

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Комп'ютерні засоби контролю параметрів розподілених
об'єктів з детермінованим вибором вузлів сенсорної мережі

(тема)

Виконала: студентка 2 курсу, групи СКСм-
18-1 Срібна М.О.
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані
комп'ютерні системи

(повна назва освітньої програми)

Керівник д.т.н., проф. Кривуля Г.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____

(підпис)

Чумаченко С.В.

(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Автоматизації проектування обчислювальної техніки _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 – Комп'ютерна інженерія _____

Тип програми _____ Освітньо-професійна _____

Освітня програма _____ Спеціалізовані комп'ютерні системи _____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентці _____ Срібній Марії Олександрівні

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Комп'ютерні засоби контролю параметрів розподілених об'єктів з детермінованим вибором вузлів сенсорної мережі _____

затверджена наказом по університету від _____ 04 _____ листопада _____ 2019 р. № 1624 Ст 2.

Термін подання студентом роботи до екзаменаційної комісії _____ 20 _____ грудня _____ 2019 р.

3. Вихідні дані до роботи _____

Збір розподілених даних в бездротових сенсорних мережах _____

Детермінований вибір вузлів сенсорної мережі _____

Кількість датчиків в сенсорній мережі

4. Перелік питань, що потрібно опрацювати в роботі

Аналіз предметної області

Актуальні проблеми розгортання БСМ

Постановка задачі

Алгоритм розташування датчиків БСМ

Моделювання проекрованої БСМ

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

16 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2019	
2	Аналіз предметної області	02.09.2019 – 12.09.2019	
3	Аналіз джерел з проблемної галузі	13.09.2019 – 28.09.2019	
4	Вивчення актуального стану та проблем предметної області	29.09.2019 – 05.10.2019	
5	Розробка алгоритму розташування датчиків БСМ	06.10.2019 – 16.10.2019	

6	Розміщення вузлів БСМ на основі вибраного алгоритму	17.10.2019 – 20.10.2019	
7	Аналіз отриманих результатів	20.11.2019 – 25.11.2019	
8	Оформлення пояснювальної записки	26.11.2019 – 09.12.2019	
9	Оформлення графічного матеріалу	10.12.2019 – 12.12.2019	
10	Перевірка виконаного проекту керівником	13.12.2019 – 18.12.2019	
11	Захист роботи	20.12.2019	

Дата видачі завдання 1 вересня 2019 р.

Студент _____
(підпис)

Керівник роботи _____ д.т.н., проф. Кривуля Г.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 95 сторінок, 43 рисунка, 8 таблиць, 42 джерела за переліком джерел.

ДАТЧИК, СИСТЕМА, БЕЗДРОТОВІ СЕНСОРНІ МЕРЕЖІ, РОЗГОРТАННЯ, МОНІТОРИНГ, ТЕМПЕРАТУРА, ВОЛОГІСТЬ, ПРОТОКОЛ, АЛГОРИТМ

У даній атестаційній роботі проводиться дослідження існуючих технологій та проектування оптимальної системи контролю параметрів розподілених об'єктів з детермінованим вибором вузлів сенсорної мережі. Використання БСМ стає широко поширеною практикою, здебільшого через їх ефективність та швидкість роботи. Відповідно, з розвитком різноманітних мініатюрних сенсорів і датчиків, об'єднаних в БСМ, можна дистанційно

контролювати такі параметри, як температура і вологість повітря, ґрунту, стан механізмів та систем на підприємстві, стан здоров'я та багато інших.

ABSTRACT

Explanatory note for this work contains 95 pages, 43 figures, 8 tables and 42 sources according to the list of sources.

SENSOR, SYSTEM, WIRELESS SENSOR NETWORKS, DEPLOYMENT, MONITORING, TEMPERATURE, HUMIDITY, PROTOCOL, ALGORITHM

This attestation work is dedicated to study of existing technologies and design of the optimal system for collecting the parameters for distributed objects with deterministic selection of the sensor network nodes. The use of wireless sensor networks is widespread practice, mainly because of their efficiency and speed. According to the development of a variety of miniature sensors integrated in the wireless sensor network you can remotely control parameters such as temperature, humidity of air and soil, industrial machines and systems conditions, health condition and many others.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	Ошибка! Закладка не определена.
ВСТУП	7
1 ПОСТАНОВКА ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	9
2 БЕЗДРОТОВІ СЕНСОРНІ МЕРЕЖІ.....	10
2.1 Актуальні проблеми використання та організації БСМ	13
2.2 Збір розподілених даних в БСМ з детермінованим вибором	15

2.3	Забезпечення масштабованості БСМ	16
2.4	Рішення задач багаторазового сферичного покриття площі	27
3	ПРОТОКОЛИ МАРШРУТИЗАЦІЇ БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖ	42
3.1	Фактори вибору протоколу маршрутизації БСМ	42
3.2	Класифікація протоколів маршрутизації БСМ	44
4	АЛГОРИТМ РОЗТАШУВАННЯ ДАТЧИКІВ БСМ	46
4.1	Поведінковий алгоритм TPSMA	46
4.2	Розміщення вузлів БСМ на основі алгоритму TPSMA	47
4.3	Модель проектованої БСМ	48
4.4	Результати моделювання	50
4.5	Розташування сенсорних вузлів	51
4.6	Реалізація систем підтримки прийняття рішень на базі нечіткої логіки	52
5	АНАЛІЗ РЕЗУЛЬТАТІВ	67
	ВИСНОВКИ	80
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БСМ – бездротова сенсорна мережа

IEEE – Institute of Electrical and Electronics Engineers

GPIO – General Purpose Input Output, інтерфейс введення/виведення загального призначення

PAN – Personal Area Network, персональна мережа

TPSMA – Territorial Predator Scent Marking Algorithm

DARPA – Defense Advanced Research Projects Agency, агентство передових оборонних дослідницьких проєктів

MeshLogic – універсальна база для створення БСМ

ZigBee – бездротовий стандарт передачі даних

Wireless-G – стандарт передачі даних в бездротовій мережі швидкості 54 Мбіт/с

CSV – Comma-Separated Values, відмежовувальний файловий формат для представлення табличних даних

ВСТУП

БСМ в останні роки привернули значний інтерес серед інженерної спільноти та наукових дослідників. Різні університети створюють різноманітні додатки сенсорних мереж, такі, як мережа моніторингу житлового середовища Каліфорнійського університету в Берклі та Коледжу Атлантики, проєкт Zebranet для моніторингу звичок тварин від Принстонського університету, БСМ, що контролює діяльність вулканів в Еквадорі, створена Гарвардським

університетом, Університетом Нью-Гемпшира та Університетом Північної Кароліни разом. Крім того, DARPA створила самовідновлюване інтелектуальне мінне поле на основі БСМ.

Різні сфери досліджень пов'язані з БСМ, такі, як безпека, охоплення, зв'язок в сенсорній мережі тощо. Дана робота розглядає проблему зони покриття мобільних і бездротових датчиків. У проблемі зони покриття кожен датчик охоплює певну підобласть і загальну покриту область сенсорної мережі, що складається з окремих покритих областей кожного вузла. Максимізація загальної зони покриття всієї сенсорної мережі є головною метою проблеми покриття території. Проблема зонального покриття тісно пов'язана з роботою систем в додатках, таких як виявлення цілей і відстеження цілей, моніторинг на полі бою, національна безпека, індивідуальний захист і моніторинг звичок тварин.

Елементи передбачуваної системи на базі БСМ матимуть можливість спілкуватися між собою при низькому рівні енергоспоживання, а також поставляти зібрані дані на мобільний або стаціонарний комп'ютер. Основні компоненти сенсорної мережі складаються з одного або декількох датчиків, які підключені до мікроконтролеру і радіомодулів. Велика кількість цих крихітних вузлів-датчиків розгортається на місцевості випадковим або визначеним чином, для виконання колективних дій з вимірювання параметрів навколишнього середовища на великій площі або навіть в важкодоступних областях місцевості.

На даний час ступінь впровадження технології в різноманітні галузі промисловості вже досить висока. Ручний збір даних і параметрів навколишнього середовища використовується лише в одиничних випадках. Також він вразливий до впливу людського чинника, що впливає на точність результатів вимірювання. Використання бездротових датчиків може скоротити час і необхідні зусилля для моніторингу навколишнього середовища і отримання даних. Використання систем моніторингу забезпечує

скорочення часу реагування на несприятливі фактори і умови, покращує контроль якості продукції та забезпечує зниження витрат праці. Використання подібних технологій дозволить дистанційно вимірювати фактори і параметри навколишнього середовища, такі як температура і вологість повітря, ґрунту, рівень води і безліч інших.

Основною причиною використання технологій БСМ є необхідність зміни розташування датчиків, в разі чого традиційна побудова мережі може потребувати істотного збільшення витрат, енергії і часу на вирішення подібних проблем. Передбачувана система спрямована на зниження витрат і зусиль на розгортання мережі, а також підвищення гнучкості і мобільності обраних точок моніторингу.

1 ПОСТАНОВКА ЗАВДАННЯ НА ПРОЕКТУВАННЯ

Об'єкт дослідження – процеси переробки інформації в розподілених технічних системах.

Предмет дослідження – моделі та методи контролю параметрів розподілених об'єктів з використанням БСМ.

Метою роботи є дослідження технологій комп'ютерного контролю параметрів розподілених об'єктів на основі БСМ з детермінованим вибором вузлів сенсорної мережі.

В процесі реалізації мети дослідження планується розглянути варіанти реалізації БСМ MeshLogic та ZigBee, провести їх порівняльний аналіз. В роботі реалізується алгоритм заданого розташування вузлів сенсорної мережі для оптимального покриття площини контролю розподіленого об'єкта.

Передбачувана система заснована на модулі IEEE 802.15.4 / ZigBee для передачі даних на обраний сервер. Модуль ZigBee / IEEE 802.15.4 з'єднаний з різними датчиками, що мають аналогові виходи через мультиплексор, який

використовується для стробування сигналу. Таким чином можна вибрати два конкретних сигнали на основі значень системи GPIO з модуля БСМ.

Сигнали вимірюються і перетворюються на значення, які потім передаються на обраний сервер, підключений до тієї ж мережі через шлюзовий елемент, наприклад, стандартний Wireless-G маршрутизатор.

Сервер може бути підключений до мережі бездротовим способом або за допомогою дротового Ethernet-з'єднання. Він зберігає отримані дані у файлі текстового формату CSV, призначеному для представлення табличних даних, які можуть бути прямо імпортовані в файл бази даних типу Excel або будь-яке інше програмне забезпечення для виконання аналізу і відображення даних.

2 БЕЗДРОВОТІ СЕНСОРНІ МЕРЕЖІ

Сенсорна мережа - це розподілена мережа необслуговуваних мініатюрних електронних пристроїв (вузлів мережі), які здійснюють збір даних про параметри зовнішнього середовища і передачу їх на базову станцію за допомогою ретрансляції від вузла до вузла за допомогою бездротового зв'язку. Вузол мережі, званий сенсором, містить датчик, що сприймає дані від зовнішнього середовища (власне сенсор), мікроконтролер, пам'ять, радіопередавач, автономне джерело живлення і іноді виконавчі механізми. Можлива також передача керуючих впливів від вузлів мережі до зовнішнього середовища [1].

Сенсорні мережі будуються на основі протоколів IEEE 802.15.4 [2], ZigBee [3] і DigiMesh [4]. Функціонально сенсорна мережа містить вузли трьох видів: (1) функціональні вузли (Ф-вузли), які здійснюють збір інформації в деякій околиці точки розміщення даного вузла; (2) транзитні вузли (Т-вузли), що виконують тільки передачу інформації і управління маршрутизацією; (3) базові станції (БС), що здійснюють глобальну координацію, організацію та

установку параметрів мережі. Сенсорна мережа може бути організована як сукупність підмереж (кластерів), пов'язаних координаторами або шлюзами взаємодії «сенсорна мережа - корпоративна мережа». Шлюзи в цьому випадку повинні виконувати функції координатора і здійснювати зв'язок з сервером (координатором сенсорної мережі), який керує всією сенсорною мережею. Функціонально до базових станцій віднесемо координатор PAN і шлюз взаємодії «сенсорна мережа - корпоративна мережа».

БСМ є мережею просторово розподілених автономних датчиків, що використовуються для моніторингу будь-яких фізичних параметрів, зокрема параметрів навколишнього середовища, і спільної передачі даних на головний сервер. Контрольовані параметри залежать від функції сенсорних вузлів в мережі, наприклад фізичні датчики (температури, вологи, радіохвильової частоти датчиків), хімічні датчики (електропровідності, рН), біологічні датчики (наприклад, датчик мікроорганізмів), датчики, орієнтовані на національну безпеку, а також багато інших знову і знову винайдених датчиків. В даний час датчик БСМ є не тільки одиночним елементом, він оснащений безліччю сенсорних елементів і називається сенсорним вузлом.

БСМ зазвичай складаються з двох основних частин: системи для генерації і збору значень з розподілених датчиків; системи обробки і локального зберігання значень з датчиків перед відправкою цих даних користувачу.

Система генерації та збору значень (сенсорна мережа) складається з сотень самоорганізовуваних сенсорних вузлів, які взаємодіють і обмінюються інформацією між собою за допомогою радіосигналів, і створюють мережу типу ad hoc. Ці датчики представляють собою невеликі блоки, здатні зберігати обмежену кількість даних і мають обмежену швидкість обробки даних і ширину каналу радіозв'язку. Сенсорні вузли можуть бути розгорнуті певним або випадковим чином і можуть покривати більшу площу з високою щільністю в залежності від необхідних вимог до розміщення.

Система обробки і локального зберігання може бути звичайним сервером, що зберігає значення датчиків і перетворює їх в формат, зручний для розпізнавання та подальшого відтворення користувачу [5].

Як правило, БСМ створюються для вирішення конкретного завдання (моніторингу навколишнього середовища, спостереження за об'єктом і т.д.) і призначені для оптимізації та підвищення продуктивності базового сценарію розгортання. Значення датчиків можуть бути попередньо оброблені системою генерації і збору значень під оптимальну продуктивність мережі і далі передані сенсорною системою на сервер обробки через Інтернет або локальну мережу. Крім того, сервер обробки надає користувачам сервіси управління і контролю даними, а також зручний користувацький інтерфейс.

У БСМ існує багато різновидів пристроїв і сенсорних вузлів, що використовуються для забезпечення стабільної роботи мережі з високою продуктивністю. На основі функціональних і комунікативних ролей вони поділяються на чотири основні типи.

Оскільки сенсорний вузол складається з багатьох елементів, він може виконувати в БСМ більше однієї функції одночасно, і також, в залежності від застосування, БСМ може містити всі ці типи вузлів або тільки деякі з них.

- **Sub-node** (кінцевий пристрій): вузол без маршрутизації. Цей вузол може тільки збирати дані з контрольованого навколишнього середовища і передавати їх дані на інші вузли.

- **Head-node** (маршрутизуючий вузол): вузол, здатний приймати дані від іншого вузла і направляти їх до подальшого вузла, вузла-шлюзу. Залежно від топології мережі цей вузол також може виконувати функції звичайного вузла збору даних з контрольованого навколишнього середовища або бути точкою маршрутизації.

- **Sink-node** (gateway, вузол-шлюз): вузол, який збирає або запитує дані з інших вузлів і передає їх в іншу мережу. Може виконувати функції

звичайного вузла збору даних з контрольованого навколишнього середовища, за винятком пересилання даних іншим вузлам мережі.

- Controller (контролер або координатор): центральний орган управління для координації адресації вузлів і з'єднань в мережі, маршрутизації, передачі даних за розкладом, синхронізації між вузлами і т.д. Залежно від топології мережі цей елемент може як використовуватися, так і бути зовсім відсутнім в БСМ.

2.1 Актуальні проблеми використання та організації БСМ

Незважаючи на те, що БСМ вже тривалий час широко використовуються в багатьох областях, все ще існують деякі значні недоліки, які не дозволяють їм стати першим технологічним вибором у багатьох випадках. Можна виділити два основних недоліки, які в основному і чинять подібний вплив: значне енергоспоживання, а також необхідність надмірності або використання великої кількості резервних вузлів для підтримки стабільної роботи подібних високопродуктивних мереж.

Як правило, БСМ розгортаються для контролю або моніторингу протягом тривалого періоду часу (кілька місяців або років), в разі чого часта заміна джерел живлення (акумуляторів) в сотнях сенсорних пристроїв, що використовуються, не є можливою і доцільною. Таким чином, кількість енергії, споживаної кожним сенсорним вузлом мережі є одним з обмежень, які слід враховувати при проектуванні БСМ. При зменшенні витрат енергії кожного вузла мережі збільшується термін служби всієї системи. Відповідно, споживана потужність БСМ може бути оптимізована зменшенням кількості енергії, споживаної окремими вузлами з продовженням їх життєвого циклу, шляхом вибору оптимальної топології мережі і протоколу маршрутизації.

Для забезпечення максимальної ефективності роботи сенсорний вузол повинен знаходитися в сплячому режимі 90% робочого часу (рисунок 2.1) і «прокидатися» лише згідно з розкладом, або в зв'язку з примусовою активацією [7]. Значна частина енергії вузла витрачається протягом активного періоду, тому дотримання належних інтервалів сплячого режиму є основоположною умовою його роботи протягом тривалого періоду часу. У БСМ на основі підходу single-hop (топологія зірка), де вузол безпосередньо пов'язується з базовою станцією, на відміну від мереж на основі підходу multihop (топології сітка, дерево та кластерні топології), інтервал сплячого режиму залежить тільки від вимог до застосування подібної мережі. На рисунку 2.1 зображений робочий цикл сенсорного вузла.



Рисунок 2.1 - Робочий цикл сенсорного вузла БСМ

Існують деякі методи, розроблені для підвищення енергоефективності вузлів. Наприклад, техніки агрегування даних скорочують передачу даних проміжними вузлами, однак це призводить до можливого підвищення кількості помилок в переданих даних і виникнення затримок при їх передачі.

Іншою актуальною проблемою при побудові і використанні БСМ є необхідність в досить великій кількості вузлів для підтримки гарної продуктивності мережі. Як правило, при розгортанні БСМ вузли розташовуються таким чином, щоб покрити всю область моніторингу. У той же час, вони повинні мати можливість з'єднання з іншими вузлами для обміну даними. Часто кількість необхідних для збереження безперебійного з'єднання

в мережі вузлів в кілька разів більше кількості, необхідної для підтримки оптимального покриття.

2.2 Збір розподілених даних в БСМ з детермінованим вибором

Незважаючи на те, що датчики в БСМ є автономними і можуть працювати незалежно від мережі, вони мають обмеження по споживанню енергії, швидкості обробки даних і обмежений обсяг пам'яті. Як правило, значення з датчиків збираються і обробляються за межами мережі. Всі програми, які містяться за межами мережі датчиків, будуть підключатися до неї за допомогою шлюзу.

У БСМ існує два способи збору значень з датчиків: single-hop і multi-hop, представлені на рисунку 2.2.

Мережа single-hop - це мережа, де кінцеві точки (тобто джерело та призначення) є єдиними станціями в мережі. Оскільки існує тільки одна інша станція, крім джерела (тобто, один «хоп» - пункт призначення), це позначається мережею з одним хопом. Проте, незважаючи на логічно прямий зв'язок, передача даних в мережі single-hop потребує значних додаткових затрат енергії, коли вузли будуть розташовані на значно більшому рівні від базових станцій або будуть мобільними і віддаленими від неї.

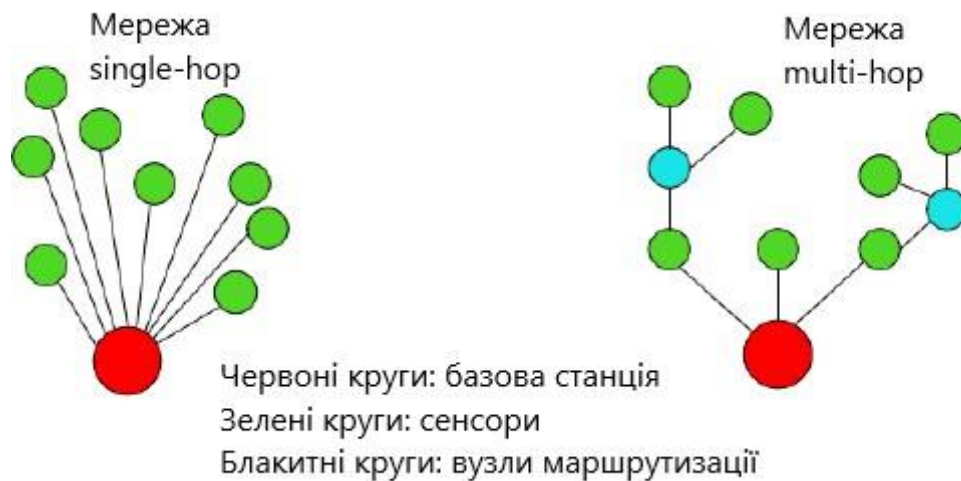


Рисунок 2.2 - Способи збору даних з датчиків в БСМ

Multi-hop мережа - мережа, де, крім двох кінцевих точок, між джерелом і кінцевим пунктом існує принаймні одна інша станція. Підхід multi-hop може вирішити проблему затрат енергії на передачу даних міжвіддаленими вузлами шляхом ретрансляції пакетів даних між джерелом і пунктом призначення з використанням проміжних вузлів (вузлів маршрутизації). Під час операції передачі даних, проміжні вузли можуть також збирати дані і передавати їх разом для підвищення ефективності та зменшення енерговитрат [6]. Також, при використанні підходу multi-hop, передача даних може ґрунтуватися на трьох різних стратегіях маршрутизації: проактивній, реактивній і гібридній.

2.3 Забезпечення масштабованості БСМ

Усі алгоритми, описані вище, розглядають лише однакові датчики з нерегульованим діапазоном. Це обмеження породжує алгоритми, які не можуть досягти одночасно забезпечення 100% покриття та енергоефективності, оскільки в діапазонах датчиків занадто багато перекриттів. У нижчеописаному алгоритмі розроблено розподілений метод використання датчиків з регульованим діапазоном для забезпечення 100% покриття, а також економії енергії для тривалого моніторингу. Цей алгоритм

можна у безлічі різних випадків (наприклад, безпосередньо після випадкового розгортання, після вибору та групування сенсорних вузлів), в тому числі й після переміщення датчиків.

Є два припущення для даного розподіленого алгоритму регулювання діапазону моніторингу:

1. Усі датчики мають регульований радіус моніторингу. Кожен датчик може скоротити свій радіус моніторингу, знизивши потужність.
2. Усі датчики знають своє поточне розташування в межах області моніторингу, а її деталі, такі, як межі області покриття, попередньо встановлюються в їх пам'ять.

У алгоритмі використовується діаграма Вороного - метод розподілу області. Якщо набір з N точок знаходиться в області A , область буде поділена на N підобластей, кожна з яких має всередині лише одну точку. Будь-яка точка $L(x_l, y_l)$ в i -й підобласті (A_i) матиме коротшу відстань до точки $I(x_i, y_i)$ в цій підобласті, ніж будь-яка інша точка $J(x_j, y_j)$. Це можна сформулювати за допомогою наступного рівняння: $A_i = \{l \in A, D_i \leq D_j, \text{ для всіх } i \neq j\}$.

