

**Міністерство освіти і науки, молоді та спорту України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії**

До захисту допущено  
Завідувач кафедри  
комп'ютерної інженерії  
к.т.н., доц. О.М.Березький

\_\_\_\_\_ 20\_\_ р.

**ДИПЛОМНА РОБОТА**  
освітньо-кваліфікаційного рівня "Магістр"  
зі спеціальності 8.05010201 "Комп'ютерні системи та мережі"  
на тему:

**БАГАТОШЛЯХОВА МАРШРУТИЗАЦІЯ НА ОСНОВІ  
АЛГОРИТМІВ МУРАШКОВИХ КОЛОНІЙ**

Студент групи КСМм - 51  
Ясінчук В.І.

\_\_\_\_\_ підпис

Науковий керівник  
к.т.н., доцент Яцків Н.Г.

\_\_\_\_\_ підпис

Консультант з нормоконтролю

\_\_\_\_\_ Прізвище, ініціали

\_\_\_\_\_ Підпис

Міністерство освіти і науки, молоді та спорту України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

“Затверджую”

Зав. кафедри комп'ютерної  
інженерії  
к.т.н., доцент О.М. Березький

“ \_\_\_\_ ” \_\_\_\_\_ 2011 р.

**З А В Д А Н Н Я**

НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Ясінчука Володимира Івановича

1. Тема дипломної роботи: Багатошляхова маршрутизація на основі алгоритмів мурашкових колоній

затверджена наказом університету № \_\_\_\_\_ від „\_\_\_\_” \_\_\_\_\_ 2011 р.

2. Термін здачі закінченої дипломної роботи \_\_\_\_\_

3. Вихідні дані до роботи: безпроводні комп'ютерні мережі з різною топологією, алгоритм мурашкових колоній

4. Структура дипломної роботи

1. Аналіз методів та протоколів маршрутизації в безпроводних мереж

2. Метод маршрутизації та основі мурашкових алгоритмів

3. Реалізація та дослідження протоколу маршрутизації на основі мурашкових алгоритмів

5. Перелік ілюстративного матеріалу

1. Принцип алгоритмів мурашкових колоній

2. Багатошляхова маршрутизація в безпроводних мережах.

3. Результати дослідження алгоритму мурашкових колоній.

4. Протокол маршрутизація в безпроводних мережах.

6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Консультант	Завдання видав	Завдання прийняв

7. Дата видачі завдання “ \_\_\_\_ ” \_\_\_\_\_ 2011 р.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№	Назва структурних частин ДР	Термін виконання	Примітка
1	Аналіз методів та протоколів маршрутизації в безпроводних мереж	11.2011 – 02.2012	
2	Метод маршрутизації та основі мурашкових алгоритмів	02.2012 – 04.2012	
3	Реалізація та дослідження протоколу маршрутизації на основі мурашкових алгоритмів	04.2012 – 05.2012	

Студент \_\_\_\_\_  
(підпис)

Керівник дипломної роботи \_\_\_\_\_  
(підпис)

## РЕФЕРАТ

Дипломна робота на тему “Багатошляхова маршрутизація на основі алгоритмів мурашкових колоній” на здобуття освітньо-кваліфікаційного рівня “Магістр” зі спеціальності “Комп’ютерні системи та мережі” написана обсягом 79 сторінок і містить 22 ілюстрації, 4 таблиць, 3 додатки та 31 джерело за переліком посилань.

**Мета роботи.** Метою дипломної роботи є розробка алгоритму та протоколу багатошляхової маршрутизації на основі алгоритму мурашкових колоній.

**Методи дослідження і апаратура.** Під час проведення досліджень використовувались методи теорії інформації та кодування, теорії проектування комп’ютерних систем та мереж, алгоритми оптимізації.

При обробці результатів досліджень використовувались методи цифрового та імітаційного моделювання і програмної емуляції роботи алгоритмів.

Отримав подальший розвиток метод пошуку оптимального маршруту передачі даних в спеціалізованих мережах, який здійснює пошук незалежних маршрутів передачі даних на основі алгоритму мурашкових колоній.

Розроблено програмне забезпечення для пошук незалежних маршрутів передачі даних на основі алгоритму мурашкових колоній. Розроблено протокол маршрутизації, який здійснює пошук оптимального маршруту на основі алгоритму мурашкових колоній. Результати роботи можуть знайти широке використання при розробці протоколів маршрутизації.

Можливими напрямками подальших досліджень є продовження досліджень з пошуку оптимального маршруту з використанням багатокритеріальної оптимізації.

**Ключові слова:** КОМП’ЮТЕРНА МЕРЕЖА, БАГАТОШЛЯХОВА МАРШРУТИЗАЦІЯ, АЛГОРИТМ МУРАШКОВИХ КОЛОНІЙ, ПРОТОКОЛ ПЕРЕДАЧІ ДАНИХ, КАНАЛ ЗВ’ЯЗКУ.

## ABSTRACT

Thesis: "Multipath routing based algorithm ant colonies" of education and qualification of "Master" specialty "Computer systems and networks" written up 79 pages and contains 22 illustrations, 4 tables, 3 appendices and 31 sources references.

**The aim of the work.** The aim of the thesis is to develop algorithms and protocol Multipath routing based algorithm ants colonies.

**Methods and apparatus.** During the research methods used in information theory and coding theory design of computer systems and networks, algorithms for optimization.

When processing the results of research methods used and the digital simulation and software emulation of algorithms.

Developed further the method of finding the optimal route of data transmission in ad hoc networks, which searches for tracks of independent data-based algorithm ants colonies.

A software for searching independent routes data based algorithm ants colonies. A routing protocol that searches the optimum route based on the algorithm ants colonies. The results can be widely used in the implementation of routing protocols in wireless computer networks.

Possible directions for further research is continuing research to find the optimal route using multiobjective optimization.

**Keywords:** COMPUTER NETWORKS, MULTIPATH ROUTING ALGORITHM ANTS COLONIES, PROTOCOLS OF DATA TRANSFER, THE COMMUNICATION CHANNEL.

## ЗМІСТ

Вступ	7
1 Аналіз методів та протоколів маршрутизації в безпроводних мережах	10
1.1 Аналіз протоколів маршрутизації в безпроводних мережах	10
1.2 Класифікація протоколів маршрутизації та вимоги до них	22
1.3 Аналіз рій-орієнтованих алгоритмів маршрутизації	27
1.4 Постановка задачі дослідження	32
2 Метод маршрутизації та основі мурашкових алгоритмів	33
2.1 Принципи алгоритмів мурашкових колоній	33
2.2 Алгоритм багатошляхової маршрутизації на основі алгоритму мурашкових колоній	40
2.3 Область використання мурашкових алгоритмів оптимізації	47
2.4 Метрики оцінки вартості маршруту передачі даних	49
3 Реалізація та дослідження протоколу маршрутизації на основі алгоритму мурашкових колоній	53
3.1 Порівняння протоколів маршрутизації	53
3.2 Пошук маршруту передачі на основі алгоритму мурашкових колоній	55
3.3 Розробка програмного забезпечення пошуку маршруту в безпроводній сенсорній мережі	59
3.4 Розробка протоколу багатошляхової маршрутизації на основі алгоритму мурашкових колоній	66
Висновки	72
Список використаних джерел	74
Додаток А. Головна функція роботи програми	77
Додаток Б. Функція вибору точки в яку перейде мурашка	78
Додаток В. Довідка про використання	79

## ВСТУП

**Актуальність теми.** Останнім часом бездротові мережі є основним напрямком розвитку мережевої індустрії. Основною перевагою створення спеціалізованих мереж (мереж довільної структури) є можливість встановлення з'єднання між терміналами звідусіль і в будь-який момент часу без допомоги фіксованої, а на практиці часто жорстко визначеної і централізованої мережевої інфраструктури.

Відповідно до найбільш поширених уявлень мобільна мережа довільної структури - це мережа, спонтанно або довільно сформована без будь-якої центральної адміністративної системи і складається з мобільних вузлів, що використовують бездротовий інтерфейс для передачі пакетних даних. Вузли в мережі такого типу можуть служити як маршрутизаторами, так і хост-вузлами, і вони можуть пересилати пакети даних для інших вузлів і підтримувати виконання програм користувача.

Підкласом мобільної мережі, що використовує принцип доступу до вузлів в залежності від ситуації, що склалася, є мережі MESH. У мережі MESH вузли пов'язані один з одним. Організація мереж комірчастої структури (MESH) дозволяє маршрутизувати дані, голос і команди між вузлами цієї мережі. Мережа враховує безперервні підключення та зміни конфігурації мережі при виникненні проблем (несправний вузол або заблокований шлях), вибираючи оптимальний шлях («стрибаючи» від вузла до вузла, поки не буде досягнутий адресу призначення).

MESH - це мережева технологія, яка дозволяє недорогим вузлам мережі рівним за статусом забезпечувати двосторонній зв'язок з іншими вузлами в цій мережі і автоматично вибирати оптимальний шлях для проходження пакетів. Така можливість дозволяє ефективно розширювати існуючу інфраструктуру мережі, яка використовується для спільного доступу, з одночасним збільшенням її можливостей і дозволяє значно знизити обмеження по пропускній здатності пов'язані зі збільшенням числа

підключених бездротових клієнтів, які притаманні звичайним бездротовим мережам.

Найбільш важливим етапом розвитку MESH-мереж є завдання забезпечення необхідної якості обслуговування, тому що через мобільність вузлів та обмеження ресурсів відмови в бездротових мережах відбуваються частіше, ніж у провідних. Крім того, в таких мережах постійно зростає обсяг трафіку реального часу, чутливого до змін затримок, пропускної здатності каналів зв'язку, втрати пакетів [1 - 4]. Одним із засобів забезпечення якості обслуговування в MESH-мережах є протоколи маршрутизації.

Існуючі протоколи маршрутизації розроблені для дротових мереж без урахування динамічної топології, обмежень енергії бездротових адаптерів, пропускної спроможності каналів зв'язку. У зв'язку з цим виникає необхідність у розробці нових адаптивних методів маршрутизації, що реагують на зміни маршрутів між вузлами та враховують обмеження спеціалізованих безпроводних мереж.

**Мета і завдання дослідження.** Метою дипломної роботи є розробка алгоритму та протоколу багатошляхової маршрутизації на основі алгоритму мурашкових колоній.

Для вирішення цієї мети у дипломній роботі поставлено такі завдання:

- 1) провести аналіз методів та протоколів маршрутизації в безпроводних мереж;
- 2) розробка алгоритму пошуку незалежних маршрутів передачі даних;
- 3) розробка протоколу багатошляхової маршрутизації;
- 4) на основі розробленого алгоритму здійснити програмну реалізацію пошуку незалежних маршрутів передачі даних.

*Об'єкт дослідження* – процес визначення маршруту передавання даних в безпроводних комп'ютерних мережах.

*Предмет дослідження* – методи та алгоритми маршрутизації в безпроводних комп'ютерних мережах.



**Методи досліджень.** Під час проведення досліджень та обробки їх результатів використовувались методи теорії інформації та кодування, теорії проектування комп'ютерних систем та мереж, алгоритми оптимізації.

**Наукова новизна одержаних результатів.**

Отримав подальший розвиток метод пошуку оптимального маршруту передачі даних в спеціалізованих мережах, який здійснює пошук незалежних маршрутів передачі даних на основі алгоритму мурашкових колоній.

**Практичне значення отриманих результатів.**

Розроблено програмне забезпечення для пошук незалежних маршрутів передачі даних на основі алгоритму мурашкових колоній.

Розроблено протокол маршрутизації, який здійснює пошук оптимального маршруту на основі алгоритму мурашкових колоній.

**Публікації та апробація ДР.** Результати дипломної роботи доповідались та опубліковані в матеріалах школи-семінар молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології»

# 1 АНАЛІЗ МЕТОДІВ ТА ПРОТОКОЛІВ МАРШРУТИЗАЦІЇ В БЕЗПРОВІДНИХ МЕРЕЖ

## 1.1 Аналіз протоколів маршрутизація в безпроводних мережах

Протоколи маршрутизації, що використовуються в MESH-мережах, засновані на традиційних алгоритмах маршрутизації, таких як дистанційно-векторний алгоритм маршрутизації і маршрутизація з урахуванням стану зв'язків. В залежності від ситуації, що склалася в MESH-мережах реалізуються різні стратегії маршрутизації, такі як пряма маршрутизація, маршрутизація в залежності від географічного місця розташування, гібридна, ієрархічна і т. д.

### 1.1.1 Проактивні протоколи маршрутизації. Пряма маршрутизація.

Mesh-маршрутизатори утворюють бездротову транспортну Mesh-мережа для надання мобільним абонентам інтегрованого доступу до зовнішніх мереж, оснащених терміналами різного рівня (рисунок 1.1) [5].

В основі проактивних протоколів маршрутизації лежать дистанційно-векторні алгоритми і алгоритми маршрутизації по стану каналу зв'язку (Link State, LS). Маршрутизація здійснюється на основі таблиць. Для збору та підтримки актуальної інформації про маршрути, маршрутизатори періодично обмінюються службовою інформацією, навіть коли система знаходиться в стійкому стані. До проактивних протоколів маршрутизації відносяться відомі протоколи маршрутизації, добре протестовані і широко використовуються в сучасних проводових мережах: протокол RIP, заснований на дистанційно векторному алгоритмі, протокол OSPF, заснований на алгоритмі станом каналу.

Найбільш відомим проактивним протоколом маршрутизації є протокол - Fisheye State Routing (FSR). FSR є простим і ефективним протоколом типу LS, який підтримує топологію мережі в кожному вузлі і оновлює інформацію про стан каналів. Головна відмінність між FSR протоколами і протоколами

типу LS полягає в способі поширення маршрутної інформації. По-перше, при використанні FSR відбувається обмін інформацією про стан зв'язків тільки між сусідніми вузлами, а не по всій мережі. Інформація в таблиці станів каналів постійно оновлюється за рахунок інформації, отриманої від сусідніх вузлів. По-друге, обмін інформацією про стани каналів оновлюється періодично, так як постійні часті оновлення руйнують канали в мобільній мережі. Крім того, періодичні розсилки інформації про стан каналів відбуваються на різних частотах для різних входів в залежності від відстані до поточного вузла.

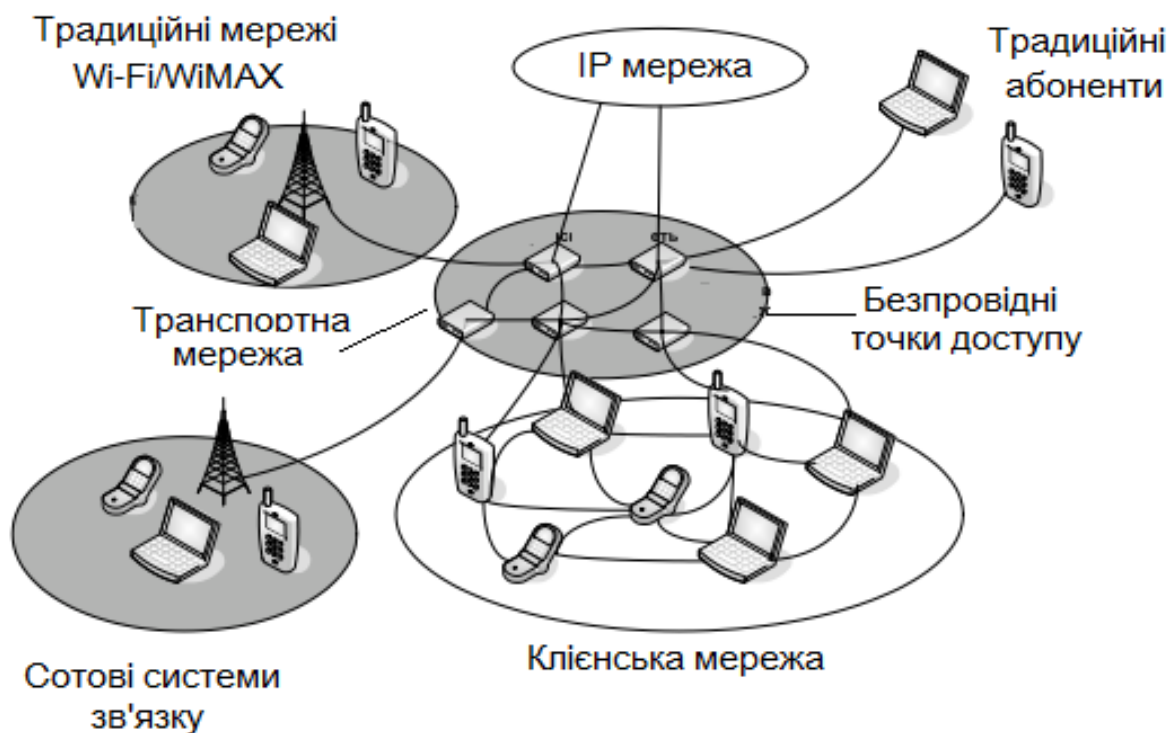


Рисунок 1.1 – Архітектура Mesh-мережі

Пакети, призначені для найбільш віддалених вузлів (поза визначеною областю), передаються на більш низьких частотах, ніж для передачі до сусідніх вузлів. При використанні FSR визначається точна відстань і інформація про маршрут до сусіднього вузла і неточний найкоротший шлях при з'єднанні з далеким вузлом. Однак ця неточність компенсується при

наближенні пакета до призначеного вузла, тобто маршрут стає більш визначеним при наближенні до шуканого вузла. Подібний принцип використовується і в Fuzzy Sighted Link State (FSLs) [6].

З точки зору використання проактивних протоколів маршрутизації в бездротових мережах необхідно відзначити наступні недоліки:

- в основу даних протоколів покладені алгоритми вибору єдиного найкоротшого маршруту, які не завжди ефективні з точки зору якості сервісу;

- проактивні протоколи засновані на традиційних (класичних) алгоритмах маршрутизації, розроблених для стаціонарних дротових мереж, без урахування динамічної природи мобільних мереж і особливостей бездротової середовища зв'язку;

- періодичний обмін службовими повідомленнями тягне за собою різке збільшення службового трафіку і непродуктивна витрата пропускну здатності та енергії мобільних терміналів;

- недостатньо швидко реагують на зміни топології мобільної мережі, тобто час збіжності використовуваних алгоритмів маршрутизації більше, ніж частота зміни маршрутів.

На підставі сказаного можна зробити висновок, що відомі протоколи маршрутизації, засновані на проактивному підході неефективні для використання в мобільних мережах. Однак проактивний підхід має явну перевагу, яка полягає в тому, що коли вузлу потрібно відправити пакет, потрібний маршрут вже побудований в таблицях маршрутизації. Цей факт є суттєвим для скорочення часу формування маршруту передачі даних, тому розглянуті алгоритми можуть бути використані в якості базових алгоритмів для розробки протоколів маршрутизації для мереж з динамічною топологією.

### 1.1.2 Реактивні протоколи маршрутизації

Для вирішення задач маршрутизації в мобільних мережах більш ефективні реактивні протоколи маршрутизації (AODV, DSR, TORA), які формують маршрут передачі даних на вимогу. Такі протоколи не вимагають

ніяких зайвих витрат при виникненні змін в топології мобільної мережі, особливо при незначному русі вузлів. Реактивні протоколи засновані на лавинних алгоритмах маршрутизації і алгоритмах маршрутизації від джерела і здатні динамічно реагувати на зміни топології мобільної мережі [7].

