Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА СИСТЕМИ З ВИРІШЕННЯ ЛОГІСТИЧНОЇ ПРОБЛЕМИ ТРАНСПОРТУ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-19-2
Ясько О.С.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Любченко В.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Яську Олегу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи з вирішення логістичної проблеми транспорту

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 27 травня 2023 р.

3. Вихідні дані до роботи джерела інформації, що описують основну частину теоретичних досліджень, документація системи моделювання AnyLogic.

4. Перелік питань, що потрібно опрацювати в роботі

1. Провести порівняльний аналіз ефективності існуючих методів вирішення транспортної задачі.

2. Побудувати генетичні оператори, що відображають специфіку розв'язуваної задачі ланцюга постачання у мережі магазинів.

3. Побудувати імітаційну модель для оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів та провести експерименти.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Моделювання логістичних проблем транспорту; ціль та задачі дослідження; модель класичної VRP; узагальнення та розширення VRP; еволюційне моделювання за допомогою генетичних алгоритмів; загальна схема вирішення задачі за допомогою генетичного алгоритму; основні генетичні оператори в системі; модель в системі AnyLogic; агенти Main, Manufacturer, Supplier та Truck.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-15.04.23	
3	Аналіз літератури з досліджуваної проблеми	16.04.23-20.04.23	
4	Аналіз генетичного алгоритму вирішення ТЗ	21.04.23-28.04.23	
5	Розробка імітаційної моделі для вирішення ТЗ	29.04.23-12.05.23	
6	Програмна реалізація	13.05.23-21.05.23	
7	Оформлення пояснювальної записки	22.05.23-24.05.23	
8	Перевірка на плагіат	26.05.23	
9	Рецензування	27.05.23	
10	Підготовка презентації та доповіді	28.05.23-02.06.23	
11	Занесення роботи в електронний архів	05.06.23	
12	Попередній захист кваліфікаційної роботи	05.06.23	

Дата видачі завдання 10 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ _____доц. Любченко В.А._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 63 с., 4 табл., 40 рис., 1 дод., 31 джерело.

ТРАНСПОРТНА ЗАДАЧА, МАРШРУТИ, ГЕНЕТИЧНИЙ АЛГОРИТМ, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, ФІТНЕС-ФУНКЦІЯ, КРОСИНГОВЕР.

Об’єктом роботи є планування та оптимізація транспортних перевезень.

Метою роботи є підвищення ефективності системи планування та оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів.

Використано методи еволюційного моделювання, оптимізації, імітаційного моделювання. Проведено дослідження моделей та методів оптимізації транспортних перевезень, порівняльний аналіз існуючих методів вирішення транспортної задачі. Побудовано генетичні оператори, що відображають специфіку розв’язуваної задачі ланцюга постачання у мережі магазинів. Реалізовано імітаційну модель для оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів.

TRANSPORT PROBLEM, ROUTES, GENETIC ALGORITHM, SIMULATION MODELING, FITNESS FUNCTION, CROSSOVER.

The object of the work is the planning and optimization of transportation.

The purpose of the work is to improve the efficiency of the system of planning and optimization of transportation in the network of stores of the city based on genetic algorithms.

Methods of evolutionary modeling, optimization, and simulation modeling were used. The study of models and methods of transportation optimization, a comparative analysis of existing methods of solving the transport problem was carried out. Genetic operators reflecting the specifics of the solvable problem of the supply chain in the chain of stores have been constructed. A simulation model was implemented to optimize transportation in the city’s chain of stores based on genetic algorithms.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

VRP – Vehicle Routing Problem (завдання маршрутизації)

ТЗ – транспортний засіб

ГА – генетичний алгоритм

ВСТУП

Завдання маршрутизації (Vehicle Routing Problem, VRP) є одним з найбільш важливих і одночасно складних типів завдань комбінаторної оптимізації в галузі транспортної логістики.

Підвищений інтерес до VRP викликаний одночасно її практичною значущістю і значною складністю. Завдання найбільш часто зустрічається серед компаній, що виконують розвезення товару з деякого пункту виробництва або складу до пунктів споживання або точок роздрібної торгівлі але існують і інші області застосування, найбільш типові серед них: сервісне обслуговування, кур'єри і поштові служби, військова логістика, збір відходів, очищення вулиць, приватні перевезення, маршрутизація громадського транспорту, доставка швидкого харчування тощо.

Актуальність роботи полягає в питанні пошуку ефективних наближених алгоритмів, евристик та оптимізаційних підходів, що дадуть змогу компаніям, організаціям отримати прийнятну якість розв'язання в розумний час.

1 АНАЛІЗ ОСОБЛИВОСТЕЙ ТА МЕТОДІВ ВИРІШЕННЯ ЛОГІСТИЧНИХ ПРОБЛЕМ ТРАНСПОРТУ

1.1 Логістичні проблемі транспорту

Транспортування, яке включає переміщення матеріальних ресурсів, незавершеного виробництва або готової продукції транспортним засобом, є важливою комплексною активністю в логістичному ланцюзі [1]. Транспортна логістика є функціональною сферою логістики, яка оптимізує логістичні операції для пересування матеріальних потоків від постачальника до кінцевого споживача з використанням транспортних засобів. Транспорт є складовою основних функціональних галузей логістики, пропонуючи на ринку товарів і послуг свою продукцію – транспортні послуги, за які отримує прибуток. Проблеми транспортної галузі поглиблюються через неспроможність підприємств-споживачів транспортних послуг забезпечити достатні обсяги перевезень та відповідний рівень дохідності [2].

Вибір типу транспортної складової логістичних систем залежить від факторів, таких як вид вантажу, вартість перевезень, мета транспортування, відстань та якість транспортних шляхів. У сучасних умовах транспортне обслуговування визначається оптимальним співвідношенням витрат та прибутку, а також мінімізацією загальних логістичних витрат. У зв’язку зі складною економічною ситуацією в країні, працівники транспортної галузі повинні приділяти велику увагу питанням організації та управління перевезеннями, підвищенню якості послуг та вибору каналів руху товару [3, 4]. Однією з головних цілей логістики в галузі транспортування є забезпечення безперебійного переміщення товарів та транспортних засобів від пункту відправлення до призначення.

Принципи транспортної логістики є ключовими напрямками для покращення транспортних технологій, інтеграції виробничих та транспортних процесів [5]. Різновиди перевезень, які залежать від видів транспортних засобів,

мають важливе значення для логістики транспортування. Сьогодні транспортна логістика зосереджується на визначенні обсягів перевезень, напрямків та номенклатури транспортованих вантажів, а також на визначенні суб'єктів товарного ринку, які належать до підсистем логістики транспортування.

1.2 Огляд постановок задач маршрутизації транспортних засобів

Логістична проблема транспорту може бути вирішена шляхом застосування різних методів та стратегій. Основними методами є оптимізація маршрутів, зменшення часу погрузки та розвантаження, покращення управління запасами, ефективне використання транспортних засобів, технології відслідковування вантажів та зниження витрат на логістику.

Крім того, можливість використання різних видів транспорту (автомобільний, морський, повітряний, залізничний та ін.) може сприяти вирішенню логістичних проблем транспорту. Наприклад, використання мультимодальних перевезень, які поєднують різні види транспорту для досягнення кінцевої точки, може знизити час доставки та витрати на логістику.

Класична задача маршрутизації транспортних засобів VRP (Vehicle Routing Problem) – задача комбінаторної оптимізації, в якій для парку однотипних транспортних засобів (ТЗ) потрібно визначити оптимальний набір замкнутих маршрутів від єдиного депо до множини віддалених клієнтів [6, 7]. На практиці критерій оптимальності може виражатися будь-якими витратами на об'їзд клієнтів, але частіше за все відповідає довжині маршрутів. На рисунку 1.1 показаний типовий приклад побудови маршрутів для парку з трьох ТЗ при мінімізації сумарної довжини маршрутів. У центрі рисунку розташоване депо, в якому спочатку знаходиться парк з декількох однотипних ТЗ, а на деякій відстані від нього розташовані клієнти, яких потрібно відвідати. Відстань між усіма пунктами вважається відомою. Оптимальне рішення являє собою набір найкоротших маршрутів для ТЗ через всіх клієнтів з поверненням в депо.

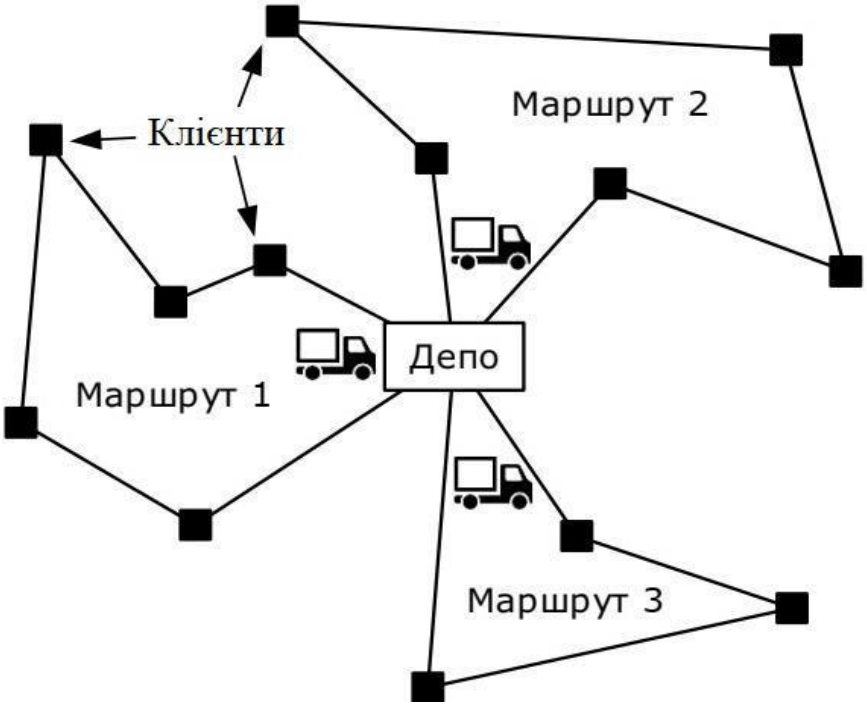

Рисунок 1.1 – Приклад рішення класичної VRP

Класична VRP може бути представлена на графі $G = (V, E)$ з множиною вершин $V = \{0, n\}$ і множиною ребер E. Вершини $i = 1, n$ відповідають клієнтам, тоді як вершина 0 відповідає депо. Також задається матриця $n \times n$ невід'ємних вартостей c_{ij}, пов'язаних з кожним ребром $(i, j) \in E$ і визначають витрати на пересування між вершинами *i* та *j*. Тоді в термінах теорії графів рішення класичної VRP зводиться до побудови декількох гамільтонових циклів мінімальної довжини на підграфі графу G з однією спільною вершиною-депо [8]. При відсутності обмеження вантажопідйомності і довільній кількості ТЗ завдання зводиться до побудови k циклів із загальною вершиною 0, які в сукупності містять всі вершини графа і мінімізують суму вартостей об'їзду клієнтів, тобто отримуємо множинне завдання комівояжера (TSP) з k ізольованими контурами з загальною точкою 0). Таке завдання можливо перетворити до звичайної TSP додаванням *k-1* додаткових копій вершини 0 і суміжних з нею ребер.

Задача комівояжера – це одна з класичних задач комбінаторної оптимізації, яка полягає в тому, щоб знайти найкоротший можливий шлях, який проходить через задану множину точок (міст), при цьому кожне місто потрібно відвідати тільки один раз і повернутися до початкової точки [9]. Ця задача теоретично має розв'язання, але для великих множин міст, коли кількість можливих шляхів зростає експоненційно, знайти точне розв'язання стає непрактичним через велику обчислювальну складність. Тому, для таких випадків застосовуються евристичні алгоритми, які дають наближені розв'язання, але не гарантують знаходження найкращого розв'язку. Таких алгоритмів є дуже велика кількість. Деякі з них відійшли на другий план через недостатню швидкість вирішення запитів та точності їх результатів [10].

