

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження методів оптимізації маршруту мобільного стоку сенсорної інфокомунікаційної мережі

Виконав:
студент 2 курсу, групи ТСМм-19-1.
Кривошопка М.В.
(прізвище, ініціали)

Спеціальність: 172 Телекомунікації та радіотехніка
(код і повна назва спеціальності)

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

Освітня програма: Телекомунікаційні системи та мережі
(повна назва освітньої програми)

Керівник: доцент кафедри ІКІ ім. В.В. Поповського.
Мельнікова Л.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Лемешко О.В.
(прізвище, ініціали)

2021р.

Харківський національний університет радіоелектроніки

Факультет _____ Інфокомунікацій _____
 Кафедра _____ Інфокомунікаційної інженерії імені В.В. Поповського _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 172 Телекомунікації та радіотехніка _____
 (код і повна назва)
 Тип програми _____ освітньо-наукова _____
 (освітньо-професійна або освітньо-наукова)
 Освітня програма _____ Телекомунікаційні системи та мережі _____
 (повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри _____
(підпис)

« _____ » _____ 2021р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Кривошопкі Миколі Володимировичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів оптимізації маршруту мобільного стоку сенсорної інфокомунікаційної мережі
 затверджена наказом по університету від «10» березня 2021р. №343 Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 24.05.2021р. _
3. Вихідні дані до роботи: сенсорна безпроводова мережа з кількістю вузлів мережі 10, 20, 50; відстань між вузлами мережі 1-10 км; розподілення відстанів між вузлами нормальне з параметрами м.о. = 5 км, с.к.о. = 1 км; рівномірне розподілення в діапазоні 1-10 км; метод гілок і меж ; мурашиний алгоритм; алгоритм імітації випалювання
4. Перелік питань, що потрібно опрацювати в роботі:
 - 1) Загальні відомості про безпроводові сенсорні мережі
 - 2) Математична постановка задачі оптимізації маршруту мобільного стоку і методи її вирішення .
 - 3) Дослідження методів оптимізації маршруту мобільного стоку сенсорної інфокомунікаційної мережі

5. Перелік графічного матеріалу із зазначенням креслень, плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал у вигляді ppt-презентації. ..

1. Вступ. Загальні відомості про БСМ. 2. Стандарти БСМ. 3. Математична постановка задачі комівояжера. 4. Методи вирішення задачі комівояжера. 5. Мурашиний алгоритм. 6. Алгоритм імітації випалювання. 7. Результати вирішення задачі комівояжера методом гілок і меж. 8. Початкові данні і результат вирішення задачі комівояжера для $n=50$. 9. Рішення мурашиного алгоритму. 11. Рішення алгоритму імітації випалювання. 12. Висновки. 13. Список публікацій.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Основна частина	доцент кафедри ІКІ ім. В.В. Поповського Мельнікова Любов Іванівна		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	15.02.2021	Виконано
2	Збір матеріалів для дослідження	28.02.2021	Виконано
3	Розробка 1 розділу	19.03.2021	Виконано
4	Розробка 2 розділу	02.04.2021	Виконано
5	Розробка 3 розділу	12.04.2021	Виконано
7	Оформлення кваліфікаційної роботи	18.05.2021	Виконано

Дата видачі завдання 15 лютого 2021 року _____

Студент _____ Кривошопка М.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ доцент Мельнікова Л.І.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 82 с., 24 рис., 14 табл., 2 додатки, 29 джерел.

АЛГОРИТМ ІМІТАЦІЇ ВИПАЛЮВАННЯ, БЕЗПРОВІДНА СЕНСОРНА МЕРЕЖА, ЗАДАЧА КОМІВОЯЖЕРА, МОБІЛЬНИЙ СТОК, МУРАШИНИЙ АЛГОРИТМ, ОПТИМІЗАЦІЯ.

Об'єкт дослідження: дослідження методів вирішення задачі оптимізації маршруту мобільного стоку в безпроводовій сенсорній мережі.

Предмет дослідження: безпроводова сенсорна мережа з мобільним стоком.

Методи дослідження: аналіз, збір, узагальнення результатів і формування висновків, методи оптимізації, методи імітаційного моделювання.

У кваліфікаційній роботі вирішена задача дослідження методів вирішення задачі оптимізації маршруту мобільного стоку в безпроводовій сенсорній мережі.

Проведено порівняння евристичних алгоритмів, мурашиного алгоритму та алгоритму імітації випалювання, та точного методу гілок і меж для задачі визначення оптимального маршруту мобільного стоку в сенсорній безпроводній мережі на основі рішення задачі комівояжера.

ABSTRACT

Explanatory note: 82 pages, 24 figures, 14 tables, 2 appendices, 29 sources.

ANT ALGORITHM, BURNING SIMULATION ALGORITHM, COMMERCIAL FRIEND, MOBILE STOCK, OPTIMIZATION, WIRELESS SENSOR NETWORK.

Object of study: research of methods for solving the problem of optimizing the route of mobile runoff in a wireless sensor network.

Subject of study: wireless sensor network with mobile drain.

Research methods: analysis, collection, generalization of results and formation of conclusions, optimization methods, methods of simulation modeling.

In the qualification work the problem of research of methods of the decision of a problem of optimization of a route of a mobile drain in a wireless sensor network is solved.

A comparison of heuristic algorithms, ant algorithm and firing simulation algorithm, and the exact method of branches and boundaries for the problem of determining the optimal route of mobile runoff in a sensor wireless network based on the solution of the salesman problem.

ЗМІСТ

Перелік умовних позначень символів, одиниць, скорочень і термінів...	7
Вступ.....	8
1 Загальні відомості про безпроводові сенсорні мережі.....	10
1.1 Поняття безпроводової сенсорної мережі.....	10
1.2 Структура безпроводних сенсорних мереж.....	11
1.3 Практичне застосування безпроводових сенсорних мереж.....	13
1.4 Перелік стандартів.....	14
1.5 Моделі передачі даних в безпроводових сенсорних мережах.....	19
1.6 Поняття часу життя мережі і методи збільшення часу життя безпроводової сенсорної мережі.....	23
2 Математична постановка задачі оптимізації маршруту мобільного стоку і методи її вирішення.....	28
2.1 Математична постановка задачі комівояжера.....	28
2.2 Методи вирішення задачі комівояжера.....	30
2.3 Метод гілок і меж.....	31
2.4 Мурашиний алгоритм.....	34
2.5 Алгоритм імітації випалювання.....	44
3 Дослідження методів оптимізації маршруту мобільного стоку сенсорної інфокомунікаційної мережі.....	50
3.1 Постановка задачі алгоритмів оптимізації маршруту.....	50
3.2 Визначення оптимального значення кількості ітерацій для розв'язання задачі.....	58
3.3 Порівняльний аналіз алгоритмів для задачі оптимізації шляху...	62
Висновки.....	78
Перелік джерел посилань.....	80
Додаток А Код програми мурашиного алгоритму.....	83
Додаток Б Код програми алгоритма імітації випалювання.....	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І
ТЕРМІНІВ

БМЗ - безпроводна мережа зв'язку
БСМ - безпроводна сенсорна мережа
ЗК - задача комівояжер
ЗЛП - задача лінійного програмування
К - комутатор
М - маршрутизатор
МК - мережійний координатор
МС - мобільний сток
СМ - сенсорна мережа
У.О.- условні одиниці
АСО - ant colony optimization
FFD - Full Function Device
IEEE - institute of electrical and electronics engineers
МЕН - Micro-Energy Harvesters
NL - Network Lifetime (час життя)
QoS - Quality of Service
RFD - Reduced Function Device
SD - Smart Dust

ВСТУП

Безпроводові сенсорні мережі являють собою активно розвинуті системи автоматизації та управління, моніторингу та контролю. Взаємодіючи з управляючими пристроями, датчики створюють розподілену, самоорганізовану систему збору, обробки і передачі інформації. Поняття «самоорганізована мережа» визначається як система, в якій пристрої «вміють» самі знаходити один одного і формувати мережу, а разі виходу з ладу будь-якого з вузлів можуть встановлювати нові маршрути для передачі повідомлень.

Спочатку виникнувши в якості узагальненої концепції розумного пилу (Smart Dust), безпроводові сенсорні мережі (БСМ) поступово набували конкретних обрисів і на сьогоднішній день є досить добре стандартизованими. Аналіз стандартів показує, що сенсорні мережі з базовим алгоритмом роботи ідентичні іншим цифровим безпроводових мереж з розділяються доступом до середовища передачі. У той же час БСМ розраховані на передачу невеликих обсягів даних з малою частотою - це дозволяє переводити пристрій в режим низького споживання енергії для збільшення часу їх автономної роботи. Тому актуальним завданням є дослідження і розробка моделей, що описують мережу з точки зору потужності, споживаної пристроями [1].

Технологія сенсорних мереж не вимагає для побудови мережі дорогих кабелів разом з допоміжним обладнанням (кабельними каналами, шафами і т.д.), так як сенсорна мережа підтримує основні інтерфейси і протоколи, які застосовуються в даний час, є можливість інтегрувати її в існуючу мережу без проведення масштабної реконструкції.

Мініатюрні і тому які не потребують енерговитрат датчики (термін експлуатації може досягати кілька років) забезпечують можливість їх розміщення в важкодоступних місцях і на великих територіях. Безпроводові рішення незамінні, коли необхідно пов'язати в мережу постійно рухомі або часто переміщувані вузли. Однак же недоліком безпроводових рішень виявляється їх менша надійність, як в сенсі гарантованої доставки даних за обмежений час, так і в сенсі захисту переданої інформації від несанкціонованого доступу.

До найбільш простим методам збільшення часу життя БСМ відносяться поліпшення апаратних характеристик пристроїв: зменшення енергоспоживання

окремих компонентів, оптимізація їх розміщення на кристалі або друкованої плати або збільшення ємності батарей. Останні дослідження в області мініатюрних перетворювачів альтернативної енергії (МЕН, Micro-Energy Harvesters) відкрили ряд можливостей для створення повністю автономних вузлів сенсорної мережі при збереженні їх невеликих розмірів. Відомий ряд готових рішень для підключення сенсорних вузлів до мініатюрних сонячних батарей, перетворювачів вібраційної енергії і термогенератор на основі елемента Пелетьє. [2]. Однак на сьогоднішній день жодне з рішень по збору і перетворення альтернативної енергії ще не знайшло масового застосування в реальних мережах збору даних, що складаються з сотень вузлів, Перш за все, через високу вартість, що включає в себе витрати на регулярне обслуговування [3].

Розробка і введення сенсорних мереж в усі сфери життя надасть величезну кількість переваг людству. Тематика сенсорних безпроводових мереж ще не достатньо вивчена. На даний момент існує ряд невирішених проблем наприклад збільшення часу життя сенсорної мережі. Одним із методів рішення даної проблеми є введення мобільного стоку [4].

Мета даної кваліфікаційної роботи полягає в дослідженні методів оптимізації шляху мобільного стоку сенсорної інфокомунікаційної мережі.

1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО БЕЗПРОВОДОВІ СЕНСОРНІ МЕРЕЖІ

1.1. Поняття безпроводової сенсорної мережі

Сенсорна мережа являє собою розподілену самостійну конфігурацію безпроводову мережу, що складається з малогабаритних інтелектуальних сенсорних пристроїв [4]. Кожне пристрій оснащений мікро контролером, приєднаним до антенної системи, елементом живлення і набором датчиків для вимірювання параметрів навколишнього середовища, наприклад, температури, освітленості, вібрації, тиску, рівня шуму та інших.

Сенсорні мережі є окремим випадком ситуаційних (в англійській літературі - ad hoc) мереж, які представляють собою розподілені системи рівноправних вузлів, в яких кожен вузол може обмінюватися даними з сусідніми вузлами. Відмінність сенсорних мереж в тому, що в них елементи можна чітко розділити по набору виконуваних функцій. Виходячи з цього виділяють три основних типи мережевих вузлів:

- кінцеве пристрій, що виконує функції збору інформації з датчиків і передачі їх в мережу. Як правило, більшу частину часу він знаходиться в режимі зниженого енергоспоживання, в якому основні споживачі енергії - мікроконтролер і приймач - вимкнені;
- маршрутизатор представляє собою елемент мережі, що виконує функції ретрансляції даних, що приходять з кінцевих пристроїв, до точки збору даних (координатору);
- координатор (шлюз, стік) є елементом, який приймає дані з усієї мережі і передає їх з додатком верхнього рівня по провідному або більш високошвидкісного безпроводового інтерфейсу. Як правило, координатор має постійне джерело живлення і, на відміну від інших вузлів мережі, не обмежений в ресурсах [4].

Такий поділ використовується для побудови централізованих систем моніторингу та управління, в яких координатор або сам обробляє інформацію з усієї мережі, або передає її на пристрій з великою кількістю ресурсів (наприклад, персональний комп'ютер). Розподілені системи на сенсорних мережах поки мало поширені через обмежений функціонал кінцевих вузлів.

На сьогоднішній день в області сенсорних мереж широко застосовуються апаратні рішення з наступними характеристиками:

- RISC-процесор з частотою від 8 до 32 МГц;
- обсяг оперативної пам'яті від 8 до 192 Кбайт;
- обсяг зовнішньої флеш-пам'яті від 0,5 до 8 Мбайт.

Особливо важливими є характеристики споживаної потужності в різних режимах. Типові значення наведені в табл. 1.1 [5].

Таблиця 1.1 - Режими роботи безпроводових модулів БСМ

Режим роботи	Позначення потужності	Типове значення, мВт
Прийом	P_{rx}	52
Передача	P_{tx}	45
Обробка	P_a	20
Режим сну	P_s	0.03

1.2. Структура безпроводових сенсорних мереж

Концепція сенсорних мереж передбачає, що будь-який вузол працює від автономного джерела живлення. Якщо таким джерелом є звичайний акумулятор (типорозміри AA, AAA та подібні) з не поповнюваною в процесі роботи енергією, то в певний момент часу він розряджається та автономний пристрій припиняє працювати. Через те, що будь-який елемент мережі виконує деякий набір задач, що від початку покладений на нього, вихід з ладу може значити наступне: у випадку, якщо набір задач, що виконується вузлом, не критичний, то можна казати про зниження якості обслуговування мережі (тимчасовому або остаточному). Поняття якості обслуговування (QoS), у том числі стосовно БСМ, детально досліджується в роботі [2]. Часовий спад якості обслуговування передбачає, що через деякий час задачі вузла, що вийшов з ладу, будуть перекладені на інші пристрої мережі. При остаточному падінні якості обслуговування мережа припиняє виконувати одну або декілька функцій.

Розрізнення сенсорних мереж у тому, що в них елементи можна чітко розділити за набором функцій, що вони виконують. Виходячи з цього виділяють три основних типи вузлів мережі [6].

1) Кінцевий пристрій, що виконує функцію збирання даних з датчиків та передачі їх в мережах. За звичай, більшу частину часу він знаходиться в режимі зниженого енергоспоживання, в якому основні споживачі енергії - мікроконтролер и прийомопередавач - вимкнені.

2) Маршрутизатор являє собою елемент мережі, що виконує функції ретрансляції даних, які надходять з кінцевих пристроїв. Відомо, що в протоколах маршрутизації традиційних мереж використовуються метрики, що націлені на збільшення пропускної здатності мережі або зменшення затримок даних, що передаються. Схожими метриками можуть слугувати кількість проміжних вузлів (хопів) до адресата, пропускна здатність каналу зв'язку, рівень навантаження лінії. У сенсорних мережах часто застосовується метрика залишкової енергії вузлів на шляху до стоку.

3) Координатор (шлюз, стік) є елементом, що приймає дані зі всієї мережі та передає їх додатку верхнього рівня по провідному або більш високошвидкісному безпроводному інтерфейсу. За звичай, координатор має постійне джерело живлення, на відміну від інших вузлів мережі, також він не обмежений в ресурсах. Якщо вузол є ключовою ланкою мережі, наприклад, виконуючи задачі маршрутизації більшої кількості потоків, його відмова та неможливість динамічної заміни означає відмову всієї мережі.

Такий розподіл використовується для побудови централізованих систем моніторингу та керування, в яких координатор або сам оброблює інформацією зі всієї мережі, або передає її на пристрої з більшою кількістю ресурсів (наприклад, персональний комп'ютер). Розподілені системи в сенсорних мережах не розповсюджені через обмежений функціонал кінцевих вузлів.

На рисунку 1.1 представлена можлива топологія безпроводової сенсорної мережі. В ній знаходяться велика кількість кінцевих пристроїв, що передають інформацію до маршрутизаторів та координатор, що займається збором даних та передачу їх до центрального комп'ютера. Потім зібрана інформація піддається аналізу, на основі якого можна приймати різноманітні рішення залежно від сфери, в якій задіяна побудована безпроводна сенсорна мережа.

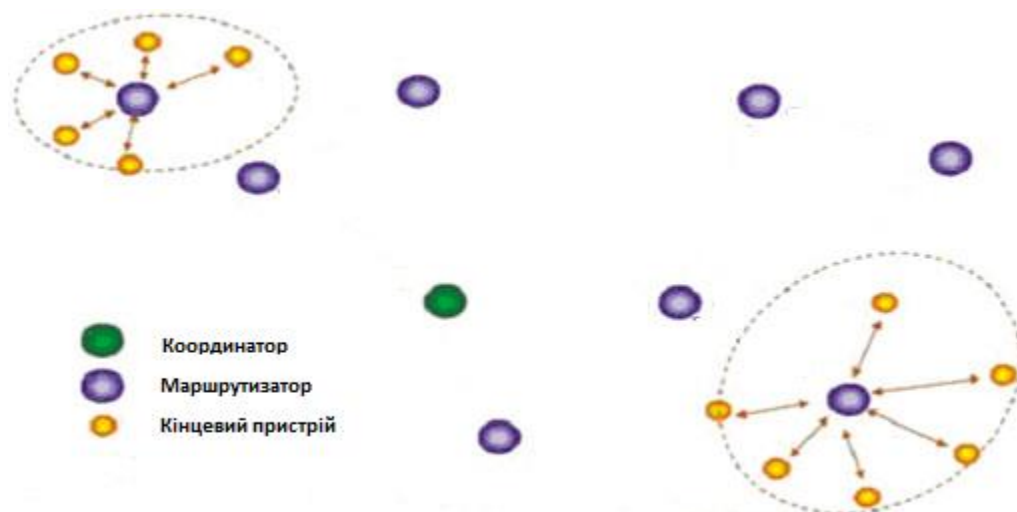


Рисунок 1.1 - Можлива топологія безпроводних сенсорних мереж

1.3. Практичне застосування безпроводових сенсорних мереж

Сенсорні мережі традиційно знаходять своє застосування в різних системах моніторингу та управління, таких як охоронне-пожежне спостереження, клімат-контроль, віддалений збір свідчень з побутових і промислових датчиків, екологічний моніторинг.

Існує і ряд нестандартних додатків, спочатку невластивих БСМ, але, тим не менш, активно вивчаються - передача мультимедіа даних, даних інерційних датчиків в системах стеження або захоплення руху.

Останнім часом збільшується число додатків, які передбачають використання мобільних елементів, що актуально в контексті розглянутих в даній роботі методів. Серед найбільш перспективних програм мобільних БСМ можна відзначити наступні [7]:

- мережі підводного моніторингу. В останні роки були вивчені можливості розгортання підводних сенсорних мереж для моніторингу стану світового океану в окремих його областях. Використання традиційних БСМ під водою стикається з цілою низкою труднощів. По-перше, це на кілька порядків більший коефіцієнт загасання, що вимагає використання більш потужних передавачів. По-друге, більш висока ймовірність виходу компонентів з ладу, через яку доводиться закладати в структуру мережі надмірність. Нарешті, під водою відсутні будь-які можливості використовувати альтернативні джерела енергії. Тому для цього додатка моделі і методи збільшення часу автономної роботи БСМ особливо актуальні;

- системи контролю місця розташування речей в просторі. Найпоширеніші приклади - контроль товарів на складі або пересувного обладнання на великому підприємстві;
- системи персональної телемедицини, що складаються з безлічі носіїв сенсорів і дозволяють контролювати ключові показники життєдіяльності людини, при цьому фіксуючи його місцезнаходження при виникненні небезпечних відхилень від норми.;
- у зв'язку з усіма вищеописаними додатками слід також відзначити тенденцію виходу все більшої кількості пристроїв в глобальну мережу інтернет, яка, згідно з рядом досліджень, вже привела до нового покоління інтернету - інтернету речей, в якому звичайні предмети стають повноправними учасниками загального обміну інформацією і взаємодії. Сенсорні мережі є однією з базових технологій інтернету речей, що дозволяють будь-якого предмету отримувати доступ до глобальної мережі.

1.4 Перелік стандартів

Безліч стандартів передачі даних такі як Wi-Fi, WiMAX, Bluetooth, HomeRF, ZigBee і так далі ділять між собою діапазон 2,4 ГГц, тим самим створюючи один одному перешкоди.

1.4.1 Стандарт Wi-Fi

Wi-Fi - торгова марка Wi-Fi Alliance для безпроводових мереж на базі стандарту IEEE 802.11. Ноутбук або комунікатор без підключення до мережі Інтернет сьогодні є практично марним шматком «заліза». Завдяки широкому використанню Wi-Fi для вирішення проблеми підключення до Інтернету цей термін став добре відомим.

Незважаючи на те, що спочатку в деяких прес-релізах WECA фігурувало словосполучення Wireless Fidelity (безпроводова точність), на даний момент від такого формулювання відмовилися, і термін Wi-Fi ніяк не розшифровується.

Продукти, що призначалися спочатку для систем касового обслуговування, були виведені на ринок під маркою WaveLAN і забезпечували швидкість передачі даних від 1 до 2 Мбіт / с. Творець Wi-Fi - Вік Хейз (Vic Hayes) знаходився в команді, що брала участь в розробці таких стандартів, як IEEE 802.11b, IEEE 802.11a і IEEE

802.11g. Зазвичай схема мережі Wi-Fi містить не менше однієї точки доступу і не менше одного клієнта. Також можливе підключення двох клієнтів в режимі точка-точка (Ad-hoc), коли точка доступу не використовується, а клієнти з'єднуються за допомогою мережевих адаптерів безпосередньо [7].

Точка доступу передає свій ідентифікатор мережі (SSID) за допомогою спеціальних сигнальних пакетів на швидкості 0,1 Мбіт / с кожні 100 мс. Тому 0,1 Мбіт / с - найменша швидкість передачі даних для Wi-Fi. Знаючи SSID мережі, клієнт може з'ясувати, чи можливе підключення до даної точки доступу. При попаданні в зону дії двох точок доступу з ідентичними SSID приймач може вибирати між ними на підставі даних про рівень сигналу. Стандарт Wi-Fi дає клієнтові повну свободу при виборі критеріїв для з'єднання.

Пристрої Wi-Fi широко поширені на сучасному ринку. Сумісність устаткування гарантується завдяки обов'язковій його сертифікації з логотипом Wi-Fi. Випромінювання від Wi-Fi - пристроїв в момент передачі даних на два порядки (в 100 разів) менше, ніж від стільникового телефону. Технологія дозволяє розгорнути мережу без прокладки кабелю, що може зменшити вартість розгортання і / або розширення мережі. Місця, де не можна прокласти кабель, наприклад, поза приміщеннями і в будівлях, що мають історичну цінність, можуть обслуговуватися безпроводовими мережами. Технологія дозволяє мобільним пристроям мати доступ до мережі.

