

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія оптимізації процесу збору відходів в територіальних громадах»

Виконала: студентка 2-го курсу, групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Хазівалієва І.І.

(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН

Месюра В.І.

(прізвище та ініціали)

« » _____ 2022 р.

Опонент: к.т.н., доцент каф. КСУ

Ковалюк О.О.

(прізвище та ініціали)

« » _____ 2022 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

«16» _____ 12 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та
 автоматизації Кафедра комп'ютерних наук
 Рівень вищої освіти II-й (магістерський)
 Галузь знань – 12 «Інформаційні технології»
 Спеціальність – 122 «Комп'ютерні науки»
 Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
 д.т.н., проф. Яровий А.А.

15.09

2022 року

ЗАВДАННЯ


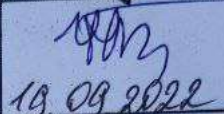
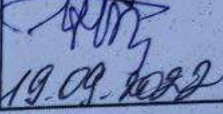
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Хазівалієва Ірина Ігорівна

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія оптимізації процесу збору відходів у територіальних громадах
 керівник роботи к.т.н., професор кафедри КН Месюра В. І.
 затверджені наказом вищого навчального закладу від «14» 09 2022 року № 203
2. Строк подання студентом роботи 18.11 2022 року
3. Вихідні дані до роботи:
 - кількість сміттєвих баків ≤ 10000 ;
 - кількість вантажівок ≤ 50 ;
 - кількість кластерів ≤ 150 ;
 - скорочення довжини маршруту $\geq 5\%$;
 - операційна система Windows 7/8/10.
4. Зміст текстової частини: Вступ, аналіз задачі збору відходів у територіальних громадах, постановка задачі оптимізації збору відходів у територіальних громадах; розробка інформаційної технології оптимізації процесу збору відходів; програмна реалізація інформаційної технології оптимізації процесу збору відходів; тестування; аналіз результатів тестування розробленої інформаційної технології; економічна частина, висновки, перелік використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) Розробка математичної моделі оптимізації процесу збору відходів, Нечітке прогнозування заповненості та доступності сміттєвих контейнерів, Нечітка кластеризація контейнерів, Нечіткий контролер оптимізації значень параметрів генетичного алгоритму, Схема нечіткого генетичного алгоритму, Аналіз результатів тестування програми

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Месюра В.І., к.т.н., проф. каф. КН	 14.09.2022	 14.09.2022
4	Буреннікова Н.В., к.е.н., доц. каф. ЕПВМ	 19.09.2022	 19.09.2022


1. Дата видачі завдання 14.09 2022 року


КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз задачі збору відходів у територіальних громадах. Постановка задач дослідження	14.09.2022 - 01.10.2022	
2	Опис процесу кластеризації та формування нечіткого генетичного алгоритму інформаційної технології	02.10.2022 - 16.10.2022	
3	Програмна реалізація розробленої інформаційної технології, тестування та оцінка параметрів	17.10.2022 - 07.11.2022	
4	Підготовка економічної частини	08.11.22 - 21.11.22	
5	Апробація та/або впровадження результатів дослідження	23.11.22 - 01.12.2022	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.22 - 14.12.2022	

Студент

Керівник роботи


(підпис)


(підпис)

Хазівалієва І. І

Месюра В. І.

АНОТАЦІЯ

УДК 004.8

Хазівалієва І. І. Інформаційна технологія управління процесом збору відходів в територіальних громадах.

Магістерська кваліфікаційна робота зі спеціальності 122–комп'ютерні науки, освітня програма– системи штучного інтелекту. Вінниця: ВНТУ, 2022. 111 с.

Наукр. мові. Бібліогр.: 35 назв; рис.: 18; табл. 21

Магістерська кваліфікаційна робота присвячена розробці інформаційної технології оптимізації процесу збору відходів. У загальній частині роботи досліджено предметну область збору відходів у територіальних громадах. Розглянуто узагальнене управління цим процесом та його деталізацію на прикладі Вінницької територіальної громади.

Проаналізовано методи оптимізації маршруту та обґрунтовано вибір нечіткої логіки для розв'язання задачі. Запропоновано математичну модель формування маршруту сміттєвозу. Розроблено нечіткий підхід до прогнозування заповненості контейнерів та їх доступності. Вирішено задачу нечіткої кластеризації контейнерів, та оптимізації маршруту сміттєвозу з нечіткого генетичного алгоритму. Здійснено проектування інформаційної технології управління процесом збору відходів у територіальних громадах, її програмну реалізацію мовою програмування JavaScript та тестування її результатів.

Ілюстративна частина складається з шести плакатів та кореслень. В економічному розділі наведено комерційний та технологічний аудит науково-технічної розробки, спрогнозовано витрати, проаналізовано рентабельність систем-аналогів, розраховано економічну ефективність та комерційний потенціал розробки, який демонструє абсолютну ефективність вкладених інвестицій при періоді окупності близько 0,153 роки.

ABSTRACT

Khazivaliieva I. I. Information technology of waste collection management optimization in in territorial communities. Master's thesis in specialty 122 - computer science, educational program - artificial intelligence systems. Vinnytsia: VNTU, 2022.

In Ukrainian language. Bibliographer: 35 titles; fig.: 18; tabl. 21.

The master's thesis is devoted to the development of information technology of waste collection management optimization in in territorial communities. The subject area of waste collection in territorial communities is investigated in the general part of the work. The general management of this process is considered and given more detailed example of the Vinnytsia territorial community.

Route optimization methods are analyzed and fuzzy logic for solving the problem is substantiated. A mathematical model of truck routing during waste collection is formulated based on determining the feasibility of including a garbage container to the truck route. The degree of loading of the container and the influence of external factors are used during the research. It's determined by the intensity of traffic on the way to the container and the condition of the road associated with weather conditions.

A solution to the problem using fuzzy clustering and a fuzzy genetic algorithm is proposed. Design of information technology of waste collection management optimization in in territorial communities. The software implementation using the JavaScript programming language and testing of its results was carried out.

The illustrative part consists of posters with simulation results. A commercial and technological audit of the scientific and technical development is given in the economic section. The costs are forecast, the profitability of similar systems is analyzed, the economic efficiency and commercial potential of the development are calculated, which demonstrates the absolute effectiveness of the investments made with a payback period of about 0.153 years.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ.....	12
1 ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНИХ ГРОМАДАХ.....	12
1.1 Аналіз задачі збору відходів у територіальних громадах.....	12
1.2 Управління процесом збору відходів у Вінницькій територіальній громаді.....	15
1.3 Аналіз існуючих методів визначення маршруту	18
1.4 Обґрунтування вибору методу оптимізації маршруту збору відходів у територіальних громадах.....	21
1.5 Постановка задачі визначення оптимального маршруту збору відходів у територіальних громадах.....	22
1.6 Висновки до першого розділу.....	24
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНИХ ГРОМАДАХ	25
2.1 Розробка структури інформаційної технології оптимізації процесу збору відходів у територіальних громадах.....	25
2.2 Розробка математичної моделі оптимізації процесу збору відходів ..	27
2.3 Нечіткий підхід до визначення ступеню заповненості та доступності сміттєвих контейнерів.....	30
2.4 Нечітка кластеризація контейнерів	38
2.5 Оптимізація маршруту збору відходів у територіальних громадах на основі нечіткого генетичного алгоритму.....	44
2.6 Розробка нечіткого контролера для автоматичної оптимізації значень параметрів нечітких генетичних операторів	49
2.7 Висновки до другого розділу	56
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНІЙ ГРОМАДІ	57
3.1 Обґрунтування вибору мови програмування	57
3.2 Розробка структури програмного забезпечення інформаційної технології оптимізації процесу збору відходів у територіальних громадах.....	60

3.3 Розробка програмного забезпечення підсистеми генетичного алгоритму оптимізації процесу збору відходів	66
3.4 Аналіз результатів тестування програми.....	71
3.5 Висновки до третього розділу	74
4 ЕКОНОМІЧНА ЧАСТИНА	75
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки інформаційної технології оптимізації процесу збору відходів у територіальних громадах.....	75
4.2 Прогнозування витрат на виконання науково-технічної інформаційної технології оптимізації процесу збору відходів у територіальних громадах.....	79
4.3 Розрахунок економічної ефективності науково-технічної м інформаційної технології оптимізації процесу збору відходів у територіальних громадах за її можливої комерціалізації потенційним інвестором	88
4.4 Висновки до четвертого розділу.....	93
ВИСНОВКИ.....	93
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96
ДОДАТКИ.....	100
Додаток А Результат перевірки на плагіат в онлайн-системі UNICHECK.....	102
Додаток Б (обов'язковий) Лістинг програми	103
Додаток В ІЛЮСТРАТИВНА ЧАСТИНА	107
Додаток Г Інструкція користувача	111
Додаток Г Довідка про впровадження.....	115

ВСТУП

Актуальність теми дослідження.

Питання екології сьогодні є надзвичайно актуальним. Щороку, за офіційними даними, українці продукують 11 млн тон сміття, це десь 300 кг на людину. Згідно даних Мінрегіонбуду об'єм усього нашого сміття 1267 млн — це 506 пірамід Хеопса. Один з найбільших у Європі полігонів для відходів у Грибовичах під Львовом, де знаходиться гора сміття висотою у 20-поверховий будинок на площі у 50 футбольних полів [1].

Управління процесом збору відходів складається з процесів збору, вивезення, переробки, рециркуляції, моніторингу. Основними недоліками системи управління відходів – це відсутність бази даних поводження з відходами та труднощі в отриманні точних даних, наприклад, про кількість і склад відходів [2].

Економічний фактор в процесі збору сміття важливий як для муніципальних підприємств, так і для звичайних жителів міста. Кожен мешканець багатоквартирного будинку сплачує щомісяця 19,95 грн за вивіз і захоронення ТПВ за 33,9 кг сміття на місяць, що в день складає 1,1 кг [3]. Найбільший вплив має збір і транспортування відходів, адже він становить 60-80% від всієї системи витрат управління процесом збору відходів [4].

Оптимізація маршрутів вивезення сміття територіальних громад в окремий полігон або на станцію утилізації дозволить зменшити їх фінансові витрати та зменшити забруднення сміттєвозами оточуючого середовища.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 "Розробка прикладних інтелектуальних інформаційних технологій та систем" та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є оптимізація маршруту збору відходів на основі нечіткого обчислення ступеню завантаженості та доступності сміттєвих контейнерів і автоматичного налаштування оптимальних значень параметрів нечітких генетичних операторів.

Для досягнення поставленої мети необхідно розв'язати такі основні задачі:

- провести аналіз предметної області процесу збору відходів у територіальних громадах;
- розглянути існуючі методи вирішення задачі маршрутизації збору відходів;
- розробити структуру інформаційної технології оптимізації процесу збору відходів в територіальних громадах;
- розробити математичну модель організації процесу об'їзду контейнерів;
- розробити нечіткій підхід до обчислення ступеню завантаженості та доступності сміттєвих контейнерів;
- розробити нечіткий контролер для автоматичної оптимізації значень параметрів нечітких генетичних операторів;
- здійснити програмну реалізацію інформаційної технології управління процесом збору відходів у територіальних громадах;
- дослідити ефективність запропонованих методів і алгоритмів за допомогою експериментів;

– проаналізувати результати проведених обчислювальних експериментів

Об'єкт дослідження – процеси управління збором відходів у територіальній громаді.

Предметом дослідження є нечіткі моделі, алгоритми та програмні засоби оптимізації маршрутів збору відходів

Методи дослідження. Методичною основою для дослідження послужили роботи вітчизняних і зарубіжних вчених в області нечіткого моделювання транспортних задач і методів нечіткого логічного виведення.

У роботі використано теоретичні та змішані методи наукових досліджень. Використано моделі та методи нечіткого логічного виведення, нечіткої

кластеризації, нечіткої оптимізації генетичних алгоритмів, а також методи об'єктно-орієнтованого проектування і програмування для програмної реалізації запропонованої інформаційної технології управління процесом збору відходів у територіальній громаді.

Наукова новизна одержаних результатів:

- вперше запропоновано нечіткій підхід до визначення ступеню заповненості та доступності смітєвих контейнерів, що дозволяє оптимізувати маршрут смітєвозу за рахунок вилучення із маршруту не повністю заповнених контейнерів;

- удосконалено генетичний алгоритм оптимізації маршруту за рахунок введення нечіткого контролера для автоматичної оптимізації значень параметрів нечітких генетичних операторів, який на відміну від існуючих дозволяє автоматично визначати оптимальні значення ймовірності схрещування та мутації, що забезпечують підвищення точності генетичного алгоритму;

- знайшла подальшого розвитку інформаційна технологія оптимізації процесу збору відходів у територіальних громадах, структура якої відрізняється наявністю трьох відокремлених етапів, що забезпечує зменшення розмірності задачі та відповідне підвищення її продуктивності

Практичне значення одержаних результатів полягає у наступному:

- розроблено алгоритм нечіткого прогнозування ступеня наповненості та доступності смітєвого контейнеру

- розроблено алгоритм оптимізації маршруту вантажівок комунальних підприємств для збору сміття із застосуванням генетичного алгоритму

- на основі методів штучного інтелекту розроблено програмне забезпечення інформаційної технології збору відходів у територіальній громаді.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується математично коректною постановкою наукової задачі, використанням апробованого математичного апарату, глибоким аналізом стану справ в предметній області, чисельною перевіркою алгоритмів при вирішенні

конкретних задач маршрутизації збору відходів програмною реалізацією розроблених методів і пропозицій та результатами виконаних обчислювальних експериментів з розробленими моделями маршрутизації.

Особистий внесок автора. Всі дослідження проведені в даній магістерській кваліфікаційній роботі. У тезах написаних у співавторстві, автору належать такі результати: [1] - розробка нечіткого підходу до прогнозування заповненості та доступності контейнера, [2] - розробка генетичного алгоритму процесу збору міських відходів, [3] - постановка задачі визначення маршруту збору відходів, [4] - аналіз проблеми збору муніципального сміття

Апробація результатів роботи. Основні теоретичні та практичні результати роботи опробовані на міжнародній конференції «Контроль і управління в складних системах - 2022» 15 листопада 2022 р. та на двох науково-технічних конференціях підрозділів Вінницького національного технічного університету: І НТКП ВНТУ-2021, м. Вінниця, 10 – 12 березня 2021 р.; ІІ НТКП ВНТУ-2022, м. Вінниця, 30 – 31 травня 2022 р.

Публікації. На тему магістерської кваліфікаційної роботи опубліковано статтю у журналі категорії Б [1] та 3 тез доповідей [2 –5]

Подано заявку №202204025 від 07.09.22 на отримання автурського права на комп'ютерну програму «Rubish IT».

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНИХ ГРОМАДАХ

1.1 Аналіз задачі збору відходів у територіальних громадах

Територіальна громада – це об'єднання громадян села/ селища, міста в межах одного району, які своєю більшістю можуть вирішувати певні питання [6]. Одним з таких питань є збір відходів в межах своєї громади.

У багатьох громадах України впроваджено сортування та роздільний збір сміття, на що виділяють значну частину свого бюджету. Наступні проблеми, з якими стикнулись мешканці стали утилізація та вивіз побутових відходів.

Раніше, коли кожне село чи місто існувало сам-на-сам, у кожного існували власні комунальні підприємства, які надавали послуги із вивезення відходів, а також були свої тарифи та правила щодо цього процесу. Після децентралізації та приєднання населених пунктів в одну громаду організації, які надавали ці послуги змінились та проблема загострилась [7].

Нині існують різні методи збору відходів, які залежать від типу встановлених контейнер. Обласні центри активно впроваджують новий підхід підземних контейнерів з використанням датчиків рівня сміття, які при певному відсотку наповненості баку передають інформацію до вантажівки. Живлення здійснюється за рахунок сонячних батарей та облаштованих індикаторів. Такий підхід дає змогу зменшити кількість поїздок до ділянок, де немає великої кількості сміття.

Даний спосіб боротьби із засміченими вулицями є досить ефективний, проте він залишається дорого вартісним. Наприклад, згідно з джерела [8], Тернопільська територіальна громада на придбання автономної підземної смітцевої системи європейського зразка у східному мікрорайоні витратила 1 450 000 гривень у 2020 році. Чимало коштів витрачається на встановлення та

підтримання таких майданчиків. На рисунку 1.1 зображено приклад підземних контейнерів.



Рисунок 1.1 – Підземний контейнерний майданчик з датчиками наповненості сміття

В цей же час у маленьких містечках та селах продовжують використовувати звичайні баки нерідко навіть без сортування. Європейський посібник з розумного управління відходами спільнот [9] рекомендує застосовувати такі кроки по запобіганню утворення великої кількості відходів:

1. Мінімізація обсягів відходів по відношенню до обсягів виробництва, росту населення, зайнятості.
2. Підвищення рівня інформованості та залучення уваги громадськості.
3. На виробництвах запроваджувати замкнутий циклресурсів
4. Стимулювати споживачів купувати продукти без упаковки
5. Екомаркування

Втім позбутися сміття повністю не вдасться, тому нині система збору відходів функціонує за наперед спланованими графіками. У 2020 році «Київкомунсервіс» запровадив інтеративну карту з різною корисною інформацією щодо відходів [10]. За допомогою неї жителі міста можуть дізнатись графік вивозу сміття того чи іншого будинку, тип контейнерів у своєму

районі та його прив'язку до певної ділянки, об'єми відходів, відфільтрувати контейнери за типом сміття тощо.

Головною задачею управління процесом збору сміття є прокладання оптимального маршруту між населеними пунктами громади. Суттєвим нюансом є різні типи контейнерів, що викликає невизначеність у кількості сміття. Якщо з розумними ділянками все прозора та зрозуміло, зі звичайними баками, які складають переважну більшість, інша ситуація.

Орієнтуватись на зіставленні графіки збору сміття можна, проте нерідко трапляється ситуація, коли бак напівпорожній або ж навпаки – біля нього вже гора сміття. Проте ігнорувати такі ділянки також неможна.

Велику роль грає місцезнаходження контейнерів, оскільки це впливає на кількість відходів, які потрапляють до них. Так, до міських установ, лікарень, підприємств, навчальних заходів, торгових центрів та інших великих будівель потрібно приїжджати щодня, подекуди й кілька разів на добу. В той час як до поодиноких будинків такої необхідності немає. Якщо ж поруч розташовані контейнери різних типів виникає проблема подвійного прокладання маршруту вантажівок з датчиками та без в один район.

Крім цього при прокладанні маршруту збору відходів, як і в інших транспортних задачах, виникає велика кількість неповноти інформації про майбутню подорож. Одним з таких факторів впливу на поїздку є погодні умови, які передбачити можливо, проте не на сто відсотків, які в свою чергу впливають на швидкість руху, стан дороги, безпечність.

Також відхилення від графіку можуть спричинити можливі технічні або технологічні збої транспорту. На загальний час збору також впливає часове вікно завантаження відходів до кузова вантажівки, завантаженість трафіку, ремонтні роботи доріг при переїзді між населеними пунктами у територіальній громаді [11].

Необхідно враховувати, що до одного контейнеру чи ділянки може під'їжджати одна вантажівка, кількість сміття у кузову вантажівки має бути розраховане так, щоб він не повертався повторно в один й той самий пункт знову.

При затримці в декілька хвилин на одній з точок чи при незвичайній ситуації увесь час поїздки поступово збільшується та відходить від графіку збору відходів.

1.2 Управління процесом збору відходів у Вінницькій територіальній громаді

Вінницька територіальна громада загальною площею 256.6 км² утворилась 2020 року. Вона налічує майже 390 тисяч жителів та складається з 9 населених пунктів: міста Вінниці, селища міського типу Десна та Великих Крушлинців, Вінницьких Хуторів, Гавришівки, Малих Крушлинців, Писарівки, Стадниці та Щіток [12].

За даними програми поводження з побутовими відходами Вінницької територіальної громади [13] на одного мешканця припадає 220-250 кг сміття на рік. Крім цього багато підприємств нехтує правилами поводження з відходами та порушують санітарно-захисну зону. Щороку з території Вінницької громади вивозиться близько 600 тисяч м³ побутових відходів, що приблизно складає 92 тисяч тон.

Контейнери для збору побутових відходів у багатоповерховій забудові встановлюються на спеціально відведених бетонованих чи асфальтованих майданчиках. У Вінниці поступово оновлюють контейнери для збору відходів: конструкції роблять з металопрофілю, який є найбільш практичним варіантом, і за потреби його легко можна замінити [14]. Проте у селах досі залишаються старі варіанти контейнерів.

Нині в експлуатації знаходиться 161 вантажівка, орієнтовно до 2030 року планується купити ще 175 машин. У вищезазначеній програмі вказано, що основою для визначення маршрутів сміттевозів є загальний обсяг побутових відходів, відстань від місць обслуговування до місць розміщення, кількість і тип сміттевозів і їхня продуктивність, необхідність обов'язкового дотримання графіків вивозу відходів.

Велика частина територій обслуговується у нічний час. Найбільша частина сміття потрапляє до Стадницького полігону, який має площу майже 14 га, який накопичує близько 13 млн. м³ відходів. Також сміття у себе розміщає комунальне підприємство «ЕкоВін», вантажівки якого нерідко можна побачити у місті.

Оцінити загальну ситуацію з відходами у громаді можна за допомогою SWOT-аналізу, який наведений у регіональному плані управління відходами Вінницької області на період до 2030 року [15].

Для покращення даної ситуації доцільним є правильне управління логістикою в громаді, яка має відповідати технологічній схемі, що наведена на рисунку 1.2, що затверджена регіональним планом збору відходів.