В силу обмеженості класична постановка слабо співвідноситься з практикою. Сьогодні існує велика кількість різновидів VRP, велика частина яких є комбінаціями декількох основних розширень класичного варіанту. Всі вони відрізняються, головним чином, різними реальними обмеженнями, що накладаються на одержуване рішення [11, 12].

Варто відмітити, що розробка та застосування новітніх технологій, таких як штучний інтелект, машинне навчання та Інтернет речей, можуть також допомогти вирішити логістичні проблеми транспорту, забезпечуючи більш ефективний та точний контроль за процесом транспортування вантажів.

Різновиди VRP можна представити у вигляді ієрархічної схеми (рис. 1.2). З неї видно, що в основі більшості різновидів VRP мається на увазі базове обмеження вантажопідйомності (CVRP), яке в явному вигляді при описі може не вказуватися. Очевидно, за рахунок сполучень розглянутих вище різновидів VRP можна отримати множину варіантів її постановок.

На рисунку 1.3 в якості прикладу показано зв'язок декількох відомих похідних варіантів VRP, узагальнюючих чотирьох базових розширень [13]: відкриту (OVRP); з множиною депо (MDVRP); з обмеженням вантажопідйомності (CVRP); з часовими вікнами (VRPTW) [14].

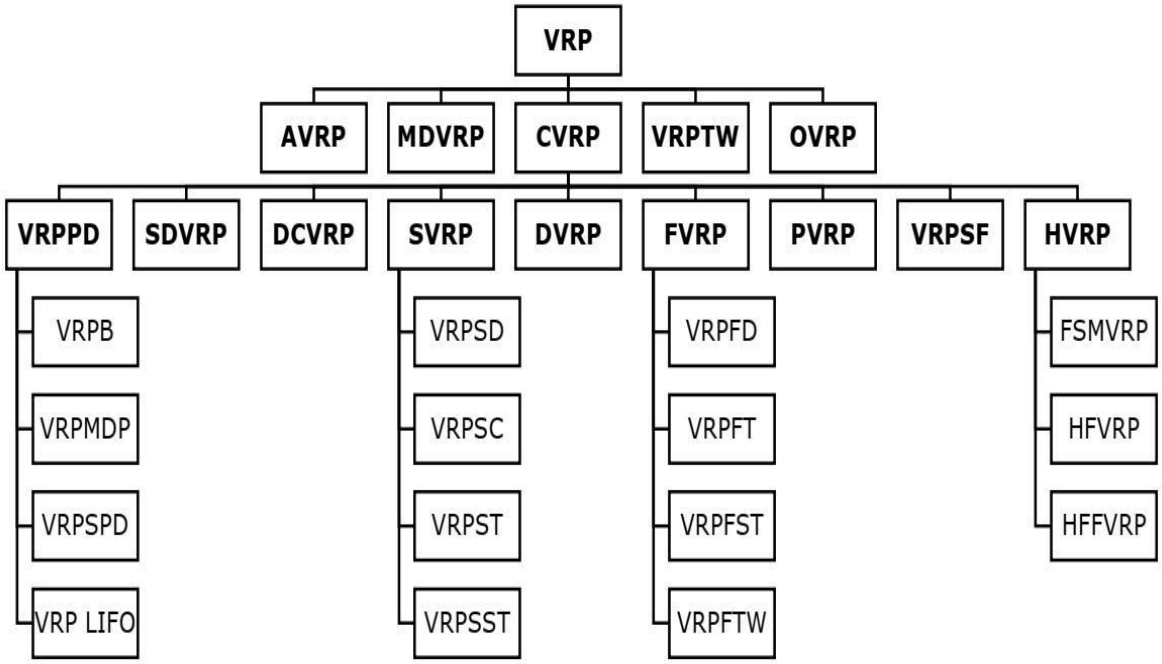

Рисунок 1.2 – Ієрархічна схема узагальнень і розширень VRP

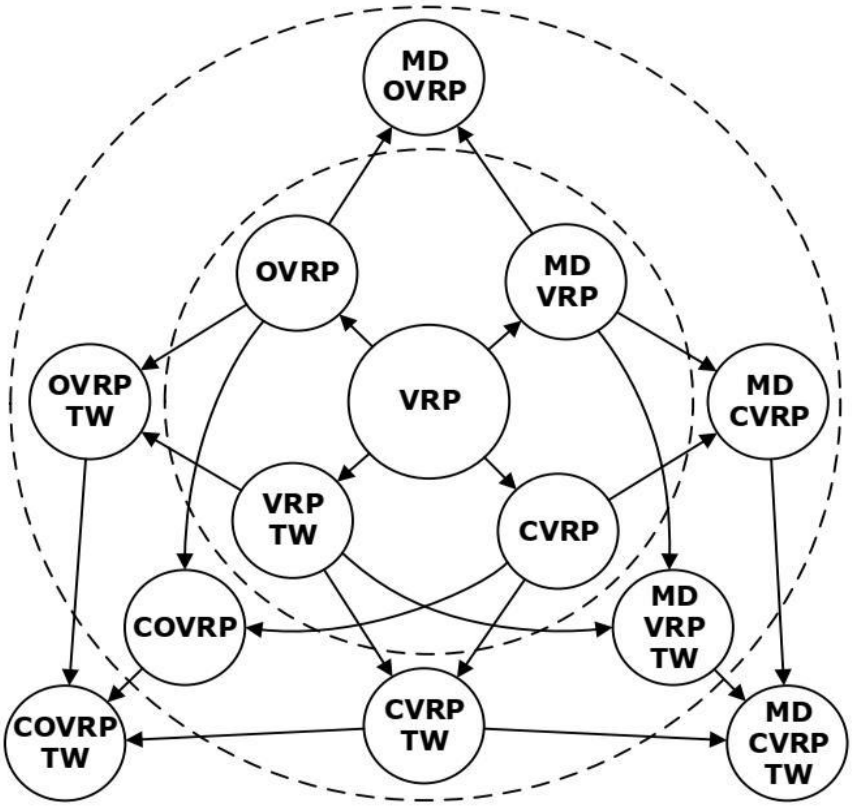

Рисунок 1.3 – Схема зв’язку узагальнюючих різновидів VRP

У літературі їх позначення прийнято формувати за принципом додавання до абревіатури букв, що характеризують складові розширення [15]. Відповідно

до загальноприйнятої системи назви завдання і її скорочення можуть виходити досить довгими, наприклад, Multi-Depot Capacitated Vehicle Routing Problem with Time Window (MDC VRPTW) – задача маршрутизації транспорту з множиною депо, обмеженням по вантажопідйомності і часовими вікнами.

В таблиці 1.1 наведені різновиди завдання, які можуть розглядатися як узагальнення класичного варіанту VRP, із зазначенням достатніх умов відомості до окремого випадку відповідно ієрархічної схеми (рис. 1.2). Існують відомі моделі розглянутих різновидів VRP із загальним підходом до формулювання, наприклад, як задач цілочисельного програмування.

Для вирішення всіх найбільш відомих узагальнень і розширень VRP існує значна кількість методів, кожен з яких має свої переваги і недоліки – в цьому ключі кожен варіант завдання заслуговує окремого огляду.

В результаті аналізу можна зробити висновок, що найбільш поширені узагальнення і розширення VRP можуть бути об’єднані в рамках узагальненої постановки, що дозволило б розробляти рішення на основі єдиної моделі.

Транспортна задача з обмеженням за часом описується наступним чином [16]. Є деяка кількість автотранспорту, один склад (депо) і деяка кількість клієнтів. Для кожного транспортного засобу потрібно скласти маршрут, протягом якого транспортний засіб відвідує ряд клієнтів (наприклад, з метою доставки будь-якого вантажу). На маршрут кожного транспортного засобу накладається ряд обмежень. Кожен маршрут повинен починатися і закінчуватися в депо. Загальна кількість товарів, необхідних для доставки клієнтам на даному маршруті даного транспортного засобу, не повинна перевищувати його вантажопідйомність. Кожен клієнт обслуговується лише одним транспортним засобом і лише одного разу, тобто не допускається відвідування одного клієнта двома і більше транспортними засобами. Кожен клієнт повинен бути обслужений в певний проміжок часу, цей проміжок визначається двома значеннями; перше значення визначає час прибуття транспортного засобу до клієнта, друге – час відправлення.

Таблиця 1.1 – Зв’язок узагальнень VRP з класичним варіантом

Задача	Особливість	Умова взведення до класичної VRP
AVRP	Асиметрична матриця вартостей	Ваги дуг для взаємно протилежних напрямків рівні
CVRP	Обмеження вантажопідйомності	Вантажопідйомність кожного ТЗ більше або дорівнює сумарному попиту всіх клієнтів
DCVRP	Обмеження відстані	Допустима довжина маршруту більше або дорівнює максимальній довжині Гамільтона циклу через всі вершини графа
VRPTW	Часові вікна	Нижнє значення часового вікна менше або дорівнює часу шляху від вершини-депо до вершини-клієнта, верхнє – найбільш пізнього з усіх можливих
MDVRP	Множина депо	Ваги ребер, що з’єднують вершини-депо з вершинами клієнтами, приймаються однаковими для всіх депо, тобто розташування всіх депо збігається
PVRP	Розширений період планування	Можливі відвідування всіх клієнтів обмежені одним єдиним днем
VRPPD	Крім доставки потрібно вивезення	Величина запитів на вивезення для всіх вершин-клієнтів дорівнює нулю
SVRP	Випадкові параметри	Щільність ймовірності випадкового параметра більше нуля тільки для певного вузького діапазону
DVRP	Динамічні параметри	Час до зміни параметрів одно або перевищує початковий час виконання завдання
HVRP	Різні ТС	Параметри транспорту і ваги ребер графа для всіх видів транспорту рівні

Для даної задачі сформульовано такі цілі (цільові функції):

– мінімізувати загальну кількість транспортних засобів, необхідних для обслуговування всіх клієнтів;

– мінімізувати загальний час обслуговування всіх клієнтів і загальну відстань, пройдену всіма транспортними засобами.

1.3 Методи вирішення задачі маршрутизації транспорту

Як було зазначено, VRP є завданням комбінаторної оптимізації, в якій число допустимих маршрутів експоненційно зростає при збільшенні числа клієнтів, і належить до класу складності NP [15, 16]. Повний перебір можливих її рішень хоча і дозволяє знайти оптимум, але вимагає колосальної кількості часу обчислень навіть при відносно невеликій розмірності задачі (14 і більше). Пропонувалися такі точні методи вирішення VRP, як, наприклад, метод гілок і меж, метод гілок з відсіканням. Вони є розвитком методу повного перебору, на відміну від останнього – з відсівом підмножин допустимих рішень, свідомо не містять оптимальних рішень. Однак час їх обчислень все одно зростає занадто швидко, а в гіршому випадку – як і при повному переборі.

Пошук рішень VRP почався в 60-і роки XX століття. Евристичні методи, які в наші дні називають класичними, розроблені переважно у період 60-90 років минулого століття. Протягом останніх двадцяти років велика кількість успішних і надзвичайно ефективних евристик були запропоновані для вирішення VRP. Нові підходи дають все більш точні результати. Але при цьому більшість з них не в змозі забезпечити хороший компроміс між якістю і продуктивністю. Одночасне вдосконалення алгоритмів як з точки зору ефективності, так і з точки зору продуктивності є відкритим завданням.

В даний час зусилля спрямовані переважно на новий напрямок в теорії і практиці штучного інтелекту – так звані метаевристики [17].

Метаевристики є сучасним потужним і надзвичайно популярним класом оптимізаційних методів, що дозволяють знаходити рішення для широкого кола

завдань з різних додатків. Сила метаевристик полягає в їх здатності вирішувати складні завдання без знання простору пошуку, саме тому ці методи дають можливість вирішувати складні для задачі оптимізації. У метаевристичних методах упор робиться на ретельному вивченні найбільш перспективних частин простору рішень [18]. Якість одержуваних рішень виходить вище, ніж у отриманих класичними евристиками. Особливість метаевристичних алгоритмів в тому, що вони не дають точного опису порядку дій для вирішення завдання, і кожен з них повинен бути додатково конкретизованим шляхом підбору значень керуючих параметрів.

