

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Запорізька політехніка"

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)

Кафедра радіотехніки та телекомунікацій
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
магістра
(ступінь вищої освіти)

на тему **РОЗРОБКА ТА ПРОЕКТУВАННЯ ЛОКАЛЬНОЇ**
ОБЧИСЛЮВАЛЬНОЇ МЕРЕЖІ

Виконав(ла): студент(ка) 2м курсу, групи БК-922м

Спеціальності _____

172 «Телекомунікації та радіотехніка»
(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Інформаційні мережі зв'язку»

АНОШИН Валерій Валерійович
(ПРИЗВИЩЕ та ініціали)

Керівник КАБАК Владислав Семенович
МОРОЗ Гаррі Володимирович
(ПРИЗВИЩЕ та ініціали)

Рецензент _____
(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Запорізька політехніка"
(повне найменування закладу вищої освіти)

Факультет Інформаційної безпеки та електронних комунікацій
 Кафедра Радіотехніки та телекомунікацій
 Ступінь вищої освіти Магістр
 Спеціальність 172 "Телекомунікації та радіотехніка"
(код і найменування)
 Освітня програма (спеціалізація) Інформаційні мережі зв'язку
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри РТТ
к.ф.-м.н., доц. САМОЙЛИК С.С.
 « » січня 2024 року

С.С.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА)

АНОШИНУ Валерію Валерійовичу

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проекту (роботи): Розробка та проектування локальної обчислювальної мережі

керівник проекту (роботи) к.т.н., доц. КАБАК Владислав Семенович

ст. викладач МОРОЗ Гаррі Володимирович

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

затверджені наказом закладу вищої освіти від "14" листопада 2023 року №443

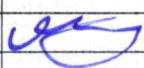
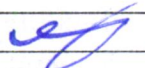
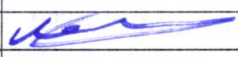
2. Строк подання студентом проекту (роботи) 10 грудня 2023 р.

3. Вихідні дані до проекту (роботи) Побудовано модель формування й обробки сигналів і перевірена її адекватність на задачі оцінювання імовірнісних характеристик оптимального некогерентного виявлювача.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Для вирішення задачі проектування мережі WLAN було використано ітеративний процес. В кожному з розділі представлено емпіричний аналіз моделі, а також запропоновано програмну реалізацію для відповідної задачі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) презентація роботи в Microsoft PowerPoint

6. Консультанти розділів проекту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
I-4	КАБАК В.С., к.т.н., доцент		
	МОРОЗ Г.В., ст. викладач		
Нормо-контроль	МОРОЗ Г.В., ст. викладач		

7. Дата видачі завдання «04» вересня 2023 року.

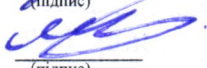
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області	4.09-1.10	виконано
2	Проблема проектування фіксованої бездротової локальної мережі	2.10-14.10	виконано
3	Багатоканальна фіксована бездротова локальна мережа: проблема проектування	14.10-24.10	виконано
4	Проблема проектування кучової безпроводної локальної мережі	24.10-2.11	виконано
	Оформлення пояснювальної записки	2.11-2.12	виконано
3	Створення презентації у Power Point	10.12	виконано

Студент(ка)

Керівник проекту (роботи)


(підпис)


(підпис)

Валерій АНОШИН

(ім'я, ПРИЗВИЩЕ)

Гаррі МОРОЗ

(ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до магістерської роботи: 95 стр., 17 табл., 12 рис., 23 джерела.

ГНСС, МЕАНДРОВИЙ СИГНАЛ, ПСЕВДОВИПАДКОВА ПОСЛІДОВНІСТЬ, ЩІЛЬНІСТЬ РОЗПОДІЛУ, ВІДНОШЕННЯ СИГНАЛ/ШУМ.

Об'єкт проектування – бездротова локальна мережа.

Мета роботи – представити моделі, алгоритми розв'язання та емпіричний аналіз, які можуть бути використані для отримання якісного проекту для різних версій задачі проектування WLAN. В рамках цього дослідження ми оцінюємо програмне забезпечення, доступне на сервері NEOS (NEOS, 2010), комерційне програмне забезпечення від систем LINDO. Оскільки обмеження і цілі проблеми проектування WLAN.

Метод дослідження – теоретично-розрахунковий, з використанням ЕОМ.

Побудовано модель формування й обробки сигналів і перевірена її адекватність на задачі оцінювання імовірнісних характеристик оптимального некогерентного виявлювача.

За виконання дипломної роботи застосовувалися сучасні методи проектування. Проектування здійснювалося на спеціальному програмному забезпеченні й обчислювальній техніці.

ЗМІСТ

	С.
Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз предметної області.....	8
1.1 Бездротові локальні мережі.....	8
1.2 Поточний стан досліджень.....	10
1.3 Мета дослідження	22
2 Проблема проектування фіксованої бездротової локальної мережі.....	24
2.1 Оптимізаційна модель та алгоритми розв'язання	24
2.2 Проект для офісної будівлі.....	39
2.3 Підсумки та висновки	41
3 Багатоканальна фіксована бездротова локальна мережа: проблема проектування.....	43
3.1 Оптимізаційні моделі.....	43
3.2 Алгоритми та програмні реалізації	47
4 Проблема проектування кучової безпроводної локальної мережі	59
4.1 Оптимізаційні моделі.....	59
4.2 Алгоритми та програмні реалізації	65
4.3 Емпіричний аналіз.....	66
4.4 Підсумки та висновки	69
Висновки	70
Додаток А.....	74
Додаток Б	76
Додаток В.....	79
Перелік джерел посилань	93

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AP	–	точка доступу
MD	–	мобільний пристрій
WLAN	–	бездротові локальні мережі

ВСТУП

Бездротові локальні мережі (WLAN) забезпечують високошвидкісний доступ до даних у будинках, офісах, аеропортах, кав'ярнях і бібліотеках. Деякі міста навіть розробили широкомасштабні мережі для використання своїми громадянами. Крім того, навіть спортивні команди, проектують свої нові стадіони з урахуванням можливості доступу до бездротових мереж [1]. Швидкість передачі даних в бездротових мережах постійно зростала протягом останнього десятиліття, що дозволило використовувати багато нових додатків, деякі з яких продаються в таких різноманітних місцях, як магазини додатків Apple, Android і Amazon.

Проблему проектування WLAN можна визначити наступним чином: "Маючи набір мобільних терміналів з відомим або невідомим розташуванням і бюджетне обмеження на кількість точок доступу, що встановлюються, визначити комбінацію розташування і частотного призначення для кожної точки доступу, яка забезпечить максимальну пропускну здатність мережі".

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Бездротові локальні мережі

Бездротові локальні мережі з'єднують два або більше комп'ютерів чи пристроїв за допомогою технологій розширення спектру або OFDM-модуляції, які дозволяють здійснювати зв'язок на обмеженій території. Зв'язок між пристроями в WLAN зазвичай вимагає обмеженої мобільності та роумінгу, а також використання неліцензованого, безкоштовного промислового, наукового та медичного (ISM) діапазону. Це на відміну від стільникових мереж, які підтримують високу мобільність абонентів і працюють в ліцензованих діапазонах спектра з набагато вищими рівнями потужності передачі. Використання ресурсів ISM-спектру при низькій потужності передачі означає, що технології WLAN повинні демонструвати стійкість до помилок і витримувати завади від інших пристроїв в тих же діапазонах спектру, наприклад, бездротових телефонів, мікрохвильових печей і відкривачів гаражних воріт WLAN засновані на стандарті IEEE 802.11, які визначають набір частот, що можуть бути використані для побудови цих бездротових мереж. Найпоширенішим протоколом є IEEE 802.11 b/g, який визначає 11 каналів у діапазоні 2,4 ГГц, як показано в таблиці 1.1. Оскільки сигнал займає приблизно 22 МГц спектра, це дає 3 канали, що не перекриваються (1, 6 і 11). Багато європейських країн дозволяють використання двох додаткових каналів і використовують обладнання, яке дозволяє використовувати чотири канали, що не перекриваються (1, 5, 9 і 13).

Таблиця 1.1 – Канали IEEE 802.11 b/g

Частота Спектр (ГГц)	Канал	Частота Спектр (ГГц)	Канал
2.412	1	2.442	7
2.417	2	2.447	8
2.422	3	2.452	9
2.427	4	2.457	10
2.432	5	2.462	11
2.437	6	—	—

Структура бездротової локальної мережі складається з мобільних пристроїв (MD), які взаємодіють з точками доступу (AP). Бездротова точка доступу забезпечує зв'язок між пристроями, підключеними до бездротової мережі, і дротовою мережею, як правило, на основі Ethernet.

Основними рушійними факторами зростаючої популярності WLAN є доступність недорогого обладнання, простота і швидкість установки в різних умовах, а також ефективне і раціональне використання ресурсів спектра ISM. Простота і швидкість встановлення бездротових точок доступу пояснюється, серед іншого, тим, що більшість розгортань використовують системи Power Over Ethernet (PoE) для безпечної передачі електроенергії разом з даними по стандартному кабелю CAT5 в мережі Ethernet.

Сучасні стандарти PoE (IEEE 802.3af і 802.3at) дозволяють розподіляти до 25 Вт потужності на кожну точку доступу. Це значно спрощує встановлення та зменшує витрати, оскільки для розгортання нової точки доступу потрібен лише один тип кабелю (Ethernet CAT5) без проведення окремих електромонтажних робіт та отримання дозволів. У той час як для створення бездротової локальної мережі в житловому середовищі достатньо встановити одну або дві точки доступу, для бездротових мереж, призначених для корпоративного або університетського кампусу, може знадобитися

набагато більше точок доступу. Україна має приклади розгорнутих міських мереж WLAN.

Наприклад, в місті Львові була запущена міська громадська Wi-Fi мережа, яка забезпечує мешканців та відвідувачів міста безкоштовним доступом до Інтернету. Ця мережа створена для зручності місцевих мешканців, туристів та бізнесменів, які відвідують місто. Проте, схоже на американський приклад, в містах України можуть виникати експлуатаційні витрати, пов'язані з розвитком та підтримкою міських мереж WLAN. Ці витрати можуть обмежувати розгортання та розвиток таких мереж, але все ж можуть бути ефективними для забезпечення доступу до Інтернету у громадських місцях.

Основна проблема проектування бездротових локальних мереж включає в себе рішення щодо кількості та розташування радіопередавачів/приймачів (AP), частоти, яка буде призначена кожному AP, і потужності сигналу кожного AP. Багато організацій встановлюють WLAN як альтернативу кабельним мережам, не витрачаючи багато часу на планування і проектування.

Деякі просто розміщують точки доступу в сітці, вмикають їх і додають нові в міру виникнення проблем.

Однак ретельне планування забезпечує високу продуктивність при мінімальних витратах.

1.2 Поточний стан досліджень

Фахівці, які відповідають за розгортання бездротових локальних мереж, зазвичай використовують для визначення місця розташування точок доступу або обстеження об'єкта, або програмні інструменти. Деякі з них можуть використовувати комбінацію цих двох стратегій, а також свої загальні знання про середовище, в якому буде працювати мережа WLAN.

Обстеження об'єкта використовується тоді, коли проектувальник має ресурси для емпіричного тестування декількох різних потенційних схем WLAN. Програмні інструменти використовуються, коли ресурси обмежені, але вони не такі точні, як обстеження на місці. Існує кілька різних моделей, які використовуються при створенні програмних засобів для проектування WLAN. Моделі на основі покриття зосереджені на забезпеченні.

Точки доступу достатньо близько, щоб асоціюватися з кожним мобільним пристроєм (MD), і не враховують інтерференцію між точками доступу. Моделі на основі пропускної здатності намагаються призначити місця і частоти для кожної точки доступу таким чином, щоб максимізувати загальну пропускну здатність мережі, яка може бути змодельована кількома різними способами. Ми використовуємо виключно пропускну здатність каналу Шеннона [2]; однак існують й інші математичні моделі пропускної здатності мережі, і вони вимагають інших інструментів оптимізації.

За останні п'ятнадцять років з'явилося багато опублікованих рукописів, що стосуються різних стратегій для цієї важливої проблеми дизайну. Крім того, існує кілька власних програмних засобів, розроблених для комерційного використання. Деякі з попередніх робіт використовують алгоритми оптимізації, а інші – спеціально розроблені евристичні методи.

1.2.1 Дослідження сайту

В [3] описано проект публічної бездротової локальної мережі, що охоплює міжнародний аеропорт Майамі. Точка доступу була встановлена на пересувному візку, прикріпленому до верхньої частини телескопічної опори. Це дозволило Гейеру і його колегам протестувати різні місця на стелі. На другому візку розмістили ноутбуки та обладнання для визначення рівня сигналу. За допомогою цього обладнання вони змогли обстежити аеропорт і

розробити конструкцію з хорошими характеристиками покриття.

В [4] звітують про свій досвід використання обстеження території для розміщення трьох точок доступу на двох поверхах академічної будівлі, де розташовані лабораторії та офіси викладачів. Автори вибрали шість потенційних місць для встановлення точок доступу, щоб покрити 253 зони попиту. Дані про рівень сигналу були зібрані в кожній з 253 точок попиту шляхом повільного обертання мобільного приймача в кожній точці. Рівень сигналу чутливий до орієнтації мобільного телефону, тому найбільші 20% і найменші 20% значень були відкинуті. Середнє значення з решти 60% було використано в моделі цілочисельного програмування для визначення оптимального розташування трьох точок доступу. Використання трьох точок доступу є бажаним, оскільки кожній з них можна призначити унікальну частоту і таким чином уникнути втручання.

Огляди об'єктів можуть бути корисними для розробки проекту, орієнтованого на покриття для районів з низькою щільністю, але можуть не дати хорошого проекту, орієнтованого на пропускну спроможність для районів з високою щільністю. Наше дослідження спрямоване на проблему проєктування в районах з високою щільністю населення. Тобто, вимірювання пропускну здатності не є частиною обстеження ділянки, але можуть бути включені в програмні інструменти. Деякі клієнти вважають, що обстеження ділянок є досить дорогим.

1.2.2 Моделі покриття

Якщо метою є покриття, а поширення сигналу може бути визначено на множині місць розташування MD(мобільний пристрій), то для вибору мінімальної кількості місць розташування точок доступу з множини можливих місць можна використовувати просту модель цілочисельного програмування. Нехай S позначає множину потенційних місць розташування точок AP(точка доступу) а M – множину місць розташування MD Нехай $C_m \in$

Спозначає множину точок доступу в межах досяжності MD m . Тоді модель покриття є простою:

Мінімізувати $P_s \in S$ w_s за умови:

$P_{ws} \geq 1 \quad s \in C_m$	$\forall m \in M$
$w_s \in \{0,1\}$	$\forall s \in S,$

де w_s вказує на те, чи встановлена точка доступу в потенційному місці s . Оптимальне рішення гарантує покриття, але не враховує пропускну здатність мережі.

Ця модель була використана для покриття квадратної ділянки розміром 250 метрів, як показано на рисунку 1.1. Кожен квадрат сітки має розмір 50 на 50 метрів MD розміщені в 36 точках сітки, а потенційні місця для розміщення точок доступу – в 16 внутрішніх точках сітки. Для ефективного зв'язку відстань між точками доступу та MD не повинна перевищувати 120 метрів.

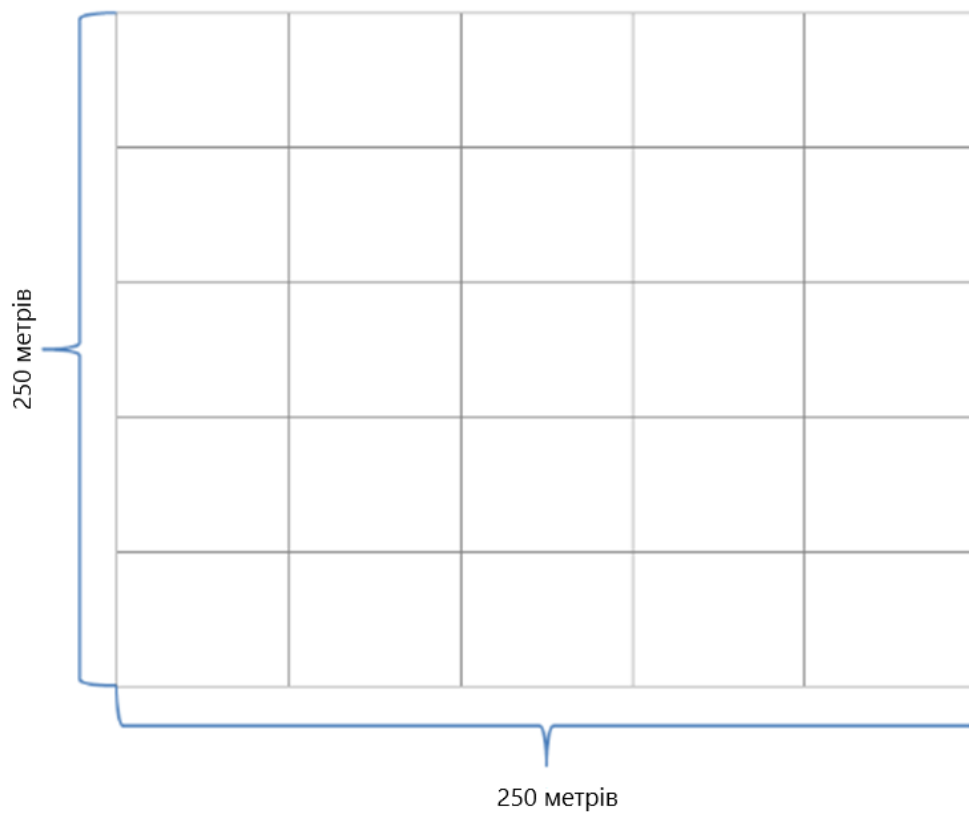


Рисунок 1.1 – Ділянка з накладеною сіткою

Оптимальне рішення передбачає розміщення чотирьох точок доступу в наступних точках сітки: (50,50), (150,50), (50,150), (200,150), як показано на рисунку 1.2. Точки сітки в основному покриті, але є невеликі ділянки біля точок (125,250) і (250,0), які не покриті. Для усунення цих "мертвих зон" можна було б використати більш дрібну сітку.

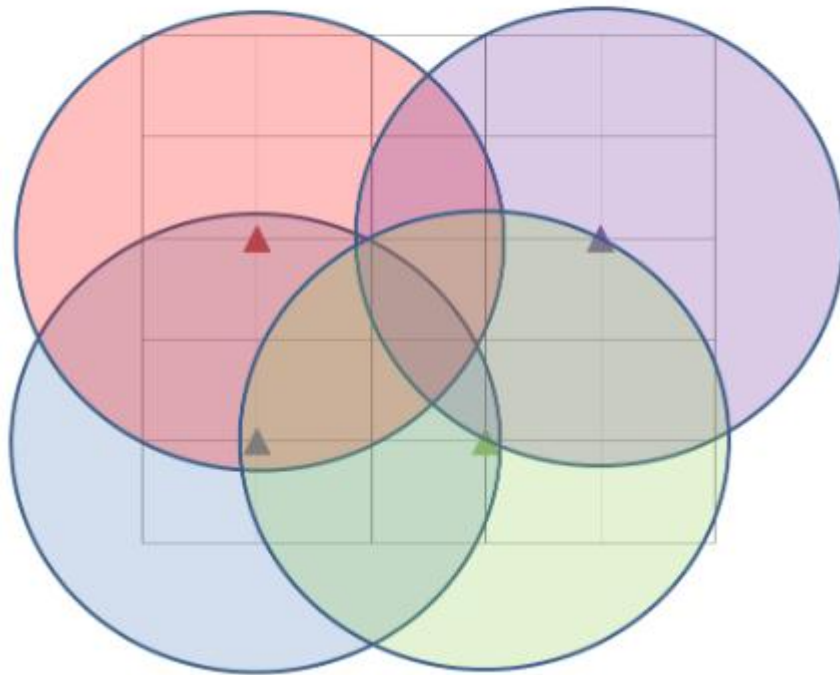


Рисунок 1.2 – Штрафний термін може бути визначений для мобільних телефонів у точках мережі, де зв'язок неможливий

Більш детальну модель, орієнтовану на покриття, розробили Бахрі та Чемберленд [5]. Вони розробили оптимізаційну модель для розміщення точок доступу, потужності передачі та розподілу частот. Метою є мінімізація необхідної загальної кількості точок доступу, які необхідно встановити, щоб гарантувати покриття. Їх модель є нелінійною цілочисельною математичною програмою, яка, як показано, є NP-важкою. Автори пропонують використовувати алгоритм пошуку табу для отримання розв'язку. Коефіцієнти затухання базуються на адаптації формули Хати (1980). Їх алгоритм був емпірично оцінений на десятиповерховому будинку з кількома сотнями станцій. Вони порівнюють свої розв'язки з нижньою межею, отриманою шляхом релаксації підмножини складних обмежень.

1.2.3 Моделі спроможності

Моделі ємності не тільки намагаються покрити задану площу, але й намагаються максимізувати пропускну здатність бездротової локальної мережі. Моделі пропускну здатності, як правило, дозволяють користувачеві вибирати між багатьма потенційними місцями розміщення точок доступу. Одне з перших досліджень цієї проблеми було виконано Шералі та ін. (1996). Вони розробили набір нелінійних оптимізаційних моделей для визначення оптимального розташування точки доступу. Мобільні пристрої розміщуються в точках сітки і використовуються стандартні моделі для визначення $g_i(x,y,z)$, втрат на шляху в точці сітки i , пов'язаної з точкою доступу в точці (x,y,z) . Однією з цільових функцій є мінімізація суми всіх зважених втрат шляху.

Другою цільовою функцією є мінімізація максимальної втрати шляху плюс вищезгаданий штраф Третя модель є комбінацією перших двох. Автори оцінюють три нелінійні методи розв'язання (метод Гука і Дживса, квазіньютонівський метод і метод спряжених градієнтів) і повідомляють про успішне розв'язання кількох невеликих задач.

