.5 наведена топологія оптимізованої комп'ютерної мережі CALS-системи

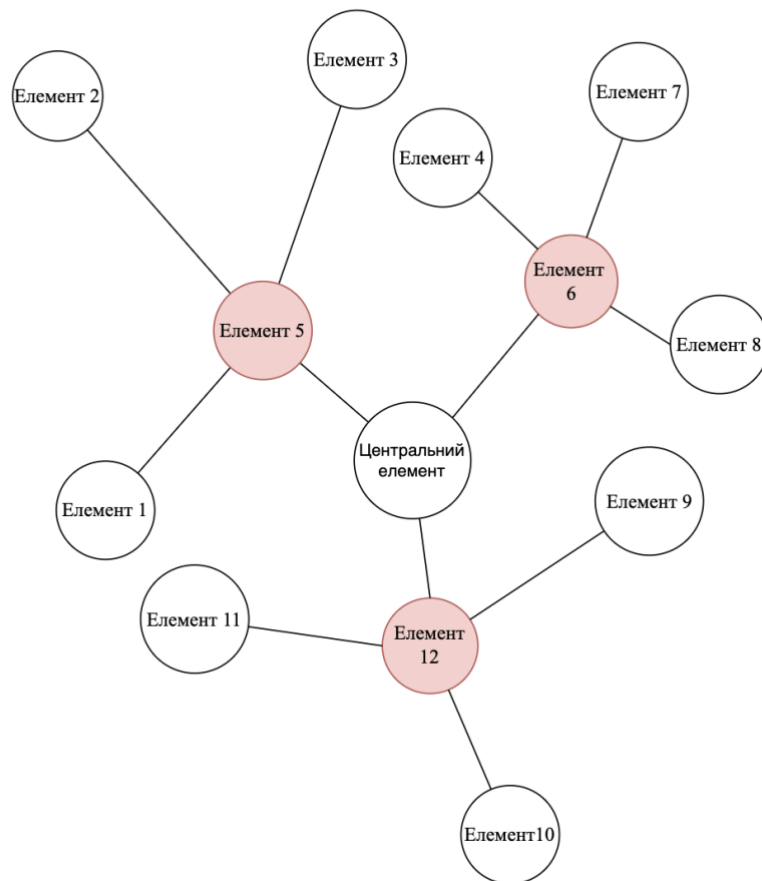


Рисунок 2.5 – Топологія оптимізованої комп'ютерної мережі CALS-системи

Тепер за методом покоординатного спуску з використанням функції «Multi Start» слід змінити розташування вузлів та ще раз оптимізувати топологію мережі. Результат оптимізації наведений на рисунку 2.6.

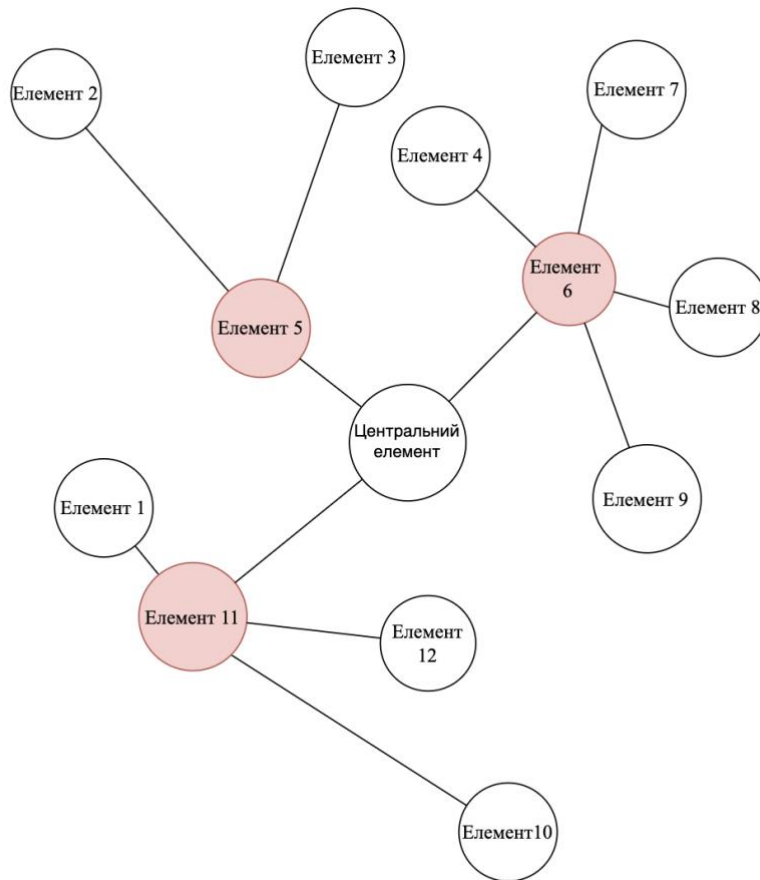


Рисунок 2.6 – Оптимізована топологія комп'ютерної мережі CALS-системи

Порівнюючи топології комп'ютерної мережі CALS-системи на рисунках 2.4 та 2.5 можна зробити висновок, що оптимізована топологія дійсно має інше розташування вузлів. Такі дії слід виконувати певну  $N$  кількість ітерацій. Отриманий на кожній ітерації критерій ефективності необхідно запам'ятовувати. Після виконання  $N$ -ї ітерації слід обрати серед множини значень те, яке найбільш близьке до вимог обраної цільової функції.

### 2.3 Розробка цільової функції

У базовому своєму розумінні, цільова функція – це математичний вираз, який використовується для того, щоб описати певний об’єкт (процес, рішення тощо) та порівняти його з іншими, які мають таку ж саму природу. Вона зв’язує основну мету оптимізації з керованими змінними, на основі яких, вирішується задача оптимізації. Мета використання цільової функції зводиться до знаходження такого значення, при якому цільова функція набуває екстремуму відносно всього відрізка значень.

У базовому виді, цільову функцію можна представити на основі формули (2.1)

$$Y = f(x_1, x_2, \dots, x_n), \quad (2.1)$$

де  $Y$  – певна лінійна функція;

$x_1, x_2, \dots, x_n$  – параметри функції.

Основу інфраструктури CALS-систем складають комп’ютерні мережі, що реалізують доступ усієї множини користувачів з центральним елементом, в якому зберігається інформація щодо всіх етапів життєвих циклів поточних, перспективних та попередніх продуктів, через проміжні вузли. У залежності від складності системи проміжними вузлами комп’ютерних мереж можуть слугувати локальні сервера, робочі станції окремих підрозділів, точки доступу тощо. Тобто в межах комп’ютерних мереж CALS-систем, маємо центральний вузол, певну сукупність вузлів та елементів. Всі вузли системи підключаються до центрального вузла. В свою чергу, до вузлів під’єднуються елементи. Під елементами можна розуміти користувачів системи. Основною метою оптимізації таких комп’ютерних мереж – є зниження ціни, витраченої на їх створення та подальшої експлуатації. Слід розуміти, що правильно спроектована топологія комп’ютерних мереж CALS-систем допоможе зменшити описані вище витрати. Таким чином, необхідно розробити таку цільову функцію, яка чітко описує вплив топології комп’ютерних мереж CALS-систем на

ціну їх створення та експлуатацію. У формулі (2.2) представлена цільова функція, яка відповідає всім необхідним потребам [20]

$$C = C_C + \sum_{i=1}^{n_E} (c_U + c_{iC}) \cdot r_{ii} + c_E \cdot n_E + \sum_{i=1}^{n_E} \sum_{j=i+1}^{n_E} c_{ij} \cdot r_{ij} \rightarrow \min, \quad (2.2)$$

де  $C_C$  – вартість центру;

$n_E$  – кількість елементів мережі;

$c_U$  – вартість вузла;

$c_{iC}, c_{Ci}$  – вартість зв'язку між  $i$ -м вузлом і центром та центром і  $i$ -м вузлом;

$c_E$  – вартість елемента;

$c_{ij}$  – вартість зв'язку між  $i$ -м та  $j$ -м елементами.