Одним з таких протоколів є Ad hoc On Demand Distance Vector (AODV). Для кожного об'єкта в мережі таблиці маршрутизації містять інформацію про наступний вузол на шляху до вузла одержувачу, часові мітки і службову інформацію. Маршрути підтримуються тільки в тому випадку, якщо вони дійсно використовуються. Щоб уникнути зациклення і для підтвердження того, що маршрут дійсно «свіжий», використовуються порядкові номери. AODV використовує такі типи пакетів: привітальні повідомлення, запит про маршрут зв'язку (RREQ) (рисунок 1.2), відповідь про маршрут зв'язку (RREP), помилка маршруту зв'язку (RERR).

type	flags	resvd	hopcnt
<b>broadcast_id</b>			
<b>dest_addr</b>			
<b>dest_sequence_#</b>			
<b>source_addr</b>			
<b>source_sequence_#</b>			

Рисунок 1.2 – Таблиця маршрутизації протоколу AODV

Якщо при передачі інформації через вузол шлях до адресата не відомий, ініціюється широкомовна передача повідомлень RREQ всім «сусідам». Маршрути, за якими слідував запит, фіксуються на проміжних вузлах. Повідомлення RREQ будуть розмножені через мережу у всіх напрямках, поки не досягнуть адресата або вузла з досить «свіжим» маршрутом, після чого по зворотному шляху від одержувача до відправника

надсилається повідомлення типу RREP (рисунок 1.3), встановлює на проміжних вузлах шлях для подальшої передачі пакетів даних.

<b>type</b>	<b>flags</b>	<b>rsvd</b>	<b>prsz</b>	<b>hopcnt</b>
<b>dest_addr</b>				
<b>dest_sequence_#</b>				
<b>source_addr</b>				
<b>lifetime</b>				

Рисунок 1.3 – Широкомовна передача повідомлень в протоколі AODV

Основні проблеми протоколу AODV полягають в наступному [8]:

- розриви в маршруті передачі даних призводять до втрати пакетів даних або ініціюють комплексну процедуру відновлення маршруту;
- обсяг контрольного трафіку зростає із збільшенням щільності та розміру мережі;
- довгі маршрути занадто нестійкі;
- не підтримуються множинні маршрути до вузла одержувача.

Переваги реактивних протоколів:

- вузли містять тільки активні маршрути;
- маршрути розраховуються тільки по необхідності;
- швидкий відгук на зміни, викликані мобільністю вузлів.

Недоліки реактивних протоколів:

- можливість високої латентності між необхідністю послати пакет і отриманням маршруту;
- необхідність утримання кешем маршрутів.

### 1.1.3 Маршрутизація в залежності від географічного положення

Маршрутизація в залежності від географічного положення (Geographic Addressing and Routing, GeoCast) дозволяє розсилати повідомлення у всі вузли в певній географічній області, використовуючи географічну інформацію замість логічних адрес вузла [9].

Географічний адресу виражений трьома способами: точка, коло (з крапкою в центрі і радіусом), і багатокутник (список точок, наприклад,  $T(1)$ ,  $T(2)$ , ...,  $T(n-1)$ ,  $T(n)$ ,  $T(1)$ ). Точка представлена географічними координатами (широта і довгота). Коли повідомлення буде направлено за адресою - багатокутник або коло, кожен вузол в межах географічної області багатокутника / кола отримає повідомлення. Географічний маршрутизатор (GeoRouter) обчислює свою зону дії (певна географічна область) (рисунок 1.4).

Ця зона дії обмежена єдиним закритим багатокутником. GeoRouters обмінюються інформацією між прилеглими зонами багатокутників для побудови таблиць маршрутизації. Цей підхід дозволяє побудувати ієрархічну структуру складається з маршрутизаторів GeoRouters. Кінцеві користувачі можуть вільно пересуватися в мережі.

Передача даних починається від хоста, що відсилає і одержує «географічні» повідомлення (GeoHost). Пакети даних надсилаються до місцевого GeoNode (що знаходиться в кожній підмережі), який здійснює відправку пакетів місцевим маршрутизатору GeoRouter. GeoRouter спочатку перевіряє, перетинає його зона дії прилеглу область маршрутизації [10].

Якщо частина області маршрутизації не покрита, GeoRouter посилає копію пакета до його головного маршрутизатору для подальшої маршрутизації поза його власної зони дії. Потім перевіряє зону дії його «дочірніх» маршрутизаторів для можливого перетину з іншими областями. Усім «дочірнім» маршрутизаторам, які перетинають потрібну область, посилають копію пакета. Коли зона дії маршрутизатора знаходиться в межах потрібної області, маршрутизатор пересилає пакет до GeoNodes. Географічні

повідомлення, які надходять періодично розсилаються і зберігаються протягом певного часу.

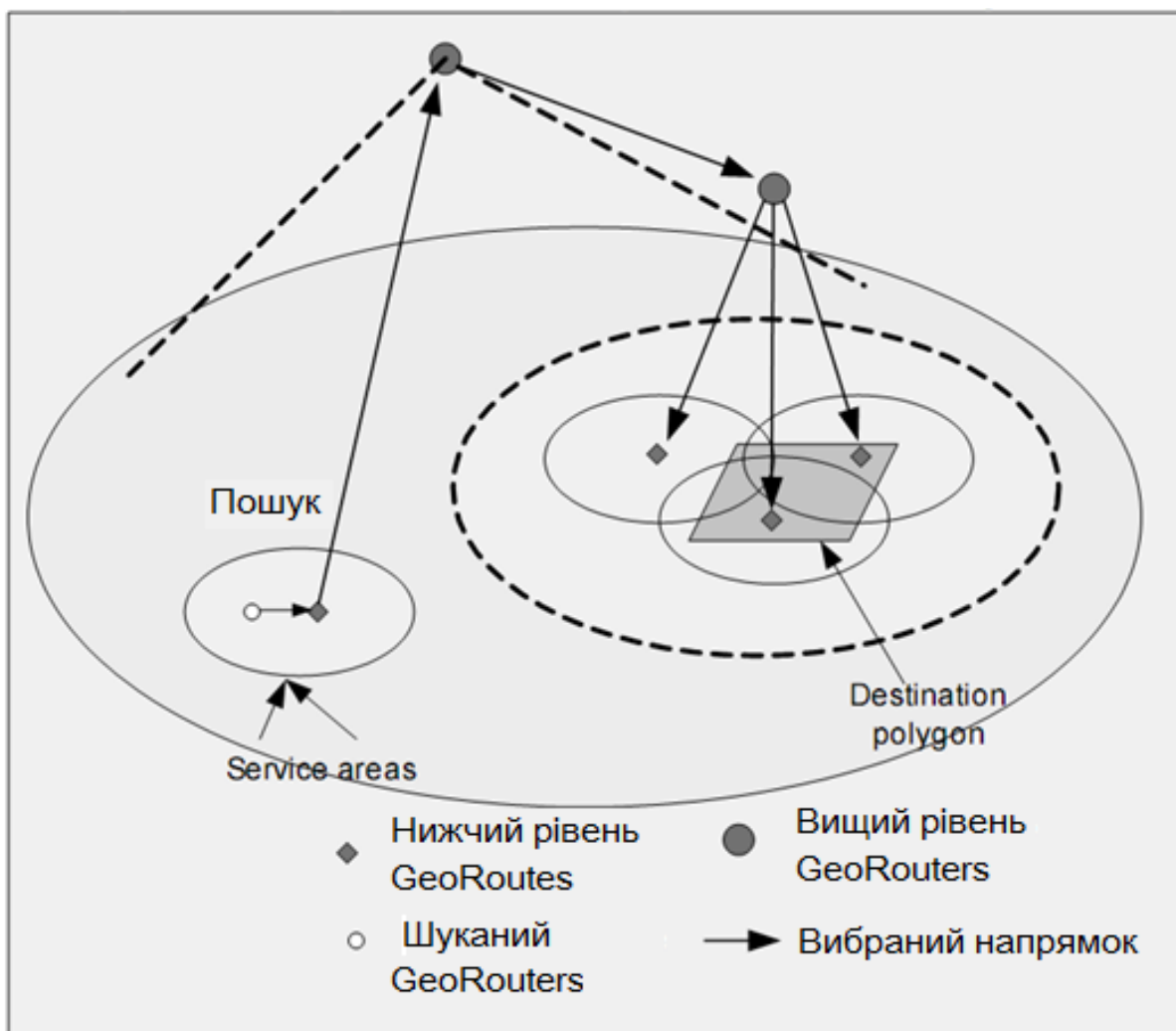


Рисунок 1.4 – Маршрутизація в залежності від географічного положення

Найбільш використовуваний протокол географічної маршрутизації - The Location-Aided Routing (LAR), заснований на маршрутизації від джерела. Протокол використовує інформацію про місцезнаходження для мінімізації області виявлення нового маршруту, тобто для виявлення найменшого маршруту до зони запиту. Як наслідок, кількість повідомлень про запит шуканого маршруту скорочена. Використовуючи інформацію місця розташування, LAR виконує пошук маршруту через обмежену розсилку (надсилає запити до зони запиту).



Як тільки вузли в зоні запиту відправляють повідомлення про встановлення потрібного маршруту, ЛАР забезпечує дві схеми для визначення зони запиту [11].

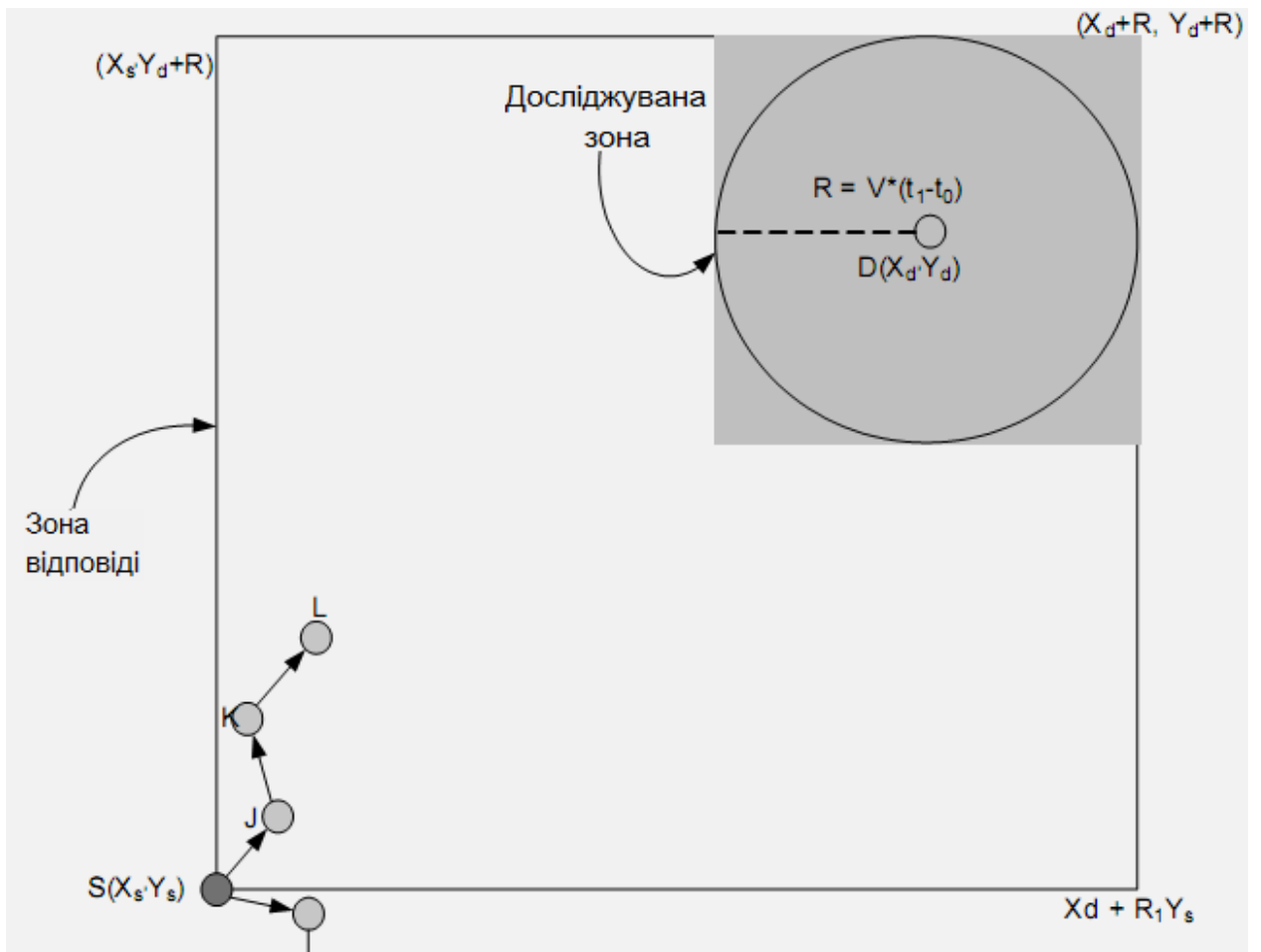
Схема 1: Джерело оцінює шукану область (очікувана зона), в якій буде встановлено з'єднання за певний час. Положення і розмір області визначаються на підставі попереднього місця розташування. Найменша прямокутна область, яка включає шукану зону і джерело, є зоною запиту (рисунок 1.5 а). Координати чотирьох кутів зони приєднані до запиту маршруту і до джерела. Під час розсилки запиту маршруту тільки вузли в зоні запиту відправляють повідомлення-відповідь.

Схема 2: Джерело обчислює відстань до одержувача, заснованому на шуканому місцезнаходженні. Це відстань, поряд з місцем розташування одержувача, включено до повідомлення запиту маршруту та розіслано сусідам. Коли вузол отримує запит, він обчислює свою відстань до одержувача. Вузол передасть повідомлення запиту далі тільки в тому випадку, якщо його відстань до вузла-одержувача менше або дорівнює відстані, включеній до повідомлення запиту. Наприклад, на рисунку 1.5 б, вузли I і J відправляють запити від S. Перш, ніж вузол передає запит, він оновлює дані про відстань в повідомленні з його власним відстанню до одержувача [12].

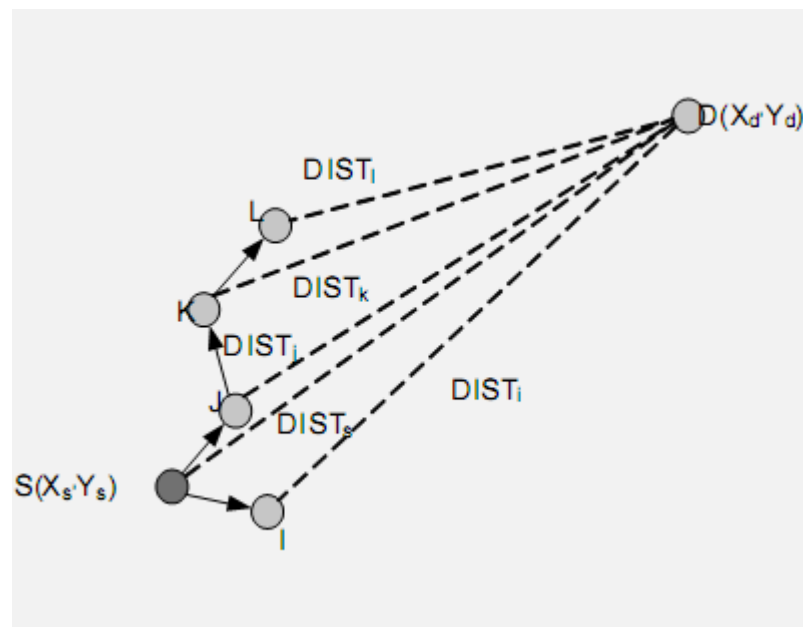
Передбачається, що всі вузли оснащені приймачами системи мобільного позиціонування (GPS, ГЛОНАСС) і можуть визначити своє географічне положення. Між мобільними об'єктами в мережі відбувається постійний обмін інформацією про місцезнаходження один одного.

Переваги протоколів географічної маршрутизації:

- Географічна маршрутизація не вимагає витрат на створення маршруту і його підтримку в актуальному стані;
- Можливість виявити необхідний вузол, знаючи тільки його стан;
- Здатність прокласти маршрут, враховуючи енергію вузлів.



а)



б)

Рисунок 1.5 – Протокол географічної маршрутизації

Недолік протоколів географічної маршрутизації полягає в неефективності отримати маршрут до вузла по його ідентифікаційним номером.

#### 1.1.4 Ієрархічна маршрутизація в спеціалізованих мережах

Протокол ієрархічної маршрутизації Hierarchical State Routing (HSR) є багаторівневим об'єднанням в кластери. Протокол підтримує логічну ієрархічну топологію при використанні рекурсивно згрупованої схеми. Вузли на одному логічному рівні згруповані у групи. Вибрані вузли, так звані - clusterheads, на більш низькому рівні стають членами наступного більш високого рівня. Потім на наступному рівні утворюють групи, і так далі. Мета об'єднання в кластери полягає в тому, щоб зменшити маршрутизацію (тобто, зберігання таблиць маршрутизації, обробка, і передача) на кожному рівні. Приклад ієрархічної структури з трьома рівнями продемонстрований на рисунку 1.6. Взагалі, є три види вузлів в групі: clusterheads (наприклад, вузли 1, 2, 3, і 4), шлюзи (наприклад, вузли 6, 7, 8, і 11), і внутрішні вузли (наприклад, вузли 5, 9, і 10). Clusterhead діє як місцевий координатор для передач інформації в межах групи.

На першому рівні відбувається об'єднання в кластери, де кожен вузол контролює стан зв'язку зі своїми сусідами і передає службову інформацію в межах групи.

Clusterhead підсумовує інформацію про стан зв'язку в межах даної групи і передає її до сусідніх вузлів групи (через шлюз). Можливість з'єднання між сусідніми clusterheads призводить до формування рівня 2 групи. Як показано на рисунку 1.6, записи про стан зв'язку на рівні 2 вузла містять "дійсні" зв'язки в С2. "Дійсний" зв'язок між сусідніми вузлами 1 і 2 рівня 1 формує шлях від clusterhead 1 до clusterhead 2 через шлюз 6. «Дійсний» зв'язок може бути розглянутий як "тунель", здійснений через більш низький рівень.