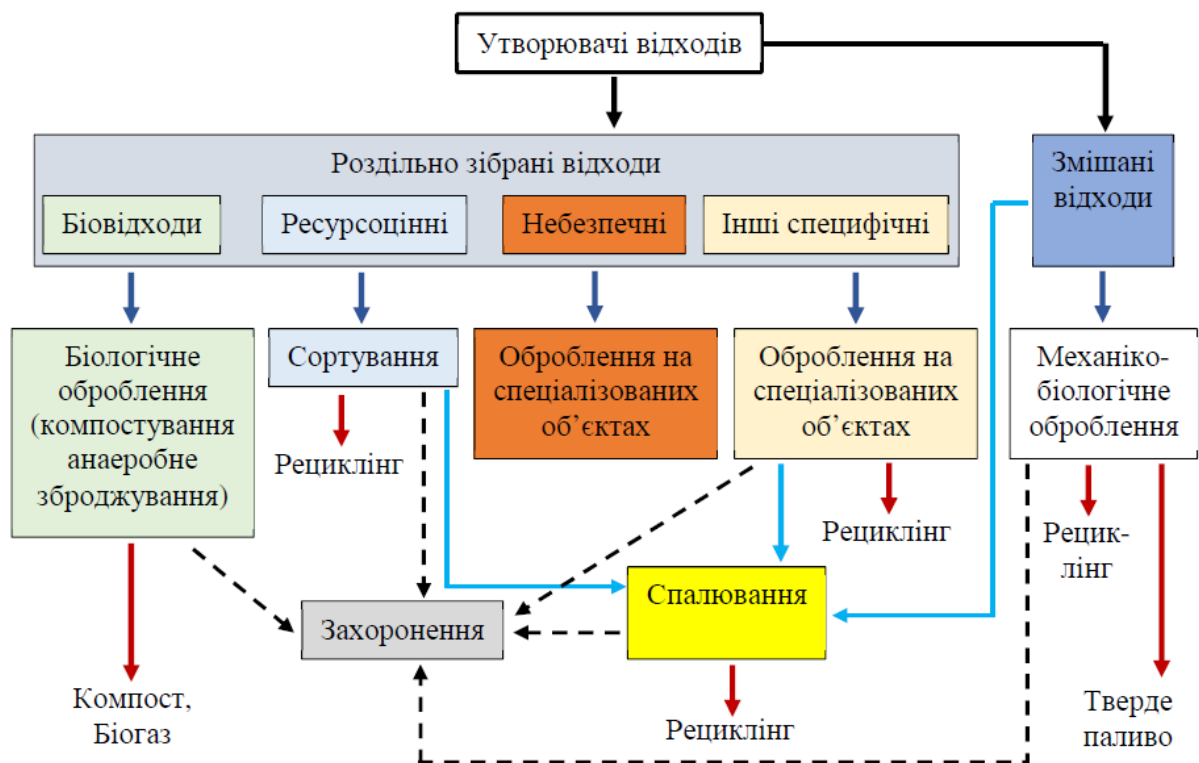


Рисунок 1.2 – Схема технологічної логістики управління відходами

Таблиця 1.1 –SWOT-аналіз управління відходами у Вінницькій громаді

Внутрішні сильні сторони	Внутрішні слабкі сторони та загрози
<ul style="list-style-type: none"> - наявність в області позитивного досвіду співробітництва територіальних громад у сфері управління відходами; - наявність позитивного досвіду співфінансування з бюджетів різних рівнів ініціатив громад у сфері управління відходами; - наявність у суб'єктів господарювання усталеної практики управління власними відходами: промислові, небезпечні та сільськогосподарства. 	<ul style="list-style-type: none"> - низька якість первинного обліку, узагальнення та аналізу даних щодо потоків відходів на усіх етапах управління відходами; - низький організаційний рівень послуги управління муніципальними відходами в багатьох населених пунктах; - відсутність в області системи управління відходами: будівництва та знесення, небезпечними ускладі побутових відходів; - велика кількість місць захоронення відходів, більшість з яких не мають обслуговуючої організації; - наявність на території області об'єктів, на яких накопичені небезпечні відходи; - низький рівень охоплення населених пунктів послугами з роздільного збирання сміття; - відсутність моніторингу впливу на довкілля на більшості місць захоронення.
Зовнішні можливості	Зовнішні загрози
<ul style="list-style-type: none"> - фінансування з державного бюджету; кредитного фінансування; інвестицій - залучення експертної та фінансової грантової допомоги. 	<ul style="list-style-type: none"> - невизначеність майбутніх економічних механізмів управління відходами як загроза економічній стабільності

1.3 Аналіз існуючих методів визначення маршруту

На сьогодні відомо багато методів визначення оптимального маршруту. Усі вони зводяться до знаходження шляху через пункти призначення за найменший або близький до нього час. Основою програмних продуктів такого типу є використання методів штучного інтелекту для досягнення поставленої мети. Серед таких виділяють точні, евристичні алгоритми, нейронні мережі та нечітку логіку.

Згідно з результатами попередніх досліджень [16] за великої кількості точок маршруту евристичні алгоритми працюють набагато швидше за точні. Серед евристичних алгоритмів, найкращі результати за часом пошуку та довжиною знайденого шляху показав генетичний алгоритм.

Генетичний алгоритм пошуку маршруту заснований на природному відборі, основою якого є мутації генів та їх комбінації, які формуються та передаються нащадкам. Виживають особини з максимальною функцією пристосованості. Приклад генетичних операторів зображено на рисунку 1.3.

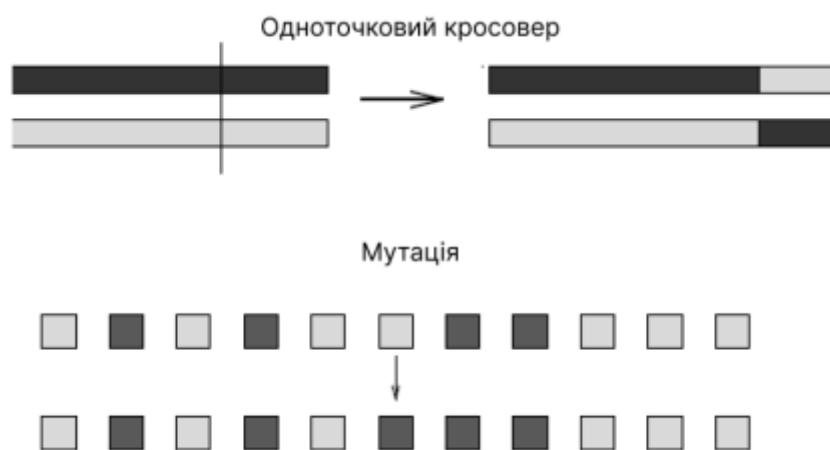


Рисунок 1.3– Приклади генетичних операторів

В алгоритмі присутні переборний та градієнтний методи, що дозволяє знайти точку максимуму цільової функції та обирати випадкові значення параметрів, які будуть поступово змінюватися для росту цільової функції. Їх комбінація дає вигреш у часі, проте оптимальний результат не завжди буде отриманий з першого проходження алгоритму [17].

Методи, засновані на використанні штучних нейронних мереж, дають змогу значно підвищити оперативність рішення цього класу задач, забезпечуючи достатню точність результату. Ефективними є: мережа Хопфілда, Кохонена та рекурентна нейронна мережа. Оглянемо принцип дії кожної з них.

Мережа Хопфілда є повнозв'язною нейронною мережею з симетричною матрицею зв'язків, що функціонує до моменту, коли наступний стан мережі в точності дорівнює попередньому: початковий стан є вхідним образом, а при рівновазі отримують вихідний образ.

Для визначення маршруту кожна точка маршруту задається як шар мережі, після чого складається матриця з N-шарів, яка враховує вагу ребер між вузлами. Далі вводять порогову оцінку для того щоб бінаризувати значення. Такий процес повторюється декілька разів, поки матриця не стане оптимальною. Приклад структури нейронної мережі Хопфілда для визначення маршруту наведено на рисунку 1.4 [18].

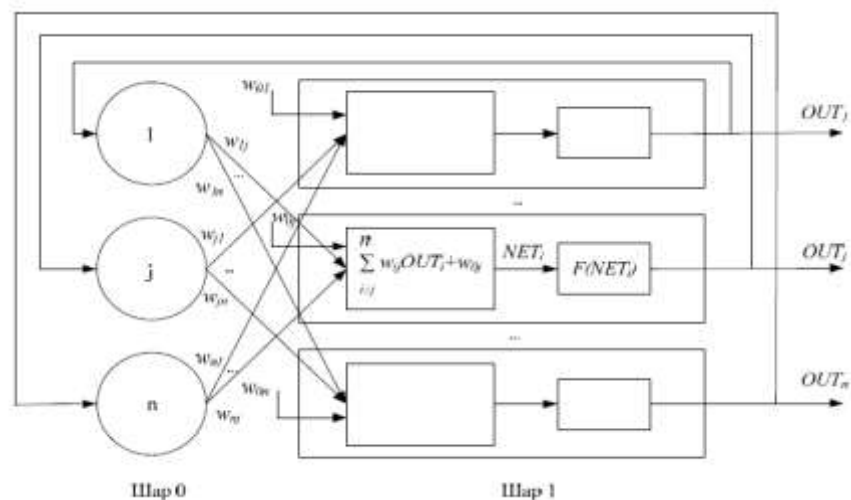


Рисунок 1.4 – Структура нейронної мережі Хопфілда для визначення маршруту

Популярним методом вирішення задачі комівояжера є застосування рекурентної нейронної мережі. В таких мережах виходи нейронних елементів наступних шарів мають синаптичні з'єднання з нейронами попередніх шарів. Це призводить до можливості обліку результатів перетворення нейронною мережею інформації на попередньому етапі для обробки вхідного вектора на наступному етапі функціонування мережі.

Вона використовується при задачах великих розмірностей. Деревовидна структура алгоритму передбачає часто розпаралелення процесів для знаходження оптимального маршруту, адже процеси в різних гілках можуть відбуватись незалежно один від одного. Проте таке розпаралелення використовує значний обсяг пам'яті [19].

Мережа Кохонена Використовується мережа, яка складається з двох одновимірних шарів нейронів. Вхідний шар складається з трьох нейронів, а вихідний – з N (за кількістю міст). Кожен нейрон вхідного шару пов'язаний із кожним вихідним нейроном. Після завершення процесу навчання, розташування міст у маршруті визначиться положенням його образу в кільцевому вихідному шарі. Недоліком є те, що трапляються випадки, у яких два або більше міст відображаються на той самий вихідний нейрон. У випадку, коли міст більше 10 така мережа може оптимізувати лише 25% маршруту. Для сотень міст вона рідко покращує результат. На рисунку 1.5 показано приклад мережі Кохонена для пошуку шляху [19].

Нечітка логіка застосовується у випадках, коли інформація є неповною, неточною, вона використовує логіко-лінгвістичні змінні, які ближчі до людського мислення і нечіткості реального світу. Основою такого підходу є використання нечітких підмножин, ступені належності елементів яких певним чітким множинам знаходяться у межах від 0 до 1, де 0 – абсолютна належність відповідній чіткій множині, а 1 – повна неналежність.

База знань таких систем подається нечіткими продукційними правилами типу «ЯКЩО [нечітка лінгвістична змінна передумови], ТО [нечітка лінгвістична змінна висновку]» [20].

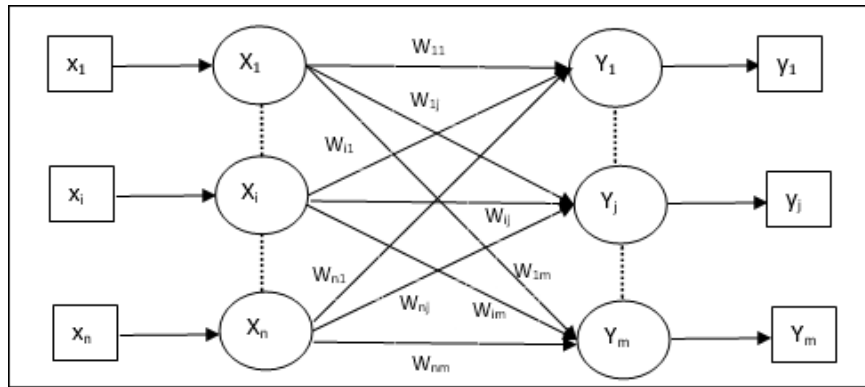


Рисунок 1.5– Структура нейронної мережі Кохонена для визначення маршруту

На входи системи нечіткого виведення можуть подаватися як нечіткі, так і чіткі значення. В останньому випадку чіткі данні фазифікуються для подальшої обробки методами нечікої логіки. Навчання нечіткої моделі для визначення маршруту може здійснюватися шляхом оптимізації параметрів функції належності та вагових коефіцієнтів правил нечіткої бази знань. В результаті обчислення істинності кожного правила здійснюється логічне виведення висновку. Отримані висновки агрегуються в нечітку відповідь і можуть бути дефазифіковані у чіткій результат [21].

1.4 Обґрунтування вибору методу оптимізації маршруту збору відходів у територіальних громадах

Під час збору відходів важливими є етапи визначення точок об'їзду вантажівками та прокладання маршруту до них. Для оптимізації часу поїздки необхідно «відкинути» контейнерні майданчики, які можуть бути пустими або напівпустими, а також врахувати фактори доступності сміттєвих контейнерів, що впливають на складання маршруту.

Суттєвим при цьому є врахування неповноти інформації щодо наповненості сміттєвих контейнерів та до значень факторів впливу під час поїздки. Наприклад, щоденним фактором є погода та стан покриття дороги, що часто затримують водіїв,

особливо в холодний період. Також впливає й сам тип автомобіля, день тижня, часу доби, можливі технічні несправності, завантаженість трафіку і т.д [22].

Здійснений аналіз літератури виявив успішні приклади вирішення задачі прокладання маршруту з використанням нечіткої логіки. Наприклад, у [23] доведено ефективність застосування нечіткої логіки для оптимізації маршрутів Хорватії шляхом прогнозування можливих затримок під час транспортування та відповідного корегування шляхів, що знижує транспортні витрати. У [24] показано, як застосування нечітких вікон при прокладанні маршруту забезпечує мінімізацію загальної довжини маршруту та вартості доставки товару.

На прикладі оптимізації маршрутів України [25] за допомогою введення нечітких коефіцієнтів та врахування параметрів транспортної системи показано, як нечіткі фактори впливають на маршрут перевезень. Запропоновано збільшення ваг ребер графу при поганому дорожньому покритті та трафіку і зменшення ваг при переході на нормальний стан дороги та незавантажений трафік, що покращило процес формування маршрутів.

Вихідні дані нашої задачі характеризуються неповнотою інформації з наповненості контейнерів, погодних умов, стану покриття дороги, трафіку під час поїздки сміттєвозу і т. ін. Зважаючи на наведені приклади робимо висновок, що для розв'язання задачі збору відходів використання нечіткої логіки у поєднанні з генетичним алгоритмом матиме кращий результат, ніж використання нейронних мереж.

1.5 Постановка задачі визначення оптимального маршруту збору відходів у територіальних громадах

Задача збору відходів функціонально зводиться до вирішення задачі комівояжера, що ускладнена неповнотою інформації про майбутній маршрут та його об'єкти.

Подамо розташування сміттєвих контейнерів територіальної громади як нечіткий граф $\tilde{G} = (X, \tilde{F})$, де множина вершин $X = \{x_i\}$; $i \in I = \{1, 2, \dots, n\}$, подає

множину контейнерів, а нечітка множина ребер $\tilde{F} = \{\mu_{\tilde{F}}(x_i, x_j)/(x_i, x_j)\}, (x_i, x_j) \in X^2$ відображає ступень доступності фрагментів доріг, які зв'язують контейнери x_i та x_j між собою.

$$\mu_{\tilde{F}_1}(x_i, x_j)/(x_i, x_j) \quad (1.1)$$

На множині вершин X визначимо нечітку підмножину заповнених контейнерів $\tilde{V} = \{v_i\}$, значення функції належності $\mu_{\tilde{V}}(x_i)$ якої визначатиме ступень наповненості контейнера x_i . На рисунку 1.7 показано приклад нечіткого графу, де вершини позначають контейнерів, а ребра – можливі варіанти проїзду. При цьому зафарбована вершина – це повний контейнер, незафарбована – пустий, напівзафарбована – недостатньо заповний.

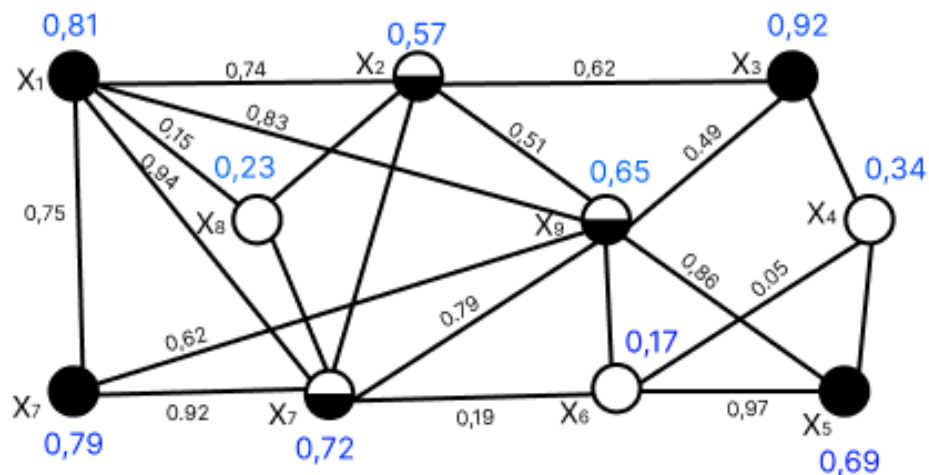


Рисунок 1.7 – Нечіткий граф маршрутів збору відходів

При формуванні маршруту необхідно враховувати такі фактори:

- кожна вантажівка має унікальний набір контейнерів та відвідує кожен контейнер один раз за один маршрут;
- кожен маршрут розпочинається з початкової точки (гаража вантажівки) та закінчується станцією утилізації;

- якщо в кузові вантажівки не вистачає місця для забору сміття з наступного контейнера, водій прямує до найближчої станції утилізації;
- якщо погодні умови незадовільні та майданчики заповнені неповністю, то точку розташування контейнер можна пропустити до покращення умов;
- якщо поїздка потрапляє на години пікової завантаженості дороги, водій має їхати об'їздними шляхами або змінити порядок об'їзду контейнерів;

При розрахуванні відстані потрібно враховувати, що вантажівка проїжджає між кварталами, тому доцільно використовувати Манхеттенську відстань, яка обчислюється за формулою

$$p(x_i, y_i) = \sum_i^n |x_i - y_i|, \quad (1.2)$$

де $x = (x_1, x_2, \dots, x_n)$ та $y = (y_1, y_2, \dots, y_n)$ – координати i -го контейнера, який необхідно проїхати.

Нехай d – відстань, яку необхідно проїхати, c_{ij} – відстань між контейнером i та j , $i \neq j$. Необхідно знайти таку впорядкованість контейнерів, яка б мінімізувала значення тривалості маршруту:

$$\sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(n)}, c_{\pi(1)}) \quad (1.3)$$

При цьому d має задовільняти умови:

1. $d(c_i, c_j) \geq 0$ для $1 \leq i, j \leq n$ та $d(c_i, c_j) = 0$ тоді і тільки тоді, коли $i = j$.
2. $d(c_i, c_j) = d(c_j, c_i)$ при $1 \leq i, j \leq n$.
3. $d(c_i, c_j) \leq d(c_i, c_k) + d(c_k, c_j)$ при $1 \leq i, j, k \leq n$.

1.6 Висновки до першого розділу

1. Проаналізовано предметну область збору відходів у територіальних громадах, де показано основні варіанти збору сміття в залежності від типу контейнеру.

2. Проведено аналіз збору відходів на території Вінницької територіальної громади на основі регіонального плану поводження з відходами до 2030 року, який надав можливість оцінити реальну ситуацію зі сміттям в межах громади

3. Досліджено відомі методи пошуку маршруту для вирішення задачі формування маршруту проїзду вантажівок до контейнерів, що зводиться до задачі комівояжера з обмеженнями, яке засвідчило доцільність використання генетичного алгоритму для оптимізації маршруту збору відходів та застосування інструментарію нечіткої логіки, що дозволяє компенсувати неповноту вхідної інформації щодо наповненості сміттєвих контейнерів та їх доступності, а також підвищити ефективність роботи генетичного алгоритму за рахунок використання нечітких генетичних операторів.

4. Сформульовано постановку задачі збору відходів у територіальних громадах, що дозволило врахувати основні обмеження задачі та визначити найбільш зручну форму її подання у вигляді нечіткого графу.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНИХ ГРОМАДАХ

2.1 Розробка структури інформаційної технології оптимізації процесу збору відходів у територіальних громадах

Процес збору відходів у територіальних громадах складається зі збору та вивезення їх до станції утилізації. Підвищення ефективності процесу формування маршрутів вивезення сміття пропонується здійснити за рахунок декомпозиції задачі на три підзадачі, що дозволить значною мірою зменшити розмірність, що показано на рисунку 2.1.



Рисунок 2.1 – Загальна структура інформаційної технології оптимізації процесу збору відходів у територіальних громадах

Етап підготовки

При організації управління вивезенням відходів виникають ситуації невизначеності рівня сміття на певних ділянках маршрутів, що призводить до марних витрат на об'їзд пустих або недостатньо заповнених контейнерів.

Додаткові проблеми виникають при цьому з організацією вивезення сміття у межах територіальних громад, де маршрути сміттевозів включають фрагменти шосейних доріг та путівців, що вимагає врахування при оптимізації маршрутів додаткових параметрів, пов'язаних з урахуванням завантаженості шосейних доріг та стану путівців в періоди складних погодних умов.

Отже, на етапі підготовки доцільно з використанням нечіткої логіки оцінити рівень заповненості та доступності смітєвих контейнерів та вилучити з маршрутів незаповнені і/або недоступні контейнери.

Етап кластеризації

Для зручності організації управління вивезенням сміття величезну кількість смітєвих контейнерів територіальної громади необхідно кластеризувати (розподілити) на окремі маршрути. Для зменшення розмірності задачі доцільно здійснити дворівневу кластеризацію:

- чітку кластеризацію верхнього рівня, яка для спрощення задачі управління розподілить територію громади на «адміністративні» кластери;
- нечітку кластеризацію нижнього рівня, метою якої є розподілення «адміністративних» кластерів на окремі маршрути сміттевозів.

При цьому на етапі попереднього формування маршрутів до кожного з кластерів будуть включатися лише достатньо наповнені та доступні за результатами нечіткого прогнозування контейнери, що належать відповідному маршруту.

Нечіткість кластерів дозволить в режимі реального часу динамічно коригувати маршрут сміттєвоза в разі виявлення в процесі забору сміття переоцінки/недооцінки завантаженості контейнерів при попередньому прогнозуванні, за рахунок контейнерів зі значеннями функції належності маршруту меншими за 0,5.

Етап маршрутизації

Оскільки сміттєві контейнери розташовані в різних частинах кластеру, об'їжджати їх можна в різному порядку, тобто формувати різні маршрути. При цьому водію необхідно об'їхати один контейнер лише один раз в межах одного маршруту. Вирішення цієї проблеми запропоновано з використанням нечіткого генетичного алгоритму. При цьому пропонується під час формування маршрутів, автоматично налаштовувати оптимальні значення ймовірності генетичних алгоритмів.

2.2 Розробка математичної моделі оптимізації процесу збору відходів

Охарактеризуємо кожен кластер парою параметрів:

$$n_i = (Q_i, R_i), \quad i = 1, \dots, m \quad (2.1)$$

де Q_i – кількість контейнерів, що необхідно об'їхати в i -му кластері; допускається спільне перевезення одним транспортним засобом їх відходів, а їх кількість відходів контейнерів Q_i , вимірюється в одних і тих самих одиницях (кг, т); R_i – технологічні координати (x_i, y_i) i -го пункту (географічні, маршрутні).

Диспетчер комунального підприємства може керувати маршрутом з f типами контейнерів. Кожен тип контейнера характеризується параметрами: O_i – наповність (вантажопідйомність), яка вимірюється в тих же одиницях, що і Q_i ; C_i

[грн /км] – вартістю збору сміття на 1 км. Співвідношення параметрів Q і O є таким, що для кожного з контейнерів знайдеться, принаймні один водій з достатньою наповненістю кузова:

$$\max \{O\} \geq \max \{Q\} \quad (2.2)$$

Може виявитися, що місткість кузова однієї машини є достатньою для збору відходів зі контейнерів, тобто виконана умова:

$$O_i \geq \sum_{i=1}^m Q_i \quad (2.3)$$

В такому випадку допустимим є вибір кільцевого або однорейсового маршруту, що можна подати як вирішення задачі комівояжера, яке буде залежати від співвідношення вартості перевезення. Тобто маршрут може бути як оптимальний, так і близький до оптимального, бо він не зіставляється з допустимими рішеннями збору іншими транспортними засобами в кілька рейсів, які є більш дешевими та з меншою вантажопідйомністю.

Існує випадок, коли співвідношення між параметрами Q_i , і O_i є таким, що виключається завантаження в одну вантажівку відходів більше ніж одного контейнеру. У цьому випадку оптимальний план містить $K = n$ рейсів – маршрутів заїзду в кожну з точок із поверненням в точку виїзду.

У загальному випадку, кількість рейсів-маршрутів задовольняє умові: $1 < R < n$.