У [6] досліджували використання генетичного алгоритму для проектування WLAN. Вони використовували формулу пропускну здатності Шеннона [2] для оцінки пропускну здатності заданої WLAN і працювали з чисельністю популяції 30 осіб для кожної генерації алгоритму або моделі. Наша модель також використовує формулу Шеннона для оцінки пропускну здатності. Однак моя стратегія рішення базується на оптимізації, тоді як Адікельс та його колеги використовують метаевристику. Методи оптимізації намагаються забезпечити гарантію оптимальності, тоді як метаевристика не дає жодних гарантій.

В [7] розробили модель нелінійного цілочисельного програмування, яка намагається врахувати перешкоди, притаманні MAC-протоколу Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), реалізованому в

бездротових локальних мережах. Припущення про трафіковий попит, частка часу передачі, доступна для MD, наближено дорівнює оберненому значенню загальної кількості конкуруючих MD плюс одиниця. Вони називають це ефективністю мережі і розробили спеціальні алгоритми для максимізації цієї нелінійної цільової функції. Оптимальний розв'язок їхньої моделі цілочисельного програмування дає найкращі сайти точок доступу з набору сайтів-кандидатів, які максимізують ефективність для заданого набору БР, що використовують один канал.

В [8] обговорюють використання комбінації декількох цільових функцій зі штрафними функціями та використання пошуку за табу для максимізації пропускної здатності. Вони не намагаються явно мінімізувати завади, натомість вирішили штрафувати місця розташування точок доступу, які, ймовірно, спричиняють велику кількість завад. Вони також повертають кілька потенційних рішень, намагаючись дати проектувальнику вибір, оскільки будь-яке рішення може бути непридатним з інших причин.

Зауважимо, що всі згадані вище алгоритми вимагають використання фіксованого набору потенційних місць розташування точок доступу. Вибір місць розташування точок доступу з набору потенційних місць схожий на задачу р-медіани, яка є задачею розміщення об'єктів, що намагається мінімізувати витрати для набору відомих користувачів. В [9] провели чудовий огляд стану проблеми р-медіани, яка класифікується як NP-Hard. Існують значні відмінності між проблемою проектування WLAN і проблемою р-медіани в тому, що при розміщенні точок доступу необхідно також враховувати інтерференцію між точками доступу і MD, пов'язаними з іншими точками доступу. Проблема р-медіани стосується лише вартості, яка паралельна сигналу між MD і пов'язаною з ним точкою доступу в задачі проектування WLAN. Тому задача проектування WLAN є особливим, більш складним випадком задачі р-медіани.

Іншим поширеним методом, який використовується для максимізації

пропускної здатності мережі при проектуванні WLAN, є розміщення точок доступу в геометричній сітці, як описано в [10]. При проектуванні WLAN за цією методикою точки доступу зазвичай розміщуються в трикутній сітці, гарантуючи, що сусідні точки доступу не мають однакових призначень каналів. В [10] обговорюють використання каналів, що перекриваються, для зміни форми використовуваної сітки на квадрат з діагональними ребрами. Ця методика є особливо корисною, якщо відомостей про потенційні місця розташування БС мало або взагалі немає.

1.2.4 Інструменти пропрієтарного дизайну

Програмний інструмент WirelessSystemsEngineering (WISE) був розроблений [11]. Це складний інструмент з чотирма модулями:

- модуль для зберігання оцифрованої версії будівлі;
- модуль для розрахунку втрат при поширенні між парами точок;
- модуль для вибору місця розташування точки доступу;
- графічний модуль для відображення рішення. Автори вказують, що в

III модулі вони використовують власну евристику.

Система на основі AutoCAD під назвою Site Modeling Tool (SMT) була розроблена Panjawni та ін. [12]. Зона покриття кожної точки доступу моделюється у вигляді багатокутника, і вони намагаються визначити крайні точки багатокутників. Їх система підтримує кілька поверхів і забезпечує візуальне відображення запропонованого рішення.

П'ятиступеневий підхід був використаний Хіллом [13] при розгортанні WLAN для Університету Карнегі-Меллона. Його група провела обстеження 65 будівель загальною площею близько трьох мільйонів футів². Він рекомендує використовувати дизайн, орієнтований на покриття, в районах з низькою щільністю, і дизайн, орієнтований на пропускну здатність, в районах з високою щільністю цих районах.

Дві системи, визначені як середовища вирішення проблем для проектування та аналізу бездротових мереж, описані Skidmore [14]. Комерційна версія називається SitePlanner і включає систему моделювання для оцінки продуктивності. Модуль оптимізації намагається мінімізувати певну комбінацію дефіциту покриття і частоти бітових помилок.

OPTNETTechnologies, Inc. також має рішення для проектування бездротових локальних мереж, яке було використано Міністерством військово-морського флоту. Система була розроблена для оптимізації розміщення передавачів на кораблях ВМС, а також включає модуль моделювання для оцінки пропускної здатності.

1.2.5 Призначення каналів

Після розміщення точок доступу певним методом, кожній точці доступу необхідно призначити частоту, на якій вона буде передавати. Однак через малу кількість доступних частот, якщо в системі більше трьох точок доступу, між ними виникатимуть певні перешкоди. Деякі системи дозволяють точкам доступу сканувати інші сусідні точки доступу і намагатися вибрати частоту з низьким рівнем перешкод. Наприклад, деякі виробники точок доступу (див. [15]) розробляють своє обладнання таким чином, щоб воно могло вибирати найкращу частоту для зв'язку. Кожна точка доступу прослуховує трафік на різних частотах і вибирає частоту, яка забезпечує найкращий прийом. Інші системи забезпечують фіксовану частоту для кожної точки доступу. Крім того, більшість алгоритмів вибору частоти, описаних в літературі, дозволяють використовувати лише три канали, що не перекриваються. В [16] представляють евристичну процедуру для призначення частот, яка тісно пов'язана з моделями призначення частот для клітинних мереж. Вони демонструють свою процедуру на клевер-листі, обслуговуваному шестикутником з точками доступу з антеною, що має кут

огляду 60 градусів. Тип отриманого рішення проілюстровано на рисунку 1.3 [16].

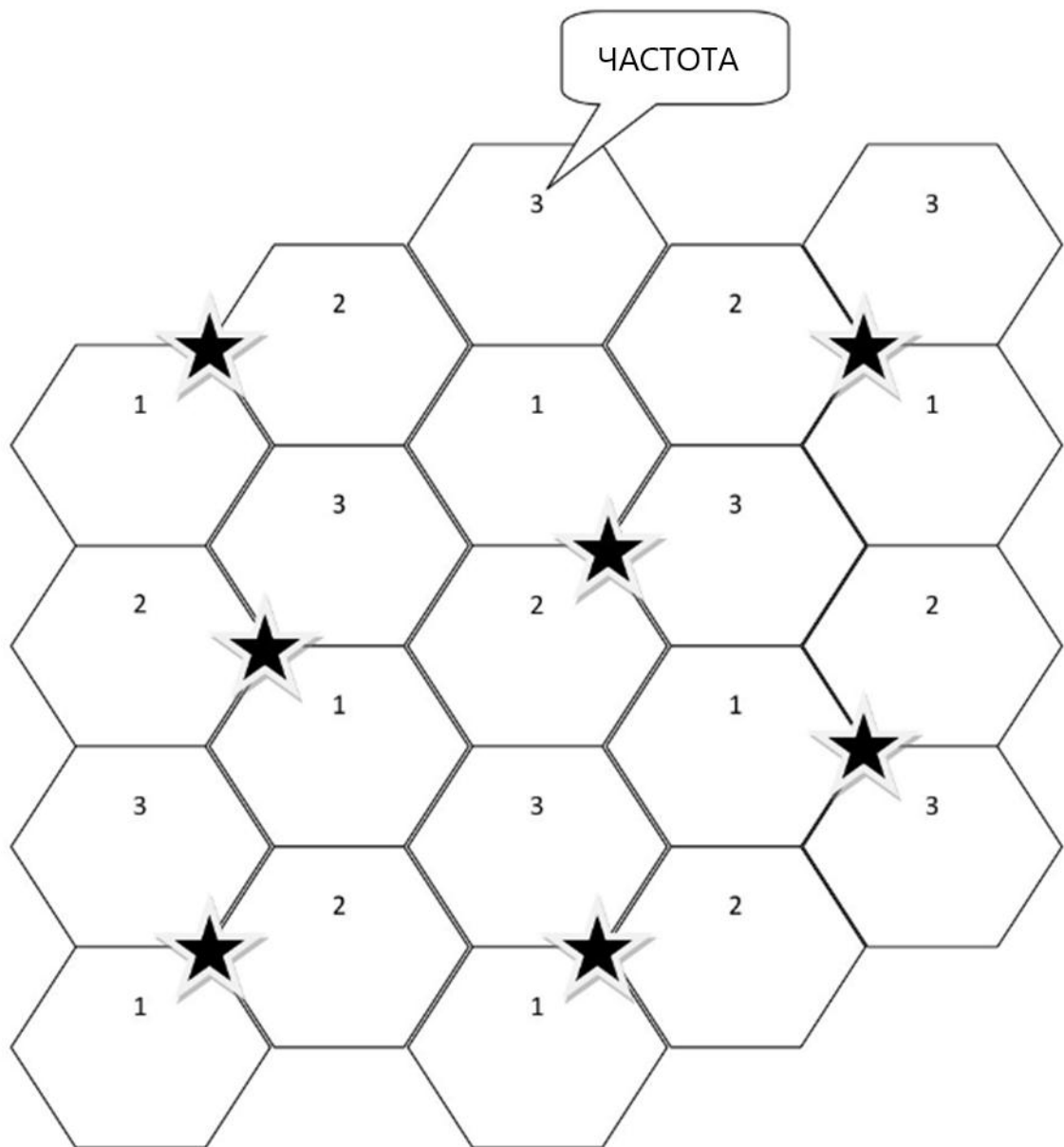


Рисунок 1.3 – Присвоєння частоти за допомогою трьох точок доступу з антеною під кутом 60 градусів на кожній зірці

Vanhatupa та ін. [17] представляють модель розподілу частот для середовищ, де співіснують декілька WLAN. Вони розглядають проблему призначення частот для нової точки доступу, враховуючи набір існуючих

фіксованих місць розташування точок доступу, таким чином, щоб пропускна здатність була максимальною. Показано, що їхнє формулювання еквівалентне знаходженню розв'язку задачі розфарбовування зваженого графа, яка є NP-повною. Тому для отримання наближених розв'язків застосовуються евристичні методи.

Kauffmann та ін. розглядають проблему розподілу частот точок доступу в середовищі, де в одній зоні розгорнуто декілька пристроїв WiFi. Вони також пропонують алгоритми, які дозволяють бездротовим клієнтам вибирати найкращу точку доступу для зв'язку. Рекомендуються розподілені алгоритми, які не потребують координації між різними мережевими пристроями. Вони стверджують, що робоча група IEEE 802.11k може включити прості модифікації програмного забезпечення.

Більшість моделей, описаних в літературі, використовують лише три частоти, що не перекриваються, як описано вище. Однак Mishra та ін. [18, 10] продемонстрували, як використання каналів, що перекриваються, може підвищити пропускну здатність мережі в деяких випадках.

Вони розробили величину, що називається I-фактор, яка вказує на ступінь перекриття між двома каналам.

I-фактор є модифікацією алгоритму випадкового ущільнення, описаного в роботі [19], який призначає канали за допомогою алгоритму оптимізації, розробленого для вирішення проблем рівноправності. Вони провели імітаційний аналіз на рівні пакетів, порівнюючи частково перекриті канали з алгоритмом випадкового ущільнення. У своєму емпіричному дослідженні вони виявили, що використання одинадцяти частот, що частково перекриваються, збільшує загальну пропускну здатність від 1,6 до 3,0 разів, залежно від щільності MD. Вони також помітили, що використання каналів, що частково перекриваються, може забезпечити пропускну здатність до 70% від теоретичної максимальної пропускну здатності, визначеної за умови використання одинадцяти каналів, що не перекриваються.

1.3 Мета дослідження

Проектування бездротової локальної мережі включає в себе визначення місць розташування точок доступу, а також визначення різних призначень каналів. Звичайно, оптимальне розташування точок доступу залежить від розташування MD. Існує дві типології цих проблем: фіксована проблема і проблема з рухомими точками. У фіксованій проблемі розташування MD відоме і не змінюється. Офіс зі стаціонарними комп'ютерами, підключеними до бездротової локальної мережі, є одним з прикладів фіксованої проблеми. Аеропорт з мандрівниками, які використовують ноутбуки в різних місцях, є прикладом проблеми, пов'язаної з рухомими даними.

Проблему проектування WLAN можна визначити наступним чином: "Маючи набір MD з відомим або невідомим розташуванням і бюджетне обмеження на кількість точок доступу, що встановлюються, визначити комбінацію розташування і частотного призначення для кожної точки доступу, яка забезпечить максимальну пропускну здатність мережі".

Мета цього дослідження – представити моделі, алгоритми розв'язання та емпіричний аналіз, які можуть бути використані для отримання якісного проекту для різних версій задачі проектування WLAN. В рамках цього дослідження ми оцінюємо програмне забезпечення, доступне на сервері NEOS (NEOS, 2010), комерційне програмне забезпечення від систем LINDO. Оскільки обмеження і цілі проблеми проектування WLAN є складними, я використав ітеративний процес для вирішення проблеми.

Моя перша ітерація, описана в Розділі 2, обговорює методи розв'язання задачі проектування одноканальної фіксованої WLAN. Я припускаю, що місця розташування всіх MD відомі, і дозволяємо призначати точки доступу лише на одну частоту. У Розділі 3 я обговорив методи вирішення проблеми проектування багатоканальної фіксованої WLAN. Я як і раніше припускаю, що місця розташування всіх MD відомі, але дозволяємо призначати точки

доступу на будь-яку частоту, як таку, що не перекривається, так і таку, що перекривається. У розділі 4 я обдумав власні методи розв'язання задачі проектування багатоканальної кучової WLAN. У кожному розділі я представляю модель і програмну реалізацію для відповідної задачі, а також емпіричний аналіз. Розділ 5 містить підсумок моєї роботи й мої висновки.

2 ПРОБЛЕМА ПРОЕКТУВАННЯ ФІКСОВАНОЇ БЕЗДРОТОВОЇ ЛОКАЛЬНОЇ МЕРЕЖІ

2.1 Оптимізаційна модель та алгоритми розв'язання

У цьому розділі описано методи розв'язання задачі проектування одноканальної стаціонарної бездротової локальної мережі. У цій главі я розробляю модель максимізації пропускної здатності для розміщення точок доступу в одноповерховій будівлі, припускаючи, що доступний лише один бездротовий канал, а розташування точок доступу є фіксованим і відомим. На відміну від інших моделей, що використовуються для розміщення точок доступу, я не вимагаю набору потенційних місць розташування точок доступу і дозволяю розміщувати точки доступу в будь-якому місці будівлі. Я використовую формулу пропускної здатності Шеннона (Shannon, 1949) для оцінки сигналу та перешкод між точками доступу та MD.

У роботі Кеннінгтона та ін. [20] я представив модель, яку можна використовувати для оптимізації проектування WLAN. Модель має нелінійну цільову функцію, лінійні обмеження і численні двійкові змінні. Особливі випадки виникають внаслідок фіксації деяких змінних рішень, таких як кількість точок доступу, що встановлюються (бюджетне обмеження), рівні потужності точок доступу та/або попередній вибір точок доступу, що встановлюються. Ця модель передбачає використання каналів, що не перекриваються, і користувач повинен вибрати з набору потенційних місць розташування точок доступу, враховуючи відомі місця розташування всіх MD. Таким чином, загасання між усіма можливими місцями розташування точок доступу і всіма фіксованими місцями розташування MD можна розрахувати заздалегідь і надати як вхідні дані для розв'язувача. Для цієї моделі було розроблено кілька потужних метаевристик і проведено

емпіричний аналіз. У цьому дослідженні ми представляємо модель, яка не потребує набору потенційних місць розташування ПД як вхідних даних. Тобто, координати (x, y) кожної точки доступу є змінною рішення, яка повинна бути визначена розв'язувачем моделі.

2.1.1 Проблема зі змінним розташуванням точки доступу

Таблиця 2.1 містить всі множини та змінні, що використовуються в цій моделі.

Таблиця 2.1 – Множини, константи та змінні, що використовуються у фіксованій моделі

Ім'я	Тип	Опис
S	набір	Набір AP
M	набір	Набір всіх MD
P	набір	Набір потенційних рівнів потужності
θ_m	константа	Розташування MDm на осі x
φ_m	константа	Розташування MDm на осі y
CH	константа	Висота стелі – висота MD
	константа	Мінімальна координата x для розміщення точок доступу
	константа	Максимальна координата x для розміщення точок доступу
	константа	Мінімальна координата y для розміщення точок доступу
	константа	Максимальна координата y для розміщення точок доступу
N_m	константа	Шум, пов'язаний з MDm
P_s	константа	Налаштування живлення для точок доступу
A_s	змінна	Множина всіх MD, пов'язаних з AP; $A_s \subset M$
x_s	змінна	Розташування точок доступу на осі абсцис
y_s	змінна	Розташування точок доступу на осі y
g_{sm}	змінна	Загасання між APs і MDm
d_{sm}	змінна	Відстань між точками APs і MDm
h_m	змінна	Точка доступу, якій призначено MDm; $h_m \in S$

Кінець таблиці 2.1

Ім'я	Тип	Опис
t_m	змінна	Сигнал на відстані MD m від точки доступу, якій він призначений
u_m	змінна	Сумарна завада на відстані MD m
v_m	змінна	Загальна кількість лікарів загальної практики, закріплених за h_m

Для спрощення ми обмежуємо кожний MD зв'язуватися лише з AP, яка має найсильніший сигнал. Це означає, що не існує вимоги мінімального SIR для зв'язку. Зауважимо, що кожна точка доступу $s \in S$ обмежена прямокутником, який визначається:

$$\{(x_s, y_s) | x_s^{min} \leq x_s \leq x_s^{max}, y_s^{min} \leq y_s \leq y_s^{max}\}. \quad (2.1)$$

Оскільки місцезнаходження MD відоме, алгоритм попередньої обробки:

$$d_{sm} = \sqrt{(x_s - x_m)^2 + (y_s - y_m)^2} + CH^2 \quad \forall s \in S, m \in M$$

а загасання між AP*s* MD*m* задається формулою:

$$g_{sm} = 10^{-1.46d_{sm}3.27} \quad \forall s \in S, m \in M$$

Для встановлення цих меж можна використати ритм. Подібна стратегія була прийнята Skidmore та ін. [14]. Відстань між точками AP*s* MD*m* визначається за формулою. Виведення (2.2) можна знайти у Додатку А.

Припустимо, що MD*m* призначена для точки доступу h_m ; тоді рівень сигналу в точці m задається формулою:

$$t_m = Pg_{hmhm} \forall m \in M.$$

Змінна u_m позначає перешкоди на MD m , коли він спілкується з AP s_i задається формулою:

$$u_m = \sum_{s \in S \setminus \{h_m\}} P_{sgsm} + N_m \quad \forall m \in M.$$

Оскільки кожна точка доступу зв'язується з усіма MD, підключеними один до одного з часовим поділом, перешкоди на MD виникають лише від інших точок доступу, що використовують ту саму частоту.

Отже, використовуючи формулу потужності Шеннона, ми шукаємо:

$$\text{maximize}_X [\log(1 + t/u_{mm})/v_m], m \in M \quad (2.5)$$

де:

$$v_m = 1/A_{hm} \quad \forall m \in M. \quad (2.6)$$

Наступні обмеження гарантують, що кожна точка доступу буде розміщена в межах можливого регіону.

$$x_s^{\min} \leq x_s \leq x_s^{\max} \quad \forall s \in S \quad (2.7)$$

$$y_s^{\min} \leq y_s \leq y_s^{\max} \quad \forall s \in S \quad (2.8)$$

Верхня та нижня межі визначаються схемою, яка розділяє MD в набори S.

Для розробки моделі (2.1)-(2.8), описаної вище, було зроблено кілька суттєвих спрощень роботи WLAN. По-перше, WLAN використовують

дискретний набір можливих швидкостей передачі, кожна з яких має певні вимоги до співвідношення сигнал/шум. Формула Шеннона передбачає роботу з безперервними швидкостями. По-друге, моя модель не враховує уникнення зіткнень. Однією з проблем бездротової передачі даних є те, що обладнання не може слухати під час передачі. Тому виявлення колізій неможливе. Отже, уникнення колізій реалізується за допомогою CSMA/CA (множинний доступ з розділенням каналів і уникненням колізій). Це означає, що MD, який бажає передати, повинен слухати протягом заздалегідь визначеного часу, щоб визначити, чи передає інший MD в межах діапазону. Якщо канал виявляється вільним, то починається передача. Звичайно, два MD, які намагаються передавати, можуть бути поза зоною досяжності один одного. Це називається проблемою прихованого вузла. По-третє, я використовую одну частоту, тоді як WLAN має три частоти, що не перекриваються. Нарешті, я використовую просту модель поширення, яка базується лише на відстані. Через складну природу поширення радіосигналу, зокрема вплив стін, меблів та інших електронних пристроїв, наша модель може лише приблизно передбачити рівень сигналу для будь-якої області відносно точки доступу. Навіть з цими спрощуючими припущеннями, задача (2.1)-(2.8) є складною задачею нелінійного програмування.

Цільова функція не є ні увігнутою, ні опуклою, і вона розривається в точці, де MD призначається до іншого AP. Подальше обговорення увігнутості функції можна знайти в Додатку Б. Хоча існують методи розв'язання неопуклих задач оптимізації, розриви роблять проблему майже неможливою для розв'язання за допомогою методів глобальної оптимізації [21].

2.1.2 Алгоритм для задачі визначення місцезнаходження змінної точки доступу

Pinter [21] надає чудовий посібник з розробки алгоритмів глобальної оптимізації. Я розробив стратегію розбиття, схожу на процедуру розгалуження та обмеження для цілочисельного програмування, щоб отримати хороший розв'язок задачі розміщення змінних точок доступу.