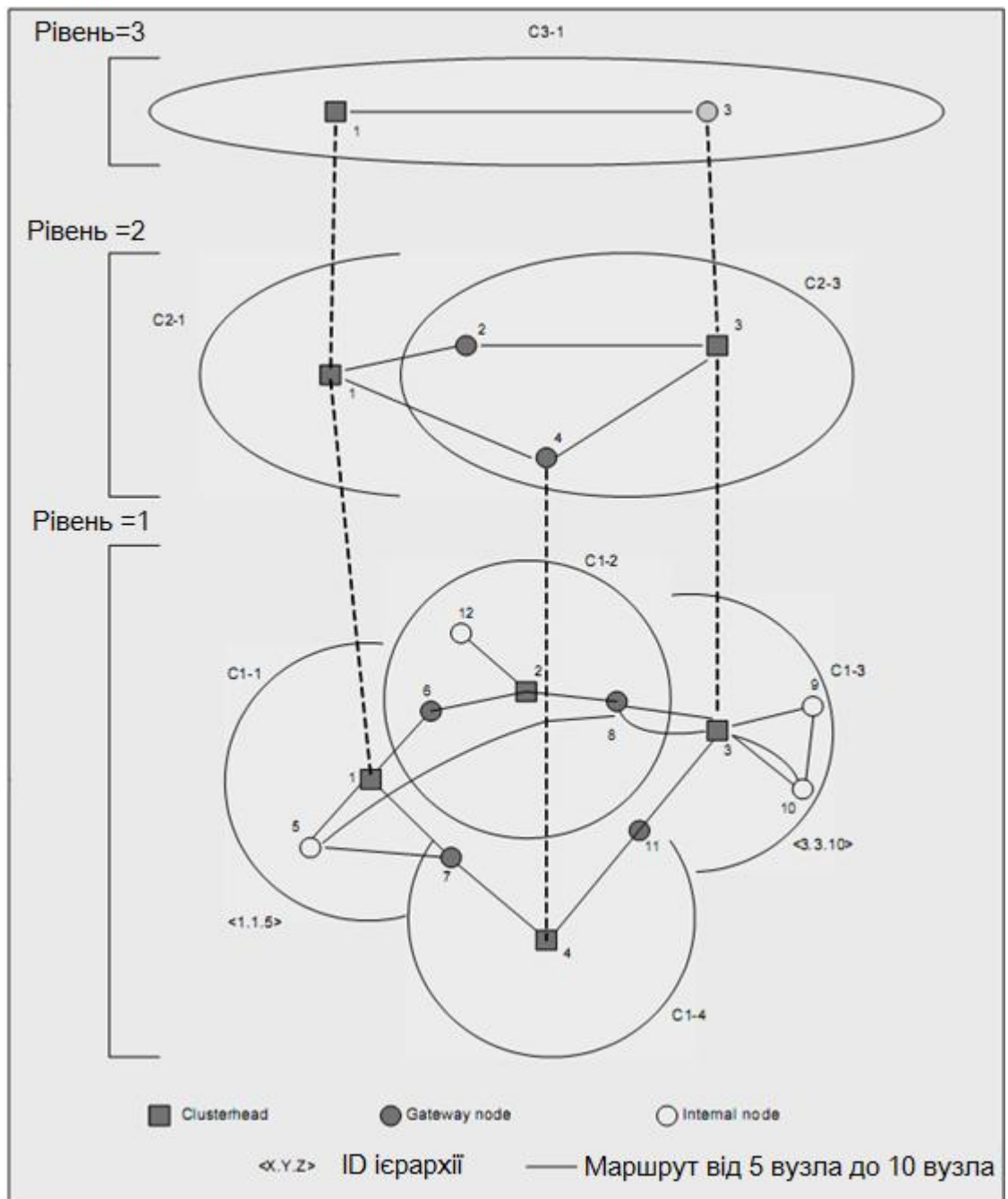


Рисунок 1.6 – Ієрархічна маршрутизація

Застосовуючи вищезгадану процедуру рекурсивного об'єднання в кластери, нові «головні» вузли групи обрані на кожному рівні, і потім стають, більш високорівневою групою. Якщо будуть вимагатися параметри QoS, clusterheads буде обробляти інформацію про рівень, якому він належить і передає її на більш високий рівень. Після отримання інформації про стан

зв'язку на одному рівні кожен вузол розсилає її до вузлів груп нижчого рівня. В результаті у кожного фізичного вузла є "ієрархічна" інформація про топологію через ієрархічну адресу кожного вузла.

#### 1.1.5 Гібридна маршрутизація

Гібридні протоколи маршрутизації мають властивості як дистанційно-векторних протоколів маршрутизації, так і протоколів за станом каналів. Маршрутна інформація розсилається тільки своїм сусідам і відбувається це так само як і в протоколах за станом каналів, при внесенні будь-яких змін в топології мережі [13].

Гібридні протоколи, такі як ZRP і HSLS, використовують змішаний підхід, коли для маршрутизації до певної безлічі вузлів «ближньої зони» використовується проактивний протокол, а до вузлів «дальньої зони» - реактивний протокол (наприклад для ZRP це протоколи IARP і IERP). У кожного з підходів є свої переваги і недоліки.

Так, наприклад, проактивні протоколи забезпечують найвищу швидкість маршрутизації, але генерують при цьому значний обсяг службових пакетів. Реактивні протоколи, хоча і надають мінімальний вплив на пропускну здатність мережі, надають маршрутну інформацію зі значною затримкою. Гібридні протоколи, що поєднують в собі переваги проактивних і реактивних протоколів в цілому є більш ефективними.

Слід зазначити, що для забезпечення QoS більшість протоколів неприйнятні і необхідна їх адаптація або розробка нових динамічних алгоритмів для забезпечення необхідного рівня якості сервісу. У зв'язку з цим, до протоколів маршрутизації в MESH - мережах висуваються спеціальні вимоги.

В даний час існує безліч моделей QoS [7-9], які використовуються в провідних мережах. В об'єднаних мережах завдання забезпечення QoS розглядають на макрорівні, тобто на рівні взаємодії складових локальних мереж. Мережева структура на макрорівні змінюється відносно рідко, що

також дозволяє використовувати відомі методи. Проте в бездротових мережах маршрутизація та забезпечення необхідного QoS, ґрунтуються на динамічній природі цих мереж [10-12], що зумовлює використання динамічних алгоритмів забезпечення QoS. Існуючі моделі QoS неприйнятні для використання в мобільних мережах Ad Hoc і необхідна їх адаптація для використання в таких мережах.

## 1.2 Класифікація протоколів маршрутизації та вимоги до них

Існуючі на даний момент протоколи маршрутизації можна класифікувати наступним чином [1, 2, 14 ].

За типом даних, які використовуються для маршрутизації:

1) топологічні. Використовують інформацію про існуючі мережеві з'єднання між вузлами мережі;

2) географічні. Використовують дані про географічне положення вузлів, одержувані звичайно за допомогою супутникової навігації, для прогнозування можливості або неможливості зв'язку між окремими вузлами і про можливі в даній мережі маршрути.

За принципом роботи:

1) проактивні або табличні (англ. proactive, table-driven). Періодично розсилають по мережі службові повідомлення з інформацією про всі зміни в її топології. В результаті кожен вузол мережі на основі даної інформації будує маршрути до всіх інших вузлів і зберігає їх в таблиці маршрутизації, звідки вони зчитуються при необхідності надіслати повідомлення будь-якому адресату;

2) реактивні або ті, що працюють за запитом (англ. reactive, on-demand). Складають маршрути до конкретних вузлів лише при виникненні необхідності у передачі їм інформації. Для цього вузол-відправник широкомовно розсилає по мережі повідомлення-запит, який повинен дійти до

вузла-адресата. У відповідь адресат посилає повідомлення підтвердження, з якого відправник дізнається необхідний маршрут і записує його в свою таблицю маршрутизації. Для повторних відправок повідомлень даному адресату маршрут просто зчитується з таблиці. Якщо буде виявлено його руйнування, то запускається так звана процедура підтримки маршруту, яка фактично полягає в пошуку нового маршруту до адресата;

3) гібридні (англ. hybrid). Дані протоколи комбінують механізми проактивних і реактивних протоколів. Як правило, вони розбивають мережу на безліч підмереж, усередині яких функціонує проактивний протокол, а взаємодія між ними здійснюється реактивними методами. У великих мережах це дозволяє скоротити розміри таблиць маршрутизації, які зберігають вузли мережі, тому що їм необхідно знати точні маршрути лише для вузлів підмережі, до якої вони належать. Також скорочується і обсяг службової інформації, який розсилається по мережі, тому що основна її частина поширюється лише в межах підмереж.

За критерієм визначення оптимальності маршруту:

1) протоколи вектора відстані (англ. distance-vector, hop-count). Завжди вважають оптимальним маршрут, який містить найменше число хопів (ретрансляції пакета) між відправником та адресатом;

2) протоколи зі складною метрикою маршрутів або протоколи стану каналів (англ. link-state). Використовують комплексну оцінку маршрутів за кількома параметрами, в які, крім числа хопів, зазвичай входять затримка на доставку пакета, пропускна здатність каналу та ін. За кількістю задіяних в маршрутизації рівнів EMBOS :

1) однорівневі. Працюють чисто на мережевому рівні, тому володіють універсальністю і мережевою прозорістю;

2) міжрівневі. В процесі роботи взаємодіють з іншими рівнями, наприклад, отримуючи від них інформацію про додаткові метрики маршруту або навіть використовуючи їх заголовки для передачі службової інформації. Теоретично дозволяють здійснювати маршрутизацію більш оптимальним

чином, але прив'язані до конкретних реалізацій інших рівнів ЕМВОС і тому не є універсальними.

За наявності підтримки декількох маршрутів до одного адресата:

1) одношляхові (англ. single-path). При обробці інформації про топологію мережі дані протоколи вибирають лише один, найбільш оптимальний, маршрут до адресата і заносять його в свою таблицю маршрутизації;

2) багатошляхові (англ. multi-path). На відміну від одношляхових, заносять в таблицю маршрутизації два або декілька найбільш оптимальних маршрутів до адресата. У разі виявлення руйнування основного маршруту дані протоколи просто зчитують з таблиці запасний, а не ініціалізують процедуру повторного побудови маршруту.

Також при виявленні перевантаження мережі за основним маршрутом багатошляхові протоколи можуть перерозподіляти частину навантаження на резервні.

Для успішного застосування в Ad hoc-мережах протоколи маршрутизації повинні мати наступні властивості [15]:

1) бути розподіленими. Всі вузли в мережі повинні бути здатні здійснювати маршрутизацію і не мати жорстко закріплених за собою функцій;

2) забезпечувати надійну доставку пакетів в умовах постійно мінливих топології мережі, коли використання класичних механізмів гарантованої доставки, як, наприклад, на транспортному рівні в протоколі ТСП, утруднено.

3) забезпечувати малий час побудови маршруту в умовах постійно мінливих топології мережі;

4) володіти механізмами оперативного виявлення розриву маршруту та його відновлення;

5) не допускати утворення петель в маршрутах;

6) розсилати при функціонуванні якомога менший обсяг службової інформації;



7). мати високу масштабованість, тобто забезпечувати високу продуктивність мережі при різних її розмірах;

8) підтримувати QoS.

Кожен клас протоколів потенційно має свої переваги та недоліки при використанні в умовах мобільних спеціалізованих мережах. Наприклад, проактивні протоколи володіють явною перевагою перед реактивними у часі побудови маршруту. У проактивних протоколів цей процес, по суті, відбувається заздалегідь, і потрібно лише зчитати маршрут з таблиці, тоді як реактивним протоколам необхідно розіслати ширококомовний запит і дочекатися підтвердження від адресата. Однак проактивним протоколам необхідно постійно здійснювати ширококомовні розсилання, на що може витрачатися значна частка пропускну здатності мережі, особливо в умовах великих мереж з високою мобільністю вузлів.

Однак наведені вище міркування є суто абстрактними і відірваними від конкретних практичних реалізацій. Адекватне порівняння протоколів з чисто теоретичних позицій ускладнене тим, що на процес передачі даних в Ad hoc-мережах впливає велика кількість різних факторів, багато з яких носять випадковий характер і слабо піддаються строгому математичному аналізу. Тому основним інструментом порівняльного аналізу протоколів маршрутизації при роботі в Ad hoc-мережах є імітаційне моделювання, яке в цілях економії часу і засобів спочатку здійснюється за допомогою комп'ютерних програм-симуляторів, без застосування реального обладнання [16].

### 1.2.1 Багатошляхова маршрутизація

В алгоритмах багатошляхової маршрутизації для кожного адресата обчислюється кілька непересічних шляхів, що дозволяє оптимально використовувати канали зв'язку і підвищувати їх загальну пропускну здатність. Крім того, багатошляхова маршрутизація забезпечує простий механізм для збільшення ймовірності надійної доставки даних за рахунок

відправлення декількох копій даних за різними маршрутами. Однак, використання протоколів багатошляхової маршрутизації призводить до збільшення енергетичних витрат і підвищення трафіку мережі.

Багатошляхова маршрутизація використовується в провідних мережах для підвищення надійності і можливості збалансування навантаження трафіку в мережі [16-18]. В останні роки цей метод поширюється на бездротові сенсорні мережі з метою досягнення кращої енергоефективності та надійності в разі збоїв вузлів мережі.

В роботах [19, 20] багатошляхова маршрутизація пропонується для покращення енергетичної ефективності сенсорних мереж за рахунок зниження частоти відкриття маршруту багатошляхове поширення динамічної маршрутизації від джерела (Dynamic Source Routing - DSR) і дистанційно-векторний алгоритм маршрутизації (Ad hoc On-Demand Distance Vector - AODV) на вимогу.

Дані протоколи з однією базовою станцією відкривають кілька альтернативних маршрутів. Коли відбувається збій основного маршруту при передачі даних, трафік даних переключється на найкращий альтернативний маршрут. Нове відкриття маршруту відбувається тільки тоді, коли всі попередні виявлені шляхи не вдається використати. Протокол багатошляхової маршрутизації на вимогу, запропонований в [21], здатний знаходити маршрути, які не перетинаються. Це досягається шляхом порівняння інформації вузла з інформацією маршруту, яку несе запит повідомлення. Запит повідомлення потім направляється вибірково в деякі вузли. В [19, 20] енергозбереження досягається в основному за рахунок зниження частоти відкриття маршруту. В [21] багатошляхова маршрутизація (MultiPoint Relay - M-MPR) спрямована на збільшення пропускну здатності трафіку БСС. Процес відкриття багатьох маршрутів використовує переваги географічної інформації відомої датчикам вузлів. У M-MPR, бездротовий вузол обмежує число запитів для поширення маршрутизації і пересилає запит тільки до обмеженого числа розташованих нижче за напрямом передачі

вузлів для того, щоб уникнути зациклення. Балансування навантаження досягається шляхом вибіркового пересилання даних в один з розташованих нижче на маршруті вузлів. Недоліком цієї схеми є те, що немає можливості вибрати оптимальний маршрут, так як в наявності кожного вузла знаходиться тільки часткова інформація про мережу. Схема балансування навантаження не очікує даних одночасно від декількох пересічних маршрутів.

В роботі [22] підхід багатошляхової маршрутизації пропонується для поліпшення протоколу спрямованої дифузії [23] з метою підвищення стійкості вузла до збоїв. В [22] також досліджується можливість швидкого пошуку альтернативних шляхів, що з'єднують джерела з базовою станцією, коли вузли ізольовані або відбувається відмова вузла. Запропоновано локалізовані алгоритми для пошуку безлічі непересічних розгалужених шляхів. Використання безлічі шляхів, дозволяють уникнути періодичної лавинної розсилки низькошвидкісних повідомлень, які необхідні для підтримки альтернативних шляхів в оригінальному протоколі спрямованої дифузії [23]. Використання декілька розгалужених шляхів працює краще з точки зору забезпечення накладних витрат, особливо для мереж з низькою щільністю вузлів [22]. Вони також стверджують, що підхід використання декількох шляхів може поліпшити надійність мережі без шкоди для енергоефективності. Використання багатошляхової маршрутизації є одним з найбільш ефективних способів підвищення надійності передачі даних на мережевому рівні безпроводних комп'ютерних мереж [24, 25].

### 1.3 Аналіз рій-орієнтованих алгоритмів маршрутизації

Існує великий обсяг роботи з головною проблемою маршрутизації в мережі. Бездротові мережі мають ряд труднощів, пов'язаних з динамічним характером їх топології, пов'язаних з рухом вузлів, радіоперешкодами, помилками вузлів, і новими додатками. Різні протоколи маршрутизації які

були запропоновані і найбільш ефективні схеми як правило, залежать від конкретних характеристик операційної системи (наприклад, розподіл підключення і зміна топології).

Ройовий інтелект [10, 11] в даний час активно вивчається для застосування маршрутизації в комунікаційних мережах. France Telecom і British Telecommunications (BT) додали ройовий інтелект до своїх телефонних мереж. MCI Worldcom також серйозно досліджує ройовий інтелект для управління телефонною мережею в Сполучених Штатах.

Потенційні переваги ройового інтелекту в порівнянні зі звичайними централізованими підходами телекомунікацій є надзвичайно переконливими. New Scientist [26] недавно повідомив тривожні новини про проблеми з мережею BT, і про дослідження компаніями ройового інтелекту в якості можливого їх рішення. У BT 24 мільйони користувачів координуються через звичайного web-диспетчера, що, в 1995, складався з 30 програм з вимогами середньої пам'яті 350 гігабайтів. Велика частина часу витрачається для перевірки, що всі елементи мережі працюють. Слід також постійно оновлюватися в міру підключення нових абонентів, нових послуг і нових проблем які виникають. Чим більшою стає мережа тим все важче адаптуватися, і помилка у центрі може мати потенційно катастрофічні наслідки в усій мережі. Розподілений характер ройового інтелекту дозволяє уникнути вузьких місць, які є результатом постійного використання такого централізованого управління.

В даний час розроблені алгоритми маршрутизації для сенсорних мереж зазвичай припускають рівні об'єми даних і пріоритети від кожного датчика в мережі. Це часто не так. Наприклад, сенсори сейсмічних та акустичних мереж як правило, мають відносно низьку швидкість передачі даних в той час як зображення і спектрометричні дані з них потрібно зібрати з високою роздільною здатністю, що вимагають високої швидкості передачі даних. У сенсорній мережі, в якій є гетерогенні датчики які вимагають різної швидкості передачі даних і обмеженої потужності та пропускної здатності,

алгоритми інтелектуальної сенсорної маршрутизації потрібні не тільки для координації існуючих датчиків для спілкуватися один з одним методами управління потужністю, ефективними протоколами, а й для знаходження і адаптування до видалення і додавання різних вузлів з високими і низькими рівнями передачі даних і зміна їх активності.

Рій базовані алгоритми маршрутизації [26-28] впливають з розуміння основних принципів роботи біологічних роїв, таких як мурахи чи бджоли. Ці рої, часто містять тисячі або десятки тисяч елементів, щодня виконують надзвичайно складні завдання глобальної оптимізації та розподілу ресурсів, використовуючи при цьому тільки локальну інформацію. Рій може виконувати такі складні завдання за рахунок інтелекту, що виходить з колективу всіх її елементів. Це в той час, як кожен такий окремий елемент має порівняно невеликий інтелект, не в змозі зрозуміти або змінити поведінку рою на глобальному або навіть часто широкому регіональному масштабі. Як приклад, поки ефективність пеленгації (маршрутизація) окремих мурах, здається дуже низькою, через випадкову поведінку, насправді ефективність маршрутизації колонії мурах надзвичайно висока, якщо судити по живучості видів, шляхом знаходження шляху до різних джерел їжі. Основними принципами, що регулюють ці рої, що діють дуже динамічно, випадково, і часто у ворожому середовищі є рутини і норма, а не виняток. Вони пропонують глибоке розуміння і керівництво в розробці алгоритмів, спрямованих на інтелектуальне управління системами з аналогічними вихідними характеристиками, такими як бездротові мережі зв'язку. Ройовий інтелект методів маршрутизації підвищує надійність та своєчасність передачі даних в гетерогенній багатовузловій мережі бездротового зв'язку. Вони будуть значною мірою сприяти досягненню мети надійного поширення покриття зв'язку в мережі. Вони крім того, зменшують накладні витрати для розширення мережі, зростання їх через функції масштабування.

## 1.4 Постановка задачі дослідження

Технології мобільних однорангових мереж набули популярності, оскільки вони мають здатність забезпечити миттєве рішення для бездротової мережі в місцях/ситуаціях, коли немає заздалегідь розподіленої інфраструктури. MANETs виявилась привабливим рішенням у широкому діапазоні середовищ від розсіяного військового розгортання до простої мережі групи ноутбуків на офісному приміщенні. У всіх випадках існує необхідність створення групи вузлів (ноутбуків, настільних комп'ютерів, КПК, стільникових телефонів і т.д.), щоб згрупувати і створити мережу, яка може запропонувати послуги, такі як спільне використання файлів, обмін повідомленнями, спільне використання ресурсів і т.д.