Стандарт шифрування WEP може бути відносно легко зламаний навіть при правильній конфігурації (через слабку стійкість алгоритму). Незважаючи на те, що нові пристрої підтримують досконаліший протокол шифрування даних WPA і WPA2, багато старих точки доступу не підтримують його і вимагають заміни.

Прийняття стандарту IEEE 802.11i (WPA2) в червні 2004 року зробило доступною більш ефективну схему аутентифікації і шифрування, яка застосовується в новому обладнанні. Для реалізації протоколів WPA і WPA2 потрібно більш надійний пароль, ніж той, який зазвичай призначається користувачем.

Стандарт IEEE 802.11 визначає два режими роботи мережі - Ad-hoc (BSS - Basic Service Set) і інфраструктурний ESS - Extended Service Set. Режим Ad-hoc (інакше званий «точка-точка») - це проста мережу, в якій зв'язок між станціями (клієнтами) встановлюється безпосередньо, без використання спеціальної точки доступу. У режимі інфраструктурної ESS безпроводова мережа складається, як

мінімум, з однієї точки доступу, підключеної до провідної мережі, і деякого набору безпроводових клієнтських станцій.

Для організації безпроводової мережі в замкнутому просторі застосовуються передавачі з антенами направленими у всі сторони. Слід мати на увазі, що через стіни з великим вмістом металевої арматури (в залізобетонних будівлях такими є несучі стіни) радіохвилі діапазону 2,4 ГГц іноді можуть взагалі не проходити, тому в кімнатах, розділених подібною стіною, доведеться ставити свої точки доступу. Потужність, яку випромінює передавачем точки доступу або ж клієнтської станції, що працює за стандартом IEEE 802.11, не перевищує 0,1 Вт, але багато виробників безпроводових точок доступу обмежують потужність лише програмним шляхом, і досить просто підняти потужність до 0,2 - 0,5 Вт. Для порівняння - потужність, яку випромінює мобільним телефоном, на порядок більше (в момент дзвінка - до 2 Вт). Оскільки, на відміну від мобільного телефону, елементи мережі розташовані далеко від голови, в цілому можна вважати, що безпроводові комп'ютерні мережі більш безпечні з точки зору здоров'я, ніж мобільні телефони.

Продукти для безпроводових мереж, що відповідають стандарту IEEE 802.11, пропонують чотири рівні засобів безпеки: фізичний, ідентифікатор набору служб (SSID), ідентифікатор управління доступом до середовища (MAC ID - Media Access Control ID) і шифрування [7].

Багато організацій використовують додаткове шифрування (наприклад, VPN) для захисту від вторгнення. На даний момент основним методом злому WPA2 є підбір пароля, тому рекомендується використовувати складні цифро-буквені паролі для того, щоб максимально ускладнити завдання підбору пароля.

1.4.2 Стандарт WiMAX

WiMAX (англ. Worldwide Interoperability for Microwave Access) - телекомунікаційна технологія, розроблена з метою надання універсального безпроводового зв'язку на великих відстанях для широкого спектру пристроїв (від робочих станцій і портативних комп'ютерів до мобільних телефонів). Заснована на стандарті IEEE 802.16, який також називають Wireless MAN (WiMAX слід вважати жаргонним назвою, так як це не технологія, а назва форуму, на якому Wireless MAN і був узгоджений) [7].

WiMAX підходить для вирішення завдання з'єднання точок доступу Wi-Fi один з одним і іншими сегментами Інтернету, а також забезпечення безпроводового

широкопasmугового доступу як альтернативи виділеним лініям і xDSL. WiMAX дозволяє здійснювати доступ в Інтернет на високих швидкостях, з набагато більшим покриттям, ніж у Wi-Fi - мереж. Це дозволяє використовувати технологію в якості магістральних каналів, продовженням яких виступають традиційні виділені і xDSL-лінії, а також локальні мережі. В результаті подібний підхід дозволяє створювати масштабовані високошвидкісні мережі в рамках міст.

WiMAX - це система далекої дії, що покриває кілометри простору, яка зазвичай використовує ліцензовані спектри частот (хоча можливо і використання неліцензованих частот) для надання з'єднання з інтернетом типу точка-точка провайдером кінцевому користувачеві. Різні стандарти сімейства 802.16 забезпечують різні види доступу, від мобільного (схожий з передачею даних у мобільних телефонів) до фіксованого (альтернатива провідного доступу, при якому безпроводове обладнання користувача прив'язане до місця розташування).

На відміну від WiMAX Wi-Fi - це система більш короткої дії, зазвичай покриває десятки метрів, яка використовує неліцензовані діапазони частот для забезпечення доступу до мережі. Зазвичай Wi-Fi використовується користувачами для доступу до їх власної локальної мережі, яка може бути і не підключена до Інтернету. Якщо WiMAX можна порівняти з мобільним зв'язком, то Wi-Fi швидше схожий на стаціонарний безпроводовий телефон (радіотелефон).

WiMAX і Wi-Fi мають абсолютно різний механізм Quality of Service (QoS). WiMAX використовує механізм, заснований на встановленні з'єднання між базовою станцією і пристроєм користувача. Кожне з'єднання засноване на спеціальному алгоритмі планування, який може гарантувати параметр QoS для кожного з'єднання. Wi-Fi, в свою чергу, використовує механізм QoS подібний до того, що використовується в Ethernet, при якому пакети отримують свої пріоритети. Такий підхід не гарантує однаковий QoS для кожного з'єднання.

Набір переваг притаманний всьому сімейству WiMAX, однак його версії істотно відрізняються один від одного. Розробники стандарту шукали оптимальні рішення як для фіксованого, так і для мобільного застосування, але поєднати всі вимоги в рамках одного стандарту не вдалося. Хоча ряд базових вимог збігається, націленість технологій на різні ринкові ніші призвела до створення двох окремих версій стандарту (вірніше, їх можна вважати двома різними стандартами). Кожна з специфікацій WiMAX визначає свої робочі діапазони частот, ширину смуги пропускання, потужність випромінювання, методи передачі та доступу, способи

кодування і модуляції сигналу, принципи повторного використання радіочастот та інші показники. Тому WiMAX-системи, засновані на версіях стандарту IEEE 802.16 e і d, практично несумісні.

Основна відмінність двох технологій полягає в тому, що фіксований WiMAX дозволяє обслуговувати тільки статичних абонентів, а мобільний орієнтований на роботі з користувачами, що пересуваються зі швидкістю до 150 км / г.

Мобільність означає наявність функцій роумінгу і «безшовного» перемикання між базовими станціями при пересуванні абонента (як відбувається в мережах стільникового зв'язку). В окремому випадку мобільний WiMAX може застосовуватися і для обслуговування фіксованих користувачів. З винаходом мобільного WiMAX все більший акцент робиться на розробці мобільних пристроїв, в тому числі, спеціальних телефонних трубок (схожих на звичайний мобільний смартфон), і комп'ютерної периферії (USB-радіо модулів і PC card). Устаткування для використання мереж WiMAX поставляється кількома виробниками і може бути встановлено як в приміщенні (пристрої розміром зі звичайний xDSL-модем), так і поза ним. Слід зауважити, що обладнання, яке розраховане на розміщення усередині приміщень і не вимагає професійних навичок при установці, звичайно, зручніше, проте здатне працювати на значно менших відстанях від базової станції, ніж професійно встановлені зовнішні пристрої. Тому обладнання, встановлене всередині приміщень, вимагає набагато більших інвестицій в розвиток інфраструктури мережі.

У загальному вигляді WiMAX мережі складаються з таких основних частин: базових і абонентських станцій, а також обладнання, що зв'язує базові станції між собою, з постачальником сервісів і з Інтернетом. Структура мереж сімейства стандартів IEEE 802.16 схожа з традиційними GSM-мережами (базові станції діють на відстанях до десятків кілометрів, для їх установки не обов'язково будувати вежі - допускається установка на дахах будинків при дотриманні умови прямої видимості між станціями). WiMAX застосовується як для вирішення проблеми «останньої милі», так і для надання доступу в мережу офісним та районним мережам.

Для з'єднання базової станції з абонентською використовується високочастотний діапазон радіохвиль від 1,5 до 11 ГГц. В ідеальних умовах швидкість обміну даними може досягати 70 Мбіт / с, при цьому не потрібно забезпечення прямої видимості між базовою станцією і приймачем. Між базовими

станціями встановлюються з'єднання (прямої видимості), що використовують діапазон частот від 10 до 66 ГГц, швидкість обміну даними може досягати 140 Мбіт / с. При цьому принаймні одна базова станція підключається до мережі провайдера з використанням класичних дротових з'єднань. Однак, чим більше число БС підключено до мереж провайдера, тим вище швидкість передачі даних і надійність мережі в цілому [7].

1.4.3 Стандарт Bluetooth

Bluetooth забезпечує обмін інформацією між такими пристроями як персональні комп'ютери (настільні, кишенькові, ноутбуки), мобільні телефони, принтери, цифрові фотоапарати, мишки, клавіатури, джойстики, навушники, гарнітури на надійній, недорогій, повсюдно доступній радіочастоті для ближнього зв'язку. Безпроводовий канал дозволяє цим пристроям повідомлятися, коли вони знаходяться в радіусі від 1 до 200 м один від одного (дальність сильно залежить від перешкод і завад), навіть у різних приміщеннях. Варто відзначити, що компанія AIRcable випустила Bluetooth-адаптер Host XR з радіусом дії близько 30 км. Для спільної роботи Bluetooth-пристроїв необхідно, щоб всі вони підтримували загальний профіль. Профіль - набір функцій або можливостей, доступних для певного пристрою Bluetooth. Технологія Bluetooth спирається на неліцензованому (практично скрізь крім Росії) частотний діапазон 2,4 - 2,4835 ГГц. При цьому використовуються широкі захисні смуги: нижня межа частотного діапазону становить 2 ГГц, а верхня - 3,5 ГГц. Частота (положення центру спектра) задається з точністю ± 75 кГц. Дрейф частоти в цей інтервал не входить. Кодування сигналу здійснюється за дворівневою схемою GFSK (Gaussian Frequency Shift Keying). Логічного 0 і 1 відповідають дві різні частоти. В обумовленій частотній полосі виділяється 79 радіоканалів по 1 МГц кожен [8].

1.4.4 Стандарт HomeRF

HomeRF - безпроводова технологія, спеціально орієнтована на мережі, створювані в домашніх умовах. Головна ідея HomeRF полягає в тому, що у домашніх користувачів потреби зовсім відмінні від потреб корпоративних користувачів. Це означає, що і рішення, які для них потрібні, спеціально для них і розроблені. HomeRF прагне працювати в цій ніші ринку, поставляючи пристрої, які

досить легко встановлюються, прості у використанні і більш доступні, ніж сучасні безпроводові рішення масштабу підприємства.

HomeRF заснований на декількох існуючих стандартах передачі голосу і даних і об'єднує їх в єдине рішення. Воно працює в смузі частот ISM 2,4 ГГц з використанням FHSS. Скачки по частотах відбуваються зі швидкістю від 50 до 100 разів на секунду. Позбавлення від інтерференції відбувається за допомогою рознесення сигналів за часом і частоті.

HomeRF використовує радіопередавачі низької потужності, які подібні до тих, що використовуються в персональних безпроводових мережах стандарту 802.15 на основі технології Bluetooth. Різниця між двома технологіями полягає в тому, що HomeRF орієнтована тільки на ринок домашніх користувачів, включаючи SWAP (Standard Wireless Access Protocol - стандартний протокол безпроводового доступу), який в рамках HomeRF дає можливість більш ефективно обробляти мультимедіа додатки. Передавачі діють на відстані 40-50 м від базової станції і можуть бути вбудовані в картки типу Compact Flash [8].

1.4.5 Стандарт ZigBee

ZigBee - назва набору мережевих протоколів верхнього рівня, які використовують малі, малопотужні радіопередавачі, засновані на стандарті IEEE 802.15.4. Цей стандарт описує безпроводові персональні обчислювальні мережі (WPAN). ZigBee націлена на додатки, яким потрібен тривалий час автономної роботи від батареї і висока безпека передачі даних при невеликих швидкостях їх передачі.

Основна особливість технології ZigBee полягає в тому, що вона при відносно невисокому енергоспоживанні підтримує не тільки прості топології безпроводового зв'язку («точка-точка» і «зірка»), а й складні безпроводові мережі з комірчастою топологією з ретрансляцією і маршрутизацією повідомлень. Области застосування даної технології - це побудова безпроводових мереж датчиків, автоматизація житлових і споруджуваних приміщень, створення індивідуального діагностичного медичного обладнання, системи промислового моніторингу та управління, а також при розробці побутової електроніки і персональних комп'ютерів.

Ім'я бренду походить від поведінки медових бджіл, після повернення їх у вулик. Мережі, утворені за протоколом ZigBee почали розглядатися з 1998, коли

виникла необхідність в само-організованих системах зв'язку ZigBee націлений на додатки, яким потрібен тривалий час автономної роботи від батареї і висока безпека передачі даних, при невеликих швидкостях передачі. ZigBee працює в промислових, наукових і медичних (ISM-діапазон) радіодіапазоні: 868 МГц в Європі, 915 МГц в США і в Австралії, і 2,4 ГГц в більшості країн в світі (під більшістю юрисдикцій країн світу). Так як ZigBee пристрій більшу частину часу перебуває в сплячому режимі, рівень споживання енергії може бути дуже низьким, завдяки чому досягається тривала робота від батареї.

Беручи до уваги такі критерії, як ціна чіпів, дешевизна і швидкість освоєння технології, низьке енергоспоживання і стійкість перед перешкодами, можна сказати, що ZigBee нерідко є зараз кращим вибором. Чіпи для реалізації ZigBee випускають такі відомі фірми, як Texas Instruments, Freescale, Atmel, STMicroelectronics, OKI і т.д. Це гарантує низькі ціни на комплектуючі для даної технології. ZigBee - це технологія, що заповнює нішу низько швидкісних безпроводових мереж з низьким енергоспоживанням, призначених для систем управління з великою кількістю вузлів, таких як системи освітлення в будівлях, системи спостереження за парком промислового обладнання і т.д.

В даний час досить доступними є модулі ZigBee: ETRX2, ETRX3, випущені фірмою Telegesis. Для ознайомлення з ними існують стартові набори, що включають в себе модуль-координатор, який має USB-роз'єм, і три інших модуля, які можна налаштувати на роботі роутера або кінцевого пристрою, що займає датчиками температури і освітленості, тестовими кнопками і т.д. [7].

Порівняння основних стандартів приведені в таблиці 1.2.

Таблиця 1.2 - Порівняльна характеристика основних стандартів безпроводових мереж

Стандарт	ZigBee (IEEE 802.15.4)	Wi-Fi (IEEE 802.11b)	Bluetooth (IEEE 802.15.1)
1	2	3	4
Частотний діапазон, ГГц	2,4-2,483	2,4-2,483	2,4-2,483
Пропускна здатність, кбіт / с	250	11000	723,1

Продовження таблиці 1.2

1	2	3	4
Розмір стека протоколу, кбайт	32-64	більше 1000	більше 250
Час неперервної роботи від батареї, дні	100-1000	0,5-5	1-10
Максимальне число вузлів в мережі	65536	10	7
Діапазон дії, м	10-100	20-300	10-100

Наведені в таблиці 1.2 характеристики показують, що кращим вибором для сенсорної мережі є стандарт ZigBee, так як його час неперервної роботи від батареї найбільший але в залежності від поставленої задачі можна використовувати і інші технології.

1.5 Моделі передачі даних в безпроводових сенсорних мережах

Прийнято виділяти чотири базових моделі збору даних в БСМ.

1) Подієва модель використовується в мережах, які фіксують деякі події в певній галузі, наприклад спрацьовування тривожного датчика, вихід фізичного параметра за допустимий діапазон, близькість розряду акумулятора і ряд інших. Важливою особливістю є те, що в загальному випадку точний час настання подій ніяк не можна передбачити, тому при дослідженні мереж використовуються імовірнісні характеристики виникнення подій як випадкового процесу. Як правило, до самої мережі пред'являються деякі вимоги за максимальною затримки передачі інформації про подію в центр збору даних [9].

2) При передачі за розкладом, так само як і в подієвої моделі, процес відправки повідомлень ініціюється елементами системи, а не центром збору даних, проте він прив'язаний до певного розкладу, що дає можливість синхронізувати роботу вузлів мережі.

3) Збір даних за запитом застосовується в системах, що накопичують інформацію про деяке явище, щоб потім на вимогу передати її в центр. Прикладами

можуть служити система екологічного моніторингу, система моніторингу будівельних конструкцій та подібні до них. Затримки передачі даних, як правило, некритичні.

4) Часто застосовується гібридна модель, що поєднує в собі особливості перших трьох. Наприклад, тестові повідомлення про стан мережі можуть передаватися через фіксовані інтервали, в той же час інформація про виникаючі події передається згідно подієвої моделі [4].

1.6. Поняття часу життя мережі і методи збільшення часу життя безпроводової сенсорної мережі

Концепція сенсорних мереж передбачає, що будь-який вузол працює від автономного джерела живлення. Якщо таким джерелом є звичайний акумулятор (типорозміри AA, AAA і подібні) з непоповнювані в процесі роботи енергією, то в певний момент часу він розряджається і автономний пристрій перестає працювати. Так як будь-який елемент мережі виконує певний набір завдань, спочатку покладений на нього, вихід з ладу може означати наступне [10]:

- у разі, якщо набір завдань, що виконується вузлом некритичний, то можна говорити про падіння якості обслуговування мережі (тимчасовому або остаточному). Поняття якості обслуговування (QoS), в тому числі стосовно БСМ, детально досліджується в роботі. Тимчасове падіння якості обслуговування передбачає, що через деякий час завдання вийшов з ладу вузла будуть перекладені на інші пристрої мережі. При остаточному падінні якості обслуговування мережу перестає виконувати одну або кілька функцій;

- якщо вузол є ключовою ланкою системи, наприклад, виконуючи завдання маршрутизації великого числа потоків, то його відмова і неможливість динамічної заміни означає відмову всієї мережі.

Так як в загальному випадку всі елементи БСМ є автономними, обов'язково настає момент, коли мережу більш не може вирішувати покладені на неї завдання. Час від початку роботи мережі до даного моменту називається часом життя або часом автономної роботи мережі (*network lifetime*). Проблема полягає в тому, що в кожному окремому випадку момент виходу мережі з ладу може визначатися по-різному, в залежності від вимог до якості обслуговування [9].

У зв'язку з вищеописаним важливо також сказати про дві ключові функції сенсорних мереж - самоорганізації та самовідтворення. Самоорганізація є процес самостійного освіти, настройки і підтримки роботи безпроводової мережі з динамічним регулюванням параметрів і логіки її роботи в залежності від зовнішніх факторів. Метою самоорганізації є створення автономної, автоматично настроюється мережі, яка після розгортання може функціонувати без втручання оператора.

Самовідновлення тісно пов'язане з самоорганізацією і передбачає, що при виході з ладу окремих вузлів мережі через певний інтервал часу мережа перебудовується і знову починає виконувати покладені на неї функції.

Функції самоорганізації і самовідновлення, як правило, описуються в стандартах мережевого рівня еталонної моделі OSI.

При вирішенні практичних завдань на базі автономних БСМ виникають дві основні задачі, пов'язані з показником часу життя [10].

1) Оцінка передбачуваного часу життя мережі при заданих характеристиках апаратних засобів і алгоритмах її роботи.

2) Збільшення часу життя за рахунок застосування ряду методів і алгоритмів.

Зокрема, слід згадати про зв'язок поняття часу життя з поняттям енергоефективності (energy efficiency) для сенсорних мереж. Енергоефективність часто вживається в якості характеристики стандартів, алгоритмів і протоколів, а її досягнення або підвищення ставиться в якості однієї із завдань технічних проектів в самих різних областях.

На сьогоднішній день універсального визначення поняття енергоефективності немає - воно залежить від конкретної предметної області, проте часто використовується наступний підхід: енергоефективність залежить від того, наскільки раціонально в деякій системі використовується, надається їй ззовні енергія. Так вдається наступне визначення коефіцієнта енергоефективності:

$$\varepsilon = \frac{W_{\text{п}}}{W_{\text{п}} + W_{\text{нп}}}, \quad (1.1)$$

де $W_{\text{п}}$ - корисно використана енергія;

$W_{\text{нп}}$ - непродуктивні витрати.

В роботах, присвячених БСМ, поняття енергоефективності часто не відрізняються від поняття часу автономної роботи. Тобто вважається, що більший час автономної роботи забезпечує більшу енергоефективність. Однак для спростування цього факту можна навести такий простий приклад: мережа, в якій відсутня передача корисних даних, а енергія витрачається тільки на паразитні процеси ($W_{\text{нп}}$), очевидно, має на порядок більший час автономної роботи, ніж мережу, яка транслює дані. Але оскільки енергія не витрачається в корисних цілях ($W_{\text{п}} = 0$), коефіцієнт ефективності дорівнює нулю.

Тому необхідно таке визначення часу життя мережі, яке б включало в себе деякі вимоги щодо забезпечення якості обслуговування, яким мережа повинна задовольняти.

Перейдемо до опису можливих методів збільшення часу автономної роботи БСМ. До найбільш простих відносяться поліпшення апаратних характеристик пристроїв: зменшення енергоспоживання окремих компонентів, оптимізація їх розміщення на кристалі або друкованої плати або збільшення ємності батарей. Дослідження даних можливостей відноситься до суміжних галузей (електроніка, радіофізика, хімія, схемотехніка та ін) [11].

Проте, слід зазначити, що у способі є як фізичні (передача даних по радіоканалу на задану відстань, так само як і обробка даних мікропроцесором, вимагають певних енергетичних витрат) і вартісні обмеження (використання більш енергоефективних компонентів призводить до подорожчання систем). Крім того, використання великих по ємності батарей неминуче призводить до збільшення розміру пристроїв, в той час як сама концепція сенсорних мереж передбачає їх мініатюрність

З точки зору програмних алгоритмів обробки даних на вузлах системи можливі наступні варіанти:

- стиснення даних. Даний метод має свої межі, крім того в сенсорних мережах самі дані, як правило, невеликі за обсягом, тому їх стиснення не дає великого ефекту;

- накопичення даних і їх подальша передача великими блоками. Метод заснований на тому, що в сучасних безпроводових стандартах будь-яка передача цифрового пакету пов'язана з додатковими накладними витратами. Тому вигідніше передавати дані великими блоками в одному пакеті [9].

Останні дослідження в області мініатюрних перетворювачів альтернативної енергії (МЕН, Micro-Energy Harvesters) відкрили ряд можливостей для створення повністю автономних вузлів сенсорної мережі при збереженні їх невеликих розмірів. Відомий ряд готових рішень для підключення сенсорних вузлів до мініатюрних сонячних батарей, перетворювачів вібраційної енергії і термогенератор на основі елемента Пелетье [13].

Однак на сьогоднішній день жодне з рішень по збору і перетворення альтернативної енергії ще не знайшло масового застосування в реальних мережах збору даних, що складаються з сотень вузлів, перш за все, через високу вартість, що включає в себе витрати на регулярне обслуговування. Але в перспективі даний підхід може стати одним з провідних і в кінцевому рахунку вирішити проблему обмеженого часу життя БСМ.