Наведемо приклади метаевристичних алгоритмів, які вже успішно застосовувалися для вирішення VRP:

- пошук з винятками;
- модельований відпал;
- генетичний алгоритм;
- мурашиний алгоритм;
- нейронні мережі.

Сильною стороною розглянутих алгоритмів є швидкість роботи навіть при великій розмірності задачі і можливість подолання локального мінімуму в процесі пошуку. Відповідно до думки дослідників такі алгоритми, як генетичні і мурашині, зараз ще відносно слабко вивчені, хоча містять великий простір для подальшого пошуку поліпшень, потенційно здатних привести до помітного зростання якості.

В цій роботі планується використовувати саме генетичні алгоритми (ГА).

ГА – це клас метаевристичних алгоритмів, які застосовуються для вирішення широкого кола комбінаторних задач, в тому числі і для вирішення транспортної задачі [19]. Дані алгоритми є випадково-спрямованими пошуковими алгоритмами і засновані на ітеративній адаптації популяції (індивідів, хромосом, рішень) протягом певного проміжку часу, заданого у вигляді кількості ітерацій роботи алгоритму. Це означає, що спочатку створюється популяція рішень. Потім на кожній ітерації алгоритму до вихідної

популяції застосовуються оператори мутації, кросинговер а й селекції, в результаті чого популяція рішень піддається зміні (еволюції); краще рішення в кінцевій отриманій популяції є остаточним рішенням завдання. Процедура вибору рішень називається селекції і здійснюється на основі оцінки цільової функції рішення (часто в літературі цю функцію називають «функція придатності рішення» або «фітнес-функція») [20, 21], призводить до того, що в результаті отримуємо рішення-нащадки, з яких формується нова популяція; потім до рішень нової популяції застосовується оператор мутації, дана нова популяція на наступній ітерації буде вихідною для отримання наступної популяції і т.д. Застосування оператора кросинговеру має бути побудовано таким чином, щоб закріпити в нащадках кращі властивості рішень-батьків (у разі транспортної задачі рішенням-нащадкам повинні передаватися «кращі» маршрути рішень-батьків), застосування оператора мутації має розширити простір пошуку за рахунок часткової довільної зміни рішень.

1.4 Сучасні застосунки для роботи з логістичними задачами

1.4.1 OptaPlanner

OptaPlanner – це відкрите програмне забезпечення на базі Java, яке використовує алгоритми штучного інтелекту, щоб оптимізувати маршрутизацію транспорту та планування виробничих процесів.

Переваги сервісу:

– відкритий код: OptaPlanner є відкритим сервісом, що дозволяє користувачам використовувати, змінювати та розповсюджувати його безкоштовно;

– підтримка різноманітних задач: OptaPlanner дозволяє розв’язувати не тільки задачу комівояжера, але й інші задачі, такі як планування розкладів тижня, занять, розподіл ресурсів та інші;

– ефективність: OptaPlanner використовує різні оптимізаційні алгоритми, що дозволяє знайти ефективні рішення для складних задач;

– легкість використання: OptaPlanner має простий інтерфейс та документацію, що дозволяє легко використовувати його навіть для користувачів з обмеженим досвідом.

Недоліки:

– обмежені можливості маштабування: OptaPlanner може мати проблеми з обробкою великих об’ємів даних та складних задач;

– відсутність графічного інтерфейсу: OptaPlanner не має графічного інтерфейсу, що може бути не зручним для користувачів, що шукають візуалізацію результатів.

1.4.2 Мова моделювання AMPL

AMPL – це мова моделювання та система оптимізації, яка дозволяє створювати та розв’язувати складні математичні моделі, зокрема задачі комівояжера.

Переваги AMPL:

– AMPL надає зручний та потужний інтерфейс для формулювання та розв’язування математичних програм;

– інтерфейс AMPL дозволяє використовувати різні мови програмування, такі як C, C++, Java, Python та інші, для розширення функціональності та забезпечення інтеграції з іншими програмними засобами;

– AMPL має широку підтримку різних типів задач математичного програмування, включаючи лінійне та нелінійне програмування, задачі цілочисельного програмування, задачі динамічного програмування та інші;

– AMPL надає потужний набір інструментів для аналізу та візуалізації результатів розв’язку задач, включаючи графіки, таблиці та інші інструменти.

Недоліки AMPL:

– для використання AMPL потрібно мати досить глибокі знання в області математичного програмування та досвід роботи зі збіркою та виконанням коду на конкретній мові програмування;

– AMPL є комерційним продуктом, тому використання його може бути досить дорогим для невеликих команд або індивідуальних користувачів;

– навіть з великою кількістю інструментів та можливостей, AMPL не є універсальним інструментом для розв’язування всіх типів задач математичного програмування, тому для певних типів задач може бути необхідно використовувати інші інструменти.

1.4.3 UPS WorldShip

UPS WorldShip – це програмне забезпечення для управління логістичними процесами, яке використовує задачу комівояжера для оптимізації маршрутів доставки посилки.

Переваги UPS WorldShip:

– надійність: UPS є одним з провідних постачальників логістичних послуг та має досвід у доставці вантажів в усьому світі. Їхнє програмне забезпечення, включаючи WorldShip, є дуже надійним та має великий досвід роботи з клієнтами;

– інтеграція з іншими системами: UPS WorldShip дозволяє інтегруватися з іншими системами, що дозволяє спрощувати процеси та підвищувати ефективність.

Недоліки UPS WorldShip:

– висока вартість: UPS WorldShip є платним сервісом, що може бути високою вартістю для малих бізнесів або фізичних осіб;

– складний інтерфейс: для новачків може бути складно зрозуміти та використовувати UPS WorldShip через складний інтерфейс та велику кількість функцій.

1.4.4 Середовище моделювання AnyLogic

Програмний продукт AnyLogic являє собою середовище імітаційного моделювання та використовується для різних сфер: виробничі системи; логістика та ланцюжки поставок; ринок та бізнес-процеси; охорона здоров'я; телекомунікації та інформаційні системи; соціальні та екологічні системи тощо. Він має можливість створювати моделі за допомогою всіх трьох сучасних підходів: дискретно-подієвого, агентного та системної динаміки.

Імітаційні моделі ланцюга поставок AnyLogic дозволяють отримати глибше розуміння та оптимізувати складні системи та процеси. Це потужний інструмент аналізу ланцюга постачання, який можна інтегрувати з поточним робочим процесом.

Додатковою перевагою AnyLogic для моделювання логістичних систем є вбудовані механізми використання ГІС-системи на базі OpenStreetMap. OpenStreetMap – це вільний, керований спільнотою проєкт, метою якого є створення вільної та редагованої карти світу. Дані OSM можна використовувати з будь-якою метою, в тому числі комерційною.

1.5 Постановка задачі

Таким чином, розв'язання задачі комівояжера є актуальним у багатьох галузях, де застосовуються замкнуті системи з чіткими часовими рамками. Тому буде розглянута задача підвищення ефективності вирішення транспортної задачі для ланцюга постачання продуктів у мережі магазинів у місті за допомогою генетичних алгоритмів.

Об'єктом роботи є планування та оптимізація транспортних перевезень.

Метою роботи є підвищення ефективності системи планування та оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів.

Для досягнення мети необхідно вирішити такі завдання:

– провести порівняльний аналіз ефективності існуючих методів вирішення транспортної задачі;

– побудувати генетичні оператори, що відображають специфіку розв’язуваної задачі ланцюга постачання у мережі магазинів;

– побудувати імітаційну модель для оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів;

– провести експерименти.

2 ГЕНЕТИЧНИЙ АЛГОРИТМ РІШЕННЯ ТРАНСПОРТНОЇ ЗАДАЧІ ЛАНЦЮГА ПОСТАЧАННЯ ПРОДУКТІВ У МЕРЕЖІ МАГАЗИНІВ

2.1 Обґрунтування вибору алгоритму

Розглянемо та порівняємо раніше згадані існуючі алгоритми для вирішення задачі комівояжера.

2.1.1 Алгоритм імітації відпалу (Simulated Annealing)

Алгоритм імітації відпалу (Simulated Annealing) – це стохастичний метод глобальної оптимізації, який імітує процес відпалу металу, коли його поступово охолоджують. Ідея полягає у тому, що початковий стан системи (наприклад, функція витрат) має високу енергію (високе значення функції витрат), і з часом знижується з температурою (через випадкові зміни в стані системи), що дозволяє системі «охолоджуватись» і прийти до більш оптимального стану.

Основні Кроки алгоритму імітації відпалу:

Крок 1. Вибір початкового стану системи (наприклад, випадковий вектор параметрів).

Крок 2. Встановлення початкової температури і коефіцієнта зниження температури.

Крок 3. Генерація випадкового стану системи (наприклад, змінюючи деякі параметри) і обчислення його енергії (функції витрат).

Крок 4. Якщо новий стан системи має меншу енергію, то він приймається як новий поточний стан системи.

Крок 5. Якщо новий стан системи має більшу енергію, то він приймається як новий поточний стан системи з деякою ймовірністю, яка зменшується зі зниженням температури (ця ймовірність дозволяє алгоритму вийти з локального мінімуму і продовжувати пошук).

Крок 6. Повторення Кроків 3-5 зі зменшенням температури до досягнення заданої максимальної кількості ітерацій або до досягнення заданої мінімальної температури.

Основним параметром алгоритму є коефіцієнт зниження температури, який впливає на швидкість збіжності алгоритму та його ефективність. Зазвичай коефіцієнт зниження температури зменшується з кожною ітерацією в залежності від заданого розкладу.

Переваги алгоритму імітації відпалу полягають у тому, що він дозволяє вирішувати задачі з великою кількістю локальних мінімумів та змінними функціями витрат. Крім того, він може бути застосований до задач, які не мають аналітичного розв'язку та задач з великою кількістю параметрів.

Кожний Крок алгоритму імітації відпалу має математичне обґрунтування.

Крок 1. Вибір початкового стану системи: нехай є деяка функція $f(x)$ від початкового стану системи x, яку необхідно мінімізувати. У цьому випадку x це перестановка вершин (міст) у тому порядку, який відповідає порядку їх відвідування, а $f(x)$ це довжина відповідного шляху.

Крок 2. Встановлення початкової температури і коефіцієнта зниження температури: візьмемо як базове рішення якийсь стан x_0. Базове рішення буде відповідати випадковій перестановці. Далі необхідно намагатися його покращувати.

Нехай T_0 – початкова температура, а α – коефіцієнт зниження температури.

Крок 3. Генерація випадкового стану системи: нехай y – випадковий стан системи, згенерований зміною деяких параметрів у x. Нехай $f(x)$ та $f(y)$ – енергія (функція вартості) x та y відповідно.

Крок 4. Якщо новий стан системи має меншу енергію, то він приймається як новий поточний стан системи: якщо $f(y) < f(x)$, покласти $x = y$.

Крок 5. Якщо новий стан системи має більшу енергію, то він приймається як новий поточний стан системи з деякою ймовірністю, яка зменшується зі

зниженням температури: якщо $f(y) \geq f(x)$, покласти $x = y$ з ймовірністю $P_{min}\left(1, e^{\frac{f(x)-f(y)}{T}}\right)$, де T – поточна температура.

Крок 6. Повторення Кроків 3-5 зі зменшенням температури до досягнення заданої максимальної кількості ітерацій або до досягнення заданої мінімальної температури.

Переваги алгоритму імітації відпалу полягають у тому, що він дозволяє вирішувати задачі з великою кількістю локальних мінімумів та змінними функціями витрат. Крім того, він може бути застосований до задач, які не мають аналітичного розв'язку та задач з великою кількістю параметрів.

Недоліки алгоритму імітації відпалу полягають у тому, що він може потребувати великої кількості ітерацій для досягнення оптимального розв'язку, особливо якщо початковий стан системи дуже віддалений від оптимального. Крім того, він може бути досить чутливим до значень параметрів, таких як початкова температура та коефіцієнт зниження температури, тому їх вибір може вплинути на якість розв'язку.