Витрати на створення та експлуатацію центра  $C_C$  та елементів  $C_E$  майже не залежать від кількості вузлів  $n_U$  та їх можна вважати постійною величиною. Витрати на створення та експлуатацію вузлів  $C_U$  мають лінійну залежність від кількості вузлів  $n_U$  та розраховується за формулою (2.3) [20]

$$C_U = c_U \cdot n_U. \quad (2.3)$$

У свою чергу зі збільшенням кількості вузлів, збільшуються витрати на створення та експлуатацію зв'язків між центром та вузлами  $C_{UC}$ , а витрати на зв'язки між вузлами та елементами  $C_{UE}$  – знижуються.

Важливо розуміти, що характер функції  $C(n_U)$  дозволяє використовувати ідею направленої перебору MDR (Method of Coordinate Descent) по кількості вузлів для рішення задачі синтезу радіально-вузлової топологічної структури ТРС комп'ютерних мереж CALS-систем.

На рисунку 2.7 наведений характер залежності витрат на мережу від кількості вузлів [20].

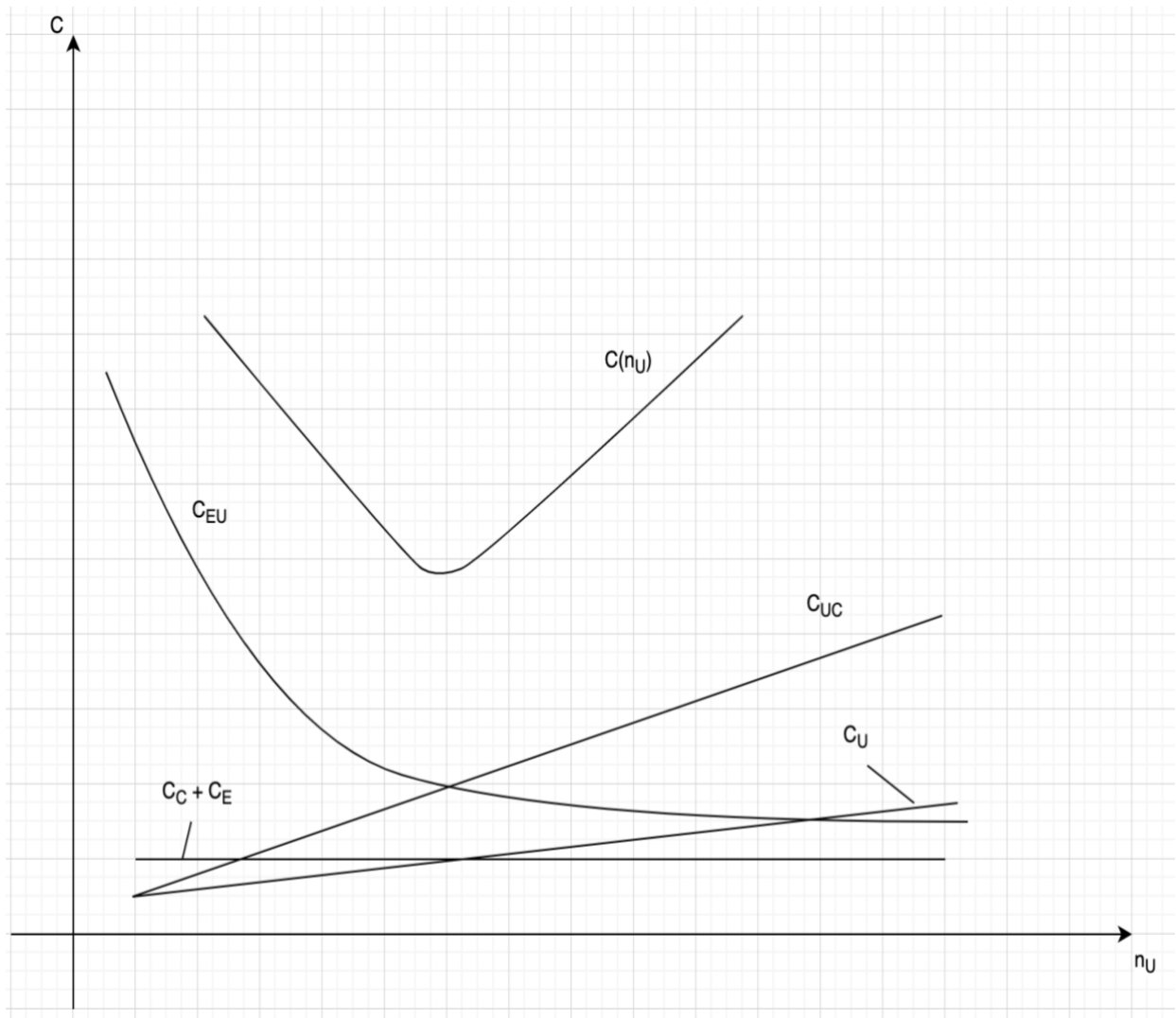


Рисунок 2.7 – Характер залежності витрат на мережу від кількості вузлів

## 2.4 Розробка алгоритму оптимізації комп'ютерних мереж

Алгоритм методу покоординатної оптимізації вузлів мережі з функцією «Multi Start».

1. Задати вхідні дані, а саме: кількість елементів  $e$ ; значення кількості ітерацій  $N$ ; значення поточної ітерації  $i := 1$ ; значення вартості центрального елемента  $C_c$ ;

значення вартості вузлів  $C_U$  та елементів  $C_E$ ; значення вартості зв'язку між центральним елементом та вузлом.

2. Визначити початкове розміщення вузлів  $w_i$  та розрахувати для нього значення цільової функції  $C(w_i)$  за формулою (2.2).

3. Реалізувати метод покоординатної оптимізації та запам'ятати отримане значення цільової функції  $C(w)$ .

4. Порівняти значення цільової функції  $C(w_i)$ , отримане після першої ітерації з еталонним значенням  $C(w)$ , у випадку якщо  $C(w_i^l) \leq C(w)$ , то  $C(w) := C(w_i)$ ,  $w := w_i$ .

5. Перевірити значення поточної ітерації  $i$ , та порівняти з значенням кількості ітерацій  $N$ . У випадку, якщо  $N > i$ , то інкрементувати  $i$  ( $i := i + 1$ ) та перейти до кроку 2. У протилежному випадку перейти до кроку 6. Запам'ятати рішення.

6. Закінчення алгоритму: отримане найкраще рішення щодо розташування вузлів з найменшим значенням критерія витрат  $C(w^0)$ , які були розглянуті в ході алгоритму.

На рисунку 2.9 наведена схема алгоритму покоординатної оптимізації топології комп'ютерних мереж з використанням функції «Multi Start» за показником наведених витрат.



Рисунок 2.8 – Алгоритм методу покоординатного спуску з функцією «Multi Start»,  
аркуш 1