Мобільність створює в безпроводових мережах ряд нових завдань, які потрібно вирішувати при проектуванні систем з рухомими вузлами, причому майже на всіх рівнях моделі OSI. Ключові завдання, пов'язані з підтримкою мобільності окремих вузлів мережі, вирішуються на мережевому рівні. Вони включені в концепцію управління мобільністю.

Управління мобільністю (Mobility Management) включає в себе набір алгоритмів і протоколів, необхідних для того, щоб мобільні пристрої могли ефективно обмінюватися даними по безпроводовому каналу в ситуації, коли їх місце розташування в просторі змінюється з плином часу. Приведена наступна класифікація завдань управління мобільністю [14].

1) Контроль місцезнаходження мобільних пристроїв, що задає такі основні параметри:

- спосіб визначення місця розташування мобільних пристроїв в просторі;
- метод адресації мобільних пристроїв;
- формат і розміщення баз даних з інформацією про географічне положення мобільних елементів;
- алгоритм оновлення інформації про місцезнаходження.

2) Управління перемиканням - щоб змінити точку з'єднання мобільного елемента з мережею, що включає наступні завдання:

- визначення умов запуску процедури перемикання (ослаблення сили сигналу, перевантаження каналу зв'язку, наявність кращого з'єднання, зміна топології мережі та ін.);

- встановлення нового з'єднання;
- організація маршрутизації пакетів через нову точку доступу.

Однак якщо для традиційних мереж рішення перерахованих вище завдань спрямоване головним чином на забезпечення ефективного перебудування мережі (наприклад, при переході мобільного елемента від однієї базової станції до іншої) за критерієм мінімізації затримок, то в сенсорних мережах критерієм виступає мінімізація додаткової споживаної енергії.

Для вирішення завдань маршрутизації в БСМ з мобільним стоком були запропоновані спеціалізовані алгоритми і протоколи. Протокол прямого поширення (Directed Diffusion) працює таким чином, що вузли мережі не дозволяють отримувати стійкі таблиці маршрутизації. Запити інформації кожен раз ініціюються стоком, після чого вони поширюються по мережі за допомогою ширококомовної трансляції. Вузол-адресат, який отримав такий запит, проводить збір необхідних даних і відправляє їх по зворотному маршруту, створеному в процесі вихідного запиту. Маршрут при цьому зберігається і може використовуватися повторно.

Був запропонований вдосконалений варіант протоколу для підтримки мобільності стоку. Він відрізняється від початкового протоколу тим, що фіксує в структурі мережі спеціальний віртуальний вузол (Relay Node), що розташовується в безпосередній близькості зі стоком і ініціює процедуру відновлення маршрутів в разі, якщо стік змінює своє положення.

Схожі принципи представлені в протоколах ART (Adaptive Reversal Tree), ERUP (Efficient Route Update Protocol) і SOT (Sink Oriented Tree).

2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ОПТИМІЗАЦІЇ МАРШРУТУ МОБІЛЬНОГО СТОКУ І МЕТОД ЇЇ ВИРІШЕННЯ

2.1 Математична постановка задачі комівояжера

Мобільний сток повинен відвідати кожен з n ретранслятора, відстані між якими відомі, тільки один раз і повернутися в початковий пункт. Його маршрут повинен мінімізувати сумарну довжину пройденого шляху [12].

Формалізуємо цю задачу, як задачу комівояжера.

Задача комівояжера - одна з найвідоміших задач комбінаторної оптимізації, що полягає в знаходженні самого вигідного маршруту, що проходить через зазначені міста хоча б по одному разу, з подальшим поверненням в початкове місто. В умовах задачі вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій і т.д.) і відповідні матриці відстаней, вартості і т.д. Як правило, вказується, що маршрут повинен проходити через кожне місто тільки один раз - в такому випадку вибір здійснюється серед гамільтонових циклів [15].

Формалізуємо задачу в термінах теорії графів. Якщо немає певних вимог до топології мережі, то безліч маршрутизаторів $N, |N| = n$, де n - кількість маршрутизаторів, можна розглядати як безліч вершин повнозв'язного неорієнтованного графа $G(N, E, C)$ розмірності n . Дугою графа $(i, j) \in E, |E| = n^2$ назовемо ланку маршруту мобільного стоку між i -м і j -м маршрутизаторами. Нехай відома матриця $C = |c_{ij}|$ відстаней між пунктами i -м і j -м ($i = \overline{1, n}; j = \overline{1, n}; i \neq j$). Матриця відстаней являє собою матрицю вагових функцій дуг, яка називається метрикою вартості з'єднання.

Модель задачі комівояжера матиме вигляд:

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij}. \quad (2.1)$$

при обмеженнях:

$$\sum_{i=1}^n X_{ij} = 1, j = \overline{1, n}; \quad (2.2)$$

$$\sum_{i=1}^n X_{ij} = 1, i = \overline{1, n}; \quad (2.3)$$

$$X_{ij} \in \{0,1\}, i = \overline{1, n}, j = \overline{1, n}; \quad (2.4)$$

$$u_i - u_j + nx_{ij} \leq n - 1, i, j = 2, \dots, n, i \neq j; \quad (2.5)$$

$$u_i \geq 0, i \in N. \quad (2.6)$$

Ще одне обмеження сформулюємо наступним чином: шукані змінні $X_{ij} \in \{0,1\}$ повинні утворювати повний контур, що включає всі пункти. Обмеження (2.2) говорить про те, що комівояжер повинен в кожен пункт $j = \overline{1, n}$ заїхати тільки один раз, а обмеження (2.3) - з кожного пункту $i = \overline{1, n}$ виїхати тільки один раз. Обмеження (2.2) - (2.4) і додаткове обмеження на маршруті комівояжера створюють так званий гамільтоновий контур (по імені ірландського математика У. Гамільтона) [16]. Ця вимога забезпечується умовами (2.5), (2.6), де u_i номер кроку, на якому відвідується i -й пункт

Не в кожному графі існує гамільтоновий контур. Сформулюємо достатню умову існування цього контуру: якщо ступінь кожної вершини (число ребер, що виходять з вершини) графа, який має n вершин $n \geq 3$, не менше $n/2$, то на цьому графі можна побудувати гамільтоновим контур. Іншими словами, якщо в групі з n осіб $n \geq 3$ кожен має принаймні $n/2$ знайомих, то всю групу можна посадити навколо столу таким чином, що кожен з них буде знайомий з двома сусідами по столу.

Якщо сформульована достатня умова не виконується для всіх вершин, то це ще не означає, що в графі немає гамільтонового контуру. Умова (необхідна) обов'язкової присутності даного контуру в графі поки не знайдено [7].

Запропоновано досить велика кількість методів пошуку на графі гамільтонових контурів мінімальної довжини, в основі яких закладено підхід організованого, відмінного від повного, перегляду «перспективних» маршрутів.

2.2 Методи вирішення задачі комівояжера

Методи розв'язання задачі комівояжера досить різноманітні і розрізняються застосовуваним інструментарієм, точністю знаходження рішення і складністю необхідних обчислень. Ось лише деякі з них:

- повний перебір (метод «грубої сили», англ. «Brute Force») - полягає в послідовному розгляді всіх можливих маршрутів і виборі з них оптимального. Метод найпростіший і точний, але неефективний і при великій кількості міст його застосування стає тяжким через значні витрати часу і ресурсів на перебір величезної кількості варіантів вирішення завдання. Для прискорення і підвищення ефективності повного перебору використовуються різні прийоми: метод гілок і меж, паралельні обчислення, радужні таблиці;

- випадковий перебір - в цьому випадку обчислюються не всі можливі варіанти маршруту, а лише деякі вибрані у випадковому порядку (наприклад, за допомогою генератора випадкових чисел). З розглянутих варіантів потім вибирається найкращий. Звичайно, найімовірніше отримане рішення не буде оптимальним (втім, воно не буде і найгіршим), але зате даний метод вимагає менших витрат часу і обчислювальних ресурсів, а тому в деяких випадках його застосування виправдано;

- динамічне програмування - ключова ідея полягає в обчисленні та запам'ятовуванні пройденого шляху від вихідного міста до всіх інших, подальшому збільшенні до нього відстаней від поточних міст до решти, і так далі. У порівнянні з повним перебором цей метод дозволяє істотно скоротити обсяг обчислень;

- жадібні алгоритми (англ. «Greedy») - засновані на знаходженні локально оптимальних рішень на кожному етапі обчислень і допущенні, що знайдене таким чином підсумкове рішення буде глобально оптимальним. На кожній ітерації вибирається найкраща ділянка шляху, яка включається в підсумковий маршрут. Метод простий, але його великий недолік в тому, що може виникнути ситуація, коли виявиться, що початкова і кінцева точки маршруту рознесені далеко один від одного і їх доведеться з'єднувати довгим відрізком шляху, що значно знизить ефективність рішення. До жадібних алгоритмів відносяться: метод найближчого сусіда (англ. «Nearest Neighbour»), модифікований метод найближчого сусіда (англ. «Double Ended Nearest Neighbour»), метод найдешевшого включення і т. д. ;

- метод мінімального острівного дерева - пошук маршруту ведеться на графі. Для знаходження оптимального шляху застосовуються різні інструменти: алгоритм Прима, алгоритм Краскала, алгоритм Борувка;
- метод імітації випалювання - один з численних методів Монте-Карло;
- метод еластичної мережі - кожен з можливих маршрутів розглядається як відображення кола на площину;
- мурашиний алгоритм - евристичний метод, заснований на моделюванні поведінки мурах, що шукають шляхи від своєї колонії до джерел їжі. Першу версію такого алгоритму запропонував доктор наук Марко Дориго в 1992 році. Цей метод дозволяє відносно швидко знайти гарне, але не обов'язково оптимальне рішення;
- генетичний алгоритм - ще один евристичний метод, що полягає у випадковому доборі та комбінуванні вихідних параметрів з використанням механізмів, що імітують природній відбір в процесі еволюції (успадкування, мутації, кроссинговер). Незважаючи на досить широкі можливості застосування (і не тільки в логістиці), цей метод часто стає об'єктом критики;
- метод гілок і меж - один з методів дискретної оптимізації, який є розвитком методу повного перебору, але відрізняється від нього відсівом в процесі обчислення підмножин неефективних рішень. Вперше був запропонований в 1960 році англійським професором Алісою Ленд і австралійським математиком Елісон Дойг [17].

2.3 Метод гілок і меж

Для вирішення завдання комівояжера методом гілок і меж необхідно виконати наступний алгоритм (послідовність дій).

- 1) Побудова матриці з вихідними даними - в таблицю заносяться відстані (C_{ij}) між містами (в осередку типу AA, BB і т. д. Ставиться символ M - умовно нескінченно велике число); при цьому рядки відповідають містам відбуття, а стовпцями містам прибуття.
- 2) Знаходження мінімумів по рядках - вкожному рядку визначається мінімальне число (d_i) і виписується в окремий стовпець.
- 3) Редукція рядків - з значень осередків кожного рядка віднімаємо відповідний мінімум ($C_{ij} = C_{ij} - d_i$), не зачіпаючи при цьому клітини з M;

4) Знаходження мінімумів за стовпцями - в кожному стовпці визначається мінімальне число (d_j) і виписується в окремий рядок.

5) Редукція стовпців - з значень осередків кожного стовпчика віднімаємо відповідний мінімум ($C_{ij} = C_{ij} - d_j$), не зачіпаючи при цьому клітини з М.

6) Знаходження кореневої нижньої межі (робимо це тільки один раз, в наступні рази пункт 6 пропускаємо) - обчислюємо нижню межу (мінімально можливу на поточному етапі довжину маршруту) в стартовій (кореневої) точці рішення, як розмір знайденої раніше мінімумів ($H_0 = \sum d_i + \sum d_j$) і починаємо побудова графа (схеми) рішення з внесення в нього кореневої вершини.

7) Обчислення оцінок нульових клітин - вважаємо оцінки (p_{ij}) для кожного осередку з нулями, як суму мінімумів по рядку і стовпцю, в яких розташовується нульова клітина, не враховуючи при цьому саму нульову клітину.

8) Вибір нульової клітини з максимальною оцінкою - шукаємо серед нульових клітин, яка володіє найбільшою оцінкою (якщо таких осередків кілька, вибираємо будь-яку), і отримуємо пару гілок (варіантів) рішення задачі: з включенням в маршрут відрізка шляху відноситься до вибраної комірки і без включення.

9) Редукція матриці - викреслюємо відносяться до обраної клітці рядок і стовпець, а також замінюємо значення осередки відповідної зворотному шляху на М.

10) Обчислення нижньої межі першої гілки (що включає відрізок шляху) - знову знаходимо мінімуми по рядках, проводимо редукцію рядків, знаходимо мінімуми за стовпцями, проводимо редукцію стовпців, після чого обчислюємо локальну нижню межу, як суму попередньої локальної нижньої межі і мінімумів ($H_k = H_k - 1 + \sum d_i + \sum d_j$), і додаємо вершину в граф.

11) Обчислення нижньої межі другої гілки (що не включає відрізок шляху) - вважаємо локальну нижню межу, як суму попередньої локальної нижньої межі і оцінки обраної раніше нульовий клітини ($H_k^* = H_k - 1 + p_{ij}$), і додаємо вершину в граф.

12) Вибір гілки з мінімальним значенням нижньої межі - серед ще не гілками вершин вибираємо володіє мінімальним значенням локальної нижньої межі (незалежно від того, яку гілку розглядаємо в даний момент).

13) Якщо повний маршрут ще не знайдений, продовжуємо рішення, якщо знайдений - переходимо до пункта 10 - якщо маршрут ще не знайдений, то хід

подальшого вирішення залежить від обраної гілки: (а) перша гілка - переходимо до пункту 7, (б) друга гілка - в клітку з максимальною оцінкою ставимо M і переходимо до пункту 2, (в) інша гілка - повертаємося до відповідних їй етапу рішення і таблиці даних.

14) Побудова повного маршруту і визначення його довжини - з'єднуємо всі знайдені раніше відрізки шляху в повний маршрут і вважаємо його загальну довжину (дані беремо з вихідної таблиці). Це короткий опис методики розв'язання задачі комівояжера. Більш докладно етапи обчислення оптимального маршруту описані нижче [18].

Основна ідея методу гілок і меж полягає в тому, що спочатку будують нижню межу φ довжин безлічі маршрутів Z . Потім безліч маршрутів розбивається на дві підмножини таким чином, щоб перше підмножина Z_{ij}^1 складалося з маршрутів, що містять деяку дугу (i, j) , а інше підмножина Z_{ij}^1 не містило цієї дуги. Для кожного з підмножин визначаються нижні межі по тому ж правилу, що і для початкового безлічі маршрутів. Отримані нижні межі підмножин Z_{ij}^1 і Z_{ij}^1 виявляються не менше нижньої межі безлічі всіх маршрутів, тобто $\varphi(Z) \leq \varphi(Z_{ij}^1)$, $\Phi(Z) \leq \varphi(Z_{ij}^1)$.

Порівнюючи нижні межі $\varphi(Z_{ij}^1)$ і $\varphi(Z_{ij}^1)$. Можна виділити те, підмножина маршрутів, яке з більшою ймовірністю містить маршрут мінімальної довжини.

Потім одна з підмножин $\varphi(Z_{ij}^1)$ або $\varphi(Z_{ij}^1)$ за аналогічним правилом розбивається на два нових $\varphi(Z_{ij}^2)$ і $\varphi(Z_{ij}^2)$. Для них знову відшукуються нижні межі $\varphi(Z_{ij}^2)$, і $\varphi(Z_{ij}^2)$ і т.д. Процес розгалуження триває до тих пір, поки не знайдеться єдиний маршрут. Його називають першим рекордом. Потім переглядають обірвані гілки. Якщо їх нижні межі більше довжини першого рекорду, то задача вирішена. Якщо ж є такі, для яких нижні межі менше, ніж довжина першого рекорду, то підмножина з найменшою нижньою межею піддається подальшому розгалуження, поки не переконуються, що воно не містить кращого маршруту.

Якщо ж такий знайдеться, то аналіз обірваних гілок триває щодо нового значення довжини маршруту. Його називають другим рекордом. Процес рішення закінчується, коли будуть проаналізовані всі підмножини.

Для практичної реалізації методу гілок і меж стосовно до задачі комівояжера вкажемо прийом визначення нижніх меж підмножин і розбиття множини маршрутів на підмножини (розгалуження).

Для того щоб знайти нижню межу скористаємося наступним міркуванням: якщо до елементів будь-якого ряду матриці завдання комівояжера (рядку або стовпцю) додати або відняти з них деяке число, то від цього оптимальність плану не зміниться. Довжина ж будь-якого маршрутом комівояжера зміниться на цю величину.

Віднімемо з кожного рядка число, рівне мінімального елементу цього рядка. Віднімемо з кожного стовпця число, рівне мінімального елементу цього стовпця. Отримана матриця називається наведеної по рядках і стовпцях. Сума всіх забраних чисел називається константою приведення.

Константу приведення слід вибирати в якості нижньої межі довжини маршрутів.

Для виділення претендентів на включення в безліч дуг, за якими проводиться розгалуження, розглянемо в наведеній матриці всі елементи, рівні нулю. Знайдемо ступеня Θ_{ij} нульових елементів цієї матриці. Ступінь нульового елемента Θ_{ij} дорівнює сумі мінімального елемента рядку i і мінімального елемента в стовпці j (при виборі цих мінімумів c_{ij} - не враховується). З найбільшою ймовірністю шуканого маршруту належать дуги з максимальним ступенем нуля.

Для отримання платіжної матриці маршрутів, що включає дугу (i, j) викреслюємо в матриці рядок i і стовець j , а щоб не допустити утворення циклу в маршруті, замінюємо елемент, який замикає поточну ланцюжок на нескінченність.

Безліч маршрутів, що не включають дугу (i, j) отримуємо шляхом заміни елемента c_{ij} на нескінченність [12].

2.4 Мурашиний алгоритм

Мурашиний алгоритм (алгоритм оптимізації наслідуванням мурашиної колонії, англ. ant colony optimization, ACO) - один з ефективних поліноміальних алгоритмів для знаходження наближених рішень завдання комівояжера, а також вирішення аналогічних завдань пошуку маршрутів на графах. Суть підходу полягає в аналізі та використанні моделі поведінки мурах, що шукають шляхи від колонії до джерела живлення, і являє собою метаевристическую оптимізацію. Перша версія алгоритму, запропонована доктором наук Марко Доріго в 1992 році, Була спрямована на пошук оптимального шляху в графі [19].

2.4.1. Біологічні принципи поведінки мурашиної колонії.

Мурахи відносяться до соціальних комах, що створює колективи. Колективна система здатна вирішувати складні динамічні завдання з виконання спільної роботи, яка не могла б виконуватися кожним елементом системи окремо в різноманітних середовищах без зовнішнього управління, контролю або координації. У таких випадках говорять про ройовий інтелект, як про хитромудрих способи кооперативного поведінки, тобто стратегії виживання.

Одним з підтверджень оптимальності поведінки мурашиних колоній є той факт, що мережа гнізд суперколоній близька до мінімального остовного дерева графа їх мурашників.

Основу поведінки мурашиної колонії становить самоорганізація, що забезпечує досягнення загальних цілей колонії на основі низкоуровневого взаємодії. Колонія не має централізованого управління, і її особливостями є обмін локальної інформацією тільки між окремими особинами (прямий обмін - їжа, візуальні і хімічні контакти) і наявність непрямого обміну, який і використовується в мурашиних алгоритмах. Таким чином, в загальному випадку розглядаються сліпі мурахи, не здатні відчувати близькість їжі.

Непрямий обмін - стігмержі (stigmergy), являє собою рознесені в часі взаємодія, при якому одна особина змінює деяку область навколишнього середовища, а інші використовують цю інформацію пізніше, коли в неї потрапляють. Біологи встановили, що таке відкладене взаємодія відбувається через спеціальне хімічна речовина - феромон (pheromone), секрет спеціальних залоз, відкладається при переміщенні мурашки. Концентрація феромону на шляху визначає перевагу руху по ньому.

Адаптивність поведінки реалізується випаровуванням феромона, який в природі сприймається мурахами протягом декількох діб. Можемо провести певну аналогію між розподілом феромону в навколишньому колонію просторі, і «глобальною» пам'яттю мурашника, що носить динамічний характер.

2.4.2 Концепція мурашиних алгоритмів

Ідея мурашиного алгоритму - моделювання поведінки мурах, пов'язаного з їх здатністю швидко знаходити найкоротший шлях від мурашника до джерела їжі і адаптуватися до умов, що змінюються, знаходячи новий найкоротший шлях. При

своєму русі мураха мітить шлях феромоном, і ця інформація використовується іншими мурахами для вибору шляху. Це елементарне правило поведінки і визначає здатність мурах знаходити новий шлях, якщо старий виявляється недоступним.

Оригінальна ідея виходить від спостереження за мурахами в процесі пошуку найкоротшого шляху від колонії до джерела живлення.

На рисунку 2.1 зображена основна ідея мурашиного алгоритму [20].

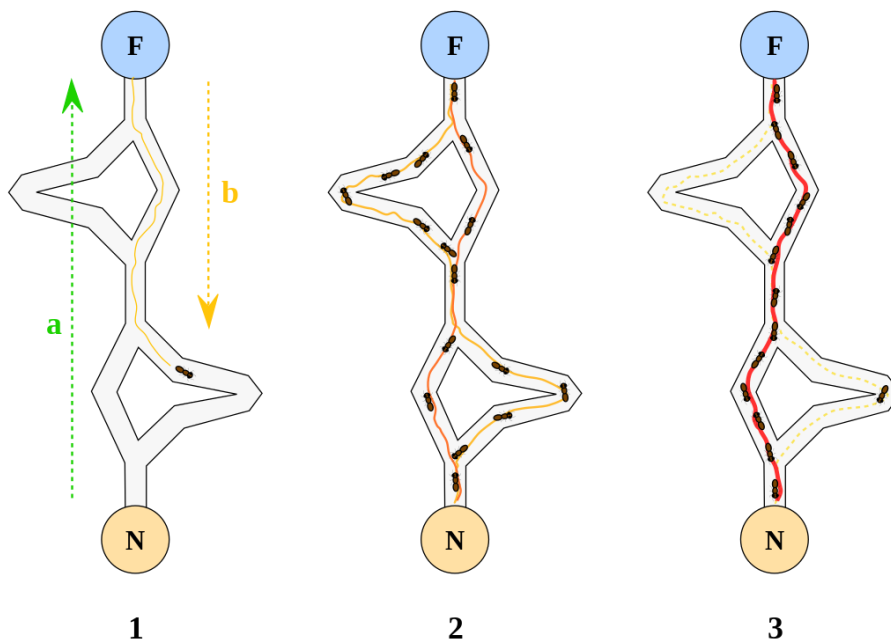


Рисунок 2.1 - Основна ідея мурашиного алгоритму

- 1) Перший мураха знаходить джерело їжі (F) будь-яким способом (a), а потім повертається до гнізда (N), залишивши за собою стежку з феромонів (b).
- 2) Потім мурахи вибирають один з чотирьох можливих шляхів, потім зміцнюють його і роблять привабливим.
- 3) Мурахи вибирають найкоротший маршрут, так як феромони з довших шляхів швидше випаровуються.