де D_i - відстань між L та I , а D_j - відстань між L та J .

У даному випадку область A буде цільовою областю охоплення, а точки - датчики, що забезпечують покриття. З припущень вище кожен датчик може обчислити діаграму Вороного локально, транслюючи своє місцезнаходження та отримуючи інформацію про їх місцезнаходження від інших датчиків. Тому діаграми Вороного можна використовувати для визначення підобласті області, найближчої до кожного датчика. Діаграми Вороного були використані в мобільних сенсорних мережах для переміщення в [7]. Два датчики, що мають однакову вершину підобласті Вороного, називаються сусідами.

При застосуванні вищезгаданих алгоритмів, що використовують однакові та нерегульовані датчики, завжди з'являються отвори в покритті та надлишкове покриття (див. рисунок 2.3). Шість датчиків однакового діапазону

розгорнуті для покриття прямокутної області. Підобласть зеленого кольору покрита трьома датчиками, тобто покриття надлишкове. Підобласть червоного кольору не покрита жодним датчиком і являє собою отвір покриття.

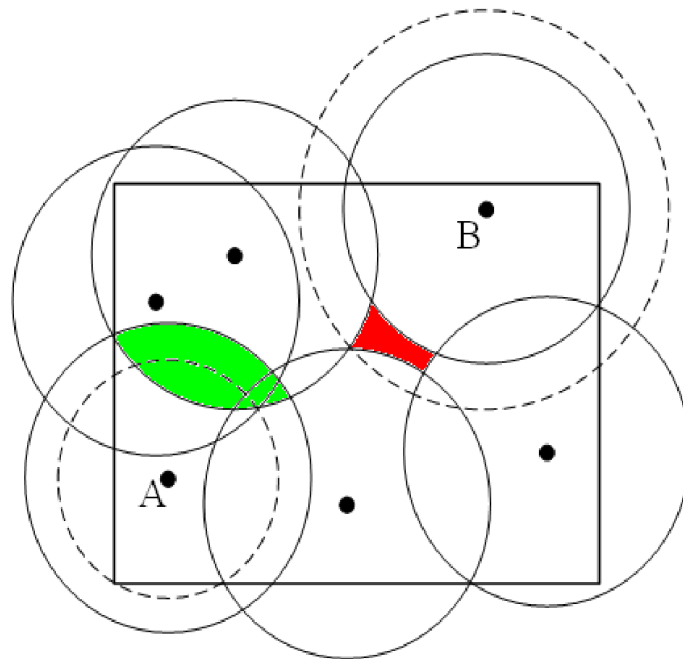


Рисунок 2.3 – Надлишкове покриття та отвори

Алгоритм вирішує наступне питання: як можна використати датчики з регульованим діапазоном, щоб уникнути отворів покриття та надлишкового покриття? Перша задача - забезпечити повне охоплення, оскільки цього вимагає більшість систем спостереження. Друга задача - зменшити область моніторингу, щоб скоротити надмірність, заощадивши енергію. Наприклад, проблема, показана на рисунку 2.3, може бути вирішена, якщо область покриття можна зменшити до області, позначеної пунктирними лініями, зображеними для датчиків А і В. Таким чином, датчик А економить деяку кількість енергії, але не створює непокритих ділянок. Крім того, датчик В споживає більше енергії, щоб забезпечити повне покриття області моніторингу.

У розподілених алгоритмах кожному датчику нелегко отримати інформацію про те, чи покрита вся область. Значно простіший спосіб

контролю за коефіцієнтом покриття всієї області - розділити область моніторингу і дозволити кожному датчику нести відповідальність за покриття лише близької до нього підобласті. Цей метод реалізується за допомогою діаграми Вороного, як описано вище.

Описано метод досягнення першої задачі - повне охоплення. Однак дотримання виключно цього методу призводить до марних витрат енергії. Приклад такої ситуації показано на рисунку 2.4. П'ять датчиків розгорнуто в області моніторингу 10 на 10 метрів. Червоні лінії позначають найдальшу вершину для кожного датчика.

Хоча вся сенсорна область охоплена п'ятьма датчиками, підобласть Вороного для датчика D здебільшого також охоплена й іншими датчиками, що оточують його. Датчик D може просто зменшити діапазон покриття на відстань DM, на якій M є точкою перетину діапазонів датчиків B і E. Навпаки, якщо будь-який з датчиків A, B, C і E використовує менший радіус покриття, деякі частини області будуть не покриті. Потрібно визначити певні датчики як «важливі», щоб відрізнити їх від інших і пріоритизувати в рамках алгоритму.

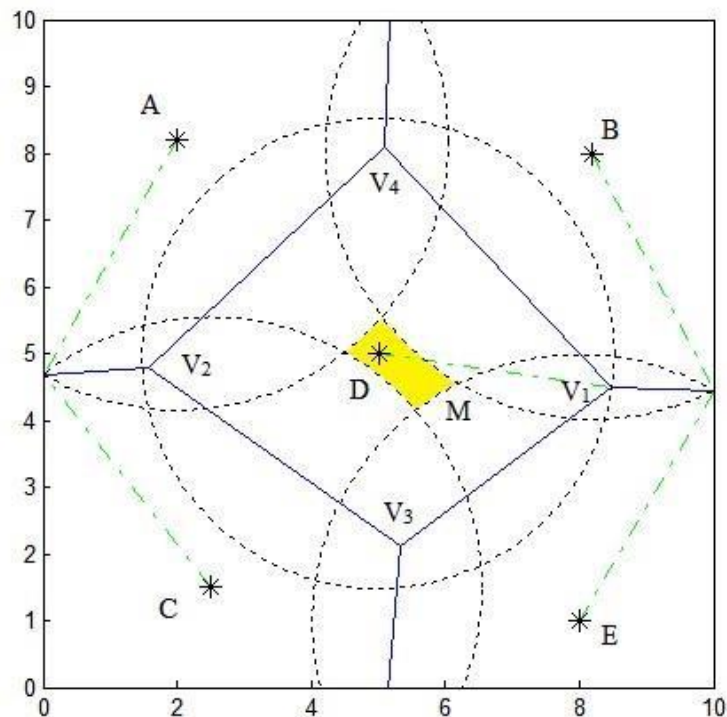


Рисунок 2.4 – Вибір радіусу моніторингу на основі діаграми Вороного.

Визначення 1: Датчик важливий, якщо зменшення будь-якої частини його діапазону моніторингу призводить до якоїсь непокритої підобласті в області моніторингу.

У алгоритмі датчик вважається «важливим», якщо він може бути описаний будь-яким із наступних трьох випадків, показаних на рисунку 2.5:

1. Якщо радіус покриття датчика дорівнює відстані між ним самим та кутом області моніторингу, цей датчик є «важливим» (датчик А на рисунку 2.5)

2. Якщо два сусідніх датчика встановлюють діапазон їх покриття для покриття спільної вершини підобласті на межі області моніторингу, то обидва датчика вважаються «важливими» (датчики В і С на рисунку 2.5)

3. Якщо три сусідні датчики встановлюють діапазон їх покриття таким чином, усі три датчики вважаються «важливими». (Датчик D, E і F на рисунку 2.5)

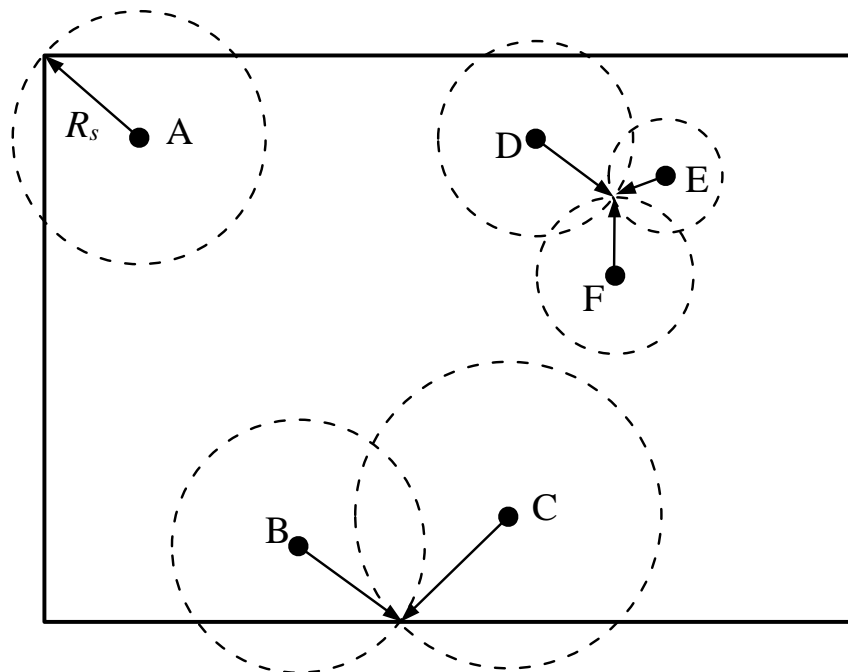


Рисунок 2.5 – Визначення «важливих» датчиків

Алгоритм спрямований на ітераційне перетворення всіх «не важливих» датчиків у «важливі» датчики, зменшуючи радіус моніторингу до мінімально можливого, забезпечуючи економію енергії.

Щоб уникнути проблеми, коли два датчика одночасно зменшують радіус їх моніторингу, що згодом може призвести до створення непокритих підобластей, кожен датчик виконує процес зменшення діапазону індивідуально один за одним на основі набору пріоритетів.

Визначення 2: Пріоритет i -го датчика визначається таким чином:

$$pr_i = \frac{1}{L_i},$$

де L_i - орієнтовний термін служби датчика, який пов'язаний з його енергією, що залишилася, і радіусом покриття. Датчик з більшим радіусом покриття матиме коротший термін експлуатації, а отже, і більший пріоритет (адже всі датчики мають однаковий акумулятор та компоненти).

Це визначення пріоритету полягає в тому, щоб усі датчики споживали енергію рівномірно. Датчик, що ближче до закінчення терміну експлуатації, отримає більш високий пріоритет, щоб скоротити його радіус.

Для того, щоб оцінити ефективність алгоритму, було проведено моделювання з двома різними сценаріями. В обох сценаріях датчики спочатку випадково розміщені. У першому сценарії виконується алгоритм потенційного поля [8], а потім застосовується регулювання діапазону покриття. У другому сценарії безпосередньо виконується алгоритм. Оскільки початкове розгортання є випадковим, алгоритм виконується доволі велику кількість разів, а потім проводиться аналіз середніх результатів. У кожному сценарії розглянуто три випадки з 20, 30 та 40 датчиками, розгорнутими відповідно. Ситуації розглядаються як з постійного, так і з дискретним регулюванням дальності. Дискретні діапазони знаходяться у множині $[1, 2, 3, \dots]$ метрів.

На рисунку 2.6 показані результати обох сценаріїв для постійного регулювання дальності. Оскільки перша задача - досягти 100% покриття, порівнюються результати середнього радіусу моніторингу за алгоритмом (чорні лінії) зі значенням, яке спочатку було необхідним для отримання 100% покриття (сині лінії). Наприклад, у сценарії 1, якщо всі датчики мають однаковий радіус моніторингу, для покриття всієї області моніторингу потрібен радіус моніторингу в середньому 20,9 метрів після 10000 пробігів моделювання. Після того, як виконано процес регулювання дальності, можна покрити область, використовуючи радіус моніторингу 18,2 метра. В обох сценаріях видно, що даний алгоритм може значно зменшити радіус моніторингу порівняно з випадком, коли використовуються датчики однакового радіусу покриття. В середньому радіус моніторингу може бути скорочений на 13% - 20% для сценарію 1, і 55% - 58% для сценарію 2. Покращення значніше у сценарії 2, коли датчики розміщуються випадковим чином без будь-яких попередніх схем переміщення. Це пояснюється тим, що щільність датчиків не може бути збалансована за даних обставин, і якщо спочатку використовується однаковий діапазон моніторингу для всіх датчиків, то для покриття всього сенсорного поля необхідний дуже великий радіус моніторингу.

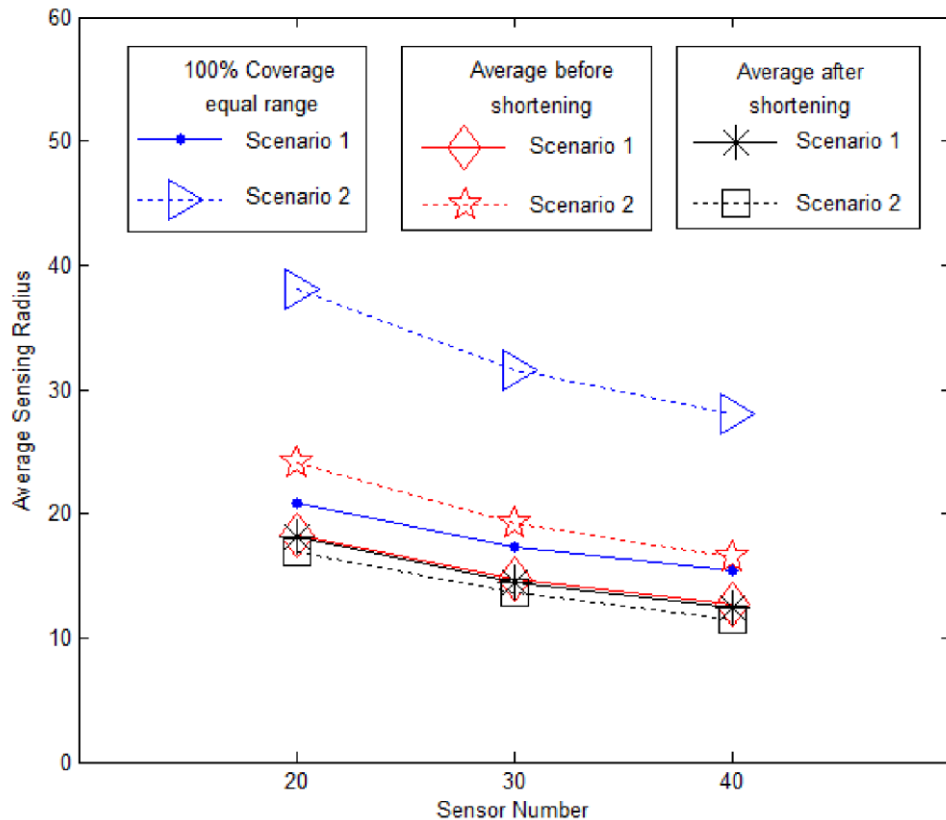


Рисунок 2.6 – Результати моделювання з різною кількістю датчиків

Результати моделювання для датчиків дискретного діапазону дуже схожі на отримані від датчика безперервного регулювання дальності. У випадку дискретного діапазону датчики повинні використовувати в середньому на 0,4 метра більше порівняно з датчиками безперервного регулювання дальності. Теоретично, якщо діапазон моніторингу розподілений рівномірно, різниця між значенням тривалості та округлим дискретним значенням становить в середньому 0,5 метра. У результаті отримано на 0,1 метра менше, оскільки коли для деяких датчиків було округлено радіус чутливості, сусідні датчики змогли використовувати менший радіус чутливості.

Для того, щоб конкретно оцінити процес скорочення, також показано середній радіус моніторингу після кроку 1, що виконується лише діаграмою Вороного перед процесом скорочення. Видно, що навіть коли датчики розподілені досить рівномірно, процес скорочення все ще може призвести до скорочення радіусу чутливості в середньому, щоб заощадити деяку кількість

енергії. За сценарієм 2 процес скорочення має більшу перевагу: може забезпечити 29% - 30% скорочення радіусу моніторингу в середньому для всієї сенсорної мережі.

У БСМ комунікаційні витрати також є важливим джерелом споживання енергії. Тому розглянемо комунікаційні витрати алгоритму.

З моделювання з'ясовано, що в сценарії 1, який застосовує алгоритм після використання потенційного поля, на етапі 2 – фільтрації зайвих датчиків – не знайдено жодного надлишкового датчика. Отже, надмірні датчики відсутні, якщо датчики розподіляються рівномірно, наприклад, після алгоритму потенційного поля. Таким чином, в цьому сценарії не потрібно проходити крок фільтрації надлишкових датчиків. Якщо пропускається крок 2, з кожного датчика буде надіслано на один пакет менше.

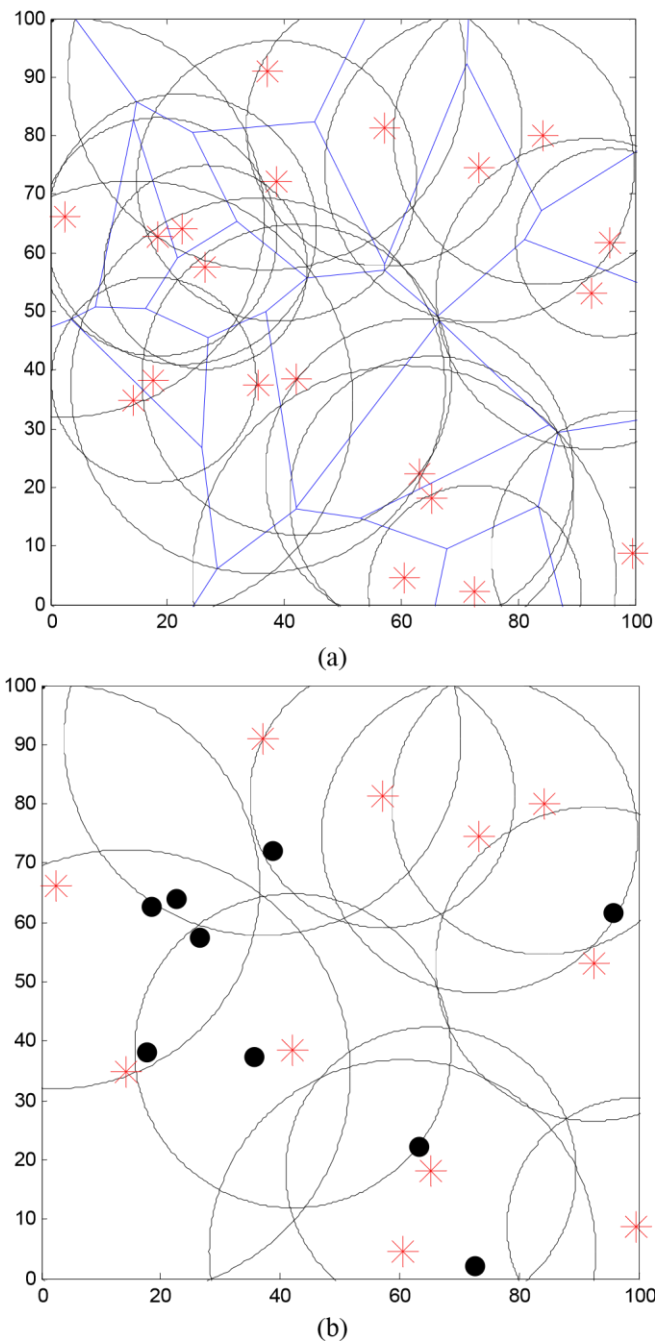


Рисунок 2.7 – Приклад результату для сценарію 2

Щоб перевірити накладні витрати комунікації, обчислюється середня кількість пакетів, що надсилаються для запуску алгоритму. На рисунку 2.8 показано залежність між кількістю датчиків і кількістю пакетів. Видно, що кількість пакетів приблизно лінійна на шкалі сенсорної мережі. Кількість пакетів, відправлених в сценарії 1, що включає алгоритм потенційного поля, менша, ніж у сценарії, що виконує алгоритм безпосередньо після випадкового

розгортання. Це пояснюється тим, що в сценарії, що включає алгоритм потенційного поля, датчики не потрібно перевіряти на надлишковість, і від кожного датчика буде відправлено на один пакет менше.

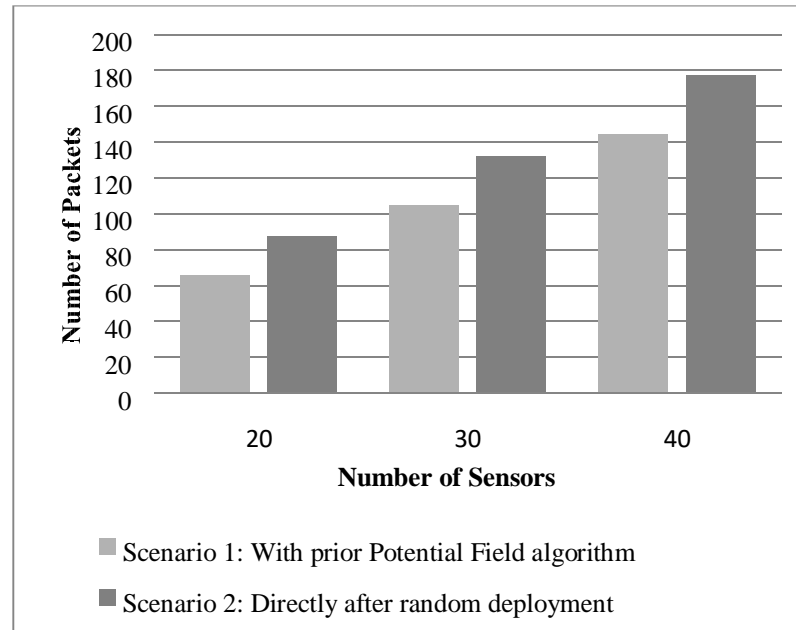


Рисунок 2.8 – Накладні витрати комунікації за алгоритмом

Отже, при використанні датчиків з регульованим діапазоном вирішується проблема отворів в покритті та проблему надлишковості при використанні нерегульованих датчиків однакового діапазону. У порівнянні з аналогічними немасштабованими сенсорними мережами можна заощадити до 50% енергії, тоді як покриття БСМ може досягати 100%. Алгоритм може використовуватися як для рівномірно, так і для нерівномірно розподілених сенсорних мереж. Він має більші показники регулювання діапазону моніторингу, коли датчики розподілені не рівномірно. Крім того, алгоритм може бути об'єднаний з існуючими рішеннями з переміщенням датчиків, плануванням або іншими алгоритмами, пов'язаними з покриттям, для забезпечення кращого покриття та економії енергії.

2.4 Рішення задач багаторазового сферичного покриття площі

2.4.1 Результати рішення модельних задач багаторазового сферичного покриття

Наведемо результати деяких обчислювальних експериментів по багаторазовому покриттю одиничного квадрата. На рисунку 2.9, *a, б, в, г, д, е* зображені оптимальні покриття одиничного квадрата, отримані з точністю $\varepsilon = 0.0001$ алгоритмом 3.2-К [9], для $N = 2, 3, 4, 5, 8, 15$ відповідно. У таблиці 2.1 наведені мінімальні радіуси одноразового покриття в задачі про оптимальне багаторазове с-сферичне покриття для відповідних значень N , отримані з вказаною точністю алгоритмом 3.2-К, у порівнянні з оптимальними рішеннями, отриманими за допомогою алгоритму 3.1[9]. Результати, отримані алгоритмом 3.2-К для зазначених значень N , узгоджуються з результатами, описаними в роботах [10], [11]. Незначна розбіжність в оптимальних значеннях радіусів покриття пояснюється, перш за все, похибкою алгоритму (наближене обчислення компонент узагальненого градієнта цільової функції, дискретизація області і тд.), а також обчислювальної похибкою.