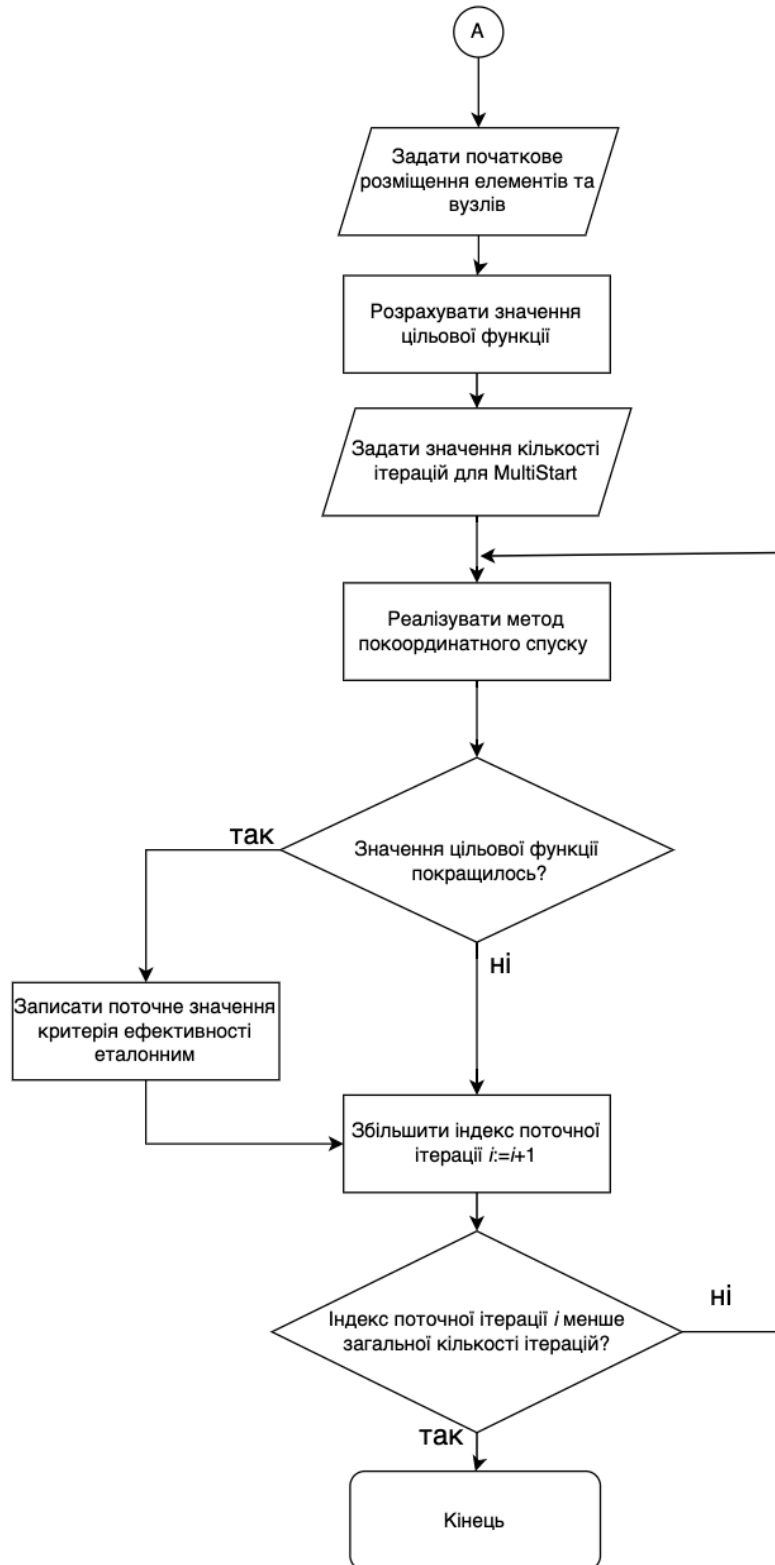


Рисунок 2.8, аркуш 2

## 2.5 Висновки до другого розділу

У ході написання другого розділу був описаний метод покоординатного спуску з використанням прямого перебору. Окрім цього, велику увагу було приділено аналізу цього алгоритму. Алгоритм наведений на рисунку 2.4.

Також, була запроваджена модифікація методу покоординатного спуску шляхом додавання функції «Мультистарт» до нього. Проведений детальний опис модифікації методу покоординатного спуску. Велику увагу було приділено створення алгоритму методу оптимізації покоординатного спуску з використанням функції «Мультистарт». Алгоритм методу з впровадженої модифікацією можна побачити на рисунку 2.8.

Виходячи з того, що математичне забезпечення технології оптимізації комп'ютерних мереж CALS-систем складають математичні моделі, методи та алгоритми розв'язання задач, сформовано склад її базових компонентів. До нього ввійшли перевірені на практиці математична модель синтезу мережі за показником наведених витрат, метод спрямованого перебору варіантів з покоординатною оптимізацією розміщення вузлів мережі та відповідний алгоритм.

На основі результатів аналізу особливостей методу покоординатної оптимізації розміщення вузлів висунута гіпотеза щодо його удосконалення на основі технології «Мультистарт». За результатами удосконалення передбачається підвищення точності методу покоординатної оптимізації. Для базового й удосконаленого методу розроблено схеми алгоритмів розв'язання задачі оптимізації топологічних структур мереж. Ефективність запропонованої модифікації методу передбачається перевірити експериментальним шляхом після програмування відповідних алгоритмів.

### 3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОПТИМІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ CALS-СИСТЕМ

#### 3.1 Вибір мови програмування

На поточний час існує багато різноманітних мов програмування, які використовуються для різних цілей. До найбільш популярних можна віднести такі мови програмування, як Java, JavaScript, C#, Python, Node JS, PHP тощо [21].

В рамках написання магістерської роботи, було обрано мову програмування Java. Java – це об'єктно орієнтована мова програмування, яка була випущена до ринку у 1995му році компанією під назвою «Sun Microsystem» [22]. При процесі зборки додатку, Java-програма компілюється до байт-коду, який при виконанні інтерпретується віртуальною машиною JVM для конкретної платформи [22]. З цього можна зробити висновок, що Java являє собою кросплатформену мову програмування. У своєму розумінні кросплатформенність – це здатність програмного забезпечення працювати з кількома апаратними платформами чи операційними системами.

У таблиці 3.1 наведені основні переваги мови програмування Java.

Таблиця 3.1 – Основні переваги мови програмування Java

Перевага	Опис
1	2
Кросплатформеність	Код програми, написаний один раз, буде працювати з будь-якою апаратною платформою або операційною системою за допомогою віртуальної машини Java (Java Virtual Machine) [22].

Продовження таблиці 3.1

1	2
Ком'юніті	Java – є дуже поширеною мовою програмування, якою користується велика кількість людей [22]. Виходячи з цього, можна знайти рішення майже будь-якої проблеми, яка виникла при написанні коду або налаштуванні середовища програмування.
Надійність	Мова Java є строго типізованою. Мається на увазі, що будь-який вираз або змінна має певний тип вже на момент компіляції, що спрощує виявлення проблем, ще на початку написання коду. Компілятор сам підказує програмісту, де потенційно може знаходитися помилка [23].
Об'єктно-орієнтованість	Усі бібліотеки, написані колись для Java – це класи, які відповідають за функціональність мови. Будь-який застосунок на Java – це набір класів, що описують різні об'єкти [23]. Це добре, тому що дозволяє створювати складні, але прості у підтримці програми. Java — мультипарадигменна мова, тобто підтримує безліч принципів програмування, що дозволяє ефективно вирішувати різні завдання.
Гнучкість	Використовуючи мову програмування Java можна розробити будь-який додаток, наприклад інтернет додаток, систему штучного інтелекту, високонавантажenu систему тощо [23].

До основних недоліків Java можна віднести [24]:

- за рахунок того, що всі програми виконуються поверх віртуальної машини Java (JVM), вони споживають більше пам'яті, якщо порівнювати з аналогами;
- відсутність контролю над збіркою сміття. Java звільняється від зайвих даних у випадкові інтервали часу;
- обмежена підтримка функцій графічного інтерфейсу;
- Java не надає підтримки для низькорівневого програмування. Ніяк не можна отримати доступ до ресурсів системного рівня за допомогою мови програмування Java.

### 3.2 Вибір середовища розробки

У загальному своєму розумінні, середовище розробки – це середовище, яке автоматизує або додає певний функціонал, який бере участь у циклі розробки програми.

Для розробників завжди потрібен редактор програмування або інтегроване середовище розробки (Integrated Drive Electronics), яке вносить свій вклад у поліпшення процесу розробки додатку, використовуючи обрану мову програмування. Він полегшує процес розробки, шляхом поліпшення опиту при використанні фреймворків та бібліотек класів.

Основними вимогами, які необхідно поставити перед собою при виборі IDE (Integrated Drive Electronics), повинні бути наступні:

- зручність використання;
- швидкість роботи та високий рівень продуктивності;
- додатковий функціонал, який покриває потреби виникаючі при написанні коду та розгортанні додатку;
- зручна інтеграція з такими провідними системами, як git тощо;
- задоволення всіх вимог щодо проведення процесу створення додатку;

– рівень особистості, навички програміста та його роль у процесі організації.