Ми почнемо з набору MD у відомих місцях і кількості точок доступу, які потрібно призначити ($|S|$). MD розподіляються на підмножини $|S|$ таким чином, щоб всі MD в даній підмножині знаходились відносно близько один до одного. Для прикладу, показано на рисунку 2.1 з $|S| = 3$, дев'ять MD були призначені трьом підмножинам. Спочатку кожній точці $|S|$ присвоюється місце в центрі кожної підмножини і визначаються межі для x- та y-координат кожної точки.

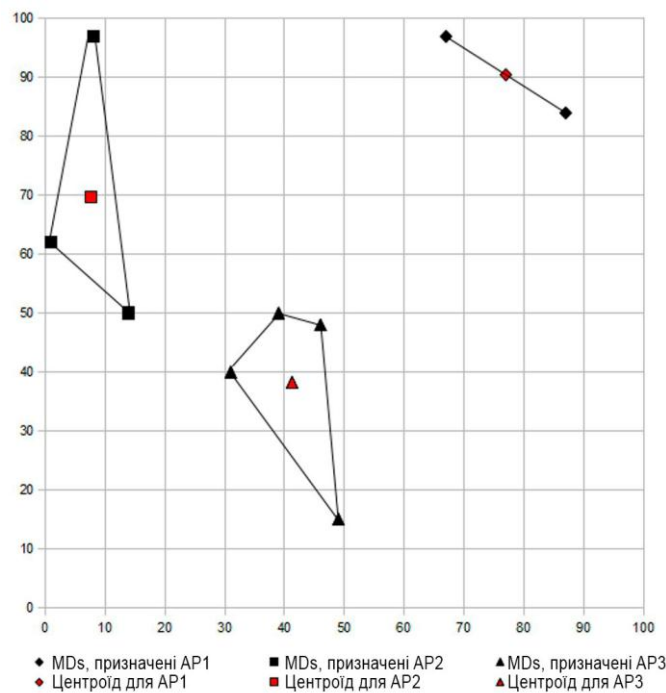


Рисунок 2.1 – Приклад задачі з дев'ятьма MD та трьома AP

Таким чином, APs обмежується:

$$\{(x_s, y_s) : x_s^{\min} \leq x_s \leq x_s^{\max}, y_s^{\min} \leq y_s \leq y_s^{\max}\},$$

як показано на рисунку 2.2, де межі виглядають як прямокутники. Моє поточне розв'язуване завдання дає нижню межу цільової функції.

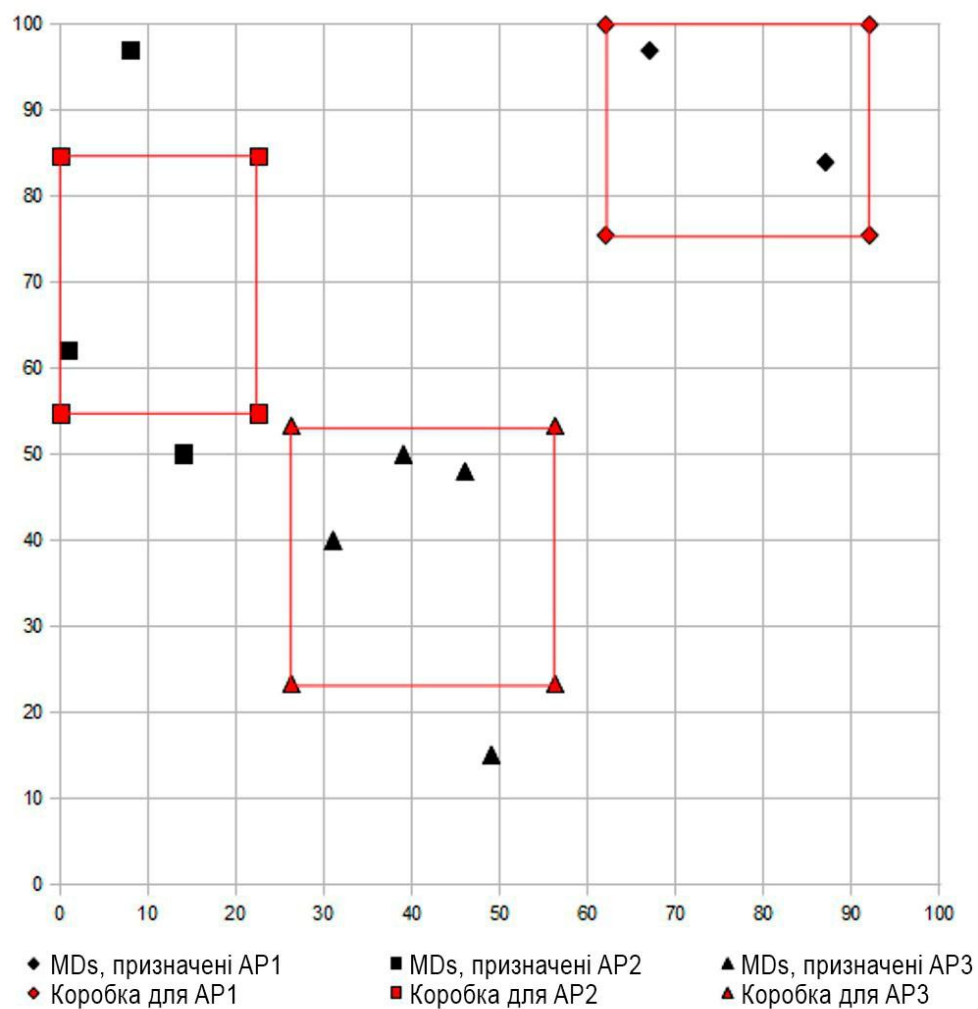


Рисунок 2.2 – Границі для прикладу задачі з трьома точками

Далі ми отримаємо оцінку найбільшого градієнта (Λ) рівняння (2.5) над допустимою областю, визначеною (2.7) і (2.8). Нехай (x_1, y_1) , (x_2, y_2) , ..., $(x_{|S|}, y_{|S|})$

$y_{|S|}$) – будь-який допустимий розв'язок (2.7) і (2.8). Тоді цільова функція (2.5) може бути використана для отримання потужності цього розв'язку, позначеної через p . Нехай d_i – найбільша відстань від розв'язку $i \in \{1, \dots, |S|\}$ до границі, заданої (2.7) і (2.8), і нехай $d = (d_1, \dots, d_{|S|})$ позначає вектор цих відстаней. Тоді $p + \Lambda \|d\|$ використовується як оцінка верхньої межі. Якщо різниця в межах знаходиться в межах заданого допуску, то остаточний розв'язок знайдено. В іншому випадку обирається AP і створюються дві підзадачі.

Нехай $y_s^{max} - y_s^{min} > x_s^{max} - x_s^{min}$. Тоді перша підзадача обмежує точки доступу до

Множина $\{(x_s, y_s) : x_s^{min} \leq x_s \leq x_s^{max}, y_s^{min} \leq y_s \leq (y_s^{max} + y_s^{min})/2\}$ та

підзадача два обмежує

AP s to $\{(x_s, y_s) : x_s^{min} \leq x_s \leq x_s^{max}, (y_s^{min} + y_s^{max})/2 \leq y_s \leq y_s^{max}\}$.

Якщо $y_s^{max} - y_s^{min} \leq x_s^{max} - x_s^{min}$, то підпроблеми розбиваються по координаті x . Підпроблеми, визначені за допомогою AP 2, проілюстровані на рисунку 2.3. Для кожної підзадачі обчислюються нові верхня і нижня межі, і ця стратегія повторюється до тих пір, поки різниці меж у всіх підзадачах не задовольняють допуск на закінчення. Цей процес повторюється кілька разів для різних початкових розділів, і загальна найкраща пропускна спроможність вважається рішенням.

Як показано на рисунках, допустима область для пошукового алгоритму може бути описана у термінах прямокутників та областей. Кожна точка доступу, $s \in S$, повинна лежати в одній з областей, позначається через $B_s = \{(x, y) | x_s^{min} < x < x_s^{max}, y_s^{min} < y < y_s^{max}\}$. Область, $r \in R$ називається визначаються блоками $|S|$, по одному для кожної точки доступу. Клітинки $|S|$, що відповідають області r , позначаються через $B_1^r, \dots, B_{|S|}^r$. Використовуючи ці позначення, наш глобальний алгоритм виглядає наступним чином.

Глобальний алгоритм.

Параметри налаштування: N – розмір початкових комірок в області 1, η – кількість випадкових вибірок для кожної ітерації, Λ – оцінка найбільшого градієнта в будь-якій можливій точці, і β – допуск на розмір області, який використовується для завершення процедури.

Використовуйте евристику розбиття для визначення $A_1, \dots, A_{|S|}$ та N для визначення $B_1^1, \dots, B_{|S|}^1$ в області 1. Нехай $R \leftarrow \{1\}$.

Випадковим чином виберемо n розкладів AP у кожній клітинці області 1. Позначимо через ζ найкращий знайдений розв'язок. За допомогою Λ визначте верхню межу для області 1, скажімо, p_1 при $\max(p_t | t \in R) > \zeta$ то.

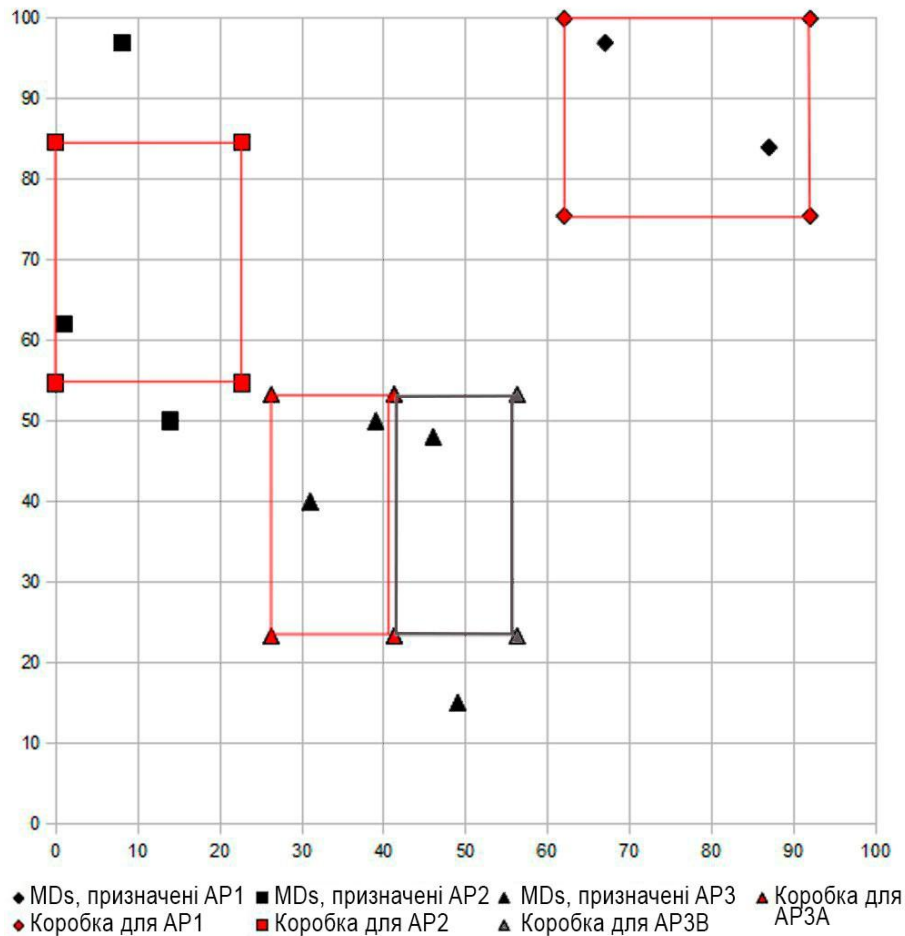


Рисунок 2.3 – Підзадачі, створені шляхом розбиття допустимої області AP 2

Нехай $r \leftarrow \operatorname{argmax}(p_t : t \in R)$. Якщо загальний розмір області r менший за β , то перервемо роботу.

Нехай через B_b^r позначено коробку в області r з найбільшою стороною. Нехай $u \leftarrow |R| + 1$ і $R \leftarrow R \setminus \{r\}$.

Створіть два нових регіони (u та $u + 1$), зберігши усі поля регіону r , окрім B_b^r , та розбивши B_b^r по довшій стороні для кожного з нових регіонів. $R \leftarrow R \cup \{u, u + 1\}$.

Використовуючи ту саму стратегію, що й на кроці 2, поновіть ζ , якщо потрібно, і визначте p_u та p_{u+1} . end while.

Найкраще рішення можна знайти на кроках 2 і 6.

Алгоритм має чотири параметри, що налаштовуються, розмір початкової області для кожної точки доступу (N), кількість зразків, взятих у кожній підзадачі (η), константа Ліпшиця (Λ) та константа термінального допуску (β). Якщо вибрані області занадто великі, алгоритм збігається повільно. Якщо області занадто малі, може бути знайдено гірший розв'язок. Константа Ліпшиця є оцінкою максимального градієнта над підзадачею. Якщо Λ недооцінити, наближений до оптимального розв'язок може бути не знайдено, але якщо Λ переоцінити, алгоритм збігається дуже повільно.

Як зазначалося раніше, цільова функція розривається, коли змінюється призначення AP для будь-якого MD. Тому ми використовуємо додаткові обмеження, щоб запобігти будь-яким змінам у призначенні. Наступні обмеження гарантують, що множини A_s для $s \in S$ залишаються фіксованими для всіх можливих місць розташування точок.

$$d_{hmm} \leq d_{sm} \quad \forall m \in M, s \in S \setminus \{h\}_m \quad (2.9)$$

Тепер ця модель має неперервну цільову функцію в меншій допустимій області. Це дозволяє нам використовувати ряд різних методів для пошуку розв'язку в цій області. Додавання обмежень (2.9) гарантує, що розв'язання

нашої моделі дасть гарний розв'язок лише за умови, що MD в A_s призначені s . Тому нам потрібно визначити гарний алгоритм розбиття MD, а потім нам також потрібно оптимізувати інші можливі розбиття, щоб гарантувати, що ми знайдемо гарний загальний розв'язок. Очевидно, що буде велика кількість нездійснених розділів. Наприклад, якщо дві MD, які знаходяться на великій відстані одна від одної, розмістити в одній підмножині, то неможливо буде знайти позицію точки доступу, яка задовольняє (2.9).

Ми використовуємо евристику, щоб знайти хороше початкове розбиття з кількістю підмножин, що дорівнює $|S|$. Ми розміщуємо MD, які знаходяться на найбільшій відстані один від одного, в окремі підмножини.

Потім ми повторюємо процес, максимізуючи мінімальну відстань між нашими новими MD і вже розміщеними MD. Після того, як ми розмістили $|S|$ якірних MD в окремих підмножинах, ми розміщуємо решту MD у підмножинах з найближчими якірними MD.

Наприклад, враховуючи розташування MD у таблиці 2.2 і припускаючи використання трьох точок доступу, нам знадобиться три підмножини. У таблиці ми бачимо, що максимальна відстань між точками 2 і 3 становить 92 км. Отже, ці точки будуть розміщені в різних підмножинах. Нам потрібно вибрати одну додаткову підмножину, максимізуючи мінімальну з двох відстаней між вже розміщеними підмножинами. Таким чином, MD 7 буде обрано з відстанню 79. Продовжуючи наш приклад, ми розмістимо MD 1 у множині з MD 7, MD 4 розмістимо з MD 2, і так далі, поки не отримаємо розбиття $A_1 = \{2,4,5,6\}$, $A_2 = \{3,8,9\}$, і $A_3 = \{1,7\}$. Отримавши розбиття, ми застосовуємо алгоритм, описаний вище. Інший розділ створюється шляхом перепризначення одного або декількох MD на інший A_s . Базова процедура застосовується кілька разів до різних розділів і повертається найкращий знайдений розв'язок. Детальні псевдокоди для нашого алгоритму можна знайти в [22] та в Додатку В.

Таблиця 2.2 – Розташування та відстані між MD для прикладу задачі

MD			Відстань							
			2	3	4	5	6	7	8	9
1	67	97	84	59	67	55	53	24	71	75
2	49	15		92	31	36	33	79	49	67
3	8	97			61	56	62	80	47	36
4	31	40				13	17	71	20	37
5	39	50					7	59	25	40
6	46	48						55	32	47
7	87	84							81	89
8	14	50								18
9	1	62								

2.1.3 Емпіричний аналіз

Після попередньої обробки для визначення (2.7), (2.8) і (2.9) та підстановки (2.1), (2.4) і (2.6) у (2.5) маємо модель зі складною нелінійною цільовою функцією та обмеженнями (2.7)-(2.9). Після застосування процедури розбиття множини $A_1, \dots, A_{|S|}$ є фіксованими для кроку 1. Я буду використовувати алгоритм без назви тому це буде «Алгоритм А». Початкова оцінка Λ була визначена шляхом обчислення великої кількості градієнтів вибірки. Для того, щоб переконатися, що Λ не є завищеною оцінкою, максимальне знайдене значення було збільшено в 10 разів. Значення Λ , використані в емпіричному аналізі, є варіаціями навколо цього значення. При аналізі розв'язувача я використовував повний фактор з різними параметрами налаштування, а також з різною кількістю як MD, так і AP. Результати можна знайти в таблиці 2.3.

Таблиця 2.3 – Емпіричний аналіз для налаштування параметрів, «Алгоритм А»

Параметри			5APs,30MDs		5APs,40MDs		6APs,30MDs		6APs,40MDs	
n	Λ	N	ζ	Час	ζ	Час	ζ	Час	ζ	Час
4	0.09	25	5.69	00:00:08	6.27	00:00:03	8.34	00:00:01	7.04	00:00:02
4	0.10	25	5.64	00:00:14	6.43	00:00:09	8.34	00:00:01	7.04	00:00:16
4	0.11	25	5.81	00:00:05	6.28	00:00:23	8.34	00:00:06	7.04	00:01:05
8	0.09	25	5.54	00:00:06	6.27	00:00:04	8.34	00:00:01	7.04	00:00:04
8	0.10	25	5.72	00:00:11	6.27	00:00:15	8.34	00:00:03	7.04	00:00:33
8	0.11	25	5.73	00:00:12	6.27	00:00:34	8.34	00:00:12	7.04	00:02:11
12	0.09	25	5.70	00:00:18	6.27	00:00:06	8.34	00:00:01	7.04	00:00:06
12	0.10	25	5.73	00:00:04	6.27	00:00:23	8.34	00:00:05	7.04	00:00:49
12	0.11	25	5.69	00:00:03	6.59	00:00:46	8.34	00:00:18	7.04	00:03:15
4	0.09	20	5.92	00:00:28	6.25	00:00:01	8.34	00:00:01	7.04	00:00:00
4	0.10	20	5.71	00:00:24	6.32	00:00:01	8.34	00:00:01	7.14	00:00:01
4	0.11	20	5.73	00:00:25	6.25	00:00:04	8.34	00:00:01	7.04	00:00:02
8	0.09	20	5.65	00:00:17	6.41	00:00:01	8.34	00:00:01	7.25	00:00:00
8	0.10	20	5.10	00:00:02	6.25	00:00:02	8.34	00:00:01	7.04	00:00:02
8	0.11	20	5.07	00:00:30	6.25	00:00:04	8.34	00:00:01	7.04	00:00:04
12	0.09	20	5.71	00:00:09	6.25	00:00:01	8.34	00:00:01	7.04	00:00:01
12	0.10	20	5.65	00:00:11	6.25	00:00:03	8.34	00:00:01	7.04	00:00:03
12	0.11	20	5.29	00:00:07	6.25	00:00:06	8.34	00:00:01	7.04	00:00:06
4	0.09	15	5.31	00:00:01	6.11	00:00:01	8.34	00:00:01	7.04	00:00:01
4	0.10	15	5.37	00:00:01	6.35	00:00:01	8.34	00:00:01	7.04	00:00:01
4	0.11	15	5.94	00:09:27	6.40	00:00:01	8.34	00:00:01	7.04	00:00:01
8	0.09	15	5.71	00:00:46	6.11	00:00:01	8.34	00:00:01	7.04	00:00:01
8	0.10	15	5.80	00:00:19	6.35	00:00:01	8.34	00:00:01	7.18	00:00:01
8	0.11	15	5.79	00:01:49	6.23	00:00:01	8.34	00:00:01	7.04	00:00:01
12	0.09	15	5.75	00:00:33	6.42	00:00:01	8.34	00:00:01	7.04	00:00:01
12	0.10	15	5.81	00:07:21	6.35	00:00:01	8.34	00:00:01	7.14	00:00:01
12	0.11	15	5.74	00:03:07	6.35	00:00:01	8.34	00:00:01	7.12	00:00:01

Я застосував дисперсійний аналіз до результатів, наведених у таблиці 2.3, і виявили, що жоден з параметрів налаштування не мав значного впливу на продуктивність у жодному з чотирьох експериментів, що свідчить про те, що «Алгоритм А» є надійним. В експерименті з шістьма точками доступу та 40 MD виявив, що розмір коробки суттєво впливає на час виконання на рівні

значущості 0,05. У жодному з інших експериментів не було отримано значущих результатів щодо часу. У таблиці 2.4 наведено Параметри розв'язувача LINDO API.