Загалом, алгоритм імітації відпалу є ефективним методом глобальної оптимізації для різноманітних задач, особливо для тих, де неможливо використати інші методи, такі як градієнтний спуск або метод Нелдера-Міда.

2.1.2 Алгоритм пошуку заборон (Tabu search))

Алгоритм пошуку заборон – це метод розвязання задачі комівояжера, який полягає в поступовому виключенні «заборонених» ребер з графа. Заборонені ребра – це ті, які гарантовано не входять у найдешевший цикл комівояжера, тому їх можна виключити з розгляду, що дозволяє зменшити кількість розв'язків, які потрібно перевірити.

Алгоритм пошуку заборон полягає у наступному:

– створити граф, який представляє задачу комівояжера;

– знайти мінімальне остовне дерево графа (MST) за допомогою алгоритму Пріма або Крускала;

– для кожного ребра, яке не входить до MST, знайти найкоротший шлях між його кінцями, використовуючи алгоритм Дейкстри або Флойда-Уоршелла;

– відсортувати ці ребра за зростанням довжини;

– поступово виключати найкоротші заборонені ребра з графа до тих пір, поки не буде знайдено найкоротший цикл комівояжера.

У більшості застосувань критерій спрямованості має вигляд: *a tabu m*, який доступний для вибору, якщо він може досягти рішення *sk EB m* краще, ніж найкраще рішення S^*, отримане до ітерації k, тобто, якщо

$$f(S^*) > f\left(S_k \bigoplus m\right). \tag{2.1}$$

Використання такого підходу, який штрафує за часто виконувані рухи, може бути реалізовано наступним чином. Спочатку підраховується, скільки разів виконується кожен хід m, щоб обчислити його частоту f_m. Потім пов’язується штрафний час з кожним ходом:

$$p \cdot m = \left\{ {}^{o}_{w \cdot f_m}. \right. \tag{2.2}$$

Якщо відповідає критерію спрямованості, інакше, де w – константа. Тоді значення ходу дорівнює:

$$f(x + m) - f(x) + p \cdot m. \tag{2.3}$$

Також існує метод для прискорення пошуку, який має назву «паралельна обробка». За еквівалентних витрат розподілені комп’ютери є потенційно

потужнішими, ніж послідовні. Часте одночасне обстеження різних переїздів із сусідства дозволяє досягти коефіцієнта прискорення, близького до ідеального.

$$\frac{T_n + T_{,,}}{\frac{T_n}{p} + T_{,,}}, \tag{2.4}$$

де p – кількість процесорів;

T_n – послідовний час для виконання «розпаралелення» обчислень;

$T_{,,}$ – час, пов'язаний з «непаралелізованою» частиною алгоритму.

Перевагою алгоритму пошуку заборон є те, що він дозволяє зменшити кількість можливих розв'язків задачі комівояжера, що дозволяє зменшити час розв'язання задачі. Крім того, алгоритм є досить простим у реалізації та добре працює на малих та середніх розмірах графів.

Недоліком алгоритму пошуку заборон є те, що він не є гарантованим способом знаходження оптимального розв'язку задачі комівояжера, оскільки не всі заборонені ребра можуть виявитися дійсно забороненими в остаточному циклі комівояжера. Також алгоритм може бути досить часовим до виконання на великих розмірах графів, оскільки для кожного незабороненого ребра потрібно знайти найкоротший шлях між його кінцями, що може займати досить багато часу, особливо якщо граф має багато вершин та ребер. Іншим недоліком алгоритму є те, що він не враховує можливість наявності петель у графі, тобто ребер, які з'єднують вершину з самою собою. У деяких випадках це може призвести до некоректного результату.

Отже, алгоритм пошуку заборон є одним з методів розв'язання задачі комівояжера, який дозволяє зменшити кількість можливих розв'язків та скоротити час розв'язання задачі на малих та середніх розмірах графів. Проте, він не є гарантованим способом знайдення оптимального розв'язку та може бути обмежений до виконання на великих розмірах графів.

2.1.3 Алгоритм оптимізації мурашиної колонії (Ant colony optimization)

Алгоритм оптимізації мурашиної колонії (Ant Colony Optimization, ACO) є метаевристичним алгоритмом, який був розроблений на основі спостережень про поведінку мурашок при пошуку харчових джерел. Основна ідея алгоритму полягає в тому, що мурахи, які шукають їжу, залишають позначки (феромони) на шляху від джерела їжі до мурашника. Інші мурахи використовують ці феромони як підказки для визначення напрямку до джерела їжі. У метафоричному сенсі, мурахи представляють розв'язки задачі, а шлях між джерелом їжі та мурашником – це простір можливих розв'язків. Феромони, які залишають мурахи, можна розглядати як показник якості розв'язків. ACO використовує цей принцип для знаходження оптимального розв'язку задачі. Алгоритм починається зі створення початкової мурашиного колонії, яка складається зі згенерованих випадковим чином розв'язків. Кожна мураха проходить шлях через граф розв'язків, залежно від ймовірності переходу з одного розв'язку до іншого, яка визначається на основі феромонів та евристичної інформації. Після проходження шляху, мураха оновлює феромони на цьому шляху відповідно до якості отриманого розв'язку. Згенеровані розв'язки з кращою якістю зберігаються в пам'яті, і використовуються для створення нової мурашиної колонії у наступному кроці. Алгоритм продовжується до досягнення заданої умови зупинки, такої як досягнення визначеної максимальної кількості ітерацій або до того моменту, коли знайдений розв'язок відповідає заданим критеріям якості. Під час роботи алгоритму важливо правильно налаштувати його параметри, такі як параметри ваги феромонів, параметри інтенсивності випаровування феромонів та параметри впливу евристичної інформації. Параметри ваги феромонів та інтенсивності випаровування феромонів можуть змінюватися в процесі роботи алгоритму, щоб забезпечити баланс між збереженням корисної інформації та уникненням передчасної збіжності [7].

Основа цього алгоритму полягає у тому, що на всіх вершинах, незважаючи на те, що це постачальник вантажу, клієнт, розташовуються мурахи, далі

починається рух мурах, вибір напряму руху мурах визначається максимальним, або мінімальним значенням, що приймає формула із параметрами ваги графу.

Вірогідність проходу маршрутом і:

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^{N} l_k^q \cdot f_k^p}, \quad (2.5)$$

де $l_i -$ параметр, зворотний вазі графа (довжина, час необхідний на подолання, затори, стан дорожнього покриття і т.д.);

f_i –кількість феромону на і-му перегоні;

$q -$ величина, що визначає «жадібність» алгоритму;

$p -$ величина, що визначає «стадність» алгоритму.

Також для правильності роботи алгоритму функція повинна відповідати таким властивостям:

$$q + p = 1, \quad (2.6)$$

$$0 \leq P_i \leq \sum_{k=0}^{N} l_k^q \cdot f_k^p, \quad (2.7)$$

$$P_i = \frac{A}{B}, A \leq B, \quad (2.8)$$

$$\sum_{i=0}^{k} P_i = 1. \quad (2.9)$$

Число маршрутів, що необхідно розрахувати зростає за фактеріальною залежністю відповідно до збільшення кількості пунктів. Відповідно до формули:

$$M = (n-1)!, \quad (2.10)$$

де $M -$ кількість маршрутів, необхідна кількість розрахунків на першому кроці;

$n -$ кількість пунктів в складеній групі.

До його переваг можна віднести: велику кількість параметрів ваги «феромонів», або графів, які можна змінювати, щоб найбільшим чином наблизитися до оптимального вирішення задачі, тобто має гнучкість до змін експлуатації транспортних засобів; алгоритм мурашиної колонії може змінювати свої параметри під час роботи, що дозволяє йому бути більш ефективним в різних умовах, надавати майже миттєвий зворотній зв'язок як для водіїв так і для диспетчерів.

Проте цей алгоритм може бути дорогим з обчислювальної точки зору, особливо, коли маємо справу з великомасштабними проблемами, через необхідність генерувати та оновлювати сліди феромонів. Також, його ефективність може значною мірою залежати від вибору таких параметрів, як кількість мурах, швидкість оновлення сліду феромону та початковий рівень феромону. Підібрати оптимальні значення цих параметрів може бути важко, що може вплинути на якість отриманого результату.

Отже, підбиваючи підсумки порівнянь, можна сказати, що обраний алгоритм, а саме генетичний, приваблює своєю можливістю роботи з великими об'ємами даних та можливістю їх паралельної обробки.

2.2 Загальні відомості про генетичні алгоритми

Як було зазначено в розділі 1, генетичний алгоритм – це евристичний алгоритм, який моделює процес природної еволюції, щоб розв'язати задачу оптимізації. Він використовує поняття генетики, такі як рекомбінація, мутація та селекція, для того, щоб створити нові покоління розв'язків та покращувати їх якість. Цей алгоритм може бути використаний для розв'язання різноманітних

задач оптимізації, включаючи задачу комівояжера (TSP), задачу розкладу (scheduling problem), задачу покриття (covering problem), та інші.

У генетичному алгоритмі використовуються наступні Кроки [22, 23]:

Крок 1. Ініціалізація: створення початкової популяції розв’язків.

Крок 2. Оцінка якості кожного розв’язку в популяції.

Крок 3. Селекція: відбір найкращих розв’язків з популяції для створення наступного покоління.

Крок 4. Рекомбінація: створення нових розв’язків шляхом комбінування даних з батьківських розв’язків.

Крок 5. Мутація – випадкове змінення окремих елементів розв’язків для збереження різноманітності.

Крок 6. Оцінка якості нових розв’язків (Генетична оцінка популяції).

Крок 7. Повторення – повторення Кроків 3-6 до досягнення кінцевої умови зупинки.

Застосування генетичного алгоритму може дати наближені розв’язки задачі оптимізації, які можуть бути досить близькі до оптимальних розв’язків, особливо для великих і складних задач. Враховуючи потужність та гнучкість алгоритму, він є ефективним засобом розв’язання складних задач оптимізації в різних сферах, включаючи науку, інженерію, фінанси, транспорт, логістику та інші. Таким чином, еволюцію штучної популяції – пошуку множини рішень деякої проблеми формально можна описати блок-схемою, що складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів. (рис. 2.1).

Рисунок 2.1 – Схема генетичного алгоритму

Розглянемо математичний апарат генетичного алгоритму. Кожний Крок генетичного алгоритму має математичне обґрунтування.

Крок 1. Ініціалізація:

$$P = \{C_1, C_2, \dots, C_k\}. \tag{2.11}$$

Кожна кожна хромосома C_i визначається як:

$$C_i = \{g_1, g_2, \dots, g_l\}, \tag{2.12}$$

де кожен ген g_j ініціалізується як g_j = random_value();

P – популяція розміром k;

l – довжина кожної хромосоми, а *random_value()* генерує випадкове значення для гена у вказаному діапазоні.

Крок 2. Оцінка:

$$F = \{f_1, f_2, \dots, f_k\}. \tag{2.13}$$

Крок 3. Селекція:

$$P' = \{C_1', C_2', \dots, C_k'\}, \tag{2.14}$$

де кожна вибрана хромосома C_i' визначається як C_i' = *select_chromosome(P, F)*;

P' – нова популяція;

select_chromosome(P, F) – функція, яка вибирає хромосому з популяції P на основі її придатності F.

Крок 4. Рекомбінація:

$$O = \{C_1'', C_2'', \dots, C_k''\}, \tag{2.15}$$

де кожна хромосома нащадка C_i'' визначається як C_i'' = *recombine*(C_i', $C'_{\mathrm{i+1\ mod\ N}}$);

O – множина хромосом нащадків, recombine(C_i, C_j) – функція, яка об'єднує генетичний матеріал батьківських хромосом C_i' і C_j' для створення нової хромосоми нащадка.