Серед експериментів з вибору між двома шляхами нерівної довжини, що ведуть від колонії до джерела живлення, біологи помітили, що, як правило, мурашки використовують найкоротший маршрут. Модель такої поведінки полягає в наступному:

- мурашка (так званий «Бліц») проходить випадковим чином від колонії;
- якщо він знаходить джерело їжі, то повертається в гніздо, залишаючи за собою слід з феромона;

- ці феромони привертають інших мурах, що знаходяться поблизу, які найімовірніше підуть по цьому маршруту;
- повернувшись в гніздо, вони зміцнять феромонних стежку;
- якщо існує 2 маршрути, то коротшим, за той же час, встигнуть пройти більше мурашок, ніж по довгому;
- короткий маршрут стане більш привабливим;
- довгі шляхи, в кінцевому підсумку, зникнуть через випаровування феромонів.

Мурахи використовують навколишнє середовище як засіб спілкування. Вони обмінюються інформацією непрямим шляхом, через феромони, в ході їх «роботи». Обмін інформацією має локальний характер: тільки ті мурахи, які знаходяться в безпосередній близькості від стежок феромонів, можуть дізнатися про них. Така система називається стігмергією і справедлива для багатьох соціальних тварин (була вивчена для випадку будівництва стовпів в гніздах термітів). Даний механізм вирішення проблеми дуже складний і є хорошим прикладом самоорганізації системи. Така система базується на позитивній (інші мурахи зміцнюють феромонних стежку) і негативною (випаровування феромонної стежки) зворотного зв'язку. Теоретично, якщо кількість феромонів залишатиметься незмінним з плином часу по всіх маршрутах, то неможливо буде вибрати шлях. Однак через зворотний зв'язок, невеликі коливання призведуть до посилення одного з маршрутів і система стабілізується до найкоротшому шляху [20].

2.4.3 Узагальнений алгоритм

Будь-мурашиний алгоритм, незалежно від модифікацій, уявімо в наступному вигляді.

Поки умови виходу не виконані.

- 1) Створюємо мурах.
- 2) Шукаємо рішення.
- 3) Оновлюємо феромон.
- 4) Додаткові дії {опціонально}.

Тепер розглянемо кожен крок в циклі більш детально.

- 1) Створюємо мурах.

Стартова точка, куди поміщається мураха, залежить від обмежень, що накладаються умовами завдання. Тому що для кожного завдання спосіб

розміщення мурах є визначальним. Або всі вони поміщаються в одну точку, або в різні з повтореннями, або без повторень.

На цьому ж етапі задається початковий рівень феромону. Він ініціалізується невеликим позитивним числом для того, щоб на початковому етапі ймовірності переходу в наступну вершину були нульовими.

2) Шукаємо рішення.

Ймовірність переходу з вершини i в вершину j визначається за такою формулою.

$$p_{ij}(t) = \frac{\tau_{ij}^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum \tau_{ij}^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}, \quad (2.7)$$

де τ_{ij} - рівень феромону;

d - евристичне відстань;

α, β - константні параметри.

При $\alpha = 0$ вибір найближчого міста найбільш імовірний, тобто алгоритм стає жадібним. При $\beta = 0$ вибір відбувається тільки на підставі феромону, що призводить до субоптимальних рішень.

Тому необхідний компроміс між цими величинами, який знаходиться експериментально.

3) Оновлюємо феромон.

Рівень феромона оновлюється відповідно до наведеної формулою:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum \frac{Q}{L_k}, \quad (2.8)$$

де ρ - інтенсивність випаровування;

$L_k(t)$ - ціна поточного рішення для k -ого мурашки;

Q - параметр, що має значення порядку ціни оптимального рішення, тобто

$\frac{Q}{L_k(t)}$ - феромон, що відкладається k -им мурахою, що використовують ребро (i, j) .

4) Додаткові дії.

Зазвичай тут використовується алгоритм локального пошуку, проте він може також з'явитися і після пошуку всіх рішень [20].

2.4.5 Етапи вирішення задачі за допомогою мурашиних алгоритмів

Для того щоб побудувати відповідний мурашиний алгоритм для вирішення будь-якої задачі, потрібно.

- 1) Уявити завдання у вигляді набору компонент і переходів або набором неорієнтованих зважених графів, на яких мурахи можуть будувати рішення.
 - 2) Визначити значення сліду феромону.
 - 3) Визначити евристику поведінки мурашки, коли будуємо рішення.
 - 4) Якщо можливо, то реалізувати ефективний локальний пошук.
 - 5) Вибрати специфічний АСО алгоритм і застосувати для розв'язання задачі.
- б) Налаштувати параметр АСО алгоритму.

Також визначальними є:

- кількість мурах;
- баланс між вивченням і використанням;
- поєднання з жадібними евристичними або локальним пошуком;
- момент, коли оновлюється феромон.

2.4.5 Застосування мурашиних алгоритмів для задачі комівояжера

Завдання формулюється як задача пошуку мінімального за вартістю замкнутого маршруту по всіх вершин без повторень на повному зваженому графі з n вершинами. Змістовно вершини графа є містами, які повинен відвідати комівояжер, а ваги ребер відображають відстані (довжини) або вартості проїзду. Це завдання є NP-важкою, і точний переборний алгоритм її рішення має факторіальною складність [20].

Моделювання поведінки мурах пов'язано з розподілом феромону на стежці - ребрі графа в задачі комівояжера. При цьому ймовірність включення ребра в маршрут окремого мурашки пропорційна кількості феромону на цьому ребрі, а кількість відкладала феромона пропорційно довжині маршруту. Чим коротше маршрут, тим більше феромона буде відкладено на його ребрах, отже, більшу кількість мурах буде включати його в синтез власних маршрутів. Моделювання такого підходу, що використовує тільки позитивний зворотний зв'язок, призводить

до передчасної збіжності - більшість мурашок рухається по локально оптимальним маршрутом. Уникнути цього можна, моделюючи негативний зворотний зв'язок у вигляді випаровування феромону. При цьому якщо феромон випаровується швидко, то це призводить до втрати пам'яті колонії і забування хороших рішень [20].

Тепер з урахуванням особливостей задачі комівояжера, можемо описати локальні правила поведінки мурах при виборі шляху.

1) Мурахи мають власну «пам'ять». Оскільки кожне місто може бути відвіданий тільки один раз, то у кожного мурашки є список вже відвіданих міст - список заборон. позначимо через J і k список міст, які необхідно відвідати мурашки k , що знаходиться в місті i .

2) Мурахи мають «зором» - видимість є евристичне бажання відвідати місто j , якщо мураха знаходиться в місті i . Будемо вважати, що видимість обернено пропорційна відстані між містами:

$$n_{ij} = \frac{1}{D_{ij}}. \quad (2.9)$$

3) Мурахи мають «нюхом» - вони можуть вловлювати слід феромона, що підтверджує бажання відвідати місто j з міста i на підставі досвіду інших мурах. Кількість феромону на ребрі (i, j) в момент часу t позначимо через $\tau_{ij}(t)$.

На цій підставі можемо сформулювати вероятностно-пропорційне правило, що визначає ймовірність переходу k -ого мурашки з міста i в місто j :

$$\begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha * [n_{ij}]^\beta}{\sum [\tau_{ij}(t)]^\alpha * [n_{ij}]^\beta}, j \in J_{i,k}, \\ P_{ij,k}(t) = 0, j \notin J_{i,k} \end{cases} \quad (2.10)$$

де α, β - параметри, що задають ваги сліду феромону.

При $\alpha = 0$ алгоритм вироджується до жодного алгоритму (буде обраний найближче місто). Зауважимо, що вибір міста є імовірнісним, правило (1) лише визначає ширину зони міста j ; в загальну зону всіх міст $J_{i,k}$ кидається випадкове число, яке і визначає вибір мурашки. Правило (1) не змінюється в ході алгоритму, але у двох різних мурах значення ймовірності переходу будуть відрізнятися, тому що вони мають різний список дозволених міст.

4) Пройшовши ребро (i, j) , мураха відкладає на ньому деяку кількість феромону, яке повинно бути пов'язано з оптимальністю зробленого вибору. Нехай $T_k(t)$ є маршрут, пройдений мурахою k до моменту часу t , $L_k(T)$ - довжина цього маршруту, а Q - параметр, що має значення порядку довжини оптимального шляху. Тоді відкладається кількість феромону може бути задано у вигляді [16]:

$$\Delta\tau_{ij,k(t)} = \begin{cases} \frac{Q}{L_k(t)}, & (i, j) \in T_k(t) \\ 0, & (i, j) \notin T_k(t) \end{cases}. \quad (2.11)$$

Правила зовнішнього середовища визначають, впершу чергу, випаровування феромону. Нехай $\rho \in [0,1]$ є коефіцієнт випаровування, тоді правило випаровування має вигляд

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t); \quad (2.12)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij,k}(t), \quad (2.13)$$

де m - кількість мурах в колонії.

Напочатку алгоритму кількості феромону на ребрах приймається рівним невеликого позитивного числа. Загальна кількість мурах залишається постійним і рівним кількості міст, кожен мураха починає маршрут зі свого міста.

Додаткова модифікація алгоритму може складатися у веденні так званих «елітних» мурах, які підсилюють ребра найкращого маршруту, знайденого з початку роботи алгоритму. Позначимо через T^* найкращий поточний маршрут, через L^* - його довжину. Тоді якщо в колонії є елітних мурах, то ребра маршруту отримають додаткову кількість феромону.

$$\Delta\tau_e = e * \frac{Q}{L^*}. \quad (2.14)$$

2.4.6 Огляд модифікацій класичного алгоритму

Результати перших експериментів із застосуванням мурашиного алгоритму для вирішення завдання комівояжера були багатообіцяючими, проте далеко не

кращими в порівнянні з вже існуючими методами. Однак простота класичного мурашиного алгоритму (названого «мурашиної системою») залишала можливості для доробок - і саме алгоритмічні удосконалення стали предметом подальших досліджень Марко Дориго і інших фахівців в області комбінаторної оптимізації. В основному, ці вдосконалення пов'язані з великим використанням історії пошуку та більш ретельно дослідженням областей навколо вже знайдених вдалих рішень. Нижче розглянуті найбільш примітні з модифікацій [21].

1) Elitist Ant System.

Одним з таких удосконалень є введення в алгоритм так званих «елітних мурах». Досвід показує, що проходячи ребра, що входять в короткі шляхи, мурахи з більшою ймовірністю будуть знаходити ще коротші шляхи. Таким чином, ефективною стратегією є штучне збільшення рівня феромонів на найвдаліших маршрутах. Для цього на кожній ітерації алгоритму кожен з елітних мурах проходить шлях, який є найкоротшим зі знайдених на даний момент.

Експерименти показують, що, до певного рівня, збільшення числа елітних мурах є досить ефективним, дозволяючи значно скоротити число ітерацій алгоритму. Однак, якщо число елітних мурах занадто велике, то алгоритм досить швидко знаходить субоптимальних рішення і застряє в ньому. Як і інші змінні параметри, оптимальне число елітних мурах слід визначати дослідним шляхом.

2) Ant-Q.

Лука Гамбарделла (Luca M. Gambardella) і Марко Дориго опублікували в 1995 році роботу, в якій вони представили мурашиний алгоритм, який отримав свою назву за аналогією з методом машинного навчання Q-learning. В основі алгоритму лежить ідея про те, що мурашину систему можна інтерпретувати як систему навчання з підкріпленням. Ant-Q підсилює цю аналогію, запозичуючи багато ідей з Q-навчання.

Алгоритм зберігає Q-таблицю, що зіставляє кожному з ребер величину, визначальну «корисність» переходу по цьому ребру. Ця таблиця змінюється в процесі роботи алгоритму - тобто навчання системи. Значення корисності переходу по ребру обчислюється виходячи з значень корисностей переходу за наступними ребрами в результаті попереднього визначення можливих наступних станів. Після кожної ітерації корисності оновлюються виходячи з довжин шляхів, доскладу яких були включені відповідні ребра.

3) Ant Colony System.

У 1997 році ті ж дослідники опублікували роботу, присвячену ще одному розробленим ними мурашиного алгоритму. Для підвищення ефективності в порівнянні з класичним алгоритмом, ними були введені три основних зміни.

По-перше, рівень феромонів на ребрах оновлюється не тільки в кінці чергової ітерації, але і при кожному переході мурах з вузла у вузол. Подруге, в кінці ітерації рівень феромонів підвищується тільки на найкоротшому зі знайдених шляхів. По-третє, алгоритм використовує змінене правило переходу: або, з певною часткою ймовірності, мураха безумовно вибирає краще - в відповідність з довжиною і рівнем феромонів - ребро, або робить вибір так само, як і в класичному алгоритмі.

4) Max-min Ant System.

У тому ж році Томас Штютцле (Tomas Stützle) і Хольгер Хоос (Holger Hoos) запропонували мурашиний алгоритм, в якому підвищення концентрації феромонів відбувається тільки на кращих шляхах з пройдених мурахами. Така велика увага до локальних оптимумам компенсується введенням обмежень на максимальну і мінімальну концентрацію феромонів на ребрах, які вкрай ефективно захищають алгоритм від передчасної збіжності до субоптимальних рішень.

На етапі ініціалізації, концентрація феромонів на всіх ребрах встановлюється рівною максимальній. Після кожної ітерації алгоритму тільки один мураха залишає за собою слід - або найбільш успішний на даній ітерації, або, аналогічно алгоритму з елітизмом, елітний. Цим досягається, з одного боку, більш ретельне дослідження області пошуку, з іншого - його прискорення.

5) Asrank.

Бернд Бульнхаймер (Bernd Bullnheimer), Річард Хартл (Richard F. Hartl) і Христині Штраусс (Christine Strauß) розробили модифікацію класичного мурашиного алгоритму, в якому в кінці кожної ітерації мурахи ранжуються у відповідність з довжинами пройдених ними шляхів. Кількість феромонів, що залишається мурахою на ребрах, таким чином, призначається пропорційно його позиції. Крім того, для більш ретельного дослідження околиць вже знайдених вдалих рішень, алгоритм використовує елітних мурах [21].

2.5 Алгоритм імітації випалювання

Алгоритм імітації випалювання (англ. Simulated annealing) - загальний алгоритмічний метод рішення задачі глобальної оптимізації, особливо дискретної і комбінаторної оптимізації. Один із прикладів методів Монте-Карло .

Алгоритм імітації випалювання - це один з методів чисельної оптимізації, в рамках якого шукаємо мінімум або максимум якоїсь функції.

Цей метод є імовірнісним і він може знайти не конкретне значення мінімуму функції, а щось, наближене до нього. Головна проблема, яку вирішує алгоритм імітації випалювання - застрявання в локальних мінімумах, коли результатом оптимізації виявляється не глобальне значення мінімуму, а якесь локальне.

Ідея методу прийшла з випалювання металів - процесу, при якому метал спочатку нагрівають, а потім поступово охолоджують [22].

Ми ітеративно вибираємо точку, в якій може бути мінімум і порівнюємо значення функції в ньому зі значенням в передущей точці. Якщо значення менше, то просто переходимо в нову точку. Якщо ж воно більше, то переходимо в нову точку з певною ймовірністю, яка залежить від значення параметра-температури (чим менше температура, тим менше ймовірність переходу). За рахунок цього можемо виходити з локальних мінімумів.

На рисунку 2.2 зображено приклад роботи алгоритму випалювання

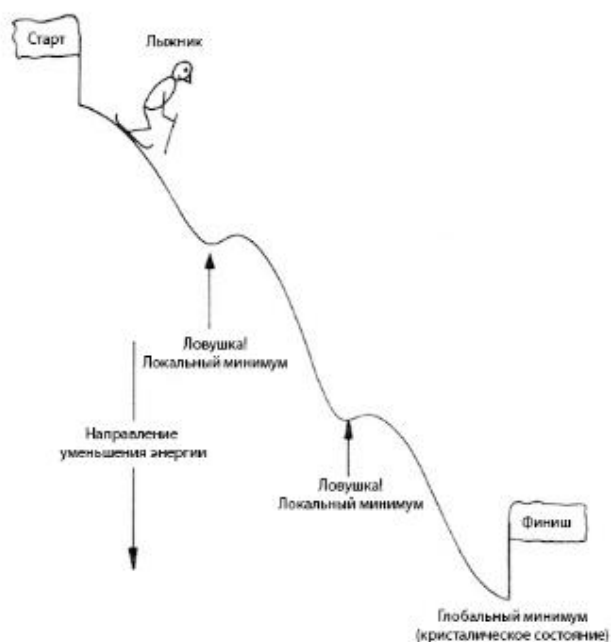


Рисунок 2.2 - Приклад роботи алгоритму випалювання

2.5.1 Основні кроки алгоритму

1) Вибір початкового рішення і початкової температури.

Доброю стратегій є випадковий вибір початкового рішення. Також в якості початкового рішення можна запропонувати рішення отримане іншими методами. Це надає алгоритму базу, на підставі якої він буде будувати більш оптимальне рішення.

Вибір початкової та порогової температуритеж слід проводити експериментально. Природними рекомендаціями можуть служити вибір порогової температуру близьку до нуля, а початкової досить високою.

2) Оцінка початкового рішення.

Цей етап повністю залежить від специфіки завдання. Єдиною вимогою є отримання в якості оцінки одного дійсного числа, яке буде характеризувати оптимальність пропонованого рішення. Це число в алгоритмі імітації випалювання прийнятно називати енергією. Якщо вибір такого числа є скрутним, то, можливо, варто відмовитися від використання пропонованого методу.

3) Основний крок алгоритму.

Основний крок при деякій температурі повторюється кілька разів. Можливо, що один раз. Також можливий варіант із залежністю числа повторів від температури:

- випадкова зміна поточного рішення;

Цей етап сильно залежить від специфіки завдання. Однак, зміна варто проводити локальні. Наприклад, для завдання комівояжера, хорошою стратегій буде обмін, в порядку проходження міст, двох випадкових міст місцями. В результаті зміни у нас буде два рішення: поточне і змінене [22];

- оцінка зміненого рішення;
- критерій допуску.

Для визначеності будемо вважати, що оптимізація полягає в мінімізації енергії. У більшості випадків цей підхід справедливий. Критерій допуску полягає в перевірці та можливу заміну поточного рішення зміненим.

Якщо змінене рішення має меншу енергію, то воно приймається за поточний. Якщо ж змінене рішення має більшу енергію, то воно приймається з імовірністю:

$$P = \exp\left(\frac{-\delta E}{T}\right), \quad (2.15)$$

де P - ймовірність прийняти змінений рішення;

δE - модуль різниці між енергією оптимального рішення і енергій зміненого рішення;

T - поточна температура.

4) Зменшення температури i , якщо температура більше деякого порога, то перехід до основного кроку.

Важливою частиною алгоритму є зменшення температури. При великій температурі ймовірність вибору менш оптимального рішення висока. Однак, в процесі роботи алгоритму температура знижується, і ймовірність вибору менш оптимального рішення знижується.

Вибір способу зменшення температури може бути різним і вибирається експериментально. Головне, щоб температура монотонно спадала до нуля. Доброю стратегією є множення на кожному кроці температури на деякий коефіцієнт трохи менший одиниці [22].

Тепер готові перейти до більш суворого опису методу. Введемо позначення: Нехай S - множина всіх станів (рішень) нашого завдання. Наприклад, для завдання комівояжора це буде безліч все можливих шляхів.

Нехай S_i - стан на i -му кроці алгоритму $s_i \in S, t_i \in R$ - температура на i -му кроці.

Для того, щоб використовувати імітацію випалювання, нам знадобиться визначити три функції:

- функцію енергії або, простіше кажучи, то що оптимізуємо;

$$E: S \rightarrow R. \quad (2.16)$$

E кожного рішення по якомусь правилу ставить у відповідність число. Залежить від конкретного завдання.

- функцію зміни температури з плином часу. Про всяк випадок зазначу, що вона обов'язково повинна бути порядку спадання;

$$T: N \rightarrow R. \quad (2.1)$$

T ставить номеру ітерації i у відповідність температуру. Функція визначає, як довго буде працювати наш алгоритм. Якщо T буде лінійною функцією, то час роботи буде відносно великим. У випадку ж із ступеневою функцією, скажимо:

$$T(i) = \frac{t_1}{\alpha^i}. \quad (2.18)$$

Все закінчиться дуже швидко, але далеко не факт, що не застрянемо в низовині, так і не діставшись до фінішу.

- функцію, яка породжує новий стан;

$$F: S \rightarrow R. \quad (2.19)$$

Функція F на основі попереднього стану породжує новий стан-кандидат, в яке система може перейти, а може і відкинути. позначимо його S_c . Спосіб отримання кандидата повністю залежить від розв'язуваної задачі, конкретну функцію Ви зможете побачити в прикладі [23].

Тепер можемо в загальному вигляді описати алгоритм методу імітації випалювання:

- на вході: мінімальна температура t_{\min} , Початкова температура t_{\max} ;
- задаємо довільний перший стан;
- $t_1 = t_{\max}$;
- $t_i > t_{\min}$;

$$s_c = F(s_i - 1); \quad (2.20)$$

$$\Delta E = E(s_c) - E(s_i - 1); \quad (2.21)$$

$$\Delta E \leq 0, S_i = S_c. \quad (2.22)$$

де $\Delta E > 0$, перехід виконується з вірогідністю $P(\Delta E) = e^{-\frac{\Delta E}{t_i}}$

- знижуємо температуру $t_{i+1} = T(i)$;
- повертаємо останній стан s .

Як бачите, все дуже просто, окремо варто пояснити тільки перехід в новий стан. Якщо енергія «кандидата» менше, він стає новим станом, в іншому випадку, перехід буде імовірнісним (тому метод відносять до класу стохастичних) [18].

Отже, є функція, що характеризує систему, і безліч координат, на якому ця функція задана. Перш за все потрібно поставити початковий стан системи. Для цього береться просто будь-яке випадкове стан. Далі на кожному k -му кроці:

- порівнюємо поточне значення F з найкращим знайденим; якщо поточне значення краще - міняємо глобальне найкраще;
- випадковим чином генеруємо новий стан; розподіл ймовірності для нього має залежати від поточного стану і поточної температури;
- обчислюємо значення функції для згенерованої точки;
- приймаємо або не беремо сгенерованное стан як поточний; ймовірність цього рішення повинна залежати від різниці функцій згенерованого і поточного станів і, звичайно, від температури (чим вище температура, тим більша ймовірність прийняти стан гірше поточного);
- якщо новий стан не прийнято, генеруємо інше і повторюємо дії з третього пункту, якщо прийнято - переходимо до наступної ітерації, знизивши температуру (але частіше перехід до наступного кроку виробляють вбуди-якому випадку, щоб уникнути довгого зациклення).

Процес зупиняється після досягнення певної температури. Речовина охолело в точці з мінімальною енергією.

2.5.2. Застосування алгоритму випалювання для задачі комівояжера

Нехай є деяка функція $f(x)$ від стану x , яку хочемо мінімізувати. В даному випадку x це перестановка вершин (міст) в тому порядку, в якому будемо їх відвідувати, а $f(x)$ це довжина відповідної колії.

Візьмемо в якості базового рішення якийсь стан x_0 (наприклад, випадкову перестановку) і будемо намагатися його покращувати.

Введемо температуру t - якийсь дійсне число (спочатку рівне одиниці), яке буде змінюватися протягом оптимізації і впливати на ймовірність перейти в сусідне стан.