Таблиця 2.1 – Мінімальний радіус одноразового покриття одиничного квадрата

N	$R(\tau_*)$ (алг. 3.2-К)	$R(\tau_*)$ (алг. 3.1)	N	$R(\tau_*)$ (алг. 3.2-К)	$R(\tau_*)$ (алг. 3.1)
2	0,5545	0,5599	9	0,2500	0,2339
3	0,5022	0,5033	10	0,2328	0,2186
4	0,3541	0,3536	11	0,2236	0,2125
5	0,3233	0,3266	12	0,2191	0,2068
6	0,3162	0,3001	13	0,2166	0,1956

7	0,2926	0,2596	14	0,2070	0,1859
8	0,265	0,2596	15	0,1892	0,1807

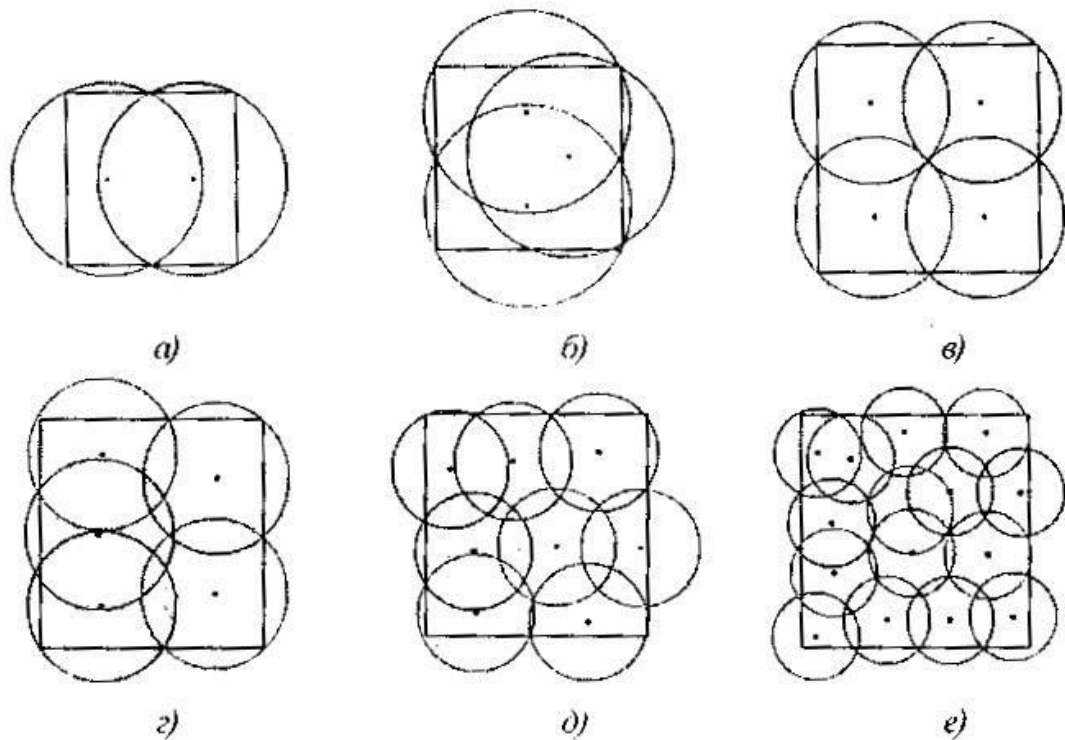


Рисунок 2.9 – Оптимальне одноразове покриття одиничного квадрата, отримане алгоритмом 3.2-К: $\|x\| = \|x\|$

Результати наведених нижче та інших обчислювальних експериментів за чисельним рішенням задачі про оптимальне багаторазове s -сферичне покриття свідчать про наявність властивості багатоекстремальності цільової функції завдання. З різних початкових наближень набору центрів $\tau^{(0)} = (\tau_1^{(0)}, \dots, \tau_N^{(0)})$ можна прийти до різних локальних рішень задачі про оптимальне багаторазове s -сферичне покриття.

На рисунку 2.11 представлено дворазове покриття одиничного квадрата 9-ю заданими центрами, отримане за допомогою алгоритму 3.1-К [9]. На рисунку 3.13 зображено відповідне оптимальне покриття квадрата, отримане алгоритмом 3.2-К.

На рисунку 2.12 представлені результати роботи алгоритму 3.2-К по багаторазовому покриттю різної кратності 16-ю центрами того ж одиничного квадрата. Там же наведені і мінімальні радіуси покриття відповідної кратності.

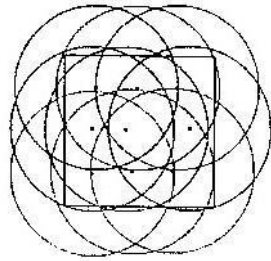


Рисунок 2.10 – 2-кратне покриття квадрата, отримане алгоритмом 3.1-К ($N = 9$, центри фіксовані): $R = 0.5488$

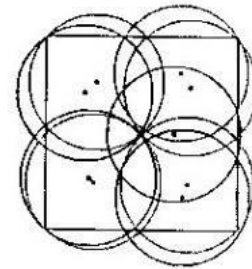
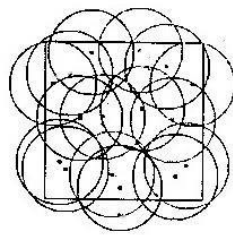
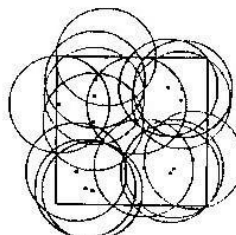


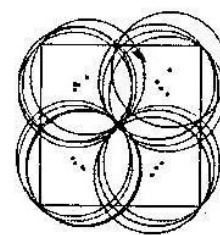
Рисунок 2.11 – Оптимальне 3-кратне покриття квадрата, отримане алгоритмом 3.2-К ($N = 9$): $R = 0.3538$



$k=2; R=0,272855$



$k=3; R=0,333016$



$k=4; R=0,361247$

Рисунок 2.12 – Оптимальне k -кратне покриття одиничного квадрата при заданій кількості $N = 16$ центрів: метрика – евклідова

Інші результати роботи алгоритмів 3.1-К і 3.2-К наведені в [12]. Порівняльний аналіз результатів обчислювальних експериментів по 2-, 3-, 4-кратному покриттю одиничного квадрата дозволив виявити переваги та недоліки алгоритму. Це, перш за все, - залежність результатів обчислень (величини мінімального радіуса сферичного покриття) від вихідних даних і параметрів алгоритму - початкове наближення координат центрів, величини кроку просторової сітки, величини кроку чисельного диференціювання при обчисленні компонент узагальненого градієнта. Серія обчислювальних експериментів показує, що для отримання радіусу покриття, близького до

оптимального радіусу, величину кроку чисельного диференціювання Δ в алгоритмі 3.2-К потрібно узгоджувати з кроком дискретизації області, з кількістю центрів і кратністю покриття. Для усунення цього недоліку пропонується застосовувати алгоритм 3.3-К [9], в якому при обчисленні компонент узагальненого градієнта цільової функції задачі про оптимальне багаторазове с-сферичне покриття використовується k-кратна діаграма Вороного. Практика розрахунків показує, що найкращі результати багаторазового покриття виходять, якщо спочатку при не дуже густій просторовій сітці знайти оптимальні розташування центрів за допомогою алгоритму 3.2-К, а потім уточнити координати цих центрів за допомогою алгоритму 3.3-К.

У таблиці 2.2 наведені результати вирішення задачі оптимального дворазового покриття одиничного квадрата, отримані в роботі [13], і для порівняння - отримані за допомогою описаного вище алгоритму, як із застосуванням формул чисельного диференціювання, так і з використанням багаторазових діаграм Вороного.

Таблиця 2.2 – Мінімальний радіус дворазового покриття одиничного квадрата

N	$R(\tau_*)[21]$	$R(\tau_*)$, отримане з допомогою	
		алгоритмів 3.2-К	алгоритмів 3.3-К
11	0.31280	0.31819	0.3164
13	0.29106	0.29740	0.2970008
15	0.26650	0.27869	0.278076
17	0.25372	0.25927	0.2657
19	0.22766	0.23753	0.236849
21	0.21601	0.21351	0.225



Рисунок 2.13 – 2-кратне оптимальне покриття одиничного квадрата, $N = 19$, отримані із застосуванням: зліва - чисельного диференціювання; праворуч - двократної діаграми Вороного

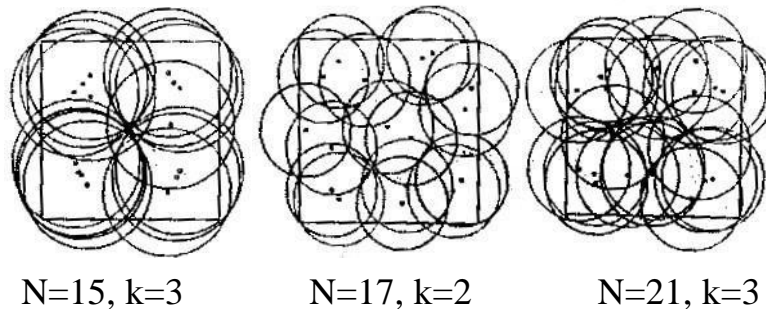


Рисунок 2.14 – k -кратне покриття одиничного квадрата, $\|x\| = \|x\|_2$

Як неважко помітити, результати вирішення деяких задач оптимального багаторазового покриття свідчать про те, що при певних значеннях N і k оптимальне розташування центрів таке, що кілька центрів можуть бути розташовані дуже близько один до одного (див. рисунки 2.13, 2.14), що на практиці не завжди піддається реалізації або небажано. Тому далі математична модель задачі про оптимальне багаторазове сферичне покриття буде уточнена деякою регулюючою умовою, що відповідає за неможливість "злипання" центрів.

2.4.2 Безперервна задача оптимального багаторазового сферичного покриття множин з обмеженнями

Нехай Ω - обмежена, замкнута множина в просторі E_n , множина $B(\tau_i, R) = \{x \in E_n: c(x, \tau_i) \leq R\}$ - c -сфера радіусу R з центром в точці τ_i із E_n , $c(x, \tau_i)$ - деяка квазіметрика (евклідова, манхеттенська). Потрібно визначити величину радіуса оптимального покриття: $R(\tau^N) = \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i)$, (2.1)

$$x \in \Omega \quad i=1, \dots, N$$

і вектор $\tau_*^N = (\tau_1^*, \dots, \tau_N^*)$, на якому досягається значення $\hat{R}(\tau_*^N)$, при цьому для кожної точки $x \in \Omega$ виконується включення

$$x \in \bigcap_{j=1}^k B(\tau_{ij}, R), \quad k \leq 1 \leq N, \quad i_j \in \{1, 2, \dots, N\}, \quad (2.2)$$

а центри задовольняють умовам:

$$\min_{(i,j): i < j} c(\tau_i, \tau_j) \geq \sigma, \quad i = 1, 2, \dots, N \quad (2.3)$$

де $\sigma > 0$ - задана величина.

Якщо ввести до розгляду множину Λ^k_N :

$$\Lambda^k_N = \{\lambda = (\lambda_1, \dots, \lambda_N): \lambda_i = 0 \vee 1, \quad i = \overline{1, N}; \sum_{i=1}^N \lambda_i = k\},$$

і на множині Ω визначити вектор-функцію $\lambda(\cdot)$ зі значеннями в

$$\Lambda = \{\lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)): \lambda(x) \in \Lambda^k_N, \quad \forall x \in \Omega\},$$

то задача про мінімальне k -кратне c -сферичне покриття з обмеженнями на розміщення центрів сфер математично може бути сформульована так:

потрібно знайти величину

$$R(\tau_1, \dots, \tau_N) = \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x), \quad (2.4)$$

а також вектор-функцію $\lambda^*(x) \in \hat{\Lambda}$ і вектор $\tau_*^N = (\tau_1^*, \dots, \tau_N^*) \in \Omega^N \subset E_n^N$, при яких в (3.3.17) досягається нижня межа, і виконуються умови (2.3).

Якщо в задачі (2.4) не цікавитися, якими саме c -сферами покривається кожна точка $x \in \Omega$, то вектор-функцію $\lambda^*(\cdot)$ зліва в рівності (2.4) можна опустити, що і будемо далі припускати.

2.4.3 Опис методу і алгоритму вирішення задачі

Для вирішення задачі (3.3.17), (3.3.16) пропонується наступний підхід [44]. Для врахування додаткових умов (3.3.16) в задачі багаторазового покриття вводиться до розгляду штрафна функція вигляду

$$P(\tau_1, \dots, \tau_N) = M \sum_{i=1}^{N-1} \sum_{j=i+1}^N \max^2(0, \sigma - c(\tau_i, \tau_j)),$$

де $M \gg 0$ – коефіцієнт штрафу, і від задачі (3.3.17), (3.3.16) відбувається перехід до задачі:

$$R_M(\tau^N) \rightarrow \min_{(\tau^1, \dots, \tau^N) \in \Omega} R(\tau_1, \dots, \tau_N) + P(\tau_1, \dots, \tau_N), \quad (2.5)$$

де

$$R_M(\tau^N) = R(\tau_1, \dots, \tau_N) + P(\tau_1, \dots, \tau_N) = \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x) +$$

$$P(\tau_1, \dots, \tau_N).$$

Чисельний алгоритм вирішення задачі (2.5), наведений далі, базується на методі узагальненого градієнтного спуску з розтягуванням простору в напрямку різниці двох послідовних узагальнених градієнтів (r-алгоритм Шора). При цьому j -ту компоненту N -мірного вектора узагальненого градієнта

$$g^R(\tau^N) = (g^{\tau_1}(\tau^N), \dots, g^{\tau_j}(\tau^N), \dots, g^{\tau_N}(\tau^N)) \quad (2.6)$$

функції $R_M(\tau_1, \dots, \tau_N)$ в точці $\tau^N = (\tau_1, \dots, \tau_N)$ будемо обчислювати за формулою:

$$g_{\tau_j}(\tau^N) = g_{R\tau_j j V}(\tau^N) + M \sum_{N, i \neq j} \max(0, \sigma - c(\tau_i, \tau_j)) \frac{\partial c}{\partial \tau_j}(\tau_j, \tau_i), \quad (2.7)$$

де $g_{R\tau_j j V}(\tau^N) \in G_{R\tau_j j V}(\tau^N)$ (див. [9]).

Представим наближений алгоритм вирішення задачі про мінімальне k -кратне s -сферичне покриття з розміщенням центрів сфер так, щоб мінімальна відстань між кожними двома центрами біла не менше заданої величини σ , складовою частиною якого є алгоритм 3.1-К, що дозволяє знаходити радіус N кіл, що утворюють k -кратне s -сферичне покриття множини при фіксованих центрах. Приведемо та проаналізуємо результати деяких обчислювальних експериментів.

Сформулюємо алгоритм вирішення задачі оптимального багаторазового покриття одиничного n -мірного куба $\Omega = [0, 1] \times [0, 1] \dots \times [0, 1]$. Для спрощення позначень в алгоритмі замість вектора τ^N будемо писати τ .

Якщо область Ω має складну форму, то заключимо Ω в n -мірний паралелепіпед Π , сторони якого паралельні осям декартової системи координат, ввівши допоміжну функцію $\rho(x)$, визначену на Π , таку, що

$$\rho(x) = \begin{cases} 0 & \text{для } x \in \Pi/\Omega \end{cases}$$

1 для $x \in \Omega$

Тоді в задачі багаторазового сферичного покриття площі під функцією $c(x, \tau_i)$ будемо розуміти функцію $c(\tau_i, \tau_j)\rho(x)$, визначену на паралелепіпеді Π і співпадаючу з $c(\tau_i, \tau_j)$ на множині Ω .

Алгоритм 3.4-К

Попередній етап.

Паралелепіпед Π покриваємо прямокутною сіткою з кроком $\Delta h_j, j = 1, \dots, n$. Множину вузлів прямокутної сітки на множині Ω позначимо Ω .

Задаємо початкове положення центрів сфер $\tau^{(0)} = (\tau_1^{(0)}, \dots, \tau_N^{(0)})$.

Знаходимо за цими центрами величину радіуса покриття

$R(\tau^{(0)}) = \max_{x \in \Omega} \max_{\lambda(x) \in \Lambda, i=1, N} c(x, \tau_i^{(0)})$, використовуючи алгоритм 3.1-К. При

цьому для кожного $i = 1, \dots, N$ обчислюємо значення функції:

$$R_i(\tau^{(0)}) = \max_{x \in T_m} \max_{x \in V(\bar{\tau}_m)} c(x, \tau_i^{(0)}).$$

Обчислюємо величину $\bar{R}_M(\tau^{(0)}) = \hat{R}(\tau_1^{(0)}, \dots, \tau_N^{(0)}) + P(\tau_1^{(0)}, \dots, \tau_N^{(0)})$.

Будуємо субградієнтну множину $G_{R^{ti}_w}(\tau^{(0)}) = \bar{c} \bar{d} \{g_{c^{ti}}(\bar{x}, \tau^{(0)})$,

$\bar{x}: c(\bar{x}, \tau_i) = R_i(\tau_i^{(0)})\}, i = 1, \dots, N$, де $g_{c^{ti}}(\bar{x}, \tau^{(0)})$ – узагальнений градієнт функції

$c(x, \tau_i)$ в точці $\tau^{(0)}$. Вибираємо вектор $g_{\bar{R}_M}(\tau^{(0)})$, задаємо початковий пробний

крок g -алгоритму $h_0 > 0$.

Перший крок алгоритму проводимо за формулою:

$$\tau^{(1)} = P_{\Omega}(\tau^{(0)} - h_0 g_{R}(\tau^{(0)})), P_{\Omega} -$$

оператор проектування на множину Ω .

Переходимо до другого кроку.

Нехай в результаті обчислень після m , $m = 1, 2, \dots$, кроків алгоритму отримано певний вектор $\tau^{(m)} = (\tau_1^{(m)}, \dots, \tau_N^{(m)})$.

Опишемо $(m+1)$ -й крок алгоритму.

Обчислюємо за відомими центрами $\tau_1^{(m)}, \dots, \tau_N^{(m)}$ величину

$$R(\tau^{(m)}) = \sup_{x \in \Omega} \min_{\lambda(x) \in \Lambda^k_N} \max_{i \in \{1, N\}} c(x, \tau_i^{(m)}),$$

використовуючи алгоритм 3.1-К вирішення задачі пошуку радіуса N кіл, що утворюють k -кратне c -сферичне покриття заданої множини Ω . При цьому для кожного $i = 1, \dots, N$ обчислюємо значення функції:

$$R_i(\tau^{(m)}) = \max_{x \in T_m} \max_{V(\bar{\tau}_m)} c(x, \tau_i^{(m)}).$$

2. Будуємо субградієнтну множину $G_{R^i}(\tau^{(m)})$. Вибираємо вектор

$$g_{RM}(\tau^{(m)}).$$

3. Проводимо $(m+1)$ -й крок r -алгоритму в H -формі [14], ітераційна формула якого має вигляд:

$$\tau^{(m+1)} = P_{\Omega} \left(\tau^{(m)} - h_m \frac{\sqrt{\frac{m+1}{\bar{R}}}}{\sqrt{(H_{m+1} g_{\bar{R}}(\tau^{(m)}), g_{\bar{R}}(\tau^{(m)}))}} H g(\tau^{(m)}) \right), \text{ де } H_{m+1} - \text{ матриця}$$

розтягування простору з коефіцієнтом α в напрямку

різниці двох послідовних узагальнених градієнтів, що має вигляд

$$H_{m+1} = H_m + \frac{(\alpha - 1)}{(H_m \xi_m, \xi_m) - (g_R(\tau^{(m-1)}), \xi_m)}$$

Якщо через округлення рахунку H_{m+1} перестає бути позитивно визначеною, заміняємо її одиничною матрицею.

Крок h_m обираємо з умови:

$$R(\tau, h > 0) = h \frac{R}{\sqrt{(H_{m+1} g_{\bar{R}}(\tau^{(m)}, g_{\bar{R}}(\tau^{(m)})) \min_{H_{m+1} g(\tau^{(m)})}}$$

4. Якщо умова

$$\|\tau^{(m+1)} - \tau^{(m)}\| \leq \varepsilon, \varepsilon > 0, \quad (2.8)$$

не виконується, то переходимо до $(m+2)$ -го кроку алгоритму, в іншому випадку – перехід до п.5

5. Припустимо $\tau_* = \tau^{(l)}$, де l – номер ітерації, на якій виконалась умова (3.3.21) завершення роботи алгоритму.

6. Обчислюємо значення мінімального радіуса покриття за формулою

$$R(\tau_*) = \max_{x \in \Omega} \lambda(\min_{x \in \Lambda_{kN}} \max_{i \in \bar{1}, N} c(x, \tau_{*i}) \lambda_i(x))$$

за допомогою алгоритму 3.1-К.

Алгоритм 3.4-К описано.

При реалізації описаного підходу постає питання про вибір штрафного коефіцієнта M . З одного боку, штраф повинен бути достатньо великим, щоб отримане значення $R(\tau_*^N)$ було якнайближче до оптимального, а вектор τ_*^N – якнайближче до умовного оптимума. З іншої сторони, занадто велике значення штрафного коефіцієнта може викликати обчислювальні труднощі при знаходженні оптимального рішення в задачі без обмежень. Тому метод штрафу краще використовувати в ітераційній формі, починаючи вибір коефіцієнта M зі значення m_1 , не надто високого, і вирішуючи задачу:

$$R(\tau^N; m_1) = R(\tau_1, \dots, \tau_N) + m_1 \sum_{i=1}^N \sum_{j=i+1}^N \max^2(0, \sigma - c(\tau_i, \tau_j)) \rightarrow_{(\tau_1, \dots, \tau_N) \in \Omega^N} .$$

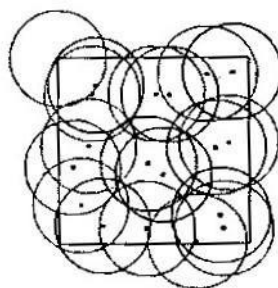
Якщо при цьому отримане рішення $\tau^N(m_1)$ буде хорошим наближенням для оптимума, то обчислення закінчуються. В іншому випадку штраф, що відповідає порушенню умов задачі, недостатньо високий, і, збільшуючи його в декілька разів (наприклад, вважаючи $m_2 = 7m_1$), слід вирішити нову задачу без обмежень і т.д.

2.4.4 Аналіз результатів обчислювальних експериментів

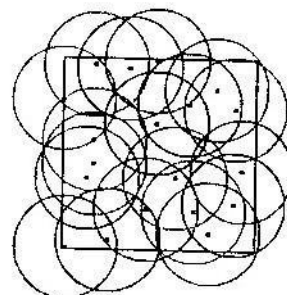
Приведемо результати вирішення декількох задач багаторазового сферичного покриття одиничного квадрата із E_2 при обмеженнях на відстані між центрами, у випадку, коли функція $c(x, \tau_i)$ задається за формулою:

$$c(x, \tau_i) = \|x - \tau_i\|_2 = \sqrt{\sum_{j=1}^n (x^j - \tau_i^j)^2} .$$

На рисунку 2.15 були представлені результати дворазового покриття одиничного квадрата 19-ю колами, що отримані при вирішенні задачі багаторазового сферичного покриття площі без урахування обмежень. Як видно, деякі центри практично співпадають. Мінімальне дворазове покриття 19-ю колами квадрата, але вже при умові розміщення центрів на допустимій відстані один від одного, представлено на рисунку 2.15 а), б)



R=0,2573



R=0,27115

а)

б)

Рисунок 2.15 – Оптимальне дворазове покриття 19-ю колами: а) при умовах (3) $\sigma = 0.05$; б) при обмеженнях (3) $\sigma = 0.15$

На рисунку 2.16 показано триразове оптимальне покриття при $N=15$. Очевидно, при реалізації такої моделі центри кіл, що утворюють багаторазове покриття, знаходяться також на деякій допустимій відстані один від одного.