IDE дотримується простого робочого принципу, який дозволяє розробникам писати логічний код у своєму редакторі середовища. Його основна функція – повідомляти компілятору, де знаходяться потенційні помилки. Функція налагодження допомагає налагодити повний код та виправити помилки [25].

До основних функцій IDE належать наступні:

– IDE повинна мати можливість заповнення коду для ідентифікації функцій Java мови;

– він повинен мати потужне управління ресурсами, яке допомагає виявити відсутні ресурси, заголовки, бібліотеки тощо;

– хороший інструмент налагодження для повного тестування розробленого додатка;

– компілювати та будувати функції.

Головною перевагою використання IDE для розробки є те, що коли компілятор інтегрований з середовищем розробки, ми отримуємо весь пакет бібліотек в одному місці, що надає можливість написати код, скомпілювати, налагодити та виконати програму в тому самому програмному забезпеченні. Також до переваг слід віднести те, що IDE мають привабливий користувальницький інтерфейс і комплектуються всіма елементами розробки програмного забезпечення, які можна використовувати для розробки програмних додатків [12].

Існує 5 найбільш популярних середовищ розробки для мови програмування Java, а саме Eclipse, IntelliJ, NetBeans, BlueJ та Jdeveloper.

### 3.2.1 Середовище розробки Eclipse

Найбільш розповсюджена IDE – це Eclipse. Eclipse – це безкоштовне IDE, яка майже завжди використовується, або у рамках навчальних цілей, або у старих проєктів чи з обмеженим бюджетом. Також Eclipse має відкритий код. Оскільки він

відкритий, він допомагає розробникам налаштувати IDE під свої вимоги та зробити додаток більш надійним. Він заснований на базовій основі Java, і, отже, являється дуже розширюваним, гнучким і сумісним з багатьма мовами, такими як C ++, Groovy, Python, Perl, C # тощо.

До основних переваг IDE Eclipse, відноситься наступне:

- Eclipse має хорошу можливість інтеграції для створення таких інструментів, як ANT та Maven;
- користувачі можуть розробляти різні додатки на одній платформі, такі як веб- та автономні програми, веб-служби тощо;
- надійні рекомендації щодо коду та налагоджувачів, які вбудовані в Eclipse.

До недоліків IDE Eclipse відноситься наступне:

- Eclipse постачається з великою кількістю перевірок для файлів JSP та HTML;
- початкове налаштування часом стає важким без належних вказівок та документації.

### 3.2.2 Середовище розробки IntelliJ IDE

На другому місці – IntelliJ IDE від компанії Jet Brains. Така IDE вважається найліпшим вибором для розробників тому, що він покриває майже всі потреби, які виникають у програміста в ході розробки додатка. Таке середовище розробки вважається найліпшим вибором для програміста [26].

IntelliJ IDE доступний як ліцензована версія спільноти Apache 2 та у власній комерційній версії. Обидва видання можуть бути використані для комерційного розвитку [26].

Він дає поради розробнику в процесі написання, аналізу коду та надає надійні інструменти рефакторингу. Він має критично важливі інструменти, такі як система контролю версій, підтримка багатьох мов та фреймворків. Він здатний аналізувати

дії розробника та автоматично запускає відповідні інструменти.

До основних переваг IntelliJ IDE можна навести наступне [26]:

- IntelliJ Idea добре знаходить повторювані блоки коду та показує помилки перед компіляцією;
- він має потужну функцію налаштування для зміни структури проекту відповідно до потреб користувача;
- хороший інтерфейс з великою кількістю варіантів інтерфейсу користувача.

### 3.2.3 Середовище розробки NetBeans

Також до п'яти найбільш розповсюджених IDE з підтримкою мови програмування Java входить NetBeans. NetBeans – це безкоштовне інтегроване середовище розробки з відкритим кодом, яким керує Apache Software Foundation. Корисно розробляти веб-додатки, настільні, мобільні, C++, HTML 5 тощо. NetBeans дозволяє розробляти додатки із набору модульних програмних компонентів, які називаються модулями. NetBeans працює на Windows, Mac OS, Linux та Solaris.

До основних переваг NetBeans, відноситься:

- NetBeans дозволяє розробникам розгортати код із власного середовища;
- користувачі можуть формувати та визначати правила для всіх мов;
- NetBeans має паралельну функцію порівняння коду, завдяки якій подібні сторінки можна писати одночасно.

До недоліків такого середовища розробки відноситься:

- він стає повільним в обробці, через великі розміри інструменту. Виходячи з цього, бажано мати полегшену версію;
- плагіни, надані NetBeans для розробки IOS та Android, можна вдосконалити;
- інтерфейс користувача має недоліки.

### 3.2.4 Середовище розробки JDeveloper

Наступним по популярності використання йде JDeveloper. JDeveloper – це інтегроване середовище розробки з відкритим кодом, що надається корпорацією Oracle. Він пропонує функції для розробки використання таких мов, як Java, XML, SQL та PL / SQL, HTML, JavaScript, BPEL та PHP. JDeveloper охоплює повний життєвий цикл розробки, від проектування та кодування, налагодження, оптимізації та профілювання до розгортання.

До основних переваг такого IDE, відноситься наступне:

- JDeveloper IDE має потужний механізм інтеграції з гнучкими програмами для розробки та компонентами версій програмного забезпечення;
- він також має гарну підтримку клієнтів для всіх користувачів у всьому світі;
- хороший зв'язок з базою даних, і користувач також може виконувати запити SQL.

До недоліків можна віднести наступне:

- складний інтерфейс для використання;
- середовище розробки стає повільним, коли користувач намагається реалізувати бізнес-процес, оскільки він займає величезний обсяг оперативної пам'яті.

### 3.2.5 Середовище розробки DrJava

Та на останньому місці серед найбільш розповсюджених середовищ розробки, які підтримують мову програмування Java відноситься DrJava. DrJava – це безкоштовне легке інтегроване середовище розробки за ліцензією BSD, де користувач може розробляти програми Java. В основному він був розроблений для студентів та менторів, щоб забезпечити їм привабливий інтерфейс та дозволити перевірити та оцінити написаний код Java.

До основних переваг IDE, відноситься:

- це дуже легкий IDE із швидким процесом обробки та компіляції коду;
- початкове налаштування не потрібно;
- його функції взаємодії дозволяють виконувати кожен клас самостійно, тому

це позитивно впливає на швидкість пошуку та усунення несправностей.

Проте у такому середовищі розробки є наступні недоліки:

- дуже простий інструмент IDE з мінімальними функціями;
- не підходить для великих додатків з використанням режиму реального часу,

оскільки програма стає дуже повільним у виконанні.

### 3.2.5 Висновки щодо вибору середовища розробки

Зважаючи увагу на всі недоліки та переваги середовищ розробки було обрано IntelliJ IDE. IntelliJ вважається один з найкращих IDE для розробки будь-яких додатків з використанням мови програмування Java. Він надає інтеграцію з усіма провідними системами такими, як Apache TomCat, GIT, Spring, Spring Boot, Hibernnet тощо. Також слід зауважити, що IntelliJ IDE має самий високий рейтинг серед розробників. Зважаючи на його такі позитивні аспекти, як висока швидкість обробки коду, надійність, коректні зауваження, широкий функціонал для розробки додатку, легкий для розуміння та привабливий інтерфейс користування саме IntelliJ IDE віддають перевагу, як при написанні комерційних програм, так і у рамках навчання.

### 3.3 Опис програмного забезпечення

Розроблений web-додаток надає змогу побудувати топологію комп'ютерної та оптимізувати її, використовуючи надані методи. Побудувати топологію можна, або випадковим чином, або задати певні координати для кожного з елементів комп'ютерної мережі.

На рисунку 3.1 представлено стартову сторінку програми.

Welcome to Topology Calculator!