Визначимо параметри плану збору: K – кількість рейсів-маршрутів, k - 1,2, ... K – номер рейсу, G – кількість контейнерів, включених до рейсу k :

$$g_1 + g_2 + \dots + g_r = n g_r \in \{1, 2, \dots, n\} \quad (2.4)$$

Маршрут збору містить розбиття номерів контейнерів $\{1, 2, \dots, n\}$ на підмножини, що не перетинаються в порядку об'їзду:

$$\{1, 2, \dots, n\} = \bigcup_r \{i_1^k, i_2^k, \dots, i_{g_r}^k\} \quad (2.4)$$

де i_g^k – контейнера який є g -м в черзі k -го маршруту.

Для станції утилізації відведемо номер $i = 0$ і встановимо, що початковим i_0^k та кінцевим $i_{1+g_k}^k$ пунктами k -го маршруту є станція:

$$i_0^k = i_{1+g_k}^k = 0; \quad k=1, \dots, K. \quad (2.5)$$

Всю визначену сукупність параметрів маршруту збору подамо вектором:

$$X = (K, g_1, \dots, g_r, i_g^k, \dots, i_{g_k}^k) \quad (2.6)$$

Довжина маршруту k -го рейсу визначиться виразом:

$$\sum_{g=1}^{1+g_k} d[i_{g-1}^k, i_g^k] \quad (2.7)$$

де $d[i_{g-1}^k, i_g^k]$ – відстань між відстань між парою сусідніх контейнерів у маршруті, сукупність яких утворює матрицю відстаней $D(i, g)$.

Для кожного рейсу має бути виконана умова вантажомісткості:

$$\sum_{g=1}^{1+g_k} Q_{i_g^k} \leq O_{g_k} \quad (2.8)$$

Сумарна вартість збору відходів у громаді визначається цільової функцією A :

$$A(X) = \sum_{k=1}^K C_r \times \sum_{g=1}^{1+g_k} d[i_{g-1}^k, i_g^k] \quad (2.9)$$

2.3 Нечіткий підхід до визначення ступеню заповненості та доступності сміттєвих контейнерів

Подамо мапу розташування сміттєвих контейнерів територіальної громади як нечіткий граф $\tilde{G} = (X, \tilde{F})$, множина $X = \{x_i\}; i \in I = \{1, 2, \dots, n\}$ вершин якого подає множину контейнерів, а нечітка множина $\tilde{F} = \{\mu_{(1)}(x_i, x_j)/(x_i, x_j)\}, (x_i, x_j) \in X^2$, ребер графа – фрагменти доріг, які зв'язують контейнери між собою.

На множині вершин X визначимо нечітку підмножину заповнених контейнерів $\tilde{V} = \{\mu_{\tilde{V}}(x)/x\}$, значення функції належності $\mu_{\tilde{V}}(x_i)$ якої визначатиме ступень наповненості контейнера x_i . Значення функції належності $\mu_{\tilde{F}}(x_i, x_j)/(x_i, x_j)$ визначатиме ступень «доступності» фрагмента дороги між контейнерами x_i та x_j .

Визначимо вхідні, $X_1 - X_3$, та вихідну Y лінгвістичні змінні:

- X_1 = Наповненість контейнера (пустий, напівповний, майже повний, повний);
- X_2 = Погода (погана, нестійка, гарна);
- X_3 = Завантаженість дороги (ніч, ранній пік, день, вечірній пік, вечір);
- Y = Актуальність забору (низька, середня, висока).

Для подання термів лінгвістичних змінних використаємо квазідзвіноподібну функцію належності $\mu_{\tilde{T}}(x)$, яка характеризує ступінь належності х нечіткому терму \tilde{T} в діапазоні $[0, 1]$ [26], як наведено на рисунку 2.1.

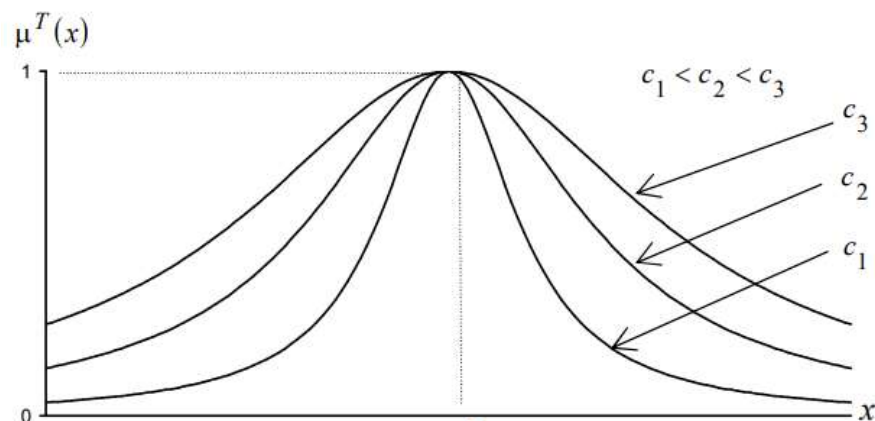


Рисунок 2.1 – Графічне подання квазідзвіноподібної функції

Така функція належності описується формулою:

$$\mu_{\tilde{T}}(x) = \frac{1}{1 + \left(\frac{x-b}{c}\right)^2} \quad (2.10)$$

Визначимо терми лінгвістичних змінних та побудуємо їх функції належності. Сьогодні багато населених пунктів впроваджують контейнерні майданчики з датчиками наповненості відходів. Проте у більшості випадків поки ще викорстовуються звичайні контейнери, інформація про наповнення яких розраховується на основі статистичних даних або може бути повідомлена через месенджер чи телефонний дзвінок (наприклад, у святкові або «особливі» дні, коли відбуваються певні масові події).

Параметр x_1 – «Наповненість контейнеру», що поданий на рисунку 2.2. $T(x_1) = \langle \text{пустий, напівповний, майже повний, повний} \rangle$. Даний фактор визначено на універсальній множині $U(y_1) = [0; 100]$, яка подає вимір відсотку завантаженості контейнера. Терми подаються нечіткими множинами:

Порожній = $\{ \langle 1/0 \rangle; \langle 1/10 \rangle; \langle 0.7/20 \rangle; \langle 0.35/25 \rangle; \langle 0.1/30 \rangle \}$;

Напівпорожній = $\{ \langle 0/0 \rangle; \langle 0.15/10 \rangle; \langle 0.7/20 \rangle; \langle 0.93/25 \rangle; \langle 1/30 \rangle \}$;

Напівповний = $\{ \langle 0/15 \rangle; \langle 0.1/25 \rangle; \langle 0.19/30 \rangle; \langle 0.84/40 \rangle; \langle 1/50 \rangle \}$;

Повний = $\{ \langle 0/60 \rangle; \langle 0.19/65 \rangle; \langle 0.91/70 \rangle; \langle 0.98/75 \rangle; \langle 1/95 \rangle \}$

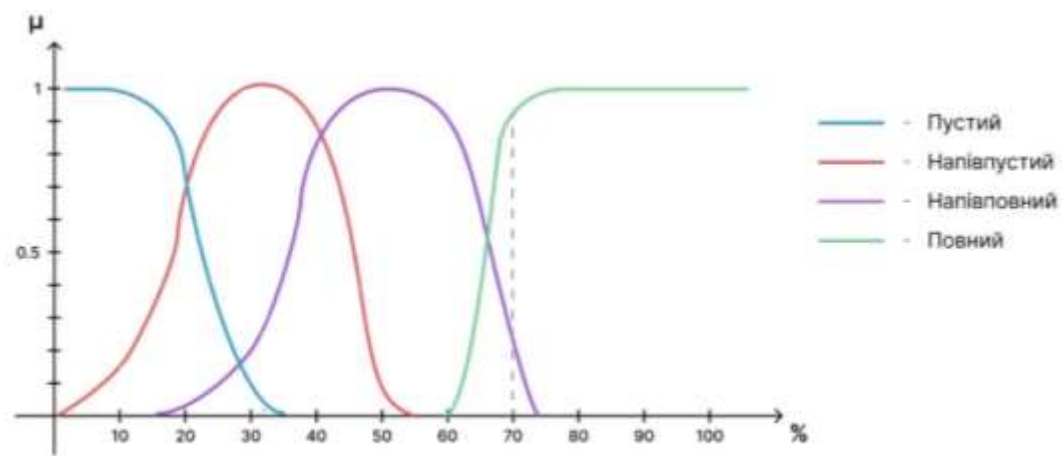


Рисунок 2.2 – Графічне подання параметра «Наповненість контейнеру»

Одним з вагомих факторів доступності контейнера є погода, яка змінює видимість, стан і прохідність (ожеледиця, снігові заноси) дороги, що призводить до зміни поведінки водія. Негода веде до зниження швидкості руху, збільшення відстані між транспортними засобами та кількості аварій [27]. Вплив погодних умов на рух автомобілю відображено в Таблиці 2.1. Водії відчують затори, в той час як транспортні оператори спостерігають зниження пропускної здатності дороги.

Проблема використання статичного підходу полягає у тому, що погода може змінюватись під час поїздки, що збільшує її тривалість. Тому при врахуванні даного фактору доцільним є використання динамічного підходу.

Таблиця 2.1 – Вплив погодних умов на водіїв

Погодні умови	Значення	Одиниці виміру	Зниження потужності двигуна (%)	Зниження швидкості (%)
Температура	1-10	°С	1	1 – 1.5
	0 – (-20)	°С	1.5	1 – 2
	<-20	°С	6 – 10	0.3 – 6
Швидкість вітру	16 – 32	км/год	1 – 1.5	1
	>32	км/год	1 – 2	1 – 1.5
Видимість	1.6 – 0.82	Км	9	6
	0.8 – 0.4	Км	11	7
	<0.4	Км	10.5	11

Параметр x_2 – «Погода». $T(x_2) = \langle \text{погана, погіршена, гарна} \rangle$. Даний фактор визначено на універсальній множині $U(y_3) = [0; 100]$, яка вимірює умовне відсоткове зниження якості дороги внаслідок зміни погоди. Графічне подання терм множини $T(x_2)$ наведено на рисунку 2.3.

Терми подаються нечіткими множинами:

Гарна = $\{ \langle 1/10 \rangle; \langle 0.96/20 \rangle; \langle 0.92/25 \rangle; \langle 0.3/30 \rangle; \langle 0.18/33 \rangle \}$;

Нестійка = $\{ \langle 0/10 \rangle; \langle 0.12/20 \rangle; \langle 0.47/25 \rangle; \langle 0.9/30 \rangle; \langle 1/40 \rangle \}$

Погана = $\{ \langle 0/40 \rangle; \langle 0.05/50 \rangle; \langle 0.39/55 \rangle; \langle 0.92/60 \rangle; \langle 1/80 \rangle \}$

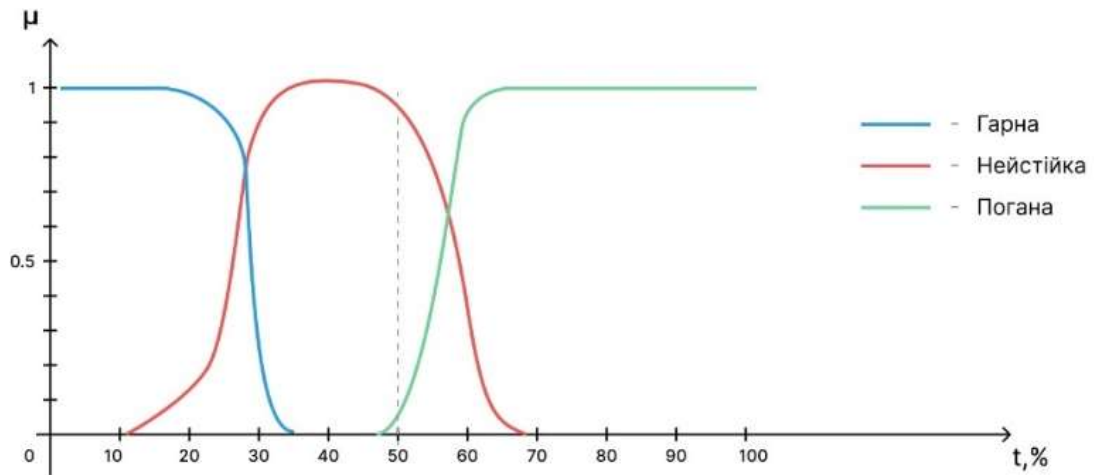


Рисунок 2.3 – Графічне представлення параметра «Погодні умови»

Суттєвий вплив на тривалість поїздки сміттєвозу має завантаженість трафіку. Ступінь завантаженості значною мірою залежить від часу доби. Обрахуватиме його на основі наявних статистичних даних або отримуватимемо з встановлених вздовж дороги камер відеоспостереження за трафіком [28]. Окремо виділимо два пікових періоди: ранковий, рух транспорту в напрямку до роботи/школи, та вечірній, коли усі повертаються додому.

Параметр x_3 – «Завантаженість дороги».

Максимальна завантаженість доріг припадає на періоди з сьомої до десятої години ранку та з сімнадцятої до дев'ятнадцятої години вечора. Таким чином, для симуляції ситуації, що змінюється, на дорогах було прийнято рішення використати погодинну матрицю.

Для симуляції різного стану доріг у різні дні тижня вирішено додається також випадкове значення певного діапазону швидкості, на яку ділиться відстань між двома точками. Результати показані у таблиці 2.2. Розрахунок швидкості руху вантажівок здійснюється наступним чином:

$$v = v' + rand(-v'', +v'') \quad (2.11)$$

v' – середня швидкість в визначений час доби

v'' – відхилення від швидкості

Таблиця 2.2 – Середня швидкість в різний час доби у межах громади

Години	Середня швидкість	Відхилення від швидкості
7 ⁰⁰ - 10 ⁰⁰	25 км/год	5 км/год
10 ⁰⁰ - 17 ⁰⁰	35 км/год	8 км/год
17 ⁰⁰ - 19 ⁰⁰	23 км/год	3 км/год
19 ⁰⁰ - 00 ⁰⁰	38 км/год	10 км/год

$T(x_3) = \langle \text{ніч, ранній пік, день, вечірний пік, вечір} \rangle$. Даний фактор визначено на універсальній множині $U(x_3) = [0; 24]$, яка вказує години доби. Графічне подання терм множини $T(x_3)$ наведено на рисунку 2.4. Терми подаються нечіткими множинами:

Ніч = $\{ \langle 0/1 \rangle; \langle 1/3 \rangle; \langle 0.6/4 \rangle; \langle 0.96/4 \rangle; \langle 0.27/6 \rangle \}$;

Ранній пік = $\{ \langle 0/1 \rangle; \langle 0.1/3 \rangle; \langle 0.73/6 \rangle; \langle 1/8 \rangle; \langle 0.19/12 \rangle \}$;

День = $\{ \langle 0/5 \rangle; \langle 0.18/9 \rangle; \langle 0.31/10 \rangle; \langle 1/12 \rangle; \langle 0.5/15 \rangle \}$;

Вечірний пік = $\{ \langle 0/10 \rangle; \langle 0.22/13 \rangle; \langle 0.58/15 \rangle; \langle 1/17 \rangle; \langle 0.19/20 \rangle \}$

Вечір = $\{ \langle 0/16 \rangle; \langle 0.35/19 \rangle; \langle 0.8/20 \rangle; \langle 0.93/21 \rangle; \langle 1/23 \rangle \}$

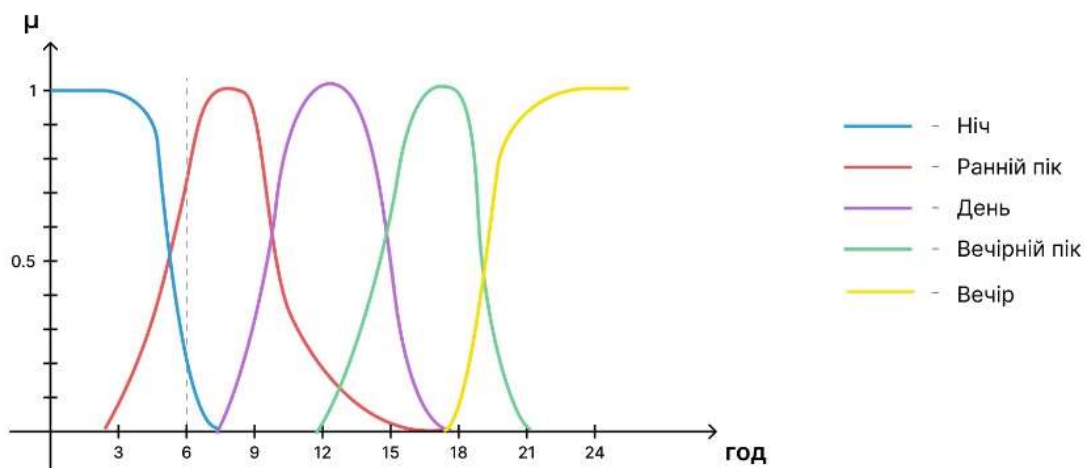


Рисунок 2.4– Графічне представлення параметра завантаженості дороги

Вихідна лінгвістична змінна Y – «Актуальність забору», визначає доцільність включення контейнера до маршруту сміттєвоза: $T(y) = \langle \text{низька, середня, висока} \rangle$. Даний фактор визначено на універсальній множині $U(x_3) = [0; 100]$, яка вказує

відсотковий вимір актуальності забору сміття. При цьому низька актуальність знаходиться в межах 0-40%, середня – 40-70%, висока – 70-100%.

Для використання алгоритму нечіткого логічного виведення Мамдані [29] сформовано нечітку базу знань, що містить 60 правил побудованих на основі нечіткої продукційної моделі висловлювань «ЯКЩО – ТО», які пов'язують вхідні змінні $x_1 - x_w$ з одним з можливих вихідних рішень. Приклади нечітких продукційних правил бази знань наведено на рисунку 2.5. Приклад фрагменту матричної нечіткої бази знань показаний у таблиці 2.3.

$$\begin{aligned} & \text{ЯКЩО}(X_1 = a_1^{11}) \text{I} (X_2 = a_2^{11}) \text{I} \dots \text{I} (X_w = a_w^{11}), \\ & \text{АБО}(X_1 = a_1^{12}) \text{I} (X_2 = a_2^{12}) \text{I} \dots \text{I} (X_w = a_w^{12}), \\ & \text{АБО} \dots (X_1 = a_1^{1\delta_1}) \text{I} (X_2 = a_2^{1\delta_1}) \text{I} \dots \text{I} (X_w = a_w^{1\delta_1}), \\ & \text{ТО } Y = d_1 \text{ і т. д.} \end{aligned} \quad (2.12)$$

де x – вхідна змінна, a з коефіцієнтами – терм відповідної вхідної лінгвістичної змінної, δ_q – кількість правил, що визначають значення вихідної змінної y .

Правило 1: ЯКЩО «Наповненість контейнера» = «напівповний» I «Погода» = «погана» I «Завантаженість дороги» = «ранній пік», ТО «Актуальність забору» = «низька».

Правило 2: ЯКЩО «Наповненість контейнера» = «напівпорожній» I «Погода» = «гарна» I «Завантаженість дороги» = «ранок», ТО «Актуальність забору» = «середня».

Правило 3: ЯКЩО «Наповненість контейнера» = «повний» I «Погода» = «нестійка» I «Завантаженість дороги» = «вечірній пік», ТО «Актуальність забору» = «середня».

Правило 4: ЯКЩО «Наповненість контейнера» = «повний» I «Погода» = «гарна» I «Завантаженість дороги» = «день», ТО «Актуальність забору» = «висока»

Рисунок 2.5 – Приклади нечітких продукційних правил бази знань

Таблиця 2.3 – Приклад фрагменту матричної нечіткої бази знань.

№ правила	Наповненість контейнера	Погода	Завантаженість дороги	Актуальність забору
1	Напівповний	Погана	Ранній пік	Низька
2	Напівпорожній	Гарна	Ранок	Середня
3	Повний	Нестійка	Вечірній пік	Середня
4	Повний	Гарна	День	Висока

Розрахуємо з використання алгоритму нечіткого виведення Мамдані ступень актуальності включення до маршруту контейнера при значеннях вхідних змінних, вказаних в таблиці 2.4.

Таблиця 2.4 – Набір тестових вхідних даних

Наповненість контейнера	Погода	Завантаженість Дороги
70	50	6

На рисунках 2.2– 2.4 вертикальними пунктирними лініями проілюстровано процес фазифікації, а в таблиці 2.5 наведене відображення її результатів у матричній нечіткій базі знань.

Таблиця 2.5 – Приклад перевірки бази знань

№ правила	Наповненість контейнера ($\mu_{НК}$)	Погода ($\mu_{П}$)	Завантаженість дороги ($\mu_{ЗД}$)
1	0,25	0	0
2	0	0,95	0,15
3	0,9	0,8	0,6
4	0,9	0,95	0,85

Результати застосування правил з обробки результатів фазифікації становлять:

$$\text{Правило 1: } \mu_{НК}^{\text{напівповний}} \& \mu_{П}^{\text{погана}} \& \mu_{ЗД}^{\text{ранній пік}} = 0,2 \& 0 \& 0 = \mu_{ТЗ}^{\text{низька}} = 0$$

$$\text{Правило 2: } \mu_{НК}^{\text{напівпорожній}} \& \mu_{П}^{\text{гарна}} \& \mu_{ЗД}^{\text{ранок}} = 0 \& 0,95 \& 0,15 = \mu_{ТЗ}^{\text{низька}} = 0$$

$$\text{Правило 3: } \mu_{НК}^{\text{повний}} \& \mu_{П}^{\text{нестійка}} \& \mu_{ЗД}^{\text{вечірній пік}} = 0,9 \& 0,8 \& 0,6 = \mu_{ТЗ}^{\text{середня}} = 0,6$$

$$\text{Правило 4: } \mu_{НК}^{\text{пустий}} \& \mu_{П}^{\text{гарна}} \& \mu_{ЗД}^{\text{день}} = 0,9 \& 0,8 \& 0,6 = \mu_{ТЗ}^{\text{висока}} = 0,85$$

Графічне подання дефазифікації методом центру тяжіння наведено на рисунку 2.6.

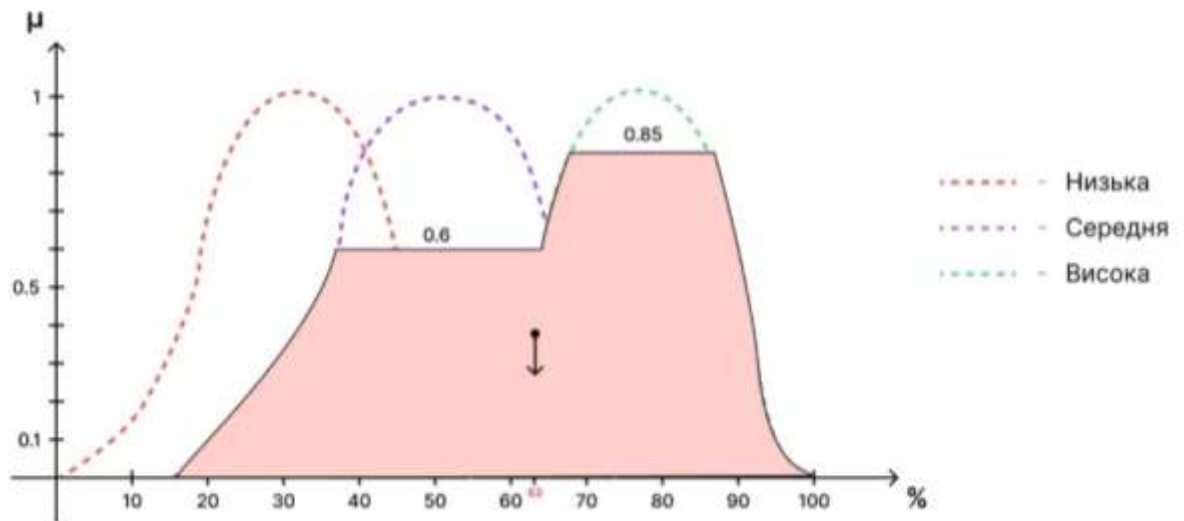


Рисунок 2.6 – Дефазифікація методом центру тяжіння

Актуальність забору сміття з контейнеру обчислюється за формулою:

$$EA = \frac{\sum_{i=1}^n \mu(EA_i) \times EA_i}{\sum_{i=1}^n \mu(EA_i)} \quad (2.13)$$

В результаті обчислень отримаємо:

$$\frac{20 * 0.1 + 25 * 0.2 + 30 * 0.33 + 35 * 0.52 + 40 * 0.6 + 45 * 0.6 + 50 * 0.6 +}{0,1 + 0,2 + 0,33 + 0,52 +}$$

$$\frac{55 * 0.6 + 60 * + 65 * 0.6 + 70 * 0.85 + 75 * 0.85 + 80 * 0.85 + 85 * 0.85 +}{+0,6 * 7 + 0,85 * 4 +}$$

$$\frac{+90 * 0.6 + 95 * 0.15 + 100 * 0}{+0,15} = 63.208$$

Отриманий результат $x = 63.208$ свідчить про актуальність включення даного контейнеру до чергового маршруту сміттєвоза.