Таблиця 2.4 – Параметри розв'язувача LINDO API

Проблеми	APs	MDs	Оцінений градієнт		Точний градієнт	
			4 Запуски	8 Запусків	4 Запуски	8 Запусків
1	3	15	3.43	3.43	3.43	3.43
2	3	15	4.92	4.92	4.92	4.92
3	3	15	4.84	4.84	4.84	4.84
4	3	20	4.78	4.78	4.78	4.78
5	3	20	4.33	4.33	4.33	4.33
6	3	20	3.76	3.76	3.76	3.76
7	3	25	4.40	4.40	4.40	4.40
8	3	25	3.35	3.35	3.35	3.35
9	3	25	3.92	3.92	3.92	3.92
10	5	15	7.55	7.55	7.55	7.55
11	5	15	6.95	6.95	6.95	6.95
12	5	15	7.55	7.55	7.55	7.55
13	5	20	6.66	6.66	6.66	6.66
14	5	20	6.15	6.15	6.15	6.15
15	5	20	6.44	6.44	6.44	6.44
16	5	25	5.45	5.45	5.45	5.45
17	5	25	5.98	5.98	5.98	5.98
18	5	25	5.93	5.93	5.93	5.93
19	7	15	9.14	9.14	9.14	9.14
20	7	15	9.49	9.49	9.49	9.49
21	7	15	9.01	9.01	9.01	9.01
22	7	20	10.36	10.36	10.36	10.36
23	7	20	7.94	7.94	7.94	7.94
24	7	20	7.87	7.87	7.87	7.87
25	7	25	7.42	7.42	7.42	7.42
26	7	25	7.89	7.89	7.89	7.89
27	7	25	7.80	7.80	7.80	7.80

Сервер оптимізації NEOS (<http://www-neos.mcs.anl.gov>) містить найсучасніше програмне забезпечення для різних оптимізаційних моделей.

Це спільний проект спільноти оптимізаторів з обчислювальними ресурсами, наданими Аргонською національною лабораторією, SunMicrosystems та кількома університетами. Який я втім зміг знайти лише на "альтернативних" ліцензійних сайтах. Потім проаналізувавши параметри налаштування LINDOAPI, які включали кількість початкових точок, а також те, чи градієнт оцінювався або обчислювався в тестових точках. Числа, наведені в таблиці 2.4, – це значення пропускну здатності, отримані в результаті кожного запуску. Зауважте, що всі чотири версії дали однакові результати для таблиці 2.3.

Тому для додаткового тестування я використав оцінений градієнт з чотирма початковими розв'язками. Це найшвидша з чотирьох протестованих процедур.

Щоб визначити найкращий розв'язувач для дослідження, я порівняв результати продуктивності шести розв'язувачів NEOS, LINDOAPI на своєму локальному обладнанні та «Алгоритм А» на невеликому наборі задач. Результати можна знайти в таблиці 2.5.

Таблиця 2.5 – Продуктивність розв'язувачів NEOS, LINDOAPI

Розмір		Локальнеобладнання		NEOS Сервер					
MDs	APs	LINDOAPI	«Алгоритм А»	LINDOGlobal	BARON	ACRS	ASA	PGA Pack	PSwarm
3	15	100	91	87	INF ¹	ERR ²	82	105 ³	47
3	25	100	94	77	INF ¹	ERR ²	83	84	45
5	15	100	99	99	INF ¹	ERR ²	100	100	42
5	25	100	101	97	INF ¹	ERR ²	97	97	37
7	15	100	96	96	INF ¹	ERR ²	96	96	29
7	25	100	97	92	INF ¹	ERR ²	92	92 ³	22

Я використав так звану модель BARON, він не зміг знайти прийнятний розв'язок для жодної з задач. Було зроблено кілька додаткових запусків з різними налаштуваннями BARON'a (https://www.gams.com/44/docs/S_BARON.html?search=baron). На жаль, жоден з цих запусків не дав прийнятного розв'язку. У двох випадках PGAPack[це бібліотека алгоритмів] повернув значення, які не задовольняють умовам (2.7) та (2.8). Всі прогони з ACRS повернули повідомлення про помилку, яке вказує на те, що розв'язок не може бути знайдений. Не було знайдено жодної документації ні для PGAPack, ні для ACRS. На основі цих результатів я виключив розв'язувачі на сервері NEOS з подальшого розгляду. Наведені результати масштабуються до потужності, визначеної LINDOAPI на локальному обладнанні. «Алгоритм А» базується на стратегії розгалужень і обмежень, і тому не такий швидкий, як LINDOAPI. Однак, оскільки обидва алгоритми надають схожі рішення, я впевнений, що LINDOAPI надає рішення, близьке до оптимального.

LINDOAPI, що працює на локальній машині, однозначно надає найкращі розв'язки для цього типу нелінійних задач, тому я буду використовувати NLP-розв'язувач у LINDOAPI (LINDO, 2010) на Java для решти цього дослідження. Також протестую «Алгоритм А», який дає результати, що не відрізняються від результатів LINDOAPI. Те, що декілька розв'язувачів дають схожі результати, є гарним доказом того, що я отримую близькі до оптимальних розв'язки.

2.2 Проект для офісної будівлі

П'ять процедур були реалізовані в програмному забезпеченні та емпірично оцінені на одному поверсі офісної будівлі. Схему форми будівлі можна знайти на рисунку 2.4.

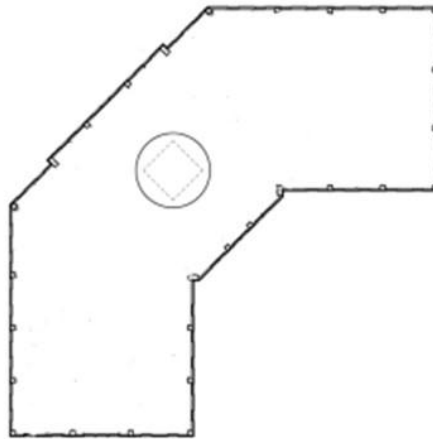


Рисунок 2.4 – Офісна будівля

На підлозі було розміщено тридцять чотири точки сітки. Ці точки є потенційними місцями розташування точок, які є вхідними даними для трьох евристик (пошук по сусідству [Neighborhood], генетична процедура [Genetic] і процедура пошуку за табу [Tabu]), розроблених у попередньому дослідженні, [20]. Всі ці евристичні процедури перебирають множину потенційних місць для пошуку оптимального рішення. Таким чином, вони намагаються вирішити дискретну версію цієї проблеми. Це відрізняється від стратегії «Алгоритм А», де пошук відбувається на множині клітинок. «Алгоритм А» намагається вирішити безперервну версію цієї проблеми. Крім того, ці 34 сайти були використані для тестування процедури повного перебору. При повному переборі випробовуються всі комбінації з 34 точок сітки, взятих $|S|$ за один раз. Це забезпечує найкраще можливе рішення, враховуючи, що точки доступу мають бути розташовані на підмножині з 34 точок сітки. «Алгоритм А» і LINDOGlobal не потребують сітки і дозволяють розміщувати точки доступу в будь-якому місці на підлозі. Емпіричні результати можна знайти в таблиці 2.6, де результати LINDOGlobal масштабуються до 100. «Алгоритм А» і LINDOGlobal показали значно кращі

результати, ніж будь-яка евристика на основі сітки, включаючи повний перебір. Звичайно, для покращення повного перебору та трьох евристичних алгоритмів на основі сітки (Neighborhood, Genetic, Tabu) можна було б використати більш дрібну сітку. Однак, ⁽³⁴⁾₍₇₎ дає понад 5 мільйонів комбінацій, які необхідно перевірити. Подвоєння цієї кількості до ⁽⁶⁸⁾₍₇₎ дає понад 900 мільйонів комбінацій з відповідним збільшенням часу обчислень. Оптимальні значення мети для «Алгоритм А» та LINDOGlobal були дуже близькими. Оскільки LINDOGlobal працює швидше, рекомендовано його для розв'язання задачі проектування фіксованої бездротової локальної мережі.

2.3 Підсумки та висновки

У цій главі представлено дослідження проблеми проектування одноканальної фіксованої бездротової локальної мережі. Для фіксованої задачі розташування всіх мобільних пристроїв відоме, і проблема полягає у визначенні розташування всіх точок доступу. Модель допускає розташування точок доступу в будь-якому місці в межах поверху задачі проектування. Однак, поточна модель дозволяє використовувати лише одну частоту. Ця модель має складну нелінійну цільову функцію, лінійні та квадратичні обмеження, а також безперервні змінні. Шість комерційних розв'язувачів плюс розв'язувач, розроблений одною дослідницькою групою, були протестовані на декількох випадково згенерованих тестових задачах. Два з цих розв'язувачів (BARON та ACRS) не змогли повернути розв'язок у жодному з тестових випадків. Один з розв'язувачів (PGAPack) повернув кілька розв'язків, які були нездійсненними.

Ще один розв'язувач (PSwarm) повернув дуже погані розв'язки. Два з розв'язувачів (LINDOGlobal, та розв'язувач, «Алгоритм А») повернув найкращі рішення. Оскільки LINDOGlobal працює дуже швидко,

рекомендовано його для цього проблемного класу.

Таблиця 2.6 – Порівняння моделей

Проб- лема	Розмір		Перерахува- ння		Neighborhood		Genetic		Tabu		FixedAPGl obal		LINDOGlobal	
	APs	MDs	Ме- жа	Час	Ме- жа	Час	Ме- жа	Час	Межа	Час	Межа	Час	Межа	Час
1	3	15	38	00:01	38	00:01	38	00:01	35	00:01	100	00:01	100	00:01
2	3	15	43	00:01	43	00:01	43	00:01	43	00:01	100	00:01	100	00:01
3	3	15	46	00:01	46	00:01	46	00:01	46	00:01	98	00:01	100	00:01
4	3	20	93	00:01	86	00:01	93	00:01	62	00:01	105	00:03	100	00:01
5	3	20	47	00:01	47	00:01	47	00:01	45	00:01	100	00:01	100	00:02
6	3	20	96	00:01	96	00:01	96	00:01	96	00:01	100	00:01	100	00:01
7	3	25	52	00:01	52	00:01	52	00:01	36	00:01	83	00:01	100	00:02
8	3	25	41	00:01	41	00:01	41	00:01	41	00:01	102	00:04	100	00:01
9	3	25	92	00:01	86	00:01	92	00:01	78	00:01	98	00:02	100	01:12
10	5	15	38	00:04	38	00:01	38	00:01	37	00:01	97	00:53	100	00:01
11	5	15	36	00:04	34	00:01	32	00:01	33	00:01	97	00:37	100	00:01
12	5	15	44	00:04	41	00:01	44	00:01	42	00:01	105	00:37	100	00:01
13	5	20	60	00:05	58	00:01	60	00:01	55	00:01	104	01:41	100	00:02
14	5	20	60	00:05	57	00:01	60	00:01	57	00:01	99	00:33	100	00:04
15	5	20	46	00:05	46	00:01	46	00:01	45	00:01	97	00:32	100	00:04
16	5	25	59	00:07	57	00:01	59	00:01	54	00:01	97	01:17	100	00:03
17	5	25	39	00:07	39	00:01	39	00:01	39	00:01	100	02:00	100	00:04
18	5	25	38	00:07	38	00:01	38	00:01	33	00:01	97	00:46	100	00:03
19	7	15	38	01:21	34	00:01	37	00:01	35	00:01	101	04:03	100	00:05
20	7	15	28	01:25	25	00:01	27	00:01	23	00:01	98	14:11	100	02:02
21	7	15	33	01:22	32	00:01	32	00:01	32	00:01	97	09:28	100	00:04
22	7	20	58	01:54	48	00:01	58	00:01	52	00:01	98	17:14	100	00:03
23	7	20	33	01:55	27	00:01	31	00:01	24	00:01	96	18:00	100	00:08
24	7	20	36	01:50	36	00:01	36	00:01	31	00:01	107	26:41	100	00:05
25	7	25	40	02:27	40	00:01	40	00:01	32	00:01	100	17:13	100	00:05
26	7	25	41	02:25	41	00:01	41	00:01	37	00:01	105	08:19	100	00:07
27	7	25	36	02:23	36	00:01	36	00:01	36	00:01	96	15:51	100	00:13

3 БАГАТОКАНАЛЬНА ФІКСОВАНА БЕЗДРОТОВА ЛОКАЛЬНА МЕРЕЖА: ПРОБЛЕМА ПРОЕКТУВАННЯ

3.1 Оптимізаційні моделі

У цьому розділі описано методи розв'язання задачі проектування багатоканальної фіксованої бездротової локальної мережі. У цьому розділі я буду використовувати модель максимізації пропускну здатності для розміщення точок доступу в одноповерховій будівлі, припускаючи, що доступні кілька бездротових каналів, які перекриваються або не перекриваються, і що розташування точок доступу є фіксованим і відомим. На відміну від інших моделей, що використовуються для розміщення точок доступу, я не вимагаю набору потенційних місць розташування точок доступу і дозволяю розміщувати точки доступу в будь-якому місці будівлі. Використовуючи формулу пропускну здатності Шеннона [2] для оцінки сигналу та перешкод між точками доступу та MD. Я також буду використовувати програмне забезпечення для реалізації даної моделі і представлю емпіричний аналіз, в якому порівняю нову реалізацію з кількома сітковими метаевристиками, а також з іншим нелінійним розв'язувачем.

У розділі 2 я представляю оптимізаційну модель для вирішення одноканальної версії задачі проектування фіксованої бездротової локальної мережі. Ця модель припускає, що не існує мінімальної вимоги до SIR [відношення сигнал-помилка] для зв'язку, і що кожна MD призначається точці доступу з найсильнішим сигналом. Алгоритм визначення розділів A_s можна знайти в цій попередній роботі, а початковий розділ розміщує MD в компактні групи. Крім того, використовується процедура попередньої обробки для отримання набору прямокутників, що не перетинаються, в яких можуть бути розташовані точки доступу. За цих обмежень множини,

обмеження та змінні для одноканальної моделі наведені в таблиці 2.1 і модель проста:

$$\text{Максимізувати } \zeta = \sum_{m \in M} [\log(1 + t_{hmm}/u_{hmm})/v_m], \quad (3.1)$$

за умови:

$$u_{hmm} = \sum_{s \in S \setminus \{h_m\}} P_s g_{sm} + N_m \quad \forall m \in M \quad (3.2)$$

$$t_{hmm} = P_{h_m} g_{h_m m} \quad \forall m \in M \quad (3.3)$$

$$d_{sm} = \sqrt{(x_s - \theta_m)^2 + (y_s - \phi_m)^2 + C^2} \quad \forall s \in S, m \in M \quad (3.4)$$

$$g_{sm} = 10^{-1.46} d_{sm}^{-3.27} \quad \forall s \in S, m \in M \quad (3.5)$$

$$v_m = |A_s| \quad \forall s \in S, m \in A_s \quad (3.6)$$

$$x_s^{\min} \leq x_s \leq x_s^{\max} \quad \forall s \in S \quad (3.7)$$

$$y_s^{\min} \leq y_s \leq y_s^{\max} \quad \forall s \in S \quad (3.8)$$

$$d_{sm} \leq d_{h_m m} \quad \forall s \in S, m \in M \quad (3.9)$$

Зауважимо, що (3.9) є набором квадратичних обмежень; отже, (3.3)-(3.9) є нелінійною математичною програмою з нелінійною цільовою функцією та лінійними і квадратичними обмеженнями. Поширюючи (3.3)-(3.9) на багатоканальний випадок, вводячи нові змінні, κ_s , що позначає призначення каналу для точки доступу s , і $\delta_{sr} \geq 0$, що позначає відсоток перекриття каналів, призначених точкам доступу s і r . Використовуючи нові змінні, (3.2) замінюється на наступний вигляд:

$$u_{hmm} = \sum_{s \in S \setminus \{h_m\}} P_s g_{sm} \delta_{sh_m} + N_m \quad \forall m \in M \quad (3.10)$$

яка враховує перекриття каналів у термінах завад. Тобто, U_{hmt} позначає інтерференцію між MDm і всіма точками доступу, крім hm, в одиницях ват. Замість того, щоб обчислювати інтерференцію між MDm і всіма іншими точками доступу, тепер обчислюємо інтерференцію між m і точками доступу, чий канал перекривають канал точки доступу, якій призначено m. Крім того, наступні обмеження визначають відповідні значення для відсотка перекриття:

$$\delta_{sr} \geq 1 - |\kappa_s - \kappa_r| \gamma \quad \forall s, r \in S | s \neq r \quad (3.11)$$

$$\delta_{sr} \geq 0 \quad \forall s, r \in S | s \neq r \quad (3.12)$$

де γ – швидкість зменшення перекриття між сусідніми каналами. Лінійна апроксимація I-фактора [10] дає значення γ 0,29. Це значення було визначено за допомогою регресії для апроксимації функції, що визначається значеннями, наведеними на рис. 6а ([10]), яка пов'язана з добре відомою характеристичною кривою втрат потужності для частот перекриття. Значення для δ_{sr} при $\gamma = 0,29$ проілюстровано в таблиці 3.1. Слід зазначити, що це лінійна апроксимація нелінійної функції, а також слід зазначити, що завади, визначені в (3.2), відрізняються від моделі Mishra та ін. Модель яку я використовую, методи глобальної оптимізації для розміщення точок доступу та призначення частот, тоді як Mishra та ін. використовують евристику та моделювання. Обидва дослідження розглядають одне і те ж питання щодо підвищення пропускної здатності, яке може бути досягнуте за рахунок часткового перекриття каналів. У України існує одинадцять різних потенційних каналів, причому різниця між каналами, що не перекриваються, становить щонайменше п'ять. Тому канали 1, 6 і 11 використовуються, коли потрібно вибрати три канали, що не перекриваються. Таким чином, проблема з каналами, що не перетинаються, має $\kappa_s \in \{1, 6, 11\} \quad \forall s \in S$ (3.13a),

а задача з каналами, що перекриваються, має $k_s \in \{1, \dots, 11\} \forall s \in S$ (3.13b)

Таблиця 3.1 – Розрахунок перекриття каналів (δ_{sr}) з використанням $\gamma = 0.29$

k_s	k_r										
	1	2	3	4	5	6	7	8	9	10	11
1	1	0.71	0.42	0.13							
2	0.71	1	0.71	0.42	0.13						
3	0.42	0.71	1	0.71	0.42	0.13					
4	0.13	0.42	0.71	1	0.71	0.42	0.13				
5		0.13	0.42	0.71	1	0.71	0.42	0.13			
6			0.13	0.42	0.71	1	0.71	0.42	0.13		
7				0.13	0.42	0.71	1	0.71	0.42	0.13	
8					0.13	0.42	0.71	1	0.71	0.42	0.13
9						0.13	0.42	0.71	1	0.71	0.42
10							0.13	0.42	0.71	1	0.71
11								0.13	0.42	0.71	1

Оновлені моделі, максимізація ζ за умови (3.3)-(3.12), (3.13a)/(3.13b) є цілочисельними нелінійними задачами оптимізації, тоді як максимізація ζ за умови (3.3)-(3.9) є неперервною задачею. Нові моделі можна використовувати для трьох каналів, що не перетинаються, або для одинадцяти каналів, що перетинаються. Зауважу, що моделі мають нелінійні цільові функції, лінійні, квадратичні та абсолютні обмеження, а також включають деякі змінні, які повинні набувати цілих значень. Отже, пошук оптимального рішення, яке можна довести, може бути досить складним і трудомістким.

3.2 Алгоритми та програмні реалізації

У цьому розділі я також буду використовувати алгоритми для реалізації нових моделей, описаних у розділі 3.1. Алгоритм для реалізації моделі (3.1)-(3.9) можна знайти в Розділі 2, і він передбачає розбиття MD за місцем розташування. Для кожного розбиття я застосовую нелінійний розв'язувач, який повертає хороший локальний оптимум. Найкращий з цих розв'язків використовується як остаточний розв'язок. Ця процедура не гарантує отримання глобального оптимуму. Однак тестування на невеликих задачах (п'ять AP і 15 MD) з відомими оптимальними розв'язками завжди давало розв'язки в межах 10% від оптимуму. Нові програмні реалізації базуються на цьому алгоритмі. Слід зазначити що їх розробив мій добрий друг, тому я мало що знаю в тим як вони працюють. Тож в мене є три нові програмні реалізації: дві для задачі призначення каналів і розміщення точок доступу, що не перетинаються («Не перет А» і «Не перет Б»), і одна для задачі призначення каналів, що перетинаються («Перет А»). Ці три програмні реалізації вимагають, щоб множини A_s , $\forall s \in S$ були визначені до розв'язання задачі максимізації ζ відповідно до (3.3)-(3.12), (3.13a)/(3.13b). Визначимо розбиття як множину множин $\{A_1, \dots, A_{|S|}\}$. Модель, розв'язана методами «Не перет А» та «Не перет Б», має цілочисельні змінні $|S|$, які можуть набувати одного з трьох можливих значень. Через вилучення деяких потенційних призначень каналів через симетрію, повний перебір передбачає розв'язання не більше $3|S|-1$ нелінійних неперервних задач. Для одного поверху будівлі площею приблизно 30 000 мет² повний перебір для призначення каналів може бути виконаний за розумну кількість комп'ютерного часу. Таким чином, заданий екземпляр «Не перет А» – це деякий розділ і призначення каналів для $|S|$ точок доступу. Пропускна здатність даного екземпляра визначається як розв'язок задачі ζ з урахуванням (3.3)-(3.12). Неперервні задачі розв'язуються за допомогою NLP-розв'язувача

LINDO в пакеті LINDOAPI (LINDO, 2011). Як «Не перет Б», так і «Перет А» використовують алгоритм з розгалуженням для визначення оптимального розміщення та частотного розподілу точок доступу. Безперервні релаксації для цих двох алгоритмів з гілками і межами використовують

$$1 \leq k_s \leq 11, s \in S, \quad (3.14)$$

замість (3.13a) або (3.13b). Для «Не перет Б» значення k_s , яке є нецілим (або цілим, таким, що $k_s \in \{ / 1, 6, 11 \}$), призведе до створення двох підзадач. Наприклад, якщо $k_s=3.3$, то буде створено одну підзадачу з $k_s = 1$, і другу підзадачу з $k_s \geq 6$. Для «Перет А» ті ж самі $k_s = 3.3$ призведуть до створення двох підзадач з $k_s \leq 3$ і $k_s \geq 4$. Неперервні релаксації, що виникають в результаті, розв'язуються за допомогою LINDOAPI.