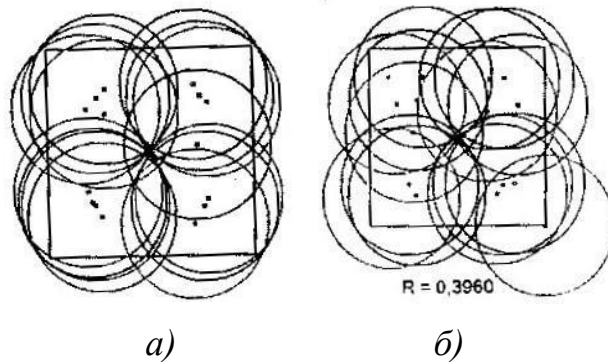


Рисунок 2.16 – Триразове оптимальне покриття одиничного квадрата: а) без обмежень; б) з контролем мінімальної відстані між центрами

Зауваження. При побудові математичних моделей задач оптимального покриття, що утворюються на практиці, обмеження на розміщення центрів куль, що покриваються, можуть мати і інший вигляд. Наприклад, можна вимагати, щоб центри були розміщені щомога рівномірніше в області (див. рисунок 2.17), або щоб ні один із них не потрапив в деякий заборонений її участок.

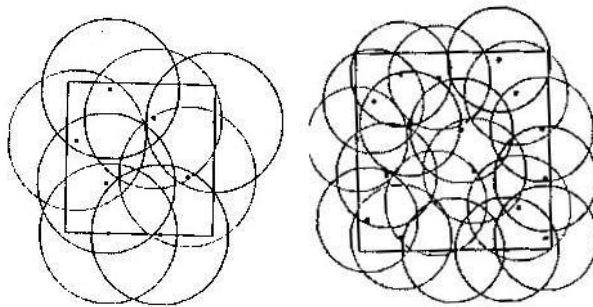


Рисунок 2.17 – Дворазове оптимальне покриття одиничного квадрата з рівномірним розміщенням центрів в області

У розділі представлені математичні моделі і методи рішення безперервних задач покриття обмеженої області сферами мінімального радіуса. Розглянуто варіанти одноразового і багаторазового покриття області.

Для безперервної задачі про оптимальне s -сферичне покриття компактної множини Ω з E_n заданим числом центрів з мінімальним радіусом запропоновано і обґрунтовано алгоритм вирішення задачі багаторазового сферичного покриття площі, заснований на використанні методу узагальненого градієнтного спуску з розтягуванням простору в напрямку різниці двох послідовних узагальнених градієнтів (r -алгоритму Н.З. Шора).

Розглянуто властивості даної задачі про оптимальне s -сферичне покриття множини Ω , встановлені її схожість і відмінність із задачею оптимального розбиття цієї множини на непересічні підмножини з розміщенням центрів підмножин, що дозволяє в тих випадках, коли це доцільно, зводити рішення задачі про покриття до задачі розбиття.

Розглянуті алгоритми: 1) дозволяють одночасно знаходити всі центри оптимального покриття; 2) не залежать від розмірності простору, якому належить множина, що покривається; 3) не пов'язані з геометричними особливостями множини, що покривається; 4) можуть бути застосовані до вирішення дискретних задач оптимального покриття.

Алгоритми, взагалі кажучи, призводять до локальних оптимальних рішень задач про оптимальне сферичне покриття. Хоча, в деяких чисельних експериментах, проведених для певних метрик, алгоритми приводили до глобальних оптимальних рішень.

Безперервні задачі багаторазового сферичного покриття узагальнені на випадок наявності додаткових умов, пов'язаних з можливим розташуванням розміщуваних центрів сфер. Розглянута програмна реалізація алгоритму

вирішення зазначених задач, заснованого на дискретизації області, що покривається, та застосуванні для вирішення задач недиференційованої оптимізації r -алгоритму Шора. При цьому для врахування обмежень в задачах залучається апарат штрафних функцій, а для наближеного обчислення компонент узагальненого градієнта розширеної цільової функції застосовуються k -кратні діаграми Вороного.

3 ПРОТОКОЛИ МАРШРУТИЗАЦІЇ БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖ

3.1 Фактори вибору протоколу маршрутизації БСМ

Тривалість функціонування і безперебійність роботи мережі є двома основними факторами, які повинні бути в обов'язковому порядку враховані при виборі протоколу маршрутизації БСМ. Однак існує і безліч інших факторів, які також повинні бути розглянуті. До них відносяться безвідмовність, енергоспоживання, масштабованість, агрегування даних, якість обслуговування, спосіб розгортання вузлів на території, спосіб передачі інформації і представлення даних.

Безвідмовність є дуже важливим фактором в БСМ, так як сенсорні вузли досить часто схильні до збоїв та відмов по безлічі різних причин, таких як вплив навколишнього середовища, виснаження джерел енергії або навіть фізичні пошкодження. Однак подібні збої в роботі вузлів і пошкодження мережі ні при якому разі не повинні призводити до втрати даних і ставити під загрозу її повноцінне функціонування. Щоб уникнути подібних сценаріїв, протокол маршрутизації повинен мати можливість перенаправляти пакети в обхід несправного вузла при виявленні його відмови або збою в його роботі.

Управління енергоспоживанням сенсорних вузлів відбувається за рахунок оптимальної комп'ютерної обробки і передачі даних. Витрата енергії також може бути мінімізована за допомогою ефективного алгоритму маршрутизації. Зменшення витрат енергії при передачі інформації може досягатися зменшенням кількості пакетів, що проходять через мережу, обмеженням їх розміру, а також шляхом обмеження кількості завдань, що виконуються кожним окремим вузлом мережі. Втрати енергії в стані очікування можуть бути зведені до мінімуму за допомогою реалізації ефективного алгоритму планування сну.

Масштабованість може бути визначена як можливість мережі адаптуватися до збільшення розмірів мережі. Кількість вузлів в БСМ може коливатися від пари одиниць до декількох тисяч. Протокол маршрутизації повинен мати можливість ефективно працювати з будь-якою кількістю вузлів і підтримувати динамічне приєднання нових вузлів без необхідності переустановлення всієї мережі.

Агрегація даних допомагає запобігти дублюванню даних в мережі. Так як БСМ мають високу щільність розміщення, сенсорний вузол може отримувати ідентичні дані від декількох сусідніх вузлів одночасно. Подібні багаторазові передачі даних також призводять до втрат енергії. Агрегування даних може бути реалізовано за допомогою усереднення, заборони дублювання даних або методу максимумів-мінімумів. Агрегація даних з використанням максимумів-мінімумів і усереднення також називається узагальненням даних. У такому випадку дані, одержувані від сусідніх вузлів, зіставляються, і тільки максимальне, мінімальне або середнє значення з усіх отриманих передаються далі в одному повідомленні.

Заборона дублювання даних - це найпростіший вид агрегування даних, при якому вузол відкидає дані, ідентичні отриманим раніше. Агрегування даних може зменшити кількість повідомлень, що передаються через мережу, але даний підхід може призвести до виникнення затримок, так як вузли повинні чекати отримання даних від усіх сусідніх вузлів до передачі об'єднаних даних. Якість обслуговування - це фактор, що залежить від затримок і надійності мережі. Виникнення затримок небажано при будь-якому типі мережі, особливо в разі наявності в ній критично важливих даних, передача яких повинна відбуватися в реальному масштабі часу.

Надійність може бути визначена як здатність мережі працювати ефективно при будь-яких умовах. Всі вищеописані чинники прямо або побічно впливають на надійність і безвідмовність роботи мережі, отже, це найбільш

важливі параметри, які слід враховувати при виборі протоколу маршрутизації БСМ.

3.2 Класифікація протоколів маршрутизації БСМ

Класифікація протоколів маршрутизації БСМ проводиться в залежності від принципу пошуку маршруту, структури мережі та ініціатора установки з'єднання. Класифікація протоколів маршрутизації БСМ приведена на рисунку 3.1.

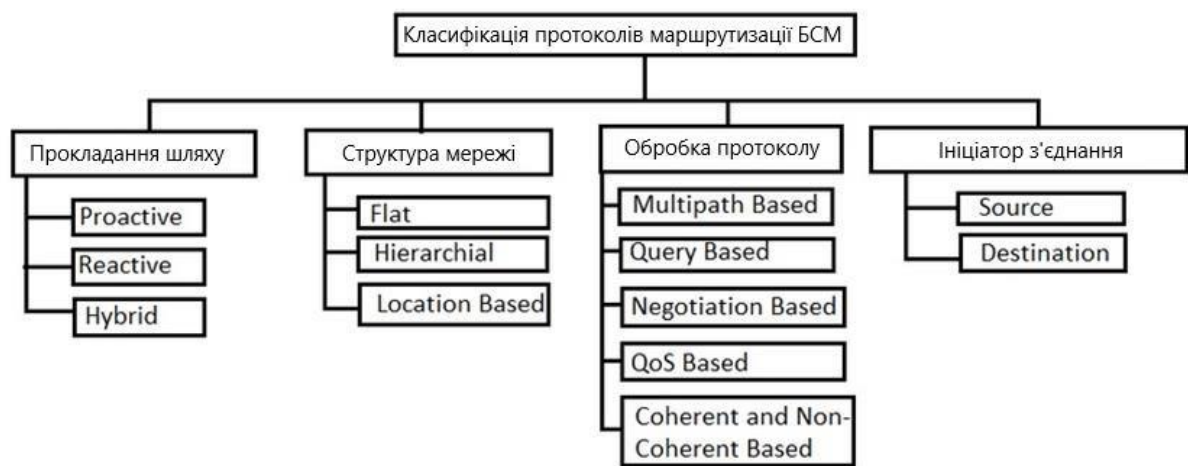


Рисунок 3.1 – Класифікація протоколів маршрутизації БСМ

Залежно від принципу пошуку шлях маршрутизації мережі може бути прокладений одним з трьох способів, а саме: проактивним (proactive), реактивним (reactive) або гібридним (hybrid) [15].

При використанні проактивних протоколів всі маршрути обчислюються заздалегідь і зберігаються в таблиці маршрутизації кожного вузла. Таким чином, зміни маршруту повинні бути поширені по всій мережі. Проактивні протоколи не є переважними в БСМ з високою щільністю розміщення вузлів, оскільки при подібному підході збільшуються накладні витрати для підтримки актуальності таблиць маршрутизації в кожному з сенсорних вузлів мережі. При використанні реактивних протоколів маршрути обчислюються тільки в

момент їх необхідності. Гібридні протоколи є комбінацією проактивних і реактивних протоколів.

БСМ також можуть бути класифіковані за типом структури мережі. За цим принципом виділяють плоскі (flat), ієрархічні (hierarchical) і безпосередні (location based) протоколи. У плоских протоколах маршрутизації всі вузли в мережі мають однакові можливості і можуть в рівній мірі брати участь в задачах маршрутизації. Однак найближчі до базової станції вузли, як правило, задіяні в більшій мірі, так як пакети всій мережі направляються через них.

В ієрархічних протоколах мережа розбивається на кластери, в кожному з яких є свій маршрутизуючий вузол. При використанні даного підходу всі елементи кластера відправляють повідомлення тільки до маршрутизуючого вузла, який, в свою чергу, пересилає всі повідомлення свого кластера вузлушлюзу. При використанні безпосередніх комунікаційних протоколів сенсорні вузли БСМ посилають пакети безпосередньо до базової станції, отже, повинні розташовуватися щодо неї в області досяжності прийому і передачі інформації.

З'єднання в БСМ може встановлюватися як джерелом (source), так і приймачем (destination) даних. У протоколах з ініціалізацією з'єднання джерелом, вузли передають дані базової станції відразу після проведення вимірювань. Такі протоколи використовують тимчасовий або подієвий підхід складання і відправки звітності звіту даних. У свою чергу, при використанні протоколів, в яких з'єднання ініціалізується приймачем, використовується запитовий підхід до звітності, коли сенсорні вузли БСМ відправляють інформацію у відповідь на запити, одержувані від приймача даних [16].

Більшість протоколів маршрутизації можна розділити на плоскі, ієрархічні або гібридні на найвищому рівні, які, в свою чергу, можуть згодом поділятися на протоколи з ініціалізацією з'єднання джерелом або приймачем. Також, в БСМ найчастіше використовується реактивний спосіб вибору шляху маршрутизації мережі.

4 АЛГОРИТМ РОЗТАШУВАННЯ ДАТЧИКІВ БСМ

Визначення оптимального розташування сенсорних вузлів на території, що відслідковується, вкрай необхідно для енергоефективного та економічно вигідного розгортання БСМ. Позиції датчиків на контрольованій ділянці повинні забезпечувати достатній ступінь покриття при збереженні тривалого терміну служби, що може бути здійснено шляхом використання ефективного механізму планування по організації і обмеження кількості задіяних сенсорних вузлів.

У даній роботі передбачається залучення алгоритму розташування вузлів БСМ, відомого як territorial predator scent marking algorithm (TPSMA). Даний алгоритм використовує біологічно засновану техніку оптимізації покриття території датчиками, яка імітує поведінку територіальних хижаків в процесі мічення (маркування) контрольованої території [17]. Алгоритм TPSMA використовує цільову функцію забезпечення максимального покриття і передбачає максимально здійсненне збільшення тривалості функціонування мережі.

4.1 Поведінковий алгоритм TPSMA

Деякі види хижаків, такі як тигри, собаки та ведмеді, можуть бути визначені як територіальні через наявність у них інстинктивного поведінкового акту систематичного захисту певної території від тварин інших видів. Територія вибирається виходячи з певних факторів, з яких найбільш частим і значущим є контроль розподілу харчових ресурсів. Більшість територіальних хижаків використовують свій запах для позначення меж

контрольованої території, що є міткою як для інших хижаків, так і для членів їх популяції.

Більшість територіальних хижаків використовують свій запах для позначення меж контрольованої території, що є міткою як для інших хижаків, так і для членів їх популяції.

Подібний спосіб комунікації дозволяє цим тваринам залишати повідомлення, що володіють відносною довговічністю. Крім того, він також може бути використаний в нічний час, під землею, або в густій рослинності.

Запахові мітки можуть полегшити комунікацію тварин, виконуючи чотири різні функції, такі, як допомога в розпізнаванні родичів, репродуктивна, тимчасова або просторова сигналізація, а також захист ресурсів. Зіставлення запахів дозволяє хижакам відрізнити членів своєї популяції, що мешкають в межах контрольованої території, від самозванців, тим самим знижуючи ймовірність несподіваних сутичок з іншими тваринами.

4.2 Розміщення вузлів БСМ на основі алгоритму TPSMA

Поведінковий алгоритм TPSMA може бути взятий за основу при розробці методики розміщення сенсорних вузлів, з метою досягнення таких цілей, як максимальне покриття та мінімальне енергоспоживання.

В даному випадку маркування області охоплення датчиками вузла БСМ ґрунтується на поведінці хижаків, що помічають запахом контрольовану територію, виходячи з певних факторів, таких, як розподіл харчових ресурсів [16]. Сенсорний вузол, в свою чергу, ідентифікує відстежувану територію, ґрунтуючись на відмічених територіях, по аналогії з зіставленням запахів хижаками в поведінковому алгоритмі TPSMA.

На рисунку 4.1 наведено псевдокод, що описує схему розміщення сенсорних вузлів на основі алгоритму TPSMA з цільовою функцією досягнення максимального коефіцієнта покриття.

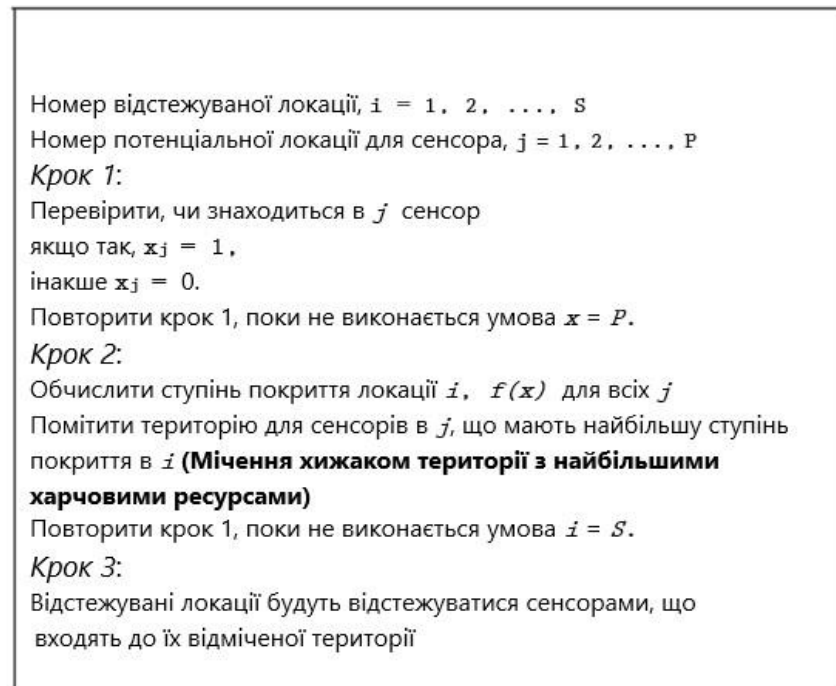


Рисунок 4.1 – Псевдокод розміщення сенсорних вузлів на основі TPSMA

4.3 Модель проектованої БСМ

На рисунку 4.2 представлено схематичне зображення контрольованої території розміром 100х100 метрів, розділеної на 16 рівних локацій.

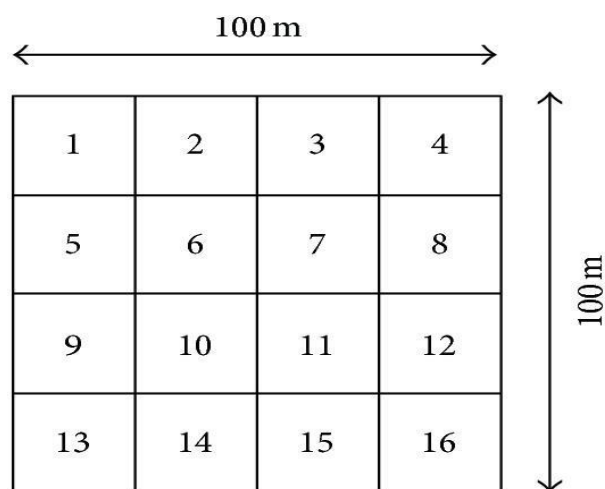


Рисунок 4.2 – Контрольований простір

В межах окремої локації може бути розміщено не більше одного вузла БСМ, а центрі кожної області розташовується точка доступу. Кожна локація може контролюватися як розташованим всередині сенсорним вузлом, так і вузлами з сусідніх контрольованих областей.

У таблиці 4.1 наведено покриття всіх 16 локацій відповідними сенсорними вузлами.

Таблиця 4.1 – Контрольовані локації відповідно до вузлів БСМ

Контрольована локація	Сенсорні вузли
1	1, 2, 5, 6
2	1, 2, 3, 5, 6, 7
3	2, 3, 4, 6, 7, 8
4	3, 4, 7, 8
5	1, 2, 5, 6, 9, 10
6	1, 2, 3, 5, 6, 7, 9, 10, 11
7	2, 3, 4, 6, 7, 8, 10, 11, 12
8	3, 4, 7, 8, 11, 12
9	5, 6, 9, 10, 13, 14
10	5, 6, 7, 9, 10, 11, 13, 14, 15
11	6, 7, 8, 10, 11, 12, 14, 15, 16
12	7, 8, 11, 12, 15, 16
13	9, 10, 13, 14
14	9, 10, 11, 13, 14, 15
15	10, 11, 12, 14, 15, 16
16	11, 12, 15, 16

Число сенсорних вузлів, розгорнутих в даному контрольованому просторі, може бути різним. Мінімальна кількість вузлів мережі можна визначити, використовуючи наступне рівняння:

$$\text{Кількість вузлів} = \frac{A}{3\sqrt{3}R_S^2/2},$$

де A - площа контрольованого простору, а R_S - радіус моніторингу сенсорного вузла. Мінімальна кількість вузлів БСМ, необхідних для цієї моделі, становить приблизно 10 вузлів. Як згадувалося раніше, в межах кожної контрольованої локації може бути розташоване не більш одного вузла. Таким

чином, число сенсорних вузлів, що розглядаються в даній роботі, варіюється від 10 до 16.

4.4 Результати моделювання

Моделювання проводилося за допомогою програми Matlab для демонстрації працездатності запропонованої схеми. У таблиці 4.2 наведені параметри моделювання з декількома стаціонарними сенсорними вузлами, розгорнутими в зоні моніторингу псевдовипадковим чином.

Таблиця 4.2 – Параметри моделювання

Parameter	Value
Number of sensor nodes	10 to 16
Monitoring area	100 m × 100 m
Sensing range, R_s	20 m
Number of surveillance locations	16
Coverage threshold, C_{th}	0.9
Uncertainty in sensor detection, R_e	10 m
α_1	1
α_2	0
β_1	1
β_2	0.5

Тест збіжності для 10 і 13 сенсорних вузлів показує, що алгоритм TPSMA сходиться після 200 ітерацій. Таким чином, передбачається, що число ітерацій, необхідних для будь-якого іншого числа сенсорних вузлів також становить 200.

Як і очікувалося, з ростом кількості датчиків коефіцієнт покриття збільшується. Однак, алгоритм TPSMA може забезпечити достатнє покриття контрольованої території при використанні меншої кількості сенсорних вузлів. Це пов'язано з тим, що цільова функція алгоритму TPSMA, запропонованого в

даній атестаційній роботі, спрямована на досягнення максимального коефіцієнту покриття.

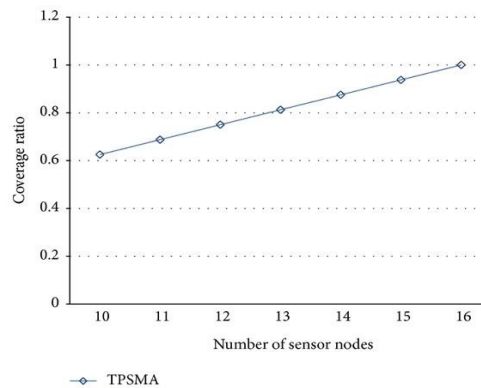


Рисунок 4.3 - Графік коефіцієнта покриття БСМ в залежності від різного числа вузлів

4.5 Розташування сенсорних вузлів

Рисунок 4.4 ілюструє розташування в межах контрольованої території десяти вузлів.

За допомогою позначок у формі червоних зірок на рисунках відображені місця розташування точок доступу в кожній з контрольованих локацій. Відмітки у формі чорних зірок вказують реальне розташування розміщених випадковим чином вузлів БСМ. Тоді як за допомогою синіх кіл схематично зображено область покриття кожного сенсорного вузла.