Ціна

Створити нову топологію комп'ютерної мережі CALS-системи

Введіть кількість елементів:

Введіть ціну одного метра кабелю:

Введіть ціну одного елементу:

Введіть ціну одного вузла:

Введіть ціну центра :

Прийняти

Рисунок 3.1 – Стартова сторінка програми

На рисунку 3.2 наведено вікно введення початкових даних для створення топології комп'ютерної мережі CALS-системи.

На рисунку 3.2 можна побачити вікно під назвою «Створити нову топологію комп'ютерної мережі CALS-системи», яке містить такі поля, як «Введіть кількість елементів», «Введіть ціну одного метра кабелю», «Введіть ціну одного елементу», «Введіть ціну одного вузла», «Введіть ціну центра» та кнопку «Прийняти». Припустимо, що комп'ютерна мережа CALS-системи має 15 елементів. Кожен з елементів коштує 1000 доларів, кожен з вузлів коштує 80 доларів, ціна одного метра кабелю – 5 доларів 90 центів, ціна центру – 2500 доларів тощо.

На рисунку 3.3 наведена створена топологія комп'ютерної мережі CALS-системи враховуючи вхідні дані.

Створити нову топологію комп'ютерної мережі CALS-системи

Введіть кількість елементів:

Введіть ціну одного метра кабелю:

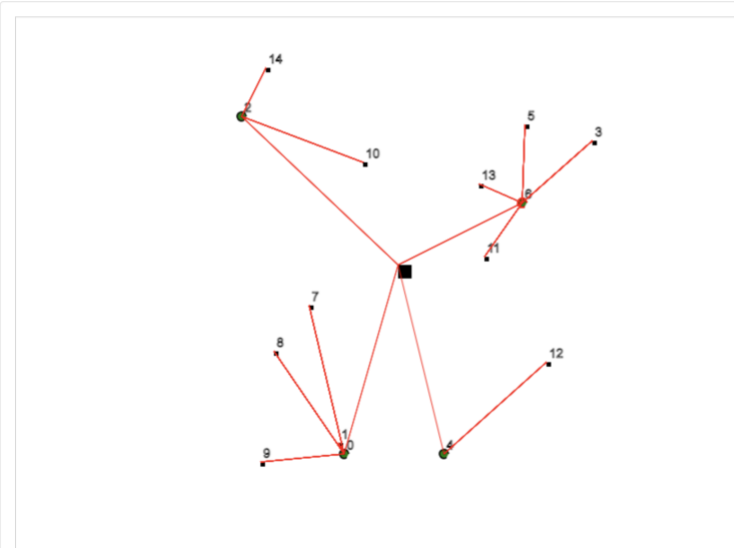
Введіть ціну одного елементу:

Введіть ціну одного вузла:

Введіть ціну центра :

Рисунок 3.2 – Вікно для введення початкових даних

Welcome to Topology Calculator!



Ціна
20899.6

Створити нову топологію комп'ютерної мережі CALS-системи

Введіть кількість елементів:

Введіть ціну одного метра кабелю:

Введіть ціну одного елементу:

Введіть ціну одного вузла:

Введіть ціну центра :

Рисунок 3.3 – Створена топологія комп'ютерної мережі CALS-системи

На рисунку 3.3 можна побачити побудовану топологію комп'ютерну мережі CALS-системи. Зеленими колами зображені вузли, Маленькими чорними квадратами – елементи, великим чорним квадратом – центр. Також можна побачити блок під назвою «Ціна». У цьому блоці відображається ціна поточної топології комп'ютерної мережі CALS-системи.

Після побудови комп'ютерної мережі CALS-системи, необхідно її оптимізувати. Для цього було запрограмовано два методи, а саме метод покоординатного спуску та метод покоординатного спуску з використанням функції «Multi Start». Для того, щоб оптимізувати топологію, слід у вікні під назвою «Оптимізувати топологію» обрати в списку один з двох запропонованих варіантів, а саме «Метод Покоординатного Спуску» або «Метод покоординатного спуску з «Multi Start». На рисунку 3.4 наведено відповідне вікно.

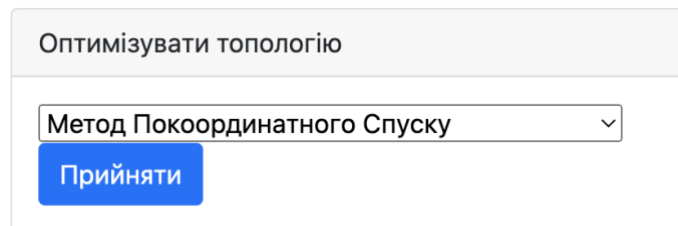


Рисунок 3.4 – Вікно для вибору методу оптимізації топології комп'ютерної мережі CALS-системи

На рисунку 3.5 зображено меню вибору методу оптимізації.

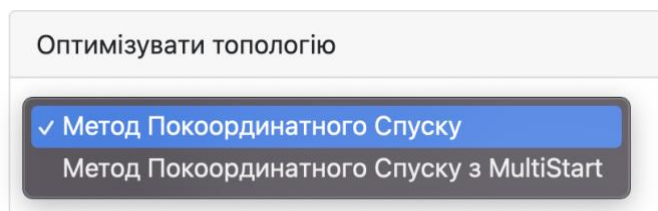


Рисунок 3.5 – Методи оптимізації комп'ютерної мережі CALS-системи

Для того, щоб порівняти ціни топології після виконання двох методів, потрібно натиснути на відповідну кнопку у вікні під назвою «Відобразити діаграму порівняння» (рис. 3.6).

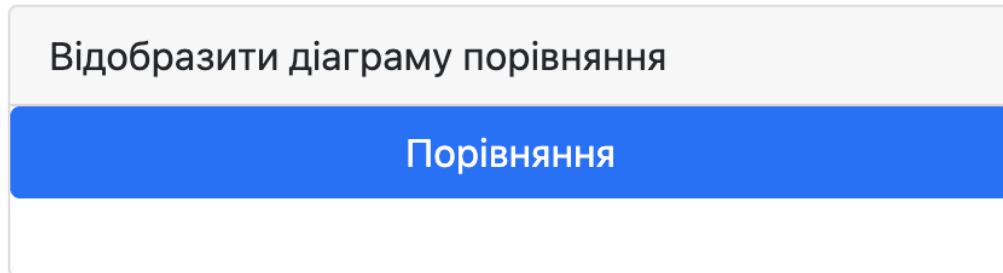


Рисунок 3.6 – Вікно з кнопкою для відображення діаграми порівняння значень оптимізації використовуючи різні методи

На рисунку 3.6 наведена оптимізована топологія комп'ютерної мережі CALS-системи після використання методу покоординатного спуску.

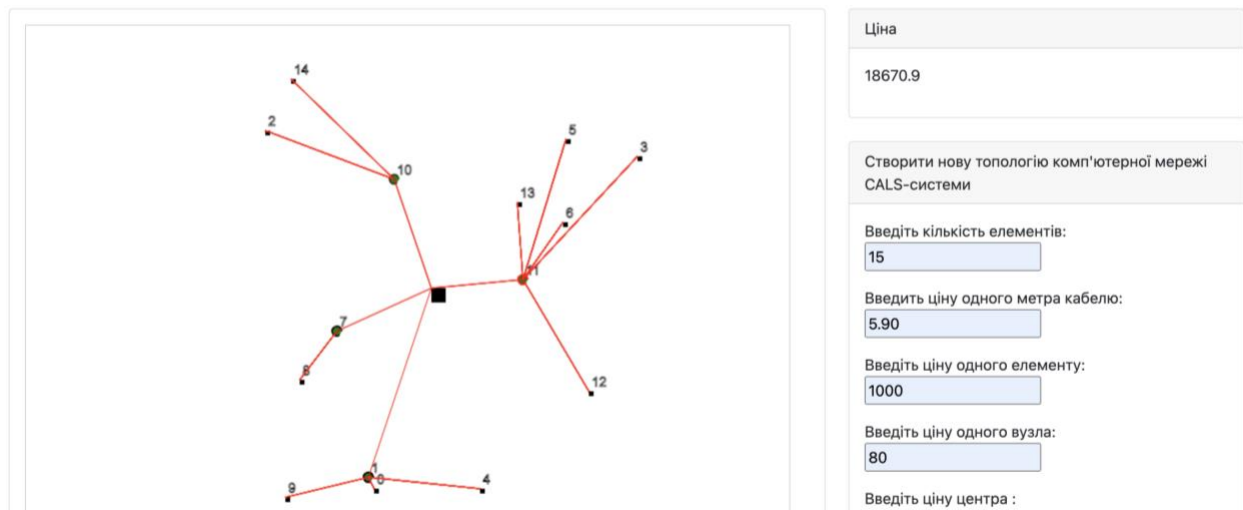


Рисунок 3.6 – Результати оптимізації топології комп'ютерної мережі CALS-системи з використанням методу покоординатного спуску

Тепер оптимізуємо топологію використовуючи метод покоординатного спуску з функцією «Multi Start». Результати оптимізації наведені на рисунку 3.7.