Крок 5. Мутація:

$$O' = \{C_1''', C_2''', \dots, C_k'''\}, \tag{2.16}$$

де кожна мутована хромосома нащадка C_i''' визначається як C_i''' = *mutate*(C_i'');

O' – множина мутованих хромосом нащадків, *mutate*(C_i) – функція, яка випадковим чином змінює генетичний матеріал хромосоми C_i для створення нової мутованої хромосоми.

Крок 6. Генетична оцінка популяції:

$$\mathrm{D} = \text{average\_distance}(\mathrm{P}), \tag{2.17}$$

де *average_distance(P)* визначається наступним чином:

$$D = \left(1/\left(\mathrm{N} * \frac{\mathrm{N}-1}{2}\right)\right) * \text{Sum}_{\{\mathrm{i}=1\}}^{\{\mathrm{N}-1\}} * \\ * \text{Sum\_\{j} = \mathrm{i} + 1\}\text{\^{}\{N\} distance}(\mathrm{C\_i}, \mathrm{C\_j}), \tag{2.18}$$

де D – середня відстань між усіма парами хромосом у популяції;

$distance(C\_i, C\_j)$ – функція, яка обчислює відстань між хромосомами C_i, C_j;

diversity – міра різноманітності популяції.

Однією з ключових переваг генетичного алгоритму є його здатність працювати зі складними та нелінійними функціями, які можуть мати багато локальних максимумів та мінімумів. Алгоритм може допомогти уникнути локальних максимумів та мінімумів та знайти глобальний максимум або мінімум.

Ще однією перевагою генетичного алгоритму є його здатність до паралельного обчислення [24]. Оскільки він працює з популяцією розв’язків, багато операцій можна виконувати паралельно на різних обчислювальних пристроях, що дозволяє прискорити процес оптимізації.

Проте, генетичний алгоритм також має свої недоліки. Наприклад, залежність від початкової популяції, недостатня ефективність у вирішенні задач зі складними обмеженнями та великою кількістю обчислень, а також недостатня здатність працювати з неперервними функціями. Усі ці фактори повинні бути враховані при виборі методу розв’язання задач оптимізації, включаючи генетичний алгоритм.

2.3 Опис операторів генетичного алгоритму

У одноточковому схрещуванні вибирається випадкова точка схрещування. Хвости двох батьків міняються місцями, щоб отримати нових нащадків [25] (рис. 2.2).

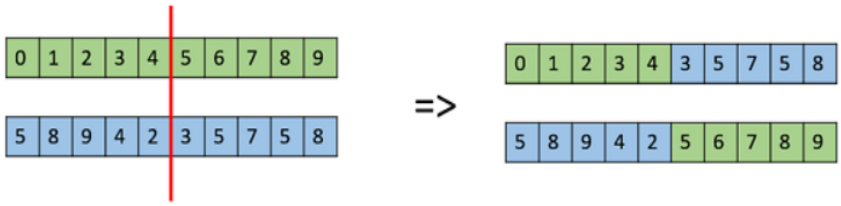

Рисунок 2.2 – Одноточковий кросовер

Ще один варіант це Partially Mapped Crossover (рис. 2.3), який власне і використовуємо в роботі. Він починається з вибору двох випадкових точок розрізу та виконує процедуру перетину за двома точками [26]. Згодом, у нових рішеннях, картування генів за межами розрізу виконується для усунення дублікатів.

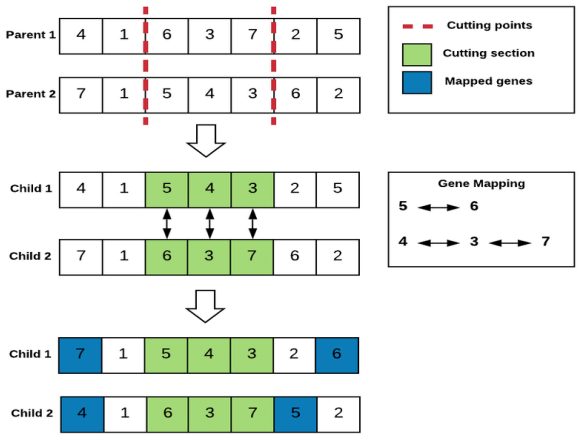

Рисунок 2.3 – Partially Mapped Crossover

Далі опишемо деякі з найбільш часто використовуваних операторів мутації [27, 28]. Подібно до операторів кросинговеру, це не вичерпний список, і розробник ГА може виявити більш корисною комбінацію цих підходів або оператор мутації, пов’язаний із проблемою.

Мутація перевороту біта. У цій мутації перевертання бітів вибираємо один або кілька випадкових бітів і перевертаємо їх. Це використовується для ГА у двійковому кодуванні (рис. 2.4).

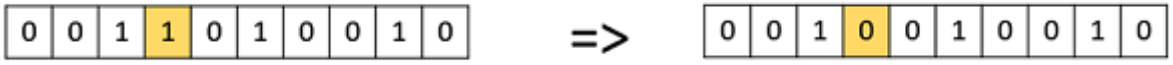

Рисунок 2.4 – Мутація перевороту біта

Випадкове скидання. Випадкове скидання є розширенням перевертання бітів для цілочисельного представлення. У цьому випадку випадково вибраному гену присвоюється випадкове значення з набору допустимих значень.

Мутація обміну. У мутації обміну навмання вибираємо дві позиції в хромосомі та міняємо їх значеннями. Це часто зустрічається в кодуваннях на основі перестановки (рис. 2.5).

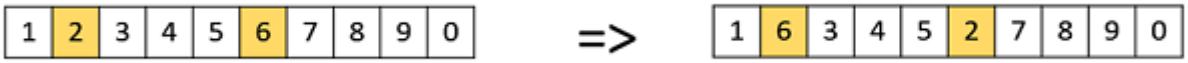

Рисунок 2.5 – Мутація обміну

Мутація Scramble. Мутація Scramble також популярна з представленнями перестановки[29, 30].

У цьому випадку з усієї хромосоми вибирається підмножина генів, а їхні значення перемішуються або перемішуються випадковим чином (рис. 2.6).

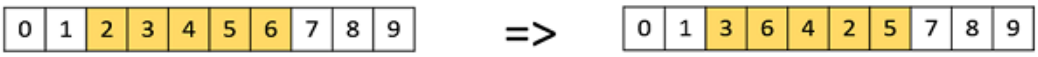

Рисунок 2.6 – Мутація Scramble

У інверсійній мутації вибираємо підмножину генів, як у мутації scramble, але замість того, щоб перетасувати підмножину, просто інвертуємо весь рядок у підмножині (рис. 2.7).

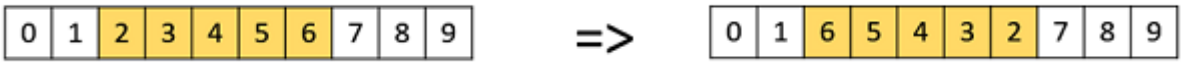

Рисунок 2.7 – Інверсійна мутація

2.4 Опис застосування генетичного алгоритму для задачі маршрутизації в ланцюгу постачань

Представимо першу популяцію як транспортну задачу із маршрутами відвідування магазинів створеної мережі (рис. 2.8). Далі транспортні засоби починають переміщення за маршрутами та відвідують магазини (рис. 2.9). Транспортні засоби повертаються до постачальника/складу (рис. 2.10). Далі вибір батьківських особин (рис. 2.11). І потім формування наступного покоління (рис. 2.12).