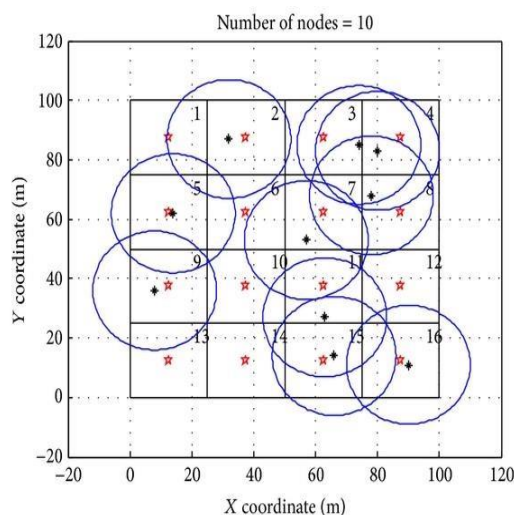


Рисунок 4.4 – Розташування десяти сенсорних вузлів

На рисунку 4.4 зображено БСМ, розгорнуту з використанням десяти сенсорних вузлів. Рисунок наочно ілюструє випадок, при якому п'ять з шістнадцяти локацій практично не охоплені областю покриття датчиків будь-яких сенсорних вузлів. Проте теоретично при збільшенні кількості вузлів площа покриття та швидкість передачі даних буде збільшуватися.

4.6 Реалізація систем підтримки прийняття рішень на базі нечіткої логіки

Розглянемо реалізацію систем підтримки прийняття рішень на базі нечіткої логіки з використанням системи MATLAB.

Основою систем нечіткої логіки є математична теорія нечітких множин, яка бере свій початок зі статті американського вченого Лотфі Заде, опублікованої під назвою "Fuzzy Sets" (нечіткі множини) в 1965 році в журналі *Information and Control*. На даний час ця теорія отримала досить глибокий розвиток, а системи на базі нечіткої логіки знайшли широке застосування в промисловості від виробництва фотоапаратів, пральних машин, мікрохвильових печей до управління великими промисловими виробництвами і систем підтримки прийняття рішення.

Ідея, що лежить в основі теорії нечітких множин, полягає в тому, що людина в своєму повсякденному житті мислить і приймає рішення на основі нечітких понять. Створення теорії нечітких множин - це спроба формалізувати людський спосіб міркувань. Розвиток обчислювальної техніки дозволяє на даний час створювати на базі теорії нечітких множин системи нечіткої логіки, які копіюють спосіб міркувань людини.

У більшості вирішуваних людиною завдань не потрібна висока точність. Навпаки, часто доводиться знаходити розумний компроміс між поняттями "точність" і "важливість" при спілкуванні з реальним світом. Так, для прийняття рішення про перехід вулиці людина не оцінює швидкість

автомобіля, що наближається з точністю до десятих часток метрів в секунду. Він визначає для себе швидкість автомобіля як "дуже швидко", "швидко", "повільно" і т.д., тобто використовує для позначення швидкості лінгвістичні змінні.

Лінгвістичною змінною називається п'ятірка $(x, T(x), X, G, M)$, де x - ім'я змінної; $T(x)$ - множина імен лінгвістичних значень змінної x , кожне з яких є нечіткою множиною на множині X ; G є синтаксичне правило для утворення імен значень x ; M є семантичне правило для асоціювання кожної величини значення з її поняттям.

Це визначення може викликати відчуття, що лінгвістична змінна - дуже складне поняття, але насправді це не так. Мета концепції лінгвістичної змінної полягає в тому, щоб формальним чином сказати, що змінна може приймати в якості значень слова з природної мови. Наприклад, якщо ми говоримо "швидка швидкість", то змінну "швидкість" слід розуміти як лінгвістичну змінну, але це не означає, що змінна "швидкість" не може приймати реальні значення.

Якщо змінна може приймати значення слів в природній мові (наприклад, "маленький", "швидкий" і т.п.), то ця змінна визначається як лінгвістична змінна. Слова, значення яких бере лінгвістична змінна, зазвичай позначають собою нечіткі множини. Лінгвістична змінна може приймати своїми значеннями або слова, або числа.

Нечітка множина - це така множина, яка утворюється шляхом введення узагальненого поняття належності, тобто розширення двоелементною множиною значень функції належності $\{0,1\}$ до відрізка $[0,1]$. Це означає, що перехід від повної належності об'єкта множини до його повної неналежності відбувається не стрибком, як в звичайних "чітких" множинах, а плавно, поступово, причому ступінь належності елемента множини виражається числом з інтервалу $[0,1]$.

Таким чином, нечітка множина $A = \{(x, A(x))\}$ визначається математично як сукупність упорядкованих пар, складених з елементів x

множини X і відповідних їм ступенів належності $A(x)$ або безпосередньо у вигляді функції $A: X \rightarrow [0, 1]$. Розглянемо приклад нечіткої множини.

Приклад 1. Множина високих людей

Нехай x є лінгвістична змінна, що позначає "зріст людини", а її функція належності до множини високих людей $A: X \rightarrow \{0, 1\}$, де X - множина, що включає в себе всі можливі значення зросту людини, задана в такий спосіб:

$$\mu_A(x) = \begin{cases} 1, & \text{якщо } x \geq 180, \\ 0, & \text{якщо } x < 180. \end{cases}$$

Тоді множина "високих людей" задається виразом $A = \{x | \mu_{A(x)}=1\}$, $x \in X$.

Графічно це представлено на рисунку 4.5 (суцільна лінія).

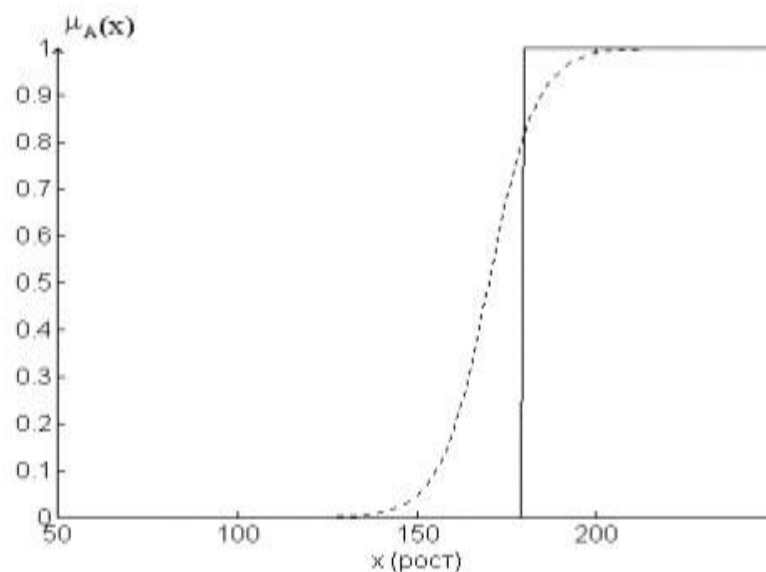


Рисунок 4.5 – Графічне зображення лінгвістичної змінної «множина високих людей»

Однак у реальному житті таке поняття, як "зріст найвищої людини", є суб'єктивним, тобто залежить від індивідуума, що робить оцінку. Одні

вважають, що високий чоловік повинен бути ростом понад 170 см, інші - більш 180 см, треті - більше 190 см. Нечіткі множини дозволяють врахувати таку

μ розмитість оцінок. Так,

нехай функція належності $A: X \rightarrow \{0, 1\}$ має вигляд, представлений на рисунку 4.5 пунктирною лінією. Тоді множина A ("високих людей") задається множиною пар $A = \{x, \mu_A(x)\}, x \in X$ і є нечіткою множиною.

Таким чином, людина ростом 145 см буде належати множині A зі ступенем належності $\mu_A(145) = 0$, людина зростом 165 см - $\mu_A(165) = 0,3$, зростом 185 см - $\mu_A(185) = 0,9$, а зростом 205 см - $\mu_A(205) = 1$.

Можна сказати, що в нечіткій логіці правдивість кожного твердження розглядається в рамках ступеню правдивості, що може також розглядатися як ступінь впевненості експерта, що робить оцінку, в тому, що елемент x належить множині A .

З вищесказаного можна зробити наступні висновки:

- 1) нечіткі множини описують невизначені поняття (швидкий бігун, гаряча вода, спекотна погода);
- 2) нечіткі множини допускають можливість часткової належності до них (п'ятниця - частково вихідний день (укорочений), погода швидше спекотна);
- 3) ступінь належності об'єкта до нечіткої множини визначається відповідним значенням функції належності на інтервалі $[0,1]$ (п'ятниця належить до вихідних днів зі ступенем належності 0,3);
- 4) функція належності ставить у відповідність об'єкту (або логічній змінній) значення ступеню його належності до нечіткої множини.

Один з важливих моментів для розуміння механізму нечіткої логіки є те, що вона може розглядатися як розширення стандартної, добре відомої булевої логіки (подібно до того, як нечіткі множини є розширенням звичайних множин). Іншими словами, якщо розглядати нечіткі значення в їх

екстремумах, тобто в 1 (абсолютна правда) і в 0 (абсолютна неправда), то можна застосовувати стандартні логічні операції: І, АБО, НЕ. У нечіткій логіці істинність кожного твердження розглядається в певній мірі. У нечіткої логіки за аналогією з операціями І та АБО вводяться операції $\min(A, B)$ і $\max(A, B)$, а операції НЕ відповідає операція $1-A$. Графічно це представлено на рисунках 4.6 - 4.8.

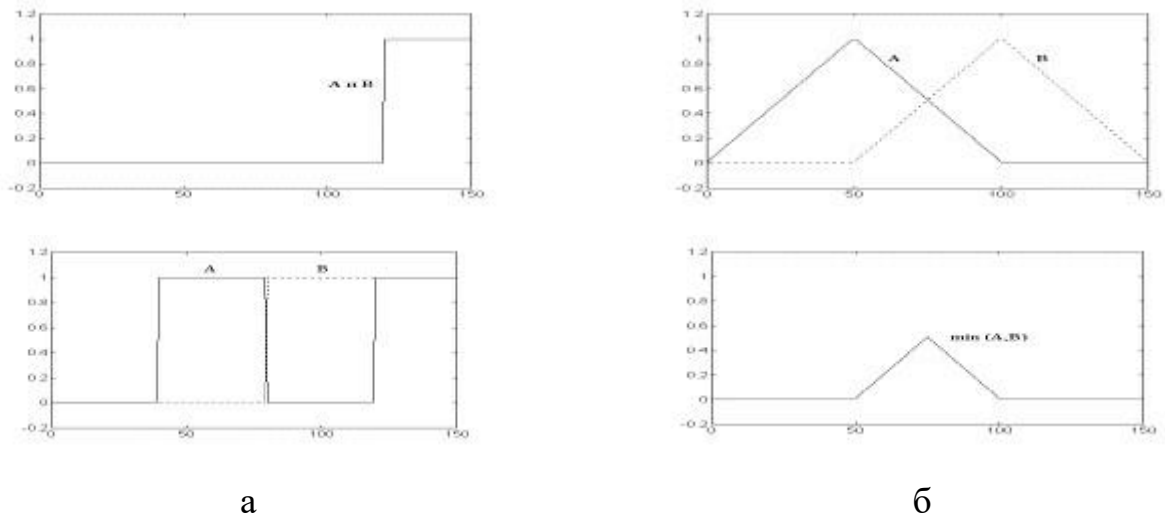


Рисунок 4.6 - Операція $\min(A, B)$: а - "чітка" логіка; б - "нечітка" логіка

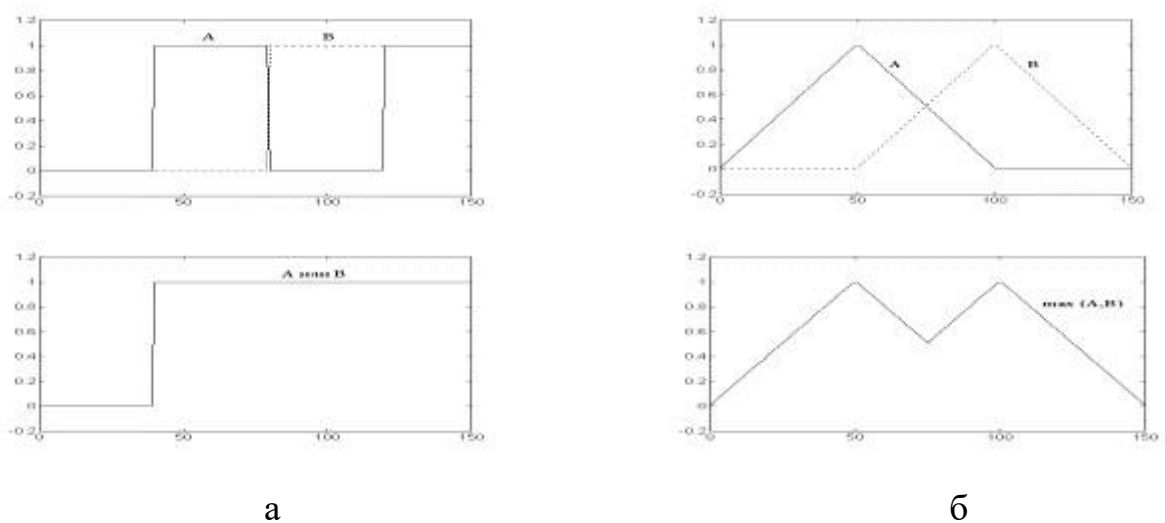


Рисунок 4.7 - Операція $\max(A, B)$: а - "чітка" логіка; б - "нечітка" логіка

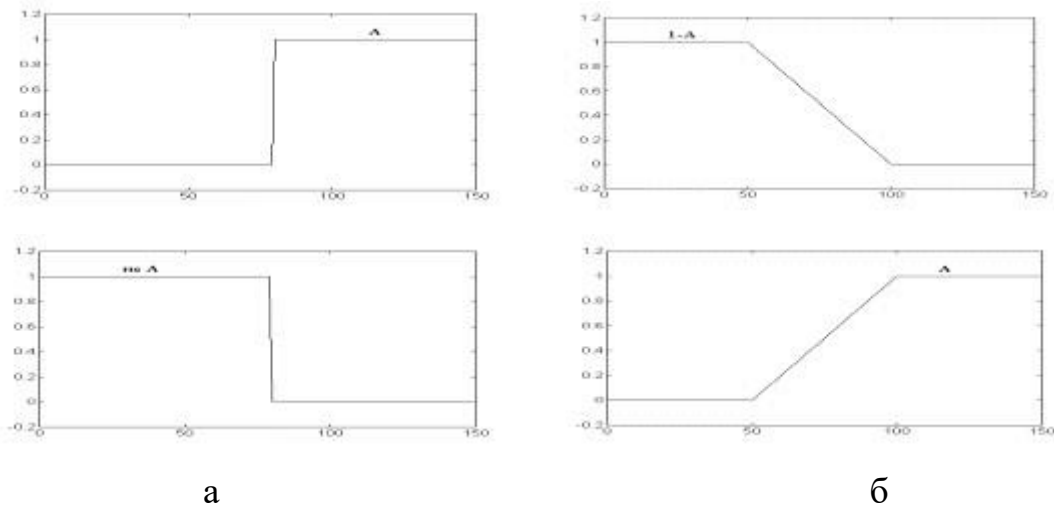


Рисунок 4.8 - Операція 1-А: а - а - "чітка" логіка; б - "нечітка" логіка

Системами нечіткої логіки називаються системи, які оперують нечіткими поняттями, такими як нечіткі множини, лінгвістичні змінні і т.п., і використовують при цьому нечітку логіку. Системи нечіткої логіки можуть бути класифіковані за трьома основними типами:

- прості системи нечіткої логіки (pure Fuzzy Logic Systems);
- нечіткі системи Такагі та Сугено (Takagi and Sugeno);
- системи нечіткої логіки з фаззіфікатором і дефаззіфікатором.

Базова конфігурація простий системи нечіткої логіки представлена на рисунку 4.9.

Базис нечітких правил містить набір нечітких IF-THEN (ЯКЩО-ТО) правил, а механізм нечіткого виводу на основі принципів нечіткої логіки використовує ці IF-THEN-правила для відображення нечітких множин з вхідної множини висловлювань X в нечіткі множини з множини висловлювань Y на виході системи.

Нечіткі IF-THEN-правила виглядають наступним чином:

$$R^{(p)}: IF x_1 \in F_{1p} \text{ і } \dots \text{ і } x_n \in F_{np} THEN y \in G^p,$$

де F_{1p} та G^p – нечіткі множини; $\vec{x} = (x_1, \dots, x_n)^T \in X$ та $x \in Y$ є вхідною і вихідною лінгвістичними змінними відповідно; $p = \bar{1}, \bar{m}$.

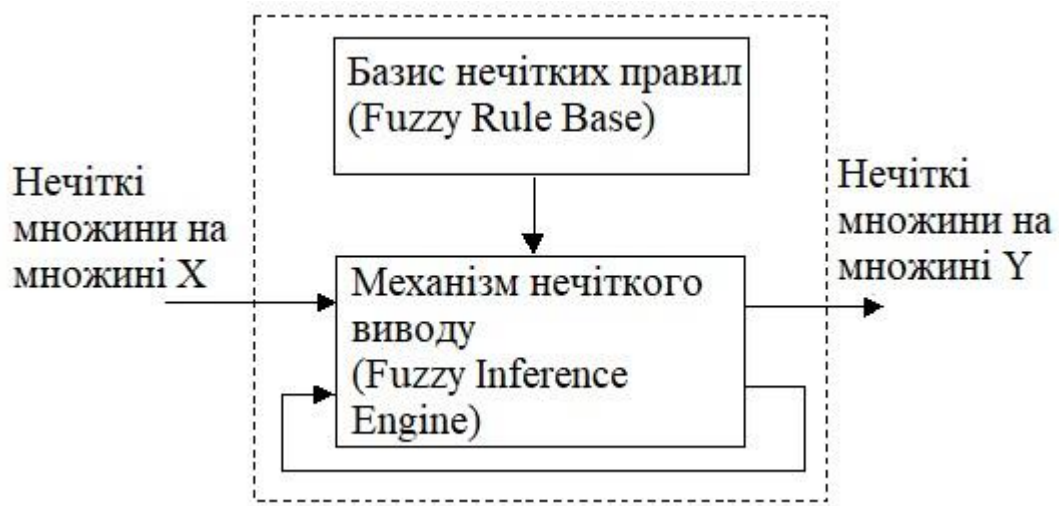


Рисунок 4.9 - Проста система нечіткої логіки

Практика показує, що ці нечіткі IF-THEN-правила забезпечують зручний механізм для представлення знань людини-експерта. Кожне нечітке IF-THEN-правило визначає нечітку множину $F_{1p} \times \dots \times F_{np} \rightarrow G_p$.

Найбільш загальним принципом нечіткої логіки, що використовується в механізмі нечіткого виводу, є композиція. Нехай A' є довільна нечітка множина з X і нехай A' є входом простої системи нечіткої логіки, представленої на рисунку 4.9. Тоді вихід, який визначається кожним нечітким IF-THEN-правилом, є нечітка множина $A' \circ R^{(p)}$ в Y з функцією належності

$$\mu_{A' \circ R^{(p)}}(y) = \sup_{x \in X} (\mu_{A'}(x) \otimes \mu_{F_{1p} \times \dots \times F_{np} \rightarrow G_p}(x, y)),$$

де оператор « \otimes » є

одна з T-норм.

Остаточний вихід простої системи нечіткої логіки є нечітка множина $A' \circ (R^{(p)}, \dots, R^{(M)})$ в Y , яке є комбінацією M нечітких множин, так що $A' \circ (R^{(1)}, \dots, R^{(M)})(y) = \mu_{A' \circ (R^{(1)}, \dots, R^{(M)})}(y) = \mu_{A'}(x) \otimes \dots \otimes \mu_{R^{(M)}}(y)$, де оператор « \otimes » є одна із T-норм.

Якщо присутній зворотний зв'язок, як показано на рисунку 4.9, то ми маємо так звану нечітку динамічну систему, тобто просту систему нечіткої логіки, у якій входи залежать від її виходів.

Структура простої системи нечіткої логіки є суттєвою частиною систем нечіткої логіки. У загальному випадку, коли лінгвістична інформація від експерта перекладається в кількісні показники, принципи нечіткої логіки використовуються для систематизації інформації. Головним недоліком простих систем нечіткої логіки є та обставина, що її входи і виходи - нечіткі множини, тоді як в більшості технічних систем входи і виходи є змінними, які приймають реальні значення.

Нечітке моделювання є одним з перспективних методів оцінки відносини «вхід-вихід» в складних нелінійних системах. Виконується таке моделювання, як правило, за допомогою нечітких систем типу Такагі-Сугено [18]. Нечітка система являє собою реальний об'єкт або процес в формі, яка легко розуміється і інтерпретується непрофесійним математиком. В системі нечіткого моделювання знання або модель досліджуваного об'єкта (процесу) відокремлені від засобів маніпулювання цими знаннями, що дозволяє дослідникам прискорити процес розробки моделі, зосередившись на питанні «що робити» замість питання «як робити» [19].

Побудова нечітких систем включає чотири основні етапи:

- формування вихідних даних - завдання типу нечіткої системи і супутніх цьому типу параметрів;
- формування структури (грубе налаштування);
- пошук оптимальних значень параметрів консеквентів і параметрів функцій належності (ФП) антицедентів (ЯКЦО-частини) на основі заданих критеріїв якості та методу оптимізації обраного критерію (тонке налаштування); так як антецеденти і консеквенти в системі взаємозалежні, цей етап є ітеративним;

- перевірка правильності побудованої системи [20].

Переважає більшість підходів до формування структури засноване на методах нечіткого кластерного аналізу, серед яких найбільш часто вживаними є метод нечітких c -середніх, алгоритми Gustafson-Kessel і Gath-Geva, метод нечіткої c -регресії (FCRM) [21], алгоритм субтрактивної кластеризації [22], а також комбінації зазначених методів [23]. Серед перерахованих вище методів FCRM, який формує форму кластера в вигляді гіперплощини, найкращий для вирішення завдань побудови нечітких систем типу Такагі-Сугено. Однак реалізація алгоритму FCRM для нечіткого моделювання стикається з наступними проблемами: 1) тривалий час виконання; 2) нестійкість до шумів; 3) висока чутливість до даних ініціалізації [24, 25].

Так як оптимізація параметрів антецедентів нечітких правил відноситься до класу NP-важких задач, виконується вона з використанням різних метаевристик і їх гібридизації [26]. Для оптимізації параметрів консеквентів використовуються різні модифікації методу найменших квадратів.

Перевірити вірність побудованої системи і уникнути перенавчання можна, використовуючи процедуру кросвалідації [27].

Останнім часом нечіткі методи моделювання сконцентровані на проблемах поліпшення інтерпретованості нечітких систем без втрати точності [28-30]. Краща інтерпретованість досягається на компактних базах правил. Однак неможливо побудувати систему одночасно з високим ступенем точності і компактності, так як ці властивості є суперечливими; на практиці одне з них переважає над іншим.