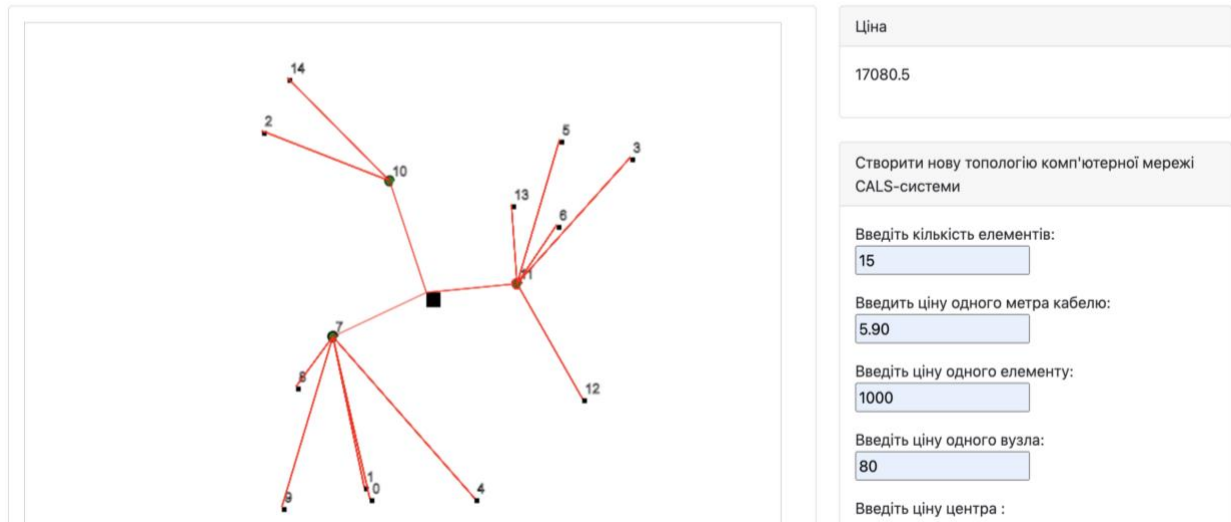


Рисунок 3.7 – Результати оптимізації топології комп'ютерної мережі CALS-системи з використанням модифікованого методу

Діаграма порівняння значень критерію оптимізації комп'ютерної мережі CALS-системи наведена на рисунку 3.8.



Рисунок 3.8 – Діаграма порівняння значень критерію оптимізації комп'ютерної мережі CALS-системи

### 3.4 Огляд результатів експерименту та його аналіз

Проведемо 20 експериментів, використовуючи описаний вище web-додаток для того, щоб порівняти ефективність методу покоординатного спуску та запропонованої оптимізації цього метода з використанням функції «Multi Start». Ціни на компоненти для створення топології комп'ютерної мережі CALS-системи в

залежності від номера експерименту наведено на таблиці 3.2.

Таблиця 3.2 – Цінові характеристик компонент мережі в експериментах

Номер експерименту	Кількість елементів	Ціна елемента, \$	Ціна вузла, \$	Ціна кабелю, \$/м	Ціна центрального вузла, \$
1	20	1000	80	5	3000
2	25	700	70	4,8	3200
3	30	750	75,8	3	3500
4	35	1100	60	3,4	2700
5	40	890	90	4,5	4000
6	45	910	95	3,3	4300
7	90	700	120	3,1	5100
8	50	830	89,99	2,8	4100
9	98	650	125,8	3,8	5200
10	15	560	50	2,3	2600
11	95	1250	140	4,7	5500
12	80	990	79	3,8	4900
13	23	810	67	2,4	2300
14	120	740	136	3,7	5200
15	150	810	140	2,8	5400
16	36	560	52	3,1	3200
17	49	738	63	2,9	3900
18	19	720	35	3,1	2600
19	210	820	148	4,3	5900
20	79	928	97	4,8	3980

У таблиці 3.3 наведені результати 20-ти експериментів з використанням

методу покоординатного спуску та запропонованої модифікації на основі процедури «Multi Start».

Таблиця 3.3 – Результати експериментів

Вартість мережі, \$			
Номер експерименту	Метод покоординатного спуску	Метод на основі «Мультистарт»	Базова топологія
1	20991,5	21432,1	26477,4
2	20619,7	19548,2	24469,5
3	24660,6	22247,9	29129,3
4	19928,9	17300,0	22424,5
5	40497,4	36898,5	43095,7
6	45853,0	44429,6	48295,0
7	68832,2	66397,5	71246,2
8	46136,4	44774,4	48552,5
9	69719,0	68198,6	70304,1
10	11380,3	9219,8	11921,3
11	125404,4	119810,3	126388,3
12	84657,2	83066,7	85454,7
13	21348,1	21080,5	22066,0
14	94935,7	94047,4	95793,4
15	125492,9	12174,2	129290,3
16	23858,6	21031,3	25888,0
17	40418,8	39738,0	42253,2
18	16812,0	15949,3	18646,7
19	168928,9	172128,6	180801,4
20	78338,8	76196,4	79838,8

Для побудови топології комп'ютерної мережі CALS-системи використовувалися вхідні дані, які наведені у таблиці 3.2 для кожного з експериментів.

Виходячи з результатів експерименту, отриманих у таблиці 3.3, за формулою (3.1) знайдемо середнє арифметичне значення ціни базової топології мережі, та ціни отриманої після проведення оптимізації

$$\bar{\chi} = \frac{\chi_1 + \chi_2 + \chi_3 + \dots + \chi_n}{n}. \quad (3.1)$$

Підставивши у формулу (3.1) значення, які були отримані в ході проведення експериментів (таблиця 3.3), отримаємо середнє арифметичне значення оцінки вартості комп'ютерної мережі CALS-системи для базової топології та після використання двох методів оптимізації. Результати занесемо до таблиці 3.4.

Таблиця 3.4 – Середнє арифметичне значення результатів експерименту

Вартість мережі, \$		
Метод покоординатного спуску	Метод на основі «Мультистарт»	Базова топологія
57540,72	51952,3	60116,815

Тепер, за допомогою засобів web-додатку, описаного вище, побудуємо діаграму порівняння значень оптимізації, використовуючи різні методи (рисунок 3.9).

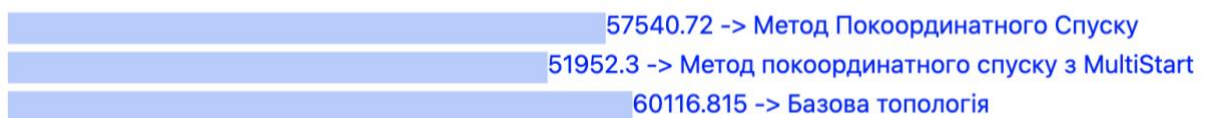


Рисунок 3.9 – Діаграма порівняння значень оптимізації комп'ютерної мережі

### 3.5 Висновки до третього розділу

За результатами аналізу особливостей сучасних мов програмування та середовищ розробки для розв'язання задачі обрано мову програмування Java та середовище розробки IntelliJ IDEA. На їх основі була здійснена розробка кросплатформеного програмного забезпечення задачі оптимізації топологічних структур комп'ютерних мереж.