Тому основним завданням в мобільній мережі є формування ефективного одного або декількох маршрутів між двома вузлами, щоб вони могли надійно спілкуватися. Така мережа характеризується наступними проблемами.

1. Топологія мережі може динамічно змінюватися залежно від випадкового переміщення вузлів.

2. Будь-який вузол може залишити/приєднатися до мережі і протокол повинен адаптуватися відповідним чином.

3. Протокол повинен забезпечувати максимальну надійність пакета в мережі для даних умов.

З урахуванням цих факторів, ключовими параметрами при розробці протоколу маршрутизації є:

1) ефективна маршрутизація.

Це насамперед вимоги протоколу – для успішного виявлення та доставки пакета від джерела до пункту призначення. Деякі із заходів для ефективної маршрутизації включають співвідношення пакетної доставки, відсоток оптимальних маршрутів і середня з кінця в кінець затримка.

1) Запобігання перевантажень.

Сильно пов'язане з попереднім параметром, для того, щоб переконатись, що протокол маршрутизації не перевантажував конкретний маршрут/вузол, що може призвести до відкидання пакетів або навіть виходу із ладу вузлів.

## 2) Споживання енергії.

Більшість мобільних вузлів є пристрої з автономним живленням: ноутбуки, КПК та інші портативні пристрої, які мають сильну потребу в енергоспоживанні. Отже, алгоритм маршрутизації повинен мінімізувати енергоспоживання окремих вузлів.

## 3) Балансування навантаження.

Деякі вузли можуть бути стратегічно розташовані в результаті присутності в найбільш оптимальних маршрутах з'єднання. Такі вузли несправедливо перевантажуються, що веде до перевантаження мережі. Отже, існує необхідність балансування навантажень на окремі вузли, як компроміс для оптимальних маршрутів, що призводить до справедливого розподілу навантаження.

Управління мережею, такою як мережею мобільних телефонів є великою проблемою. Також змінний характер спеціальних мереж запобігає вибору централізованого рішення, яке може знайти оптимальний маршрут для передачі пакетів, і в той же час звести до мінімуму різні параметри, такі як навантаження і т.д. Також, жоден вузол не може взяти на себе роль централізованого управління через обмежені можливості енергії та обробки мобільних вузлів. Тому це призвело до необхідності розробки розподіленого підходу, який включає в себе обмежену обробку та потужність окремих вузлів, але які працюють узгоджено в напрямку маршрутизації та управління мережею.

Для спеціальних мереж були розроблені різні протоколи маршрутизації, деякі проактивні як DSDV та деякі за потребою, як AODV і DSR. Однак більшість з цих протоколів намагається вирішити одну задачу –

ефективна маршрутизація – і надати алгоритми для мінімальної затримка з кінця в кінець, максимальної пропускної здатності і т.д.

Отже, існує необхідність у розробці протоколу багато шляхової маршрутизації, що в результаті буде забезпечувати загальне управління мережею додатково до простої маршрутизації пакетів. Отже, перспективним напрямом вирішення задачі пошуку оптимальних маршрутів передачі даних в безпроводних мережах є використання алгоритмів мурашкових колоній.

#### Висновок до розділу

Проведений аналіз та класифікація протоколів маршрутизації в бездротових мережах показала, що на даному етапі їх розвитку існує досить велика кількість стратегій маршрутизації, але, незважаючи на різноманітність протоколів маршрутизації в Mesh-мережах, жоден з існуючих методів маршрутизації не забезпечує ефективної передачі даних з наданням необхідних параметрів якості обслуговування. У зв'язку з цим виникає завдання розробки та дослідження нових підходів рішення задачі маршрутизації в безпроводних комп'ютерних мережах, що підвищують ефективність передачі даних з забезпеченням різними додатками необхідного рівня якості обслуговування.



## 2 МЕТОД МАРШРУТИЗАЦІЇ ТА ОСНОВІ МУРАШКОВИХ АЛГОРИТМІВ

### 2.1 Принципи алгоритмів мурашкових колоній

Останніми роками інтенсивно розробляється науковий напрям з назвою «Природні обчислення» (Natural Computing), який об'єднує математичні методи, в яких закладені принципи природних механізмів ухвалення рішень [11]:

- Genetic Algorithms – генетичні алгоритми;
- Evolution Programming – еволюційне програмування;
- Neural Network Computing – нейромережеві обчислення;
- DNA Computing – ДНК обчислення;
- Cellular Automata – клітинні автомати;
- Ant Colony Algorithms – мурашкові алгоритми.

Ці механізми забезпечують ефективну адаптацію флори і фауни до довкілля впродовж декількох мільйонів років.

Серед так званих “Soft computing techniques”, розроблених за останні десять років для важко вирішуваних завдань дискретної оптимізації, відносяться:

- генетичні алгоритми (GAs) – ґрунтуються на природному відборі і генетиці;
- мурашині алгоритми (Ant Colony Optimization – ACO, Ant Systems – AS) – моделюють поведінку мурашника.

Імітація самоорганізації мурашиної колонії складає основу мурашиних алгоритмів оптимізації. Колонія мурашок може розглядатися як багатоагентна система, в якій кожен агент (мурашка) функціонує автономно по дуже простих правилах. На противагу майже примітивній поведінці агентів, поведінка всієї системи виходить на подив розумною.

Мурашині алгоритми серйозно досліджуються європейськими ученими з середини 90-х років. На сьогоднішній день вже отримані добрі результати

для оптимізації таких складних комбінаторних завдань, як завдання комівояжера, завдання оптимізації маршрутів вантажівок, завдання розфарбовування графа, квадратичне завдання про призначення, завдання оптимізації мережених графіків, завдання календарного планування і багато інших.

Особливо ефективні мурашині алгоритми при динамічній оптимізації процесів в розподілених нестационарних системах, наприклад, трафіків в телекомунікаційних мережах.

### 2.1.1 Біологічні принципи поведінки мурашиної колонії

Принципи поведінки мурашок витримали випробування далеко не в лабораторних умовах впродовж 100 мільйонів років – саме стільки часу тому мурашки «колонізували» Землю. Мурашки відносяться до соціальних комах, що живуть усередині деякого колективу – колонії. На Землі близько двох відсотків комах є соціальними, половину з них складають мурашки – невеликі істоти масою від 1 до 5 міліграма [26].

Число мурашок в одній колонії коливається від 30 штук до декількох мільйонів. На Землі близько  $10^{+E16}$  мурашок із загальною масою, приблизно рівній масі людства. Поведінка мурашок при транспортуванні їжі, подоланні перешкод, будівництві мурашника і інших діях частенько наближається до теоретично оптимальному. Як приклад на рисунку 2.1 приведена структура взаємозв'язаних гнізд суперколонії мурашок *Formica lugubris* в Швейцарії. Мережа мурашників близька до мінімального остовному дерева, що сполучає всі гнізда колонії – вершини графа на рисунку 2.1.

Основу «соціальної» поведінки мурашок складає самоорганізація — безліч динамічних механізмів, що забезпечують досягнення системою глобальної мети в результаті низькорівневої взаємодії її елементів.

Принциповою особливістю такої взаємодії є використання елементами системи лише локальної інформації. При цьому виключається будь-яке централізоване управління і звернення до глобального образу, що

репрезентує систему на зовнішньому світі. Самоорганізація є результатом взаємодії наступних чотирьох компонентів [27]:

- випадковість;
- багатократність;
- позитивний зворотний зв'язок;
- від'ємний зворотний зв'язок.

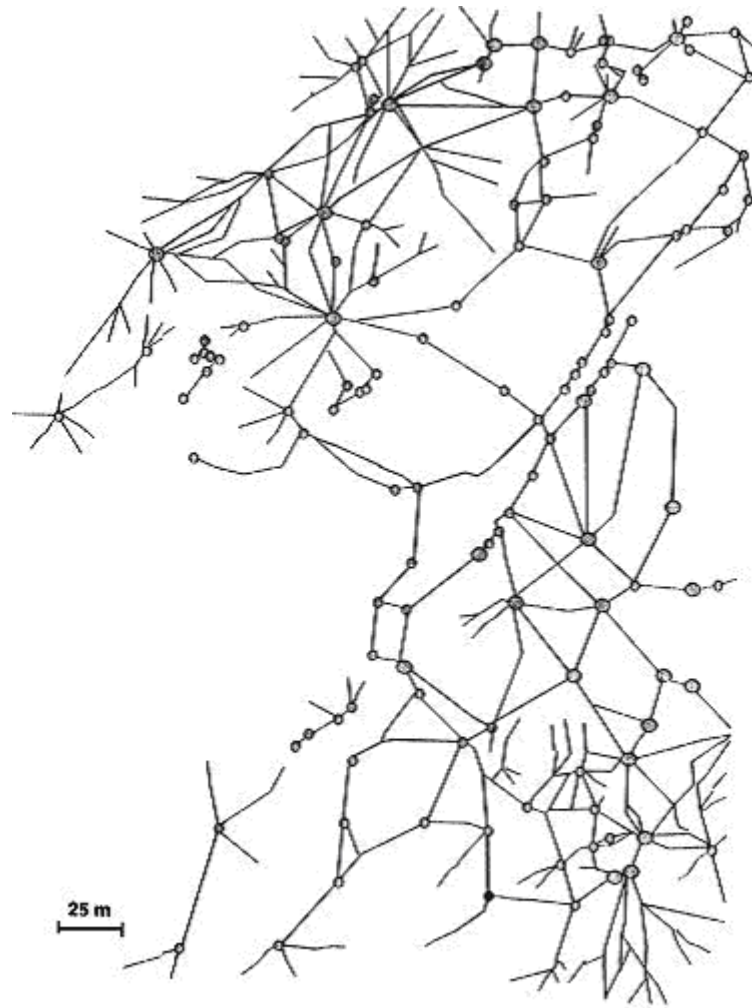


Рисунок 2.1 – Мережа гнізд суперколонії мурашок *Formica lugubris* в Швейцарії

Мурашки відносяться до соціальних комах, які створюють колективи. Колективна система здатна вирішувати складні динамічні завдання по виконанню спільної роботи, яка не могла б виконуватися кожним елементом системи окремо у всіляких середовищах без зовнішнього управління,

контролю або координації. У таких випадках говорять про ройовий інтелект (Swarm intelligence), як про хитромудрі способи кооперативної поведінки, тобто стратегії виживання.

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

Основу поведінки мурашиної колонії складає самоорганізація, що забезпечує досягнення загальних цілей колонії на основі низькорівневої взаємодії. Колонія не має централізованого управління, і її особливостями є обмін локальною інформацією лише 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 мурашки, не здатні відчувати близькість їжі.

Непрямий обмін – стігмержі (stigmergy), є рознесена в часі взаємодія, при якій одна особина змінює деяку область довкілля, а інші використовують цю інформацію пізніше, коли в неї потрапляють. Біологи встановили, що така відкладена взаємодія відбувається через спеціальну хімічну речовину – феромон (pheromone), секрет спеціальних залоз, що відкладається при переміщенні мурашки. Концентрація феромону на дорозі визначає перевагу руху по ньому.

Адаптивність поведінки реалізується випаром феромону, який в природі сприймається мурашками протягом декількох діб. Ми можемо провести деяку аналогію між розподілом феромону в просторі, що оточує колонію, і «глобальною» пам'яттю мурашника, що носить динамічний характер.

Мурашині алгоритми є імовірнісною жадібною евристикою, де вірогідність встановлюється, виходячи з інформації про якість рішення, отриманою з попередніх рішень. Вони можуть використовуватися як для статичних, так і для динамічних комбінаторних оптимізаційних завдань.

Збіжність гарантована, тобто в будь-якому разі ми отримаємо оптимальне рішення, проте швидкість збіжності невідома.

### 2.1.2 Історія створення мурашиних алгоритмів

Історія створення мурашиних алгоритмів почалося з вивчення поведінки реальних мурашок. Експерименти з аргентинськими мурашками, що проводяться Держсом в 1989 і Денеборгом в 1990 році послужили відправною крапкою для подальшого дослідження ройового інтелекту. Дослідження використання отриманих знань для дискретної математики почалися на початку 90-х років ХХ століття, автором ідеї є Марко Доріго з Університету Брюсселю, Бельгія. Саме він вперше зумів формалізувати поведінку мурашок і застосувати стратегію їх поведінки для вирішення завдання про найкоротші шляхи. Пізніше за участю Гамбарделлі, Тайлларда і Ді Каро були розроблені і багато інших підходів до вирішення складних оптимізаційних завдань за допомогою мурашиних алгоритмів. На сьогоднішній день ці методи є вельми конкурентоздатними в порівнянні з іншими евристичними і для деяких завдань дають найкращі на сьогоднішній день результати [26].

### 2.1.3 Концепція мурашиних алгоритмів

Ідея мурашиного алгоритму – моделювання поведінки мурашок, пов'язаної з їх здатністю швидко знаходити найкоротшу дорогу від мурашника до джерела їжі і адаптуватися до умов, що змінюються, знаходячи нову найкоротшу дорогу. При своєму русі мурашка мітить дорогу феромоном, і ця інформація використовується іншими мурашками для виборі шляху. Це елементарне правило поведінки і визначає здатність мурашок знаходити новий шлях, якщо старий виявляється недоступним [10, 26,].

Розглянемо приклад, показаний на рисунку 2.2, коли на оптимальній досі дорозі виникає перешкода. В цьому випадку необхідно визначити новий оптимальний шлях. Дійшовши до перешкоди, мурашки з рівною імовірністю

обходитимуть її справа і зліва. Те ж саме відбуватиметься і на зворотному боці перешкоди. Проте, ті мурашки, які випадково виберуть найкоротший шлях, будуть швидше за інших проходити, і за декілька пересувань він буде більш збагачений феромоном. Оскільки рух мурашок визначається концентрацією феромону, то наступні віддаватимуть перевагу саме цій дорозі, продовжуючи збагачувати його феромоном до тих пір, поки ця дорога з якої-небудь причини не стане недоступна.

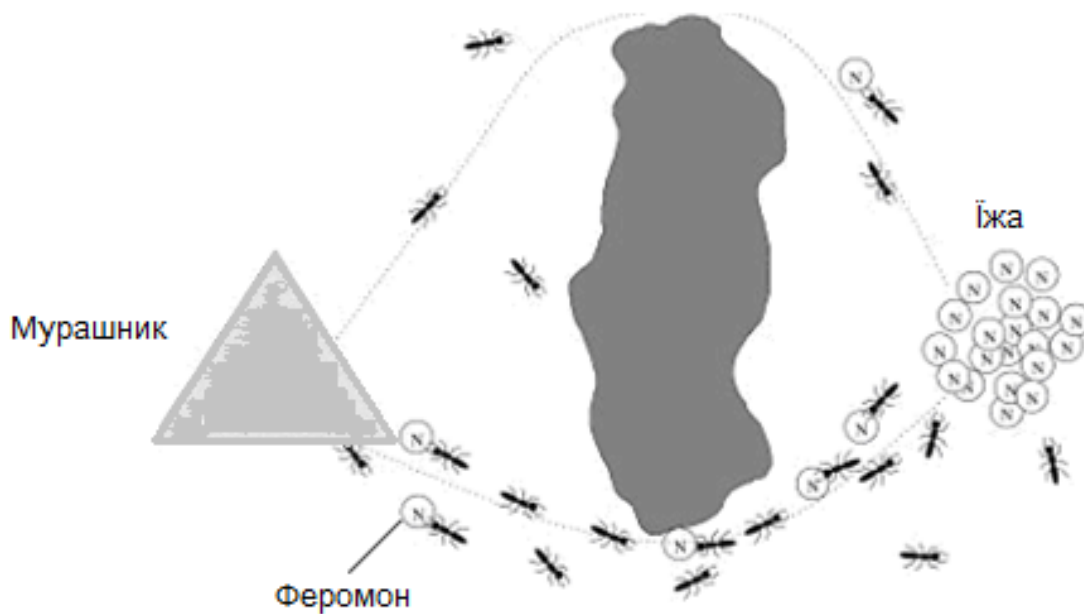


Рисунок 2.2 – Визначення нового оптимального шляху

Очевидний позитивний зворотний зв'язок швидко приведе до того, що найкоротша дорога стане єдиним маршрутом руху більшості мурашок. Моделювання випару феромону – негативного зворотного зв'язку – гарантує нам, що знайдене локальне оптимальне рішення не буде єдиним – мурашки шукатимуть і інші дороги. Якщо ми моделюємо процес такої поведінки на деякому графові, ребра якого є можливими шляхами переміщення мурашок, протягом певного часу, то найбільш збагачена

феромоном дорога по ребрах цього графа і буде рішенням задачі, отриманим за допомогою мурашиного алгоритму.

#### 2.1.4 Узагальнений мурашковий алгоритм

Будь-який мурашковий алгоритм, незалежно від модифікацій, представимо в наступному вигляді [10].

- Якщо умови не виконані.
  1. Створюємо мурашок.
  2. Шукаємо рішення.
  3. Оновлюємо феромон.
  4. Додаткові дії {опційно}.

Тепер розглянемо кожен крок в циклі детальніше.

1. Створення мурашок.

- Стартова точка, куди поміщається мурашка, залежить від обмежень, що накладаються умовами завдання. Тому що для кожного завдання спосіб розміщення мурашок є визначальний. Щоб всі вони поміщаються в одну точку, або в різних з повтореннями, або без повторень.

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

2. Пошук рішення.

- Імовірність переходу з вершини  $i$  у вершину  $j$  визначається по наступній формулі [26]

$$P_{ij}(t) = \frac{\tau_{ij}(t)^a \cdot \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{j \in \text{allowed nodes}} \tau_{ij}(t)^a \left(\frac{1}{d_{ij}}\right)^\beta},$$

де  $\tau_{ij}(t)$  – рівень феромону,  $d_{ij}(t)$  – евристична відстань, а  $\alpha, \beta$  – константні параметри.

При  $\alpha = 0$  вибір найближчого міста найбільш вірогідний, тобто алгоритм стає жадібним.

При  $\beta = 0$  вибір відбувається лише на підставі феромону, що наводить до субоптимальних рішень.

Тому необхідний компроміс між цими величинами, який знаходиться експериментально.

### 3. Оновлення феромону.

Рівень феромону оновлюється відповідно до приведеної формули

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{\substack{k \in \text{Colony that} \\ \text{used edge } (i,j)}} \frac{Q}{L_k},$$

де  $\rho$  – інтенсивність випаровування,  $L_k(t)$  – ціна поточного рішення для  $k$ -ї мурашки, а  $Q$  – параметр, що має значення порядку ціни оптимального рішення, тобто  $\frac{Q}{L_k(t)}$  – феромон, що відкладається  $k$ -ою мурашкою, що використовує ребро  $(i, j)$ .

### 4. Додаткові дії.

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

## 2.2 Алгоритм багатошляхової маршрутизації на основі алгоритму мурашкових колоній

Для того, щоб побудувати відповідний мурашковий алгоритм для вирішення якого-небудь завдання, потрібно.



1. Представити завдання у вигляді набору компонент і переходів або набором неорієнтованих зважених графів, на яких мурашки 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 алгоритм мурашкових колоній і застосувати для вирішення завдання.

6. Налаштувати параметри алгоритму мурашкових колоній.

Визначати також необхідно:

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

2.2.1 Алгоритмів мурашкових колоній для пошуку незалежних маршрутів

Завдання формулюється як завдання пошуку мінімального за вартістю маршруту по всіх вершинах без повторень на повному зваженому графові з  $n$  вершинами. Змістовно вершини графа є вузлами, які можуть бути джерелами інформації, а ваги ребер відображають відстані (довжини) або вартість шляху. Ця задача є NP - трудною, і точний алгоритм перебору її рішення має факторіальну складність.