Рисунок 2.8 – Перша популяція

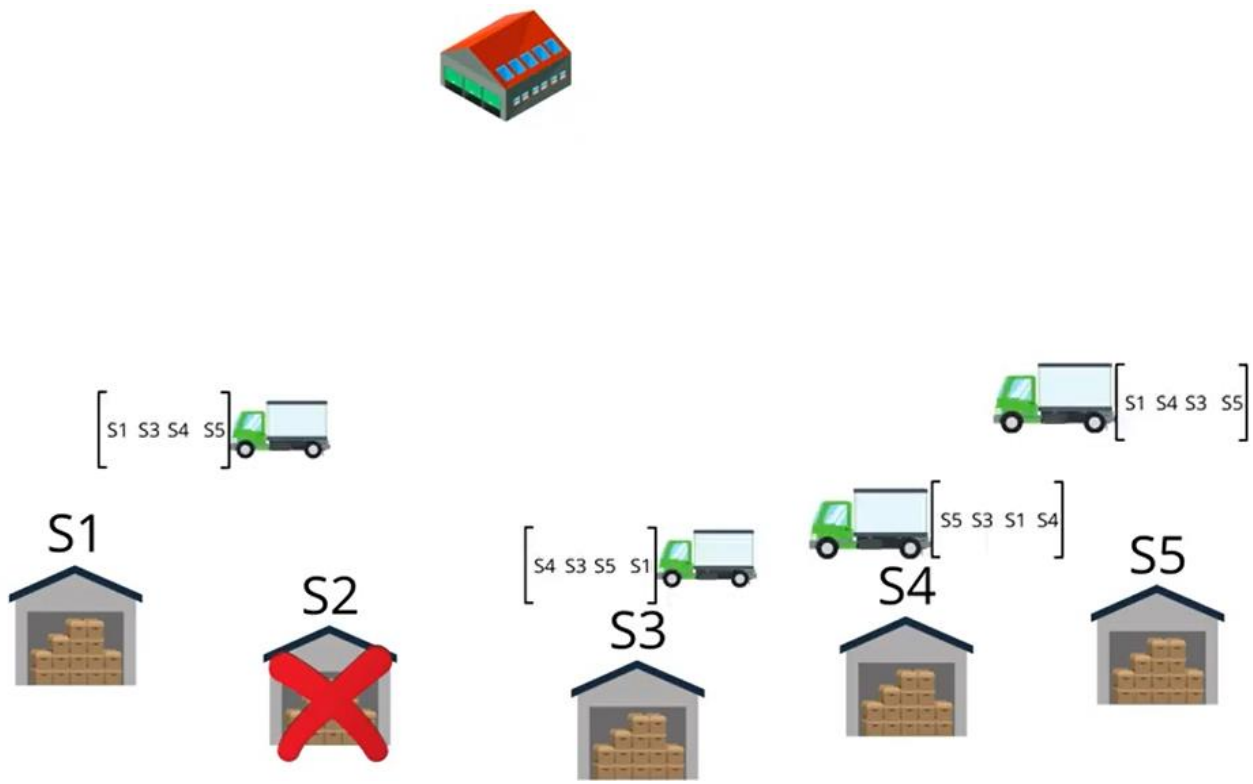

Рисунок 2.9 – Транспортні засоби рухаються за маршрутами відвідування магазинів

Рисунок 2.10 – Транспортні засоби повертаються до постачальника/складу

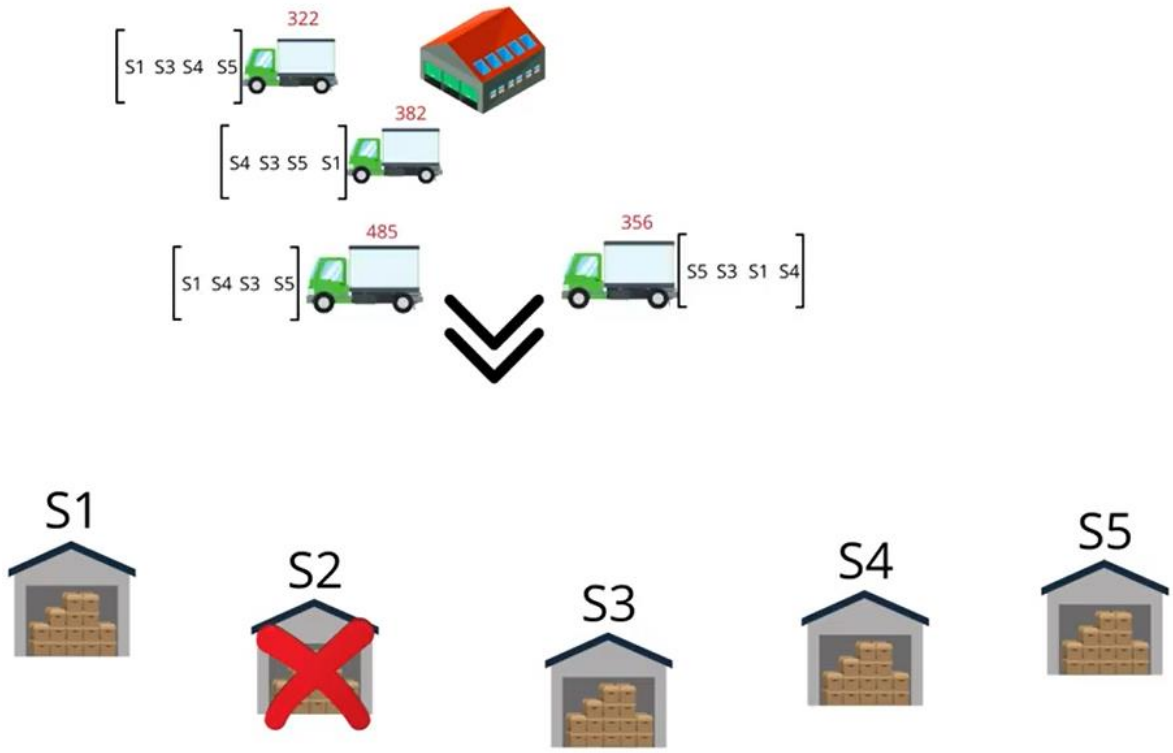

Рисунок 2.11 – Вибір батьків для схрещення

Рисунок 2.12 – Генерація потомків

3 РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ЛАНЦЮГА ПОСТАЧАННЯ ПРОДУКТІВ У МЕРЕЖІ МАГАЗИНІВ

3.1 Опис агентів моделі

Модель створюється як агентна імітаційна модель у системі AnyLogic. Взагалі вона включає наступних агентів: Main (головний агент), Manufacturer (виробник продукції), Supplier (магазини роздрібної мережі), Truck (транспортний засіб).

На рисунку 3.1 наведено структуру змінних та параметрів головного агенту Main. Він включає підмножини агентів Manufacturers (в цьому проєкті тільки один виробник продукції), Suppliers, Trucks. Уся множина магазинів поділяється на тих, які треба відвідати та ті, які не треба.

В якості параметрів моделі маємо наступні параметри генетичного алгоритму:

– mutationProbability (значення за замовчуванням 0,1);
– populationSize (значення за замовчуванням 1000);
– maxGenerations (значення за замовчуванням 25);
– crossOverProbability (значення за замовчуванням 0,8).

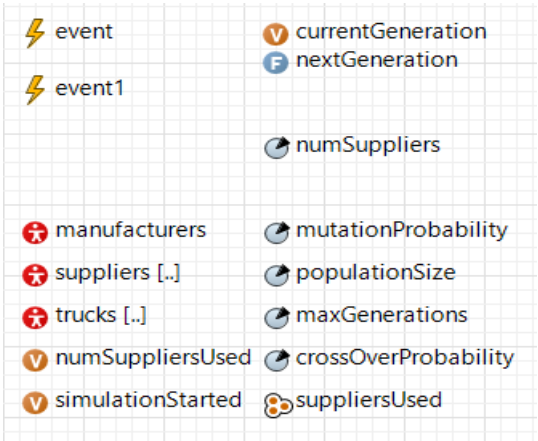

Рисунок 3.1 – Агент Main

Зображення, а також змінні агенту Manufacturer представлені на рисунку 3.2. Він містить колекцію магазинів/дистриб’юторів, що зв’язані з цим агентом виробника.

Зображення, а також змінні агенту Supplier представлені на рисунку 3.3. Він містить колекцію магазинів/дистриб’юторів, що зв’язані з цим агентом.

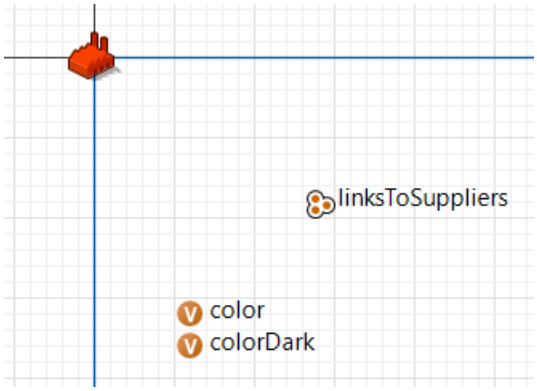

Рисунок 3.2 – Агент Manufacturer

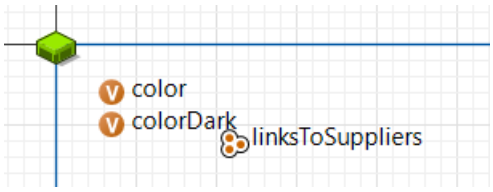

Рисунок 3.3 – Агент Supplier

Зображення, змінні, а також поведінка агенту Truck у вигляді діаграми станів представлені на рисунку 3.4.

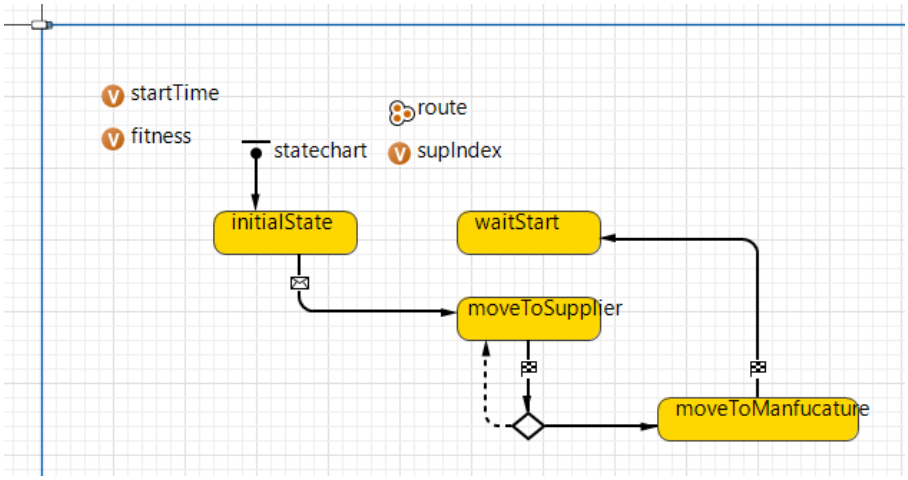

Рисунок 3.4 – Агент Truck

В якості змінних: startTime (для фіксації часу початку), fitness (значення фітнес-функції). Також він містить колекцію route – маршрути.

3.2 Опис поведінки агентів моделі

Розглянемо поведінку агенту Truck. У початковому стані фіксуємо час початку у змінній startTime (рис. 3.5).

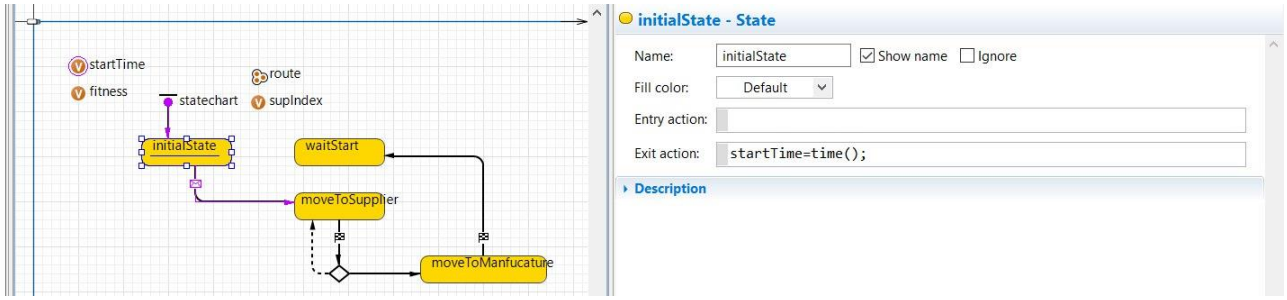

Рисунок 3.5 – Поведінка агенту Truck. Початковий стан

Перехід до стану moveToSupplier здійснюється при отриманні повідомлення «start» (рис. 3.6).

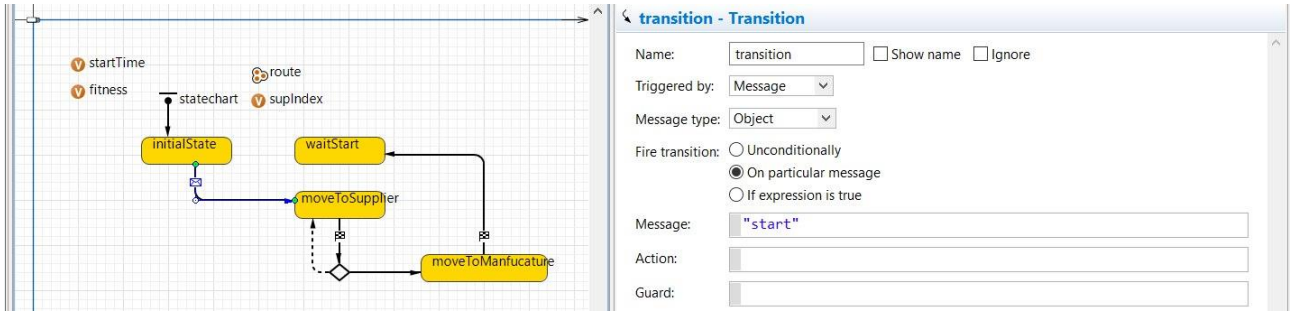

Рисунок 3.6 – Поведінка агенту Truck. Перехід до стану moveToSupplier

У стані moveToSupplier викликаємо метод moveTo вказуючи індекс чергового магазину, які беремо з колекції route (рис. 3.7).

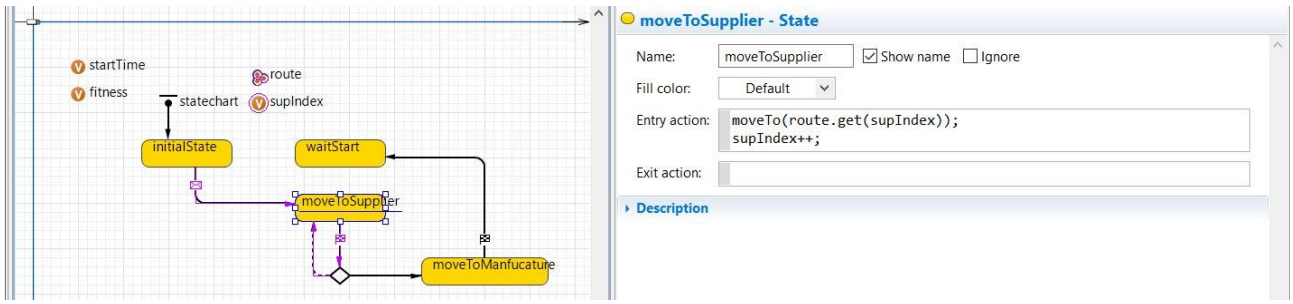

Рисунок 3.7 – Поведінка агенту Truck. Дії для стану moveToSupplier

Відповідно організуємо цикл за кількістю елементів у колекції route, що забезпечує послідовне переміщення транспортного засобу за маршрутом. Після відвідування останнього магазину (рис. 3.8) транспорт повертається до виробника (рис. 3.9).