Ще не прийдемо до оптимального рішення або поки не закінчиться час, будемо повторювати такі кроки.

- 1) Зменшимо температуру $t_k = T(t_{k-1})$.
- 2) Виберемо випадкового сусіда x - то є якийсь стан u , яке може бути отримано з x якимось мінімальним зміною.
- 3) З ймовірністю $p(f(x), f(u), t_k)$ зробимо привласнення $x \leftarrow u$.

В кожному кроці є багато свободи при реалізації. Основні евристичні ідеї такі.

1) Напочатку оптимізації наше рішення і так погане, і можемо дозволити собі високу температуру і ризик перейти в стан гірше. В кінці навпаки - наше рішення майже оптимальне, і не хочемо втрачати прогрес. Температура повинна бути високою напочатку і повільно зменшуватися до кінця [19].

2) Алгоритм буде працювати краще, якщо функція $f(x)$ «гладка» щодо цієї зміни, тобто змінюється не сильно.

3) Імовірність повинна бути менше, якщо новий стан гірше, ніж старе. Також ймовірність повинна бути більше при високій температурі.

Наприклад, можна діяти так:

– $t_k = \gamma \cdot (t_{k-1})$, Де γ це якесь число, близьке до одиниці (наприклад, 0.990.99). Воно повинно залежати від планованої кількості ітерацій: оптимізація при низькій температурі майже нічого не мінятиме;

– у випадку з перестановками цим мінімальним зміною може бути, наприклад, своп двох випадкових елементів;

– якщо y не гірше, тобто $f(y) \leq f(x)$, то переходимо в нього вбудь-якому випадку. Інакше робимо перехід в y , з імовірністю $p = e^{-(f(x) - f(y)) / t_k}$ - це експонента негативного числа, і вона дасть можливість в проміжку (0,1).

Взагалі, у виборі конкретних евристик не існує «золотого правила». Всі компоненти алгоритму сильно залежать один від одного і від завдання [24].

3 ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ МАРШРУТУ МОБІЛЬНОГО СТОКУ СЕНСОРНОЇ ІНФОКОМУНІКАЦІЙНОЇ МЕРЕЖІ

3.1 Постановка задачі алгоритмів оптимізації маршруту

Завданням даного дослідження є порівняльний аналіз точного рішення задачі комівояжера, отриманого методом гілок і меж, і наближеного рішення, отриманого евристичними методами. Точне рішення було використано для оцінки похибки евристичних рішень. Визначення похибки рішення дозволило визначити число необхідних ітерацій, а отже, оцінити час отримання допустимого рішення.

Задача комівояжера була вирішена методом гілок і меж в пакеті Matlab з використанням функції `bintprog` для мережі з 10 і 30 вузлів. Як вартості з'єднання між маршрутизаторами мережі було вибрано відстань між ними, яке моделювалося рівномірним розподілом в діапазоні (1-10) км і нормальним розподілом з математичному очікуванню, рівному 5 км, і дисперсією, яка дорівнює 1 км². Як оптимального маршруту мобільного стоку розглядався маршрут мінімальної довжини. Як результат експерименту розглядалася вибіркоче середні довжини оптимального маршруту при 500 експериментах. Маршрут без оптимізації розглядався на безлічі всіх перестановок вузлів мережі.

Для підтвердження достовірності вибіркочих середніх значень, за допомогою процедури Робінса-Монро [25] [27] були отримані рекурсивні оцінки довжини маршруту мобільного стоку. Рекурсивні оцінки довжини маршруту з використанням оптимізації та без оптимізації наведені на рис. 3.1-3.4. Тут L_1 - довжина оптимального маршруту мобільного стоку в кожному з експериментів, L_2 - довжина маршруту мобільного стоку при випадковому виборі маршрутизаторів в кожному з експериментів, m_1 -рекурсивна оцінка умовного математичного очікування довжини оптимального маршруту мобільного стоку, m_2 - рекурсивна оцінка умовного математичного очікування довжини маршруту мобільного стоку при випадковому виборі маршрутизаторів.

В таблиці 3.1 приведені значення вибіркочих середніх довжини маршруту мобільного стоку.

Таблиця 3.1. - Значення вибірових середніх довжини маршруту мобільного стоку

Вибіркові середні довжини маршруту, км,	Рівномірний закон розподілу відстаней між маршрутизаторами		Нормальний закон розподілу відстаней між маршрутизаторами	
	Кількість маршрутизаторів			
	n = 10	n = 30	n = 10	n = 30
Із застосуванням оптимізації	33,5	121,2	44,7	74,9
Без оптимізації	49,4	160,1	62,3	96,7

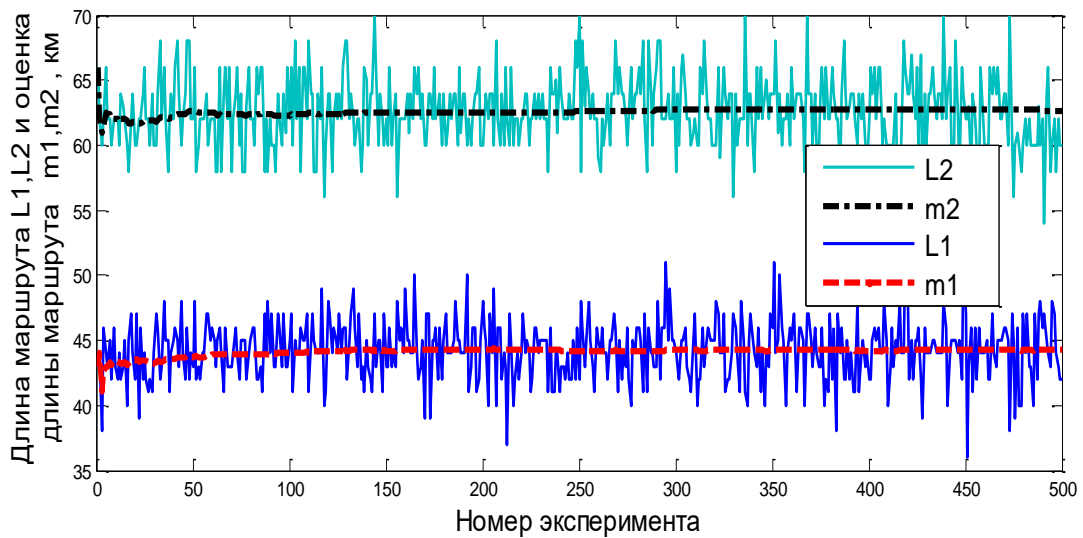


Рисунок 3.1 - Рекурсивна оцінка довжини маршруту мобільного стоку при нормальному законі розподілу відстаней між вузлами, $n = 10$.

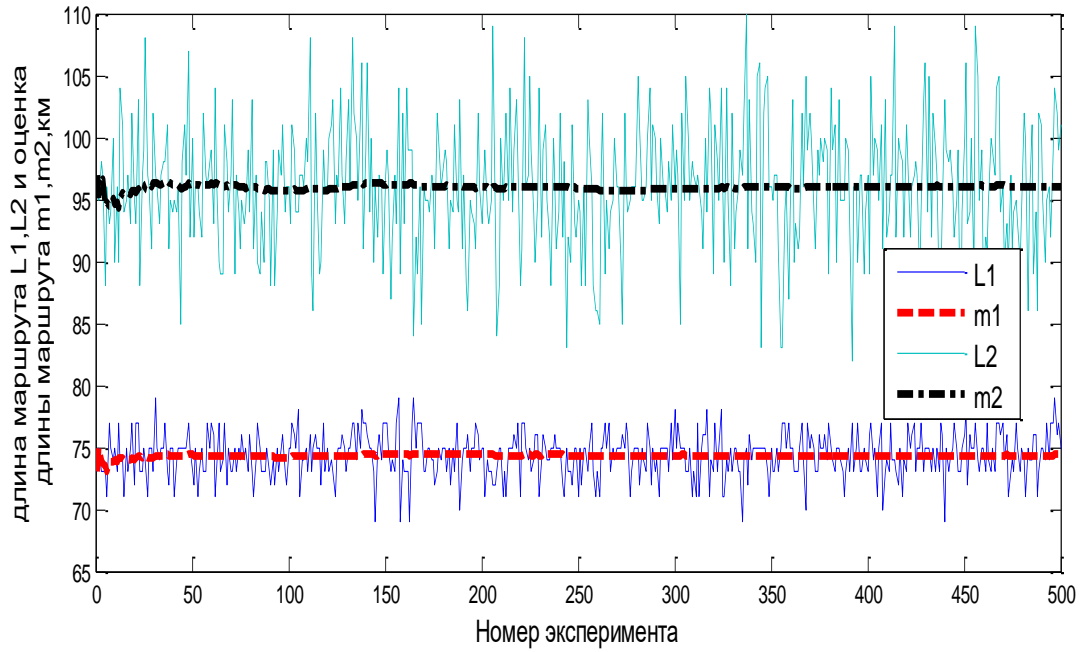


Рисунок 3.2 - Рекурсивна оцінка довжини маршруту мобільного стоку при нормальному законі розподілу відстаней між вузлами, $n = 30$.

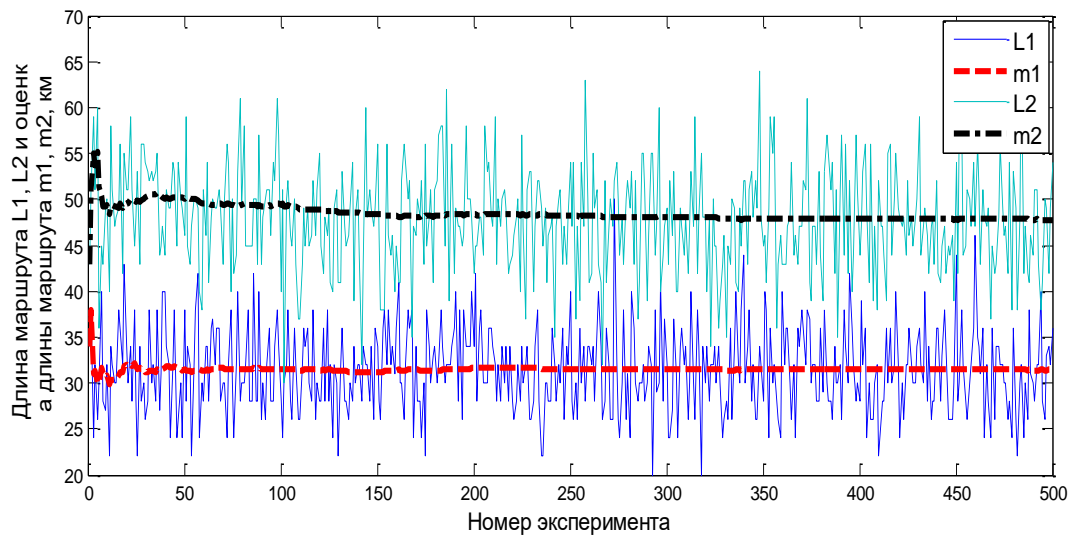


Рисунок 3.3 - Рекурсивна оцінка довжини маршруту мобільного стоку при рівномірному законі розподілу відстаней між вузлами, $n = 10$.

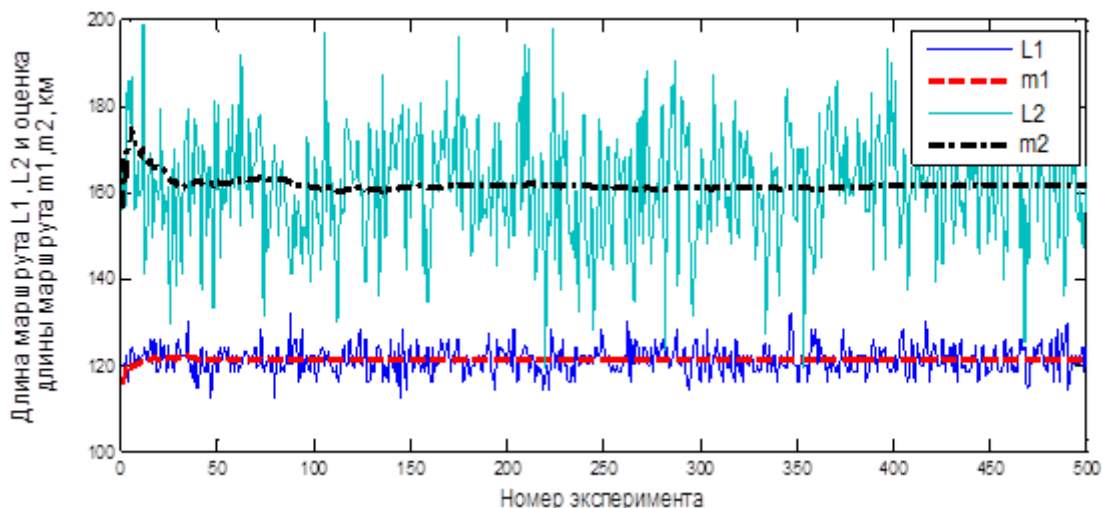


Рисунок 3.4 - Рекурсивна оцінка довжини маршруту мобільного стоку при рівномірному законі розподілу відстаней між вузлами, $n = 30$.

Для порівняння були обрані два евристичних алгоритма: алгоритм імітації випалювання і мурашиний алгоритм. Для застосування алгоритмів до вирішення завдання комівояжера необхідно задати координати пунктів.

Координати для 10, 20, 50 пунктів наведені в таблицях 3.2-3.4 відповідно.

Таблиця 3.2 - Координати для 10 пунктів

№ Пункта	X-координата	Y-координата
1	64.5560	90.1104
2	20.9221	27.7721
3	51.6975	65.7792
4	22.7203	23.2930
5	89.8661	64.8356
6	88.7157	19.3623
7	3.6055	93.7693
8	58.6599	69.9310
9	48.9370	83.2530
10	0.9711	10.5037

Таблиця 3.3 - Координати для 20 пунктів

№ Пункта	Х-координата	У-координата
1	52.1597	95.5601
2	1.9196	63.6311
3	6.3565	46.5866
4	3.0746	66.0993
5	18.1014	43.9670
6	37.2821	5.0035
7	16.6668	5.8136
8	69.2888	96.3496
9	51.8321	36.9488
10	70.3050	11.1837
11	14.1556	23.9552
12	42.5330	58.8128
13	55.1569	95.4764
14	2.0953	34.0074
15	4.5228	4.2153
16	12.7726	5.2241
17	26.4810	98.6043
18	81.6522	20.8680
19	73.5520	24.5911
20	5.8865	29.1832

Таблиця 3.4 - координати для 50 пунктівів

№ Пункта	Х-координата	У-координата
1	2	3
1	64.3143	1.0335
2	98.3094	88.1154
3	78.8417	29.4470
4	3.7566	4.3246
5	71.7370	46.3549
6	87.5993	10.8558

Продовження таблиці 3.4

1	2	3
7	28.6618	16.4648
8	76.4360	33.6406
9	24.8803	71.4431
10	90.8005	84.8320
11	82.9661	45.2861
12	67.3041	21.8704
13	51.4457	8.6148
14	11.7521	64.6400
15	6.8502	34.0586
16	57.8447	17.0838
17	53.9740	94.5100
18	13.5799	46.6322
19	75.6728	29.0275
20	58.7067	2.7511
21	12.9558	45.1491
22	11.8016	62.3956
23	5.6853	20.9846
24	10.4870	19.8081
25	83.3784	25.4139
26	36.1366	3.2274
27	53.0115	16.7289
28	81.3022	34.2655
29	81.1341	37.6206
30	2.9145	70.6759
31	4.7527	88.0327
32	56.8160	32.7086
33	88.6513	50.1101
34	8.3908	82.9287
35	81.9838	59.4841
36	95.1465	32.9118
37	84.6034	50.1301

Продовження таблиці 3.4

1	2	3
38	40.0401	98.1786
39	59.4615	81.2310
40	46.5944	90.4136
41	47.4861	43.5111
42	18.5612	22.4940
43	45.9927	89.5787
44	76.7160	39.1231
45	11.8319	52.0155
46	17.4391	6.6859
47	41.6861	23.4696
48	61.4724	40.8551
49	20.8783	27.4067
50	54.2831	45.4826

Розташування пунктів наведені на рисунках 3.5 - 3.7 для 10, 20, 50 пунктів відповідно.

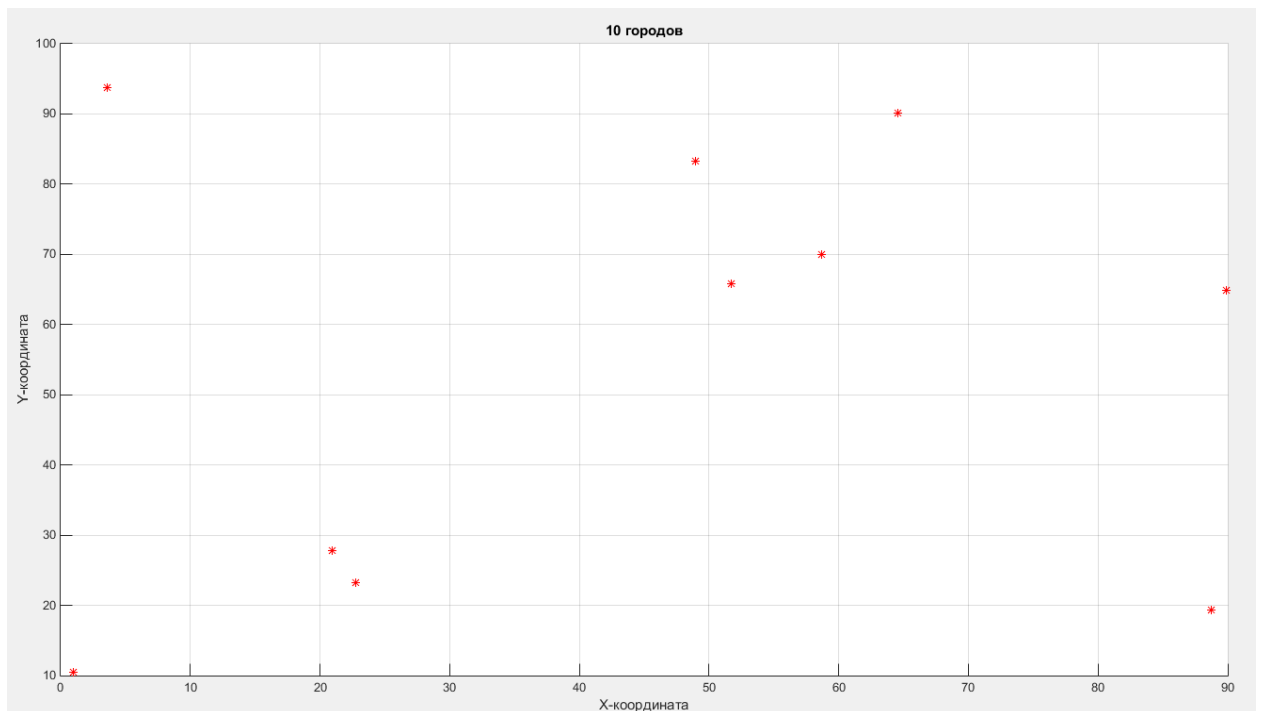


Рисунок 3.5 - Розташування 10 пунктів

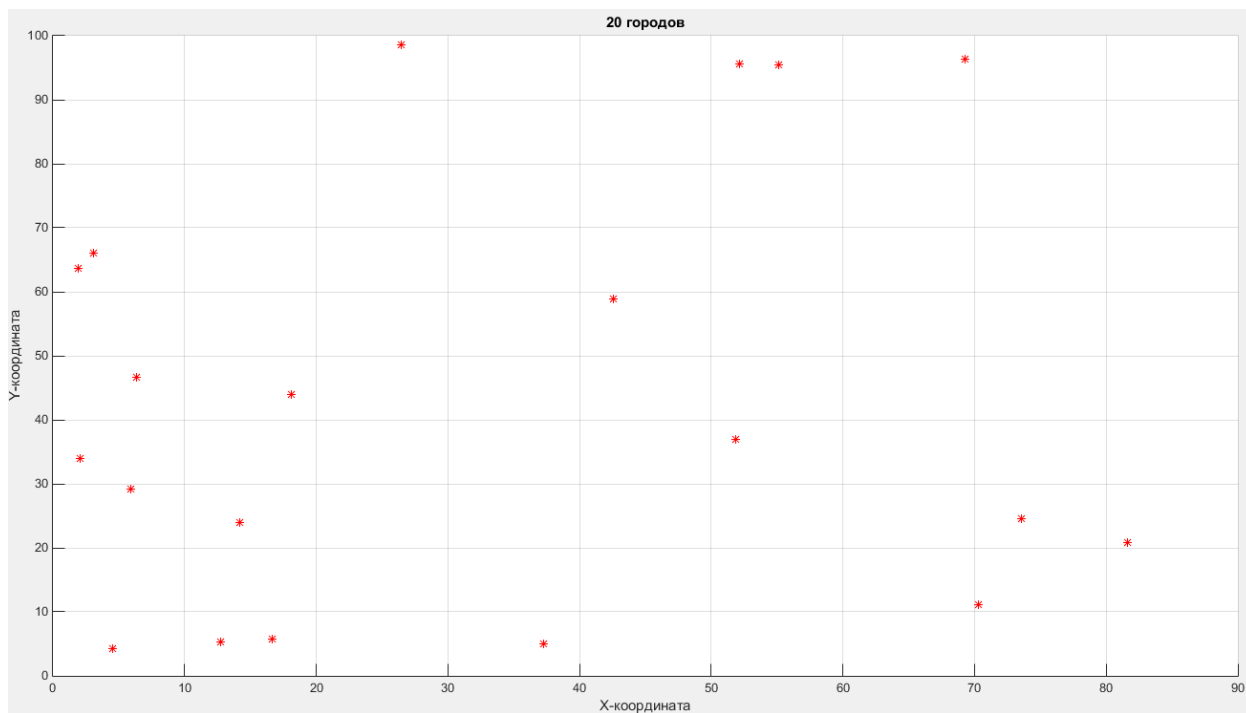


Рисунок 3.6 - Розташування 20 пунктів

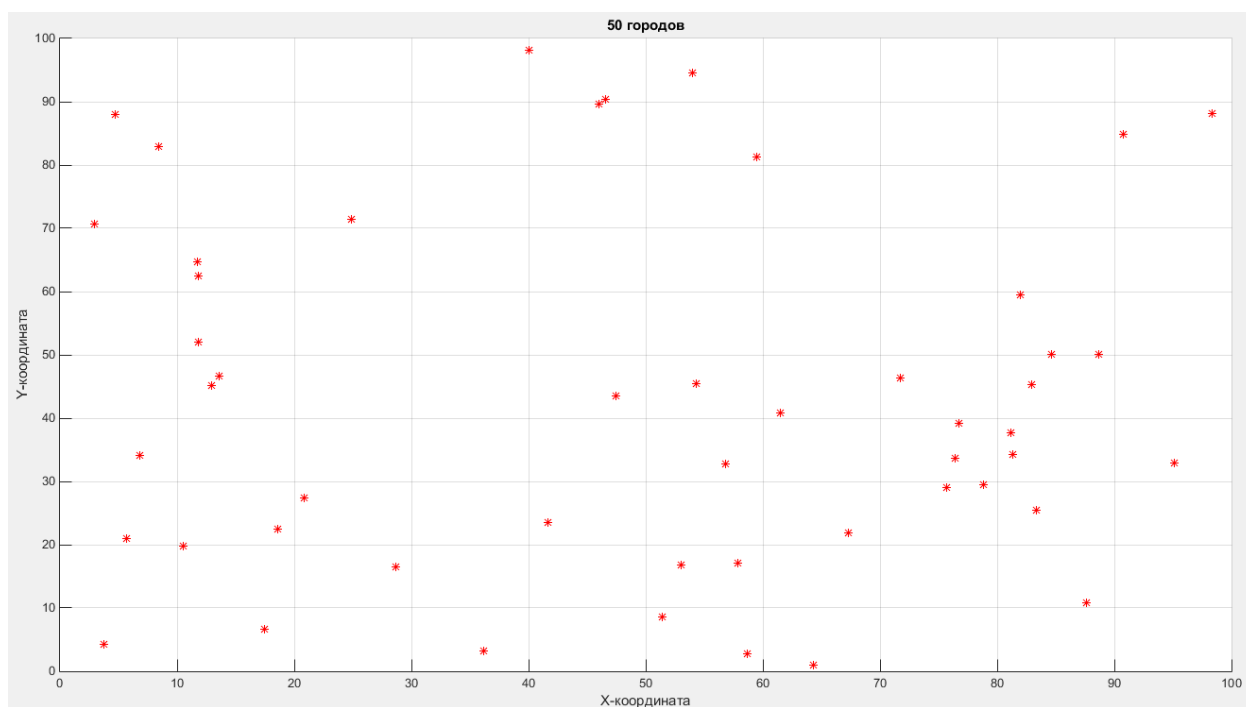


Рисунок 3.7 - Розташування 50 пунктів

Розташовані пункти на координатній осі, але може бути і на карті від цього рішення завдання не зміниться, так як шлях мобільного стоку не залежить від місцевості.