2.4 Нечітка кластеризація контейнерів

Вінницька область стала першою, хто вирішив запровадити підхід створення кластерів, тобто зон оптимального охоплення щодо управління побутовими відходами [3] для оптимізації процесу збору сміття.

Це представлено у регіональному плані поводження з відходами на період до 2030 року [13], які виділяють такі критерії визначення кластерів:

- рекомендоване охоплення населення 150 – 400 тис. мешканців;
- наявний потенціал створення полігону в кластері;
- мінімізація витрат на транспортування та захоронення відходів;
- мінімізація обсягів захоронення відходів.

З врахуванням поділу області на райони та громади, автори регіонального плану пропонують створення 6 кластерів в межах районів: Калинівський, Жмеринський, Іллінецький, М.Куриловецький, Тульчинський, Бершадський. У таблиці 2.6 наведено дані кластери з врахуванням населення та кількості утворених відходів. На рисунку 2.7 наведено альтернативні варіанти кластеризації області.

Таблиця 2.6 – Кластери Вінницької області

Кластер	Населення (осіб)	Кількість відходів (тонн/рік)
Калинівський	185145	185145
Жмеринський	649130	194739
Іллінецький	229385	68816
Мурованокуриловецький	217698	65309
Тульчинський	203737	203737
Бершадський	140518	42155
Усього по області	1625613	487684

Здійснимо нечітку кластеризацію сміттевих контейнерів «адміністративного» кластера на окремі маршрути смітевозів. З урахуванням вантажомісткості смітевозів розрахуємо кількість нечітких кластерів (маршрутів) k , де для кожного контейнера визначимо його ступінь належності до

кожного з кластерів.

При цьому необхідно враховувати наступні обмеження:

- загальна кількість контейнерів всередині кластера не повинна перевищувати допустиму місткість кузова сміттєвоза;
- контейнери, що водночас відносяться до різних кластерів, можуть використовуватися для динамічного коригування маршрута сміттєвоза в режимі реального часу в разі виявлення в процесі збору сміття переоцінки завантаженості контейнерів його маршруту при попередньому прогнозуванні на основі неповних даних.

Головними параметрами при використанні нечіткої кластеризації є фіксована кількість кластерів, центроїди та головний кластер, який є найближчим центром ваги відповідного кластеру. Для отримання центру ваги необхідно мінімізувати цільову функцію, яка є сумою відстаней від кожної точки до кожного кластера:

$$Jm = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m \|x_i - v_j\|^2 \quad (2.14)$$

де m – ваговий коефіцієнт, N – кількість контейнерів, c – кількість центроїдів, v_j – центр j -го кластеру, який співпадає з розміром вибірки, x_i – i -та вибірка, u_{ij} – приналежність x_i до центру кластера c_j .

$$u_{ik}^{(l+1)} = \left(\frac{1}{\sum_{k=1}^c \|x_k - v_i^{(l)}\| / \|x_k - v_i^{(l)}\|} \right)^{2/(m-1)} \quad (2.16)$$

3. Порівняємо значення $U^{(l)}$ та $U^{(l+1)}$, якщо (4) – припиняємо ітерацію, інакше $l=l+1$ та повертаємось в початок алгоритму

$$\|U^{(l)} - U^{(l+1)}\| \leq \varepsilon \quad (2.17)$$

4. Якщо значення функції $PFm(U, c)$ досягає мінімуму, процес кластеризації закінчується. Інакше $c = c+1$ та повертаємось в початок алгоритму. Тобто, обирається число, функція належності якого мінімальна при збільшенні c . Функція $PFm(U, c)$ обчислюється як:

$$PFm(U, c) = \frac{1}{n} \sum_{i=1}^c \sum_{j=1}^n |u_{ij} - (u_{ij})_{0.5}| \quad (2.18)$$

На рисунку 2.9 наведено «адміністративну» кластеризацію Вінницької громади, до якої входить 9 населених пунктів: Вінниця, Десна, Великі Крушлинці, Вінницькі Хутори, Гавришівка, Малі Крушлинці, Писарівка, Стадниця та Щітки.

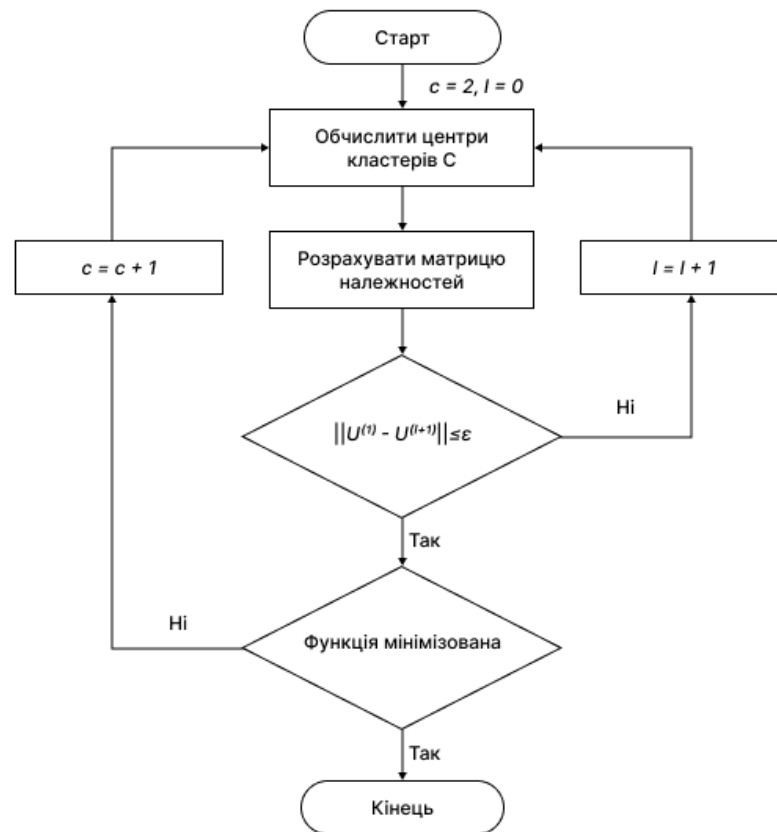


Рисунок 2.8 – Схема алгоритму нечіткої кластеризації контейнерів

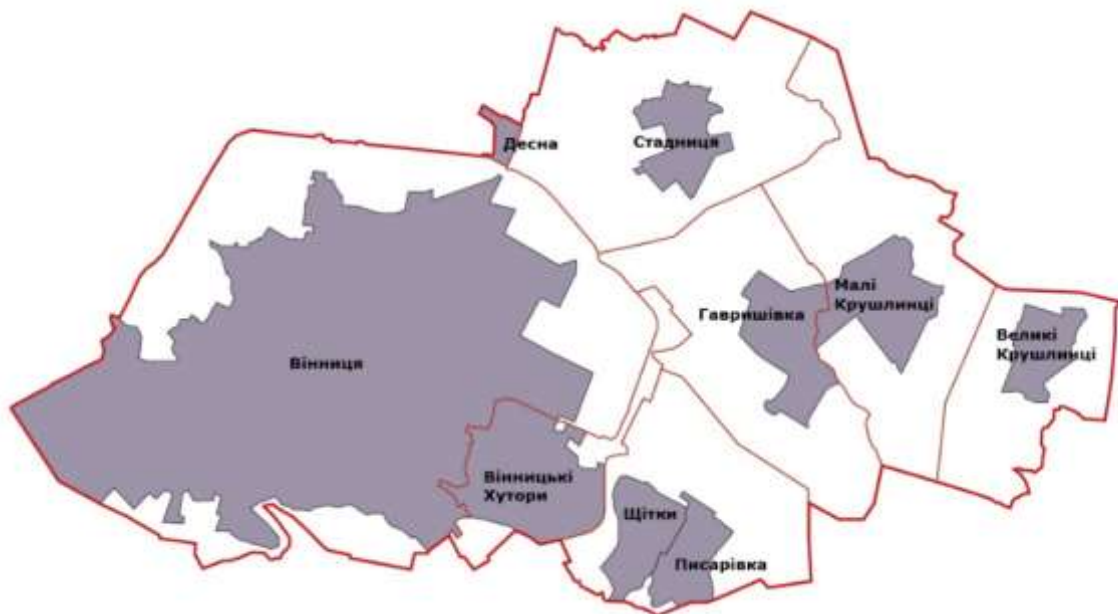


Рисунок 2.9 – Розбиття Вінницької громади на кластери

На рисунку 2.10 наведено приклад нечіткої кластеризації фрагменту вінницького «адміністративного» кластеру». Фрагмент матриці належності контейнерів нечітким кластерам показано у вигляді таблиці 2.7, цифри в якій показують значення функції належності. Контейнер належить тому кластеру, функція належності до якого контейнера має найбільше значення. Сума значень функцій належності контейнера всім нечітким кластерам має дорівнювати одиниці.

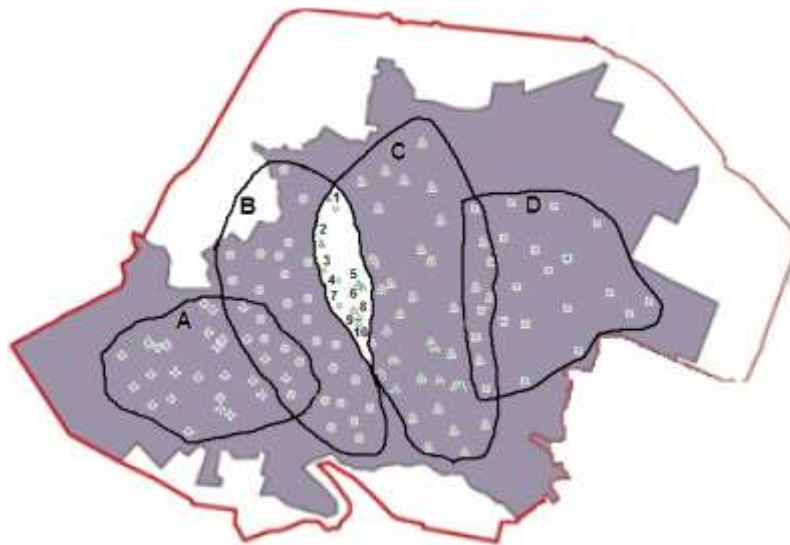


Рисунок 2.10 – Нечітка кластеризація з виділенням 4 кластерів

Таблиця 2.7 – Ступень належності контейнера до певного кластера

Номер контейнера	Кластер А	Кластер В	Кластер С	Кластер D
1	0,22	0,33	0,37	0,18
2	0,16	0,39	0,35	0,10
3	0,12	0,42	0,32	0,08
4	0,16	0,41	0,34	0,09
5	0,16	0,37	0,34	0,13
6	0,18	0,35	0,34	0,14
7	0,13	0,44	0,33	0,10
8	0,18	0,35	0,34	0,14

2.5 Оптимізація маршруту збору відходів у територіальних громадах на основі нечіткого генетичного алгоритму

Результати попередніх досліджень [3] свідчать, що для оптимізації маршруту, яка зводиться до вирішення задачі комівояжера, доцільним є використання генетичного алгоритму. При цьому якість оптимізації залежить від правильного підбору значень параметрів генетичного алгоритму, зокрема, від ймовірності використання генетичних операторів схрещування та мутації.

Отже, для вирішення задачі оптимізації маршруту збору відходів з використанням генетичного алгоритму необхідно виконати такі кроки:

1. Визначити форму подання хромосом та згенерувати початкову популяцію.
2. Виконати відбір батьківських хромосом для застосування до них генетичних операторів.
3. Сформувати базу правил для нечіткого логічного контролера визначення оптимальних значень ймовірності генетичних операторів.
4. Здійснити нечітке логічне виведення та дефазифікацію його результату для отримання керуючого впливу контролера на величину ймовірності застосування генетичних операторів.
5. Застосувати отриманий керуючий вплив до генетичних операторів.
6. Застосувати оператори схрещування та мутації до батьківських хромосом.

У задачі збору відходів хромосомами являються сміттєві контейнери, які необхідно об'їхати сміттєвозу в межах свого кластеру (маршруту). Так як наповненість контейнерів щоразу змінюється, це дає можливість формувати різні маршрути. Представимо рішення як рядок, що матиме $M+N$ -бітів, де M – хромосома, яка подає кількість точок, перестановка яких визначає маршрут сміттєвозу; N – номер кластеру, адже один автомобіль може бути задіяний у кількох ділянках громади, якщо йому вистачає місця в кузові.

Щоб не допустити повторне розміщення контейнерів в хромосомі, введемо перестановку чисел від 1 до M із зазначенням порядку для перших M -бітів, де

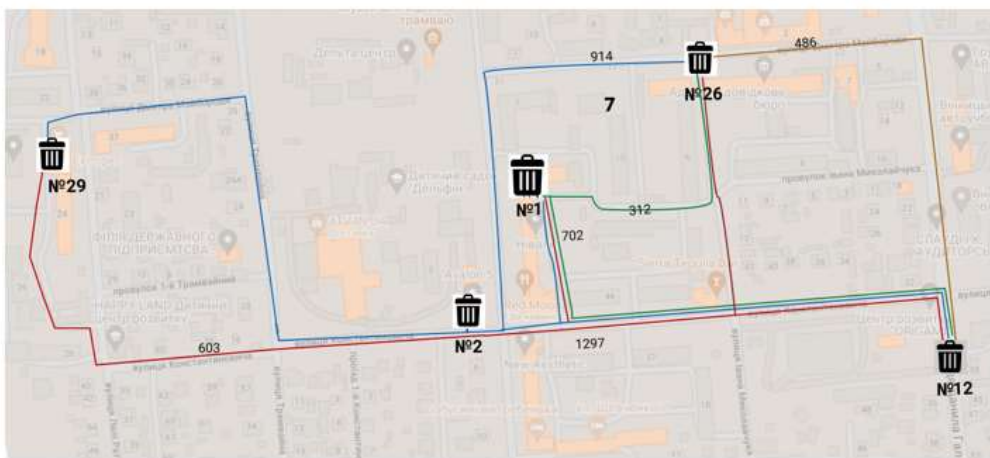
жодне число не може повторюватись. На рисунку 2.5 зображено форму подання рішення задачі з 20 біт, де справа номерами від 1 до 5 представлено п'ять кластерів Вінницького кластеру, обраного вище, а зліва – номери контейнерних майданчиків, які необхідно об'їхати.

1	16	7	23	11	32	27	4	22	17	30	19	8	28	3	1	2	3	4	5
---	----	---	----	----	----	----	---	----	----	----	----	---	----	---	---	---	---	---	---

Рисунок 2.11 – Форма подання призначення маршрутів вантажівкам

Функція пристосованості хромосом розраховується як сума відстаней між конкретними контейнерами.

На етапі кластеризації визначено наповнені контейнери, які включаються в маршрут та виконують роль хромосом. Приклад фрагменту їх розташування та матрицю відстаней між ними [30] на рисунку 2.12



	29	2	1	26	12
29		603	702	914	1297
2	603		161	505	451
1	702	161		312	445
26	914	505	312		486
12	1297	451	445	486	

Рисунок 2.12 – Варіанти маршрутів вантажівки

Приклад обрахунку функцій пристосованості окремих хромосом надасть такі результати

$$\sum_{i=1}^4 f_{29-2-1-26-12}(x) = 1/(603 + 161 + 312 + 486) = 1/1562 = 0,0006402;$$

$$\sum_{i=1}^4 f_{2-26-29-12-1}(x) = 1/(505 + 914 + 1297 + 445) = 1/3161 = 0,0003163;$$

$$\sum_{i=1}^4 f_{1-12-2-29-26}(x) = 1/(445 + 451 + 603 + 914) = 1/2413 = 0,0004144.$$

Після обрахунків можна зробити висновок, що перша хромосома з маршрутом 29-2-1-26-12 є найбільш пристосованою. Для відбору батьківських хромосом використаємо метод колеса рулетки, де кожній хромосомі відповідає частина сектору рулетки, величина якого є пропорційною значенню функції пристасованості хромосоми. Після обертання колесо випадково обирає хромосому для схрещування. Одержану рулетку представлено на рисунку 2.13, а в таблиці 2.8 наведено імовірність вибору хромосоми в якості батьківської.

Зауважимо, що значний вплив на якість пошуку генетичного алгоритму мають такі процеси як: дослідження глобального простору пошуку, що впливає здатність до пошуку рішення і допомагає уникати локальних екстремумів, та використання локального пошуку для знаходження «вдалих» рішень, який забезпечує поступове покращення рішень. Не враховування співвідношення даних процесів збільшує час роботи алгоритму і погіршує результати оптимізації через застрягання генетичного алгоритму в локальних екстремумах [30].

За значної переваги глобального пошуку (мутації) переважатиме випадковий пошук, який негативно позначиться на якості знайдених рішень. За значної переваги локального пошуку (схрещування) всі особини популяції стають однаковими, а пристосованість не зростає досить довго, що називають передчасною збіжністю генетичного алгоритму.

Таблиця 2.8 – Відбір методом колеса рулетки

Номер контейнеру	Імовірність визначення хромосоми, як батьківської
1	0,064
2	0,031
3	0,041
4	0,026
5	0,157
6	0,482
7	0,199

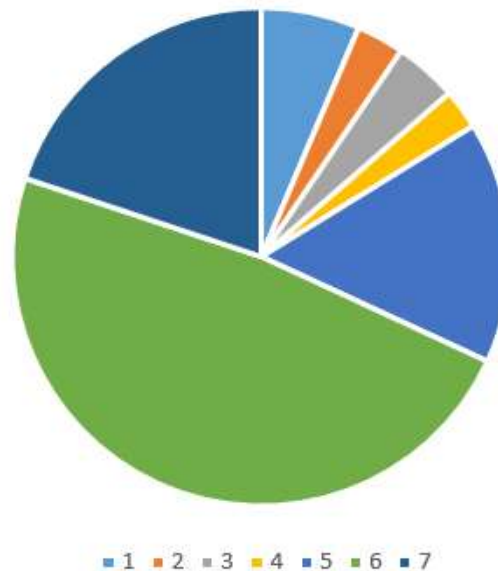


Рисунок 2.13 – Відбір методом колеса рулетки

У попередньому дослідженні [3] порівнювалось 2 методи схрещування: двоточковим кросовером та оператор схрещування з ремонтом хромосом-нащадків, другий з них показав більш оптимальні результати, тому в даному разі використовуємо його. Він передбачає обмін двох батьківських особин та ремонтування хромосом-нащадків. Якщо трапляється поява в хромосомі одного і того ж контейнеру, то здійснюється «ремонт» хромосоми, що показано на рисунку 2.14.

Здійснимо операцію мутації для урізноманітнення хромосом та уникненню затримки алгоритму в локальних мінімумах, а також передчасної зупинки роботи алгоритму. У попередній роботі порівнювалась мутації випадковою зміною позицій та зміною двох точок, остання серед яких показала кращі результати, тому для оптимізації маршруту оберемо її на рисунку 2.15.

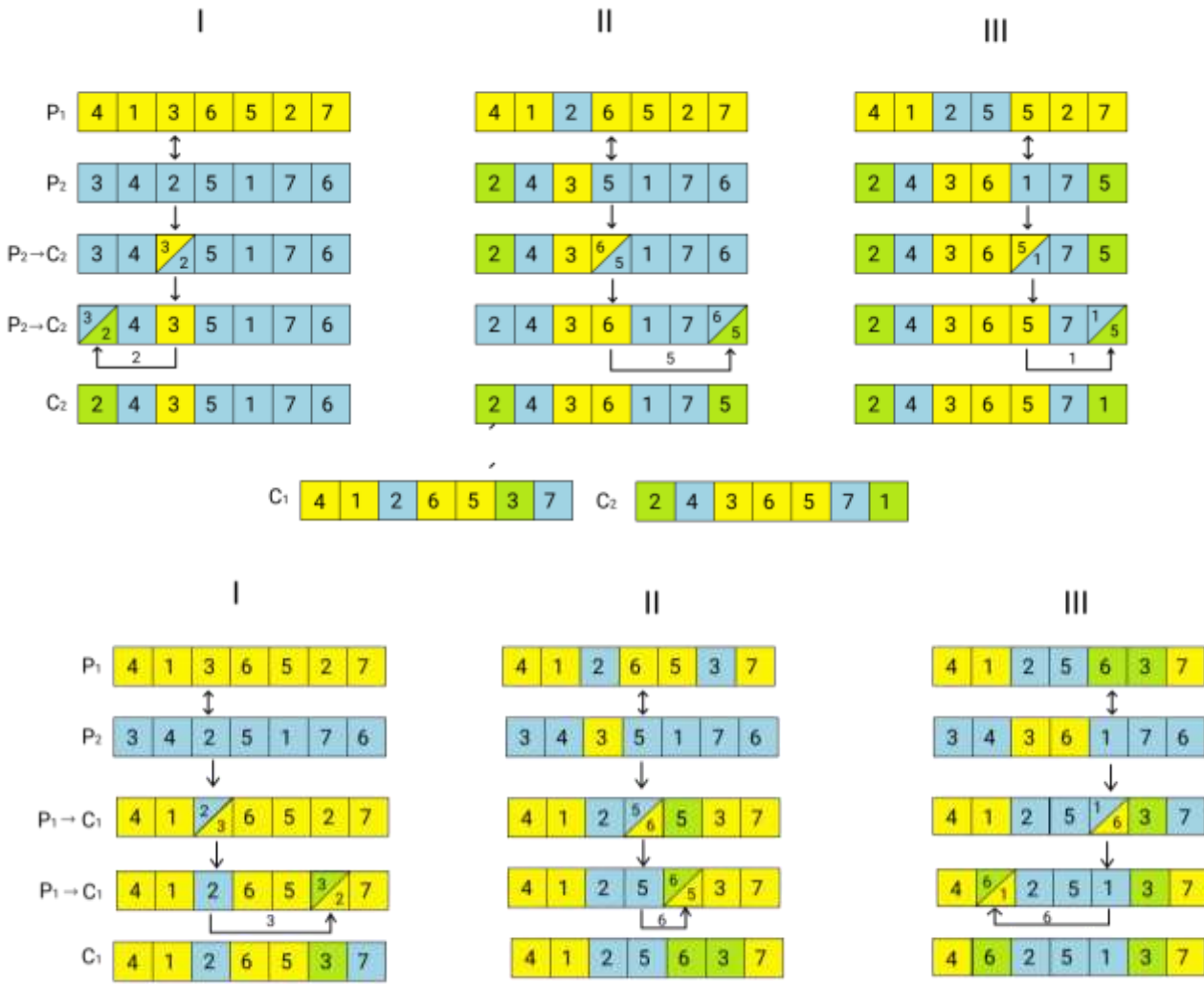


Рисунок 2.14 – Операція схрещування з ремонтом хромосом-нащадків



Рисунок 2.15 – Мутація зміною двох точок

Здійснення налаштування параметрів генетичного алгоритму відбувається з метою досягнення рівноваги між простором пошуку і використанням знайдених рішень, що є умовою отримання оптимальних результатів.