Моделювання поведінки мурашок пов'язане з розподілом феромону на стежці – ребрі графа в задачі пошуку маршруту. При цьому вірогідність включення ребра в маршрут окремої мурашки пропорційний кількості феромону на цьому ребрі, а кількість феромону, що відкладається, пропорційна довжині маршруту. Чим коротше маршрут, тим більше феромону буде відкладено на його ребрах, отже, більша кількість мурашок включатиме його в синтез власних маршрутів. Моделювання такого підходу,

що використовує лише позитивний зворотний зв'язок, наводить до передчасної збіжності – більшість мурашок рухаються по локально оптимальному маршруту. Уникнути цього можна, моделюючи негативний зворотний зв'язок у вигляді випару феромону. При цьому якщо феромон випаровується швидко, то це приводить до втрати пам'яті колонії і забуває хороших рішень, з іншого боку, великий час випару може привести до здобуття стійкого локального оптимального рішення.

Тепер з врахуванням особливостей завдання, ми можемо описати локальні правила поведінки мурашок при виборі маршруту.

1. Мурашки мають власну «пам'ять». Оскільки кожний вузол може бути відвіданий лише один раз, то у кожної мурашки є список вже відвіданих вузлів – список заборон. Позначимо через  $J_{i,k}$  список вузлів, які необхідно відвідати мурашці, що знаходиться в місті  $i$ .

2. Мурашки володіють «зором» – видимість є евристичне бажання відвідати місто  $j$ , якщо мурашка знаходиться в місті  $i$ . Вважатимемо, що видимість обернено пропорційна відстані між містами

$$\eta_{ij} = 1/D_{ij} .$$

3. Мурашки володіють «нюхом» – вони можуть уловлювати слід феромону, підтверджуючий бажання відвідати місто  $j$  з міста  $i$  на основі досвіду інших мурашок. Кількість феромону на ребрі  $(i, j)$  у момент часу  $t$  позначимо через  $\tau_{ij}$ .

4. На цій основі ми можемо сформулювати імовірно-пропорційне правило, що визначає імовірність переходу  $k$  - тої мурашки з міста  $i$  в місто  $j$  [11]:

$$\begin{cases} P_{ij,k}(t) = \frac{\tau_{ij}(t)^{\bar{\alpha}} \cdot \tau_{ij}^{\bar{\beta}}}{\sum_{l \in J_{i,k}} \tau_{ij}(t)^{\bar{\alpha}} \cdot \tau_{ij}^{\bar{\beta}}}, j \in J_{i,k}, \\ P_{ij,k}(t) = 0, j \notin J_{i,k} \end{cases} \quad (2.1)$$

де  $\alpha, \beta$  – параметри, які задають вагу сліду феромону. При  $\alpha=0$  алгоритм вироджується до жадібного алгоритму (буде вибрано найближчий вузол). Відмітимо, що вибір вузла є імовірнісним, правило (1) лише визначає ширину зони вузла  $j$ ; у загальну зону всіх вузлів  $J_{i,k}$  кидається випадкове число, яке і визначає вибір мурашки. Правило (1) не змінюється в ході алгоритму, але у двох різних мурашок значення ймовірності переходу відрізнятимуться, оскільки вони мають різний список дозволених міст.

5. Пройшовши ребро  $(i, j)$ , мурашка відкладає на ньому деяку кількість феромону, яка має бути пов'язана з оптимальністю зробленого вибору. Нехай  $T_k(t)$  є маршрут, пройдений мурашкою до  $k$  моменту часу  $t$ ,  $L_k(t)$  – довжина цього маршруту, а  $Q$  – параметр, що має значення порядку довжини оптимальної дороги.

Отже, кількість феромону, що відкладається, може бути задана у вигляді

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

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

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij}(t); \Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{i,j}(t), \quad (2.2)$$

де  $m$  – кількість мурашок в колонії.

На початку алгоритму кількості феромону на ребрах приймається рівним невеликому додатному числу. Загальна кількість мурашок залишається постійною і рівною кількості вузлів, кожна мурашка починає маршрут зі свого вузла.

Додаткова модифікація алгоритму може полягати у веденні так званих «елітних» мурашок, 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  $e$  елітних мурашок, то ребра маршруту отримають додаткову кількість феромону [10]

$$\Delta\tau_e = e \cdot Q / L^* \quad (2.3)$$

На основі описаної вище сукупності кроків автором розроблений алгоритм пошуку оптимальних шляхів при реалізації багатошляхової маршрутизації.

У запропонованому алгоритмі можна самостійно в довільному порядку розмістити вузли мережі або сформувати їх псевдовипадковим чином. Відстань між сусідніми вершинами мережі визначається:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

причому вершини вважаються сусідніми, якщо відстань  $D$  між ними  $D \leq R$ , де  $R$  – радіус зв'язку вузла мережі (табл. 2.1).

Таблиця 2.1 – Структура таблиці відстаней між вузлами

$D_{11}$	$D_{12}$	...	$D_{1n}$
$D_{21}$	$D_{22}$	...	$D_{2n}$
...	...	...	...
$D_{n1}$	$D_{n2}$	...	$D_{nn}$

На основі відстані  $D_{ij}$  визначається видимість і початкова концентрація феромона.

2.2.2 Алгоритм мурашкових колоній для задачі пошуку маршруту передачі

1. Введення матриці відстаней  $D$ .
2. Ініціалізація параметрів алгоритму –  $\alpha, \beta, e, Q$ .
3. Ініціалізація ребер – присвоєння видимості  $\eta_{ij}$  і початкової концентрації феромону.
4. Розміщення мурашок у випадково вибрані вузли без повторень.
5. Вибір початкового найкоротшого маршруту і визначення  $L^*$ .
- //Основний цикл
6. Цикл за часом життя колонії  $t = 1, t_{\max}$ .
7. Цикл по всіх мурашках  $k=1, m$ .
8. Побудувати маршрут  $T_k(t)$  за правилом (2.1) і розрахувати довжину  $L_k(t)$ .
9. Кінець циклу по мурашках.
10. Перевірка всіх  $L_k(t)$  на краще рішення в порівнянні з  $L^*$ .
11. У випадку якщо рішення краще, відновити  $L^*$  і  $T^*$ .
12. Цикл по всіх ребрах графа.
13. Відновити сліди феромону на ребрі по правилах (2.2) і (2.3).
14. Кінець циклу по ребрах.
15. Кінець циклу за часом.

16. Вивести найкоротший маршрут  $T^*$  і його довжину  $L^*$ .

Складність даного алгоритму, як неважко відмітити, залежить від часу життя колонії, кількості вузлів ( $n$ ) і кількості мурашок в колонії ( $m$ ).

### 2.2.3 Можливі модифікації мурашиних алгоритмів

Оскільки в основі мурашиного алгоритму лежить моделювання пересування мурашок по деяких дорогах, то такий підхід може стати ефективним способом пошуку раціональних рішень для завдань оптимізації, що допускають інтерпретацію з допомогою графів. Ряд експериментів показує, що ефективність мурашиних алгоритмів зростає із зростанням розмірності вирішуваних завдань оптимізації. Добрі результати виходять для нестационарних систем із змінними в часі параметрами, наприклад, для розрахунків телекомунікаційних і комп'ютерних мереж. У Інтернеті можна знайти опис використання мурашиного алгоритму для розробки оптимальної структури знімальних мереж GPS в рамках створення високоточних геодезичних і знімальних технологій.

Якість отримуваних рішень багато в чому залежить від вибраних параметрів в імовірно-пропорційному правилі вибору шляху на основі поточної кількості феромону і від параметрів правил відкладання і випару феромону. Можливо, що динамічне адаптаційне налаштування цих параметрів може сприяти здобуттю кращих рішень. Важливу роль грає і початковий розподіл феромону, а також вибір умовно оптимального рішення на кроці ініціалізації.

Перспективними шляхами поліпшення мурашиних алгоритмів є різні адаптації параметрів з використанням бази нечітких правил і їх гібридизація, наприклад, з генетичними алгоритмами. Як варіант, така гібридизація може полягати в обміні через певні проміжки часу поточними найкращими рішеннями.

## 2.3 Область використання мурашкових алгоритмів оптимізації

Приведений мурашковий алгоритм оптимізації пошуку маршруту після незначних модифікацій може використовуватися для вирішення різних комбінаторних завдань: квадратичного завдання про призначення (Quadratic Assignment Problem), завдання про оптимізацію маршрутів вантажівок (Vehicle Routing Problem), завдання календарного планерування (JobShop Schedule Planing), завдання розфарбовування графа (Graph Coloring Problem) і ін.

Мурашкові алгоритми знаходять вирішення дискретних завдань оптимізації не гірше інших загальних метаевристичних технологій і деяких проблемно-орієнтованих методів.

Високу ефективність мурашині алгоритми демонструють при оптимізації розподілених нестационарних систем. Яскравим прикладом може служити знаходження мурашиними алгоритмами оптимальних трафіків в телекомунікаційних мережах [1, 7]. Як приклад в таблиці 2.3 приведені результати маршрутизації в американській мережі NSFNET, що містить 14 вузлів з 21 двонаправленою лінією зв'язку.

Порівнювалися наступні алгоритми: AntNet – мурашиний алгоритм; OSRF – офіційний алгоритм маршрутизації мережі Інтернет; SRF – алгоритм, що використовує динамічну метрику при оцінці вартості з'єднань; Даємон – апроксимація ідеального алгоритму маршрутизації; BF – алгоритм Беллмана-Форда. У таблиці приведені середні значення часу затримки і пропускної спроможності при інтенсивному завантаженні мережі. Числа в дужках – значення середньо квадратичних відхилень при 10 кратному прогоні алгоритмів.

Мурашині алгоритми засновані на імітації самоорганізації соціальних комах за допомогою використання динамічних механізмів, за допомогою яких система досягає глобальної мети в результаті локальної низькорівневої взаємодії елементів. У роботі на прикладі завдання маршрутизації показано,

як в алгоритми вирішення дискретних завдань оптимізації упровадити складові самоорганізації мурашок: випадковість, багатократність взаємодії, негативну і позитивну зворотні зв'язки. Проведені комп'ютерні експерименти показують, що мурашині алгоритми знаходять хороші маршрути значно швидше, ніж точні методи комбінаторної оптимізації. Ефективність мурашиних алгоритмів збільшується із зростанням розмірності завдання оптимізації.

Мурашині алгоритми забезпечують рішення і інших комбінаторних завдань не гірше за загальні метаевристичних технології оптимізації і деяких проблемно-орієнтованих методів. Особливо добрі результати мурашиної оптимізації виходять для нестационарних систем, параметри яких змінюються в часі, наприклад телекомунікаційних і комп'ютерних мереж.

Важливою властивістю мурашиних алгоритмів є неконвергентність: навіть після великого числа ітерацій одночасно досліджується безліч варіантів рішення, унаслідок чого не відбувається тривалих часових затримок в локальних екстремумах. Все це дозволяє рекомендувати використання мурашиних алгоритмів для вирішення складних комбінаторних завдань оптимізації. Перспективними шляхами поліпшення мурашиних алгоритмів є online адаптація параметрів за допомогою бази нечітких правил, а також їх гібридизація з іншими методами природних обчислень, наприклад генетичними алгоритмами [11].

Таблиця 2.2 – Порівняння евристичних методів оптимізації пошуку маршруту

Тестова задача	Eil 50	Eil 75	KroA 100
Імітаційне випалювання	443	580	немає даних
Генетичні алгоритми	428	545	22761
Еволюційне програмування	426	542	немає даних
Мурашині алгоритми	425	535	21282



Таблиця 2.3 – Порівняння алгоритмів маршрутизації

Алгоритм	AntNet	OSRF	SRF	Daemon	BF
Середня затримка передачі, с	0.93 (0.2)	5.85 (1.43)	3.58 (0.83)	0.10 (0.03)	4.27 (1.22)
Пропускна здатність, біт/с x10	2.392 (0.011)	2.1 (0.002)	2.284 (0.003)	2.403 (0.010)	1.41 (0.047)

Гібридизація може здійснюватися за острівною схемою, коли різні алгоритми вирішують задачу паралельно і автономно (кожен на окремому «острові») з обміном найкращими рішеннями через певний час, або за принципом «мастер–підмастер», коли основний алгоритм — «майстер» передає вирішення типових підзадач «підмайстрові» — спеціалізованому, швидкому алгоритму.

#### 2.4 Метрики оцінки вартості маршруту передачі даних

При оцінці якості алгоритмів маршрутизації необхідно враховувати наступні критерії [5].

1. Ефективність. Алгоритм повинен відправляти пакети по «найкращих» маршрутах. Алгоритм називається оптимальним, якщо він вибирає «найкращі» маршрути.

2. Коректність. Алгоритм повинен доставляти кожен пакет, що надійшов в мережу, в точності за призначенням.

3. Складність. Алгоритм обчислення таблиць повинен використовувати якомога менше повідомлень, економно витратити час і пам'ять.

4. Стійкість. У разі зміни топології (додавання або видалення вузла) алгоритм вносить зміни в таблиці маршрутизації.

5. Адаптивність. Алгоритм вибирає маршрути, найменш завантажені.

6. Справедливість. Алгоритм повинен обслуговувати всіх користувачів в рівній мірі.

В якості метрик визначення оптимального шляху використовується:

1. Мінімальна кількість ланок. Вартість використання маршруту визначається кількістю ланок (пройдених каналів або кроків від однієї вершини до іншої) у цьому маршруту. Алгоритм маршрутизації з мінімальним числом ланок вибирає шляхи, які мають найменше число ланок.

2. Мінімальна затримка. Кожному каналу зв'язку надається динамічно змінювана характеристика, що залежить від трафіку через цей канал. Алгоритм мінімальної затримки постійно переглядає таблиці так, щоб завжди вибиралися шляхи з (майже) мінімальною загальною затримкою.

3. Заряд батареї. Для ретрансляції повідомлень вибираються вузли з великим зарядом батареї.

4. Найкоротший шлях. Кожному каналу зв'язку надається незмінний (невід'ємні) вага, і вартість шляху покладається рівною сумарній вазі всіх каналів маршруту. Алгоритм найкоротшого маршруту вибирає маршрут найменшої вартості.

5. Максимальна пропускна здатність. При виборі маршруту оцінюється пропускна здатність каналу.

6. Ймовірність помилки. Для передачі даних вибирається найбільш надійний канал.

Як видно з наведених метрик, оптимальний маршрут оцінюється не одним критерієм або показником якості, а сукупністю таких критеріїв, причому вони є однаково значущими. Іноді можна знайти рішення, яке оптимально за всіма критеріями. Однак набагато частіше виникає зворотна ситуація, коли критерії не узгоджуються один з одним. У таких випадках рішенням є компромісом за різними критеріями.

Враховуючи обмежені ресурси бездротових вузлів, актуальним завданням при виборі методу маршрутизації є оцінка складності алгоритму

пошуку оптимальних шляхів. Для порівняння зазначених алгоритмів використовують такі параметри [5]:

- складність за кількістю обміну повідомленнями. Це загальне число повідомлень, які були відправлені для виконання алгоритму;
- бітова складність. Кількість бітів для кодування безлічі різних повідомлень;
- часова складність. Час необхідний для проведення обчислень;
- складність за обсягом пам'яті. Кількість одиниць часу необхідне окремому вузлу для виконання цього алгоритму.

У таблиці 1.1 представлена часова складність основних алгоритмів пошуку найкоротшого шляху: де  $n$  – кількість вузлів;  $m$  – кількість ребер,  $m_1$  – кількість мурашок в колонії,  $t_{\max}$  – час життя колонії [6].

Таблиця 2.4 – Алгоритми пошуку найкоротшого маршруту та оцінка їх складності

Алгоритм	Часова складність
Флойда - Уоршелла	$O(n^3 / \log n)$
Белмана-Форда	$O(n \cdot m)$
Дейкстрі	$O(n^2 + m)$
Джонсона	$O(n^2 \cdot \lg n + n \cdot m)$
Мурашкових колоній	$O(t_{\max}, n^2, m_1)$

З наведених алгоритмів пошуку оптимального маршруту особливо перспективні алгоритми мурашкових колоній, тому, що їх можна використовувати для вирішення не тільки статичних, але і динамічних задач, наприклад для розробки алгоритмів маршрутизації в безпроводних мобільних мережах [6].

## Висновки до розділу

В другому розділі розкрито принципи алгоритмів мурашкових колоній, основу поведінки яких складає самоорганізація, що забезпечує досягнення загальних цілей колонії на основі низькорівневої взаємодії.

Виділено ефективні області застосування мурашкових алгоритмів оптимізації, які особливо ефективні при динамічній оптимізації процесів в розподілених нестационарних системах, наприклад, трафіків в телекомунікаційних мережах.

Розроблено алгоритм пошуку оптимального маршруту на основі алгоритму мурашкових колоній, який забезпечує пошук незалежних маршрутів передачі від джерела до одержувача.

Проведено аналіз метрик оцінки вартості маршруту передачі даних та якості алгоритмів маршрутизації. При оцінці якості алгоритмів маршрутизації необхідно враховувати наступні критерії: ефективність, коректність, складність, стійкість, адаптивність.

Оцінка складності алгоритмів пошуку найкоротшого маршруту показала переваги алгоритмів мурашкових колоній а також їх перспективність застосування для вирішення не тільки статичних, але і динамічних задач, наприклад для розробки алгоритмів маршрутизації в безпроводних мобільних мережах.

## 3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПРОТОКОЛУ МАРШРУТИЗАЦІЇ НА ОСНОВІ МУРАШКОВИХ АЛГОРИТМІВ

### 3.1 Порівняння протоколів маршрутизації

Проведемо детальне порівняння вищевказаних протоколів шляхом імітаційного моделювання їх роботи в різних умовах, сукупність яких прийнято називати сценаріями роботи мережі. На підставі аналізу отриманих результатів планується внести в найбільш перспективні протоколи доопрацювання, які підвищать їх ефективність, або ж створити на їх основі новий протокол. Нижче як приклад наведено результати одного подібного експерименту, проведеного в симуляторі «QualNet Developer 4.5»[6].

Симулятори моделюють роботу всіх рівнів базової еталонної моделі взаємодії відкритих систем (EMBBC). Для мереж радіозв'язку присутня модель радіоканалу, що враховує поширення радіохвиль в використовуваному частотному діапазоні, а також моделі, що описують простір. Сукупність перерахованих вище властивостей симулятора визначає адекватність проведених на ньому експериментів і достовірність одержуваних при цьому результатів.

Вимірювалася залежність коефіцієнта доставки пакетів (відношення числа прийнятих пакетів з даними до числа переданих) від розміру мережі для протоколів AODV, DSR, LANMAR, OLSR, OSPFv2, ZRP в умовах наступного сценарію:

- модель каналного рівня - IEEE 802.11b (Wi-Fi), з розподіленою функцією управління (MAC DCF) і жорстко зафіксованою пропускнуою здатністю 2 Мб / с;
- робочий діапазон - 2,4 ГГц;
- потужність передавачів кожного вузла - 15 дБм;
- чутливість приймачів кожного вузла - (-89) дБм;

- тип антени - ненаправлена, з коефіцієнтом посилення 0 дБ, ефективністю 0,8 і втратами в фідері 0,5 дБ, висота підвісу 1,5 м;
- модель поширення сигналу - двопротенева;
- розміри мережі - 16, 25, 64 і 100 вузлів відповідно;
- вузли мережі переміщуються зі швидкістю 16 м / с на плоскій поверхні, обмеженої квадратами з сторонами 1,4, 1,6, 2,3 і 2,8 км відповідно, з паузами по 4 с, після чого випадковим чином змінюють напрямок руху;
- кількість активних сполук в мережі - 4;
- тип трафіку - CBR (Constant Bit Rate) з розміром пакета 512 байт та інтенсивністю відправки 10 пакетів в секунду;
- протокол транспортного рівня - UDP;
- час роботи мережі - 300 с.