При побудові компактних і точних нечітких систем завдання вибору найчастіше вирішується або як двокритеріальне, або зведенням зазначених двох критеріїв до єдиного критерію. В першому випадку різні компроміси між критеріями представлені у вигляді Парето-оптимальної множини згенерованих нечітких систем [31-37]. У другому випадку з двох критеріїв або конструюється один критерій, що враховує ваги кожного [28, 38, 39], або

використовуються статистичні інформаційні критерії, інформаційний характер яких пов'язаний з концепцією інформації Кульбака-Лейблера і які алгебраїчно виражаються через суму мірки помилки апроксимації і штрафу за кількість параметрів системи [40-42]. Кращою вважається система з мінімальним значенням критерію.

Система типу Такагі-Сугено містить оптимальну кількість правил і володіє прийнятною точністю апроксимації. Перевагою систем нечіткої логіки такого типу є те, що вони описуються компактною системою рівнянь. Для них можуть бути легко розроблені методи оцінки параметрів і вибору порядку системи M . Слабке місце таких систем нечіткої логіки полягає в тому, що частина THEN в IF-THEN-правилі не є нечіткою, що не дозволяє природним чином отримувати нечіткі правила від людини-експерта.

Для того, щоб використовувати прості системи нечіткої логіки в технічних додатках, коли входи і виходи систем приймають реальні значення, найпростішим шляхом є додавання фаззифікатора до входу і дефаззифікатора до виходу простої системи нечіткої логіки.

Базова конфігурація системи нечіткої логіки з фаззифікатором і дефаззифікатором представлена на рисунку 4.10.

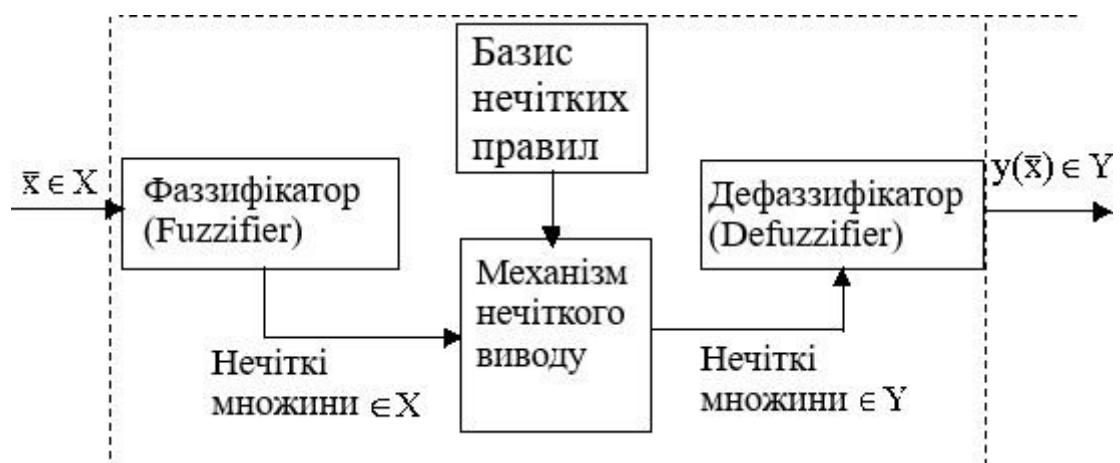


Рисунок 4.10 - Система нечіткої логіки з фаззифікатором і дефаззифікатором

Фаззіфікатор відображає "чітку" точку (або значення змінної) з X в нечіткі множини з X . Дефаззіфікатор відображає нечіткі множини з Y в чітке значення виходу. Базис нечітких правил і механізм нечіткого виведення залишаються такими ж, як у простих системах нечіткої логіки.

Системи нечіткої логіки з фаззіфікатором і дефаззіфікатором мають кілька привабливих рис:

- по-перше, вони придатні для використання в технічних системах, так як їх вхідні і вихідні змінні приймають реальні значення;
- по-друге, вони надають можливість природного переходу від висновків експерта до нечітким IF-THEN-правилам;
- по-третє, вони надають велику свободу у виборі фаззіфікатора, механізму нечіткого виведення і дефаззіфікатора, тобто можна підібрати систему нечіткої логіки, найбільш підходящу для вирішення конкретного завдання;
- по-четверте, можуть бути розроблені різні алгоритми настройки таких систем нечіткої логіки, що дозволяє ефективно поєднувати чисельну і лінгвістичну інформацію.

Базис нечітких правил складається з набору нечітких IF-THEN-правил виду. Без втрати спільності можна розглядати багатовходову систему нечіткої логіки з одним виходом (MISO-систему), тому що система з багатьма виходами (MIMO - система) може бути представлена як композиція групи систем, що мають один вихід.

Базис нечітких правил є основною частиною системи нечіткої логіки в тому сенсі, що інші її компоненти використовуються для інтерпретації цих правил і їх застосування при вирішенні конкретних завдань. Практика показує, що нечіткі IF-THEN-правила надають зручний механізм людині-експерту для вираження своїх знань.

Розглянемо використання систем нечіткої логіки для реалізації функцій прийняття рішення на міжцеховому рівні машинобудівного підприємства на прикладі функції "прийняття оптимального плану".

В ідеальній ситуації всі замовники тієї чи іншої продукції мають стовідсоткової платоспроможністю. Реальне життя вносить свої корективи, і формула для обчислення цільової функції включає коефіцієнт платоспроможності замовника. Зрозуміло, що платоспроможність є цілком нечітким поняттям і визначення коефіцієнта платоспроможності підприємства є складним завданням прийняття рішення. У сучасних умовах економіки платоспроможність підприємства визначається не тільки наявністю грошових коштів на банківському рахунку або можливістю їх своєчасного залучення з інших джерел, а й наявністю виробів або матеріалів, здатних зацікавити підприємство-постачальника як так званого "бартеру".

У зв'язку з цим функція прийняття рішення за завданням коефіцієнта платоспроможності була реалізована представленої на рисунку 4.11 системою нечіткої логіки з двома входами, одним виходом і трьома правилами. Входами системи є експертні бальні оцінки наявності грошей на рахунку замовника та наявності цікавить підприємство бартеру. Мінімальна кількість балів (0) відповідає абсолютної впевненості в тому, що замовник не має на рахунку грошових коштів і не зможе їх вчасно залучити для оплати поставлених йому виробів, а наявний в його розпорядженні бартер абсолютно не потрібен підприємству. Максимальна кількість балів (10) відповідає зворотному стану речей. Оцінки про наявність грошових коштів і корисності бартеру виносяться різними фахівцями. Оцінка наявності грошових коштів на рахунку замовника проводиться фахівцями відділу власної безпеки підприємства і здійснюється за непрямими ознаками. Корисність бартеру оцінюють фахівці з відділу постачання.

Виходом системи нечіткої логіки є значення коефіцієнта платоспроможності, який приймає значення від 0 до 1.



Рисунок 4.11 – Система нечіткої логіки з двома входами, одним виходом і трьома правилами

Система працює наступним чином. Інформаційний потік рухається зліва направо, від двох входів до одного виходу. Паралельна природа виконання правил є одним з важливих аспектів систем нечіткої логіки. Замість різкого перемикавання між моделями (правилами) в точках переходу виходить гладкий перехід від області, в якій поведінка системи описується одним правилом, до області, де воно описується іншим правилом.

Для входів системи визначені відповідні їм нечіткі множини і задані визначають ці множини функції належності. В даному випадку входи задаються на множині чисел (балів) від 0 до 10. Експертом, що брав участь в побудові системи нечіткої логіки, були задані три правила, які використовують нечіткі множини, і визначено функції їх належності. Для наявності грошей на рахунку замовника (змінна "гроші") - це нечіткі множини "маленькі", "середні" і "великі", для змінної "бартер" - "поганий" і "хороший", для змінної "платоспроможність" - "низька", "середня" і "висока".

Перш ніж застосувати той чи інший метод імплікації, необхідно враховувати вагу правила. Кожне правило може мати свою вагу (число від 0 до 1) в залежності від того, наскільки це правило є важливим. Вага правила накладається на вихід IF-частини. В даному випадку всі правила прийняті рівноцінними, і вага кожного правила дорівнює 1.

Метод імплікації визначається як формування виведення з правила на основі заданих умов. Входом процесу імплікації є число, отримане з умов (ступінь істинності умови), а виходом - нечітка множина. Для даної системи нечіткої логіки був обраний найпростіший і добре себе зарекомендував метод імплікації по мінімуму.

У разі, коли кілька правил працюють паралельно, необхідно об'єднати (агрегувати) нечіткі множини, що представляють виходи окремих правил, в одну нечітку множину для підготовки до заключного кроку - дефаззифікації. Входом процесу агрегування є набір нечітких множин, отриманих з кожного правила шляхом імплікації. Його виходом є одна нечітка множина для кожної вихідної змінної. В даному випадку для агрегування нечітких множин використовується операція об'єднання.

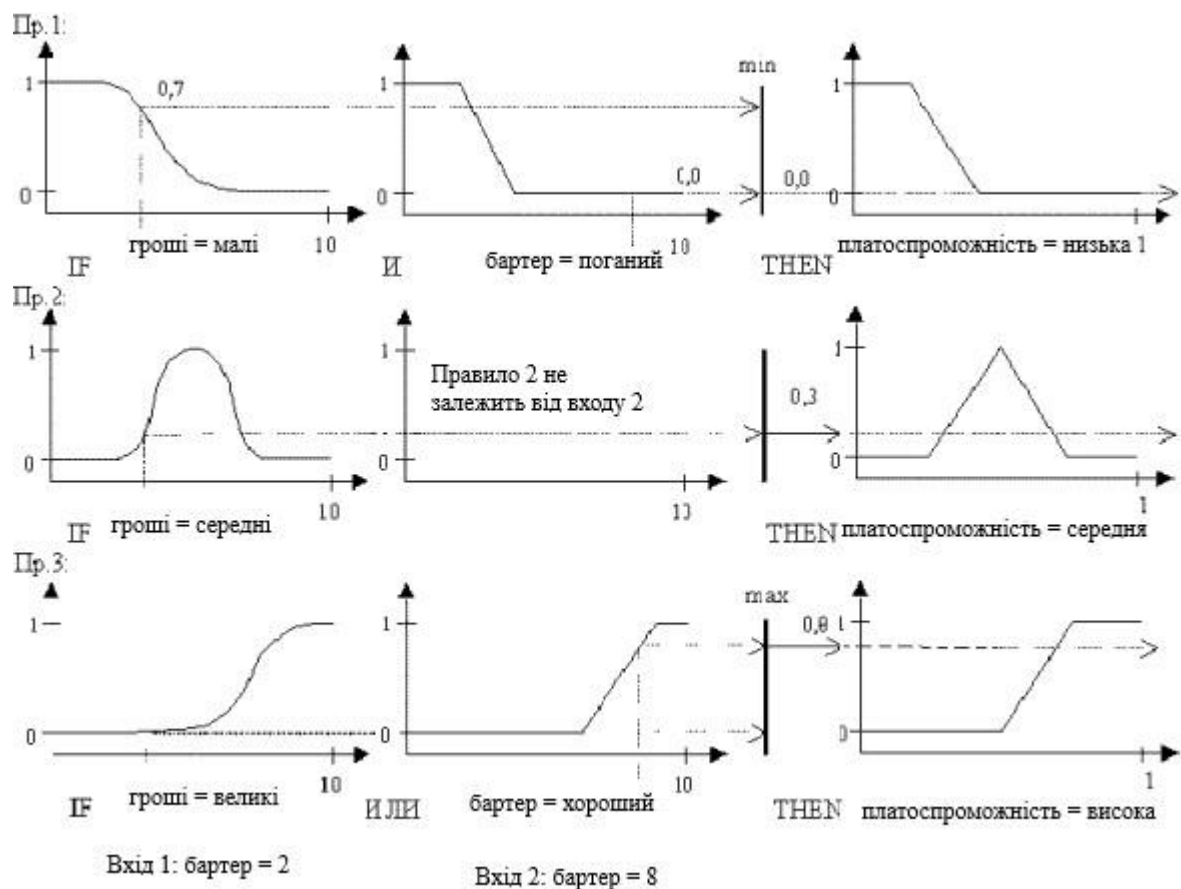


Рисунок 4.12 - Загальна схема пропонованої системи нечіткої логіки

Зауважимо, що так як методи агрегування є комутативними (що мають виконуватися завжди), то порядок, в якому стоять правила, не має значення. Для ілюстрації роботи системи на рисунку 4.12 задані входи "гроші = 3" і "бартер = 8". Виходом операції агрегування для заданих входів є нечітка множина з функцією належності, представленої на рисунку 4.13.

Дефазифікація вихідної нечіткої множини методом центру ваги дає для наведеного прикладу вихідне значення коефіцієнта платоспроможності 0,8. Зауважимо, що всі нечіткі множини задані на відповідних множинах. Для грошей і бартеру - на множині балів від 1 до 10, для коефіцієнта платоспроможності - на безперервній множині від 0 до 1.

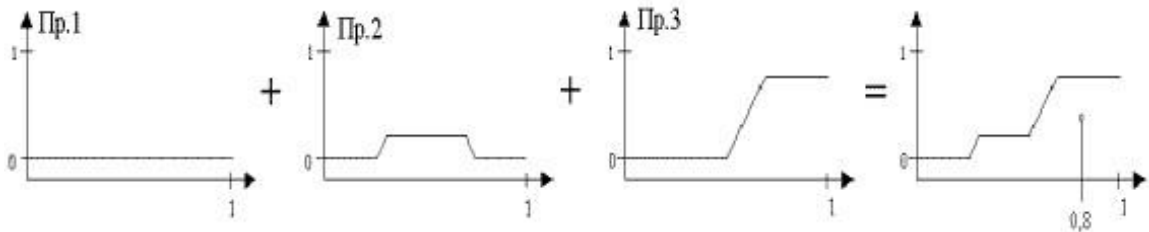


Рисунок 4.13 – Нечітка множина з функцією належності

5 АНАЛІЗ РЕЗУЛЬТАТІВ

Для перевірки правильності вибору датчиків в системі була реалізована система підтримки прийняття рішень на базі нечіткої логіки з використанням системи Matlab.

Задані зв'язки між змінними та параметри системи у вікні редактора FIS Editor, що графічно представлено на рисунку 5.1.

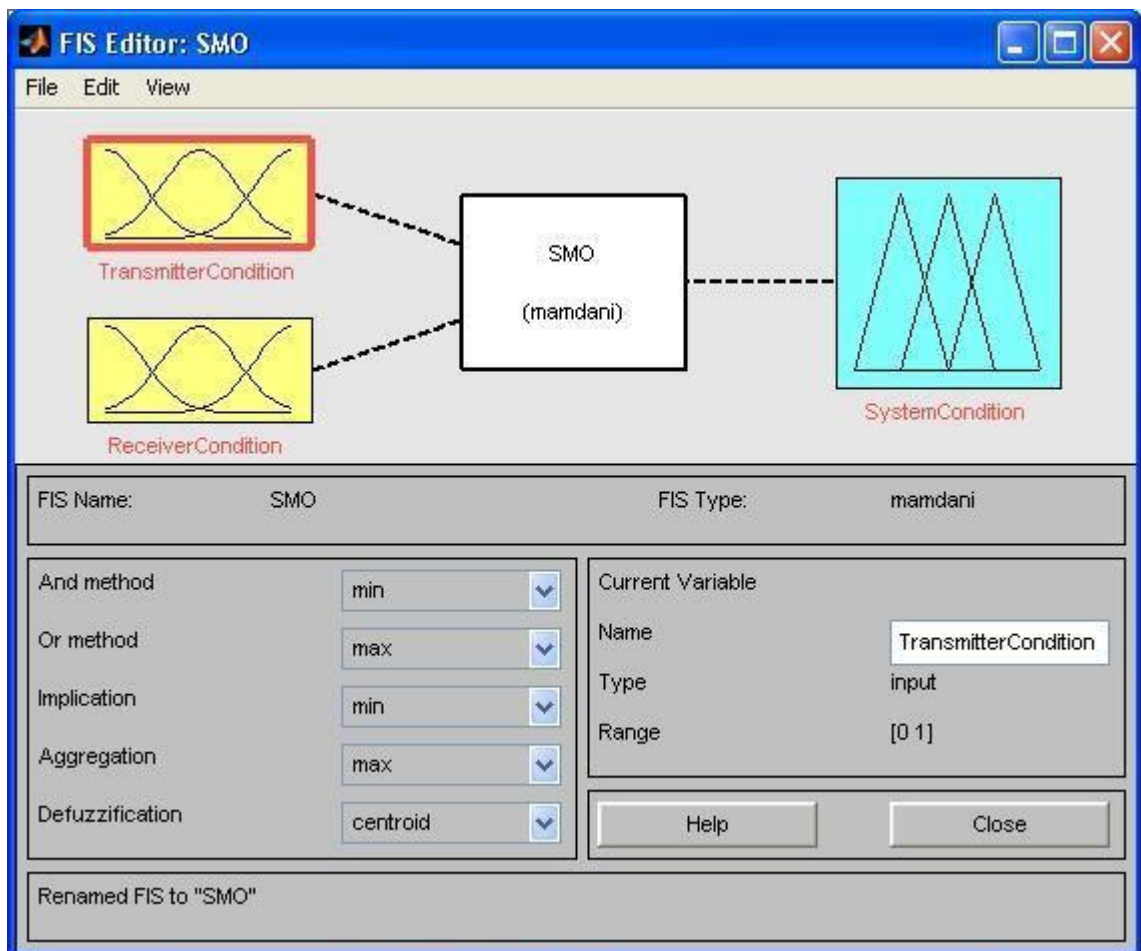


Рисунок 5.1 – Параметри системи у вікні редактора FIS Editor

Задана функцію належності змінної TransmitterCondition. Для лінгвістичної оцінки цієї змінної використано 5 термів з трикутними

функціями належності. В результаті був отриманий графічний результат, зображений на рисунку 5.2.

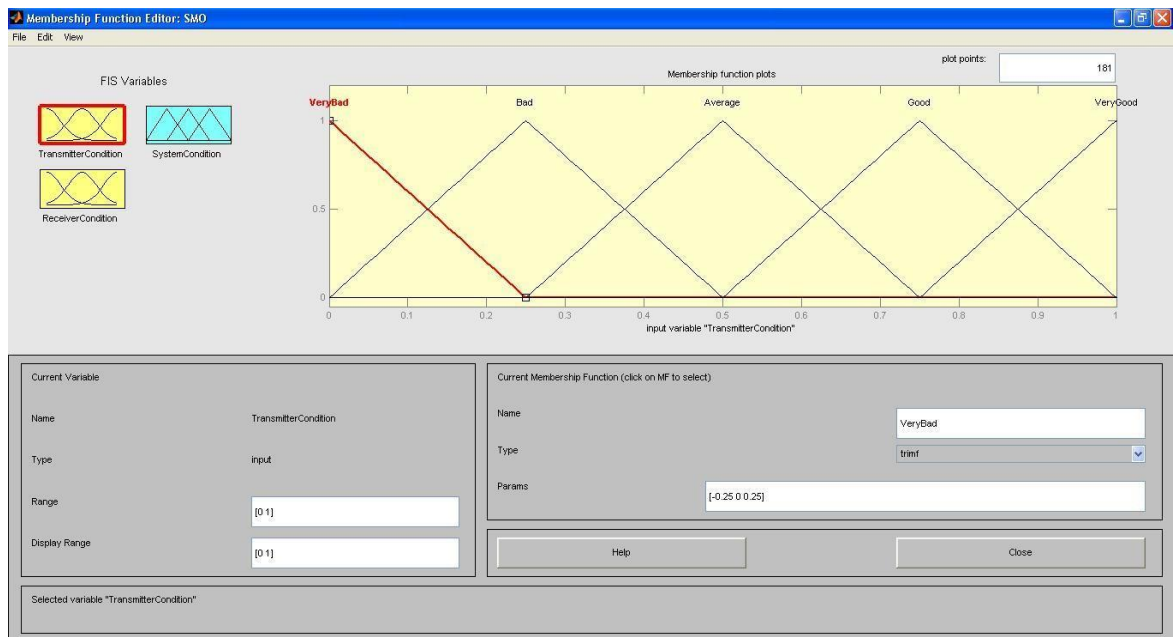


Рисунок 5.2 - Функція належності змінної TransmitterCondition

Задана функція належності змінної ReceiverCondition. Для лінгвістичної оцінки цієї змінної використано 5 термів з дзвоноподібними функціями належності. В результаті був отриманий графічний результат, зображений на рисунку 5.3.

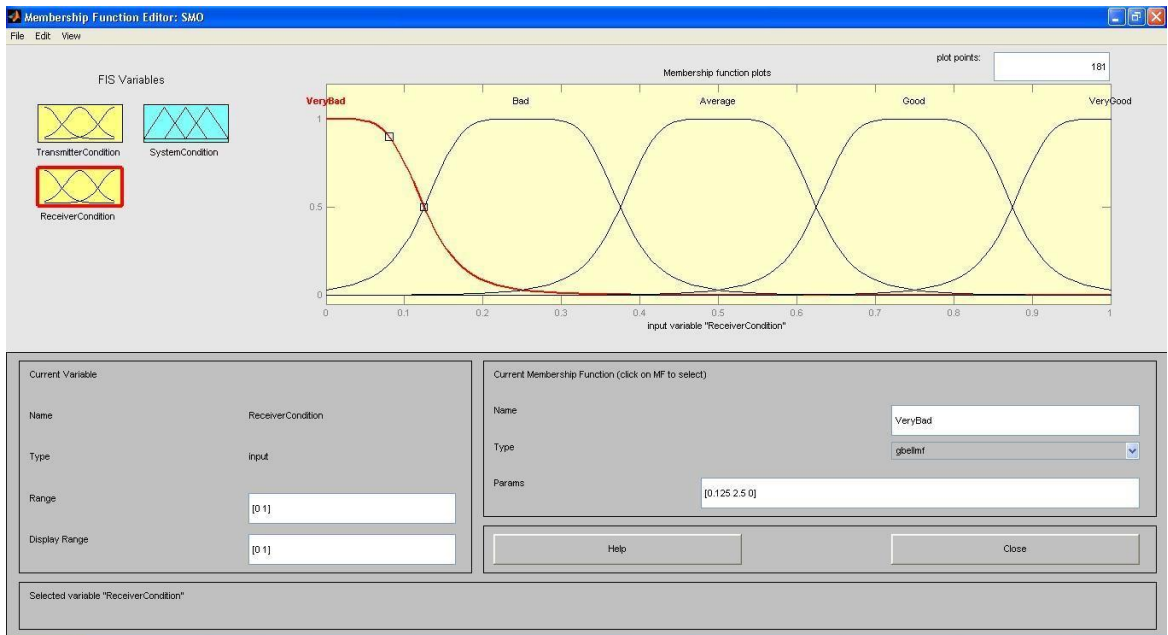


Рисунок 5.3 - Функція належності змінної ReceiverCondition

Була задана функція належності змінної SystemCondition. Для лінгвістичної оцінки цієї змінної використано 4 терма з трапецієподібними функціями належності. В результаті було отримано графічний результат, зображене на рисунку 5.4.