2.6 Розробка нечіткого контролера для автоматичної оптимізації значень параметрів нечітких генетичних операторів

Для автоматичної оптимізації значень параметрів нечітких генетичних операторів схрещування та мутації, здійснимо розробку нечіткого контролера, що дає можливість динамічно регулювати обрані параметри в процесі вирішення задачі збору відходів.

Нечіткий контролер [32] реалізує набір правил, що задають конкретні ситуації управління. У багатьох випадках достатньо знайти співвідношення між помилкою і швидкістю зміни помилки в процесі, щоб повпливати на величину, що забезпечує керування контролером. Для цього формулюються прості лінгвістичні правила.

Якщо помилка є позитивною і великою, а швидкість зміни помилки – негативна і мала, то зміна на вході процесу – позитивна і велика.

У правилі використовуємо три змінні: помилка (E), швидкість зміни помилки (C) і зміна керуючого впливу (U). Їх задають фіксованими універсальними множинами, що визначають діапазони вимірювань величин. Кожна з нечітких змінних може бути перетворена в нечіткі підмножини. Наприклад, нечітке правило може бути записано у вигляді оператора будь-якої форми:

Якщо E належить E_k , C належить C_k , то U належить U_k

Нечіткі підмножини задано набором упорядкованих пар виду:

$$\begin{aligned} E_k &= \{(e, m_{e_k}(e))\} \in E, \\ C_k &= \{(c, m_{c_k}(c))\} \in C \\ U_k &= \{(c, m_{u_k}(u))\} \in C \end{aligned} \quad (2.19)$$

де e, c, u — елементи дискретних універсальних множин, а $m_{e_k}, m_{c_k}, m_{u_k}$ — ступінь належності певного елемента до нечіткої підмножини

В результаті нечітка множина активується зі степенню, яка вираховується за формулою:

$$w_k = m_{e_k}(e) \wedge m_{u_k}(c) \quad (2.20)$$

Форма функції належності вихідної нечіткої множини залежить від кодування нечіткого правила. Широко розповсюджені два методи кодування: по мінімуму кореляції та добутку належності нечіткої множини. При мінімумі кореляції маємо:

$$m_{u_k}(y) = w_k \wedge m_{u_k}(y) \quad (2.21)$$

А при кодуванні произведенню кореляція множиться на величину:

$$m_{u_k}(y) = w_k m_{u_k}(y) \quad (2.22)$$

Для отримання функції належності вихідної нечіткої множини відбувається об'єднання всіх функцій:

$$m_0(y) = m_{01}(y) \vee m_{02}(y) \vee \dots \vee m_{0k}(y) \quad (2.23)$$

Вихідний ліквідатор нечіткості дорівнює нечіткому центру:

$$u = \frac{\int y m_0(y) dy}{\int m_0(y) dy} \quad (2.24)$$

де границі інтегралу відповідають всій універсальній множині.

На вхід контролера надходить комбінація показників ефективності генетичного алгоритму або поточних параметрів. На вихід подаються нові покращенні параметри управління алгоритмом. Даний процес показаний на рисунку 2.16.

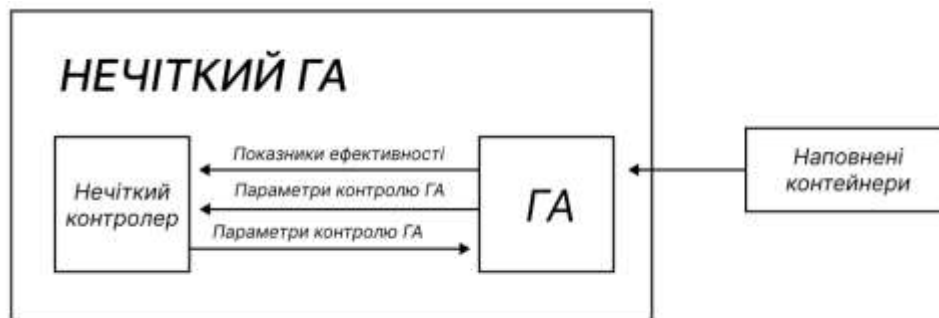


Рисунок 2.16 – Використання нечіткого контролера для автоматичної оптимізації значень параметрів нечітких генетичних операторів

Для покращення значень ймовірності генетичних операторів схрещування та мутації використаємо нечіткий логічний контролер, який включає в себе такі блоки:

- Базу правил
- Блок фазифікації, в результаті якого формується нечітка множина з відповідними функціями належності
- Блок дефазифікації, де відбувається перетворення функцій належності в конкретні рішення
- Виведення рішення

- Система контролю генетичних параметрів

Нечіткий логічний контролер оперує такими лінгвістичними змінними: NL (negative large) – значно негативний, NS (negative small) – трохи негативний, ZE (zero) – стабільний, PS(positive small) – трохи позитивний, PL (positive large) – значно поитивний. Формування керуючого впливу контролера здійснюється на основі оцінки двох параметрів:

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)} \quad (2.24)$$

$$e_2(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{best}(t)} \quad (2.25)$$

де t – часовий крок, $f_{max}(t)$ – краще значення цільової функції на ітерації t , $f_{ave}(t)$ – середнє значення цільової функції на ітерації, $f_{ave}(t-1)$ – середнє значення цільової функції на ітерації $(t-1)$. Параметри e_1 , e_2 задані на наступних інтервалах: $e_1 \in [0;1]$, $e_2 \in [-1;1]$.

Вхідний параметр $\Delta Pc(t)$ визначає ймовірність виконання схрещування, який заданий на інтервалі $\Delta Pc(t) \in [-0.1;0.1]$, для мутації візьмемо аналогічний параметр $\Delta Pm(t)$. При цьому ймовірність зміни цих параметрів сягає не більше 10%. Тоді використаємо чіткі значення, щоб змінити ймовірність Pc та Pm таким чином:

$$\begin{aligned} Pc(t) &= Pc(t-1) + \Delta Pc(t) \\ Pm(t) &= Pm(t-1) + \Delta Pm(t) \end{aligned} \quad (1.2)$$

Отже, нечіткий логічний контролер являє собою двовимірний набір нечітких чисел, які представлені у таблицях 2.4 та 2.5. Функції належності термів керючих впливів та вихідного сигналу наведені на рисунку 2.17.

Так, наприклад, якщо при вході контролера задані параметри $e_1 =$ трохи позитивний та $e_2 =$ значно негативний, то згідно таблиці отримаємо: $\Delta Pc(t) =$ трохи негативний, $\Delta Pm(t) =$ трохи позитивний. Тоді значення ймовірності кросинговера та мутації визначаються як:

$$Pc(t) = Pc(t - 1) + NS \quad (2.26)$$

$$Pm(t) = Pm(t - 1) + PS \quad (2.27)$$

При заданих параметрах e_1 та e_2 ймовірність кросинговера «незначно» погіршиться, а мутації – «значно» покращиться.

Таблиця 2.9 – Нечітке правило для оператору кросоверу ($\Delta Pc(t)$)

	e_2				
e_1	<i>NL</i>	<i>NS</i>	<i>ZE</i>	<i>PS</i>	<i>PL</i>
<i>PL</i>	трохи позитивний	стабільний	трохи позитивний	трохи позитивний	значно позитивний
<i>PS</i>	стабільний	стабільний	Стабільний	стабільний	стабільний
<i>ZE</i>	трохи негативний	значно негативний	Стабільний	стабільний	трохи негативний

Таблиця 2.10 – Нечітке правило для оператору мутації ($\Delta Pm(t)$)

	e_2				
e_1	<i>NL</i>	<i>NS</i>	<i>ZE</i>	<i>PS</i>	<i>PL</i>
<i>PL</i>	трохи позитивний	стабільний	трохи позитивний	трохи негативний	значно негативний
<i>PS</i>	стабільний	стабільний	трохи негативний	стабільний	трохи негативний
<i>ZE</i>	трохи позитивний	значно кращий	Стабільний	значно позитивний	трохи позитивний

На рисунку 2.17 показано основні блоки нечіткого генетичного алгоритму, який включає в себе вхідний та вихідний інтерфейс, базу знань та нечітку систему.

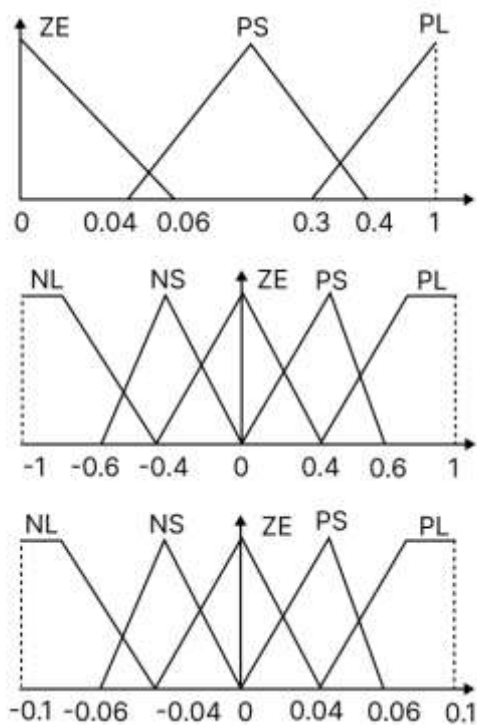


Рисунок 2.17 – Приклад використання правил нечіткого контролера

Структура нечіткого генетичного алгоритму оптимізації маршруту збору відходів у територіальних громадах зображено на рисунку 2.18, нижче наведено псевдкод алгоритму, а на рисунку 2.19 схему функціонування нечіткого генетичного алгоритму.



Рисунок 2.18 – Блоки нечіткого генетичного алгоритму

```

Початок алгоритму
t= 0 Iteration counter
Формування початкової популяції  $P(t)$ 
Поки (не виконана умова зупинки)
{
    t=t+1
    Відбір
    Оператор Кросоверу  $P(t)$ 
    Оператор Мутації  $P(t)$ 
    Розрахунок функції належності  $P(t)$ 
    Регулювання параметрів ГА
    {
        Нечіткий логічний контролер  $(e_1, e_2)$ 
        Оновлення параметрів
    }
}
Кінець алгоритму

```

Рисунок 2.19 – Псевдокод нечіткого генетичного алгоритму

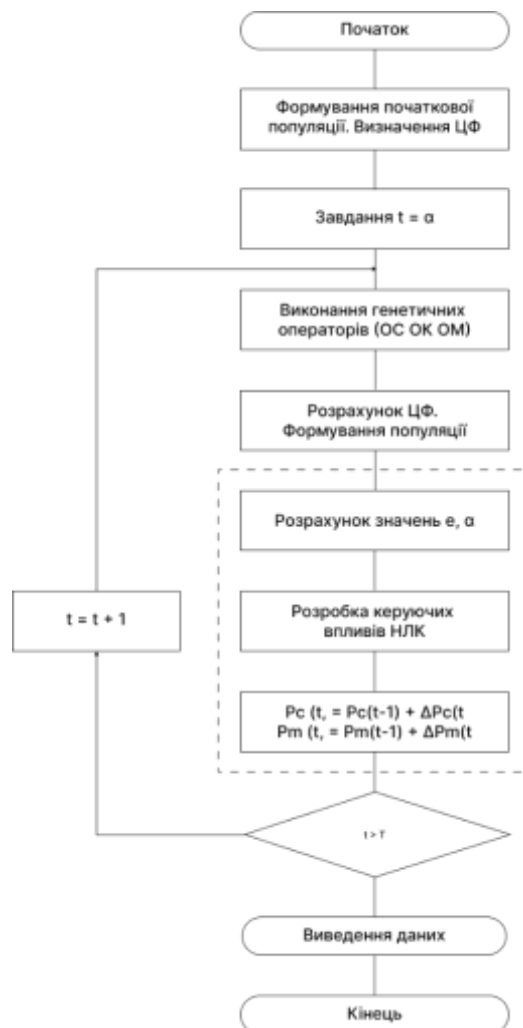


Рисунок 2.10 – Схема нечіткого генетичного алгоритму

2.7 Висновки до другого розділу

1. Розроблено структуру інформаційної технології оптимізації процесу збору відходів у територіальних громадах, яка відрізняється наявністю трьох відокремлених етапів, що забезпечує зменшення розмірності задачі та відповідне підвищення продуктивності технології.

2. Запропоновано загальну модель задачі на основі нечіткого графу, яка забезпечує можливість нечіткого обчислення ступеню заповненості сміттєвих контейнерів та їх доступності, що дозволяє оптимізувати маршрути сміттєвозів за рахунок видалення з них незаповнених та недоступних контейнерів.

3. Здійснено нечітку кластеризацію сміттєвих контейнерів на маршрути, що дозволяє динамічно коригувати маршрут сміттєвоза в режимі реального часу в разі виявлення в процесі збору сміття переоцінки завантаженості контейнерів при її попередньому прогнозуванні на основі неповних даних

4. Визначено вхідні та вихідні лінгвістичні змінні наповненості контейнеру, погодних умов та завантаженості доріг та функції належності їх терм множин, що забезпечило можливість створення нечіткої моделі обчислення ступеню наповненості та доступності сміттєвих контейнерів.

5. Обґрунтовано загальні вимоги щодо узгодження співвідношення локального та глобального пошуку генетичного алгоритму, що забезпечило основу для створення бази правил для нечіткого контролера автоматичної оптимізації значень параметрів нечітких генетичних операторів.

6. Розроблено нечіткий контролер автоматичної оптимізації значень параметрів нечітких генетичних операторів, що забезпечило покращення оптимізації маршрутів генетичним алгоритмом на 1,5-3,5%.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ ВІДХОДІВ У ТЕРИТОРІАЛЬНІЙ ГРОМАДІ

3.1 Обґрунтування вибору мови програмування

Для реалізації інформаційної технології оптимізації збору відходів необхідно створити веб-застосунок, який включає в себе фронтенд та бекенд частину. При написанні бекенду варто врахувати швидкість роботи нечіткого генетичного алгоритму. Написання одного і того ж алгоритму різними мовами програмування може змінювати довжину, обсяги та час роботи програми. Проаналізуємо популярні мови програмування в даному напрямку – JavaScript, Ruby, Python.

JavaScript – об'єктно-орієнтована скриптова мова програмування, використовується як вбудована мова для програмного доступу до об'єктів додатків. Основними архітектурними рисами є динамічна типізація, автоматичне керування пам'яттю, прототипне програмування та функції як об'єкти першого класу.

Ядро JavaScript включає ряд функцій, які дають наступні можливості: зберігання даних в змінні; активізація частин коду у відповідності з певними сценаріями; які здійснюються на сторінці сайту; створення контенту, який оновлюється автоматично; управління мультимедійними можливостями [33].

JavaScript обробляється в веб-додатках на стороні клієнта, веб-інтерфейси, які працюють на його основі, є кросплатформними. що дає можливість використання її на будь-якій операційній системі. Недоліками даної мови є проблема роботи із типами даних та обмежені можливості мови, які виправляються за рахунок підключення бібліотек та фреймворків.

Ruby – динамічна високорівнева мова об'єктно-орієнтовного програмування. Однією з основних особливостей мови є відсутність процесу компіляції, при цьому помилки в програмі можна визначити лише при запуску чи під час роботи програми. Характерною ознакою також є відсутність множинного наслідування.

Дана мова містить автоматичний прибиральник сміття, що працює для всіх об'єктів. Ruby не вимагає попереднього оголошення змінних та використовує прості домовленості для визначення області видимості. Як і JavaScript вона є кросплатформленою та може динамічно завантажувати розширення.

Як бекенд-технологія використовується відкритим вхідним лаконічним кодом. Перевагою даної мови є функція автоматизації тестування, що значно спрощує процес розробки веб-додатків. Недоліками є мала кількість бібліотек та високий час виконання процесів коду [34].

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Лаконізм мови дозволяє писати код, який легко прочитати та який набагато коротший, ніж в інших мовах. Однією з переваг мови є стандартна бібліотека, яка містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм [34].

Дана мова не є компільованою, тому швидкість виконання коду є досить низькою, проте досить зручна як мова сценарії, що для додаткових налаштувань функціональності. Python дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. Як і попередні мови, вона є кросплатформленою.

Популярність серед бекенд розробників мова набула через її простоту, легкість під час навчання та швидкість роботи коду. Крім того мова має власний веб-сервер WSGI, який відповідає на запити.

Для обрання мови, за допомогою якої можна краще відобразити роботу інформаційної технології оптимізації збору відходів, проведемо порівняльний аналіз. У таблиці 3.1 показано критерії порівняння швидкості виконання коду, а в таблиці 3.2 – використання пам'яті, що тестувалось на основі Intel Core2 Duo T7500@2.20Ghz CPU; 2 GB RAM; Linux 2.6.32 i686.

Таблиця 3.1 – Порівняння швидкості виконання коду мовами програмування

Розмір, Kb	Ruby	JavaScript	Python
256	0:00:07	0:00:03	0:00:17
512	0:00:29	0:00:21	0:01:21
768	0:01:15	0:00:51	0:03:21
1024	0:02:21	0:01:31	0:06:13
1280	0:03:45	0:02:24	0:09:58
1536	0:05:28	0:03:28	0:14:37
1792	0:07:32	0:04:43	0:20:11

Таблиця 3.2 – Порівняння використання пам'яті мовами програмування

Розмір, Kb	Ruby, Kb	JavaScript, Kb	Python, Kb
256	16,872	9,620	6,384
512	21,480	10,672	5,876
768	27,924	11,328	7,676
1024	29,681	12,704	6,388
1280	38,376	13,524	9,212
1536	44,520	14,748	6,900
1792	50,664	18,052	7,156
2048	51,176	22,912	11,516

В результаті отриманих даних обрано мову JavaScript для реалізації інформаційної технології збору відходів. Під час тестування вона показала найкращий результат швидкості виконання коду та близький до оптимального у порівнянні критерію використання пам'яті, що є суттєвим під час роботи нечіткого генетичного алгоритму.

3.2 Розробка структури програмного забезпечення інформаційної технології оптимізації процесу збору відходів у територіальних громадах

Розглянемо алгоритм розв'язку задачі збору відходів у територіальній громаді. Вхідними даними є масив відстані від одного контейнера до іншого; масив місткості кузова автомобіля. Вихідними даними є масив масивів з порядком проїзду кожної вантажівки, яку включено в масив. Загальна схема вирішення задачі збору відходів виглядає наступним чином та показана на рисунку 3.1:

Крок 1: З отриманого масиву відстані від одного контейнера до іншого генерується матриця поїздки.

Крок 2: Поки не зібрано сміття з кожної точки, в циклі повторюються такі дії:

Крок 2.1: Обирається вантажівка зі списку доступних

Крок 2.2: Знаходиться найближчий наповнений контейнер, який не об'їханий вантажівкою

Крок 2.3: Поки місткість кузова дозволяє зібрати відходи, виконуємо наступні кроки в циклі:

Крок 2.3.1: Найближчий контейнер, який не був відваний вантажівкою, додається в маршрут.

Крок 2.3.2: Від місткості вантажівки віднімається вага баку.

Крок 2.3.3: Знаходження найближчого контейнера

Крок 3: Вивід результату у вигляді масиву масивів з порядком проїзду кожного контейнера, який знаходиться в межах кластера.

Для забезпечення якісної роботи інформаційної технології оптимізації процесу збору відходів, її розділено на три режими роботи. Перший – режим диспетчера, який бачить кількість контейнерів, які необхідно об'їхати та мапу їх розташування. Він може скористатися режимами автоматичного розбиття контейнерів на кластери та формування маршрутів для кожного кластеру.



Рисунок 3.1 – Загальна схема алгоритму збору відходів

Також у диспетчера є можливість перезапустити програми автоматичного формування кластерів та маршрутів, якщо вони йому не сподобаються, внести ручні зміни в отримані автоматично кластери і/або маршрути. Він відстежує виконання задач збору відходів сміттєвозами в режимі реального часу та може дізнатись причини відхилення від маршруту: наприклад, запізнення,

незаплановані зупинки, пропущення точок об'їзду, і, таким чином запобігти зриву збору сміття у громаді [2].

У режимі водія доступна мапа сміттєвих контейнерів громади, де можна переглянути усі точки збору по громаді та перейти до тих, котрі незібрані. Кожному водію відправляється оптимізований маршрут в межах його кластеру та з врахуванням наповненості кузова автомобіля. Додаток забезпечує комунікацію між диспетчером та водієм у разі необхідності негайної передачі інформації.

Третій режим – містянин, який призначений для жителів громади. За допомогою нього вони можуть передавати дані про наповненість конкретної точки. Вони можуть повідомляти про стан контейнеру, коли викидають сміття або ж просто бачать наповнену чи переповнену ділянку. Таким чином дані у диспетчера будуть точнішими, що дозволить покращити процес оптимізації маршруту. На рисунку 3.2 наведено діаграму прецедентів даної інформаційної технології, а на рисунку 3.3 показано діаграму послідовностей.