3.2 Визначення оптимального значення кількості ітерацій для розв'язання задачі

Так як кількість ітерацій впливає на час розрахунків і обидва алгоритми є евристичними, кількість ітерацій задається спочатку перед розрахунками, то потрібно визначити оптимальні значення кількості ітерацій для кожного завдання і алгоритму окремо.

Для визначення мінімального шляху проведемо розрахунки за допомогою методу гілок і меж і порівняємо результат з результатами кожного алгоритму при різних кількостях ітерацій, знайдемо ділянки в яких виходить мінімальне відхилення від оптимального рішення.

3.2.1 Визначення оптимального значення кількості ітерацій для мурашиний алгоритм

Для 10 пунктів візьмемо кількість ітерацій в діапазоні від 1 до 50 з кроком 1.

Для 20 пунктів візьмемо кількість ітерацій в діапазоні від 10 до 500 з кроком 10.

Для 50 пунктів візьмемо кількість ітерацій в діапазоні від 100 до 5000 з кроком 100.

На рисунках 3.8-3.10 приведені отримані графіки для 10, 20 і 50 пунктів відповідно.

Як видно з графіків то для 20 пунктів вибрати оптимальну кількість ітерацій не складно так як після 50 ітерацій отримуюмо однакові значення які відповідають мінімальному шляху. Для 10 і 50 пунктів все не так однозначно, так як на графіках добре видно евристику алгоритму, адже навіть при великих значеннях ітерацій не отримуюмо мінімальний шлях і тому кількість ітерацій для них доведеться вибирати аналітично.

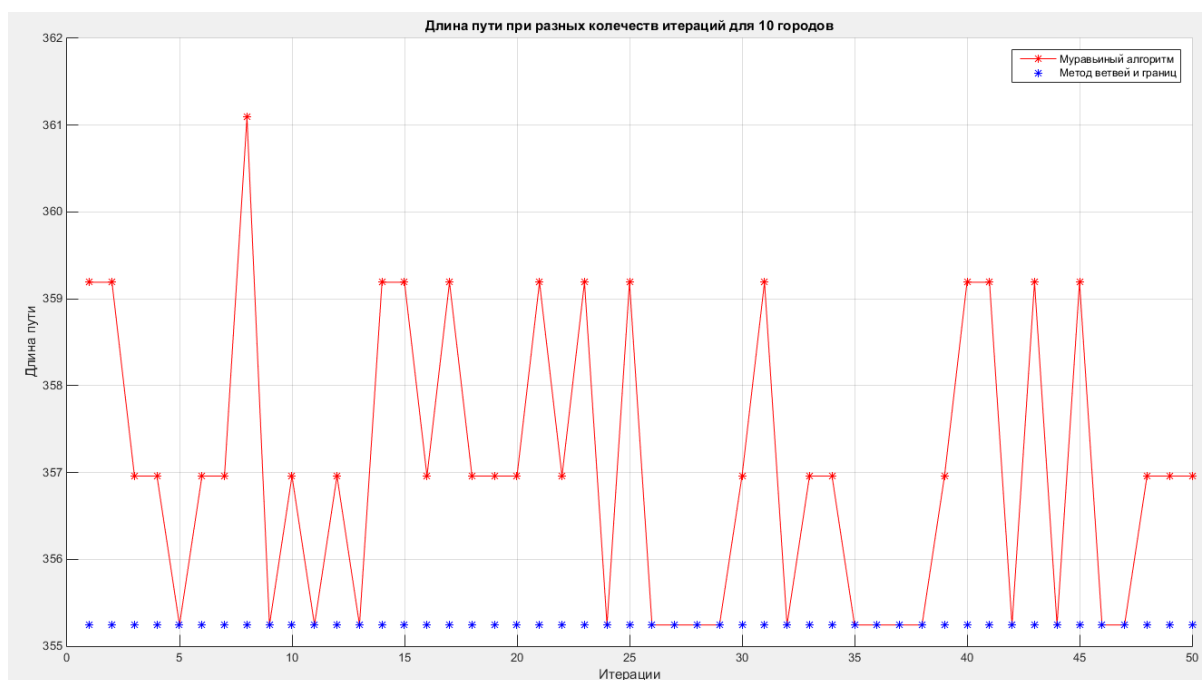


Рисунок 3.8 - Довжина шляху при різних значеннях ітерації для 10 пунктів

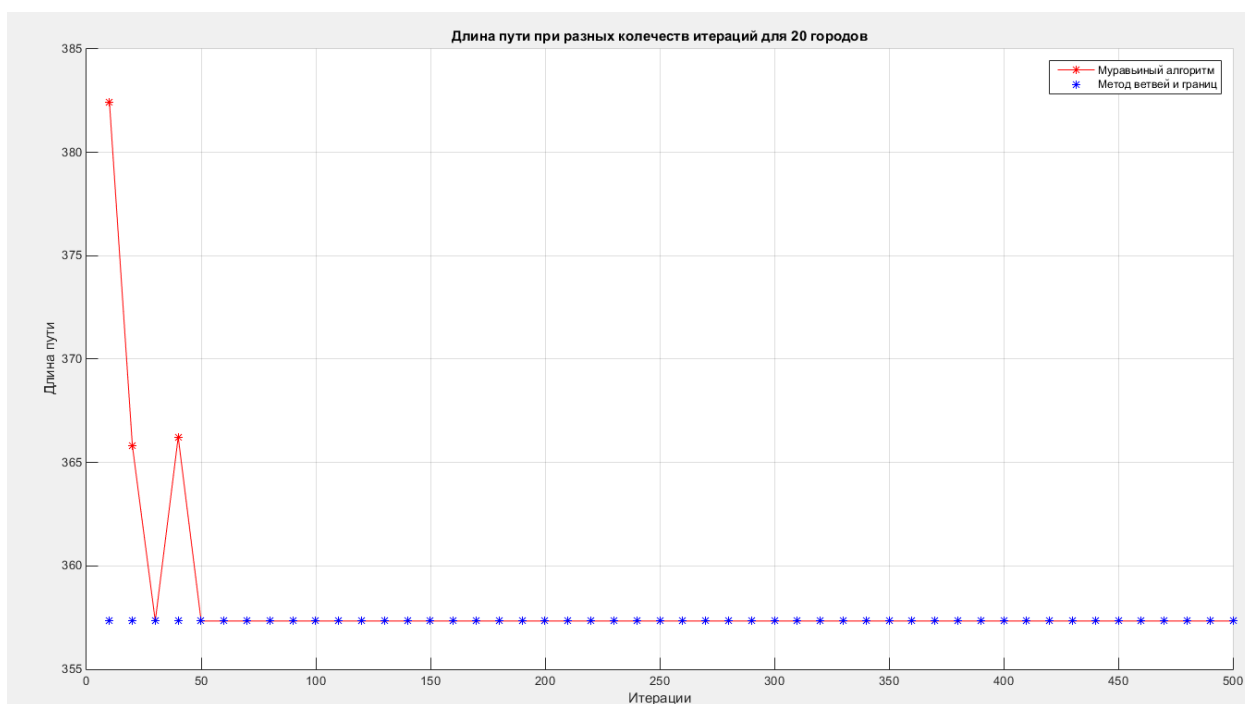


Рисунок 3.9 - Довжина шляху при різних значеннях ітерації для 20 пунктів

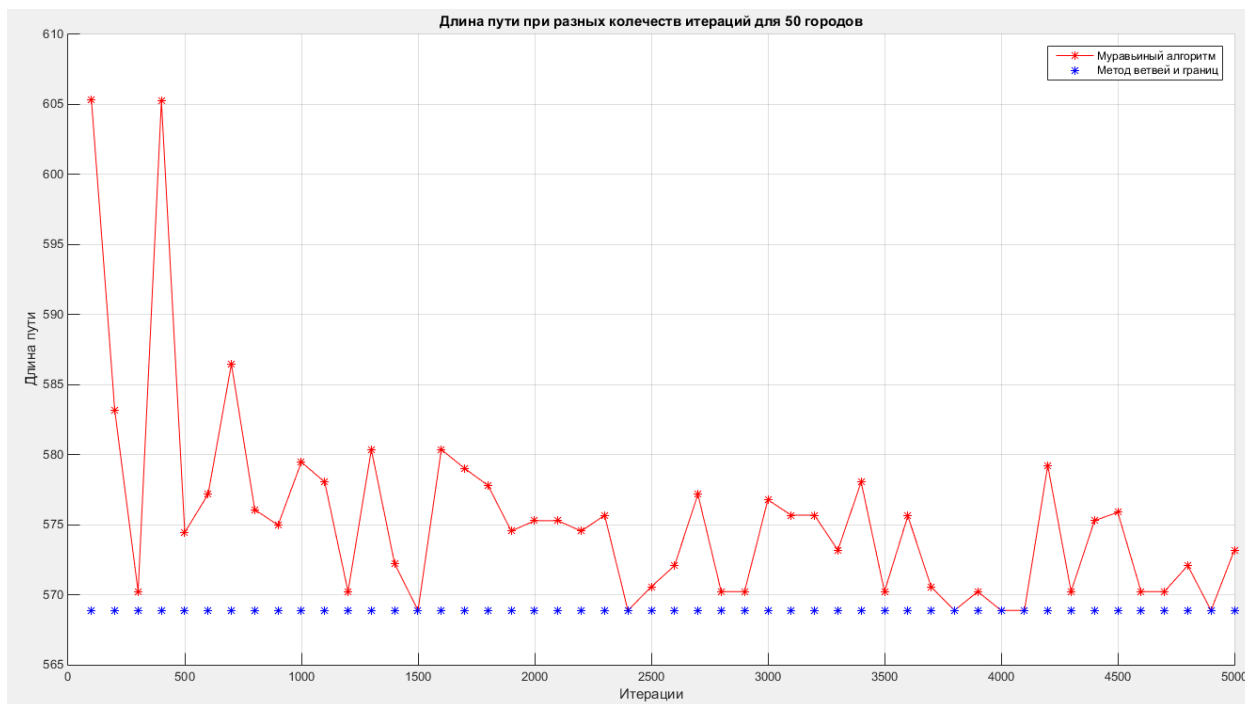


Рисунок 3.10 - Довжина шляху при різних значеннях ітерації для 50 пунктів

Для 10 пунктів виберемо 26 ітерацій, так як на ділянці 25-28 ітерацій отримуємо мінімальний шлях і це найменша кількість ітерацій коли отримуємо таке значення.

Для 20 пунктів вибираем 55 ітерацій, адже після 50 отримуємо мінімальний шлях.

Для 50 пунктів вибрали 3900 ітерацій, хоч і на графіку бачимо, що при такій кількості ітерацій у нас не мінімальне значення шляху, але на ділянці в 3800-4100 ітерацій отримуємо мінімальні значення довжини шляху.

3.2.2 Алгоритм імітації випалювання

Алгоритм імітації випалювання спочатку вимагає великої кількості ітерацій для своєї роботи і тому не можливо порівняти роботу мурашиного алгоритму і алгоритму імітації випалювання на однакових діапазонах ітерацій.

Проведемо аналіз роботи алгоритму на ділянках від 1000 до 50000 з кроком 1000 ітерацій для 10 пунктів, для 20 пунктів від 10000 до 500000 з кроком 10000, а для 50 пунктів від 1000000 до 50000000 з кроком 1000000.

На рисунках 3.11-3.13 приведені отримані графіки для 10, 20 і 50 пунктів відповідно.

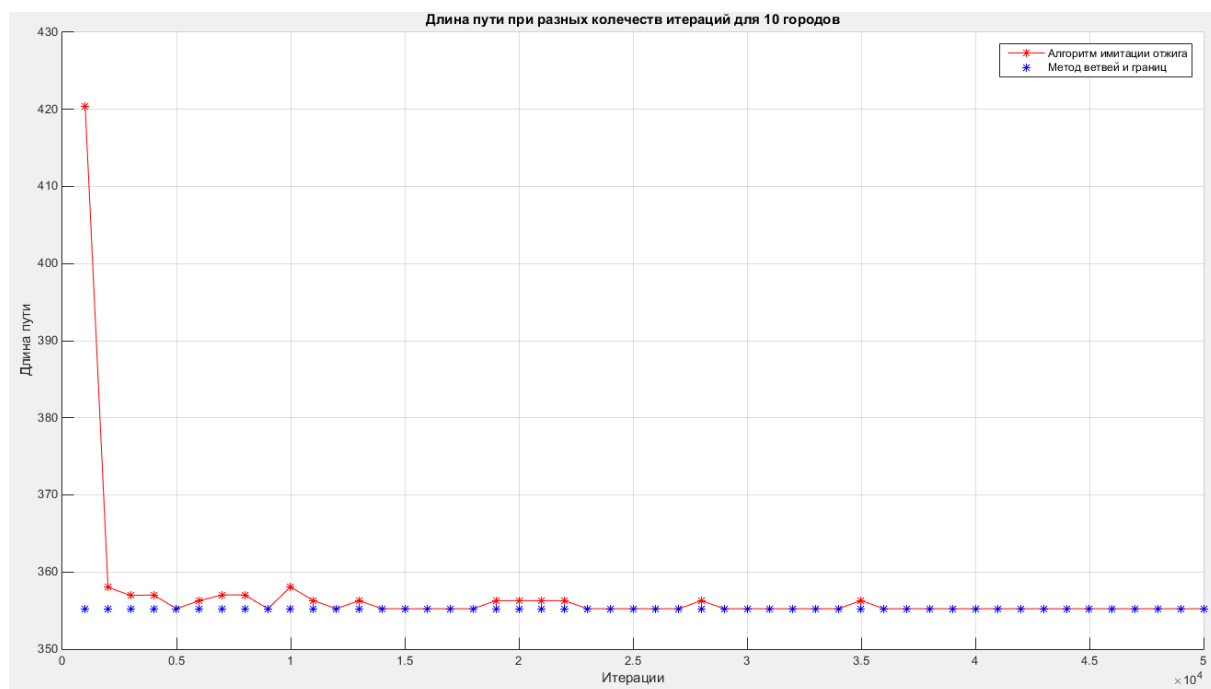


Рисунок 3.11 - Довжина шляху при різних значеннях ітерації для 10 пунктів

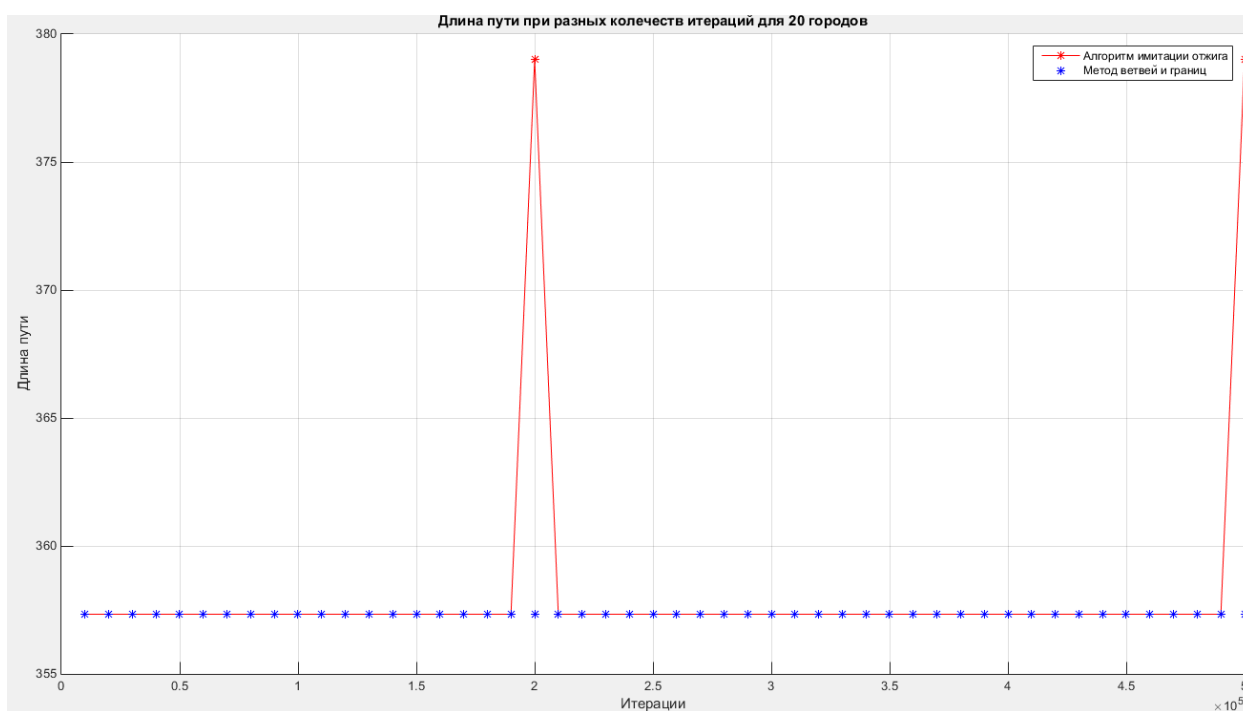


Рисунок 3.12 - Довжина шляху при різних значеннях ітерації для 20 пунктів

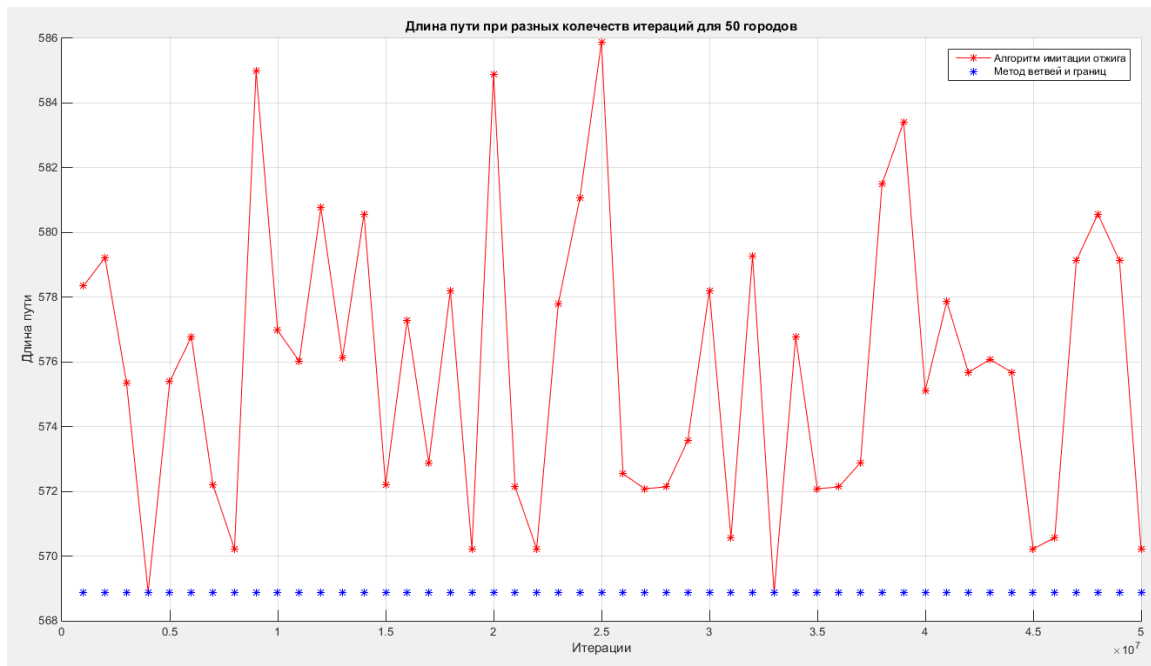


Рисунок 3.13 - Довжина шляху при різних значеннях ітерації для 50 пунктів

Як видно з графіків то вибрати кількість ітерацій для 10 і 20 пунктів не складно, так як отримали практично на всьому діапазоні вибраних ітерацій мінімальний шлях за одиничним винятками, а ось з вибором оптимальної кількості ітерацій для 50 пунктів вже складніше так як мінімальний шлях отримуємо тільки при двох значень ітерацій, саме при 4000000 і при 33000000 ітерацій, а при сусідніх значеннях отримуємо значне відхилення від мінімального шляху.

Для 10 пунктів виберемо 16000 ітерацій, так як на ділянці 1000-15000 ітерацій отримуємо невеликі відхилення від мінімального шляху.

Для 20 пунктів виберемо 250000 ітерацій, як середнє значення всього діапазону.

Для 50 пунктів виберемо 27000000 ітерацій, хоч і на графіку бачимо, що при такій кількості ітерацій у нас не мінімальне значення шлях але на ділянці в 26000000-29000000 приходило більш плавний графік без різких стрибків.

3.3 Порівняльний аналіз алгоритмів для задачі оптимізації шляху

Порівняння алгоритмів проведемо за двома пунктами: час розрахунку і довжина шляху.

Для більш точних результатів для кожного алгоритма проведемо по 20 вимірювань, розрахуємо відхилення отриманих довжин шляху від оптимального шляху отриманого за допомогою методу гілок і меж.

3.3.1 Результати моделювання для 10 пунктів

Мінімальний шлях проходить через вершини $5 \rightarrow 1 \rightarrow 8 \rightarrow 3 \rightarrow 9 \rightarrow 7 \rightarrow 10 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 5$ і має довжину 355.2446 у.о.. Не має значення яка вершина буде начальною так як у нас замкнутий контур і з якої вершини не почати але пройти через вершини в даній послідовності все одно пройдемо мінімальним шляхом.

На рисунку 3.14 зображений мінімальний шлях для 10 пунктів.