3.2.1 Емпіричний аналіз

У емпіричному аналізі проаналізував кілька наборів даних. По-перше, визначив, який з набору метаевристик дає найкращі результати, а також найкращі параметри. Після того, як завершив метаевристичний аналіз, порівняв результати найкращої метаевристики з програмними реалізаціями. Результати представлені в наступних розділах.

3.2.2 Аналіз метаевристик

Як описано в роботі Кеннінгтона та ін. [20], чотири процедури були реалізовані в програмному забезпеченні та емпірично оцінені на ділянці розміром 175 метрів x 175 метрів, як показано на рисунку 2.4. Процедура пошуку за табу і генетичні алгоритми були закодовані разом з простим пошуком по сусідству. У своїй роботі, Kennington та ін. [20] визначили, що

для невеликих задач метаевристика загалом повертала розв'язки, близькі до оптимальних. Однак було невідомо, як зміна точності сітки для потенційних місць розташування точок доступу вплине на остаточний розв'язок. Тому вони запустили кілька задач з різною кількістю точок сітки і порівняли перезавантаження. Для кожної задачі значення $|S|$ фіксоване, так само як і потужність для кожної точки доступу. Таким чином, отримана модель має лінійні обмеження і нелінійну цільову функцію. У всіх тестових випадках використовувалися три частоти, що не перекриваються. Розміри задач наведено в таблиці 3.2, а результати для генетичного, табульованого та сусіднього алгоритмів пошуку – в таблиці 3.3. Результати масштабуються до ємності, визначеної генетичним алгоритмом з 34 потенційними місцями розташування точок доступу. Крім того, розташування 15 MD в задачі 1 проілюстровано на рисунку 3.1.

Таблиця 3.2 – Список проблем для метаевристичного аналізу

Номер	AP	MD
1	5	15
2	5	20
3	5	25
4	7	15
5	7	20
6	7	25
7	9	15
8	9	20
9	9	25

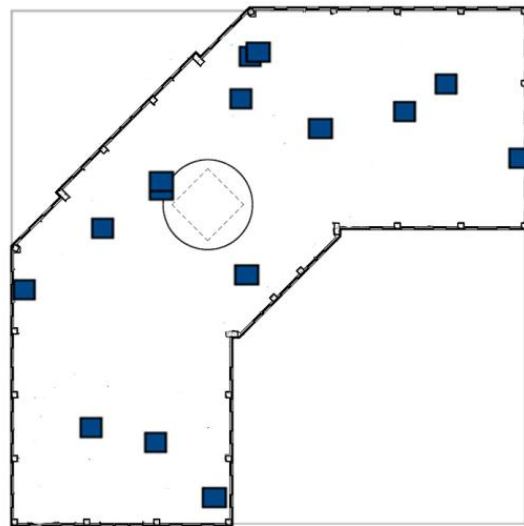


Рисунок 3.1 – Місця знаходження даних MD для проблеми 1

Використовуючи Дисперсійний аналіз [далі ANOVA] для аналізу цих результатів, я визначив, що як використаний алгоритм, так і кількість

потенційних місць розташування точок доступу суттєво впливають на пропускну здатність, тому рішення з Р-значеннями $\leq 0,001$ та $0,012$ відповідно. Оскільки Р-значення вказує на ймовірність побачити дані, розподілені подібно до моїх вихідних даних, враховуючи гіпотезу про те, що всі середні рівні, а Р-значення $\alpha = 0,05$ або нижче вказує на статистично значущу різницю, ці результати є дуже статистично значущими, що дозволяє відкинути нульову гіпотезу про те, що всі алгоритми повинні повертати однакове середнє значення. Взаємозв'язок між алгоритмом A1 та кількістю точок не є значущим на рівні $\alpha = 0,05$ з Р-значенням $0,032$. Що стосується часу, то єдиний значущий внесок був зроблений алгоритмом, який використовувався, з Р-значенням $\leq 0,001$. Графіки основних ефектів для пропускну здатності показані на рисунку 3.2, а графік взаємодії – на рисунку 3.3. Таким чином, я зміг визначити, що для цього набору тестових завдань найкращі рішення дає пошук по сусіднім районам з 165 потенційними місцями розташування точок доступу. Тому в наступному розділі я використаю результати пошуку по сусідству для порівняння з моделями оптимізації.

Таблиця 3.3 – Масштабовані потужності для метаевристик

Номер проблеми	34 Точки сітки			64 Точки сітки			99 Точок сітки		
	Genetic	Tabu	Neighbor	Genetic	Tabu	Neighbor	Genetic	Tabu	Neighbor
1	100	115	109	108	110	118	111	109	154
2	100	104	83	105	98	88	71	76	87
3	100	110	79	104	101	108	104	118	120
4	100	102	101	106	113	102	102	107	101
5	100	106	102	107	136	123	108	124	156
6	100	118	106	101	111	116	108	118	127
7	100	74	63	103	88	92	110	82	81
8	100	98	91	104	114	95	106	110	117
9	100	95	84	73	91	86	74	94	111
Номер проблеми	124 Точки сітки			165 Точок сітки			196 Точок сітки		
	Genetic	Tabu	Neighbor	Genetic	Tabu	Neighbor	Genetic	Tabu	Neighbor
1	107	72	128	113	148	152	110	105	155
2	75	76	89	85	80	107	68	74	103
3	104	73	104	104	84	113	77	79	97
4	105	90	111	109	103	143	111	105	133
5	109	131	128	113	114	152	119	116	173
6	102	96	125	113	117	152	110	73	134
7	110	85	68	108	94	89	106	89	94
8	110	114	110	108	108	131	103	96	129
9	73	90	97	79	110	119	80	99	122

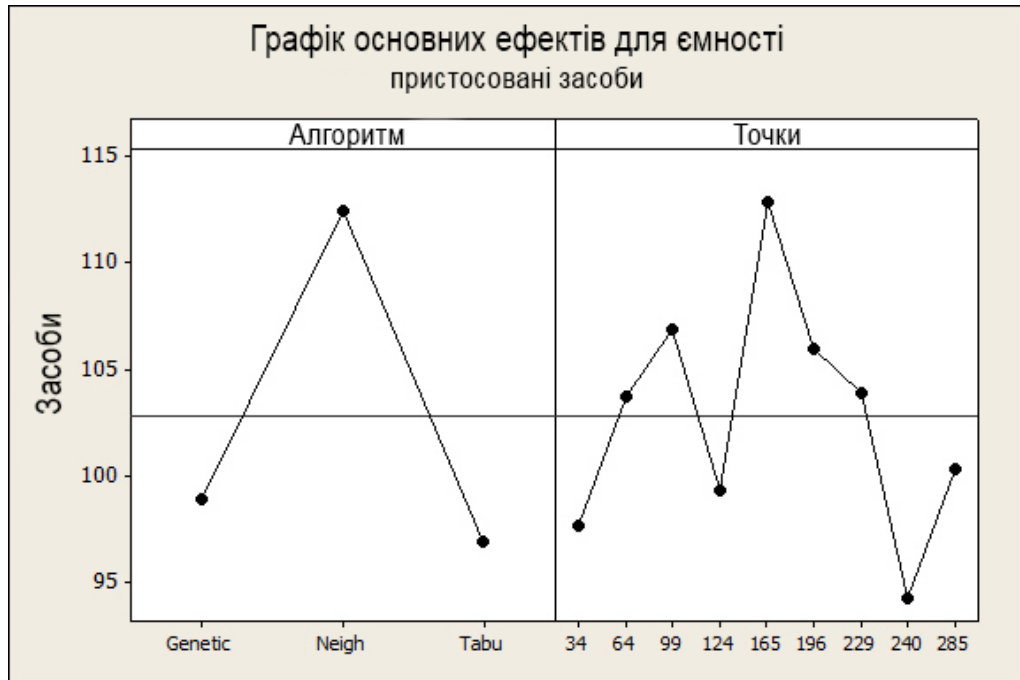


Рисунок 3.2 – Основні ефекти для пропускної здатності порівняно з алгоритмом та кількістю використаних точок

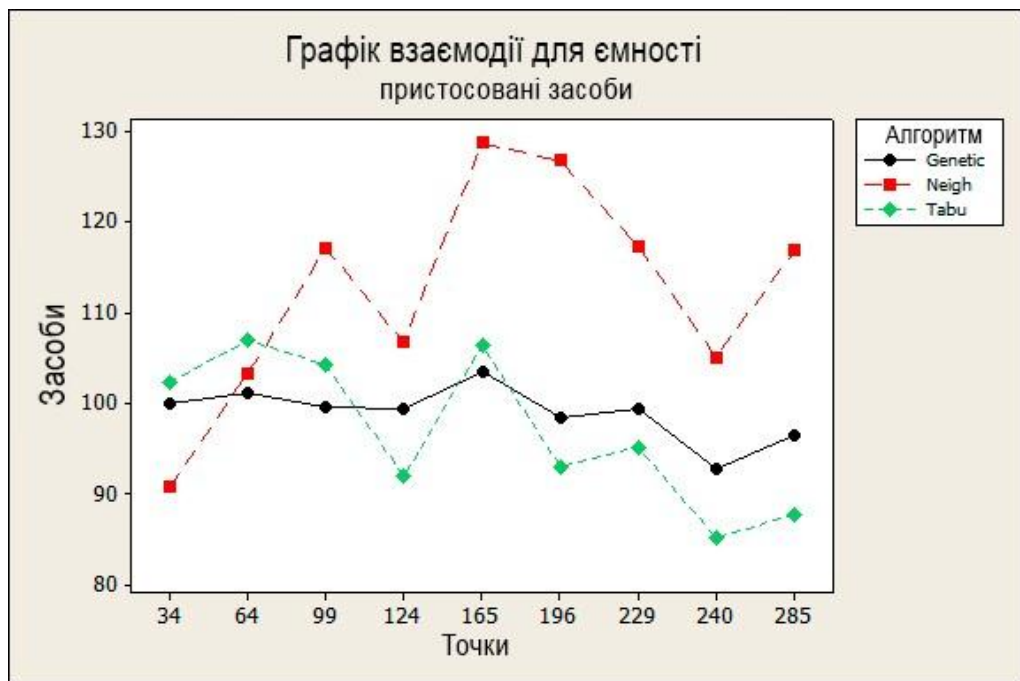


Рисунок 3.3 – Графік залежності пропускної здатності від взаємодії алгоритму/кількості точок

3.2.3 Аналіз нових програмних реалізацій

У цьому розділі я порівняю результати трьох реалізацій, «Не перет А», «Не перет Б» та «Перет А», з можливостями, визначеними за допомогою пошуку околиць у задачі Офісної будівлі. Я протестував «Не перет Б», використовуючи $\gamma = 0.29$, який було отримано з лінійної апроксимації даних, знайдених в Mishra та ін. (2006) [ТУТ].

Результати порівняння всіх реалізацій можна знайти в таблиці 3.4. Я розв'язував ці моделі, використовуючи Java для попередньої обробки та LINDOAPI для оптимізації. Всі потужності в таблиці 3.4 масштабовано до потужності, отриманої в результаті розв'язання «Не перет А» для всіх n-пар частот-кандидатів. Для кожної реалізації на кожну задачу було встановлено часовий ліміт у 2 години.

Очевидно, що «Не перет А» значно перевершує навіть найкращі метаевристики, що використовують фіксовані потенційні місця розташування точок доступу для цього набору даних. Це не дивно, оскільки будь-яке рішення, знайдене метаевристичними алгоритмами, буде знаходитися в просторі рішень «Не перет А». «Не перет Б» і «Перет А» надають можливості, які майже такі ж хороші, як і «Не перет А», але за менший час. Однак, «Перет А» забезпечив більшу пропускну здатність, ніж «Не перет А» у кількох випадках.

Це підтверджує результати моделювання Mishra та ін. (2006), що в деяких випадках використання частот, що перекриваються, може забезпечити більшу пропускну здатність бездротового зв'язку, ніж використання частот, що не перекриваються. Цікаво, що це покращення відбувається лише тоді, коли використовується найменша кількість точок доступу. Це також узгоджується з висновками Mishra та ін. (2006), які стверджують, що перекриття частот має бути корисним лише тоді, коли точки доступу з частотами, що перекриваються, знаходяться на значній відстані одна від

одної, майже поза межами діапазону передачі. Коли додаткові точки доступу розміщуються в обмеженому просторі, доступному в моєму прикладі, перекриття частот не покращує пропускну здатність мережі.

Проаналізувавши результати за допомогою ANOVA, визначив, що єдиним фактором, який впливає на пропускну здатність при масштабуванні, був використаний алгоритм з Р-значенням $< 0,01$. Однак, оскільки я масштабував потужності до потужності, знайденої за допомогою «Не перет А», це очікувано. Крім того, парний тест Тьюкі для кожної комбінації алгоритмів не виявив суттєвої різниці між результатами пропускну здатності «Перет А» та «Не перет Б», а також сильної різниці між усіма іншими парами алгоритмів з Р-значенням $< 0,01$. На жаль, це означає, що «Не перет Б» і «Перет А» дають гірші результати, ніж «Не перет А». Однак, враховуючи, що «Не перет Б» і «Перет А» працюють набагато швидше, «Не перет Б» і «Перет А» є кращим вибором при збільшенні розміру проблеми.

Таблиця 3.4 – Масштабовані можливості для всіх реалізацій

Проблема			«Не перет Б»		«Перет А»		«Не перет А»		Neighborhood	
No	APs	MDs	Ємність	Час	Ємність	Час	Ємність	Час	Ємність	Час
1	5	15	92	00:21	85	00:06	100	00:00:57	51	00:00:01
2	5	15	91	00:12	89	00:46	100	00:01:14	51	00:00:01
3	5	15	90	00:13	90	00:09	100	00:00:58	54	00:00:01
4	5	20	93	00:29	98	00:19	100	00:01:25	50	00:00:01
5	5	20	88	00:28	104	00:48	100	00:01:01	52	00:00:01
6	5	20	94	00:18	112	00:27	100	00:00:52	51	00:00:01
7	5	25	86	00:16	96	00:34	100	00:02:04	54	00:00:01
8	5	25	94	00:11	111	00:47	100	00:01:54	53	00:00:01
9	5	25	92	00:30	66	00:12	100	00:02:53	52	00:00:01
10	7	15	92	00:54	85	00:35	100	00:08:37	53	00:00:02
11	7	15	91	00:27	84	00:37	100	00:06:39	48	00:00:03
12	7	15	91	00:42	84	00:33	100	00:05:42	52	00:00:02

Кінець таблиці 3.4

Проблема			«Не перет Б»		«Перет А»		«Не перет А»		Neighborhood	
No	APs	MDs	Ємність	Час	Ємність	Час	Ємність	Час	Ємність	Час
13	7	20	85	00:39	90	00:42	100	00:10:49	51	00:00:03
14	7	20	92	00:45	94	00:45	100	00:15:34	54	00:00:04
15	7	20	92	00:31	89	02:53	100	00:08:07	51	00:00:03
16	7	25	86	00:41	98	00:51	100	01:11:23	50	00:00:04
17	7	25	88	04:04	92	01:06	100	00:36:59	50	00:00:06
18	7	25	79	01:16	92	01:22	100	00:29:32	56	00:00:08
19	9	15	85	01:03	78	01:27	100	02:00:00	51	00:01:30
20	9	15	84	01:21	94	00:51	100	02:00:00	47	00:01:41
21	9	15	92	01:03	79	00:54	100	02:00:00	60	00:01:46
22	9	20	93	01:18	78	01:21	100	02:00:00	47	00:01:29
23	9	20	86	02:09	87	01:11	100	02:00:00	51	00:02:17
24	9	20	87	00:40	78	01:18	100	02:00:00	53	00:02:13

Я також відзначаю, що «Не перет А» порушив часові обмеження для всіх задач з дев'ятьма точками доступу, я очікую, що «Не перет А» буде складніше знайти гарний розв'язок за розумний проміжок часу, якщо задачі стануть ще більшими. За винятком кількох викидів, залишки були приблизно нормальними, як показано на рисунку 3.4. Це вказує на те, що модель, визначена за допомогою ANOVA, є потенційно точною, оскільки залишки (відхилення від лінійної моделі для кожної точки даних) повинні слідувати нормальному розподілу відповідно до припущень, які вимагає ANOVA.

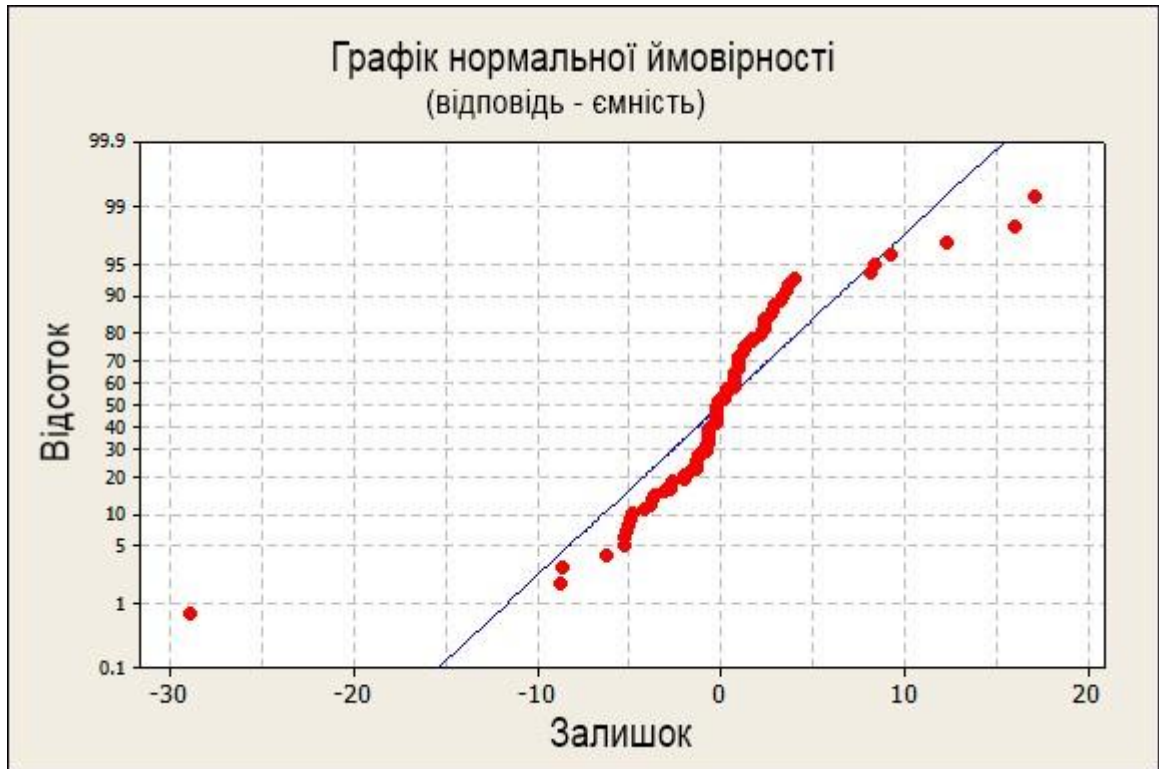


Рисунок 3.4 – Нормограма залишків для потужності

Крім того, обмеживши аналіз лише макетами з п'ятьма точками доступу, виявив за допомогою тесту Тьюкі, що не було суттєвої різниці між «Перет А», «Перет Б» та «Не перет А». Оскільки «Перет А» покращив лише деякі результати визначені за допомогою «Не перет А», це очікувано. Однак, не можна сказати, що «Перет А» покращує результати, отримані за допомогою «Не перет А». Є кілька пояснень, чому «Перет А» не перевершує «Не перет А» у всіх випадках з п'ятьма точками доступу. По-перше, дуже великий розмір допустимої області ($11|S|$ потенційних розв'язків проти $3|S|-1$ потенційних розв'язків) робить пошук глобального оптимуму дуже складним. По-друге, розміщення MD може обмежувати місця розміщення точок доступу для забезпечення оптимальної пропускної здатності. Якщо MD у даному плані є більш централізованими, збільшення потенційної пропускної здатності, створене за рахунок використання частот, що перекриваються, може бути перекрите зменшенням пропускної здатності через вимушене

розміщення точок доступу на великій відстані від MD, щоб задовольнити вимоги щодо використання переваг від перекриття. Факторами, які виявилися значущими при аналізі часової продуктивності за допомогою ANOVA, були використаний алгоритм, кількість точок доступу та взаємодія між алгоритмом і точками доступу, всі з Р-значеннями $< 0,01$. Однак ANOVA може бути не найкращим інструментом для аналізу цього конкретного набору даних, оскільки дані не виглядають нормально розподіленими, як показано на рисунку 3.5. Частково це може бути пов'язано з тим, що я встановив довільний часовий проміжок у дві години. Незважаючи на це, очевидно, що збільшення кількості точок доступу в макеті значно збільшує час, необхідний алгоритмам для завершення роботи. Також зрозуміло, що «Перет А» і «Не перет Б» працюють значно швидше, ніж «Не перет А». Оскільки «Перет А» і «Не перет Б» дають результати, які статистично не відрізняються від результатів «Не перет А» при малій кількості точок доступу, і значно швидші при великій кількості точок доступу, рекомендовано використовувати алгоритми.