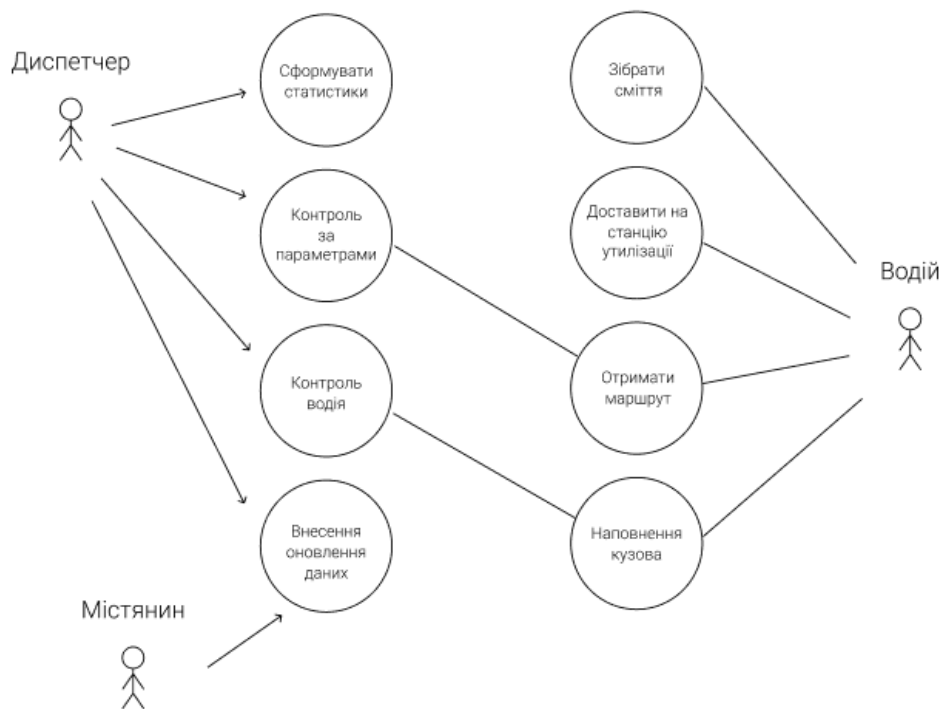


Рисунок 3.2 – Діаграма прецедентів інформаційної технології

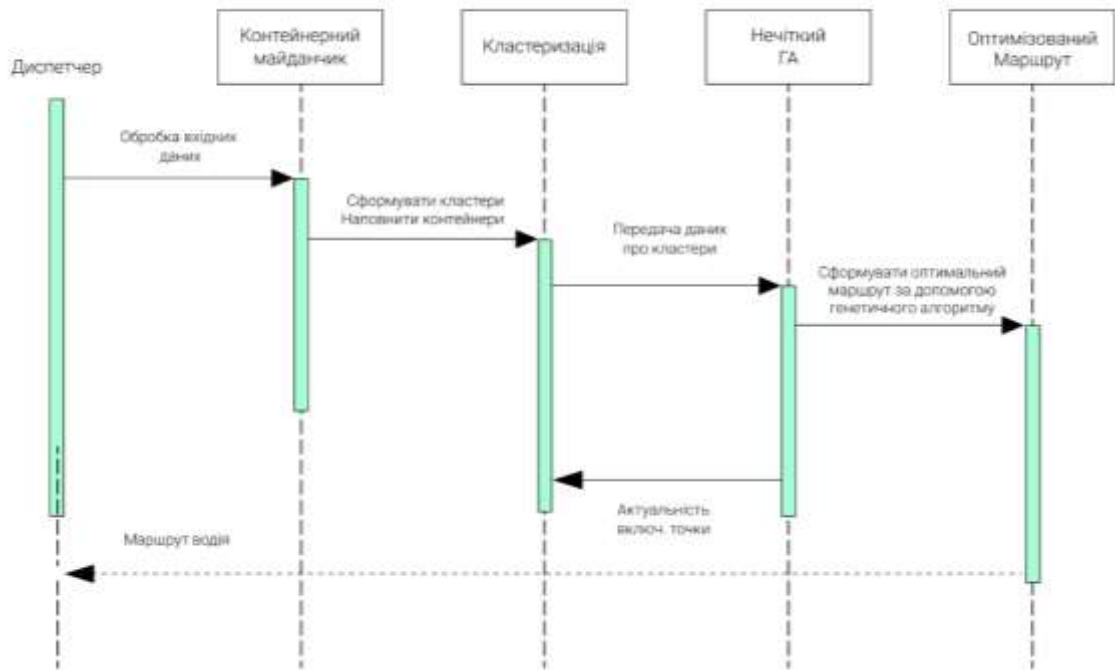


Рисунок 3.3 – Діаграма послідовностей інформаційної технології

Під час своєї роботи диспетчер формує звіти щодо статусу виконання процесу збору сміття у територіальній громаді. Вона включає номер наповненого контейнера, його адресу, номер вантажівки, що його обслуговує, місткість її кузова на момент збору та статус виконання задачі: виконаний – сміття зібране, в обробці – вантажівка прямує до точки, невідомий – водій не відповів на заявку диспетчера. Розташування контейнерних майданчиків взято з інтерактивної карти [29]. У таблиці 3.3 показано фрагмент статистики.

Таблиця 3.3 – фрагмент статистики збору сміття у м. Вінниця

Номер Контейнера	Адреса	Наповненість контейнера (%)	Вантажівка	Наповненість кузова (%)	Статус виконання
2	Трамвайна, 3	85	1	20	Виконаний
6	Стрілецька, 30А	73	9	95	Невідомий
13	Космонавтів, 41	92	7	74	В обробці
20	Пирогова, 9Б	68	3	13	В обробці
32	Павла Тичини, 15	96	5	82	Виконаний

Загальний сценарій роботи інформаційної технології складається з кількох етапів. Спочатку формується уся база контейнерів для об'їзду – як з датчиками так і без них та фіксується у центральну базу даних. Після чого відбувається обробка вхідних даних, тобто встановлення координат кожного майданчика та попереднє визначення наповненості.

Потім відбувається процес кластеризації, під час якого формуються кластери відповідно до рівня наповненості контейнера та відкидаються пусті або напівпусті точки, які точно не включаються у маршрут. Це дає можливість заощадити час водія для поїздки до більш завантажених точок у громаді.

Наступним кроком є оптимізація маршруту за допомогою нечіткого генетичного алгоритму, який постійно динамічно змінюється через постійне оновлення даних щодо наповненості контейнерів. У випадку, коли один з водіїв не може виконати своє завдання через різні причини, диспетчер має можливість направити іншого водія на дану позицію для забезпечення своєчасного вивезення сміття.

Після отримання свого маршруту, кожен водій прямує до своїх точок. Формування маршрутів здійснюється з розрахунком місткості кузова вантажівки. Коли автомобіль завантажений, він прямує до найближчої станції утилізації, а потім повертається до збору сміття. Вкінці робочої зміни водій поїде своєї до початкової точки виїзду. Схему роботи інформаційної технології оптимізації процесу збору відходів у територіальній громаді зображено на рисунку 3.4.

Структурну схему програмного забезпечення подано на рисунку 3.5, де окремими блоками виділено інтерфейс диспетчера, водія та місцянина. Диспетчер бачить стан усіх вантажівок та контейнерних майданчиків. Водію доступно лише своя конкретна ділянка в межах кластеру.

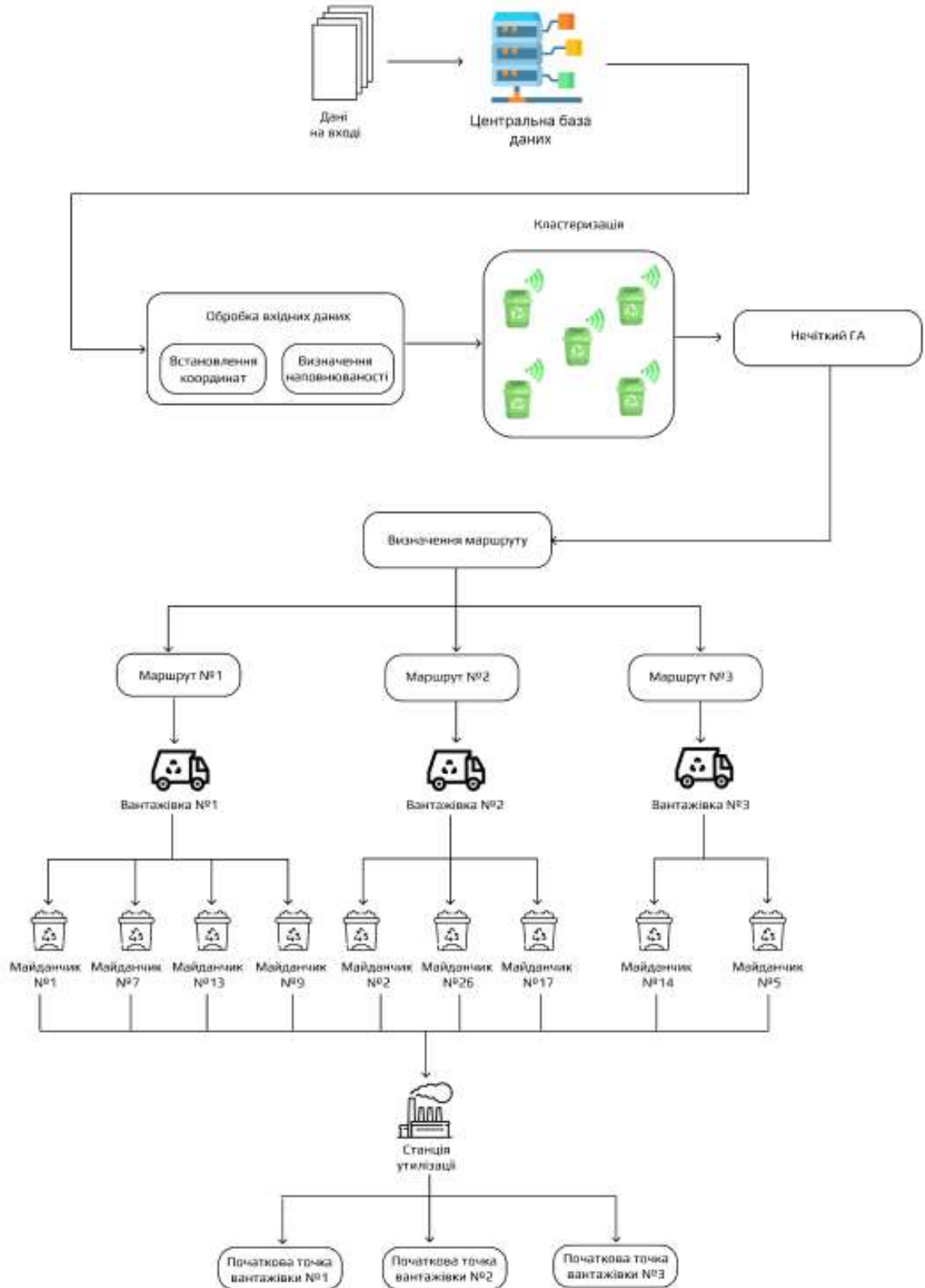


Рисунок 3.4 – Схема роботи інформаційної технології

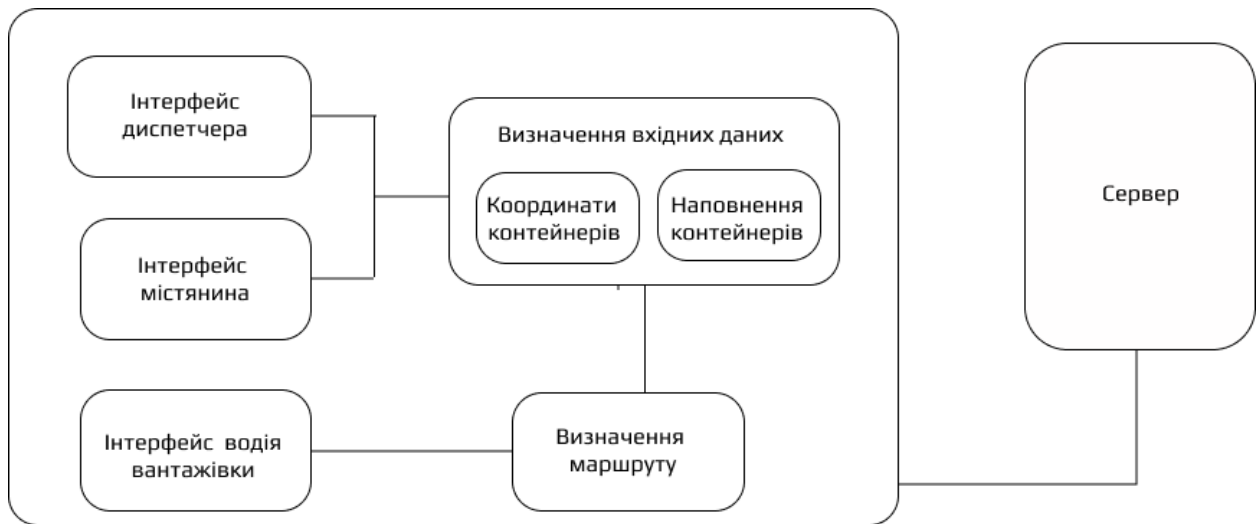


Рисунок 3.5 – Структура програмного забезпечення інформаційної технології оптимізації процесу збору відходів у територіальній громаді

Містянину доступна інтерактивна карта контейнерів, на котрій він обирає точку, про яку хоче передати інформацію, яка фіксується сервером та передається диспетчеру. Жителі міста можуть це зробити за допомогою шкали, на якій показано відсоток наповненості контейнеру та кнопка, яка сигналізує про критичний стан на одній з ділянок на випадок, коли контейнер переповнений та сміття лежить біля нього. Тоді водію поступає інформація, що туди необхідно поїхати позачергово, а ділянка у диспетчера виглядає як критична.

3.3 Розробка програмного забезпечення підсистеми генетичного алгоритму оптимізації процесу збору відходів

Для отримання оптимального маршруту обов'язковою умовою є виконання послідовних покрокових операцій генетичного алгоритму. З них можна виділити 4 основні модулів:

- 1) Клас GaChromosome – інтерфейс реалізації хромосом, який визначає методи Crossover, Mutation, CalculateFitness та CompareFitness, які представляють раніше описані генетичні операції. Тобто сюди приходять

кількість контейнерів, формується початкова популяція, визначається цільова функція, виконуються генетичні операції

```
GChromosome(actual_ratings,
predicted_ratings):
    """Get genetic feature weights using crossover
    and mutation."""

    # Generate initial binary population
    cur_gen_pop = np.random.randint(2,
    size=(INITIAL_POPULATION, FEATURE_LENGTH))
    next_gen_pop = np.zeros((cur_gen_pop.shape[0],
    cur_gen_pop.shape[1]))
    fitness_vector = np.zeros((INITIAL_POPULATION,
    2))
```

2) Клас GaChromosomeParams визначає ймовірність кросинговеру, розмір мутації та кількість точок кросовера

```
GChromosomeparams
=
np.multiply(next_g
en_pop, np.matrix(
    [np.float(np.random.normal(0, 2, 1)) if
np.random.random() < MUTATION_RATE else 1 for x in
range(next_gen_pop.size)]).reshape(next_gen_pop.shape
))
    cur_gen_pop = next_gen_pop

    best_soln =
np.array(cur_gen_pop[np.argmin(fitness_vector[:,
1])])
    return best_soln
```

3) Клас FuzzyParameters визначає метод Clone, який оптимізує генетичні оператори

```
FuzzyParameters(self,
age):
    """Get fuzzy set values of given
age."""
    return [self.young(age),
self.middle(age),
```

```

self.old(age)]

return
[self.very_bad(gim_value),
                                     self.bad(gim_value),
                                     self.average(gim_value),
                                     self.good(gim_value),
                                     self.very_good(gim_value),
                                     self.excellent(gim_value)]

```

- 4) Класи `GaDefaultChromosome` та `GaChromosomeParamsBlock` беруть участь у процесі відбору та визначають ту хромосому, яку буде включено в маршрут

```

winners =
np.zeros((WINNER_PER_GEN
, FEATURE_LENGTH))

for n in range(len(winners)):
    selected =
np.random.choice(range(len(fitness_vector)),
int(WINNER_PER_GEN/2),

replace=False) # select random indexes from pop

    wnr =
np.argmin(fitness_vector[selected, 1]) # select
one index with min fitness error (tournament)
    winners[n] =
cur_gen_pop[int(fitness_vector[selected[wnr]][0])
] # add to winner population

next_gen_pop[:len(winners)] =
winners

```

Він також використовується для покращення мутації, викликаючи метод `PrepareForMutation`, який виконує резервну копію старої хромосоми, яка після цього порівнюється з новоутвореною. Якщо мутація покращила пристосовність, то викликається метод `AcceptMutation`, інакше – `RejectMutation`.

Кросинговер, мутація та пристосування реалізується шляхом успадкування класу, а Потім користувач може перевизначити методи PerformCrossover, PerformMutation і CalculateFitness і реалізувати певні операції для цільової проблеми.

Алгоритм припиняє свою роботи за умови знаходження оптимального або близького до нього маршруту, або коли досягнута максимальна кількість ітерацій. Встановимо умову, що популяція стає стабільною, тобто значення цільової функції становить менше 0.1%. На рисунку 3.6 зображено діаграму класів, а на рисунку 3.7 схему нечіткого генетичного алгоритму.

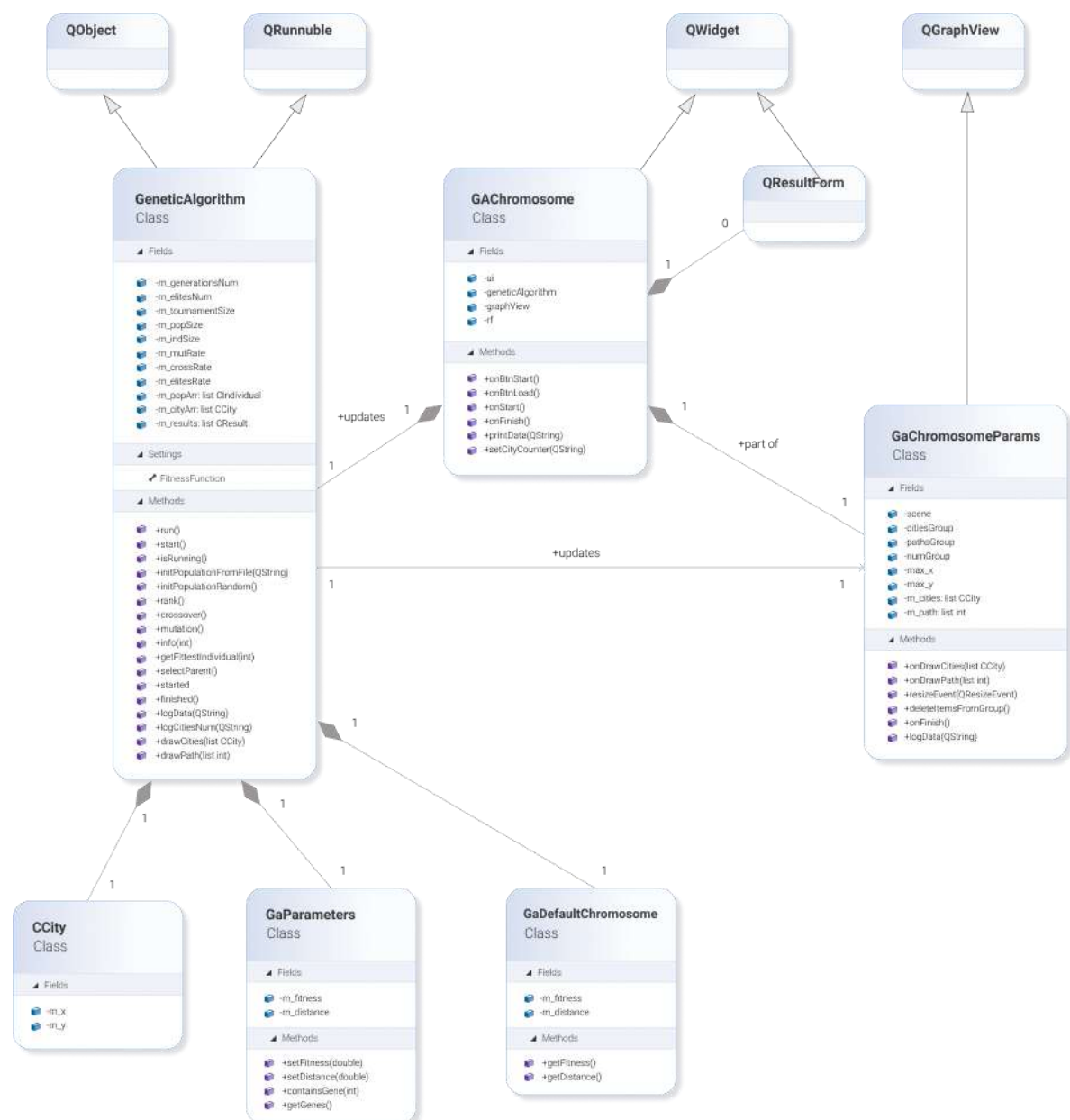


Рисунок 3.6 – Діаграма класів інформаційної технології оптимізації процесу збору відходів

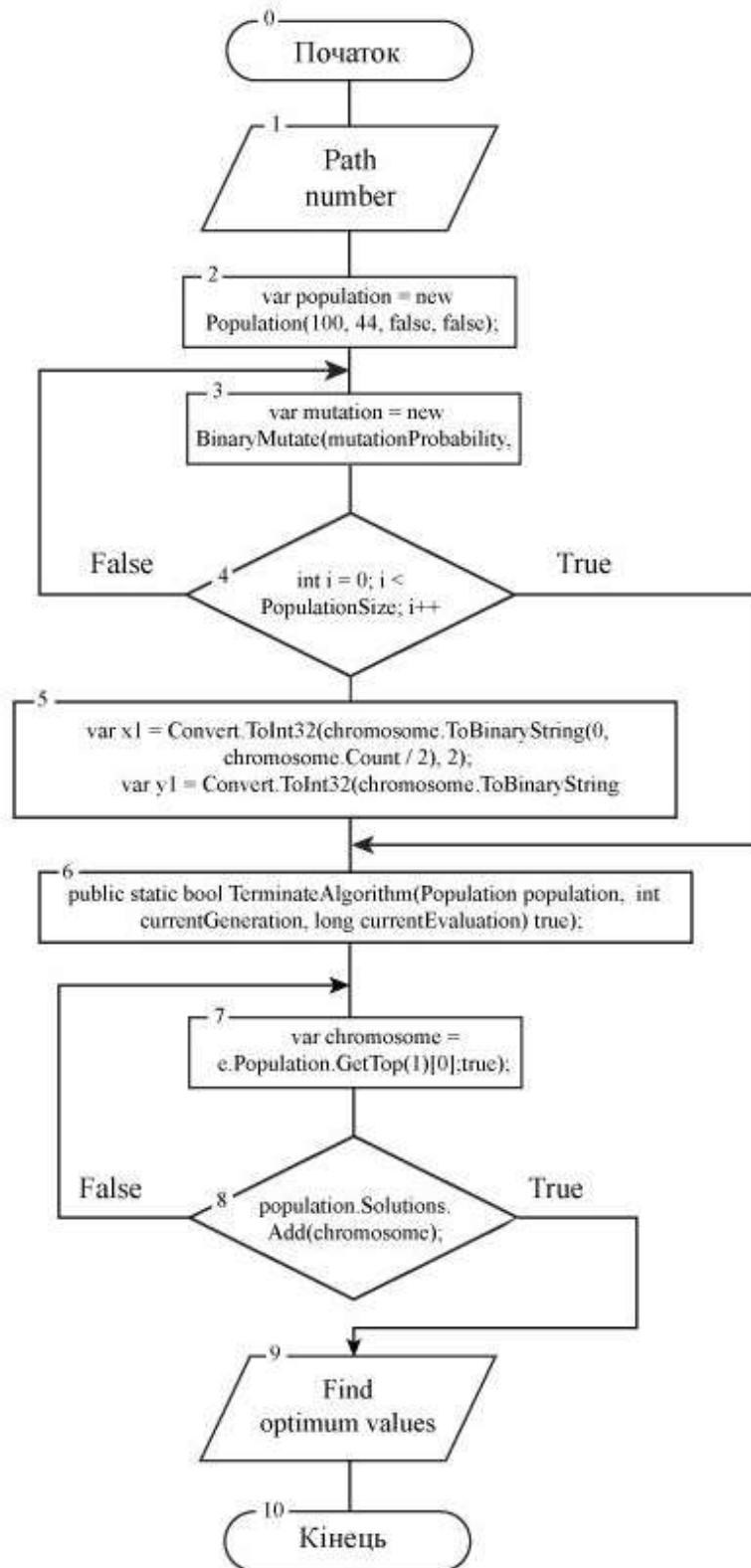


Рисунок 3.7 – Схема роботи нечіткого генетичного алгоритму

3.3 Аналіз результатів тестування програми

Тестування інформаційної технології оптимізації процесу збору відходів здійснювалось з врахуванням даних про графіки вивезення сміття та складені маршрути Вінницької територіальної громади [стаття]. Враховувалась інформація про погодні умови [*], та завантаженість дороги [*] в період з травня по червень громади.

Протестуємо програму та порівнюємо її з результатами попередніх досліджень [*]. Для оцінювання ефективності здійснено 100 запусків генетичного алгоритму з врахуванням параметрів ймовірність схрещування – 0,65- 0,87, ймовірність мутації – 0.12, критерій зупинки – відсутність покращення значення цільової функції протягом 15 поколінь, розмір популяції – 80 хромосом. У таблиці 3.4 показано результати тестування, які показують, що використання нечіткого контролера дало змогу покращити маршрут на 1.71%.