Результати моделювання, усереднені по всіх чотирьох з'єднаннях і трьох експериментів, наведено на рисунку 3.1.

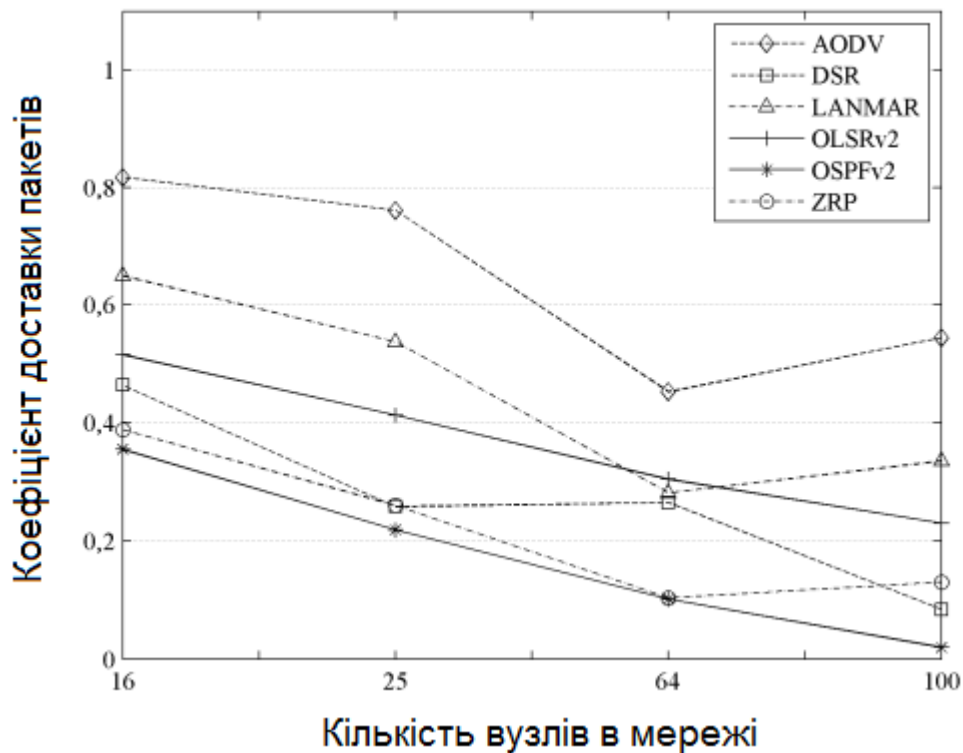


Рисунок 3.1 – Результати імітаційного моделювання

Наведений приклад демонструє можливості симуляторів як в плані завдання вхідних умов експерименту, так і в плані збору та представлення його результатів.

Представлений графік дозволяє кількісно оцінити роботу протоколів, наприклад, переконатися в тому, що класичний OSPF показує більш низькі результати в умовах мобільних Ad hoc - мереж, ніж спочатку розроблені для них протоколи маршрутизації.

### 3.2 Пошук маршруту передачі на основі алгоритму мурашкових колоній

Нижче приведений розроблений алгоритм мурашкових колоній для пошуку оптимального маршруту передачі даних, що реалізовує всі ідеї попереднього розділу (у дужках міститься словесний опис відповідного коду).

<Ввід матриці відстаней – D>

<Ініціалізація параметрів алгоритму  $\alpha$  (alpha),  $\beta$  (beta),  $e$ ,  $Q$ ,  $\tau_0$  (tau0)>

m=n % Кількість мурашок дорівнює кількості пунктів

For I=1:n % Для кожного ребра

    For j=1:n

        If i=j

            eta (i, j)=1/D (i, j); % Видимість

            tau (i, j) = tau0; % Феромон

        Else tau (i, j) = 0;

% Перехід із одного пункту в той самий неможливий

    End

End

End

For k=1:m

```

<Розмістити мурашок у випадково вибраній точці>
End
<Вибрати умовно найкоротший маршрут  $T^*$  і розрахувати його довжину  $L^*$ >
% Основний цикл
For t=1:tmax
% tmax – кількість ітерацій алгоритму
    For k=1:m % Для кожної мурашки
< Побудувати маршрут  $T_k(t)$  по правилу (2.1) розрахувати довжину  $L_k(t)$ >
        End
    End
If <"Краще рішення знайдено?">
<Обновити  $T^*$  і  $L^*$ >
    End
For i=1:n
For j=1:n % Для кожного ребра
<Обновити сліди феромона по правилах (2.2) і (2.3)>
    End
    End
    End
<Вивести найкоротший маршрут  $T^*$  і його довжину  $L^*$ >.

```

Алгоритм мурашководної колонії був запрограмований і протестований на мережі, структура якої представлена на рисунку 3.2. Для мережі з 50 вузлами алгоритм без елітних мурашок після 100 ітерацій знайшов оптимальний маршрут завдовжки 411 лише в одному випадку з десяти. Рішення можна поліпшити простим збільшенням кількості ітерацій. Довжини маршрутів мурашок на одній ітерації дещо відрізняються, тому для прискорення знаходження оптимуму необхідно штучно підсилювати найкращі поточні рішення за допомогою елітних мурашок. На рисунку 3.3 показані усереднені динаміки рішення задачі алгоритмами з різним числом мурашок.



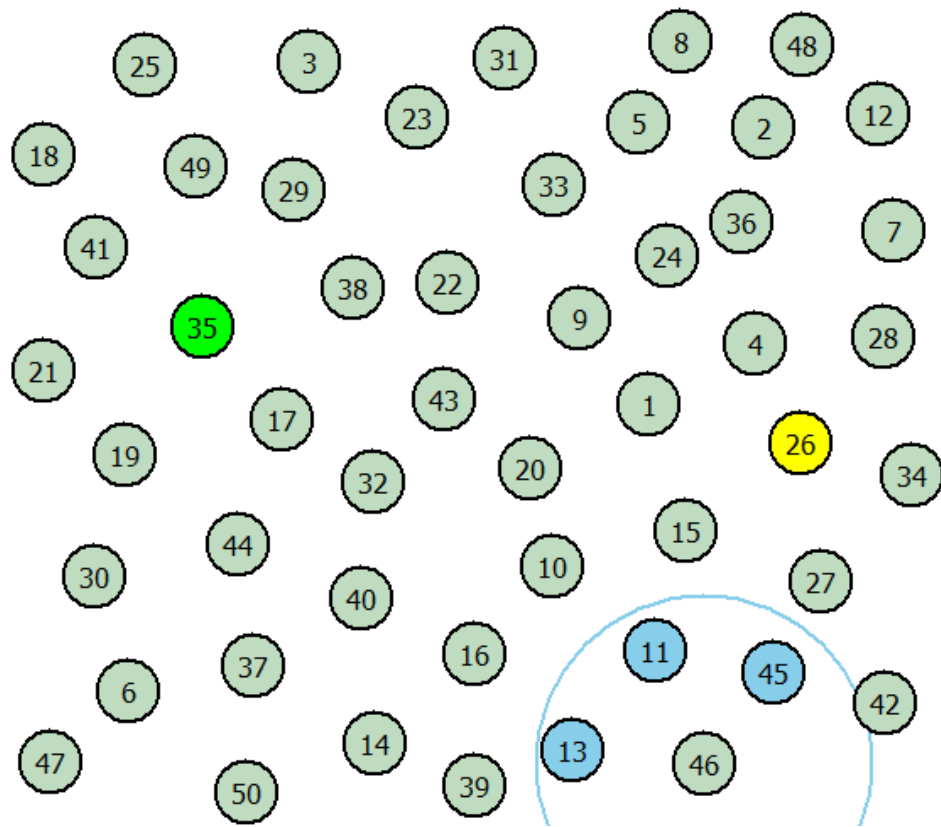


Рисунок 3.2 – Структура досліджуваної мережі

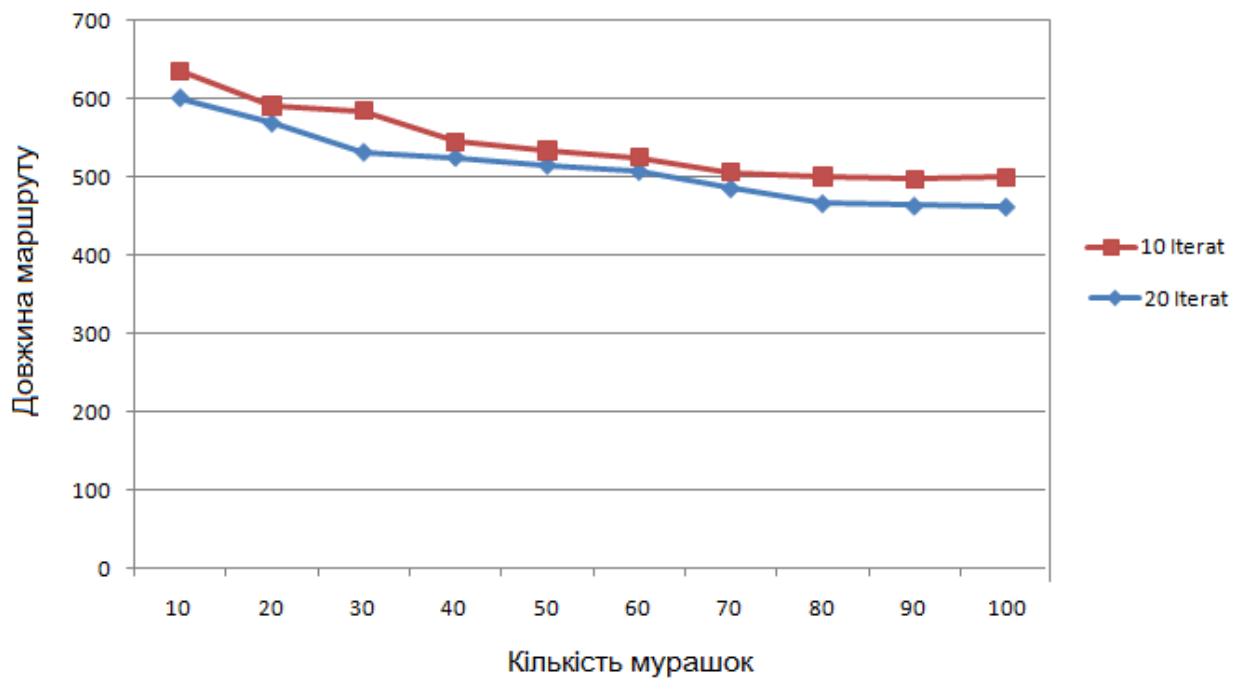


Рисунок 3.3 – Залежність довжини знайденого маршруту від кількості мурашок

Як видно з рисунку 3.3 точність знайденого маршруту підвищується при збільшенні числа ітерацій та кількості мурашок.

У десяти експериментах за 100 ітерацій алгоритм з п'яти елітними мурашками тричі знайшов оптимальний маршрут, з десятьма елітними мурашками – п'ять разів. Не дивлячись на те, що алгоритм з п'яти елітними мурашками повільніше наближається до хороших маршрутів, він набагато швидше проходить субоптимальні рішення пастки.

Використання елітних мурашок дозволяє покращити результати обчислення (рисунок 3.4).

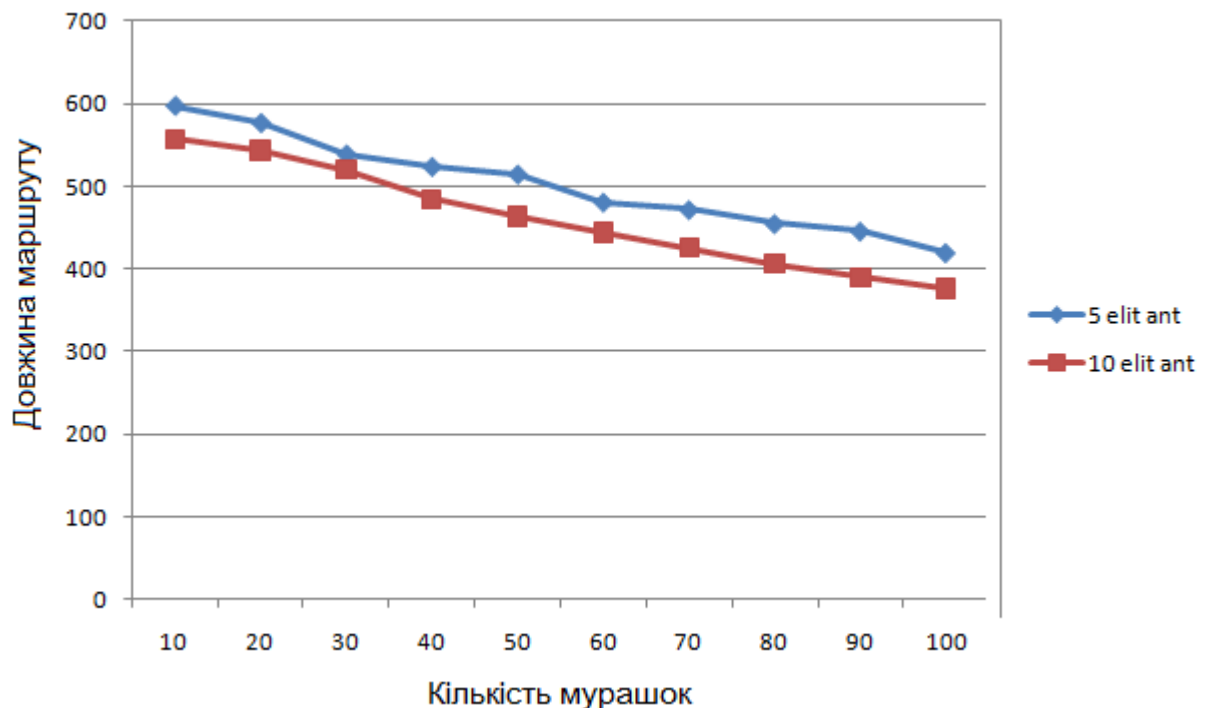


Рисунок 3.4 – Залежність довжини знайденого маршруту від кількості елітних мурашок

Експерименти для кожного алгоритму повторювалися 10 разів. На перший погляд здається, що чим більше елітних мурашок, тим краще працює алгоритм. Дійсно, алгоритми з великою кількістю елітних мурашок краще знаходять субоптимальні маршрути, проте після цього надовго застряють в

локальному оптимумі, оскільки елітні мурашки дуже сильно підсилюють ці субоптимальні рішення.

Проведені експерименти свідчать, що популяція рішень ніколи не вироджується до одного, загального для всіх мурашок маршруту. Навпаки, алгоритм продовжує синтезувати нові, інколи кращі рішення.

В порівнянні з точними методами, наприклад динамічним програмуванням або методом гілок і кордонів, мурашиний алгоритм знаходить близькі до оптимуму рішення за значно менший час навіть для завдань невеликої розмірності ( $n \geq 20$ ). Час оптимізації мурашиним алгоритмом є поліноміальною функцією від розмірності  $O(t, n * n, m)$ , тоді як для точних методів залежність експоненціальна.

### 3.3 Розробка програмного забезпечення пошуку маршруту в безпроводній сенсорній мережі

На основі описаних в розділі 2 послідовності кроків пошуку оптимального маршруту передачі даних на основі алгоритму мурашкових колоній розроблено відповідне програмне забезпечення.

Програма розроблена в середовищі Embarcadero RAD Studio XE2 на мові програмування Delphi. Embarcadero RAD Studio XE2 являє собою повнофункціональний пакет для швидкої візуальної розробки кросплатформених програм, що інтенсивно працюють з даними, для Windows, Mac OS X, .NET, PHP, веб-рішень і мобільних пристроїв. Delphi – ітеративна, структурована, об'єктно-орієнтована мова програмування, діалект Object Pascal.

Опишемо тільки основну функцію програми для знаходження найкоротшої відстані між точками за допомогою мурашкових алгоритмів (не будемо вдаватися до опису таких функцій як розставлення точок, заповнення таблиць і т.д.).

Після заповнення ввідних параметрів (розставлення точок, вводу кількості ітерацій і т.д.) і натиснення кнопки «Шукати» програма починає свою основну роботу для пошуку найкоротших відстаней за допомогою мурашкових алгоритмів. Після натиснення кнопки «Шукати» програма спершу перевіряє коректність введених даних і повідомляє про їх некоректне введення, якщо таке наявне.

Блок схему основної функції роботи програми наведено в додатку А. Отже всі необхідні поля і точки заповнені правильно. Перед початком своєї роботи програма ініціалізує початкові змінні введеними даними і засікає час початку своєї роботи. Як можна побачити з додатку А основне тіло роботи цієї функції складається з 3-х циклів. Оскільки обчислення відбуваються в тому ж потоці, який реагує на запити користувача, то програма на час роботи головної функції «підвисає», а щоб можна було орієнтуватися скільки програмі ще залишилось до кінця її роботи, то виводиться номер самої верхньої ітерації циклу (в лівому верхньому куті монітора), яка в даний момент обчислюється. В двох наступних циклах відбувається вибір переходу з точки в точку (функція, що реалізує цей вибір наведена в додатку Б). Після вибору точки виконується перевірка, на те чи зайшла дана мурашка в «тупік» (мурашка зайшла в «тупік», якщо в радіусі видимості немає точок, які вона могла б ще відвідати) або в кінцеву точку. Якщо мурашка зайшла в «тупік» або в кінцеву точку, то цикл переривається і виконується пошук шляху наступною мурашкою, інакше продовжується пошук поточною мурашкою. По закінченню роботи двох нижніх циклів добавляються нові знайдені рішення в таблиці, виконується їх сортування по зростанню знайдених шляхів та зафіксується ітерація на якій була знайдена коротша відстань від попередніх ітерації. Кількість феромону яка відкладається на кожній верхній ітерації обчислюється за формулою:

– для всіх знайдених маршрутів:

Феромон  $_{i,j,k} = 1 / \text{відстань маршруту } k$ ;

– для найкоротшого знайденого маршруту:

Феромон  $\tau_{i,j,k} = e * 1 / \text{відстань найкоротшого маршруту}$ ;

– для ребер які не ввійшли ні до одного із знайдених маршрутів:

Феромон  $\tau_{i,j,k} = 1 / \text{відстань найдовшого маршруту} / 1.5$ ;

де  $i$  та  $j$  є ребром в маршруті  $k$ ,  $e$  це – кількість елітних мурашок.

По закінченню всіх ітерацій заміряється час закінчення, від якого віднімається час початку і виводиться час роботи цієї функції в заголовку з назвою програми, а № ітерації на якій була знайдена найкоротша відстань серед всіх знайдених виводиться в вікні, яке містить таблиці результатів в заголовку з назвою вікна. На цьому робота основної функції програми закінчена.