З метою оцінки ефективності запропонованої модифікації методу з використанням розробленого програмного забезпечення було проведено 20 експериментів з оптимізації комп'ютерних мереж з різною кількістю і вартістю їх елементів.

За результатами експериментів встановлено, що запропонована модифікація методу дозволяє отримувати більш оптимальні рішення. Це дозволить на практиці отримувати варіанти побудови мереж значно меншої наведеної вартості.

## 4 ОХОРОНА ПРАЦІ

### 4.1 Загальні положення

Розглянемо питання безпечної життєдіяльності в лабораторії Інформаційних технологій. У залежності від характеру технологічних процесів, що розробляються, лабораторія може бути розташована поблизу виробництва з підвищеним рівнем небезпеки. Людина, яка працює у такій лабораторії, повинна мати базові знання у програмуванні, створенні та експлуатації комп'ютерних мереж CALS-систем, роботи з CALS-системами тощо.

У базовому своєму розумінні охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини під час трудової діяльності [27].

До виконання робіт, можуть допускатися тільки ті інженери, які пройшли відповідні навчання, відвідували семінари та стажування щодо виконання робіт на основі відповідних приладів, правила поведінки у подібних лабораторія тощо. Також важливим фактором є розуміння правил поведінки при аварійному стані приборів та надання першої допомоги потерпілим від нещасних випадків відповідно до вимог положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 № 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за № 231/10511 (НПАОП 0.00-4.12-05) [28].

### 4.2 Вимоги до фахівця під час роботи

До основних правил поведінки програміста у лабораторії забороняється робити наступні дії [29]:

- торкатися екрану ПК руками. Для виконання подібних дій, видається спеціальна указка;
  - сидіти криво або на своїх ногах. Перед робочою машиною необхідно сидіти рівно, спираючись на спинку стільця;
  - близько наблизитися при роботі з ПК до монітору. Необхідно дотримуватися безпечної дистанції, яка становить від 50 см до 60 см;
  - торкатися проводів та кабелів руками, самостійно приєднувати та відключати їх. Якщо виникають відповідні сторонні проблеми при роботі з ПК, слід покликати спеціальну людину, яка навчена вирішувати подібні проблеми;
  - приймати їжу та пити при роботі з ПК. При необхідності зробити відповідні дії, слід зробити перерву та відійти до спеціального приміщення;
  - довго працювати за робочою машиною та перевтомлювати організм;
- Програміст зобов'язаний робити:
- не розголошувати деталі роботи без відповідного указу чи дозволу керівництва;
  - працювати згідно з встановленими нормами відповідно конвенція Міжнародної організації праці від 22.06.1935 р. № 47 «Про скорочення робочого часу до сорока годин на тиждень». Програміст повинен працювати 40 годин на тиждень з одно годинним щоденним переривом;
  - у разі виникнення певних проблем зі здоров'ям зупинити роботу та зробити перерив або відвідати лікаря.

#### 4.3 Вимоги до приміщення

Приміщення лабораторії повинно мати припливно-витяжну вентиляцію і бути забезпеченим первинними засобами пожежогасіння; зокрема вуглекислотними вогнегасниками, ковдрою із негорючих матеріалів, ящиком або відром з піском і совком.

Приміщення лабораторії повинно бути обладнано витяжними шафами з верхніми та нижніми відсмоктувачами та бортиками, що попереджають стікання рідини на підлогу. В приміщеннях лабораторій, окрім загальнообмінної припливно-витяжної та місцевої витяжної вентиляції від лабораторних шаф, повинні бути пристрої для природного провітрювання (кватирки, фрамуги). Швидкість руху повітря за умови відчинення стулок витяжних шаф та працюючій вентиляції повинна бути не менше 0,5 м/с, а за умов наявності поблизу приміщень з їдкими, отруйними та з речовинами з неприємним запахом – не менше 1,0 м/с.

Усі матеріали й речовини повинні зберігатись відповідно до правил пожежної безпеки.

Лабораторія повинна бути забезпечена аптечкою з набором медикаментів і перев'язувальних засобів.

При роботі у лабораторії, фахівець повинен знаходитися у максимально безпечному та комфортному стані без зайвих фізичних навантажень. Його необхідно забезпечити зручним робочим місцем, до якого входить стіл, стілець, необхідна апаратура тощо.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні устаткування, що використовується. Висота робочої поверхні столу повинна становити 725 мм, робоча поверхня стола повинна мати ширину 800...1400 мм і глибину 800...1000 мм. Робочий стіл повинен мати простір для ніг заввишки щонайменше 600 мм, шириною — щонайменше 500 мм, глибиною лише на рівні колін — щонайменше 450 мм і лише на рівні витягнутих ніг — щонайменше 650 мм.

При розміщенні робочих місць відстань між робочими столами має бути не менше 2,0 м, а відстань між бічними поверхнями моніторів не менше 1,2 м.

Також слід правильно налаштувати світ штучне світло. Рівень загального штучного освітлення приміщення слід розрахувати на основі формули (4.1)

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де  $W$  – питома потужність, Вт/м<sup>2</sup>;

$S$  – площа приміщення, м<sup>2</sup>;

$W_{\Sigma}$  – загальна потужність освітлювальної установки, Вт.

#### 4.3 Основні вимоги безпеки перед початком робіт

До основних вимог, які необхідно зробити перед початком робіт, відноситься наступне:

- отримати дозвіл на виконання роботи. Перевірити стан та, за необхідності, наприклад в умовах COVID, одягти спецодяг, засоби індивідуального захисту;
- включити припливно-витяжну вентиляцію за 10-15 хв до початку роботи. Перевірити справність приладів, обладнання, наявність необхідних засобів захисту. У разі необхідності включити вентиляцію у витяжній шафі;
- перевірити обладнання на працездатність. У разі виявлення несправностей стосовно обладнання та засобів захисту, сповістити відповідальну особу та не приступати до роботи до усунення виявлених несправностей.

#### 4.4 Вимоги після закінчення робіт

Після закінчення роботи програміст повинен [33]:

- відключити всі прилади від електромережі;
- перевірити робоче місце за яким здійснювалася робота;
- вжити необхідні заходи особистої гігієни, а саме вимити та обробити руки, зняти спеціальний одяг та по можливості прийняти душ;

- зняти та привести до нормального стану спеціальний одяг. Якщо одяг був пошкоджений або заплямований після використання звернутися до відповідної людини та обговорити можливі заходи для усунення проблеми;
- якщо фахівець виходить останнім з приміщення лабораторії, закрити всі вікна, вимкнути світло та перевірити приміщення.

#### 4.5 Висновки до четвертого розділу

Проведено аналіз загальних положень, основні обов'язки фахівців під час роботи з розробленою програмною, вимоги після закінчення роботи тощо. Детально розглянуто вимоги щодо приміщення лабораторії. До основних положень рекомендується віднести встановлення необхідної системи вентиляції, правильного налаштувати штучного освітлення, розподілення робочих станцій, іншого обладнання.

Всіх положень та вимог, які були визначені у цьому розділі, необхідно дотримуватися для забезпечення безпечної та ефективної роботи персоналу лабораторії.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було отримано рішення важливої науково-прикладної задачі підвищення ефективності функціонування комп'ютерних мереж CALS-систем за рахунок оптимізації їх топологічних структур.

За результатами огляду й аналізу сучасного стану питання впровадження CALS-систем і технологій встановлено, що вони знаходять усе більш широке в інтегрованих виробничих компаніях. Інфраструктуру CALS-систем складають корпоративні комп'ютерні мережі, які забезпечують взаємодію між всією множиною користувачів, що забезпечують неперервний процес проєктування виробу, проєктування технологічних процесів його виготовлення, систем керування технологічними процесами тощо.