Таблиця 3.4 – Тестування інформаційної технології

Номер маршруту	1	2	3	4	5	6	7
Загальна кількість контейнерів, км	53	42	68	61	34	27	49
Довжина маршруту, класичний ГА, км	37,07	29,18	47,89	44,75	25,48	19,74	33,15
Кількість наповнених контейнерів, км	45	31	54	45	25	21	40
Довжина маршруту, класичний ГА, км	35,14	27,22	44,01	41,43	23,90	18,71	30,86
Відсоток покращення, км	5,2%	6,7%	8,1%	7,4%	6,2%	5,4%	6,9%
Довжина маршруту, ГА з контролером, км	34,73	27,02	43,19	41,12	23,31	18,37	30,33
Відсоток покращення	6,3%	7,4%	9,8%	8,1%	8,5%	6,9%	8,5%

На рисунку 3.8 зображено початкову сторінку інформаційної технології, яка виконана як веб-додаток. Вона дає можливість авторизуватись у різні режими, дізнатись про точки збору відходів у громаді, контактну інформацію про пункти утилізації, а також має інтерактивну карту контейнерних майданчиків. На рисунку 3.9 показано приклад роботи інтерфейсу диспетчера, який відкрив один з кластерів громади та моніторить ситуацію вивозу сміття на даній ділянці.

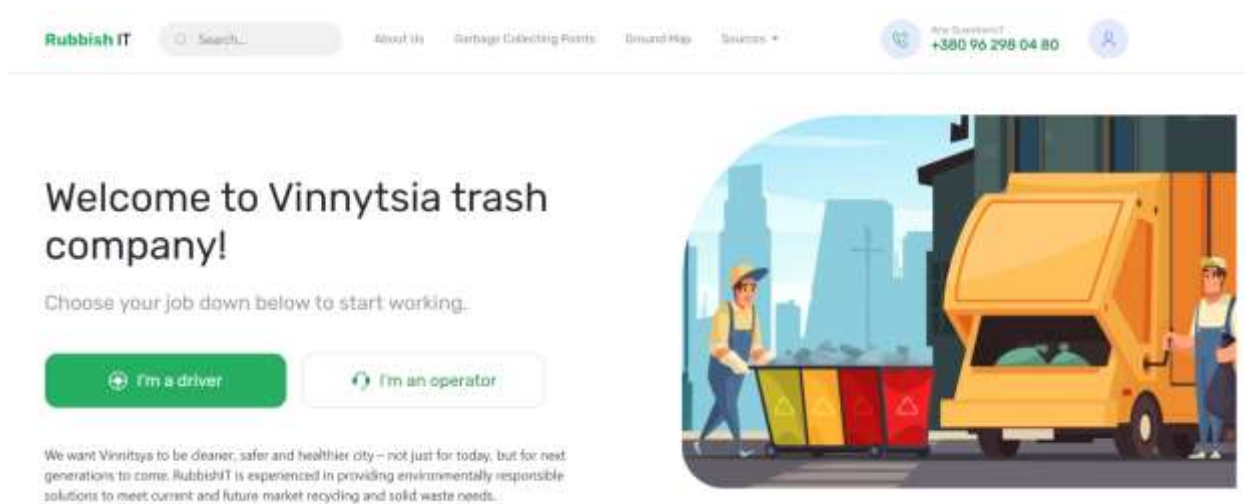


Рисунок 3.8 – Початкова сторінка інформаційної технології оптимізації процесу збору відходів

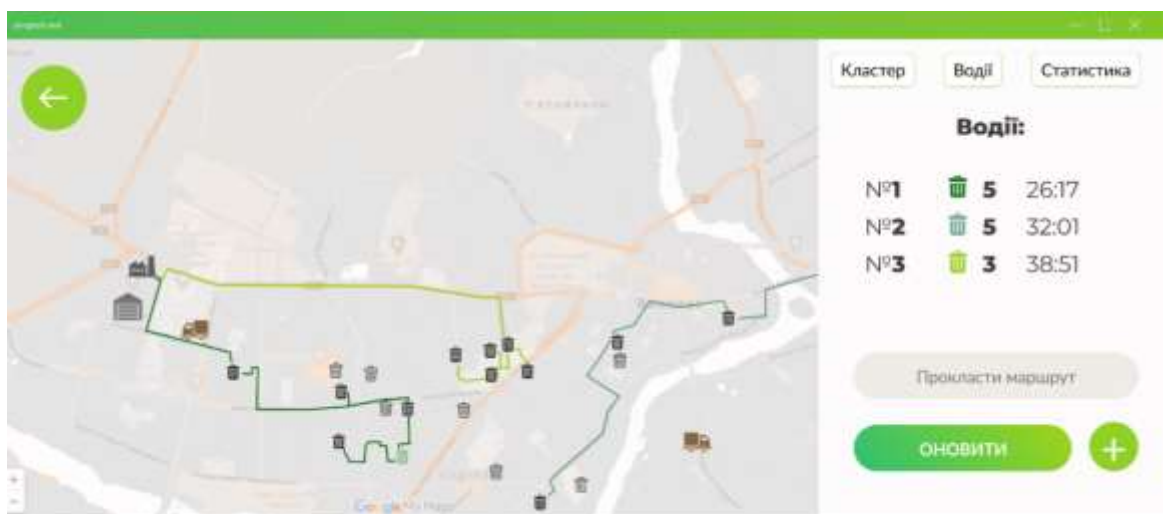


Рисунок 3.9 – Приклад роботи інтерфейсу диспетчера

На рисунку 3.10 зображено приклад роботи інтерфейсу водія, якому доступний власний оптимізований маршрут, стан наповненості кузова та найближча станція утилізації в межах його або сусіднього кластеру. На рисунку 3.11 показано інтерфейс містятнина, який передає дані щодо наповненості контейнера.



Рисунок 3.10 – Приклад роботи інтерфейсу водія

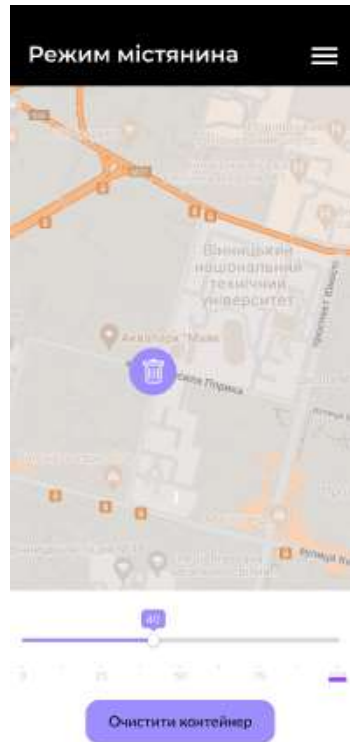


Рисунок 3.11 – Приклад роботи інтерфейса містянина

3.5 Висновки до третього розділу

1. Проаналізовано мови програмування та обрано JavaScript для розробки інформаційної технології, яка показала оптимальні результати у використанні пам'яті та швидкості роботи при реалізації нечіткого генетичного алгоритму

2. Подано структуру програмного забезпечення, яке розділено на 3 режими роботи: водія, якому надходять оновлені маршрути, диспетчера – для моніторингу роботи водіїв та кількості відходів на точках, який при потребі може перенаправити водія з одного кластера в інший, та містянина, який має змогу повідомляти про стан контейнерів та покращувати роботу диспетчера.

3. Розроблено програмне забезпечення, що дозволяє прокласти оптимальний або близький до оптимального маршрут збору відходів у територіальних громадах.

4. Роботу інформаційної технології протестовано на відкритих даних м. Рівне та порівняно з результатами попередніх досліджень. Використання нечіткого контролера дозволило покращити генетичні оператори та оптимізувати маршрут на 1,71.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки інформаційної технології оптимізації процесу збору відходів у територіальних громадах

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу інформаційної технології оптимізації процесу збору відходів у територіальних громадах, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Таблиця 4.1 – продовження

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Таблиця 4.1 – продовження

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	Номера експертів		
	Експерт 1	Експерт 2	Експерт 3

	Бали		
Технічна здійсненність концепції	4	2	4
Наявність аналогів на ринку	3	4	4
Цінова політика	4	5	4
Технічні та споживчі властивості виробу	3	4	4
Експлуатаційні витрати	4	3	2
Ринок збуту	3	4	4
Конкурентоспроможність	4	3	5
Фахівці з технічної і комерційної реалізації	3	4	4
Фінансування	5	4	3
Матеріально-технічна база	4	3	4
Термін реалізації ідеї	4	4	3
Супровідна документація	3	3	4
Сума	44	43	45
Середньоарифметична сума балів	$(44+43+45) / 3 = 44$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Провівши дослідження рівня комерційного потенціалу розробки «Інформаційної технології оптимізації процесу збору відходів у територіальних громадах» виявлено, що його рівень високий та становить 44 бала. Перевагами даної розробки є можливість контролювати рівень наповненості контейнерів у громаді та скласти оптимальний маршрут для вантажівок, враховуючи нечіткі зовнішні фактори впливу на дорогу.

4.2 Прогнозування витрат на виконання науково-технічної інформаційної технології оптимізації процесу збору відходів у територіальних громадах

Витрати, пов'язані з проведенням науково-технічної, дослідноконструкторської, конструкторсько-технологічної роботи, створенням дослідного зразка і здійсненням виробничих випробувань, під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуються за такими *статтями*:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково-виробничих цілей;
- витрати на службові відрядження;
- спецустаткування для наукових (експериментальних) робіт;
- програмне забезпечення для наукових (експериментальних) робіт;
- витрати на роботи, які виконують сторонні підприємства, установи і організації;
- інші витрати;
- накладні (загальновиробничі) витрати.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_i = 1 \frac{k \times M_{ni} \times t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників, залучених до процесу досліджень;

M_{pi} – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного дослідника, дн.;

T_p – середня кількість робочих днів в місяці, $T_p=21 \dots 23$ дні. Проведені розрахунки бажано звести до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	22300	1125,14	33	37129,62
Інженер	21000	1086,27	33	35846,91
Всього				72976,53

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які приймали участь в розробці інформаційної технології оптимізації процесу збору відходів у територіальних громадах. Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_д = З_о \cdot 11 \% / 100 \% \quad (4.2)$$

$$З_д = (72\,976,53 \cdot 11 \% / 100 \%) = 8027,42 \text{ (грн.)}$$

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Н_з = (З_о + З_р + З_{дод}) \times \frac{Н_{зп}}{100\%}, \quad (4.3)$$

де $H_{зп}$ – норма нарахування на заробітну плату.

$$Н_з = (72\,976,53 + 8027,42) \cdot 22 \% / 100 \% = 17820,87 \text{ (грн.)}$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія оптимізації процесу збору відходів у територіальних громадах».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.4)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 273,00 \cdot 1,1 - 0 \cdot 0 = 900,90 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір 500 80 г\м	273,00	3,0	0	0	900,90
Папір для записів 100 70 г\м	155,00	5,0	0	0	852,50
Органайзер офісний	174,00	2,0	0	0	382,80
Набір офісний (канцелярське приладдя)	222,00	3,0	0	0	732,60
Картридж для принтера	1465,00	1,0	0	0	1611,50
Диск оптичний CD-R	23,25	3,0	0	0	76,73

Flesh-пам'ять 64 GB	625,00	1,0	0	0	687,50
Тека для паперів	97,00	4,0	0	0	426,80
Всього					5671,33

До статті «Амортизація обладнання, програмних засобів та приміщень» відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A = \frac{Ц}{T_{\text{в}} \cdot 12} \cdot t_{\text{вик}} \quad [\text{грн.}] \quad (4.5)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 23000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, термін його фактичного використання – 2 міс.

$$A_{\text{обл}} = \frac{23000}{2} \times \frac{2}{12} = 1619,67 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик.}}{12} \quad (4.6)$$

Норму амортизації H_a приймемо за 10 %.

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер (Lenovo ideapad 3)	23000	4	2	956,33
Приміщення	1 252 000	20	2	10433,33
Ліцензійна ОС, та спеціалізовані ліцензійні нематеріальні ресурси	7800	-	2	71.042
Всього				11,460.702

До статті «Паливо та енергія для науково-виробничих цілей» належать витрати на придбання у сторонніх підприємств, установ і організацій будь-якого палива, що витрачається з технологічною метою на проведення досліджень. Стаття формується у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень.

$$B_e = \sum \frac{n W_{yi} \times t_i \times C_e \times K_{впнi}}{\eta_{i=1}} \quad (4.7)$$

$$B_e = \sum \frac{0,09 \times 310 \times 6,02 \times 0,9}{\eta_{i=1}} = 151,162$$

$$B_e = \sum \frac{0,031 \times 310 \times 6,02 \times 0,9}{\eta_{i=1}} = 52,067$$

$$B_e = \sum \frac{0,029 \times 85 \times 6,02 \times 0,9}{\eta_{i=1}} = 13,355$$

де W_{yi}^w – встановлена потужність обладнання на певному етапі розробки, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії);

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$; η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Проведені розрахунки необхідно звести до таблиці 4.6.

$$C_e = (C_{\text{опт}} + C_{\text{розп}} + C_{\text{пост}}) \left(1 + \frac{\text{ПДВ}}{100\%}\right), \quad (4.8)$$

$$C_e = (3.411 + 1.257 + 0.346) * 1.2 = 6.02 \text{ грн.}$$

де $C_{\text{опт}}$ - середня оптова ціна електроенергії, яка визначається оператором ринку (без ПДВ), грн за 1 кВт·год;

$C_{\text{розп}}$ - вартість розподілу електроенергії окремою енергорозподільною компанією (без ПДВ), грн за 1 кВт·год;

$C_{\text{пост}}$ - вартість постачання електроенергії від енергорозподільної компанії до конкретного споживача (без ПДВ), грн за 1 кВт·год.

ПДВ - величина податку на додану вартість, %, у 2022 році ПДВ=20%.

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Комп'ютер (Lenovo ideapad 3)	0.09	310	727,41
Монітор (ASUS VY249HE)	0.031	310	234,67
Освітлення	0.029	85	64.27
Всього			1026.35

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_{\text{с}} = (3_{\text{o}} + 3_{\text{p}}) \cdot \frac{H_{\text{іВ}}}{100\%}, \quad (4.9)$$

де $H_{\text{іВ}}$ – норма нарахування за статтею «Інші витрати».

$$I_{\text{с}} = 72976,53 * 115\% / 100\% = 83923,1 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нЗВ}} = (3_{\text{o}} + 3_{\text{p}}) \cdot \frac{H_{\text{нЗВ}}}{100\%}, \quad (4.10)$$

де $H_{\text{нЗВ}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нЗВ}} = 72976,53 * 126\% / 100\% = 91950,43 \text{ (грн.)}$$

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія оптимізації процесу збору відходів у територіальних громадах» належать витрати на відрядження штатних працівників, працівників

організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.11)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 23\%$.

$$B_{cv} = (72\,976,53 + 8027,42) \cdot 23 / 100\% = 18630,91 \text{ грн.}$$

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{zag} = Z_o + Z_p + Z_{dod} + Z_n + M + K_v + B_{spec} + B_{prg} + A_{obl} + \\ + B_e + B_{cv} + B_{cn} + I_v + B_{nzv}. \quad (4.12)$$

$$B_{zag} = 7276,53 + 10946,48 + 18463,06 + 1127,29 + 286,55 + 83923,1 + 91950,43 = \\ 213973,82 \text{ грн.}$$

Загальні витрати ZB на завершення науково-дослідної (науковотехнічної) роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{B_{zag}}{\eta}, \quad (4.13)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta =0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то

$\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$.

$$ЗВ = 213973,82 / 0,5 = 427\,947.64 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної м інформаційної технології оптимізації процесу збору відходів у територіальних громадах за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

з) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.14)$$

де $\pm\Delta\Pi_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_o = \Pi_o \pm \Delta\Pi_o$;

Π_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 45\%$;

\mathcal{D} – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році $\mathcal{D} = 18\%$;

Припустимо, що при прогнозованій ціні 7000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 20000 шт., протягом другого року – на 30000 шт., протягом третього року на 40000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\P_1 = (0*500 + (7000 + 500)*20000)*0,8333*0,45*(1 - 0,18) = 36\,898\,524 \text{ грн.}$$

$$\Delta\P_2 = (0*500 + (7000+500)*(20000+30000))*0,8333*0,45*(1 - 0,18) = 92\,246\,310 \text{ грн.}$$

$$\Delta\P_3 = (0*500 + (7000+500)*(20000+30000+40000))*0,8333*0,45*(1 - 0,18) = 166\,043\,358 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 295 188 192 грн.

Розраховуємо приведену вартість збільшення всіх чистих прибутків III , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-дослідної розробки:

$$III = \sum_1^T \frac{\Delta\P_i}{(1 + \tau)^t}, \quad (4.15)$$

де $\Delta\P_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\text{ПП} = (36\,898\,524 / (1+0,1)^1) + (92\,246\,310 / (1+0,1)^2) + (166\,043\,358 / (1+0,1)^3) = 33\,544\,112,73 + 76\,236\,619,83 + 124\,750\,832,46 = 234\,531\,565,02 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (4.16)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 427\,947.64 = 855\,895.28 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = \text{ПП} - PV, \quad (4.17)$$

$$E_{abc} = 234\,531\,565,02 - 611\,353,78 = 233\,675\,669,74 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.18)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 233\,675\,669,74 / 855\,895,28)} - 1 = 6,495$$

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_e}, \quad (4.19)$$

$$T_{ок} = 1 / 6,495 = 0,153 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,153 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновки до четвертого розділу

Економічна частина даної роботи містить розрахунок витрат на розробку інформаційної технології оптимізації процесу збору відходів у територіальних громадах, сума яких складає 427947.64 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним. Період окупності складе близько 0,153 роки.

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології та програмного забезпечення розв'язання задачі оптимізації процесу збору відходів у територіальних громадах.

Основні результати магістерської кваліфікаційної роботи полягають в такому:

1. Проаналізовано предметну область збору відходів Вінницької територіальної громади та засвідчено, що більшість маршрутів сміттєвозів не є

оптимальними. Виявлено такі основні недоліки постійних не гнучких маршрутів, як: проїзд сміттєвозів до незаповнених контейнерів; великі витрати часу сміттєвозів в заторах на шосейних дорогах та при проїзді путівцями за поганих погодних умов; повернення до полігону з незаповненим кузовом, – що засвідчує необхідність створення гнучких маршрутів, що враховують наповненість та доступність сміттєвих контейнерів.

2. Досліджено відомі методи формування оптимальних маршрутів сміттєвозів і обґрунтовано доцільність використання генетичного алгоритму для оптимізації маршруту збору відходів та застосування інструментарію нечіткої логіки, що дозволяє компенсувати неповноту вхідної інформації щодо наповненості сміттєвих контейнерів та їх доступності, а також підвищити ефективність роботи генетичного алгоритму за рахунок використання нечітких генетичних операторів.

3. Сформульовано постановку задачі збору відходів у територіальних громадах, що дозволило врахувати основні обмеження задачі та визначити найбільш зручну форму її подання у вигляді нечіткого графу.

4. Розроблено структуру інформаційної технології оптимізації процесу збору відходів у територіальних громадах, яка заснована на декомпозиції задачі збору відходів на три етапи: підготовки, кластеризації та маршрутизації, - що забезпечує зменшення розмірності задачі, спрощення її бізнес-процесів та відповідне підвищення продуктивності технології.

5. Запропоновано загальну модель задачі на основі нечіткого графу, яка забезпечує можливість нечіткого обчислення ступеню заповненості сміттєвих контейнерів та їх доступності, що дозволяє оптимізувати маршрути сміттєвозів за рахунок вилучення з них незаповнених та недоступних контейнерів.

6. Здійснено нечітку кластеризацію сміттєвих контейнерів на маршрути, що дозволяє динамічно коригувати маршрут сміттєвоза в режимі реального часу в разі виявлення в процесі збору сміття переоцінки завантаженості контейнерів при її попередньому прогнозуванні на основі неповних даних

7. Розроблено модель нечіткого логічного виведення, що дозволяє визначити ступень актуальності забору відходів із сміттєвого контейнеру на основі

неповних даних щодо його наповненості, погодних умов та завантаженості доріг, що дозволяє оптимізувати маршрути сміттєвозів за рахунок виключення з них частини сміттєвих контейнерів.

8. Обгрунтовано згальні вимоги щодо узгодження співвідношення локального та глобального пошуку генетичного алгоритму, що забезпечило основу для створення бази правил для нечіткого контролера, який здійснює автоматичне налаштування оптимальних значень параметрів нечітких генетичних операторів схрещування та мутації.

9. Розроблено нечіткий контролер автоматичного налаштування оптимальних значень імовірності застосування нечітких генетичних операторів, що забезпечило покращення оптимізації маршрутів генетичним алгоритмом дозатково на 1,5-3,5%.

10. Для програмної реалізації інформаційної технології оптимізації процесу збору відходів у територіальних громадах проаналізовано мови програмування та обгрунтовано вибір JavaScript, що показала оптимальні результати у використанні пам'яті та швидкості роботи при реалізації нечіткого генетичного алгоритму

11. Розроблено структуру програмного забезпечення інформаційної технології, яке розділено на 3 режими роботи: водія, якому надходять оновлені маршрути; диспетчера – для моніторингу роботи водіїв та кількості відходів в контейнерах, який при потребі може перенаправити водія з одного кластера в інший; містянина, який має змогу повідомляти про стан контейнерів та покращувати роботу диспетчера.

12. Розроблено програмне забезпечення, що дозволяє прокласти оптимальний або близький до оптимального маршрут збору відходів у територіальних громадах, з урахуванням наповненості та доступності сміттєвих контейнерів.

13. Роботу інформаційної технології протестовано на набір Вінницької територіальної громади та порівняно з результатами попередніх досліджень. Вилучення із маршруту ненаповнених та недоступних контейнерів та використання нечіткого контролера для автоматичного налаштування

імовірності використання генетичних операторів схрещування та мутації дозволило оптимізувати маршрути сміттєвозів на 6,3 – 9,8%.