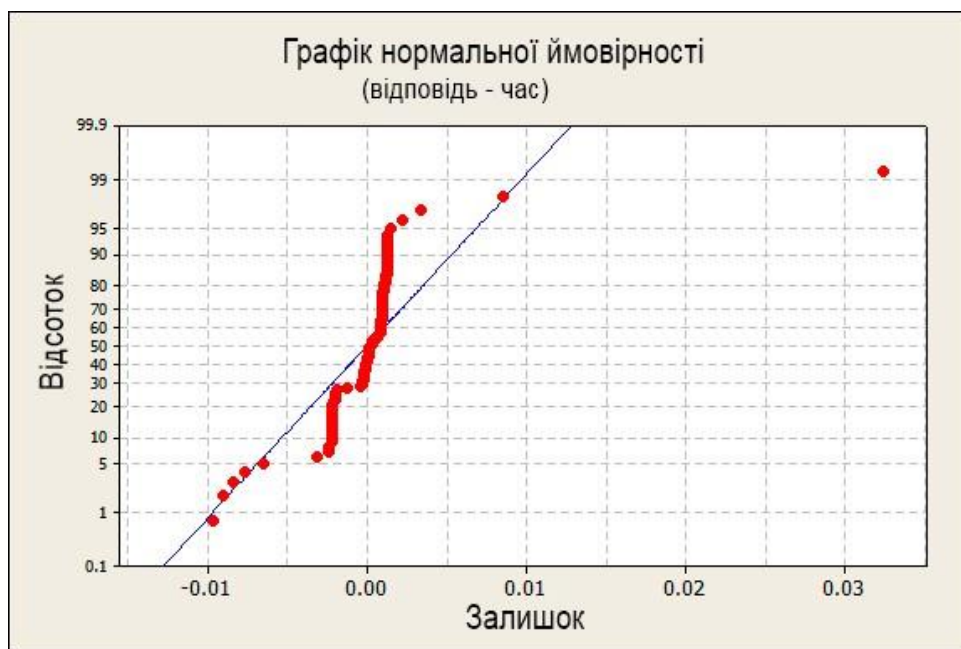


Рисунок 3.5 – Графік залежності залишків від часу

3.3 Підсумки та висновки

У цій главі представлено дослідження проблеми проектування фіксованої бездротової локальної мережі з декількома частотами, що перекриваються і не перекриваються. Для фіксованої задачі розташування всіх мобільних пристроїв (MD) відоме, і проблема полягає у визначенні розташування всіх точок доступу (AP) і всіх частотних присвоєнь. Модель допускає розташування точок доступу в будь-якому місці в межах поверху задачі проектування. Ця модель має комплексну нелінійну цільову функцію, лінійні та квадратичні обмеження, а також неперервні та цілі змінні. Визначивши, що програмна реалізація алгоритму значно перевершує метаевристику з фіксованим набором потенційних місць розташування точок доступу, а також визначивши, що в деяких випадках використання частот, що перекриваються, може забезпечити кращі результати щодо пропускної здатності, ніж використання частот, що не перекриваються. Рекомендовано використовувати алгоритми «Не перет Б» і «Перет А» для забезпечення високої якості макетів WLAN з практично невеликим часом обчислень.

4 ПРОБЛЕМА ПРОЕКТУВАННЯ КОЧОВОЇ БЕЗПРОВІДНОЇ ЛОКАЛЬНОЇ МЕРЕЖІ

4.1 Оптимізаційні моделі

У цьому розділі описано методи вирішення проблеми проектування багатоканальної кочової бездротової локальної мережі. У цьому розділі я буду використовувати модель максимізації пропускної здатності для розміщення точок доступу в одноповерховій будівлі, припускаючи, що доступні кілька бездротових каналів, які перекриваються або не перекриваються, і що існує набір потенційних сценаріїв розташування точок доступу, які виникають з певною ймовірністю. Також був представлений емпіричний аналіз декількох цільових функцій з використанням нової реалізації.

У розділі 2 та розділі 3 представлено оптимізаційні моделі для вирішення одноканальної версії задачі проектування фіксованої бездротової локальної мережі та багатоканальної версії задачі проектування фіксованої бездротової локальної мережі. Ці моделі припускають, що для зв'язку не існує вимог до мінімального SIR(відношення сигнал–помилка), і що кожна MD призначається точці доступу з найсильнішим сигналом, як правило, найближчій точці доступу. Для того, щоб спростити задачу, фіксуємо призначення точок доступу для кожного MD, розбиваючи MD на групи за призначенням, точок доступу. Алгоритм визначення розбиття можна знайти в Розділі 2, і початкове розбиття розміщує MDs в компактні групи. Однак, у кочовій задачі розташування MD не відоме апіорі, і його потрібно оцінити певним чином. Я буду використовувати оптимізаційну модель, яка використовує декілька потенційних сценаріїв, кожен з яких є екземпляром фіксованої задачі. Тому для кожного сценарію $r \in R$, де R – множина всіх

сценаріїв, мені потрібне розбиття. Ці розбиття позначимо через Ars , де s – це AP , якому призначено MD у розбитті, і вони визначаються так само, як і розбиття у фіксованій задачі. Потім проведу оптимізацію за всіма сценаріями одночасно і знайду єдиний план розміщення всіх точок доступу, який забезпечить хорошу загальну пропускну здатність. Крім того, використовується процедура попередньої обробки для отримання набору прямокутників, що не перетинаються, в яких можна розмістити точки доступу. Оскільки тепер я повинен оптимізувати за кількома сценаріями, введемо множину сценаріїв (R). Кожен сценарій $r \in R$ позначає набір MD , а також кілька пов'язаних з ним змінних і констант. Хоча ці змінні та константи подібні до тих, що використовувались у Розділі 2, існує достатньо відмінностей, щоб переформулювати їх тут. Відповідно до цих обмежень, набори, константи та змінні для цієї моделі наведені в Таблиці 4.1, а обмеження є наступними:

$$u_m^r = \sum_{s \in S \setminus \{h_{mr}\}} P_s g_{sm}^r \delta_{sh_{mr}}^r + N_m^r \quad \forall m \in M^r, r \in R \quad (4.1)$$

де u_m^r – сумарна завада між MD_m і всіма точками доступу, яким вона не призначена і які знаходяться на одній частоті у всіх сценаріях.

$$t_m^r = P_{h_{mr}} g_{h_{mr}m}^r \quad \forall m \in M^r, r \in R \quad (4.2)$$

де t_m^r – сумарний сигнал між кожним MD і точкою доступу, до якої його призначено в кожному сценарії.

$$d_{sm}^r = \sqrt{(x_s - \theta_m^r)^2 + (y_s - \phi_m^r)^2 + C^2} \quad \forall s \in S, m \in M^r, r \in R \quad (4.3)$$

який обчислює відстань між кожною точкою доступу та кожним MD у кожному сценарії.

Таблиця 4.1 – Множини, константи та змінні, що використовуються в кочовій моделі

Ім'я	Тип	Опис
S_R	множина	Набір точок доступу
M_r	множина	Набір сценаріїв
A_r	множина	Множина всіх MD у сценарії $r \in R$
s	множина	Набір усіх MD, пов'язаних з AP; $A_{rsc} \subset M_r$
θ_r	константа	Розташування MD_m у r -му сценарії на осі x
m	константа	Розташування MD_m у r -му сценарії на осі y .
ϕ_r	константа	Висота стелі – висота MD.
m	константа	Мінімальна x -координата для розміщення AP.
C	константа	Максимальна координата x для розміщення AP.
x_{min}	константа	Мінімальна y -координата для розміщення AP.
s	константа	Максимальна y -координата для розміщення AP.
x_{max}	константа	Шум, пов'язаний з MD_m у r -му сценарії.
s	константа	Налаштування потужності для APs.
y_{min}	константа	Швидкість зменшення перекриття між сусідніми каналами;
s	константа	$=0,29$
y_{max}	константа	Вага сценарію r
s		Розрахована максимальна потужність для сценарію r
N_r		
m		
$P_{sy} W_r$		
Ω_r		

Кінець таблиці 4.1

Ім'я	Тип	Опис
x_s	змінна	Розташування APs на осі абсцис
y_s	змінна	Розташування APs на осі y
δ_{st}	змінна	Призначення каналів APs
g_r	змінна	Відсоток перекриття між точками доступу sit
sm	змінна	Загасання між APs та MDm у r-му сценарії
d_r	змінна	Відстань між AP s та MD m у r-му сценарії AP, якій призначено
hmr	змінна	MDm у r-му сценарії; $hmr \in S$.
ur	змінна	Загальні перешкоди та шум на MDm у сценарії r.
m		Сигнал між MDm і точкою доступу, якій він призначений у сценарії r.
tr		Кількість MD, призначених AP, якій призначено MDm у сценарії r.
m		Розрахункова потужність сценарію r; $=f(x, y, \kappa, M_r)$.
v_r		
m		
ζ_r		

$$g_{sm}^r = 10^{-1.46 d_{sm}^r - 3.27} \quad \forall s \in S, m \in M^r, r \in R \quad (4.4)$$

Яка є мірою загасання між кожною точкою доступу і кожною точкою передачі в кожному сценарії.

$$v_m^r = |A_s^r| \quad \forall s \in S, m \in A_s^r, r \in R \quad (4.5)$$

Яка фактично є константою після того, як $A_g s$ стає постійною, і є загальною кількістю MD, призначених для кожної точки доступу в кожному сценарії.

$$x_s^{min} \leq x_s \leq x_s^{max} \quad \forall s \in S \quad (4.6)$$

$$y_s^{min} \leq y_s \leq y_s^{max} \quad \forall s \in S \quad (4.7)$$

Які є верхньою та нижньою межами для розміщення кожної точки доступу.

$$d_{sm}^r \leq d_{h_{mr}m}^r \quad \forall s \in S, m \in M^r, r \in R \quad (4.8)$$

Який змушує кожну точку доступу залишатись у регіоні таким чином, щоб жоден MD у жодному сценарії не був призначений іншій точці доступу, ніж та, що вже визначена Ars .

$$\delta_{st} \geq 1 - |\kappa_s - \kappa_t| \gamma \quad \forall s, t \in S | s \neq t \quad (4.9)$$

$$\delta_{st} \geq 0 \quad \forall s, t \in S | s \neq t \quad (4.10)$$

Які визначають загальну кількість перекриття частот між кожною парою точок доступу. Це дає пропускну здатність каналу Шеннона (Shannon, 1949) для кожного сценарію як:

$$\zeta^r = \sum_{m \in M^r} [\log(1 + t_m^r / u_m^r) / v_m^r] \quad \forall r \in R \quad (4.11)$$

де κ_s – обмеження для каналів, що не перетинаються, з:

$$\kappa_s \in \{1, 6, 11\} \quad \forall s \in S \quad (4.12a)$$

і проблема з перекриттям каналів має:

$$\kappa_s \in \{1, \dots, 11\} \quad \forall s \in S \quad (4.12b)$$

Крім того, кочова задача має багато потенційних цільових функцій, залежно від призначення мережі. Якщо потрібна висока середня пропускна здатність, то бажано максимізувати очікуване значення по всіх ζ_r , з наступною метою:

$$\text{maximize } \sum_{r \in R} W^r \zeta^r \quad (4.13a)$$

Якщо існує вимога ніколи не мати низької пропускної здатності, бажано максимізувати мінімальне значення ζ_r за всіма сценаріями, враховуючи наступну мету та додаткові обмеження:

$$\text{maximize } Z \quad (4.13b)$$

$$\text{subject to } Z \leq \zeta^r \quad \forall r \in R \quad (4.13c)$$

Можливо також, що потрібно мінімізувати жаль, з наступними об'єктивними та додатковими обмеженнями, а також з урахуванням теоретичної максимальної пропускної здатності для кожного сценарію:

$$\text{minimize } Y \quad (4.13d)$$

$$\text{subject to } Y \geq \Omega^r - \zeta^r \quad \forall r \in R \quad (4.13e)$$

Цільові функції не є ні увігнутими, ні опуклими, і розриваються в точці, де Цільова функція призначається іншій точці доступу. Хоча існують методи розв'язання неопуклих оптимізаційних задач, розриви роблять проблему майже неможливою для розв'язання за допомогою методів глобальної оптимізації (Pinter, 1996). Отже, поточна модель фіксує Arg перед кожною оптимізацією. Крім того, моделі мають лінійні, квадратичні та абсолютні обмеження, а також включають деякі змінні, які повинні набувати цілих значень.

4.2 Алгоритми та програмні реалізації

Наступні програмні реалізації базуються на роботі в розділах 2 та 3. Тепер я застосував надійні методи оптимізації до реалізації та алгоритмів, описаних у цих розділах. Є три програмні реалізації, по одній для кожної з моделей, обговорених у попередньому розділі. Ці програмні реалізації вимагають розташування всіх MD у всіх сценаріях, а також межі для всіх AP. Також я буду використовувати програму попередньої обробки, яка визначає всі множини Arg , $\forall s \in S$, $r \in R$ перед розв'язанням відповідних цільових функцій. Далі визначаємо розбиття як множину множин $\{A_1, \dots, A_{|R|} \mid |S|\}$. Розбиття визначаються шляхом створення нового сценарію, що містить всі MD для всіх сценаріїв, а потім визначення підмножин так, як якщо б існував лише один сценарій з багатьма MD. Наприклад, припустимо, що є три сценарії, кожен з яких містить три MD і дві AP. Дев'ять загальних MD будуть розділені на дві підмножини, потім кожна підмножина буде розділена на три менші підмножини залежно від того, в якому сценарії виникла MD. Після створення декількох ймовірних варіантів, кожне розбиття подається розв'язувачу разом з обмеженнями та цільовою функцією для задачі, що розв'язується. Найкращий розв'язок з усіх варіантів повертається як загальний розв'язок. Яка цільова функція буде використана, залежить від

потреб кінцевого користувача. Неперервні задачі розв'язуються за допомогою розв'язувача LINDONLP в пакеті LINDOAPI (LINDO, 2011).

4.3 Емпіричний аналіз

Реалізував ці процедури в програмному забезпеченні, і вони були емпірично оцінені на ділянці розміром 175 метрів x 175 метрів як показано на рисунку 2.4. Є 27 сценаріїв, дев'ять з 15 MD, дев'ять з 20 MD і дев'ять з 27 MD. У кожному сценарії MD були розміщені випадковим чином по всій структурі. Наприклад, розташування 15 MD в задачі 1 показано на рис. 3.1. Для кожного сценарію визначив теоретичний максимум (Ω_T) за допомогою програмних реалізацій, описаних у розділі 3. Для аналізу я використовував лише частоти, що не перекриваються. Для кожного експерименту визначив кількість використовуваних точок доступу (5, 7, 9). Потім вибрав випадкову кількість сценаріїв для оптимізації (3-6), а потім вибрав сценарії випадковим чином без повторень. Я оптимізував кожен експеримент, використовуючи всі три цільові функції. Для цього аналізу надав рівну вагу кожному сценарію і повернув розраховане середнє значення, мінімальну пропускну здатність і максимальний жаль для кожного знайденого рішення, а також час для кожного прогону. Результати були масштабовані до середнього значення Ω_T для всіх обраних сценаріїв, їх можна знайти в таблиці 4.2.

Таблиця 4.2 розділена на три секції, по одній для розв'язків, визначених з використанням кожної цільової функції. Перша секція містить значення, визначені з використанням цільової функції очікуваної вартості, друга секція містить значення, визначені з використанням цільової функції максимізації мінімальної пропускну спроможності, а остання секція містить значення, визначені з використанням цільової функції мінімізації максимального жалю. Кожен рядок таблиці містить кількість точок доступу, кількість сценаріїв, для яких проводилась оптимізація, та дані для кожної цілі. Перше значення в

кожній секції – це загальний час, необхідний для виконання оптимізації для цієї цілі, а наступні три значення – це середнє, мінімальне та максимальне значення жалю для цього прогону даних. Оскільки значення масштабуються до середнього значення Ω_g для всіх вибраних сценаріїв, значення, що повертаються, є відсотком від максимально можливого очікуваного значення, якби найкраще розташування точки доступу і призначення частоти були однаковими для всіх сценаріїв.

Таким чином, менші числа означають кращу продуктивність для максимальних значень, а більші числа означають кращу продуктивність для мінімальних і середніх значень. Проаналізувавши результати за допомогою ANOVA. Факторами, що використовувалися, були кількість точок доступу, кількість сценаріїв та мета, що оптимізувалася.

Таблиця 4.2 – Аналіз стійкої оптимізації

		Максимізація очікуваного значення				Збільшити мінімум				Мінімізація максимальної втрати			
A Ps	Сце- нарії	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)
5	4	00:01:18	85	82	21	00:01:23	80	77	24	00:02:12	83	79	20
5	4	00:00:51	78	68	27	00:00:29	75	66	30	00:04:04	78	68	27
5	5	00:02:04	73	62	40	00:01:42	73	62	40	00:01:53	73	62	40
5	4	00:01:12	76	71	29	00:01:33	76	71	29	00:01:29	82	79	22
5	6	00:01:48	80	66	32	00:01:13	81	73	34	00:04:52	80	66	32
7	4	00:12:10	68	66	43	00:05:37	62	48	47	00:10:02	61	56	44
7	3	00:11:40	75	70	41	00:14:27	76	71	40	00:18:44	70	60	37
7	5	00:03:36	61	34	68	00:00:11	61	43	60	00:06:31	63	50	52
7	3	00:03:53	76	64	29	00:04:35	76	64	29	00:11:08	74	60	33
7	3	00:04:35	75	61	40	00:03:33	74	65	37	00:00:13	68	59	43
7	6	00:12:07	64	44	56	00:06:09	65	55	48	00:00:03	65	49	51
9	3	00:27:06	66	58	38	00:30:42	66	56	40	00:47:04	70	68	36
9	5	00:43:33	72	51	43	00:34:36	58	45	64	01:32:21	66	55	40
9	4	00:24:41	71	49	47	01:13:21	72	63	44	04:40:20	75	59	37

Кінець таблиці 4.2

		Максимізація очікуваного значення				Збільшити мінімум				Мінімізація максимальної втрати			
A Ps	Сценарії	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)	Час	Серед (4.13a)	Мін (4.13b)	Втрати (4.13d)
9	4	01:00:16	69	59	42	02:47:42	69	64	38	02:51:38	69	64	38
9	5	00:42:19	68	58	37	00:21:25	62	61	37	00:07:45	65	62	33

Я провів окремий Дисперсійний аналіз для кожної залежної змінної (середня потужність, мінімальна потужність та максимальна потужність). Найбільшим джерелом варіабельності була кількість використаних AP. Для всіх трьох залежних змінних кількість використаних точок доступу була значущим фактором у поверненні потужності. Оскільки потужності були масштабовані до максимальної теоретичної очікуваної потужності (Ω_r), це означає, що з меншою кількістю точок доступу легше наблизитися до Ω_r . Крім того, кількість використаних сценаріїв була важливим фактором, оскільки менша кількість сценаріїв давала менші значення пропускну здатності. Це дивно, оскільки при меншій кількості сценаріїв логічно легше частково задовольнити всі з них. Однак, використана мета не дала статистично значущої різниці в жодному з наших вихідних параметрів. Це означає, що будь-яка з використаних реалізацій може бути використана для будь-якої з цілей, тобто, якщо ви хочете максимізувати мінімум, ви можете максимізувати очікуване значення, і отримати подібні результати. Це, ймовірно, пов'язано зі складністю задачі і нездатністю розв'язувача знайти загальний глобальний максимум або мінімум. Крім того, цілі не є повністю незалежними. Наприклад, рішення з великою мінімальною пропускну здатністю завжди матиме очікувану пропускну здатність, більшу за мінімальну, а за інших незмінних умов збільшення мінімального значення завжди збільшуватиме середнє значення.

Однак третій прогін даних з сімома точками доступу приніс

несподіваний результат. Максимальний регрет за найкраще знайдене рішення з використанням очікуваної цінності був надзвичайно високим, більшим за саму очікувану цінність. Слід також зазначити, що мінімум для цього конкретного експерименту також надзвичайно низький. Це означає, що один з п'яти сценаріїв показав особливо низьку пропускну здатність, але пропускну здатність інших чотирьох сценаріїв була достатньо високою, щоб забезпечити найкращу середню пропускну здатність з усіх знайдених. Це демонструє, як потреби кінцевого користувача можуть впливати на вибір цілі. Хоча статистично значущої різниці між цілями, використаними в сукупності, немає, в цьому конкретному випадку, а також, можливо, і в інших, використання цілі, яка не відповідає вимогам кінцевого користувача, може призвести до гірших результатів.

4.4 Підсумки та висновки

Я використав програмну реалізацію для задачі синхронізації кочової бездротової локальної мережі. Ця реалізація використовує методи оптимізації замість геометричної евристики для розміщення точок доступу. Ця реалізація вимагає набору ймовірних сценаріїв розміщення MD, що передбачає певні знання про закономірності використання в зоні покриття. Також виявив, що всі мої цільові функції працюють однаково і дають схожі результати щодо пропускну здатності для всіх досліджених вихідних параметрів. Тому рекомендовано використовувати формулу очікуваного значення, якщо немає інших варіантів, оскільки вона є найпростішою у створенні і в середньому найшвидшою. Однак слід бути обережним у виборі мети, якщо користувач має особливі потреби.

ВИСНОВКИ

У магістерській роботі представлено один з наборів оптимізаційних моделей для різних версій задачі проектування бездротової локальної мережі. Один набір моделей передбачає фіксоване і відоме розташування мобільних пристроїв, а інший допускає, що розташування мобільних пристроїв може змінюватися кожного дня (кочова задача). Ці дві моделі та програмні реалізації, які забезпечують хороші рішення для обох проблем. Також моделі дозволяють визначати місцезнаходження точок доступу в будь-якій точці поверху задачі проектування, і на основі емпіричного аналізу надають високоякісні рішення. Наскільки відомо, не існує інших оптимізаційних процедур для розміщення AP, які не потребують ані набору потенційних місць розташування AP, ані розміщення AP у геометричних фігурах. Через те, що всі моделі мають складні нелінійні цільові функції, лінійні та квадратичні обмеження, а також неперервні змінні, я не можу стверджувати, що це рішення є оптимальними. Також виявив, що розв'язувачі з пакету LINDOAPI надали найкращі рішення для цих моделей порівняно з шістьма іншими комерційно доступними розв'язувачами. Однак я не вичерпно протестували всі доступні нелінійні розв'язувачі. Існує велика кількість інших пакетів, доступних для використання в нелінійній оптимізації, йя розглянули лише невелику кількість доступних програм. OR/MSToday (Fourer, 2011) нещодавно опублікував огляд доступних розв'язувачів LP. У цьому огляді вони зазначають, які розв'язувачі здатні обробляти нелінійні обмеження, хоча нелінійні задачі не розглядаються. Розробники 26 з 45 опитаних розв'язувачів відповіли ствердно на запитання, чи їхні розв'язувачі обробляють загальні нелінійні обмеження, що принаймні деякі з цих розв'язувачів можуть обробляти неопуклі нелінійні задачі.