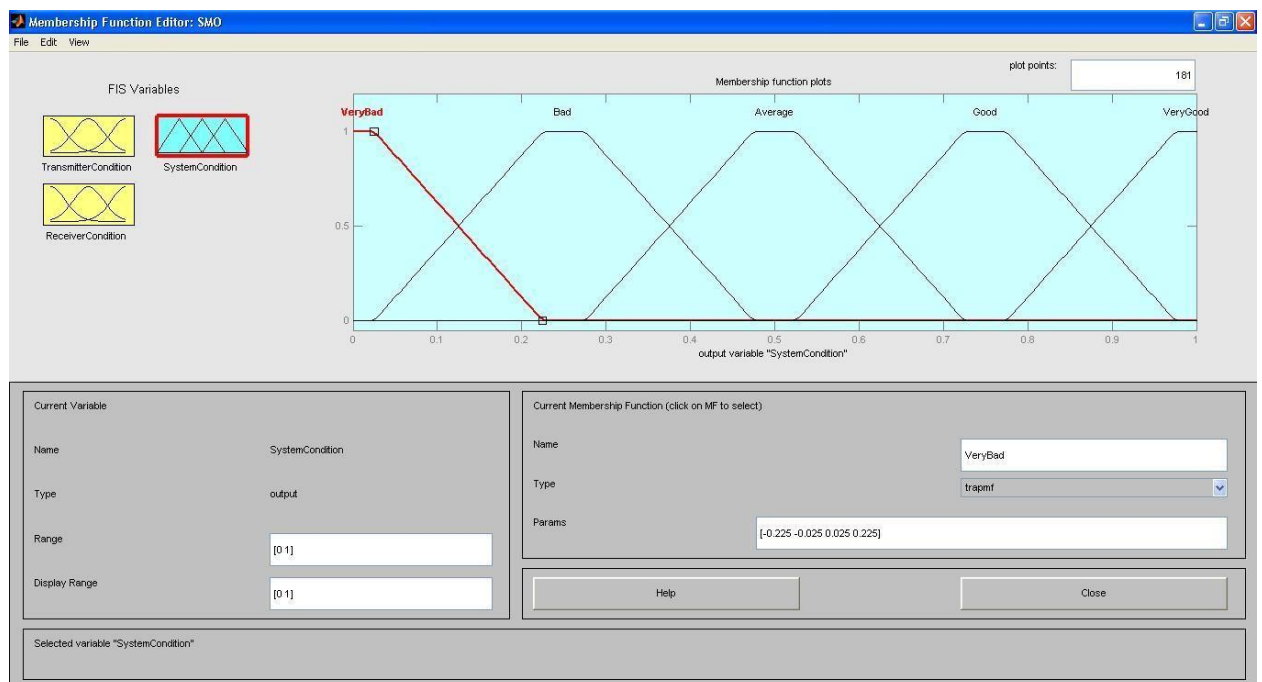


Рисунок 5.4 - Функція належності змінної SystemCondition

Введемо в систему правила логічного висновку. Правила логічного висновку представлені в таблиці 5.1.

Таблиця 5.1 - Таблиця правил логічного висновку

		TransmitterCondition				
ReceiverCondition		Very Bad	Bad	Average	Good	Very Good
	Very Bad	Very Bad	Very Bad	Bad	Bad	Average
	Bad	Very Bad	Bad	Bad	Average	Average
	Average	Bad	Bad	Average	Average	Good
	Good	Bad	Average	Average	Good	Good
	Very Good	Average	Average	Good	Good	Very Good

На рисунку 5.5 зображено вікно редактора бази знань після введення всіх 25 правил.

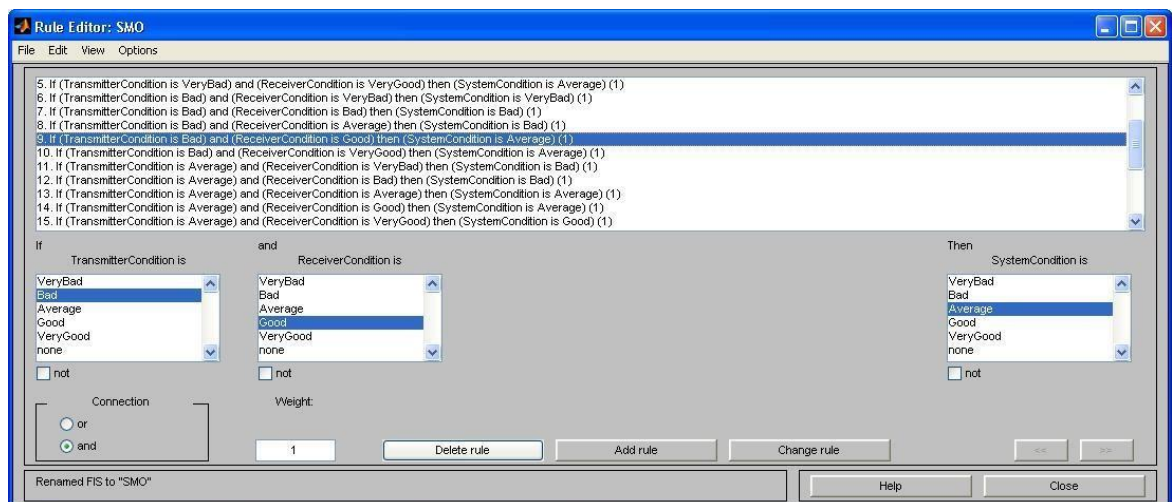


Рисунок 5.5 - База знань в RuleEditor

На рисунку 5.6 приведено вікно візуалізації нечіткого логічного висновку системи. Були вказані 25 значень входних змінних системи, які відповідають станам приймача і передавача.

Таблиця 5.2 - Агрегування підумов правил

TransmitterCondition	ReceiverCondition	SystemCondition
0	0	0.0734 – Good
0.25	0.25	0.249 – Very Bad
0.5	0.5	0.49 – Average
0.75	0.75	0.74 – Good
1	1	0.912 – Very Good
0.11	0.36	0.204 – Very Bad
0.11	0.62	0.363 – Bad
0.37	0.62	0.372 – Bad
0.38	0.11	0.215 – Very Bad
0.82	0.61	0.595 – Average
0.83	0.84	0.759 – Good

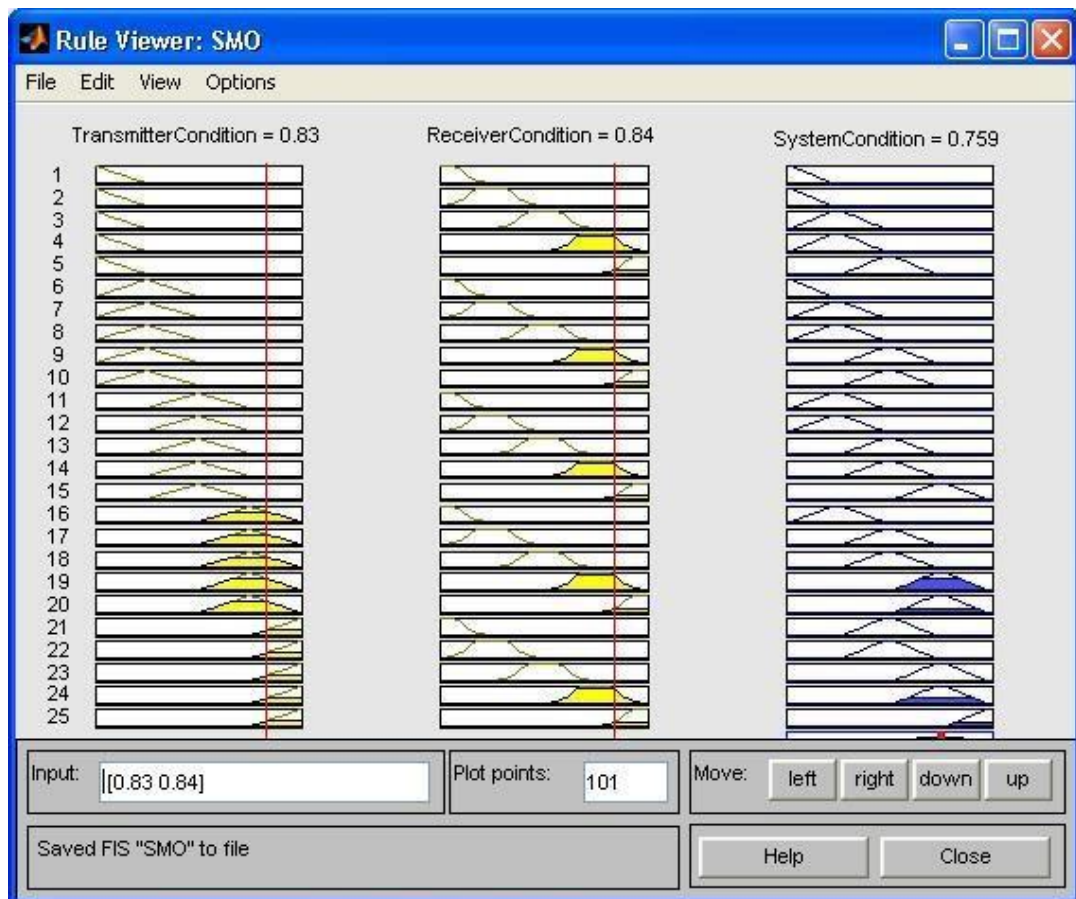


Рисунок 5.6 - Візуалізація нечіткого логічного виводу в RuleViewer

На рисунку 5.7 приведена поверхня "входи-вихід", відповідна синтезованій нечіткій системі.

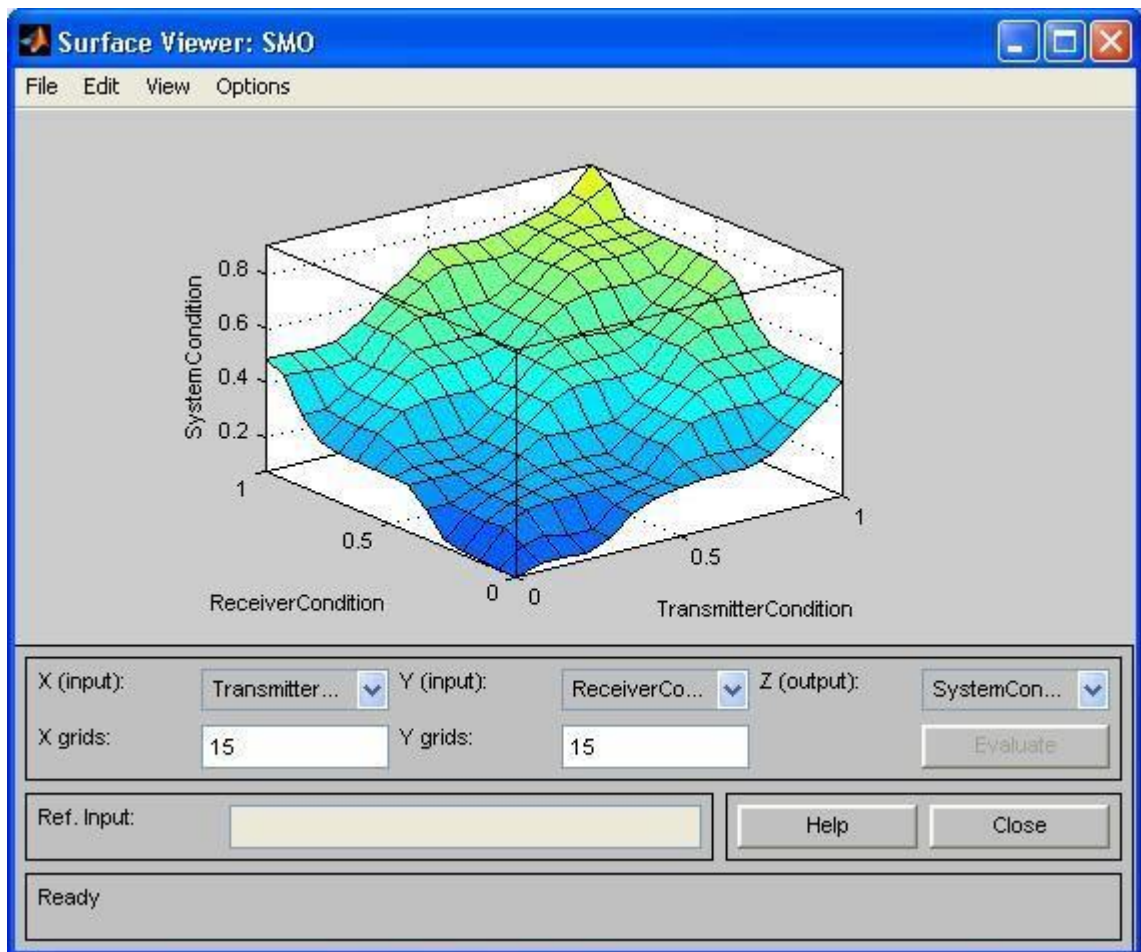


Рисунок 5.7 - Поверхня "входи-вихід" у вікні SurfaceViewer

В ході даного експерименту відповідно до вхідних даних були задані функції належності для вхідних змінних `TransmitterCondition` та `ReceiverCondition` і для вихідної змінної `SystemCondition`. Була складена таблиця правил логічного висновку, на основі даної таблиці була сформульована сукупність правил, які були введені в систему. Для вхідних значень, визначених відповідно до вхідних даних, було виконано моделювання. В результаті моделювання було отримано значення 0.759, що характеризує високий показник стану системи. Була отримана поверхня "входи-вихід", відповідна синтезованій нечіткій системі.

На наступному етапі створено нейро-нечітку мережу в середовищі Matlab, здійснено її навчання, а також використання налаштованої мережі для отримання результатів нечіткого виведення.

В середовищі Matlab вибрано кількість та тип функцій належності.

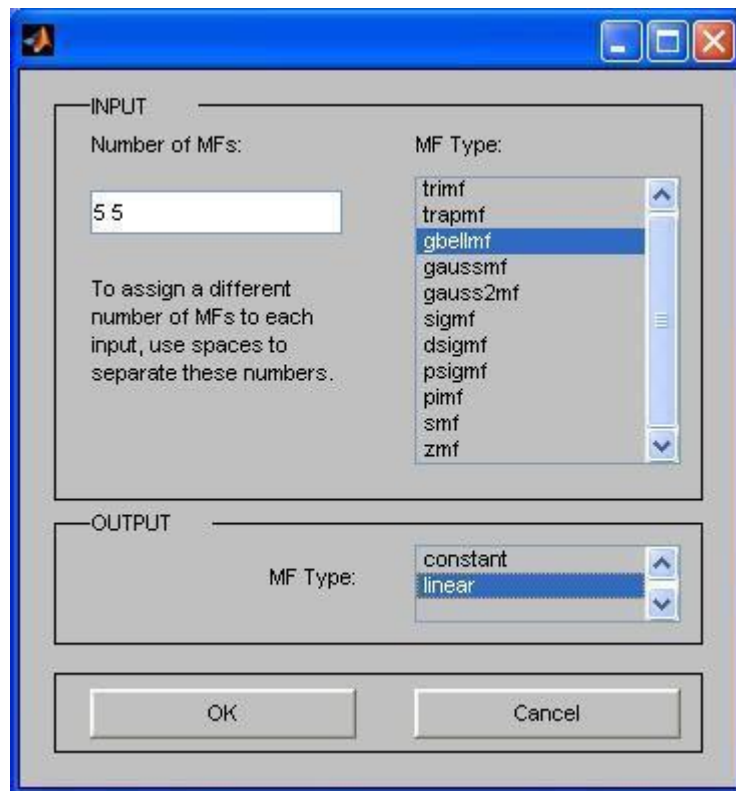


Рисунок 5.8 — Вибір кількості та типу функцій належності

Був підготовлений файл "train.dat" з навчальними даними, який містять 11 рядків. Значення якості приймача і передавача є однаковими для всіх рядків.

Зміст файлу з навчальними даними наведено в таблиці 5.3.

Таблиця 5.3 — Зміст файлу "train.dat"

TransmitterCondition	ReceiverCondition	SystemCondition
0	0	0.0734
0.25	0.25	0.249
0.5	0.5	0.49
0.75	0.75	0.74
1	1	0.912
0.11	0.36	0.204
0.11	0.62	0.363

0.37	0.62	0.372
0.38	0.11	0.215
0.82	0.61	0.595
0.83	0.83	0.759

Файл з навчальними даними був завантажений в систему. В результаті було отримано результат, зображений на рисунку 5.9.

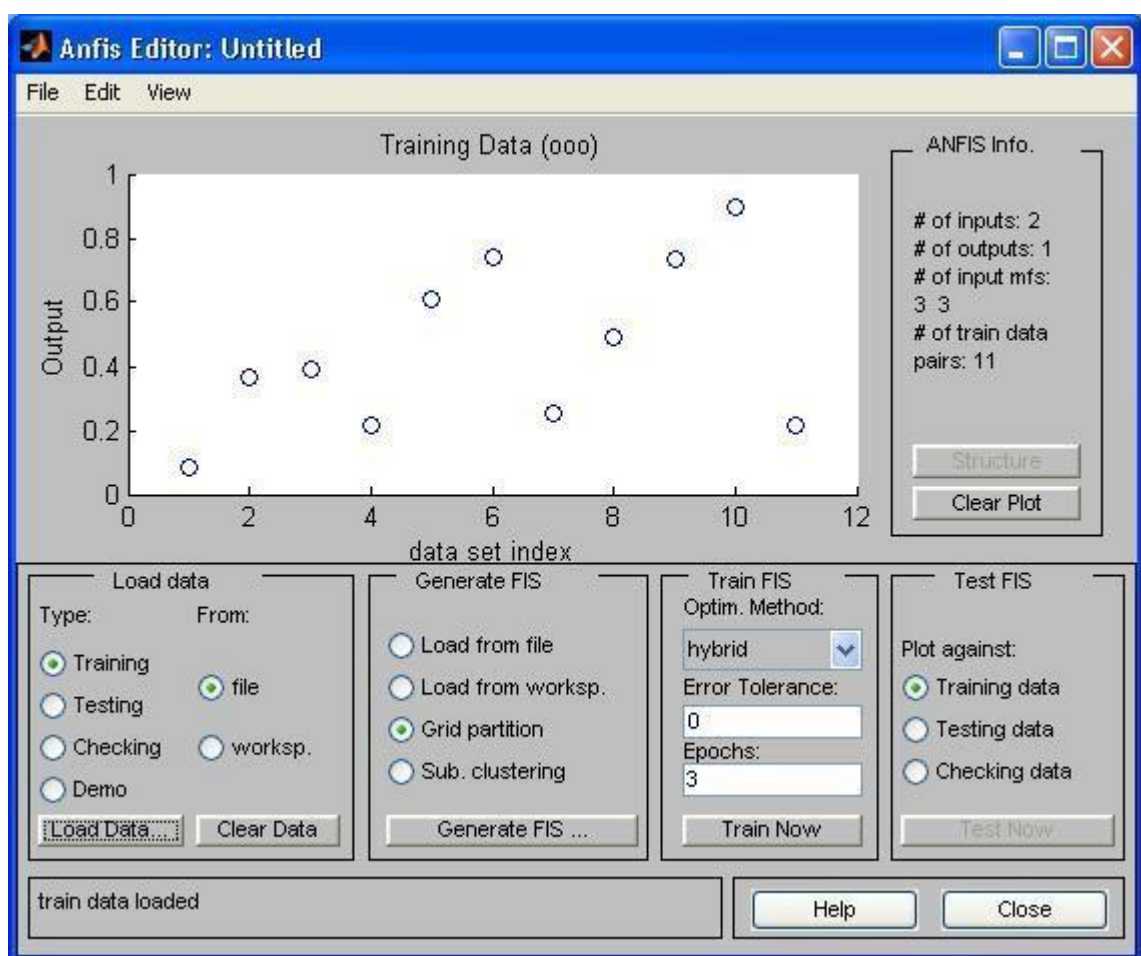


Рисунок 5.9 — Графічний інтерфейс ANFIS після завантаження файлу "train.dat"

ANFIS підтримує тільки системи типу Сугено. Крім того, системи нечіткого виводу повинні мати такі властивості:

- мати один вихід, отриманий з використанням дефаззифікації за методом середньозваженого значення;
- бути системою Сугено першого або нульового порядку;
- мати однакову вагу для всіх правил;
- всі вихідні функції належності повинні бути одного типу і бути linear або constant;
- кількість вихідних функцій належності має дорівнювати кількості правил.

Була отримана система Сугено, яка задовольняє цим вимогам. Дана система була завантажена в ANFIS.

Структура нейро-нечіткої мережі представлена на рисунку 5.10.

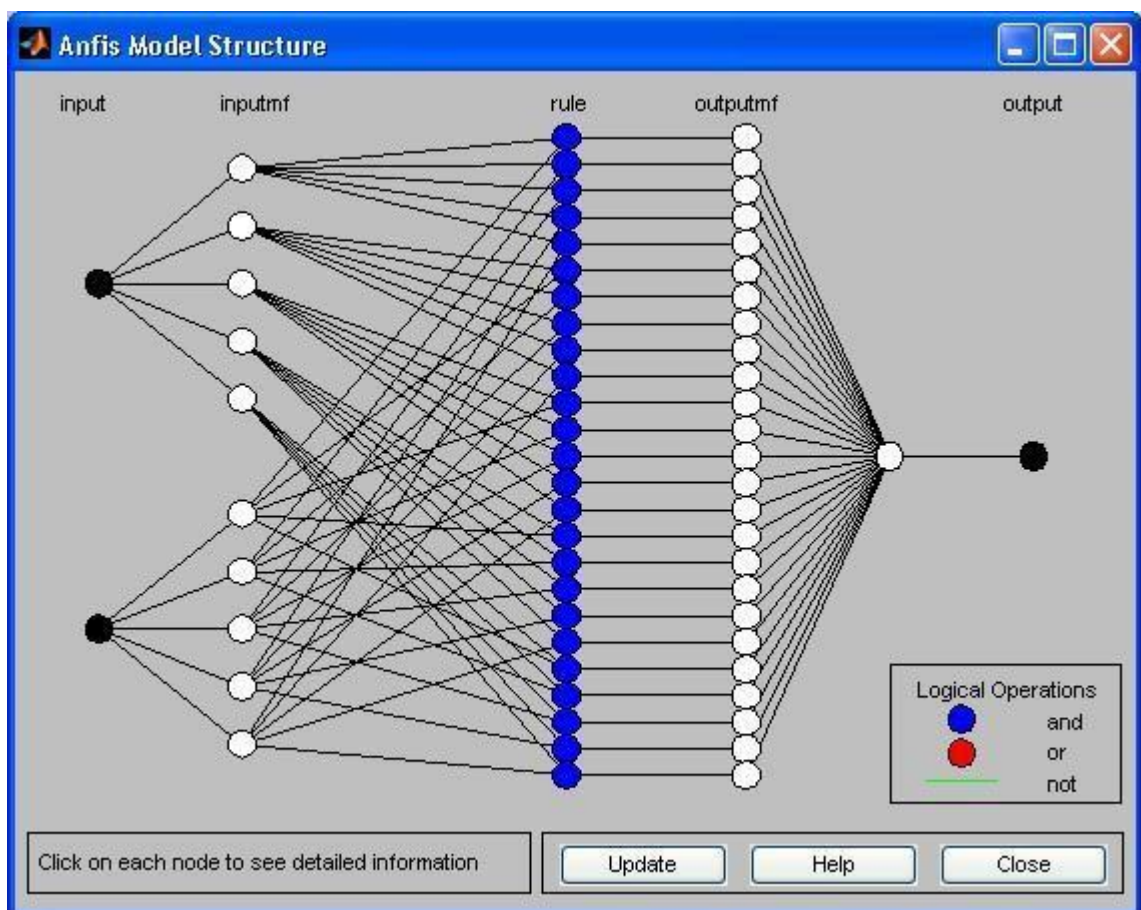


Рисунок 5.10 — Структура нейро-нечіткої мережі

Були виставлені параметри навчання нейронної мережі: метод навчання (гібридний), рівень помилки (0), кількість циклів навчання (30). В результаті навчання мережі було отримано графік зміни помилки в ході виконання циклів навчання, зображений на рисунку 5.11.