Тепер опишемо функцію, яка вибирає точку в яку перейде мурашка (блок схема цієї функції наведена в додатку Б). Ця функція викликається із основної функції. З самого початку ця функція перевіряє чи залишилися ще не відвідані точки, які входять в радіус видимості мурашки. Якщо таких точок не залишилося, то робиться відмітка про те, що дана мурашка зайшла в «тупік» і функція повертає управління головній функції. Інакше заповнюється масив ймовірності переходу з точки  $i$  в точку  $j$  за наступною формулою:

$$\begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_{i,k}} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, j \in J_{i,k}; \\ P_{ij,k}(t) = 0, j \notin J_{i,k}; \end{cases}$$

де  $\alpha, \beta$  – параметри, що задають вагу феромону.  $k$  – мурашка. При  $\alpha = 0$  алгоритм, вироджується до жадібного алгоритму (буде обрано найближчий вузол),  $\eta_{ij} = 1 / D_{ij}$ , де  $D_{ij}$  – відстань між ребром  $i$  та  $j$ ,  $t$  – час життя колонії мурашок ( ітерація верхнього циклу).

Тепер коли відомі ймовірності переходу з даної точки в інші точки, залишилось перейти в якусь з цих точок. Вибір точки в даній функції в яку перейде мурашка здійснюється за принципом «Колеса рулетки», де кожна точка на ній має свій сектор з площею, пропорційною ймовірності, для вибору точки потрібно згенерувати випадкове число (кинути кульку на рулетку) і визначити сектор, в який це число входить (на якому зупинилась кулька). Наприклад, нехай є 3 точки (рисунок 3.5) в які мурашка може перейти з ймовірність переходів 20%, 30% і 50% відповідно, а згенероване число становить 56.58312%. Це число входить в діапазон від 50 до 100, отже мурашка перейде в точку №3.

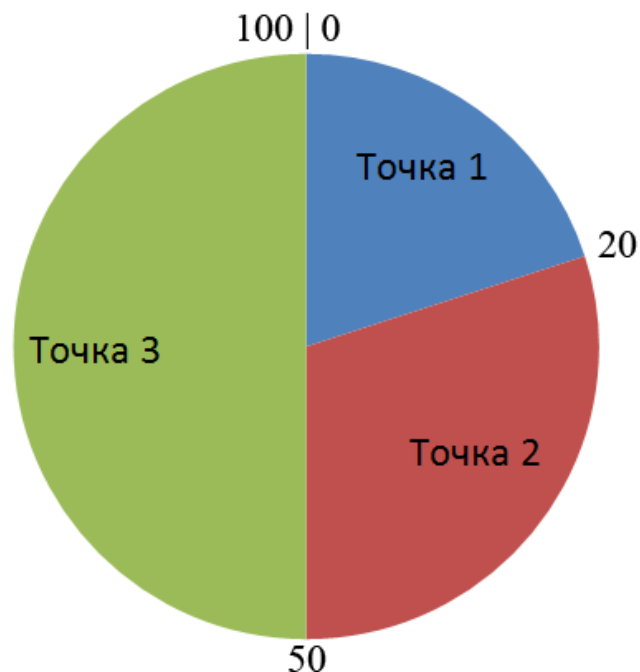


Рисунок 3.5 – Вибір ймовірності переходу в точку за принципом «Колеса рулетки»

Після вибору переходу в точку, дана точка відмічається і виконується повернення в головну функцію. Отже на цьому опис основної функції програми закінчено.

Тепер запусимо програму і спробуємо знайти найкоротшу відстань (рисунок 3.6). З рисунку 3.6 можна побачити, що знайдений шлях буде дійсно найкоротшим з заданим радіусом видимості.

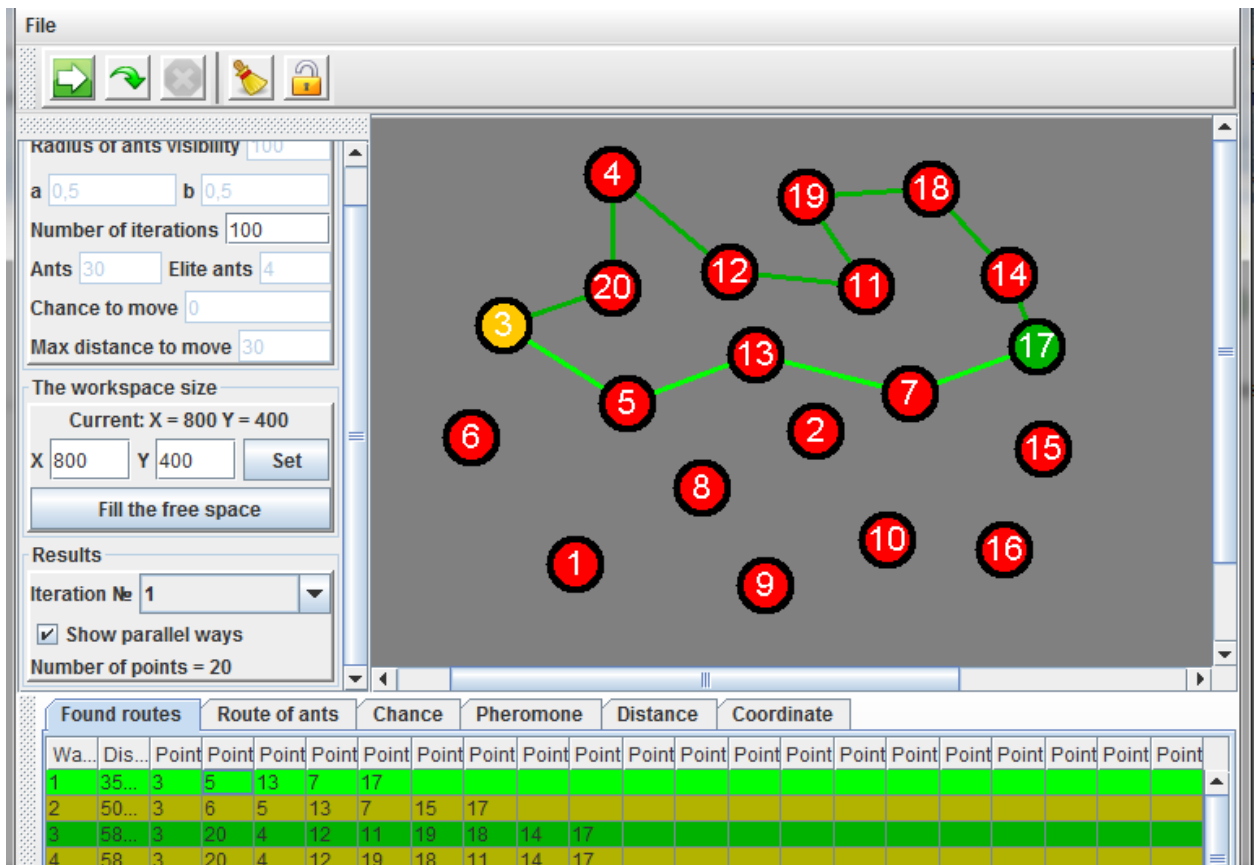
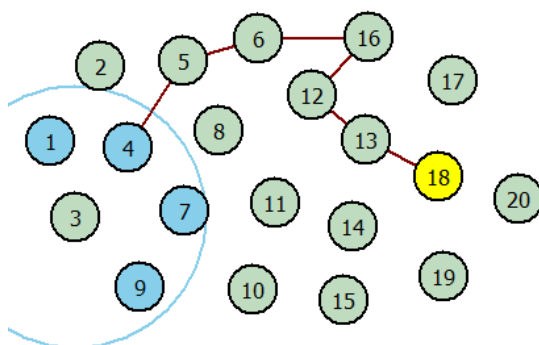
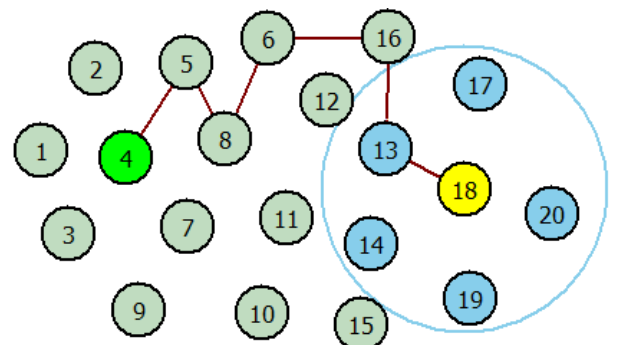


Рисунок 3.6 – Головне вікно програми пошуку незалежних маршрутів передачі

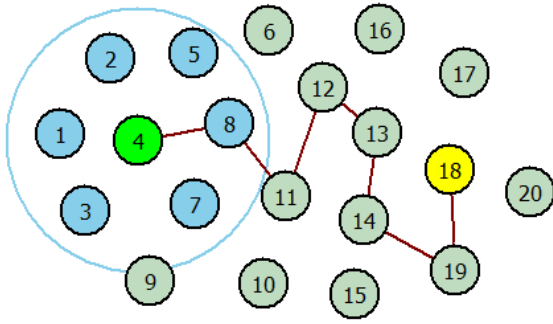
При пошуку маршруту мурашки, на кожній ітерації, генерують нові рішення (рисунок 3.7 (а-е)). У розглянутому прикладі в результаті роботи програми оптимальний шлях знайдений першою мурашкою на четвертій ітерації (рисунок 3.7 е).



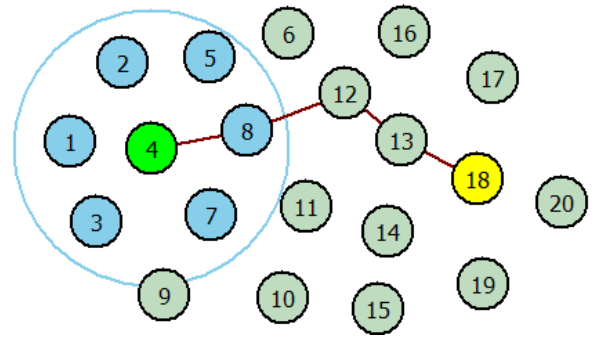
а) ітерація № 1, мурашка 1, довжина маршруту – 398



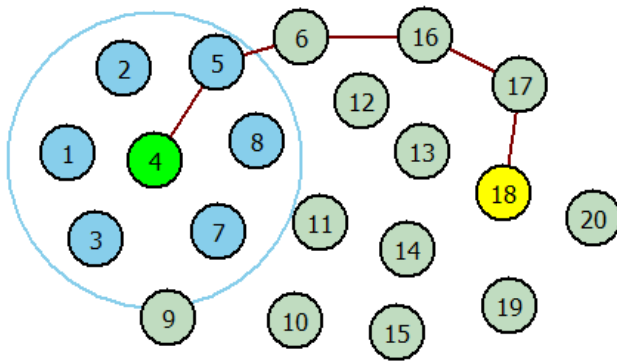
б) ітерація №1, мурашка 2, довжина маршруту – 437



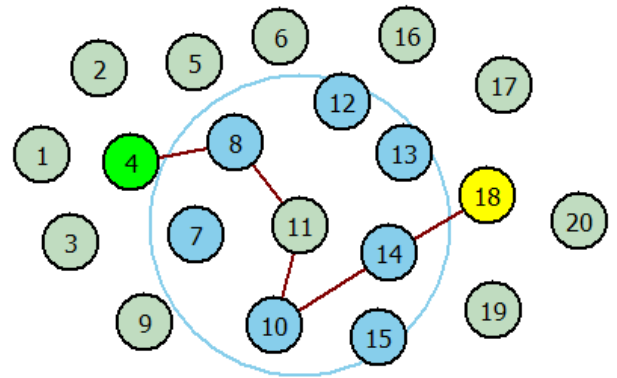
в) ітерація №1, мурашка 3, довжина маршруту – 501



г) ітерація №4, мурашка 1, довжина маршруту – 261



д) ітерація №4, мурашка 2, довжина маршруту – 367



е) ітерація №4, мурашка 3, довжина маршруту – 373

Рисунок 3.7 – Еволюція пошуку оптимального маршруту

На рисунку 3.8 приведені результати пошуку маршрутів передачі відсортовані в порядку зростання довжини.



Таблиці даних. Найкоротший шлях знайдено на ітерації №4

Ітерація № 4 Муравей № 1

Мурашки: Знайдені шляхи | Шанс переходу | Феромон | Відстань | Координати

Мура	Відст.	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка	Точка
1	261,1	4	8	12	13	18											
2	425,7	4	3	9	10	15	14	18									
3	437,6	4	5	8	6	16	13	18									
4	517,0	4	1	2	5	6	12	16	17	18							
5	577,8	4	3	9	10	14	11	12	13	18							
6	630,2	4	5	2	1	3	9	10	15	14	18						
7	633,9	4	2	1	3	9	10	15	14	19	18						
8	684,0	4	3	9	10	14	11	13	16	17	18						
9	693,8	4	1	3	9	7	10	15	19	14	13	18					
10	739,3	4	3	1	2	5	8	7	9	10	15	14	18				
11	740,2	4	3	7	11	10	15	14	13	12	16	17	18				
12	779,8	4	3	9	7	8	11	10	14	15	19	20	18				
13	811,3	4	3	1	2	5	8	11	10	15	19	14	13	18			
14	Немає	4	3	9	10	15	11	12	13	17	16	6	5	8	7	Тупик	
15	Немає	4	3	7	11	10	9	Тупик									
16	Немає	4	1	2	5	6	16	12	8	7	11	14	10	9	3	Тупик	

Рисунок 3.8 – Відображення результатів роботи програми

Для заданої структури мережі (рисунок 3.7) знайдено чотири незалежні маршрути (рисунок 3.9).

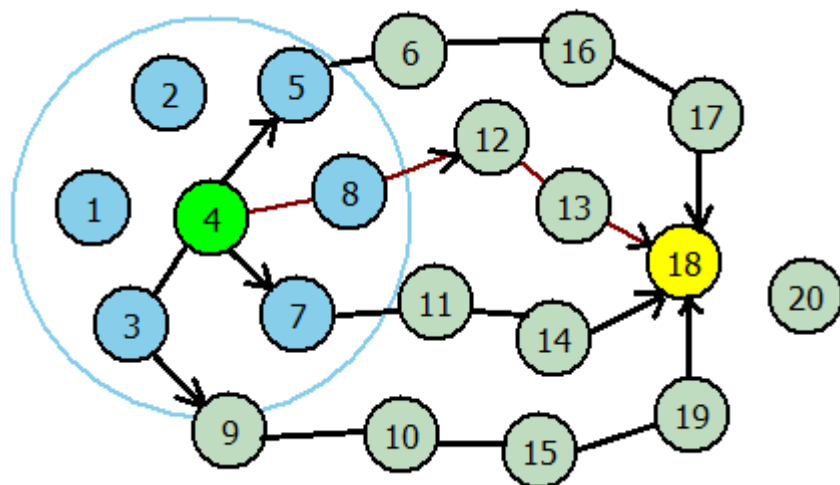


Рисунок 3. 9 – Результати пошуку незалежних маршрутів

За допомогою розробленого програмного забезпечення експериментально підтверджена достовірність результатів теоретичних положень пошуку оптимального маршруту на основі алгоритму мурашкової колонії.

### 3.4 Розробка протоколу багатошляхової маршрутизації на основі алгоритму мурашкових колоній

#### 3.4.1 Структура протоколу маршрутизації

Дані від джерела інформації передається на базову станцію за допомогою сусідніх вузлів, які працюють в якості ретрансляторів. Дані кожного джерела діляться на  $N$  частин. Кількість мурах, які беруть участь в задачі маршрутизації також дорівнює  $N$ . Вихідні дані містять таку інформацію, як ідентифікація джерела інформації, ідентифікація вузла, значення основи системи залишкових класів  $P_i$ , час і вихідні дані [27].

Вхідні дані розділяються у вузлі на  $N$  частин, згідно  $DATA(mod P_i)$  (рисунок 3.10).

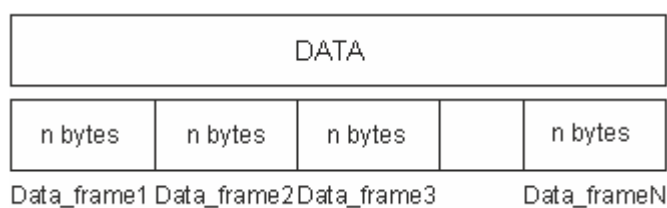


Рисунок 3.10 – Розділення вхідних даних

Розрядність даних залежить від значення модуля і визначається за формулою  $n = \lceil \log_2 P_i \rceil$ .

Після поділу вихідних даних на частини для створення пакета даних готового до передачі, до кожної частини додаються параметри маршрутизації.

До кожної частини даних додають:

IDDEA – ідентифікаційний код (код визначає призначення пакета: передаються дані, помилка або підтвердження пакета);

DSN – наступний вузол, до якого передається пакет;

FN – номер пакета  $k$  (також представляє номер агента мурашки);

$NJ_w^k$  – є порядковим номером кількості відвіданих вузлів до цього часу.

$Data\_frame^k$  –  $k$  - та частина даних, як показано на рисунок 3.11.

Перші чотири поля складають заголовок пакета (рисунок 3.11). Коли прийом всіх пакетів даних буде завершено, базова станція об'єднує їх у вихідні дані на основі формули (2.1).

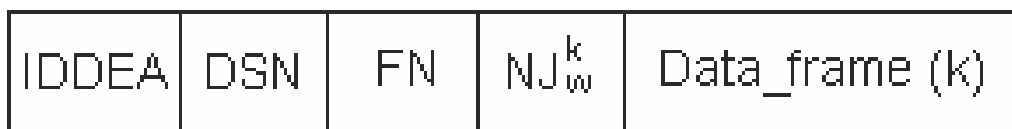


Рисунок 3.11 – Структура пакету передачі даних.

На рисунку 3.12 показаний приклад маршрутизації для вузла А. Вузол А щойно отримав пакет даних (номер агента дорівнює 5) і приймає рішення про вибір наступного пункту призначення для цього пакета.

Адрес наступного вузла передачі вибирається за допомогою рівняння (3.1), який забезпечує більш високу ймовірність  $P_{ij}^k$ . У цьому прикладі в якості шляху буде обране ребро (А, D), так як має більш високе значення феромона ( $\tau_{AD} = 0.7$ ). Для вирішального правила також мають значення рівні енергії в сусідніх вузлах В, С і D.

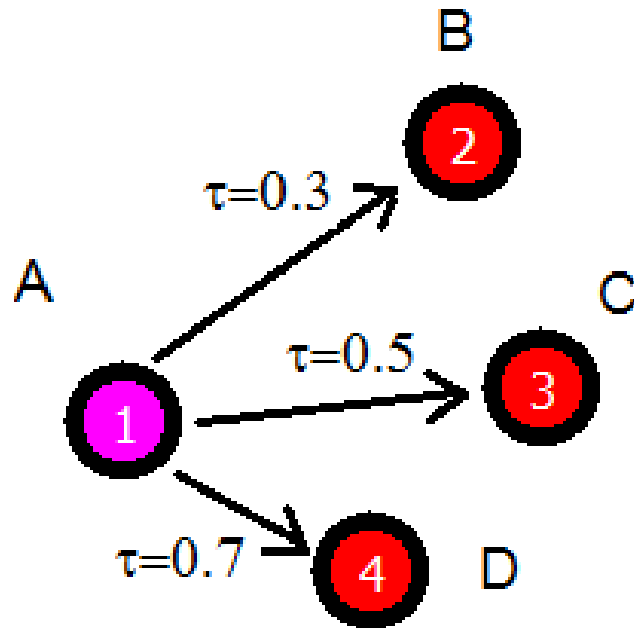


Рисунок 3.12 – Параметри мурашкового алгоритму, які запам'ятовуються у вузлах

Якщо був вибраний вузол D, потім поле DSN оновлюється і пакет передається. Вузли B і C також почують цю передачу. Вони перевіряють поле DSN і розуміють, що це не їх ідентифікатор ID, тому вони відкидають цей пакет відразу після зчитування області DSN.