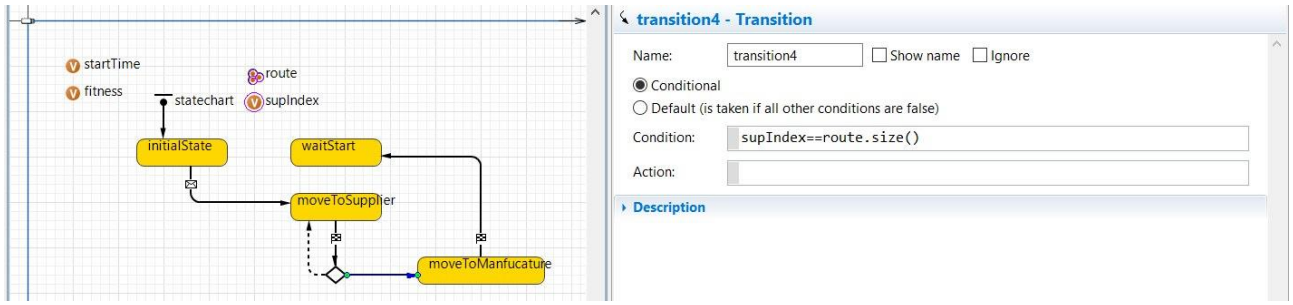

Рисунок 3.8 – Поведінка агенту Truck. Перехід до стану moveToManufacturer

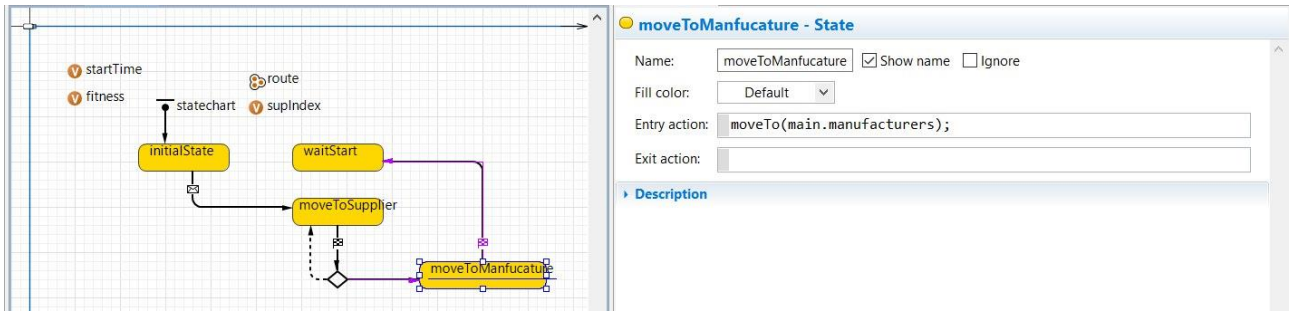

Рисунок 3.9 – Поведінка агенту Truck. Дії у стані moveToManufacturer

Після повернення транспортного засобу до виробника фіксуємо значення фітнес-функції, як час витрачений на рух за маршрутом (рис. 3.10).

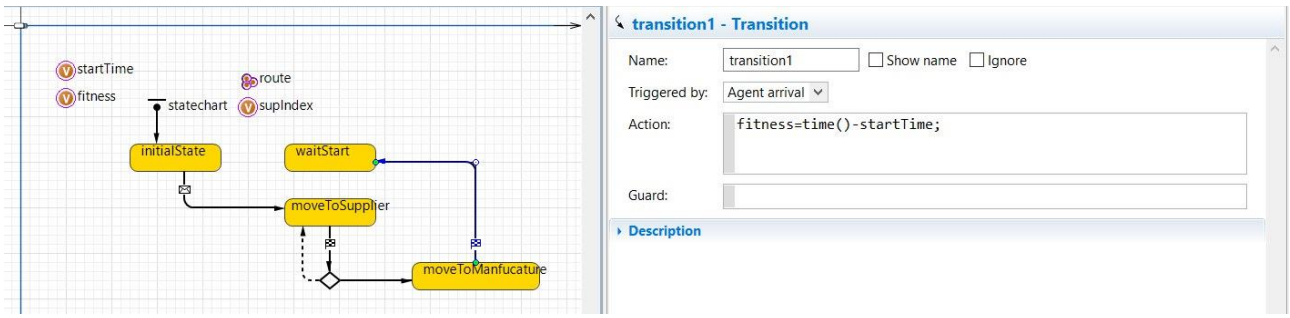

Рисунок 3.10 – Поведінка агенту Truck. Перехід у стан waitStart

Якщо залишаються ще транспортні засоби тоді запускаємо функцію nextGeneration, яка власне виконує дії генетичного алгоритму, щодо застосування операторів та генерації нової популяції (рис. 3.11).

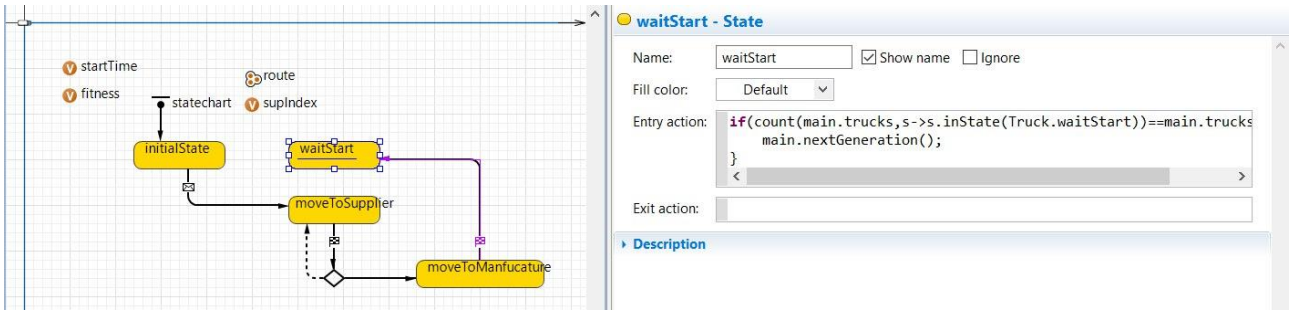

Рисунок 3.11 – Поведінка агенту Truck. Дії у стані waitStart

Далі розглянемо запуск агента Main за такими Кроками:

Крок 1. Формування та розміщення на карті колекції магазинів випадковим чином у визначеному заздалегіть регіоні (в даному випадку це полігон, що обведений та покриває місто Харків (Лістинг А.1).

Крок 2. Розміщення виробника/складу (Лістинг А.2).

Крок 3. Тепер власне розглянемо функцію формування нового покоління NextGeneration, що є реалізацією генетичного алгоритму.

Обираємо кращій за фітнес-функцією транспортний засіб, аналізуємо масив магазинів, що потребують відвідування. Встановлюємо властивість Visible для зв’язків (Лістинг А.3).

Крок 4. Аналізуємо маршрути та формуємо дані для відображення на графіку кращого та середнього значення фітнес-функції (Лістинг А.4).

Крок 5. Якщо це не остання популяція то відповідно випадковим чином обираємо з кращіх транспортних засобів пару батьків. Далі виконуємо кросовер і потім три варіанти мутації: мутація обміну, Scramble та інверсійна. Після цього формується нова популяція транспортних засобів та подається команда «start» і вони починають рухатись за новими маршрутами (Лістинг А.5).

3.3 Опис процесу моделювання

На рисунку 3.12 представлено шаблон екрану, що виступає як випробувальний стенд для моделі. Використовується елемент ГІС-карта, який налаштований на місто Харків. Сформований елемент ГІС-регіон, що покриває місто, саме в його межах. Генерується випадковим чином розташування магазинів. На головному екрані також присутні бігунки, що використовуються для налаштування параметрів генетичного алгоритму. Кнопка «Запустіть оптимізацію» призводить до запуску моделювання ланцюга постачань з використанням генетичного алгоритму.

Рисунок 3.12 – Загальний вигляд головного екрану

На рисунку 3.13 представлено те, як виглядає екран при запуску. При цьому формується початкове розташування складу виробника, магазинів чи дистриб’юторів.

Після натисканні кнопки «Запустіть оптимізацію» частина магазинів змінює своє зображення відповідно параметру «Кількість магазинів, що треба відвідати» (рис. 3.14).

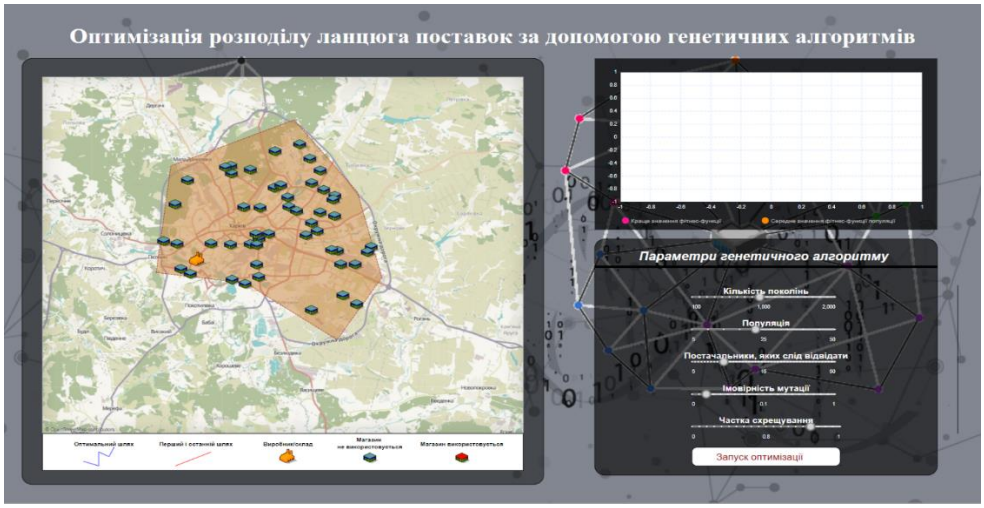

Рисунок 3.13 – Початкове розташування магазинів при запуску

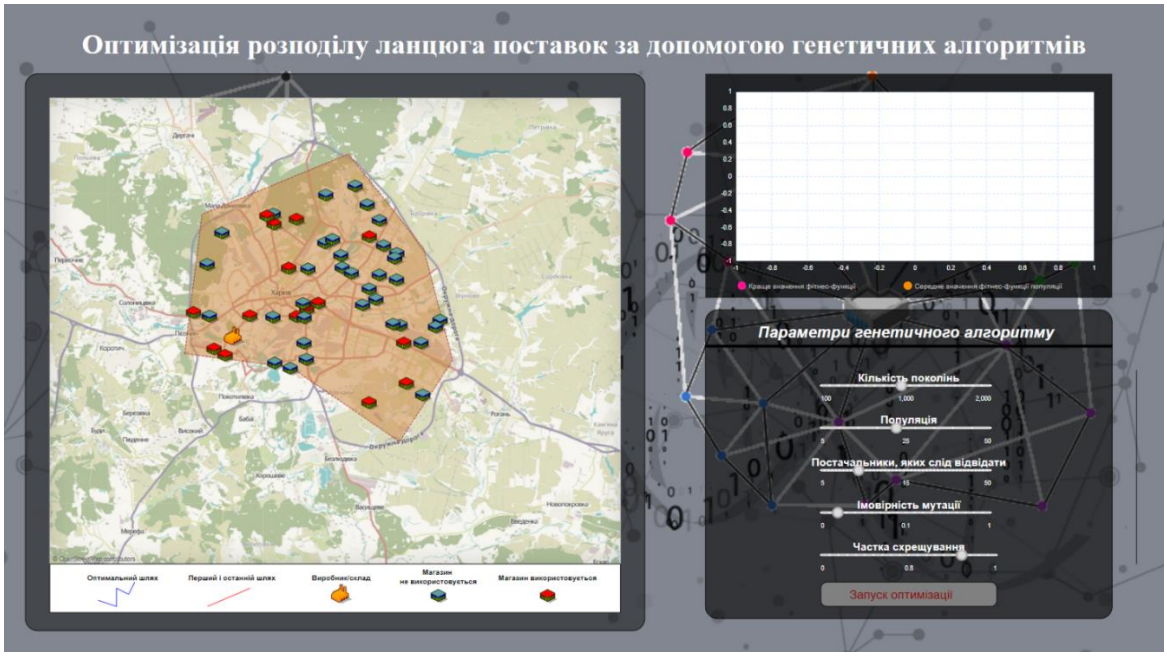

Рисунок 3.14 – Початок моделювання з визначенням магазинів, що треба відвідати

Далі здійснюється запуск транспортних засобів, які починають рухатись до магазинів (рис. 3.15). Після закінчення першого відвідування отримуємо маршрути, що відображаються на карті лініями синього кольору та перші значення фітнес-функцій, які відображаються на графіку (рис. 3.16).