14. Розраховано комерційний ефект від реалізації результатів розробки на три роки, що склала 233675669,74 грн. і період окупності, який складе близько 0,153 роки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хазівалієва І. І., Месюра В. І. «Урахування нечітких факторів при оптимізації маршруту збору відходів у територіальній громаді» в журналі «Наукові праці Вінницького національного технічного університету» [Електронний ресурс] Режим доступу: <https://praci.vntu.edu.ua/index.php/praci/article/view/662/625>
2. Хазівалієва І. І., Месюра В. І. «Інформаційна технологія управління процесом збору відходів у територіальних громадах на основі нечіткої логіки» в ЛІ Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2022) [Електронний ресурс] Режим доступу: <https://conf.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/viewPaper/14891>

3. Хазівалієва І. І., Месюра В. І. «Веб-додаток інформаційної технології оптимізації процесу збору відходів у територіальних громадах» в Міжнародна конференція «Контроль і управління в складних системах» [Електронний ресурс] Режим доступу: <https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/1645>
4. Хазівалієва І. І., Месюра В. І. «Інтелектуальна система збору міських відходів» в L Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2021) [Електронний ресурс] Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/11450>
5. Хазівалієва І. І., Мельник О. Д. «Municipal waste collection» в LI Науково-технічна конференція факультету будівництва, цивільної та екологічної інженерії (2022) [Електронний ресурс] Режим доступу: <https://publish.vntu.edu.ua/index.php/all-fbtegp/all-fbtegp-2022/paper/viewPaper/14933>
6. Правовий статус територіальної громади [Електронний ресурс] Режим доступу: <http://sts-mrada.gov.ua/?p=1498>
7. Як у Калинівській громаді вирішують питання із вивезенням твердих побутових відходів [Електронний ресурс] Режим доступу: <https://kalynivska-objednana-gromada.gov.ua/news/1616766759/>
8. Київ побудував інфраструктуру для IoT, планує встановлення датчиків і запуск "розумних" рішень для міста [Електронний ресурс] Режим доступу: <https://interfax.com.ua/news/telecom/795971.html>
9. Розумне управління відходами спільнот [Електронний ресурс] Режим доступу: https://eap-csf.eu/wp-content/uploads/Guideline_UA.pdf
10. О. В. Рибалова «Поводження з відходами», Х: НУЦЗУ, 2016, с. 224-227.
11. Децентралізація Вінницької області [Електронний ресурс] Режим доступу: <https://decentralization.gov.ua/areas/0432/gromadu>
12. Програма поводження з побутовими відходами у м. Вінниці на період 2013-2020 [Електронний ресурс] Режим доступу: <http://surl.li/escxjy>

13. У Вінниці оновлюють контейнерні майданчики [Електронний ресурс]
Режим доступу:<https://vitatv.com.ua/misto/u-vinnytsi-onovlyuyut-konteynerni-maydanch-uku>
14. Регіональний план управління відходами Вінницької області на період до 2030 року [Електронний ресурс] Режим доступу:
<extension://elhekieabhbkmcefcoobjddigjcaadp/http://www.vin.gov.ua/images/doc/vin/ODA/ogoloshenia/RPUV1507.pdf>
15. Вирішення проблеми комівояжера за допомогою нейронної мережі Хопфілда [Електронний ресурс] – Режим доступу: <https://www.researchgate.net/publication/331399707>
16. М. С. Тарков «Решение задачи коммивояжера с использованием рекуррентной нейронной сети» [Електронний ресурс] – Режим доступу: <https://doi.org/10.15372/SJNM20150308>
17. Нейросеть на практике: Задача коммивояжера [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/188938/>
18. М. С. Тарков «Решение задачи коммивояжера с использованием рекуррентной нейронной сети» [Електронний ресурс] – Режим доступу: <https://doi.org/10.15372/SJNM20150308>,
19. Борисов А.Н., Крумберг О.О., Федоров І.П. [Електронний ресурс] – Режим доступу:
extension://elhekieabhbkmcefcoobjddigjcaadp/https://iq.vntu.edu.ua/method/getfile.php?fname=69078.pdf&x=1&card_id=52157&id=69078
20. Задача комівояжера мережа Кохонена [Електронний ресурс] – Режим доступу <http://um.co.ua/4/4-17/4-177021.html>
21. Процес і система нечіткого логічного виведення [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/ne4itkalogika/necitka-logika/process-i-sistema-necitkogo-logicnogo-vivedenna>
22. В.Баланов «Аналіз факторів, які впливають на забезпечення руху вантажних потягів за розкладом» [Електронний ресурс] – Режим доступу: <http://tstt.diit.edu.ua/article/view/57057>.

23. Гонзалес-Калдерон К. «Route optimization of urban public transportation» [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/262429126_Route_optimization_of_urban_public_transportation
24. Jun Zheng «A Vehicle Routing Problem Model With Multiple Fuzzy Windows Based on Time-Varying Traffic Flow» [Електронний ресурс] – Режим доступу: <https://ieeexplore.ieee.org/document/9001084>
25. Оптимізація транспортної логістики в Україні [Електронний ресурс] – Режим доступу: <http://surl.li/есхур>
26. Митюшкин Ю. И. «Softcomputing: ідентифікація закономірностей нечіткими базами знань» [Електронний ресурс] – УНІВЕРСУМ-Вінниця. –2002. –146 с.
27. Ліценгер П. The route change of travel time based routing influenced by weather Sweden (2011) [Електронний ресурс] – Режим доступу: <https://www.diva-portal.org/smash/get/diva2:414346/FULLTEXT02.pdf>
28. Ситуаційний центр міста Вінниця (2022) [Електронний ресурс] – Режим доступу: <https://cutt.ly/cB2Rpgd>
29. Навчання нечіткої бази знань за вибіркою нечітких даних // Штучний інтелект (2006) [Електронний ресурс] <http://surl.li/есуак>
30. Контейнерні майданчики у Вінниці [Електронний ресурс] – Режим доступу: https://www.google.com/maps/d/u/0/viewer?mid=1364L8Cyg7OjL0DooGgiJ1iQ6G_GG8ZUa&ll=49.23058566629286%2C28.441538920066524&z=16
31. А. Ю. Кононюк «Нейронні мережі і генетичні алгоритми» – К:2008, с. 241
32. Геррера Ф, Лорано М. «Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions» [Електронний ресурс] – Режим доступу: <http://surl.li/есует>
33. Огляд мов програмування високого рівня [Електронний ресурс] – Режим доступу: <https://katerinka1200.github.io/index2.html>
34. Практичний аспект Ruby [Електронний ресурс] – Режим доступу: jak.koshachek.com

35. Язык программирования Python: преимущества, недостатки и область применения , [Электронный ресурс] – Режим доступа: https://skillbox.ru/media/code/dlya_chego_nuzhen_python/

ДОДАТКИ

Додаток А

Результат перевірки на плагіат в онлайн-системі UNICHECK



Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013332562

Дата перевірки:
19.12.2022 22:45:52 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.12.2022 22:47:10 EET

ID користувача:
62038

Назва документа: 122МКР-Хазівалієва12022

Кількість сторінок: 69 Кількість слів: 10815 Кількість символів: 81471 Розмір файлу: 1.58 MB ID файлу: 1013092065

8.54% Схожість

Найбільша схожість: 8.54% з Інтернет-джерелом (<https://praci.vntu.edu.ua/index.php/praci/article/download/662/625/8>..

8.54% Джерела з Інтернету

1

Сторінка 71

Не знайдено джерел з Бібліотеки

0.18% Цитат

Цитати

1

Сторінка 72

Не знайдено жодних посилань

8.71% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

3.71% Вилучення з Інтернету

38

Сторінка 73

5.99% Вилученого тексту з Бібліотеки

140

Сторінка 73

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

50

Додаток Б (обов'язковий)

Лістинг програми

```

GACromosome(actual_ratings,
predicted_ratings):
    """Get genetic feature weights using crossover
    and mutation."""

    # Generate initial binary population
    cur_gen_pop = np.random.randint(2,
    size=(INITIAL_POPULATION, FEATURE_LENGTH))
    next_gen_pop = np.zeros((cur_gen_pop.shape[0],
    cur_gen_pop.shape[1]))
    fitness_vector = np.zeros((INITIAL_POPULATION,

GACromosomeparams
=
np.multiply(next_g
en_pop, np.matrix(
    [np.float(np.random.normal(0, 2, 1)) if
np.random.random() < MUTATION_RATE else 1 for x in
range(next_gen_pop.size)]).reshape(next_gen_pop.shape
))
    cur_gen_pop = next_gen_pop

    best_soln =
np.array(cur_gen_pop[np.argmin(fitness_vector[:,
1])])
    return best_soln

FuzzyParameters(self,
age):
    """Get fuzzy set values of given
age."""
    return [self.young(age),
self.middle(age),
self.old(age)]

return
[self.very_bad(gim_value),
self.bad(gim_value),

```

```

        self.average(gim_value),
                    self.good(gim_value),

        self.very_good(gim_value),

        self.excellent(gim_value)]

winners
p.zeros((WINNER_PER_GEN,
FEATURE_LENGTH))

        for n in range(len(winners)):
            selected =
np.random.choice(range(len(fitness_vector)),
int(WINNER_PER_GEN/2),

replace=False) # select random indexes from pop

                    wnr =
np.argmin(fitness_vector[selected, 1]) # select
one index with min fitness error (tournament)
                    winners[n] =
cur_gen_pop[int(fitness_vector[selected[wnr]][0])]
# add to winner population

        next_gen_pop[:len(winners)] =
winners

// Мутація генів хромосом, випадковим чином переставляючи їх між собою
public void Mutate()
{
for (int position = 0; position < ChromosomeLength; position++)
{
        if (rand.NextDouble() < ChromosomeMutationRate)
            ChromosomeGenes, [position] = (ChromosomeGenes, [position]
+ rand.NextDouble()) / 2.0;
}
}

public void ExtractChromosomeValues(ref double, [] values)
{

```



```

        for (int i = 0; i < ChromosomeLength; i++)
            values, [i] = ChromosomeGenes, [i];
    }
var line = new GPolyline.fromEncoded({
    color: "#0000FF",
    weight: 5,
    opacity: 0.5,
    points:
public sealed class ChromosomeComparer : IComparer
{
    public int Compare(object x, object y)
    {
        if (!(x is Chromosome) || !(y is Chromosome))
            throw new ArgumentException("Not of type Chromosome");
        if (((Chromosome)x).ChromosomeFitness > ((Chromosome)y).
            ChromosomeFitness)
            return 1;
        else if (((Chromosome)x).ChromosomeFitness ==
            ((Chromosome)y).ChromosomeFitness)
            return 0;
        else
            return -1;
    }
}
{
    Chromosome g = ((Chromosome) CurrentGenerationList ,
        [PopulationSize - 1]);
    values = new double, [g.ChromosomeLength];
    g.ExtractChromosomeValues(ref values);
    fitness = (double)g.ChromosomeFitness;
}
}
public class Chromosome
{

```

```

public double, [] ChromosomeGenes;
public int ChromosomeLength;
public double ChromosomeFitness;
public static double ChromosomeMutationRate;

public Chromosome(int length, bool createGenes)
{
    ChromosomeLength = length;
    ChromosomeGenes = new double, [length];
    if (createGenes)
    {
        for (int i = 0; i < ChromosomeLength; i++)
            ChromosomeGenes, [i] = rand.NextDouble();
    }
}

public void Crossover(ref Chromosome Chromosome2, out Chromosome child1,
    out Chromosome child2)
{
    int position = (int)(rand.NextDouble() * (double)ChromosomeLength);
    child1 = new Chromosome(ChromosomeLength, false);
    child2 = new Chromosome(ChromosomeLength, false);
    for (int i = 0; i < ChromosomeLength; i++)
    {
        if (i < position)
        {
            child1.ChromosomeGenes, [i] = ChromosomeGenes, [i];
            child2.ChromosomeGenes, [i]= Chromosome2.ChromosomeGenes, [i];
        }
        else
        {
            child1.ChromosomeGenes, [i]= Chromosome2.ChromosomeGenes, [i];
            child2.ChromosomeGenes, [i] = ChromosomeGenes, [i];
        }
    }
}
}
}

```

Додаток В

Додаток В (Обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ЗБОРУ
ВІДХОДІВ У ТЕРИТОРІАЛЬНИХ ГРОМАДАХ

Виконала: студентка 2-го курсу, групи 2КН-21м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Хазівалієва І.І.

(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН

Месюра В.І.

(прізвище та ініціали)

« » 2022 р.

Вінниця ВНТУ - 2022 рік

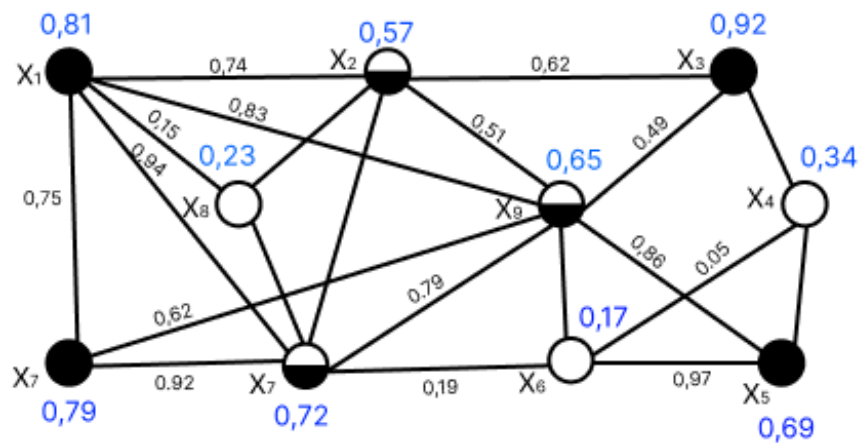


Рисунок В.1 – Розробка математичної моделі оптимізації процесу збору відходів

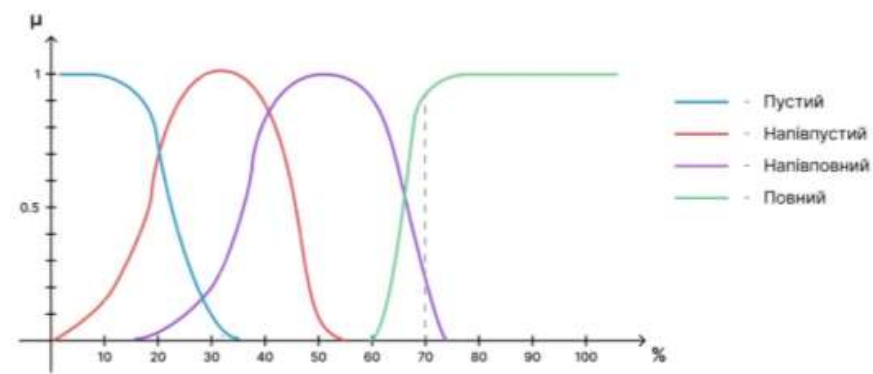


Рисунок В.2 – Нечітке прогнозування заповненості та доступності сміттєвих контейнерів

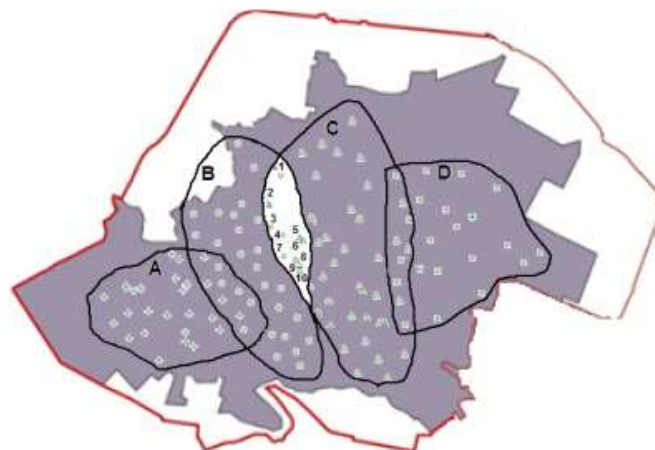


Рисунок В.3 – Нечітка кластеризація контейнерів

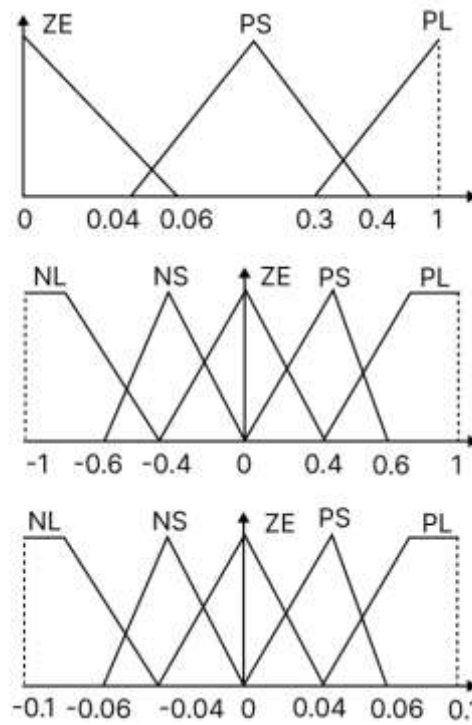


Рисунок В.4 – Нечіткий контролер оптимізації значень параметрів
генетичного алгоритму



Рисунок В.5 – Схема нечіткого генетичного алгоритму

Номер маршруту	1	2	3	4	5	6	7
Загальна кількість контейнерів, км	53	42	68	61	34	27	49
Довжина маршруту, класичний ГА, км	37,07	29,18	47,89	44,75	25,48	19,74	33,15
Кількість наповнених контейнерів, км	45	31	54	45	25	21	40
Довжина маршруту, класичний ГА, км	35,14	27,22	44,01	41,43	23,90	18,71	30,86
Відсоток покращення, км	5,2%	6,7%	8,1%	7,4%	6,2%	5,4%	6,9%
Довжина маршруту, ГА з контролером, км	34,73	27,02	43,19	41,12	23,31	18,37	30,33
Відсоток покращення	6,3%	7,4%	9,8%	8,1%	8,5%	6,9%	8,5%

Рисунок В.6 – Аналіз результатів тестування програми

Додаток Г

Інструкція користувача

Після запуску програми відкривається стартове вікно, в якому необхідно обрати режим диспетчера або водія. На початковій сторінці користувач може знайти інформацію про місця розташування контейнерів з датчиками наповненості сміття, місця утилізації та графік вивезення у громаді, що зображено на рисунку А.1.

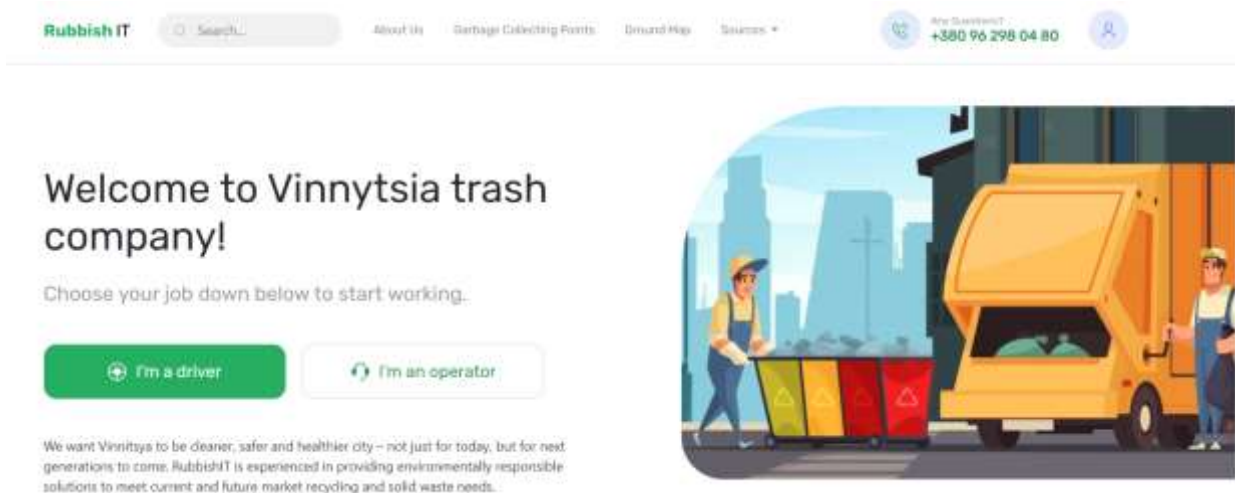


Рисунок А.1 – Стартове вікно

При авторизації в режимі диспетчера, йому доступна карта територіальної громади, яку він може ділити на кластери, в залежності від кількості водіїв, відслідковувати процес збору відходів та переправляти водія на інший кластер, якщо йому вистачає місця у кузові сміттєвоза. Також йому доступна статистика збору відходів за певний проміжок часу, що показано на рисунку А.2.

На рисунку А.3 показано приклад роботи режиму водія, маршрути в якого динамічно змінюються. Йому доступна карта контейнерів в межах свого кластера та відсоток наповненості свого кузова.

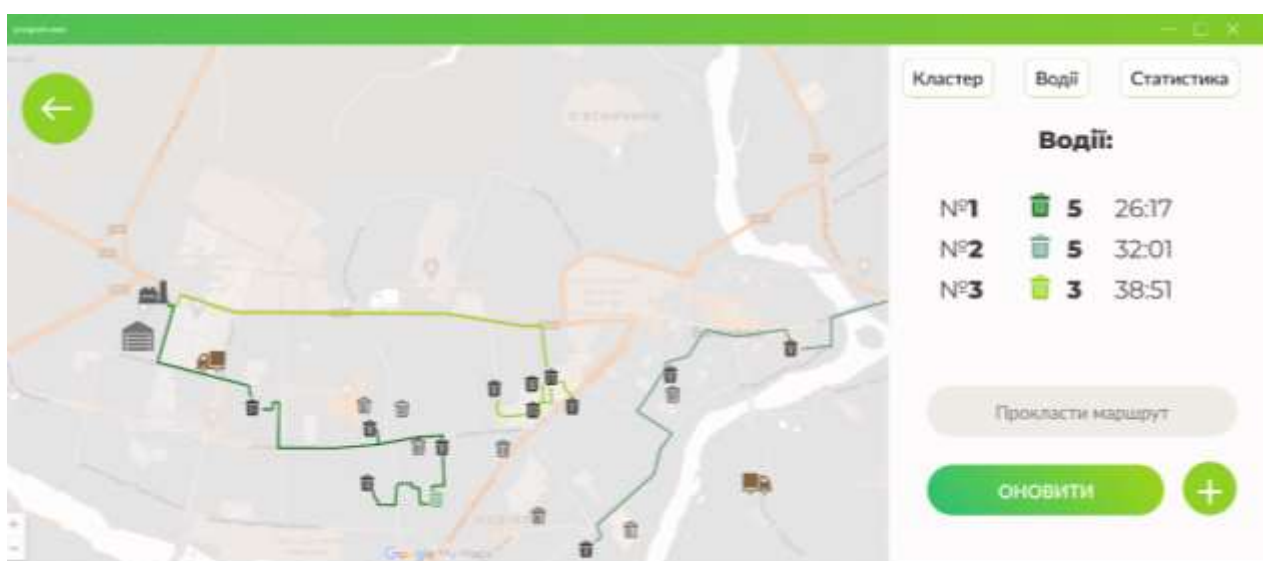


Рисунок А.2 – Інтерфейс диспетчера



Рисунок А.3– Приклад роботи інтерфейса водія

Незарєєстрований користувач стандартно вважається містянином та має можливість передати інформацію щодо наповненості контейнера або його переповненість за допомогою шкали від 0 до 100, що показано на рисунку А.4

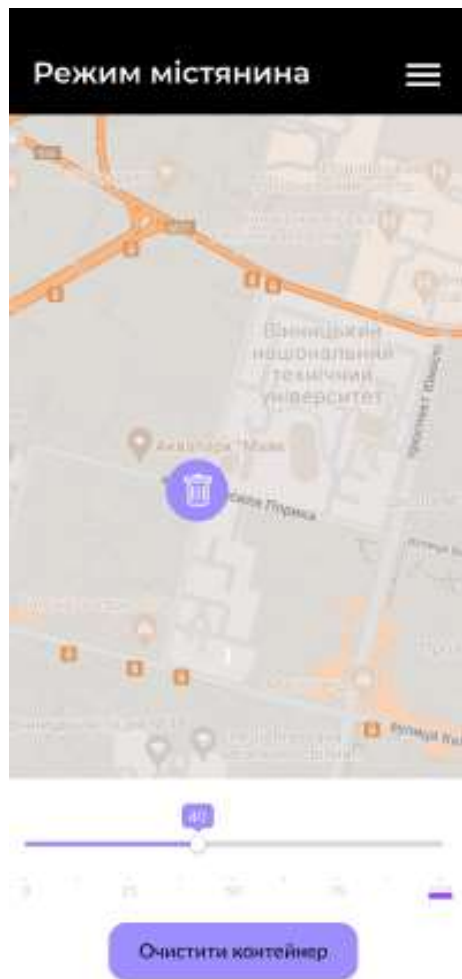


Рисунок А.4– Приклад роботи інтерфейса містянина

Додаток Г

Довідка про впровадження

Магазин «У Михайла»
смт. Теплик, вул. Незалежності, 32,
Вінницька обл. 23800
тел.: (098) 771 23 07

Довідка

Видана Хазівалієвій Ірині Ігорівні, студентці групи 2КН-21м в тому, що розроблений в магістерській кваліфікаційній роботі «Інформаційна технологія оптимізації процесу збору відходів у територіальних громадах» нечіткий генетичний алгоритм оптимізації маршруту планується для використання у проекті з доставки товарів споживачам.

Виконавчий директор
«У Михайла»

Пакула М. М.