Вузол D також перевіряє область пакета - DSN. З дозволу ID вузла D, а також, якщо FN немає у забороненому списку вузла D, тоді поле  $NJ_w^k$  оновлюється (збільшується на одиницю). Потім, на наступному вузлі оновлюється поле DSN, а також виконуються ті ж операції, що і на попередньому вузлі (вузол A).

Визначення розміру пакету даних має велике значення, так як кількість мурашок еквівалентно кількості пакетів. Таким чином, розмір пакета повинен бути визначений при початковому налаштуванні системи відповідно із середнім розміром даних і апаратними характеристиками.

Мінімальний розмір даних в пакеті, що складається з кадрів МАС повинен бути таким, як розмір кадру в нижньому рівні МАС. Якщо використовується МАС рівень протоколу IEEE 802.15.4, то максимальна корисне навантаження в кадрі складаємо до 104 байта. При моделюванні протоколу розмір поля даних приймався рівним 100 байт.

### 3.4.2 Дослідження протоколу

Після поділу даних у джерелі і їх передачі через сусідні вузли, базова станція отримує окремі частини даних з декількох шляхів [29].

Використання різних шляхів забезпечить підвищення надійності передачі даних у разі виходу з ладу основних маршрутів. Для запобігання втрати пакета в цих шляхах, використовується алгоритм виявлення та виправлення помилок і сигнали підтвердження. Після отримання пакета, базова станція обчислює значення феромона, по рівнянню (3.2), яке буде додано в  $NJ_w^k$ , (рисунок 3.13) і передає його в бік джерела, використовуючи той же шлях, який був використаний для відповідного пакета передачі даних.



Рисунок 3.13 – Структура пакету сигналу підтвердження.

Вузол, що отримав сигнал підтвердження спочатку перевіряє значення FN (рисунок 3.14). Якщо номер знаходиться в пам'яті (заборонений список) вузла, який був записаний попередньо при початковому поширенні пов'язаних пакетів, то цей пакет передається іншим сусідам уздовж шляху.

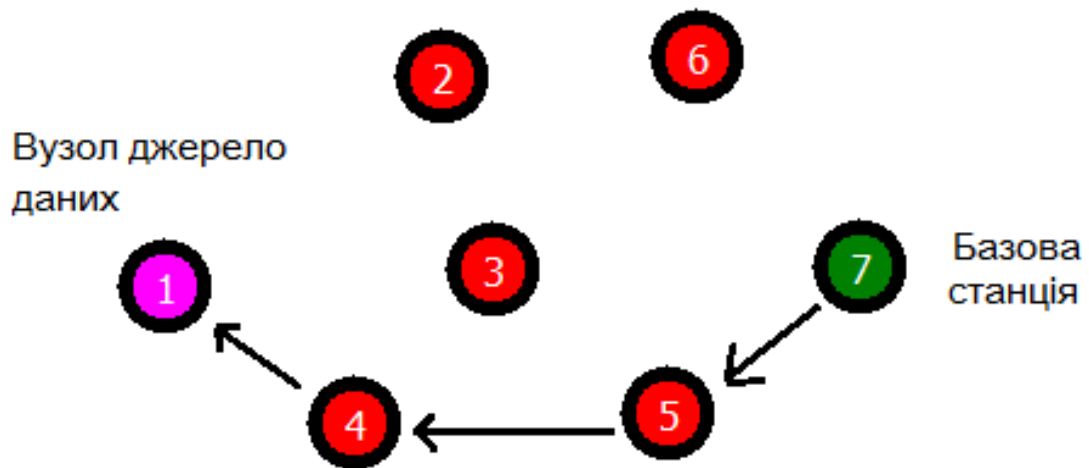


Рисунок 3.14 – Передача сигналу підтвердження приймання пакету

Вузол також оновлює значення феромону, додаючи вказаний в пакеті  $\Delta\tau^k$  за допомогою рівняння (3.3). Якщо ідентифікатор вузла не знайдений в пам'яті, то цей сигнал ігнорується, і ніякі подальші операції не будуть виконуватися. Таким чином  $\Delta\tau^k$  додається вздовж обраного шляху.

Вузол джерело даних чекає сигнал підтвердження прийому кожного пакета даних. У разі відсутності сигналу підтвердження, викликані помилками при передачі, приймається рішення про повторну передачу пакета.

Енергетичні рівні сусідніх вузлів, необхідні для прийняття рішення по правилу (3.1), для цього кожен вузол повинен повідомляти про запас енергії своїм сусідам. Коли відбувається зміна рівня енергії у вузлі, здійснюється передача цієї зміни сусіднім вузлам. Найбільш суттєві зміни енергії відбуваються після передачі даних.

Висновки до розділу.

В третьому розділі проведена порівняльна оцінка протоколів маршрутизації, яка показала, що класичний протокол OSPF показує більш низькі результати в умовах мобільних спеціалізованих мереж, ніж розроблені для них спеціалізовані протоколи маршрутизації.

На основі розробленого програмного забезпечення проведено дослідження алгоритму мурашкових колоній для пошуку маршруту передачі даних. Результати досліджень підтвердили високу ефективність використання алгоритму мурашкових колоній для пошуку незалежних маршрутів передачі даних.

Розроблено протокол багатошляхової маршрутизації на основі алгоритму мурашкових колоній, який забезпечує ефективний пошук, відкриття маршруту та передачу даних незалежними паралельними маршрутами.

## ВИСНОВКИ

В дипломній роботі розв'язано актуальну задачу розробки алгоритму та протоколу багатошляхової маршрутизації в безпроводних комп'ютерних мережах. При цьому отримано наступні результати.

1. Проведений аналіз та класифікація протоколів маршрутизації в бездротових мережах показала, що на даному етапі їх розвитку існує досить велика кількість стратегій маршрутизації, але, незважаючи на різноманітність протоколів маршрутизації в спеціалізованих мережах, існуючі методи маршрутизації не забезпечують ефективної передачі даних з наданням необхідних параметрів якості обслуговування.

2. Розроблено алгоритм пошуку оптимального маршруту в безпроводних мережах на основі алгоритму мурашкових колоній, який забезпечує пошук незалежних маршрутів передачі від джерела до одержувача.

3. Оцінка складності алгоритмів пошуку найкоротшого маршруту показала переваги алгоритмів мурашкових колоній, які характеризуються меншою часовою складністю, а також їх перспективність застосування для вирішення не тільки статичних, але і динамічних задач, наприклад для розробки алгоритмів маршрутизації в безпроводних мобільних мережах.

4. Проведена порівняльна оцінка протоколів маршрутизації, яка показала, що класичний протокол OSPF показує низьку ефективність роботи в умовах мобільних спеціалізованих мереж, ніж розроблені для них спеціалізовані протоколи маршрутизації.

5. На основі розробленого програмного забезпечення проведено дослідження алгоритму мурашкових колоній для пошуку маршруту передачі даних. Результати досліджень підтвердили високу ефективність використання алгоритму мурашкових колоній для пошуку незалежних маршрутів передачі даних.



6. Розроблено протокол багатошляхової маршрутизації на основі алгоритму мурашкових колоній, який забезпечує ефективний пошук, відкриття маршруту та передачу даних незалежними паралельними маршрутами. Використання протоколу багатошляхової маршрутизації забезпечує підвищення надійності передачі даних в безпроводних комп'ютерних мережах.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Амато Вито. Основы организации сетей Cisco, том 1. Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 512 с.
2. Томас М. Томас П. Структура и реализация сетей на основе протокола OSPF, 2-е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 816 с.
3. Kemal Akkaya, Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks* 3, 2005 – pp.325 – 349.
4. Тель Ж. Введение в распределенные алгоритмы. Пер. с англ. В.А. Захарова. – М. МЦНМО, 2009 – 616 с.
5. Винокуров В.М. Маршрутизация в беспроводных мобильных Ad hoc-сетях / В.М. Винокуров, А.В. Пуговкин, А.А. Пшенников, Д.Н. Ушарова, А.С. Филатов// Управление, вычислительная техника и информатика. Доклады ТУСУРа, № 2 (22), часть 1, 2010. – С.288 – 292
6. Су Цзюнь. Критерии оценки качества методов маршрутизации в беспроводных сенсорных сетях / Су Цзюнь // Матеріали І Всеукраїнської школи-семінару молодих вчених і студентів “Сучасні комп’ютерні інформаційні технології” (20 – 21 травня 2011 р., м. Тернопіль). Тернопіль: ТНЕУ, 2011. – С.58 – 59.
7. Carlos de Moraes Cordeiro. Ad hoc & Sensor Networks, Theory and Applications / Carlos de Moraes Cordeiro, Dharma Prakash Agrawal. – Singapore: World Scientific Publishing Co, 2006. – 642 p.
8. Julian Hsu Bhatia. Performance of Mobile Ad hoc Networking Routing Protocols in Large Scale Scenarios / Julian Hsu Bhatia, S. Tang, K. Bagrodia, R. Acriche // IEEE Military Communications Conference. – 2004. – Vol. 1. – P. 21–27.
9. Аоки М. Введение в методы оптимизации / М. Аоки. – М.: Наука, 1977. – 344 с.

10. Штовба С.Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях, 2003, №4 – С.70-75.
11. МакКоннелл Дж. Основы современных алгоритмов. – М.: Техносфера, 2004. – 368 с.
12. Олифер В. Компьютерные сети. Принципы, технологии, протоколы / Олифер В. Г., Олифер Н. А.//– СПб.: Питер, 2007. – 960 с.
13. Міночкін А. І., Романюк В. А., Сова О. Я. Аналіз методів управління навантаженням у мобільних радіомережах на транспортному рівні моделі OSI// Збірник наук праць ВІТІ НТУУ «КПІ». №3. – 2006. – С.68-72.
14. Кучерявий Е. А. Управление трафиком и качество обслуживания в сети Интернет. – СПб.: Наука и техника, 2004. – 336 с.
15. Миночкин А. И., Романюк В. А. Управление топологией мобильной радиосети // Зв'язок –2003 – № 2 – С 28-33.
16. . H. Suzuki and F.A. Tobagi. Fast Bandwidth Reservation Scheme with Multi-link & Multi-path routing in ATM Networks // Proc. of IEEE INFOCOM'92, Florence, Italy, May 1992.
17. P. Georgatsos and D. Griffin. A Management System for Load Balancing Through Adaptive Routing in Multiservice ATM Networks // Proc. of IEEE INFOCOM'96, San Francisco, CA, Mar. 1996.
18. Maxemchuk N.F. Dispersity Routing in High-speed Networks. Computer Networks and ISDN System, vol. 25, issue 6, 1993 – pp. 645-661.
19. A. Nasipuri and S.R. Das, On-demand Multipath Routing for Mobile Ad Hoc Networks. Proc. of the 8th International Conference on Computer Communications and Networks (IC3N), Boston, MA, Oct. 1999.
20. Marina M.K. and Das S.R. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. Proc. of the Ninth International Conference for Network Protocols (ICNP), Riverside, CA, Nov. 2001.
21. De S., Qiao C., and Wu H.. Meshed Multipath Routing: An Efficient Strategy in Sensor Network. Computer Networks, Special Issue on Wireless Sensor Networks, vol. 43, issue 4, Nov. 2003 – pp. 481-497.

22. Ganesan D., Govindan R., Shenker S., and Estrin D. Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. Proc. of the Second ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), Long Beach, CA, Oct. 2001

23. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. Proc. of ACM MobiCom'00, Boston, MA, Aug. 2000, pp. 56-67.

24. W. Lou. An efficient N-to-1 mutlipath routing protocol in wireless sensor networks // Proceedings of IEEE international Conference on Mobile Ad-hoc and Sensor Systems (MASS), Washington, DC, November 2005.

25. Bonavear E., Dorigo M. Swarm Intelligence: from Natural to Artificial Systems. – Oxford University Press, 1999. – 307 p.

26. Dorigo M. Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // Reader for CEU Summer University Course «Complex System».– Budapest, Central European University, 2001. – P. 1–38.

27. Reimann M. Ant Based Optimization in Good Transportation. PhD Thesis. University of Vienna. – Vienna, Austria, 2002. – 149 p.

28. Caro G. D., Dorigo M. Anet: a Mobile Agents Approach to Adaptive Routing. Technical Report IRIDA 97-12. IRIDA – Universite Libre de Brusseles. – Brussels, Belgium, 1997. – 27 p.

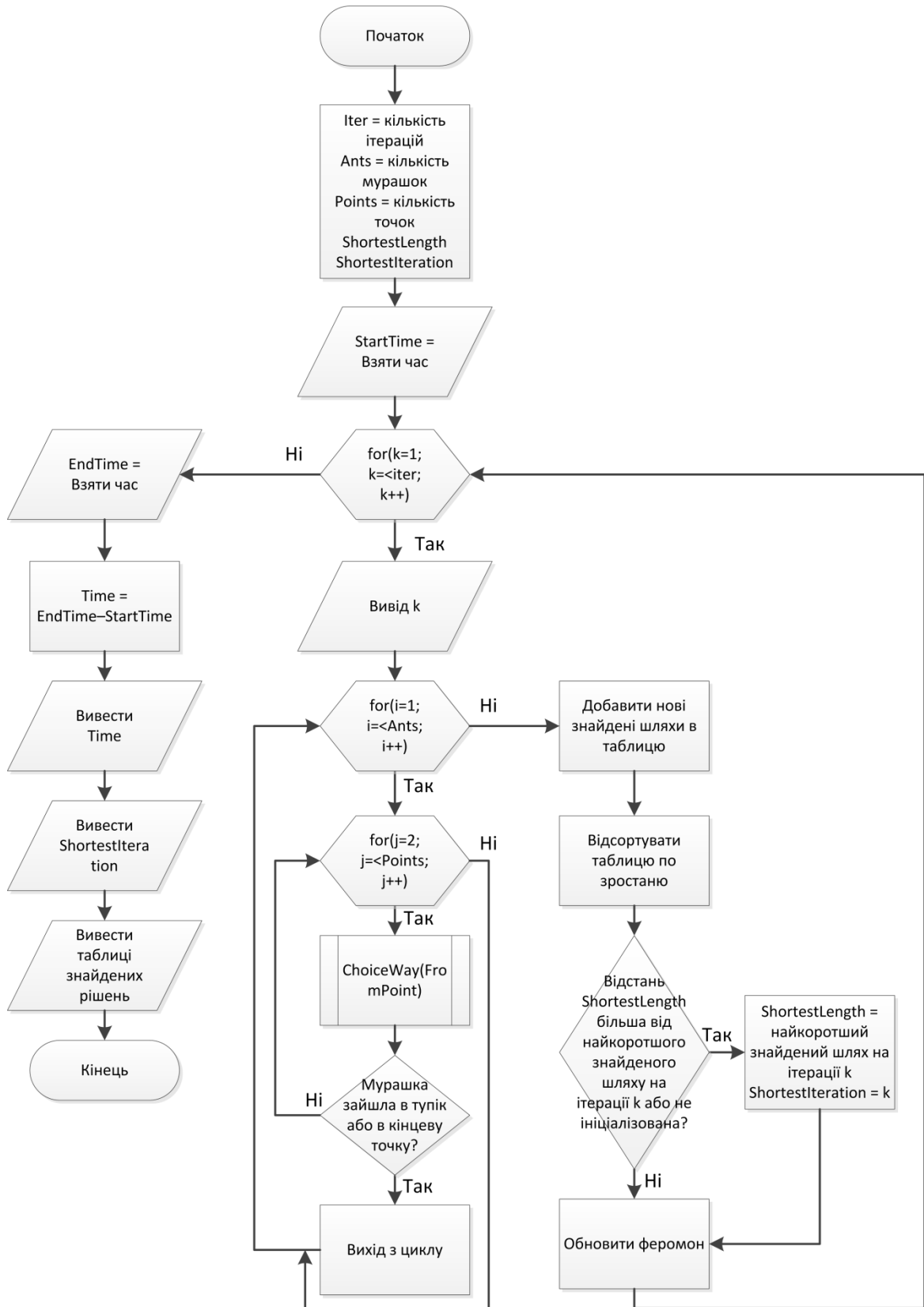
29. Cherix D. Note preliminaire sur la structure, la phenologie et le regime alimentaire d'une super colonie de Formica lugubris Zett. // Insects Sociaux 27, 1980. – P. 226–236.

30. Вимоги до оформлення дипломних робіт за освітньо-кваліфікаційними рівнями “спеціаліст” і “магістр”/ За ред. проф. Г.П.Журавля – Тернопіль: ТНЕУ, 2007. – 36 с.

31. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність „Комп’ютерні системи та мережі” / О.М. Березький, Р.Б. Трембач, Н.М. Васильків, Г.М. Мельник /Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2012.– 42 с.

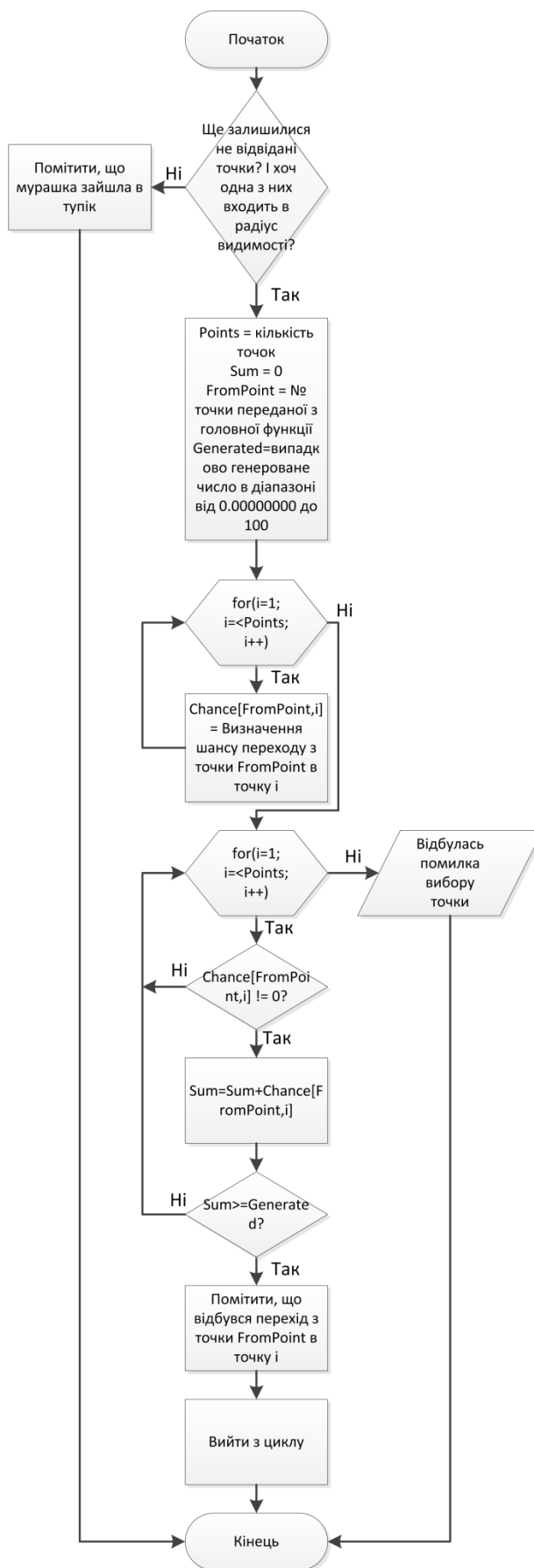
# ДОДАТОК А

## Головна функція роботи програми



## ДОДАТОК Б

### Функція вибору точки в яку перейде мурашка



## ДОДАТОК В