Це обумовлює актуальність завдань оптимізації комп'ютерних мереж CALS-систем, вирішення яких дозволяє з меншими наведеними витратами реалізувати швидкий та надійний обмін інформацією між користувачами з мінімальними наведеними витратами. На цій основі сформульовано постановку мети та задач дослідження, що передбачає, зокрема, удосконалення обраного базового методу розв'язання задачі та його програмну реалізацію.

Виходячи з мети та завдань дослідження, розроблено математичні та програмні компоненти технології оптимізації комп'ютерної мережі CALS-системи. Зокрема, запропоновано удосконалений алгоритм методу спрямованого перебору з покоординатною оптимізацією розміщення вузлів мережі на основі технології «Мультистарт». Для методів розроблено схеми алгоритмів та здійснено їх програмну реалізацію мовою Java. Було розроблено web-додаток, який надає можливість будувати та аналізувати варіанти побудови комп'ютерних мереж, використовуючи метод покоординатного спуску та його покращення з використанням функції «Multi

Start».

Використовуючи засоби розробленого web-додатку, було проведено серію експериментів, які підтвердили ефективність запропонованої модифікації базового методу.

Розроблені компоненти технології оптимізації можуть бути використані на підприємствах і в організаціях, де вирішуються задачі проектування чи реінжинірингу комп'ютерних мереж. Їхнє практичне використання дозволить за рахунок підвищення точності розв'язання задач скорочувати наведені витрати на створення й експлуатацію комп'ютерних мереж.

Результати кваліфікаційної роботи апробовані на 3-х наукових конференціях [1, 30 – 31].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Смірнов М.Є. Розроблення засобу оптимізації комп'ютерної мережі CALS-систем // Діджиталізація науки як виклик сьогодні : матеріали наук.-практ. конф. з міжнар. участю (м. Одеса, 21 грудня. 2021 р.). Одеса, 2021. С. 29-30.
2. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.
3. Y.Kravchenko, V. Taratoukhine, V. Kureichik. Continuous Acquisition and Life-Cycle Support (CALS) Simulation Models on the Basis of the ERP and CAD Technologies Integration. 2018, 540 p.
4. Product Data Management (PDM). URL: <https://www.techtarget.com/searcherp/definition/product-data-management-PDM> (дата звернення: 21.09.2022).
5. Capacity Requirements Planning (CRP). URL: <https://www.investopedia.com/terms/c/capacity-requirements-planning.asp#:~:text=Capacity%20requirements%20planning%20is%20the,to%20meet%20its%20production%20goals>. (дата звернення: 22.09.2022).
6. What Is Material Requirements Planning (MRP)?. URL: [https://www.netsuite.com/portal/resource/articles/inventory-management/material-requirements-planning-mrp.shtml#:~:text=A%20Material%20Requirements%20Planning%20\(MRP,bill%20of%20materials%20\(BOM\)](https://www.netsuite.com/portal/resource/articles/inventory-management/material-requirements-planning-mrp.shtml#:~:text=A%20Material%20Requirements%20Planning%20(MRP,bill%20of%20materials%20(BOM)). (дата звернення: 23.09.2022)
7. What is Computer Networking?. URL: [https://aws.amazon.com/what-is/computer-networking/?nc1=h\\_ls](https://aws.amazon.com/what-is/computer-networking/?nc1=h_ls) (дата звернення: 25.09.2022)
8. Ross K. Computer Networking: A Top-Down Approach. London 2020, 864 p.
9. Layers of OSI Model. URL: <https://www.geeksforgeeks.org/layers-of-osi-model/> (дата звернення: 25.09.2022)
10. Exposito E. Advanced Transport Protocols: Designing the Next Generation.

London. 2018, 304 p.

11. D. Wetherall, A. Tanenbaum. Wetherall D. Computer Networks. London, 2020, 780 p.

12. Anthony Gratton. Personal Area Networking Technologies and Protocols // Cambridge University Press : електрон. версія журн. 2019. С. 15-22. URL : <https://doi.org/10.1017/CBO9780511979132> (дата звернення: 27.09.2022)

13. Gary Donahue, Denise Donohue. Art of Network Architecture. New York, 2019, 530 p.

14. Gary Donahue, Denise Donohue. Network Warrior: Everything You Need to Know That Wasn't on the CCNA Exam. London, 2020, 620 p.

15. Sedgewick R. Algorithms in Java: Fundamentals Data Structures Sorting Searching. Berlin, 2019, 1320 p.

16. Hobson M. Mathematical Methods for Physics and Engineering: A Comprehensive Guide. London, 2019, 1540 p.

17. A Guide to Network Optimization. URL: <https://www.sunnyvalley.io/docs/network-basics/network-optimization> (дата звернення: 26.10.2022).

18. Newton O. Everything You Need to Ace Math in One Big Fat Notebook: The Complete Middle School Study Guide. Deggendorf, 2019, 1510 p.

19. Бескорвайный В. В., Иванова В.А. Модификация метода топологической оптимизации системы мониторинга производственного процесса // Технология приборостроения. 2017. №2. С. 14–17.

20. Нефьодов Л. І., Невлюдов І. Ш., Безкорвайний В. В. CALS технології і системи : навч. посіб. Харків: ВПЦ “Харків. нац. у-т радіоелектроніки”, 2021. 270 с.

21. Бескорвайный В. В., Авраменко В. П. Евристичний метод структурного синтезу територіально розподілених систем // Технологія приладобудування. 2018. С. 69–74.

22. Top 10 Java Compilers. URL: <https://uk.myservername.com/top-10-best-java->

- ides-online-java-compilers#Working\_Principle\_Of\_IDE (дата звернення: 25.10.2022).
23. Poornachandra S Java Programming. Gamburg, 2018, 980 p.
  24. Jeanne Boyarsky, Scott Selikoff. OCP Oracle Certified Professional Java SE 11 Programmer I Study Guide: Exam 1Z0-815. London, 2020, 870 p.
  25. Horstmann C. Core Java: Fundamentals. USA, 2020, 1640 p.
  26. Top 10 Java IDE. URL : <https://www.helenjoscott.com/tag/intellij-idea/> (дата звернення: 09.11.2022).
  27. Jet Brains. IntelliJ IDEA. URL: <https://www.jetbrains.com/idea/> (дата звернення: 12.11.2022).
  28. Охорона праці. URL: <https://oppb.com.ua/news/chto-takoe-ohrana-truda/> (дата звернення: 12.11.2022).
  29. ДЕРЖАВНИЙ КОМІТЕТ УКРАЇНИ З НАГЛЯДУ ЗА ОХОРОНОЮ ПРАЦІ. URL: <https://zakon.rada.gov.ua/laws/show/z0231-05> (дата звернення: 12.11.2022).
  30. Особливості нормування роботи програмістів. URL: <https://studfile.net/preview/2401216/page:4/> (дата звернення: 12.11.2022).
  31. Смірнов М., Безкоровайний В. Оптимізація топологічних структур комп'ютерних мереж CALS-систем // Всеукраїнська наук.-практ. конф. здобувачів вищої освіти і молодих учених «Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві» (м. Харків, 23 листоп. 2022 р.), Харків: ХНАДУ. URL: [https://mf.khadi.kharkov.ua/fileadmin/user\\_upload/\\_imported/uploads/%D0%9F%D0%A0%D0%9E%D0%93%D0%A0%D0%90%D0%9C%D0%90\\_%D0%90%D0%9A%D0%86%D0%A2\\_2022.pdf](https://mf.khadi.kharkov.ua/fileadmin/user_upload/_imported/uploads/%D0%9F%D0%A0%D0%9E%D0%93%D0%A0%D0%90%D0%9C%D0%90_%D0%90%D0%9A%D0%86%D0%A2_2022.pdf) (дата звернення 03.12.2022).
  32. Смірнов М., Безкоровайний В. Структурно-топологічна оптимізація комп'ютерних мереж CALS-систем // Інформаційні системи та технології: матеріали 11-ї Міжнародної наук.-техн. конф. Ч. 2 (м. Харків, 22-25 листоп. 2022 р.), Х.: ХНУРЕ, 2022. С. 7–8. URL: <https://istconf.sedep.pdf> (дата звернення 03.12.2022).