Хоча протестувавши лише декілька з цих оптимізаційних пакетів,

моделі та процедури попередньої обробки можуть бути використані в будь-якому пакеті розв'язувачів, що підтримує неопуклу оптимізацію. Також я узнав, що пакет Knitro (2011) призначений для розв'язування нелінійних задач, хоча в документації явно зазначено, що негладкі задачі можуть не давати оптимального розв'язку. Визначення того, який розв'язувач може зробити це найкраще, було б корисним напрямком для майбутніх досліджень. Зауважу, що в документації до всіх нелінійних розв'язувачів, які були знайдені зазначено, що оптимальність не може бути гарантована, якщо розв'язувана задача є негладкою. Також зазначу, що мета цього дослідження полягає в тому, щоб взяти складну негладку нелінійну задачу і, використовуючи схему розбиття, перетворити її на невелику кількість гладких нелінійних задач. У розділі 2.1.2 обговорюється, як проблема проектування WLAN стає розривною, коли MD переміщується до іншої точки доступу через переміщення точки доступу, і як наше розбиття MD на фіксовані підмножини, пов'язані з окремими точками доступу, усуває джерело розриву, і, отже, перетворює негладку задачу оптимізації на простішу неопуклу нелінійну оптимізаційну задачу. Хоча це розбиття на розділи створює безліч менших проблем, деякі з них можна відкинути як нездійсненні, а решту легше вирішити, ніж задачу проектування WLAN без фіксованих розділів MD. Хоча моделі на основі Java добре працюють у LINDO для перевірених розмірів задач, багато промислових користувачів можуть віддавати перевагу використанню електронної таблиці Excel для введення даних. OptQuest (2011) підключається до MicrosoftExcel і використовує електронну таблицю для аналізу та розв'язання нелінійних оптимізаційних задач, включаючи цілочисельні задачі. Оптимізація в OptQuest відбувається у два етапи. Спочатку OptQuest аналізує обмеження та мету, щоб визначити найкращі алгоритми для використання. По-друге, OptQuest визначає розв'язок задачі лем. Протестувавши OptQuest з моделлю та екземплярами, у розділі 3. Дозволив три частоти, що не перетинаються, і

використовували моделі та алгоритми попередньої обробки для налаштування задач для OptQuest. Використовувавши Excel з плагіном OptQuest для аналізу та розв'язання цих прикладів. Після завершення аналізу для кожного прикладу OptQuest повернув повідомлення про те, що через неопуклу природу задачі глобальний оптимум не може бути гарантовано. Результати для цих екземплярів наведено в таблиці 1, масштабовані до результатів, визначених «Не перет А» за допомогою LINDOAPI для даного екземпляра, які також можна знайти в таблиці 3.4. Також порівнюємо результати OptQuest з оптимізацією «Не перет Б» з використанням LINDOAPI, який є розв'язувачем LINDO, що розв'язує ту саму задачу, використовуючи методи цілочисельного програмування замість перебору. Не існує статистично значущої різниці між результатами, отриманими за допомогою LINDOAPI з використанням «Не перет Б» та OptQuest з використанням власного цілочисельного розв'язувача. Це ще раз підтверджує, що результати, отримані за допомогою LINDOAPI, є високоякісними, хоча гарантії оптимальності все ще немає. Це видно з продуктивності обох розв'язувачів при повному переборі всіх частотних пар. Оскільки час розв'язку для OptQuest трохи менший, ніж для LINDOAPI з використанням «Не перет Б», подальші дослідження використання розв'язувача OptQuest для моделей можуть виявитися плідними, і це буде гарним напрямком для подальших досліджень.

Слабким місцем моделі є те, що вона дозволяє розміщувати AP та MD лише на одному поверсі. Хоча додавання коефіцієнта ослаблення на поверсі (FAF) дозволило б розміщувати компоненти на різних рівнях, наразі в моделі немає методології, яка б дозволила визначити, скільки AP буде розміщено на кожному поверсі. Модифікація моделі для декількох рівнів також вимагатиме модифікації обмежень (2.9), оскільки відстань більше не буде єдиною змінною, що бере участь у визначенні загасання.

Ще однією сферою для подальших досліджень є масштабування

проблеми. Наразі протестувавши максимум дев'ять точок доступу. Деякі об'єкти мають набагато більшу кількість точок доступу. Наприклад, який небудь стадіон використовує 500 а то і більш точок доступу. В аеропортах можуть бути сотні або навіть тисячі людей, які намагаються отримати віддалений доступ до інтернету одночасно. Цей поточне проєкт потенційно може зайняти значну кількість часу, щоб вирішити будь-який з цих великих випадків без подальшого спрощення.

Таблиця 1 – Дані OptQuest

Приклад			«Не перет Б»		OptQuest		«Не перет А»	
Номер	APs	MDs	Ємність	Час	Ємність	Час	Ємність	Час
1	5	15	92	00:00:21	94	00:00:19	100	00:00:57
2	5	15	91	00:00:12	92	00:00:18	100	00:01:14
3	5	15	90	00:00:13	90	00:00:21	100	00:00:58
4	5	20	93	00:00:29	91	00:00:20	100	00:01:25
5	5	20	88	00:00:28	88	00:00:19	100	00:01:01
6	5	20	94	00:00:18	92	00:00:20	100	00:00:52
7	5	25	86	00:00:16	86	00:00:17	100	00:02:04
8	5	25	94	00:00:11	94	00:00:19	100	00:01:54
9	5	25	92	00:00:30	90	00:00:19	100	00:02:53
10	7	15	92	00:00:54	92	00:00:25	100	00:08:37
11	7	15	91	00:00:27	91	00:00:23	100	00:06:39
12	7	15	91	00:00:42	91	00:00:25	100	00:05:42
13	7	20	85	00:00:39	86	00:00:25	100	00:10:49
14	7	20	92	00:00:45	90	00:00:25	100	00:15:34
15	7	20	92	00:00:31	98	00:00:24	100	00:08:07
16	7	25	86	00:00:41	88	00:00:25	100	01:11:23
17	7	25	88	00:04:04	90	00:00:26	100	00:36:59
18	7	25	79	00:01:16	83	00:00:26	100	00:29:32
19	9	15	85	00:01:03	88	00:00:35	100	02:00:00
20	9	15	84	00:01:21	85	00:00:38	100	02:00:00
21	9	15	92	00:01:03	92	00:00:35	100	02:00:00
22	9	20	93	00:01:18	93	00:00:37	100	02:00:00
23	9	20	86	00:02:09	86	00:00:39	100	02:00:00
24	9	20	87	00:00:40	87	00:00:38	100	02:00:00

ДОДАТОК А

ВИВЕДЕННЯ ФОРМУЛИ ЗАГАСАННЯ

Для визначення загасання між двома точками необхідно розв'язати кілька рівнянь. Втрати на шляху проходження – це ослаблення сигналу, виражене в дБ, і визначається як різниця між переданою потужністю в точці джерела і прийнятою потужністю в точці призначення. Втрати наведено в Rapport (2002) на стор. 108, за формулою:

$$PL = 10 \log(P_s/P_m) \quad (A.1)$$

де P_s – передана потужність;

P_m – прийнята потужність.

Це рівняння можна переписати у вигляді:

$$P_m = P_s 10^{-PL/10} \quad (A.2)$$

Втрати на шляху також можна розрахувати безпосередньо між парою місць, як це показано в Rapport (2002) на стор. 161, за формулою:

$$PL = PL_0 + 10n \log(d_{sm}/d_0) + X_\sigma \quad (A.3)$$

де PL_0 – відомі втрати на шляху між пунктами на відстані d_0 ;

d_{sm} – відстань між джерелом і пунктом призначення;

n – константа, що визначається середовищем передачі;

X_σ – випадкова величина, що представляє шум.

Для цього аналізу ми використовуємо очікуване значення $X\sigma$, яке дорівнює 0. Шум, пов'язаний з MDm, Nm, що використовується в цьому дослідженні, є середньоквадратичним значенням $X\sigma$ і зазвичай приймається рівним 10-10.

Раппапорт (2002) наводить значення n як 3,27 на стор. 165, і ми будемо використовувати це число. Раппапорт також дає значення для PL0 для d_0 в 1 метр, яке ми позначимо як PL1. Це значення можна обчислити на с. 166, що дає значення PL1 31,5 дБ.

$$PL_f = PL_1 + 10 \cdot 3.27 \log(1 \text{ ft}/3.28 \text{ ft}) = 14.6 \text{ dB} \quad (\text{A.4})$$

Таким чином, PL при dsmft від джерела задається формулою:

$$PL = PL_f + 10 \cdot 3.27 \log(d_{sm}/1 \text{ ft}) = 14.6 + 32.7 \log(d_{sm}) \quad (\text{A.5})$$

Об'єднання (A.2) і (A.5) дає:

$$P_m = P_s 10^{-(14.6+32.7 \log(d_{sm}))/10} = P_s 10^{-1.46} d_{sm}^{-3.27} \quad (\text{A.6})$$

Переписування в нотації, наведені в Главі 2, дає згасання потужності (gsm) між AP s і MD m як:

$$g_{sm} = P_m/P_s = 10^{-1.46} d_{sm}^{-3.27} \quad (\text{A.7})$$

ДОДАТОК Б

УВІГНУТІСТЬ ЦІЛЬОВОЇ ФУНКЦІЇ

Модель проектування WLAN, що максимізує пропускну здатність, визначена (2.1)-(2.8), є нелінійною математичною програмою, оскільки цільова функція є нелінійною. Проектування в умовах бюджетного обмеження може бути досягнуто шляхом фіксації $|S|$. Модель розподілу каналів отримують, фіксуючи змінні ws для всіх точок доступу. Коли потужність передачі фіксована, цільова функція складається з суми доданків виду $g(x, y, z) = [\log(1 + x/y)]/z$. На жаль, $g(x, y, z)$ не є ні увігнутою, ні опуклою. Отже, (2.5) не є ні опуклою, ні увігнутою. Графік $g(\cdot)$ вздовж прямої лінії з сайту $(2.0^{-4}, 1.0^{-6}, 9)$ to $(1.0^{-6}, 1.0^{-6}, 24)$ показано на рисунку Б.1. Очевидно, що $g(\cdot)$ не є ні ввігнутою, ні опуклою.

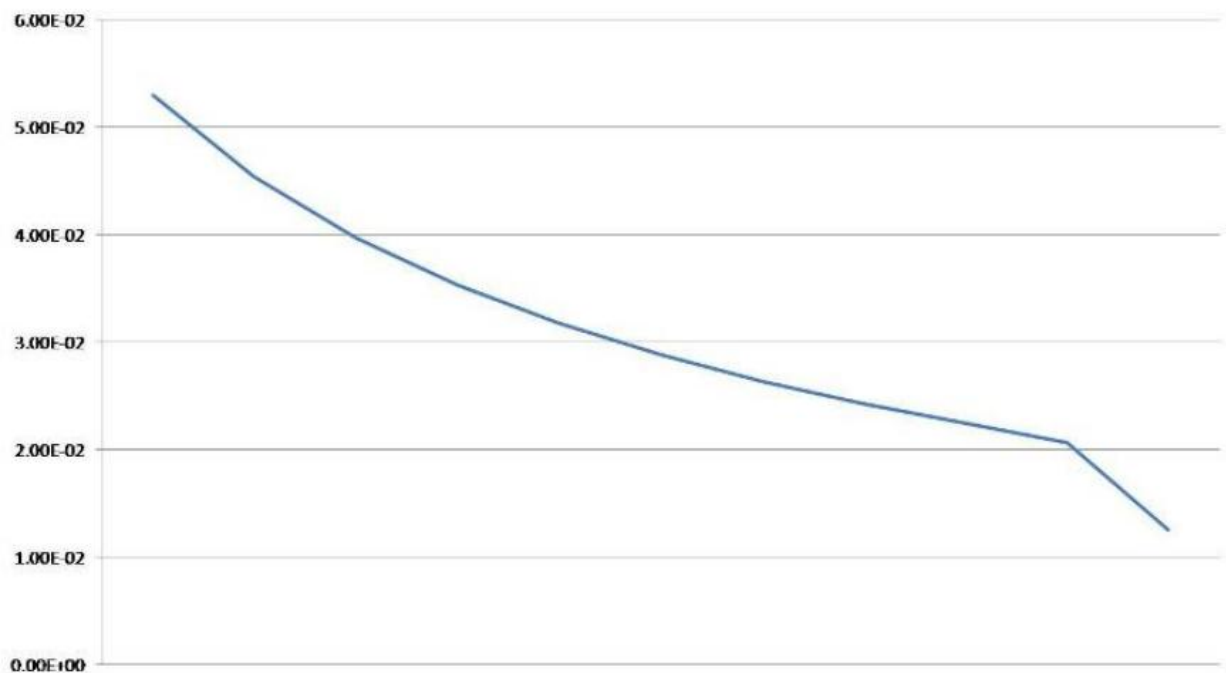


Рисунок Б.1 – Функція $g(x, y, z)$, побудована вздовж прямої лінії

Гессіан функції $g(x, y, z)$ дає конкретний доказ форми функції.

Гессенський – так:

$$H = \begin{bmatrix} -1/[y^2 z \ln(10)q^2] & [x - yq]/[y^3 z \ln(10)q^2] & -1/[yz^2 \ln(10)q] \\ [x - yq]/[y^3 z \ln(10)q^2] & [2xyq - x^2]/[y^4 z \ln(10)q^2] & x/[y^2 z^2 \ln(10)q] \\ -1/[yz^2 \ln(10)q] & x/[y^2 z^2 \ln(10)q] & [2 \ln(q)]/[z^3 \ln(10)] \end{bmatrix},$$

де $q = x/y + 1$.

Множення H з обох боків на вектор $\begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix}$ визначить увігнутість функції, оскільки функція буде увігнутою лише тоді, коли добуток є від'ємним напівскінченним [23]. Величина є від'ємною напівскінченною лише у випадку:

$$\begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix} H \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \leq 0 \text{ for all } \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix}.$$

Перемноживши вектори і спростивши, отримаємо:

$$\begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix} H \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = [-\bar{x}^2 y^2 z^2 + 2\bar{x}\bar{y}y^2 z^2 - 2\bar{x}\bar{z}(x+y)y^2 z + \bar{y}^2(x^2 + 2xy)z^2 + 2\bar{y}\bar{z}xyz(x+y) + \bar{z}^2 y^2(x+y)^2 2\ln(x/y+1)] / [y^2 z^3 \ln(10)(x+y)^2]$$

Оскільки існують значення змінних, для яких функція є як додатною, так і від'ємною, то добуток не є ні додатним, ні від'ємним напіввизначеним, а отже, функція має невизначену увігнутість. Такі значення наведено в таблиці Б.1. Це означає, що не існує простого способу визначити оптимальний

розв'язок.

Таблиця Б.1 – Значення, які дають додатні та від'ємні значення для x'
 Nx для $g(x, y, z)$

	x	y	z	x'	y'	z'
Позитивно	1.00E-06	1.00E-03	3	1.00E-05	1.00E-08	30
Негативно	1.00E-06	1.00E-03	3	1.00E-05	1.00E-09	3

ДОДАТОК В

МЕТАЕВРИСТИЧНІ ПСЕВДОКОДИ

Цей додаток містить всі псевдокоди для метаевристик, використаних для визначення пропускної спроможності в таблиці 2.6. Таблиця В.1 містить перелік усіх використаних процедур, а рисунки В.1-В.7 містять фактично використані псевдокоди.

Таблиця В.1 – Процедури для метаевристики

Neighborhood(n, M, S, n_f, G)	Шукає по сусідству точки доступу, повертає
Рисунок С.1	рішення з найвищою пропускною здатністю для M
Tabu(n, M, S, n_f, G)	Виконує табульований пошук точок доступу, повертаючи рішення
Рисунок С.2	з найбільшою ємністю для M
Genetic(n, M, S, n_f, p, G)	Генетичний алгоритм, який повертає рішення
Рисунок С.3	з найбільшою ємністю для M
DetermineFreqs(n, n_f)	Повертає множину всіх відмінних придатних для використання
Рисунок С.4	частотні n -зв'язки
GetImmigrant(n, Z, M, S, G) Рисунок С.5	Повертає випадковий розв'язок
DetermineChild(q_1, q_2, M, G, n)	Повертає два дочірніх розв'язки, що містять усі гени
Рисунок С.6	від батьків $q_1, q_2 \in Q$
GetCapacity(x^1, x^2, M, G, n)	Визначає ємність для конфігурації точки доступу
Рисунок С.7	заданих через x^1 та x^2 та конфігурацією MD , заданою через m
Rand(P)	Повертає випадковий елемент з довільної множини P
Uniform(0,1)	Повертає псевдовипадкове число, рівномірно розподілене між 0 та 1
Sort(Q)	Сортує множину Q розв'язків за зменшенням ємності

```

Begin Procedure Neighborhood( $n, M, S, n_f, G$ )
   $Z \leftarrow \text{DetermineFreqs}(n, n_f)$ ;  $x \leftarrow \text{GetImmigrant}(n, Z, M, S, G)$ ;  $stop \leftarrow 0$ 
  while  $stop = 0$ 
     $stop \leftarrow 1$ 
    for  $i$  in  $\{1, \dots, n\}$ 
       $temp \leftarrow x_i^1$ 
      for  $j$  in  $S$ 
        if  $j = temp$  then continue
         $x_i^1 \leftarrow j$ 
        for  $k$  in  $Z$ 
           $c \leftarrow \text{GetCapacity}(x^1, k, M, G, n)$ 
          if  $c > x^3$  then
             $x^3 \leftarrow c$ ;  $temp \leftarrow j$ ;  $x^2 \leftarrow k$ ;  $stop \leftarrow 0$ 
          endif
        endfor
      endfor
       $x_i^1 \leftarrow temp$ 
    endfor
  endwhile
  return  $x$ 
End Procedure

```

Рисунок В.1 – Псевдокод алгоритму пошуку сусідів

```

Begin Procedure Tabu( $n, M, S, n_f, G$ )
   $Z \leftarrow \text{DetermineFreqs}(n, n_f); x \leftarrow \text{GetImmigrant}(n, Z, M, S, G)$ 
   $y \leftarrow x$  // A solution containing the best APs thus far
   $tabu \leftarrow \emptyset$  // tabu denotes a list of APs of length  $2n$ 
   $stop \leftarrow 0$ 
  for  $i$  in  $\{1, \dots, n\}$ 
     $tabu \leftarrow tabu \cup \{x_i^1\}$  add entries to end of list
  endfor
  while  $stop < n$ 
     $\omega \leftarrow 0$  // The best capacity value thus far this iteration
     $stop \leftarrow stop + 1$ 
    for  $i$  in  $\{1, \dots, n\}$ 
       $in \leftarrow x_i^1$ 
       $temp \leftarrow x_i^1$ 
      for  $j$  in  $\{1, \dots, |S|\}$ 
        if  $j \in tabu$  then continue
         $x_i^1 \leftarrow j$ 
        for  $k$  in  $Z$ ,  $c \leftarrow \text{GetCapacity}(x^1, k, M, G, n)$ 
        if  $c > \omega$  then
           $\omega \leftarrow c; out \leftarrow i; in \leftarrow j; infreq \leftarrow k$ 
        endif
      endfor
       $x_i^1 \leftarrow temp$ 
    endfor
     $x_{out}^1 \leftarrow in; x^2 \leftarrow infreq; x^3 \leftarrow \omega; tabu \leftarrow tabu \cup \{in\}$ 
    if  $|tabu| \geq 2n$  then  $tabu \leftarrow tabu \setminus \{tabu_1\}$  remove entry from start of list
    if  $x^3 > y^3$  then  $y \leftarrow x; stop \leftarrow 0$ 
  endwhile
  return  $y$ 
End Procedure

```

Рисунок В.2 – Псевдокод алгоритму пошуку за табуляцією.

```

Begin Procedure Genetic( $n, M, S, n_f, p, G$ )
   $Q \leftarrow \emptyset$ ;  $Z \leftarrow \text{DetermineFreqs}(n, n_f)$ 
  for  $i$  in  $\{1, \dots, p\}$ 
     $Q \leftarrow Q \cup \{\text{GetImmigrant}(n, Z, M, S, G)\}$ 
  endfor
   $\text{Sort}(Q)$ ;  $j \leftarrow 0$ 
  while  $j < p$  do
     $P \leftarrow \emptyset$  // An ordered set of parent solutions;  $P \subset Q$ 
    for  $i$  in  $\{1, \dots, \lceil 2p/3 \rceil\}$ 
       $P \leftarrow P \cup Q^i$ 
    endfor
     $Q \leftarrow \emptyset$ 
    for  $i$  in  $\{1, \dots, \lfloor |P|/2 \rfloor\}$ 
       $x_1 \leftarrow \text{Rand}(P)$ ;  $P \leftarrow P \setminus \{x_1\}$ ;  $x_2 \leftarrow \text{Rand}(P)$ ;  $P \leftarrow P \setminus \{x_2\}$ 
       $[y_1, y_2] \leftarrow \text{DetermineChild}(x_1, x_2, M, G, n)$ 
       $N \leftarrow \{y_1, y_2, x_1, x_2\}$  // An ordered set of children and parent solutions
       $\text{Sort}(N)$ ;  $Q \leftarrow Q \cup \{N^1, N^2\}$ 
    endfor
     $Q \leftarrow Q \cup P$ 
    for  $i$  in  $\{|Q| + 1, \dots, p\}$ 
       $Q \leftarrow Q \cup \{\text{GetImmigrant}(n, Z, M, S, G)\}$ 
    endfor
     $\text{Sort}(Q)$ ;  $j \leftarrow j + 1$ 
  endwhile
  return  $Q^1$ 
End Procedure