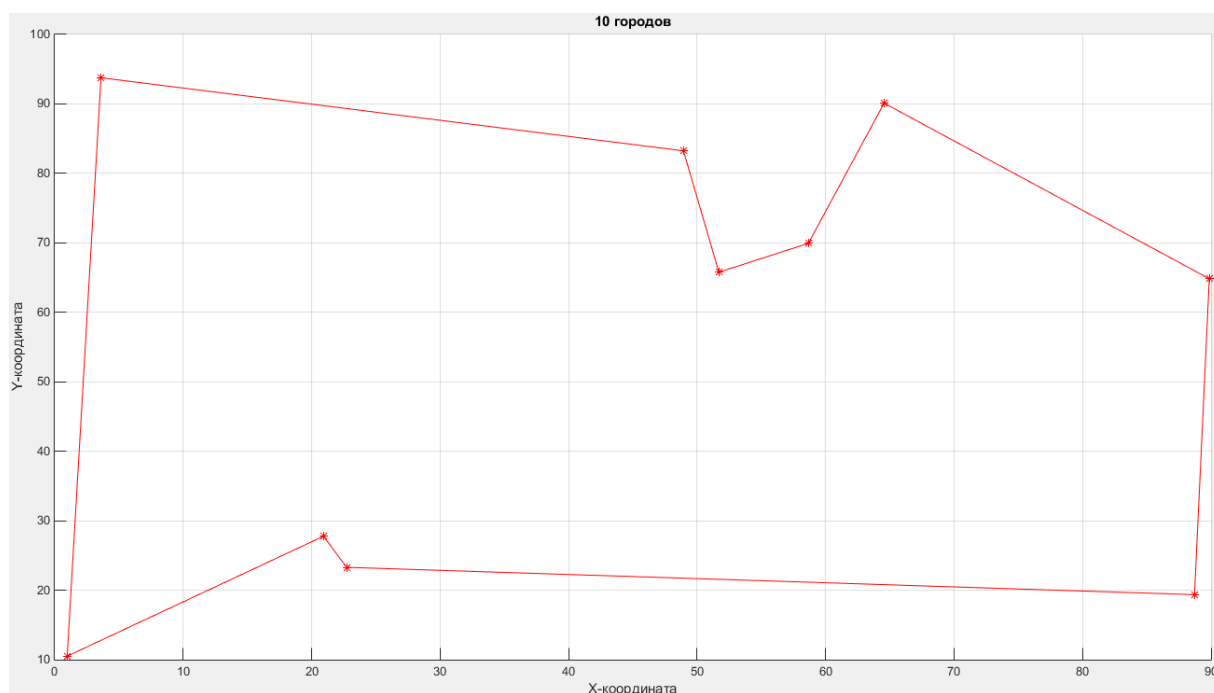


Рисунок 3.14 - Мінімальний шлях для 10 пунктів

Параметри для мурашиного алгоритму:

- кількість ітерацій (поколінь) - 26;
- кількість мурах в поколінні - 10;
- кількість пунктів - 10;
- альфа (коєф. орієнтації на феромони) - 1;
- бета (коєф. орієнтації на довжину шляху) - 2;
- р (коєф. поновлення феромонів локальне) - 0,1;

- e (коєф. поновлення феромонів глобальне) - 0,1;
- q (коєф. вибору найпривабливішого пунктіва) - 0,9;
- початкове розташування мурах - випадковий;
- мінімальній шлях 355.2446 у.о..

У таблиці 3.6 наведені результати розрахунків для мурашиного алгоритму.

Таблиця 3.6 - Результати розрахунків для мурашиного алгоритму.

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	359.1907	0.809092	1.1108	356.9009	0.2102	0.4662
2	356.9595	0.339886	0.4827			
3	355.2446	0.113958	0			
4	357.0235	0.198742	0.5008			
5	356.9595	0.128243	0.4827			
6	356.9595	0.110399	0.4827			
7	355.2446	0.148615	0			
8	355.2446	0.180334	0			
9	359.1907	0.182238	1.1108			
10	357.0235	0.195793	0.5008			
11	359.1907	0.170344	1.1108			
12	356.9595	0.200888	0.4827			
13	355.2446	0.138731	0			
14	357.0235	0.181274	0.5008			
15	356.9595	0.159103	0.4827			
16	355.2446	0.180342	0			
17	359.1907	0.180479	1.1108			
18	356.9595	0.183611	0.4827			
19	356.9595	0.208028	0.4827			
20	355.2446	0.194215	0			

Як видно з таблиці отримуємо в середньому помилку в 0.4662% від мінімального шляху при цьому середній час розрахунку становить 0.2102 секунд.

Параметри для алгоритма імітації випалювання:

- кількість пунктів - 10;
- початкова температура - 1000;
- кінцева температура - 0,1;
- формула температури - початкову температуру / k-ую ітерацію;
- число ітерацій - 16000;
- функція ймовірності прийняття - $\exp(-\Delta E / T)$;
- визначення потенційного маршруту (що породжує сімейства) - розворот частини вектора (поточного маршруту) від двох випадково обраних чисел рівномірним розподілом;
- мінімальній шлях 355.2446 у.о..

У таблиці 3.7 результати розрахунків для алгоритма імітації випалювання

Таблиця 3.7 - Результати розрахунків для алгоритма імітації випалювання

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	2	3	4	5	6	7
1	355.2446	0.751930	0	355.2446	0.5595	0
2	355.2446	0.748708	0			
3	355.2446	0.741665	0			
4	355.2446	0.593019	0			
5	355.2446	0.518144	0			
6	355.2446	0.488536	0			
7	355.2446	0.564102	0			
8	355.2446	0.509357	0			
9	355.2446	0.410670	0			
10	355.2446	0.534892	0			
11	355.2446	0.469080	0			
12	355.2446	0.569686	0			
13	355.2446	0.576537	0			
14	355.2446	0.472867	0			

Продовження таблиці 3.7

1	2	3	4	5	6	7
15	355.2446	0.613468	0	355.2446	0.5595	0
16	355.2446	0.494844	0			
17	355.2446	0.488830	0			
18	355.2446	0.548207	0			
19	355.2446	0.513769	0			
20	355.2446	0.582325	0			

Як видно з таблиці отримали 100% результат, всі виміри показали нам мінімальний шлях при середньому значенні часу 0.5595 секунд.

Алгоритм імітації випалювання показав себе краще для 10 пунктів ніж мурашиний алгоритм, але варто зауважити, що на розрахунки алгоритму випалювання пішло в 2.6618 часу більше ніж на мурашиний, але варто збільшити кількість мурах в поколінні до 20 так відразу буде видно, що отримаємо теж 100 % результат, а час залишаться майже незмінними.

Результати розрахунків для мурашиного алгоритму з кількістю мурах в поколінні рівним 20 наведені в таблиці 3.8.

Таблиця 3.8 - Результати розрахунків для мурашиного алгоритму з кількістю мурах в поколінні рівним 20

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	355.2446	0.199739	0	355.2446	0.2169	0
2	355.2446	0.245082	0			
3	355.2446	0.205790	0			

Як видно з таблиці то навіть різниця часу між 10 і 20 мурахами в поколінні для мурашиного алгоритму мінімальна, а результат отримуємо 100% при тому, що різниця під час виконання двох алгоритмів відносно велика.

3.3.2 Результати моделювання для 20 пунктів

Мінімальний шлях проходить через вершини 1→13→8→12→9→19→18→10→6→7→16→15→11→20→14→5→3→2→4→17→1 і має довжину 357.3328 у.о.. Не має значення яка вершина буде початковою так як у нас замкнутий контур і з якої вершини не почати але пройти через вершини в даній послідовності все одно пройдемо мінімальним шляхом.

На рисунку 3.15 зображений мінімальний шлях для 20 пунктів.

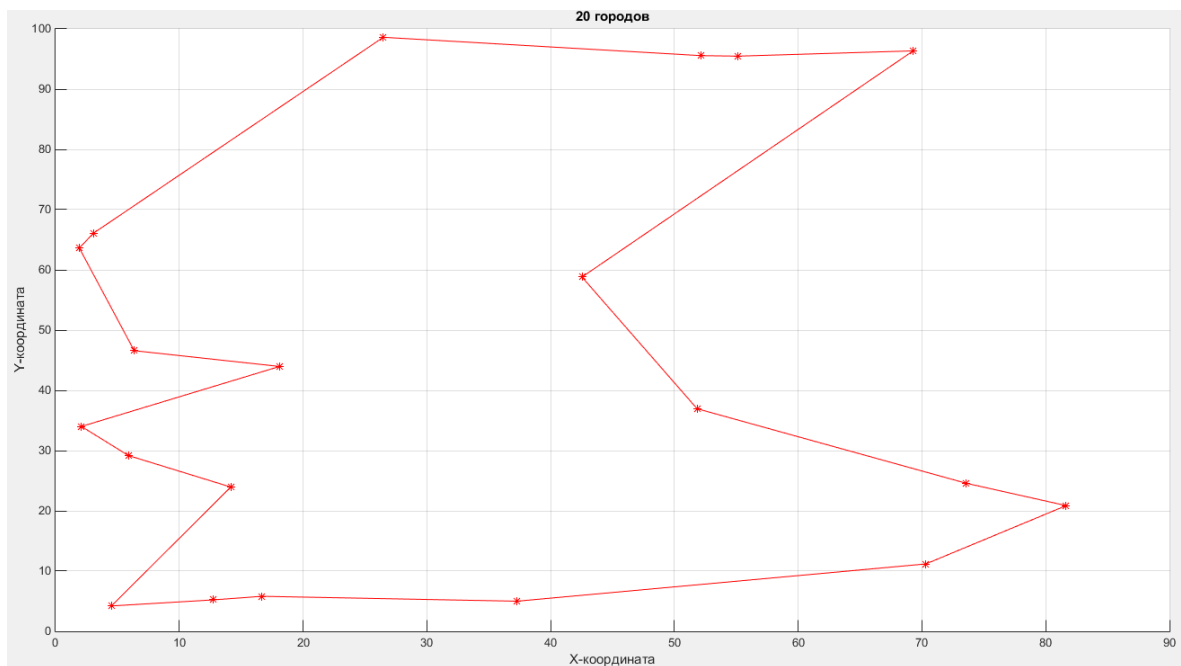


Рисунок 3.15 - Мінімальний шлях для 20 пунктів

Параметри для мурашиного алгоритму:

- кількість ітерацій (покоління) - 55;
- кількість мурах в поколінні - 10;
- кількість пунктів - 20;
- альфа (коэф. орієнтації на феромони) - 1;
- бета (коэф. орієнтації на довжину шляху) - 2;
- р (коэф. поновлення феромонів локальне) - 0,1;
- е (коэф. поновлення феромонів глобальне) - 0,1;
- q (коэф. вибору найпривабливішого пунктіва) - 0,9;
- початкове розташування мурах - випадковий;
- мінімальній шлях 357.3328 у.о..

У таблиці 3.9 наведені результати розрахунків для мурашиного алгоритму.

Таблиця 3.9 - Результати розрахунків для мурашиного алгоритму

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	357.3328	0.271445	0	357.4470	0.3112	0.0319
2	357.3328	0.283423	0			
3	357.3328	0.297148	0			
4	357.3328	0.304530	0			
5	357.3328	0.327589	0			
6	357.3328	0.277917	0			
7	357.3328	0.305225	0			
8	357.3328	0.350609	0			
9	357.3328	0.327719	0			
10	357.3328	0.285887	0			
11	359.6161	0.320670	0.6390			
12	357.3328	0.274376	0			
13	357.3328	0.318673	0			
14	357.3328	0.412933	0			
15	357.3328	0.315659	0			
16	357.3328	0.306380	0			
17	357.3328	0.322244	0			
18	357.3328	0.315672	0			
19	357.3328	0.306156	0			
20	357.3328	0.300492	0			

Як видно з таблиці отримали практично 100% результат, тільки одне вимірювання не дало нам мінімальний шлях при цьому середній час розрахунку становить 0.3112 секунди.

Параметри для алгоритма імітації випалювання:

- кількість пунктів - 20;
- початкова температура - 1000;
- кінцева температура - 0,1;
- формула температури - початкову температуру / к-ую ітерацію;

- число ітерацій - 250000;
- функція ймовірності прийняття - $\exp(-de / t)$;
- визначення потенційного маршруту (що породжує сімейства) - розворот частини вектора (поточного маршруту) від двох випадково обраних чисел рівномірним розподілом;
- мінімальній шлях 357.3328 у.о..

У таблиці 3.10 результати розрахунків для алгоритма імітації випалювання

Таблиця 3.10 - Результати розрахунків для алгоритма імітації випалювання

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	2	3	4	5	6	7
1	357.3328	0.437119	0	367.5737	0.5230	2.8659
2	357.3328	0.588930	0			
3	379.0111	0.500612	6.0667			
4	357.3328	0.550232	0			
5	398.0664	0.549847	11.3993			
6	357.3328	0.523015	0			
7	357.3328	0.538513	0			
8	357.3328	0.568568	0			
9	357.3328	0.499402	0			
10	357.3328	0.550680	0			
11	415.6489	0.514654	16.3198			
12	357.3328	0.472661	0			
13	357.3328	0.455975	0			
14	398.0664	0.482385	11.3993			
15	357.3328	0.516543	0			
16	379.0111	0.531055	6.0667			
17	379.0111	0.593576	6.0667			
18	357.3328	0.489827	0			
19	357.3328	0.507194	0			

Продовження таблиці 3.10

1	2	3	4	5	6	7
20	357.3328	0.588362	0	367.5737	0.5230	2.8659

Як видно з розрахунків то алгоритм імітації випалювання показав середній шлях 367.5737 у.о. за 0.523 секунд.

Для 20 пунктів мурашиний алгоритм показав себе краще ніж алгоритм імітації випалювання, як і в показниках середнього шляху так і часу розрахунку. Можна для алгоритма імітації випалювання зменшити кількість ітерацій або початкову температуру, що в свою чергу зменшити час самого розрахунку, але в цей же час і зменшити точність самих розрахунків.

3.3.3 Результати моделювання для 50 пунктів

Мінімальний шлях проходить через вершини 50→41→47→7→26→46→4→23→24→42→49→15→21→18→45→22→14→30→31→34→9→38→43→40→17→39→10→2→35→33→37→11→5→44→29→28→8→19→3→25→36→6→1→20→13→27→16→12→32→48→50 і має довжину 568.8768 у.о.. Не має значення яка вершина буде початковою так як у нас замкнутий контур і з якої вершини не почати але пройти через вершини в даній послідовності все одно пройдемо мінімальним шляхом.

На рисунку 3.16 зображений мінімальний шлях для 50 пунктів.

Параметри для мурашиного алгоритму:

- кількість ітерацій (покоління) - 3900;
- кількість мурах в поколінні - 10;
- кількість пунктів - 50;
- альфа (коєф. орієнтації на феромони) - 1;
- бета (коєф. орієнтації на довжину шляху) - 2;
- р (коєф. поновлення феромонів локальне) - 0,1;
- е (коєф. поновлення феромонів глобальне) - 0,1;
- q (коєф. вибору найпривабливішого пунктіва) - 0,9;
- початкове розташування мурах - випадковий;
- мінімальній шлях 568.8768 у.о..

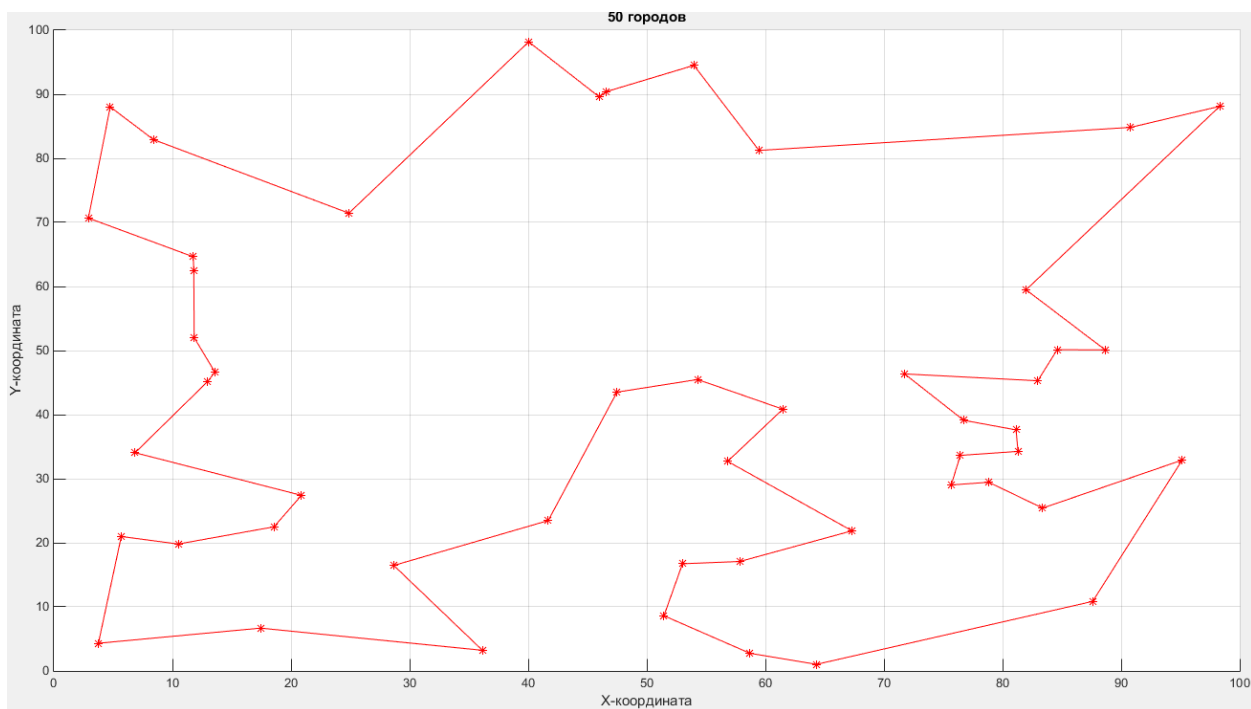


Рисунок 3.16 - Мінімальний шлях для 50 пунктів

У таблиці 3.11 наведені результати розрахунків для мурашиного алгоритму.

Таблиця 3.11 - Результати розрахунків для мурашиного алгоритму

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	2	3	4	5	6	7
1	568.8768	28.420921	0	571.8715	27.1727	0.5264
2	574.5416	27.217633	0.9958			
3	568.8768	28.092571	0			
4	568.8768	27.572295	0			
5	572.5433	26.991461	0.6445			
6	573.9818	26.697709	0.8974			
7	575.6753	26.938165	1.1951			
8	579.2713	26.963611	1.8272			
9	572.0819	27.040020	0.5634			
10	573.1822	27.043510	0.7568			
11	568.8768	27.968893	0			

Продовження таблиці 3.11

1	2	3	4	5	6	7
12	572.0819	26.918153	0.5634	571.8715	27.1727	0.5264
13	568.8768	27.295156	0			
14	573.1822	27.340024	0.7568			
15	575.889	26.679792	1.2326			
16	568.8768	27.323449	0			
17	575.1074	26.686787	1.0952			
18	568.8768	26.643718	0			
19	568.8768	26.933550	0			
20	568.8768	26.686680	0			

Як видно з таблиці мурашиний алгоритм показує відмінні результати для 50 пунктів при середньому часу 27.1727 секунди середня помилка складає всього лише 0.5264%.

Параметри для алгоритма імітації випалювання:

- кількість пунктів - 50;
- початкова температура - 100000;
- кінцева температура - 0,1;
- формула температури - початкову температуру / к-ую ітерацію;
- число ітерацій - 250000;
- функція ймовірності прийняття - $\exp(-\delta e / t)$;
- визначення потенційного маршруту (що породжує сімейства) - розворот частини вектора (поточного маршруту) від двох випадково обраних чисел рівномірним розподілом;
- мінімальній шлях 568.8768 у.о..

У таблиці 3.12 наведені результати розрахунків для алгоритма імітації випалювання

Як видно з таблиці алгоритм імітації випалювання показав непоганий результат, при середньому часу роботи в 42.6522 секунди видав помилку раною в 0.6689%.

Таблиця 3.12 - Результати розрахунків для алгоритма імітації випалювання

№	Довжина шляху (у.о.)	Час (секунд)	Помилка (%)	Середня довжина шляху (у.о.)	Середній час (секунд)	Середня помилка (%)
1	572.4875	39.521770	0.6347	572.6818	42.6522	0.6689
2	583.7623	48.814057	2.6167			
3	572.5433	43.199117	0.6445			
4	572.1377	47.953066	0.5732			
5	570.2242	43.352373	0.2369			
6	575.6275	47.239737	1.1867			
7	574.5416	48.144293	0.9958			
8	568.8768	52.656999	0			
9	572.8328	42.287291	0.6954			
10	570.7624	39.801882	0.3315			
11	568.8768	40.103212	0			
12	573.1822	38.751265	0.7568			
13	579.6166	40.197158	1.8879			
14	568.8768	39.771060	0			
15	575.3939	39.393028	1.1456			
16	572.0819	42.831310	0.5634	572.6818	42.6522	0.6689
17	572.4875	40.282783	0.6347			
18	568.8768	39.447056	0			
19	570.2242	40.389165	0.2369			
20	570.2242	38.907399	0.2369			

Проведемо ще по кілька вимірів для 100 і 500 пунктів.

На рисунку 3.17 зображено розташування 100 пунктів.

На рисунку 3.18 зображено розташування 500 пунктів.