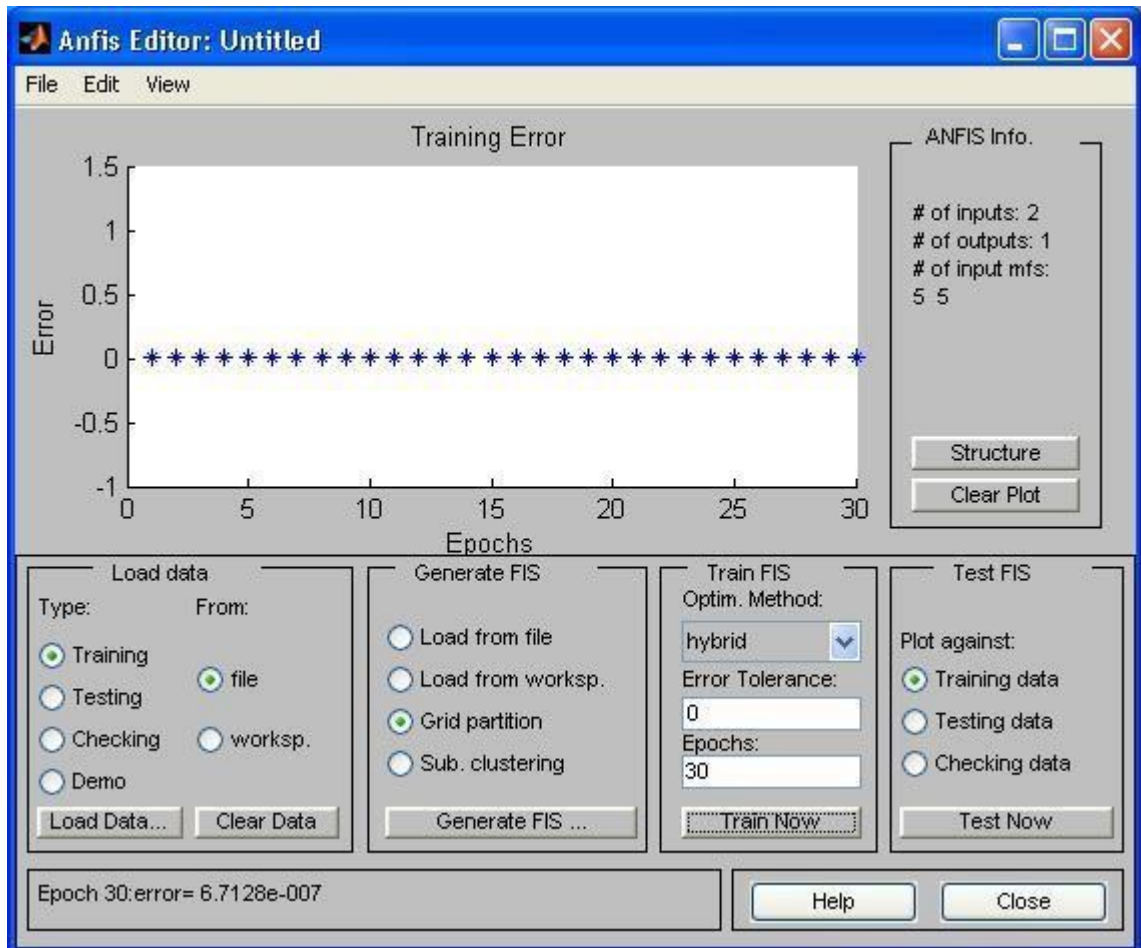


Рисунок 5.11 — Графік залежності помилки навчання від кількості циклів навчання

Як можна бачити, значення помилки навчання навіть на першому циклі навчання незначно відрізняється від 0.

Був підготовлений файл "testing.dat" з перевірочними даними в 11 рядків. Зміст файлу з перевірочними даними наведено в таблиці 5.4.

Таблиця 5.4 — Зміст файлу "testing.dat"

TransmitterCondition	ReceiverCondition	SystemCondition
----------------------	-------------------	-----------------

0.03	0.01	0.0939
0.27	0.24	0.258
0.51	0.52	0.506
0.74	0.73	0.734
0.98	1	0.888
0.09	0.36	0.19
0.1	0.62	0.358
0.38	0.62	0.381
0.34	0.14	0.229
0.82	0.60	0.579
0.83	0.81	0.751

Файл з перевірочними даними був завантажений в систему. В результаті було отримано графічний результат, зображений на рисунку 5.12.

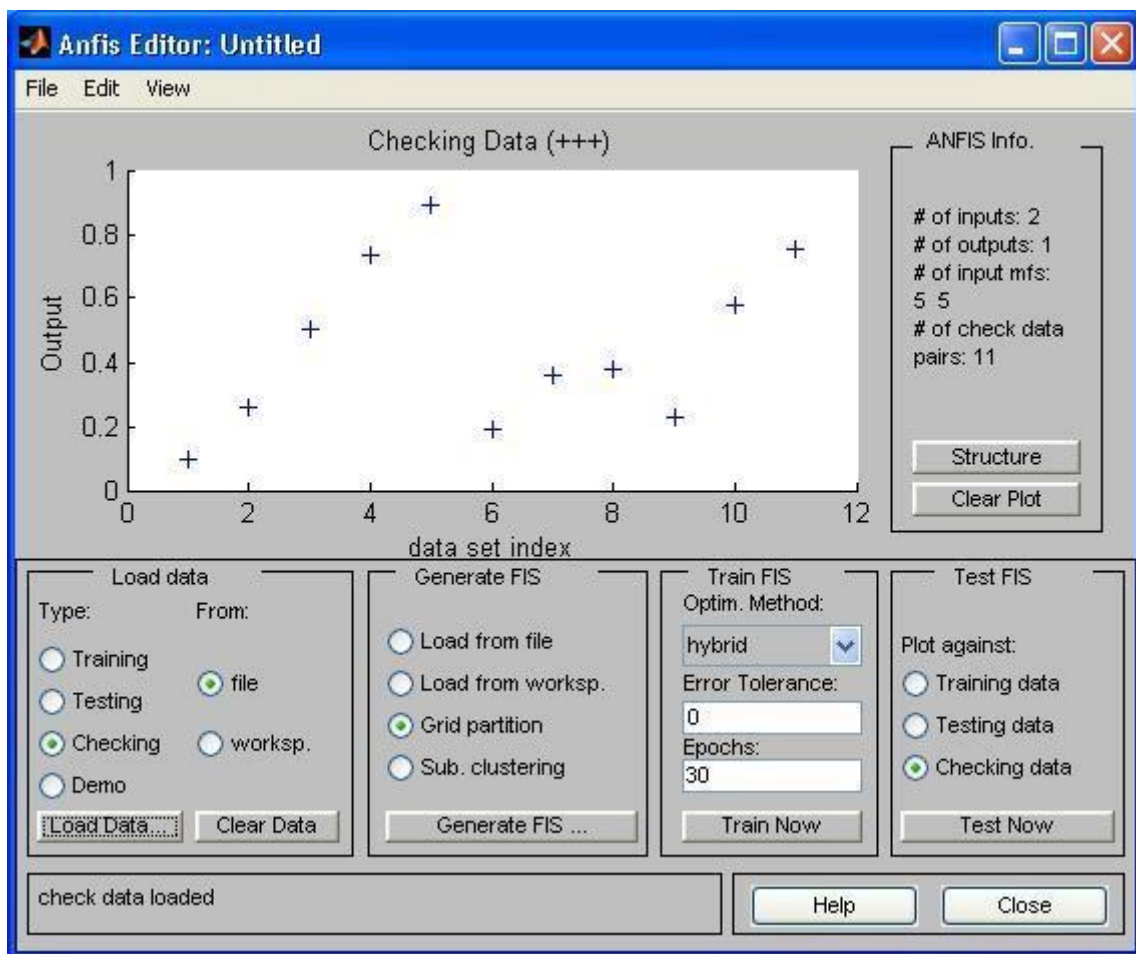


Рисунок 5.12 — Графічний інтерфейс ANFIS після завантаження файлу "testing.dat"

Результати перевірки нейронної мережі на даних з файлу "testing.dat" представлені на рисунку 5.13.

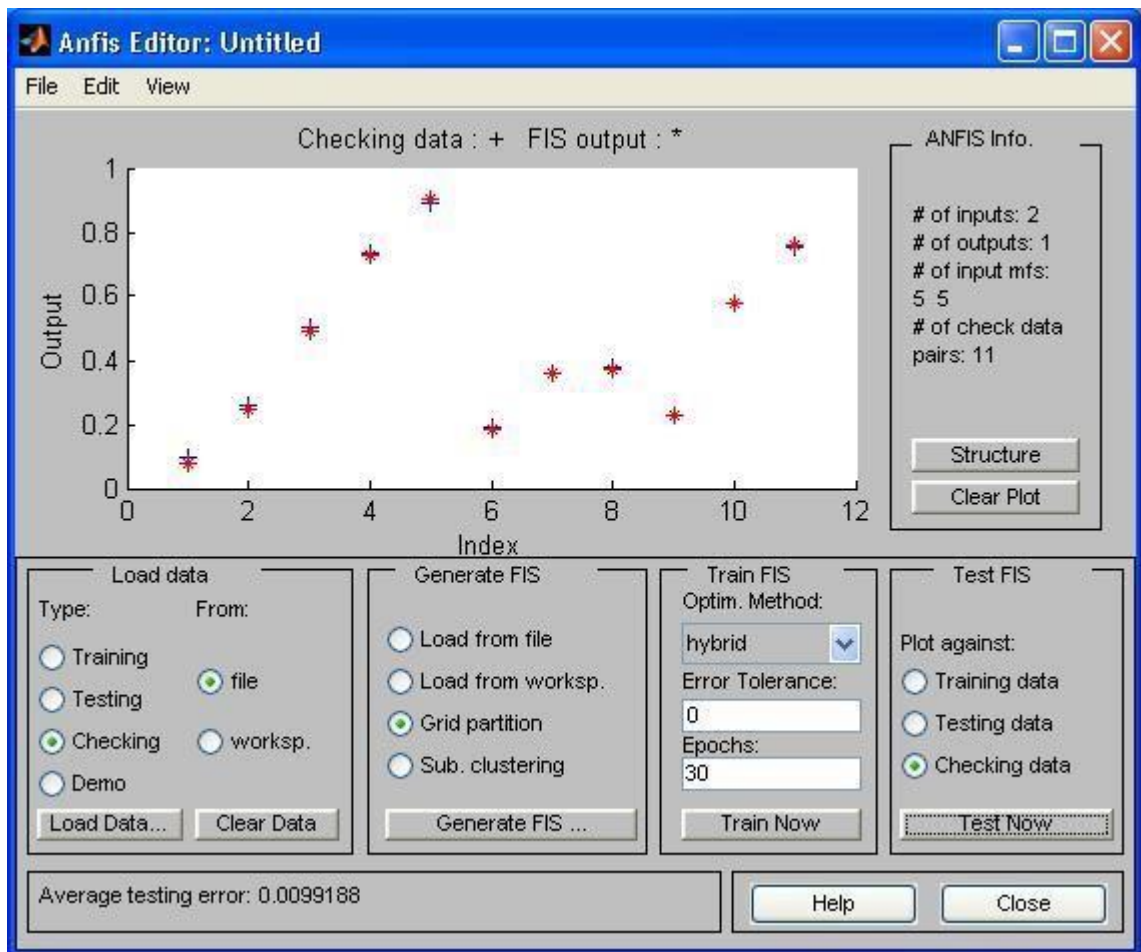


Рисунок 5.13 — Результати перевірки нейронної мережі на даних з файлу "testing.dat"

В ході даного експерименту було складено і завантажено в ANFIS файл з навчальними даними для нейронної мережі. У адаптивну систему нечіткого виводу була завантажена FIS-система типу Сугено. Було виконано навчання нейронної мережі на навчальних даних. Був складений і завантажений в адаптивну систему нечіткого виводу файл з перевірочними даними. Результати перевірки нейронної мережі показують, що середнє значення помилки при тестуванні дорівнює 0.009. Була отримана структура розробленої нейро-нечіткої мережі.

ВИСНОВКИ

В даній атестаційній роботі було проведено дослідження існуючих технологій для збору інформації з використанням БСМ та інтелектуальних датчиків, а також подальшої її відправки до віддаленої базової станції.

В результаті виконання даної роботи запропоновано оптимальний спосіб моніторингу параметрів навколишнього середовища, який може подолати недоліки, існуючі в сучасних системах БСМ. Було визначено, що моніторинг параметрів з використанням БСМ на базі плоского протоколу маршрутизації типу multi-hop призводить до відчутного збільшення тривалості функціонування і скорочення вартості розгортання подібної мережі завдяки застосуванню алгоритму TPSMA для розміщення сенсорних вузлів в межах контрольованої території.

Однак, незважаючи на існування різних типів протоколів маршрутизації в БСМ, все ще існує велика потреба в нових протоколах, які могли б бути простішими в реалізації і при цьому істотно продовжити термін служби мережі. При необхідності збільшення площі контрольованої території або при видаленні її в просторі на істотну відстань, варто задуматися про впровадження системи зчитування інформації датчиків сенсорної мережі за допомогою безпілотних літальних апаратів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Mainwaring, Wireless sensor networks for habitat monitoring [Текст] / A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson // First ACM Workshop on Wireless Sensor Networks and Applications, – Atlanta, GA, 2002.
2. P. Juang, Energyefficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet [Текст] / P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, D. Rubenstein // Proceedings of the 10th International

Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X) – San Jose, CA, 2002.

3. G. Werner-Allen, Monitoring volcanic eruptions with a wireless sensor network [Текст] / G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, M. Welsh // Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05). – 2005.

4. Self-healing Mines [Електронний ресурс] / DARPA, – Режим доступу: <https://www.globalsecurity.org/military/systems/munitions/shm.htm>

5. R. C. Shah, Energy Aware Routing for Low Energy Ad Hoc Sensor Networks [Текст] / R. C. Shah, J. Rabaey // Proceedings of the IEEE Wireless Communications and Networking Conference. – Orlando, USA, 17 - 21 March, 2002, – vol. 1. – pp. 350-355.

6. Y. Al-Obaisat, On wireless sensor networks: architectures, pro-tocols, applications, and management [Текст] / Y. Al-Obaisat, R. Braun // – 2007.

7. G. Wang, Movement-Assisted Sensor Deployment [Текст] / G. Wang, G. Cao, T. F. Porta // Transactions on mobile computing, – 2006. – vol. 5. – pp. 640652.

8. A. Howard, Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem [Текст] / A. Howard, M. Mataric, G. Sukhatme // 6th International Symposium on Distributed Autmomous Robotics System (DARS). – Fukuoka, Japan, Jun 25-27, 2002.

9. Кісельова О.М. Неперервні задачі оптимального розбиття множин і галгоритми [Текст] / Кісельова О.М., Коряшкіна Л.С. // Київ, Наукововиробниче підприємство «Видавництво «Наукова думка» НАН України». – 2015. – с.189-219.

10. Брусов В.С. Вычислительный алгоритм оптимального покрытия областей плоскости [Текст] / В.С. Брусов, С.А. Пиявский // Журн. вычисл. математики и мат. физики. – 1979. – 11, № 2. – с. 304-312.

11. Пиявский С.А. Об оптимизации сетей [Текст] / Пиявский С.А. – Изв. АН СССР. Техн. Кибернетика. – 1968. – № 1. – с. 68-80.
12. Коряшкина Л.С. Применение методов оптимального разбиения множеств к непрерывным задачам многократного покрытия [Текст] / Л.С. Коряшкина, А.А. Михалева, В.И. Навоенко // Питання прикладної математики і математичного моделювання. – 2014. – с. 141-154.
13. Галиев Ш.И. Оптимизация многократного покрытия ограниченного множества кругами [Текст] / Ш.И. Галиев, М.А. Карпова // Ж. вычисл. матем. и матем. физ. – 2010. – Т. 50, № 4. – с. 757-769.
14. Шор. Н.З. Использование модификации r-алгоритма для нахождения минимума полиномиальных функций [Текст] / Н.З. Шор, Т.А. Бардадым, Н.Г. Журбенко, А.П. Лиховид, П.И. Стецюк // Кибернетика и системный анализ. – 1999. - № 5. – с. 33-47.
15. Y. Al-Obaisat, On wireless sensor networks: architectures, pro-tocols, applications, and management [Текст] / Y. Al-Obaisat, R. Braun // – 2007.
16. M. Kuorilehto, Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment [Текст] / M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hämmäläinen, M. Hämmäläinen, T. D. Hämmäläinen // Wiley Online Library. – 2007.
17. A. R. D. Kevin Townsend, Getting Started with Bluetooth Low Energy [Текст] / A. R. D. Kevin Townsend, Carles CufÃ // O'Reilly Media. – 2014. – vol. 54.
18. Takagi T., Fuzzy identification of systems and its application to modeling and control [Текст] / T. Takagi, M. Sugeno // IEEE Transaction Systems, Man and Cybernetics. – 1985. – vol. 15, № 1. – pp. 116–132.

19. Tron E. Mathematical modeling of observed natural behavior: a fuzzy logic approach [Текст] / E. Tron, M. Margaliot // *Fuzzy Sets and Systems*. – 2004. – vol. 146. – pp. 437–450.
20. Ходашинский И.А. Идентификация нечетких систем: методы и алгоритмы [Текст] // *Проблемы управления*. – 2009. – № 4. – с. 15–23.
21. J. Abonyi, *Cluster Analysis for Data Mining and System Identification* [Текст] / J. Abonyi, B. Feil. – Birkhäuser, 2000. – p. 319.
22. R. Yager, Generation of fuzzy rules by mountain method [Текст] / R. Yager, D. Filev // *Journal of Intelligent & Fuzzy Systems*. – 1994. – vol. 2. – pp. 209–219.
23. M.R. Sadrabadi, Identification of the linear parts of nonlinear systems for fuzzy modeling [Текст] / M.R. Sadrabadi, M.H.F. Zarandi // *Applied Soft Computing*. – 2011. – vol. 11. – pp. 807–819.
24. C. Li, T–S fuzzy model identification based on a novel fuzzy c–regression model clustering algorithm [Текст] / C. Li, J. Zhou, Q. Li, X. Xiang, X. An // *Engineering Applications of Artificial Intelligence*. – 2009. – vol. 22. – pp. 646–653.
25. M. Soltani, A novel fuzzy c–regression model algorithm using a new error measure and particle swarm optimization [Текст] / M. Soltani, A. Chaari, F. Ben Hmida // *International Journal of Applied Mathematics and Computer Science*. – 2012. – vol. 22, № 3. – pp. 617–628.
26. *Handbook of Metaheuristics* [Текст] / Editors: M. Gendreau, J.-Y. Potvin. – Springer, 2010. – p. 669.
27. H. Ishibuchi, Repeated double cross-validation for choosing a single solution in evolutionary multi-objective fuzzy classifier design [Текст] / H. Ishibuchi, Y. Nojima // *Knowledge-Based Systems*. – 2013. – vol. 54. – pp. 22–31.
28. C.H. Nguyen, A discussion on interpretability of linguistic rule based systems and its application to solve regression problems [Текст] / C.H. Nguyen,

V.T. Hoang, V.L. Nguyen // Knowledge-Based Systems. – 2015. – vol. 88. – pp. 107–133.

29. M.B. Gorzalczany, A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability [Текст] / M.B. Gorzalczany, Rudzinski F. // Applied Soft Computing. – 2016. – vol. 40. – pp. 206–220.

30. M. Galende-Hernandez, Complexity reduction and interpretability improvement for fuzzy rule systems based on simple interpretability measures and indices by bi-objective evolutionary rule selection [Текст] / M. Galende-Hernandez, G.I. Sainz-Palmero, M.J. Fuente-Aparicio // Soft Computing. – 2012. – vol. 16. – pp. 451–470.

31. P. Pulkkinen, A Dynamically Constrained Multiobjective Genetic Fuzzy System for Regression Problems [Текст] / P. Pulkkinen, H. Koivisto // IEEE Transaction on Fuzzy Systems. – 2010. – vol. 18, № 1. – pp. 161–177.

32. R. Alcalá, A Fast and Scalable Multiobjective Genetic Fuzzy System for Linguistic Fuzzy Modeling in HighDimensional Regression Problems [Текст] / R. Alcalá, M.J. Gacto, F. Herrera // IEEE Transaction on Fuzzy Systems. – 2011. – vol. 19, № 4. – pp. 666–681.

33. M. Antonelli, An efficient multi-objective evolutionary fuzzy system for regression problems [Текст] / M. Antonelli, P. Ducange, F. Marcelloni // International Journal of Approximate Reasoning. – 2013. – vol. 54, № 9. – pp. 1434–1451.

34. Горбунов И.В. Методы построения трехкритериальных Паретооптимальных нечетких классификаторов [Текст] / И.В. Горбунов, И.А. Ходашинский // Искусственный интеллект и принятие решений. – 2015. – № 2. – с. 75–87.

35. Ходашинский И.А. Алгоритмы поиска компромисса между точностью и сложностью при построении нечётких аппроксиматоров [Текст] / И.А. Ходашинский, И.В. Горбунов // Автометрия. – 2013. – Т. 49, № 6. – с. 51–

61.

36. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions [Текст] / M. Fazzolari, etc. // IEEE Trans. Fuzzy Systems. – 2013. – vol. 21, № 1. – pp. 45–65.

37. J. Alcalá-Fdez, Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation [Текст] / J. Alcalá-Fdez, J. Otero // International Journal of Approximate Reasoning. – 2007. – vol. 44. – pp. 45–64.

38. A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects [Текст] / K. Spalko, K. Lapa, A. Przybył, M. Zalasinski // Neurocomputing. – 2014. – vol. 135. – pp. 203–217.

39. J. Yen, Application of Statistical Information Criteria for Optimal Fuzzy Model Construction [Текст] / J. Yen, L. Wang // IEEE Transaction on Fuzzy Systems. – 1998. – vol. 6, № 3. – pp. 362–372.

40. S.-M. Zhou, Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modeling [Текст] / S.-M. Zhou, J.Q. Gan // Fuzzy Sets and Systems. – 2008. – vol. 159. – pp. 3091–3131.

41. Ходашинский И.А. Построение компактных и точных нечетких моделей на основе статистических информационных критериев [Текст] // Информатика и системы управления. – 2014. – № 1 (39). – с. 99–107.

42. Сарин К.С. Три алгоритма генерации структуры нечеткой системы типа Такаги–Сугено [Текст] / К.С. Сарин, И.А. Ходашинский // ЗнанияОнтологии-Теории (ЗОНТ2015): матер. Всерос. конф. с междунар. участием. Российская академия наук, Сибирское отделение; Институт математики им. С.Л. Соболева, 2015. – с. 124–132.