```

Рисунок В.3 – Псевдокод генетического алгоритму

```

Begin Procedure DetermineFreqs( $n, n_f$ )
   $out \leftarrow \emptyset$ 
  for  $i$  in  $\{1, \dots, n\}$ 
     $w_i \leftarrow 1$ 
  endfor
   $k \leftarrow n$ 
  while  $k > 1$  do
     $w_k \leftarrow w_k + 1$ 
    if  $w_k > f$  then
       $w_k \leftarrow 1; k \leftarrow k - 1$ 
    else
       $u_k \leftarrow w_k; use \leftarrow \text{true}$ 
      for  $i$  in  $\{1, \dots, n_f\}$ 
         $temp \leftarrow 0$ 
        for  $j$  in  $\{1, \dots, n\}$ 
          if  $w_j = i$  then  $temp \leftarrow temp + 1$ 
        endfor
        if  $temp = 0$  then  $use \leftarrow \text{false}$ 
      endfor
      if  $use$  then  $out \leftarrow out \cup \{w\}$ 
       $count \leftarrow n$ 
    endif
  endwhile
  return  $out$ 
End Procedure

```

Рисунок В.4 – Псевдокод для визначення корисної частоти n-кортежів

```

Begin Procedure GetImmigrant( $n, Z, M, S, G$ )
   $i \leftarrow 1$ 
  while  $i \leq n$ 
     $k \leftarrow \text{Rand}(\{1, \dots, |S|\})$ 
     $use \leftarrow \text{true}$ 
    for  $j$  in  $\{1, \dots, i - 1\}$ 
      if  $k = temp_j$  then  $use \leftarrow \text{false}$ 
    endfor
    if  $use$  then  $temp_i \leftarrow k; i \leftarrow i + 1$ 
  endwhile
   $x^1 \leftarrow temp; x^2 \leftarrow \text{Rand}(Z); x^3 \leftarrow \text{GetCapacity}(x^1, x^2, M, G, n)$ 
  return  $x$ 
End Procedure

```

Рисунок В.5 – Псевдокод для визначення рішення щодо імміграції

```

Begin Procedure DetermineChild( $j, k, M, G, n$ )
   $\delta_1 \leftarrow 0; \delta_2 \leftarrow 0; [(\bar{a}_1^1, \dots, \bar{a}_n^1), (\bar{a}_1^2, \dots, \bar{a}_n^2), \bar{a}^3] \leftarrow j; [(\bar{v}_1^1, \dots, \bar{v}_n^1), (\bar{v}_1^2, \dots, \bar{v}_n^2), \bar{v}^3] \leftarrow k$ 
  for  $i$  in  $\{1, \dots, n\}$ 
    if ( $\text{Uniform}(0, 1) < 0.5$  or  $\delta_2 \geq n/2$ ) and ( $\delta_1 < n/2$ ) then
       $\delta_1 \leftarrow \delta_1 + 1; a_{\delta_1}^1 \leftarrow \bar{a}_i^1; a_{\delta_1}^2 \leftarrow \bar{a}_i^2$ 
    else
       $\delta_2 \leftarrow \delta_2 + 1; b_{\delta_2}^1 \leftarrow \bar{a}_i^1; b_{\delta_2}^2 \leftarrow \bar{a}_i^2$ 
    endif
  endfor
  for  $i$  in  $\{1, \dots, n\}$ 
    if ( $\text{Uniform}(0, 1) < 0.5$  or  $\delta_2 \geq n$ ) and ( $\delta_1 < n$ ) then
       $\delta_1 \leftarrow \delta_1 + 1; a_{\delta_1}^1 \leftarrow \bar{b}_i^1; a_{\delta_1}^2 \leftarrow \bar{b}_i^2$ 
    else
       $\delta_2 \leftarrow \delta_2 + 1; b_{\delta_2}^1 \leftarrow \bar{b}_i^1; b_{\delta_2}^2 \leftarrow \bar{b}_i^2$ 
    endif
  endfor
   $a^3 \leftarrow \text{GetCapacity}(a^1, a^2, M, G, n); b^3 \leftarrow \text{GetCapacity}(b^1, b^2, M, G, n)$ 
  return  $[a^1, a^2, a^3], [b^1, b^2, b^3]$ 
End Procedure

```

Рисунок В.6 – Псевдокод для кросовера генетичних алгоритмів

```

Begin Procedure GetCapacity( $x^1, x^2, M, G, n$ )
  for  $i$  in  $M$ 
     $temp \leftarrow \infty$ 
    for  $j$  in  $\{1, \dots, n\}$ 
       $k \leftarrow x_j^1$ 
      if  $g_{ki} < temp$  then
         $temp \leftarrow g_{ki}; \rho \leftarrow i; \sigma \leftarrow k$ 
      endif
    endfor
     $z_{\rho\sigma} \leftarrow 1$ 
  endfor
  for  $i$  in  $\{1, \dots, n\}$ 
     $\beta_i \leftarrow 10^{-10}; l \leftarrow x_i^1$ 
    for  $j$  in  $\{1, \dots, n\}$ 
      if  $i \neq j$  and  $x_i^2 = x_j^2$  then
         $\beta_i \leftarrow \beta_i + \sum_{k \in M | z_{jk}=1} g_{lk}$ 
      endif
    endfor
  endfor
   $\zeta \leftarrow \sum_{i \in M} [\log(1 + \sum_{j=1}^n g_{ji} / \beta_j) / \sum_{j \in M, k \in S | z_{ji}=1} z_{jk}]$ 
  return  $\zeta$ 
End Procedure

```

Рисунок В.7 – Псевдокод для визначення потужності

Цей розділ містить псевдокод для «Алгоритм А». Список усіх алгоритмів наведено у таблиці В.2. Весь псевдокод можна знайти на рисунках В.8-В.15.

Таблиця В.2 – Процедури для алгоритму глобальної оптимізації

$\text{Global}(R, S, M, P, n_n, \Lambda, \Gamma, J, K)^0$	Алгоритм глобальної оптимізації
Рисунок В.8	
$\text{GetUpper}(R, S, M, P, A, n_n, \Lambda, \Gamma, J, K^r, \xi)$	Повертає верхню межу місткості для регіону r
Рисунок В.9	
$\text{GetPartitions}(S, M, \Gamma)$	Повертає набір розділів MD-дисків
Рисунок В.10	Для використання в глобальній оптимізації
$\text{Init}(S, M, \Gamma)$	Повертає початковий розділ, який інші
Рисунок В.11	Базуються на
$\text{GetMax}(M, \Gamma)$	Повертає MD, які є максимальними
Рисунок В.12	Відстань між ними
$\text{Score}(A)$	Забиває розділ, щоб забезпечити
Рисунок В.13	Без дублювання
$\text{CheckA}(A, B, v)$	Перевіряє наявність дублікатів розділів
Рисунок В.14	
$\text{Compare}(A1, A2)$	Порівнює два розділи, щоб переконатися, що
Рисунок В.15	Вони не є дублікатами
$\text{Uniform}(0, 1)$	Повертає псевдовипадкове число, рівномірно розподілене між 0 та 1

```

Begin Procedure Global( $R, S, M, P, n_n, \Lambda, \Gamma, J, K^0$ )
   $A \leftarrow \text{GetPartitions}(S, M, \Gamma)$ 
  for  $i$  in  $\{1, \dots, n\}$ 
     $\xi \leftarrow 0$ 
    for  $r$  in  $R$ 
       $[p_r, \xi] \leftarrow \text{GetUpper}(R, S, M, P, A, n_n, \Lambda, \Gamma, J, K^r, \xi)$ 
    endfor
     $j \leftarrow \max_{r \in R} (p_r)$ 
    while  $j > \xi$ 
      for  $r$  in  $R | p_r = j$ 
         $a \leftarrow 0; b \leftarrow 0; c \leftarrow 0$ 
        for  $s$  in  $S$ 
          if  $xx_s^r - xn_s^r > a$ 
             $a \leftarrow xx_s^r - xn_s^r; b \leftarrow s; c \leftarrow 1$ 
          endif
          if  $yx_s^r - yn_s^r > a$ 
             $a \leftarrow yx_s^r - yn_s^r; b \leftarrow s; c \leftarrow 2$ 
          endif
        endfor
         $R \leftarrow R \cup \{|R| + 1, |R| + 2\}; p_r \leftarrow 0$ 
         $xx^{|R|-1} \leftarrow xx^r; xx^{|R|} \leftarrow xx^r; xn^{|R|-1} \leftarrow xn^r; xn^{|R|} \leftarrow xn^r$ 
         $yx^{|R|-1} \leftarrow yx^r; yx^{|R|} \leftarrow yx^r; yn^{|R|-1} \leftarrow yn^r; yn^{|R|} \leftarrow yn^r$ 
        if  $c = 1$ 
           $xx_b^{|R|-1} \leftarrow (xx_b^r + xn_b^r)/2; xn_b^{|R|} \leftarrow (xx_b^r + xn_b^r)/2$ 
        else
           $yx_b^{|R|-1} \leftarrow (yx_b^r + yn_b^r)/2; yn_b^{|R|} \leftarrow (yx_b^r + yn_b^r)/2$ 
        endif
         $[p_{|R|}, \xi] \leftarrow \text{GetUpper}(R, S, M, P, A, n_n, \Lambda, \Gamma, J, K^{|R|}, \xi)$ 
         $[p_{|R|-1}, \xi] \leftarrow \text{GetUpper}(R, S, M, P, A, n_n, \Lambda, \Gamma, J, K^{|R|-1}, \xi)$ 
      endfor
       $j \leftarrow \max_{r \in R} (p_r); d \leftarrow r | p_r = j$ 
      if  $\sqrt{\|xx^d - xn^d\|^2 + \|yx^d - yn^d\|^2} < \sqrt{N}$  then continue
    endwhile
  endfor
  return  $[xx^d, xn^d, yx^d, yn^d]$ 
End Procedure

```

Рисунок В.8 – Псевдокод алгоритму глобального пошуку

```

Begin Procedure GetUpper( $R, S, M, P, A, n_n, \Lambda, \Gamma, J, K^{|R-1|}, \xi$ )
  for  $i$  in  $\{1, \dots, n_n\}$ 
    for  $s$  in  $S$ 
       $x_s \leftarrow \text{Uniform}(0, 1)(xx_s^r - xn_s^r) + xn_s^r; y_s \leftarrow \text{Uniform}(0, 1)(yx_s^r - yn_s^r) + yn_s^r$ 
      for  $m$  in  $M$ 
         $g_{sm} \leftarrow 10^{-1.46}[(x_s - \theta_m)^2 + (y_s - \phi_m)^2 + 14^2]^{-3.27/2}$ 
      endfor
    endfor
     $\zeta_i \leftarrow \sum_{m \in M} [\log(1 + \sum_{s \in S | m \in A_s} [P_s g_{sm} / (\sum_{s1 \in S \setminus \{s\}} P_{s1} g_{s1m} + N_m)])] / |A_s|$ 
    if  $\zeta_i > \xi$ 
       $\xi \leftarrow \zeta_i$ 
    endif
  endfor
   $p_r \leftarrow \Lambda \sqrt{\|xx^d - xn^d\|^2 + \|yx^d - yn^d\|^2} + \sum_{i \in \{1, \dots, n\}} \zeta_i / n$ 
  return  $[p_r, \xi]$ 
End Procedure

```

Рисунок В.9 – Псевдокод алгоритму верхньої межі

```

Begin Procedure GetPartitions( $S, M, \Gamma$ )
   $A \leftarrow \text{Init}(S, M, \Gamma)$ ;  $B \leftarrow \{A\}$ ;  $v_1 \leftarrow \text{Score}(A)$ ;  $k \leftarrow 2$ 
  While  $k < 10$ 
     $[m1, m2] \leftarrow \text{GetMax}(M, \Gamma)$ ;  $\gamma_{m1, m2} \leftarrow 0$ 
    for  $c$  in  $\{1, \dots, k - 1\}$ 
       $A \leftarrow B_c$ ;  $i \leftarrow l|m1 \in A_l$ ;  $j \leftarrow l|m2 \in A_l$ 
      if  $i \neq j$ 
         $A_i \leftarrow A_i \setminus \{m1\}$ ;  $A_j \leftarrow A_j \cup \{m1\}$ 
        if  $\text{CheckA}(A, B, v) = 0$ 
           $\Lambda \leftarrow B \cup \{A\}$ ;  $v_k \leftarrow \text{Score}(A)$ ;  $k \leftarrow k + 1$ 
        endif
         $A_i \leftarrow A_i \cup \{m1, m2\}$ ;  $A_j \leftarrow A_j \setminus \{m1, m2\}$ 
        if  $\text{CheckA}(A, B, v) = 0$ 
           $B \leftarrow B \cup \{A\}$ ;  $v_k \leftarrow \text{Score}(A)$ ;  $k \leftarrow k + 1$ 
        endif
      endif
    endfor
  endwhile
  return  $B$ 
End Procedure

```

Рисунок В.10 – Псевдокод алгоритму отримання розділів

```

Begin Procedure Init( $S, M, \Gamma$ )
     $[m1, m2] \leftarrow GetMax(M, \Gamma)$ 
     $A_1 \leftarrow \{m1\}; A_2 \leftarrow \{m2\}; C_1 \leftarrow m1; C_2 \leftarrow m2; D \leftarrow \{m1, m2\}$ 
    for  $i$  in  $\{3 \dots |S|\}$ 
         $temp \leftarrow \infty; best \leftarrow 0$ 
        for  $j$  in  $\{1 \dots |M|\}$ 
             $temp2 \leftarrow 0$ 
            for  $k$  in  $\{1 \dots i - 1\}$ 
                if  $temp2 < \gamma_{j, C_k}$  then  $temp2 \leftarrow \gamma_{j, C_k}$ 
            endfor
            if  $temp1 > temp2$  then  $temp1 \leftarrow temp2; best = j;$ 
        endfor
         $A_i \leftarrow \{M_j\}; C_i \leftarrow M_j; D \leftarrow D \cup \{M_j\}$ 
    endfor
    for  $m$  in  $M \setminus D$ 
         $temp \leftarrow 0; best \leftarrow 0$ 
        for  $i$  in  $\{1 \dots |S|\}$ 
            if  $\gamma_{m, C_i} > temp$  then  $temp \leftarrow \gamma_{m, C_i}; best \leftarrow i$ 
        endfor
         $A_{best} \leftarrow A_{best} \cup \{m\}$ 
    endfor
    return  $A$ 
End Procedure

```

Рисунок В.11 – Отримання псевдокоду алгоритму початкового розділу

```

Begin Procedure GetMax( $M, \Gamma$ )
   $i \leftarrow 0; j \leftarrow 0; k \leftarrow 0$ 
  for  $m1$  in  $M$ 
    for  $m2$  in  $M \setminus \{m1\}$ 
      if  $\gamma_{m1,m2} < k$ 
         $i \leftarrow m1; j \leftarrow m2; k \leftarrow \gamma_{m1,m2}$ 
      endif
    endfor
  endfor
  return  $[i, j]$ 
End Procedure

```

Рисунок В.12 – Отримання максимального псевдокоду алгоритму

```

Begin Procedure Score( $A$ )
   $k \leftarrow 0$ 
  for  $i$  in  $\{1, \dots, |A|\}$ 
     $t \leftarrow 1$ 
    for  $j$  in  $\{1, \dots, |A_i|\}$ 
       $t \leftarrow t * A_{ij}$ 
    endfor
     $t \leftarrow t^{|A_i|}; k \leftarrow k + t$ 
  endfor
  return  $k$ 
End Procedure

```

Рисунок В.13 – Псевдокод алгоритму отримання балів

```

Begin Procedure CheckA( $A, B, v$ )
   $j \leftarrow \text{Score}(A); k \leftarrow 0$ 
  for  $i$  in  $\{1, \dots, |B|\}$ 
    if  $j = v_i$ 
       $k \leftarrow \text{Compare}(A, B_i)$ 
      if  $k = 1$  break
    endif
  endfor
  return  $k$ 
End Procedure

```

Рисунок В.14 – Псевдокод алгоритму повторення перевірки

```

Begin Procedure Compare( $A1, A2$ )
   $k \leftarrow 1$ 
  for  $i$  in  $\{1, \dots, |A1|\}$ 
    if  $\nexists j | A1_i = A2_j$ 
       $k \leftarrow 0$ 
    endif
  endfor
  return  $k$ 
End Procedure

```

Рисунок В.15 – Порівняння псевдокоду алгоритму

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Даллас Ковбойс. Оцінка розподілених антенних систем ТРІ зі спортивними майданчиками. [Електронний ресурс] – Режим доступу: <http://www.dallascowboys.com/cdw/> (дата звернення 11.09.2023). – Назва з екрану.
2. Shannon, 1949
3. Geier, J. 2009. Огляд сайту 802.11a/b. [Електронний ресурс] – Режим доступу: <http://www.wi-fiplanet.com/columns/article.php/1479831> (дата звернення 15.09.2023). – Назва з екрану.
4. Rodrigues, R., G. Mateus, A. Loureiro. 2000. Про проектування та план пропускної здатності ності бездротової локальної мережі. Симпозіум з мережевих операцій та управління, Гонолулу, Гаваї. – С. 335-348.
5. Бахрі А., С. Чемберленд. 2005. Про проблему проектування бездротової локальної мережі з гарантією продуктивності. // Комп'ютерні мережі 48 – С. 856-866.
6. Адікес М., Р. Білло, Б. Норман, Б. Ннаджі, Я. Раджгопал. 2002. Оптимізація макетів бездротових мереж зв'язку в приміщенні. ІЕ Transactions 34. – С. 823-826.
7. Амальді, Е., С. Бозіо, Ф. Малучеллі, Д. Юань. 2011. Вирішення нелінійних проблем покриття, що виникають при проектуванні WLAN. Дослідження операцій 59. – С. 173-187.
8. Jaffrès-Runser, K., J. Gorce, S. Ubéda. 2006. Багатоцільове QoS-орієнтоване планування для бездротових локальних мереж всередині приміщень. Конференція IEEE з транспортних технологій, 2006, VTC2006 Fall 1-5.
9. Mladenović, N., J. Brimberg, P. Hansen, J. Morena-Pérez. 2005.

Проблема р-медіани: огляд метаевристичних підходів. Європейський журнал оперних досліджень *tional Research* 179, 927-939. NEOS. 2010. Отримано 10 серпня 2010 року на OptQuest. 2011 [Електронний ресурс] – Режим доступу: <http://www-neos.mcs.anl.gov/> (дата звернення 15.09.2023). – Назва з екрану.

10. Мішра А., В. Шрівастава, С. Банерджі, В. Арбо. 2006. Частково перекриті канали не вважаються шкідливими. *SIG Metrics/Performance Conference*, червень 2006. – С. 63-74.

11. Fortune S., D. Gay, B. Kerningham, O. Landron, R. Valenzuela, M. Wright. 1995. WISE Design of Indoor Wireless Systems: Практичні обчислення та оптимізація *tion. IEEE Computational Science and Engineering* 2 58-68.

12. Panjawni, M., A. Abbott, T. Rappaport. 1996, Інтерактивне обчислення областей покриття для бездротових приміщень. *IEEE Journal on Selected Areas in Communications* 14 – с. 420-430.

13. Хіллз, А. 2001. Великомасштабний дизайн бездротової локальної мережі. *IEEE Communications Magazine* 39. – С. 98-104.

14. Свідмор, Р., А. Верстак, Н. Рамакрішнан, Т. Раппапорт, Л. Уотсон, Ж. Хе, С. Варадараджан, К. Шаффер, Ж. Чен, К. Бэ, Ж. Цзян, В. Трантер. 2004. На шляху до інтегрованих PSE для бездротового зв'язку: Досвід проектів S4W та SitePlanner. Огляд мобільних обчислень та комунікацій 8. – С. 20-34.

15. Proxim Wireless. 2009. [Електронний ресурс] – Режим доступу: http://www.proxim.com/products/ap_700/. (дата звернення 15.09.2023). – Назва з екрану.

16. Leung, K., B. Kim. 2003. Призначення частоти для бездротових мереж *IEEE 802.11. Конференція IEEE з транспортних технологій, 2003, VTC 2003-fall* 3. – С. 1422-1426.

17. Vanhatupa, T., M. Hännikäinen, T. Hamäläinen. 2007. Оцінка моделей та алгоритмів оцінки пропускну здатності для частотного

планування WLAN. Комп'ютерні мережі 51. – С. 3110-3124.

18. Мішра, А., Е. Рознер, С. Банерджі, В. Арбо. 2005b. Використання каналів, що частково перекриваються, в бездротових мережах: Перетворення небезпеки на перевагу. В *ternet Measurement Conference*, 2005. – С. 311-316.

19. Mishra, A., S. Banerjee, W. Arbaugh. 2005a. Призначення каналів для бездротових локальних мереж на основі зваженого розфарбовування. Огляди мобільних обчислень та зв'язку 9. – С. 19-31.

20. Кеннінгтон, Д., Д. Крац, Г. Спіріде. 2010a. Розділ 6: Моделювання та проектування WLAN на основі оптимізації. Проектування бездротових мереж: Оптимізаційні моделі та процедури вирішення. Ред. J. Kennington, E. Olinick, D. Rajan, Springer, New York.

21. Pinter, J. 1996. Глобальна оптимізація в дії: безперервна оптимізація та оптимізація Ліпшиця – алгоритми, реалізації та застосування. Kluwer Academic Publishers, Бостон.

22. Kennington, J., J. Kratz, D. Rajan. 2010b. Проблема проектування статичної бездротової локальної мережі. Технічний звіт 10-EMIS-05, кафедра інженерних акцій, інформації та систем, Південний методистський університет, Даллас, штат Техас, 75205. [Електронний ресурс] – Режим доступу: <http://lyle.smu.edu/~jlk/10-EMIS-05/static.pdf> (дата звернення 15.09.2023). – Назва з екрану.

23. Базараа, М., Х. Шералі, К. Шетті. 2006. Нелінійне програмування: Теорія та застосування. Wiley Interscience, Хобокен, Нью-Джерсі.