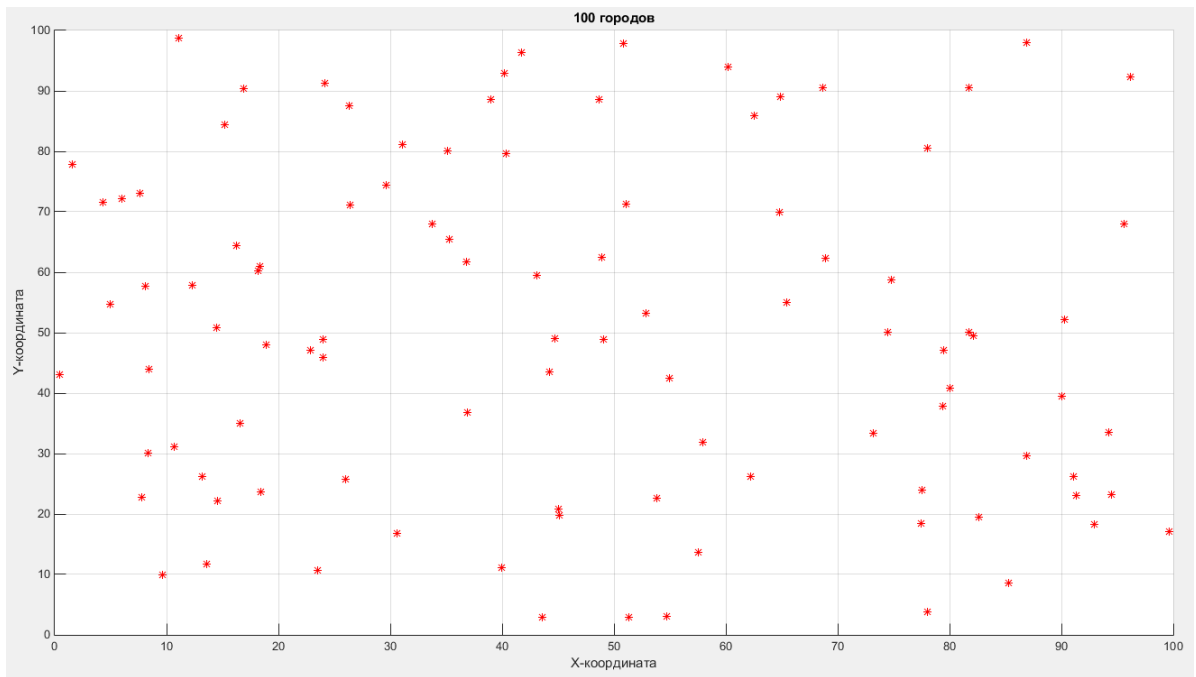


Рисунок 3.17 - Розташування 100 пунктів

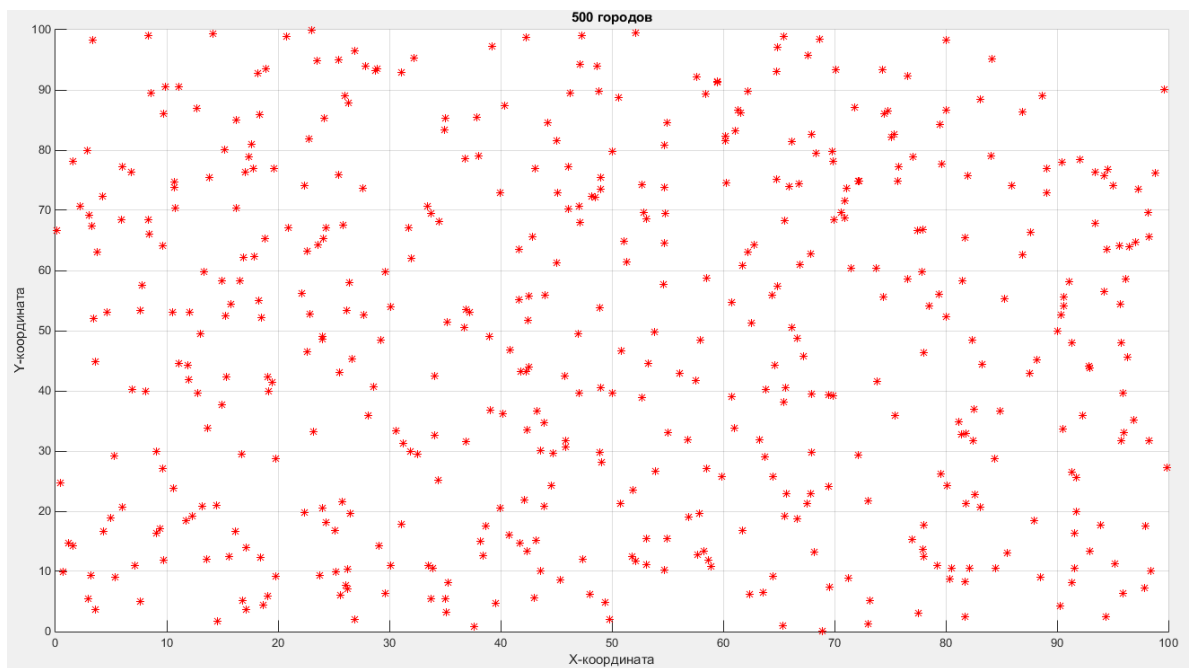


Рисунок 3.18 - Розташування 500 пунктів

Параметри для мурашиного алгоритму:

- кількість ітерацій (покоління) - 9000/20000;
- кількість мурах в поколінні - 10;
- кількість пунктів - 100/500;
- альфа (коєф. орієнтації на феромони) - 1;
- бета (коєф. орієнтації на довжину шляху) - 2;

- р (коэф. поновлення феромонів локальне) - 0,1;
- е (коэф. поновлення феромонів глобальне) - 0,1;
- q (коэф. вибору найпривабливішого пунктіва) - 0,9;
- початкове розташування мурах - випадковий.

Параметри для алгоритма імітації випалювання:

- кількість пунктів - 100/500;
- початкова температура - 100000;
- кінцева температура - 0,1;
- формула температури - початкову температуру / k-ую ітерацію;
- число ітерацій - 1000000/10000000;
- функція ймовірності прийняття - $\exp(-\delta e / t)$.
- визначення потенційного маршруту (що породжує сімейства) - розворот частини вектора (поточного маршруту) від двох випадково обраних чисел рівномірним розподілом;

У таблиці 3.11 наведені результати розрахунків.

Таблиця 3.13 - Результати розрахунків для 100 і 500 пунктів.

	Мурашиний алгоритм		Алгоритм імітації випалювання	
№	Довжина шляху (у.о.)	Час (секунд)	Довжина шляху (у.о.)	Час (секунд)
100 пунктів				
1	821.4988	189.513869	817.5347	42.239714
2	816.6737	186.088802	801.4941	43.638596
3	799.4715	140.772230	800.1126	48.471429
500 пунктів				
1	-	-	1836.4	79.681766
2	-	-	1843.1	83.506554

З таблиці видно, що алгоритм імітації випалювання при 100 пунктів працює вразі швидше і видає середній шлях рівним 806.3805 у.о., що менше ніж показав нам мурашиний алгоритм.

Кращий шлях мурашиного алгоритму для 100 пунктів, показаний на рисунку 3.19, для алгоритма імітації випалювання на рисунку 3.20.

Кращий шлях алгоритму імітації випалювання для 500 пунктів зображений на рисунку 3.21.

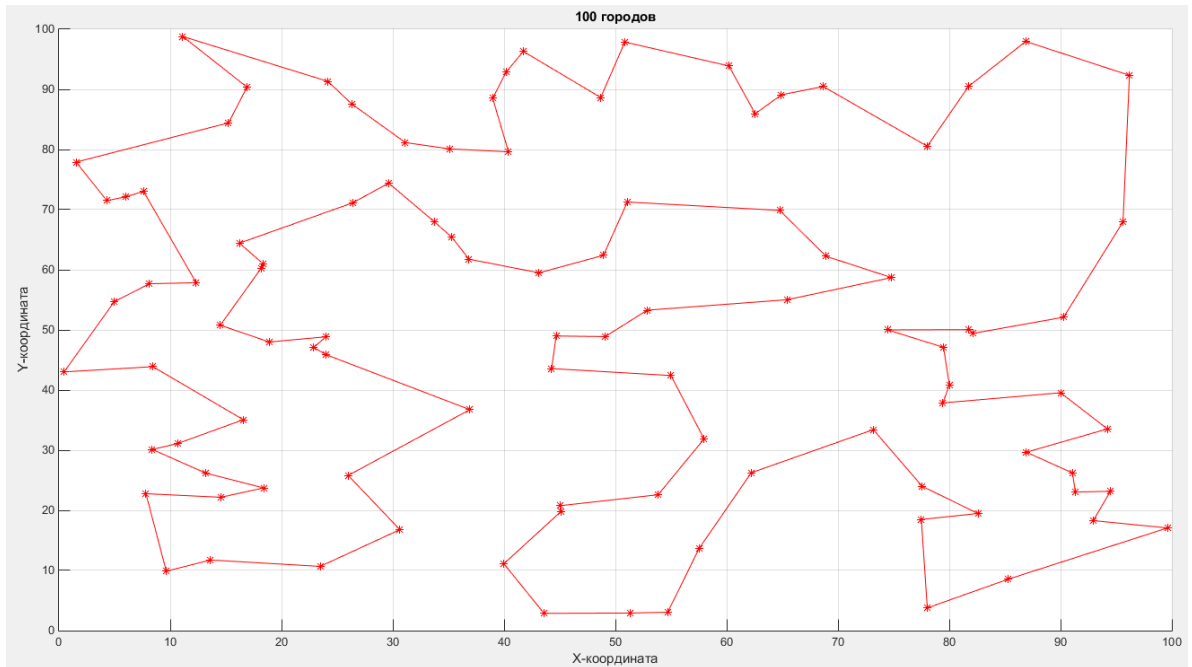


Рисунок 3.19 - Кращий шлях мурашиного алгоритму для 100 пунктів

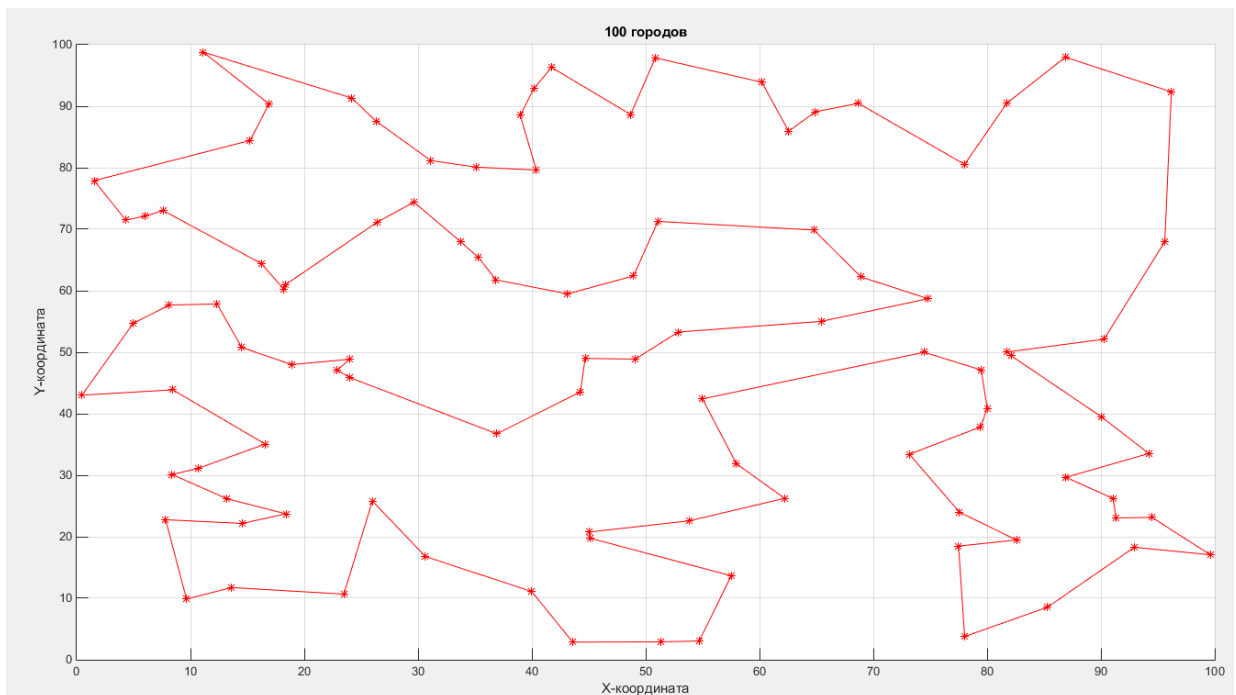


Рисунок 3.20 - Кращий шлях алгоритму імітації випалювання для 100 пунктів

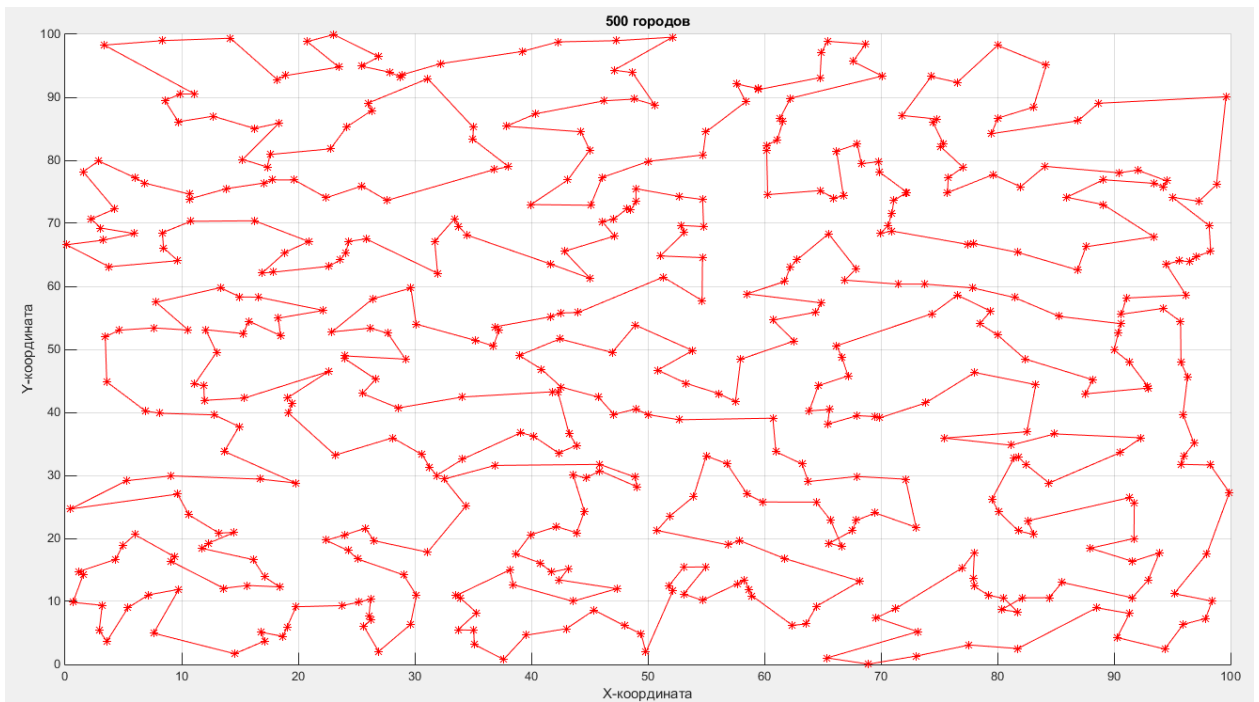


Рисунок 3.21 - Крайний шлях алгоритму імітації випалювання для 500 пунктів

Для 500 пунктів мурашиний алгоритм довго розраховує шлях і використання його при такій кількості пунктів вже не є раціональним, при цьому алгоритм імітації випалювання дає нам рішення вже за 80 секунд, що всюю чергу краще ніж мурашиний алгоритм.

З усіх розрахунків можна зробити висновок, що при не великій кількості пунктів, до 100, мурашиний алгоритм показує себе краще за всіма критеріями, але як тільки у нас велика кількість пунктів, 100 або більше, так час виконання мурашиного алгоритму дуже сильно збільшується, що говорить про не раціональності його використання при розрахунку шляху з великою кількістю пунктів.

Навіть при зменшенні вихідних параметрів мурашиного алгоритму отримаємо, кращі результати алгоритму випалювання, адже зменшивши час виконання вплинемо на результати вимірювань.

ВИСНОВКИ

У даній кваліфікаційній роботі вирішена задача дослідження методів вирішення задачі оптимізації маршруту мобільного стоку в бездротовій сенсорній мережі.

Бездротові сенсорні мережі є перспективною технологією в галузі створення побутових та промислових систем збору даних та управління.

Головним показником БСМ, що визначає їх застосовність на практиці, є час їх життя, саме тому задача максимального збільшення цього показника є актуальною як і раніше.

Розглянуто введення мобільного стоку, як одного з перспективних методів енергетичної балансування в бездротових сенсорних мережах. З метою зменшення часу затримки передачі даних при опитуванні маршрутизаторів в роботі поставлена і вирішена задача оптимізації маршруту мобільного стоку як завдання комівояжера.

Проведено порівняльний аналіз точного рішення задачі комівояжера, отриманого методом гілок і меж, і наближеного рішення, отриманого евристичними методами. Для отримання наближеного рішення були запрограмовані два евристичні алгоритми: мурашиний алгоритм та алгоритм імітації випалювання, які були реалізовані в пакеті Matlab 2014. Мурашиний алгоритм та алгоритм імітації випалювання були реалізовані для задачі комівояжера з 10, 20 та 50 пунктів з конкретними координатами для кожної задачі.

Показано, що використання евристичних алгоритмів доцільне тільки в тих випадках, коли потрібно швидко розрахувати новий маршрут для мобільного стоку. Коли швидкість розрахунку нового маршруту не є критичним, то краще використовувати точні алгоритми розрахунку маршруту. Для кожного алгоритма потрібно підбирати параметри в залежності від поставленої задачі, так як ці параметри впливають на швидкість роботи алгоритму і можуть зменшити діапазон можливих маршрутів, які можна отримати під час розрахунків.

Показано, що використання оптимізації при виборі маршруту мобільного стоку в БСМ дозволяє в середньому зменшити довжину контуру обходу мобільного стоку на 30% - 40% в порівнянні з випадковим обходом в залежності від закону

розподілу відстаней між вузлами. Таким чином, введення мобільного стоку для оптимізації маршруту опитування елементів безпроводної мережі підвищить працездатність мережі, а також збільшить час її життя та функціонування.

Отже введення мобільного стоку задля оптимізації маршруту опитування елементів безпроводної мережі підвищить працездатність системи, а також збільшить час її життя та функціонування, тому введення запрограмованого мурашиного алгоритму чи алгоритму імітації випалювання, в залежності від задачі, є ефективним та актуальним шляхом збільшення вартості та якості створеної сенсорної мережі, та зменшення витрат до яких можна призвести виведення зі строю якогось елементу мережі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Ефименко М. С., Клымов С. И., Саткенов Р. Б. Беспроводные сенсорные сети // Молодой ученый.—2018.—№51.— С. 40-42. — URL <https://moluch.ru/archive/237/55115>
2. Yang wenguo and GuoTiande, The Non-uniform Property of Energy Consumption and its Solution to the Wireless Sensor Network / 2nd International Workshop on Education Technology and Computer Science, March 6-7, 2010, pp. 186-192.
3. Vullers R., van Schaijk R., Doms I. et al. Micropower energy harvesting //Solid-State Electronics. 2009. Vol. 53, no. 7. P. 684 - 693.
4. Агафонов М. Технології безпроводової передачі даних, «Безпроводові технології» №1, 2014 р
5. Балонін Н. А. Безпроводові персональні мережі на основі ZigBee. Навчальний посібник / Н. А. Балонін, М. Б. Сергєєв. - Санкт-Петербург: ГУАП, 2014. - 58 с.
6. Плоскун, І. О. Оптимізація маршруту мобільного стоку у безпроводних сенсорних мережах [Текст] / І. О. Плоскун, В. С. Марчук // Матеріали ХХ ювілейного міжнародного форуму «Радіоелектроніка та молодь у ХХІ столітті». - 2016. - Т. 4. - С. 57-58.
7. Варгаузін В.А. Радіомережі для збору даних від сенсорів, моніторингу та управління на основі стандарту IEEE 802.15.4 // ТелеМультіМедіа. 2015. № 6. - С. 23-27.
8. Широкополосные беспроводные сети передачи информации [Текст] / В. М. Вишнеvский, А. И. Ляхов, С. Л. Портной, И. В. Шахнович - М. : Техносфера, 2005. - 592 с.
9. Мельнікова Л.І., Лінник О.В.,Кривошопка М.В.,Барсук В.О. Оптимізація маршруту мобільного стока в безпроводній сенсорній мережі // Проблеми телекомунікацій. - №1(24). - 2019. - С. 104-112.
10. Комаров М. М. Розробка і дослідження методу енергетичної балансування безпроводової стаціонарної сенсорної мережі з автономними джерелами живлення [Текст]/ М.М. Комаров.-М .: 2012 р - 125 с.

11. Комаров М. М. Позіціонірованіє датчиків безпроводової мережі як спосіб енергозбереження [Текст] / М.М. Комаров, Л.С. Восків.-М .: Датчики і системи. 2012 р - 305 с.
12. Мудров В. І. Завдання про комівояжера[Текст] / В.І. Мудров.-М .: Знание, 1969 р - 62 с.
13. Vullers, R. Micropower energy harvesting [Текст] / R. van Schaijk., I. Doms // Solid-State Electronics. - 2009. - P. 684-693.
14. Вабищевич А. Н. Определение положения в пространстве элементов беспроводной сенсорной сети с помощью инерциальных сенсоров // Тезисы докладов научно-технической конференции студентов, аспирантов и молодых специалистов МИЭМ 2010. М.: МИЭМ, 2010. С. 151-152.
15. Johnson, DS, McGeoch, LA The traveling salesman problem: a case study. Local search in combinatorial optimization / DS Johnson, LA McGeoch. - Chichester: Wiley. - P. 215-310.
16. L. Melnikova, E. Linnyk, D. Ageyev, O. Melnikova, N. Kryvoschapka and V. Barsouk, "Minimizing the Route of Sink Node in Wireless Sensor Network," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 2019, pp. 861-864, doi: 10.1109/PICST47496.2019.9061563.
17. Задача коммивояжера [Електронний ресурс]. - Режим доступу до ресурсу: https://synset.com/ai/ru/tsp/Salesman_Intro.html
18. *Галютдинов Р.Р.* Задача коммивояжера - метод ветвей и границ // Сайт преподавателя экономики. [Електронний ресурс]. - [2020]. - Режим доступу до ресурсу <http://galyautdinov.ru/post/zadacha-kommivoyazhera>
19. M. Dorigo, 1992. Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy.
20. Штовба С. Д. Муравьиные алгоритмы, Exponenta Pro. Математика в приложениях. 2004 г. № 4
21. M. Dorigo, V. Maniezzo, A. Colorni, "The Ant System: Optimization by a colony of cooperating agents" // IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26, 1, стр. 29-41, 1996 г
22. Алгоритм имитации отжига [Електронний ресурс]. - Режим доступу до ресурсу http://www.machinelearning.ru/wiki/index.php?title=Алгоритм_имитации_отжига

23. Лопатин А. С. Метод отжига / А. С. Лопатин. - Санкт-Петербург: Санкт-Петербургский государственный университет, 2005. - 17 с. - (Санкт-Петербургский государственный университет).

24. Метод отжига [Электронный ресурс] - Режим доступа до ресурсу: <https://algorithmica.org/ru/annealing>.

25. Основи теорії телекомунікаційних систем: підручник. Під ред. Поповського В.В. - Харків: ХНУРЕ, 2018. - 368 с.

26. Мельнікова Л.І., Лінник О.В., Кривошاپка М.В., Барсук В.О. Застосування евристичної процедури багатокритеріальної оптимізації до вибору варіанта мовного кодеку в IP-мережі // Проблеми телекомунікацій. - 2020. – 1(26). – С.23-32. – Режим доступу: http://pt.nure.ua/wp-content/uploads/2021/03/201_melnikova_codec.pdf

27. Кривошاپка Н.В., Барсук В.А. Пошук оптимального маршруту мобільного стоку в безпроводовій сенсорній мережі за допомогою генетичного алгоритму // Матеріали шостої Міжнародної науково-технічної конференції "Информатика, управление и искусственный интеллект (ИУИИ-2019)", - Харків, НТУ "ХПИ". – 2019. – С. 37-38.

28. А. С. Бухаров, В. А. Барсук, Н. В. Кривошاپка. Использование многокритериального подхода к оптимизации ТКС. XXIV международный молодежный форум «Радиоэлектроника и молодежь в XXI веке». – 2020. – №10. – С. 47–48.

29. Мельнікова Л.И., Барсук В.А., Кривошاپка Н.В. Адаптивное управление мощностью передачи в сетях мобильной связи // Матеріали шостої Міжнародної науково-технічної конференції «Проблеми електромагнітної сумісності перспективних безпроводових мереж зв'язку ЕМС- 2020». – Харків, ХНУРЕ. Том 4. – 2020. – С. 38-41.