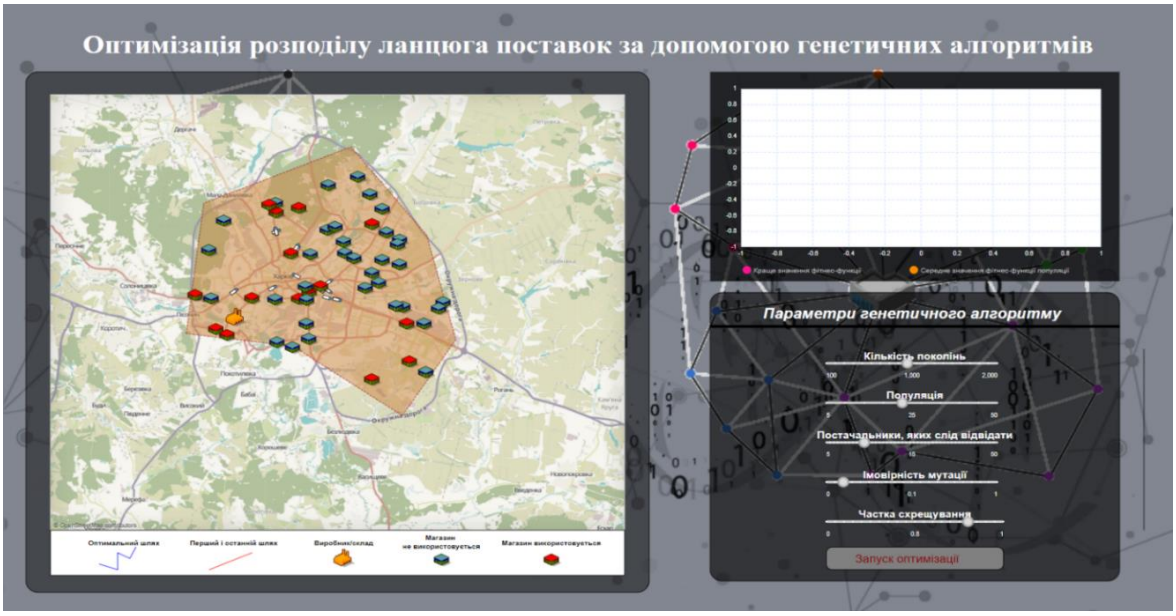

Рисунок 3.15 – Транспортні засоби рухаються до магазинів

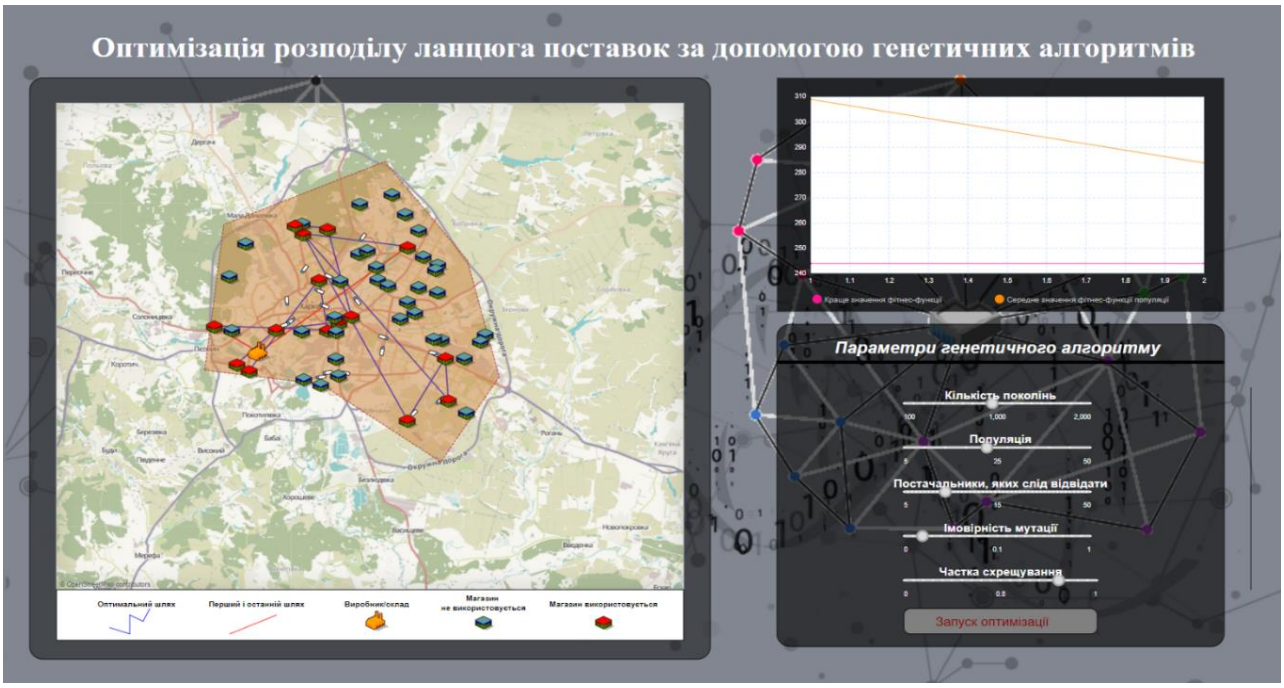

Рисунок 3.16 – Сформовані варіанти маршрутів та початкові значення фітнес-функції

Далі моделювання продовжується генерацією наступних поколінь тому на екрані можемо спостерігати змінення маршрутів та як покращується значення фітнес-функції, щодо загального часу відвідування всіх магазинів (рис. 3.17).

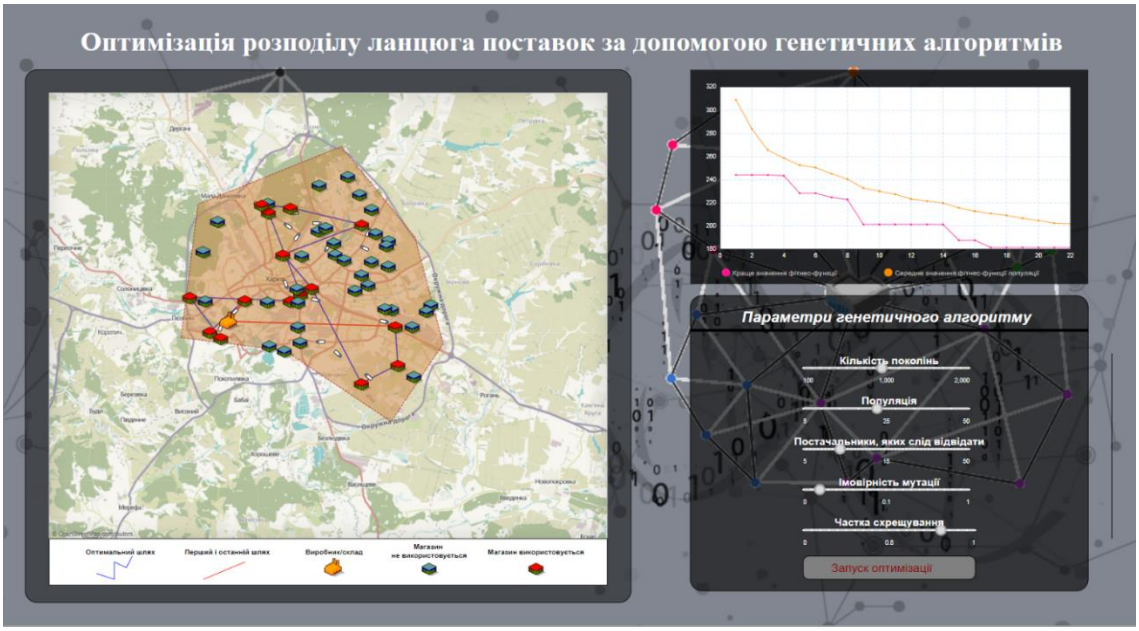

Рисунок 3.17 – Генерація поколінь та покращення фітнес-функції

На рисунку 3.18 бачимо кінцевий результат після завершення генерації заданої кількості поколінь.

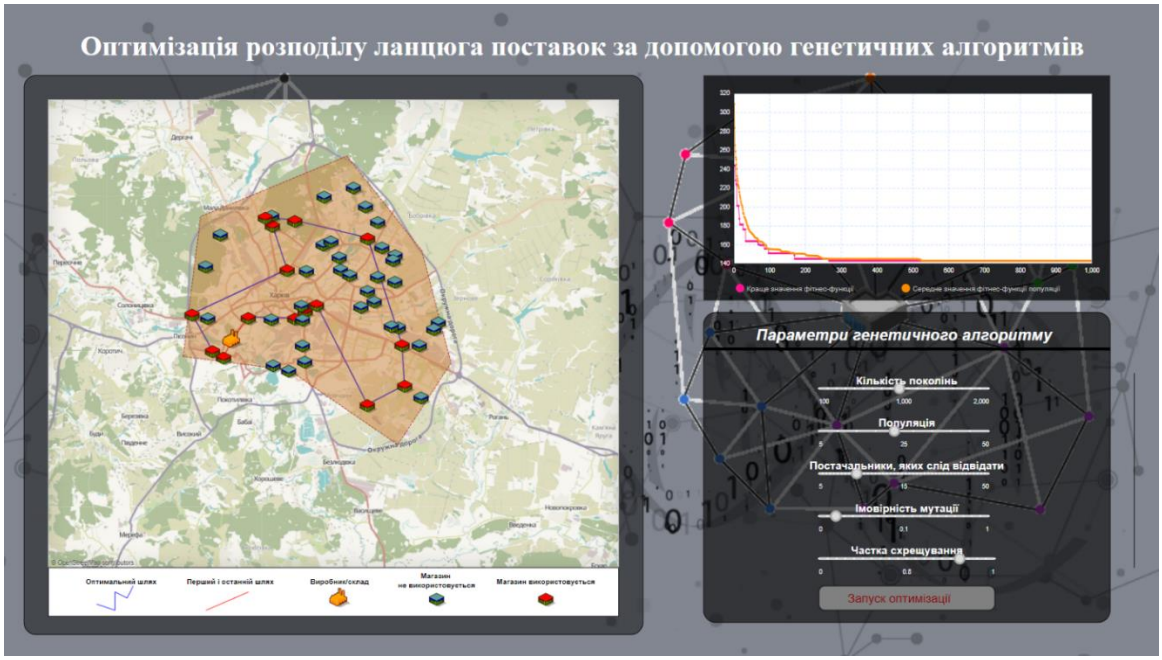

Рисунок 3.18 – Завершення генерації заданої кількості поколінь

На рисунку 3.19 бачимо сформований маршрут за результатами оптимізації за допомогою ГА.

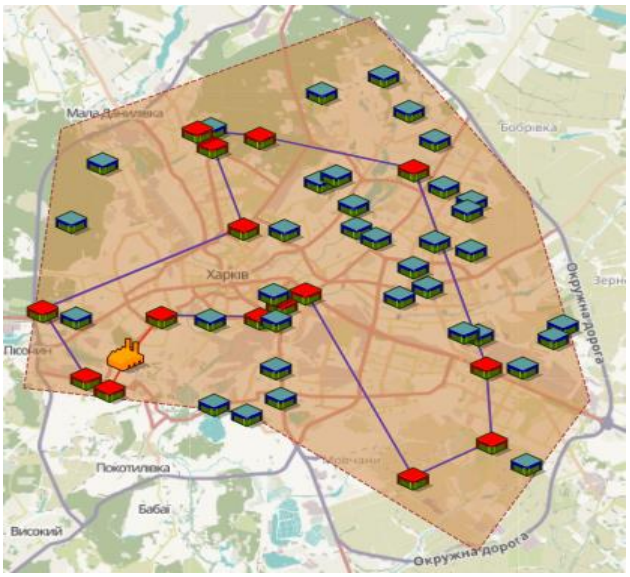

Рисунок 3.19 – Результуючий маршрут, знайдений ГА

На рисунку 3.20 представлений результуючий графік кращого та середнього значення фітнес-функції.

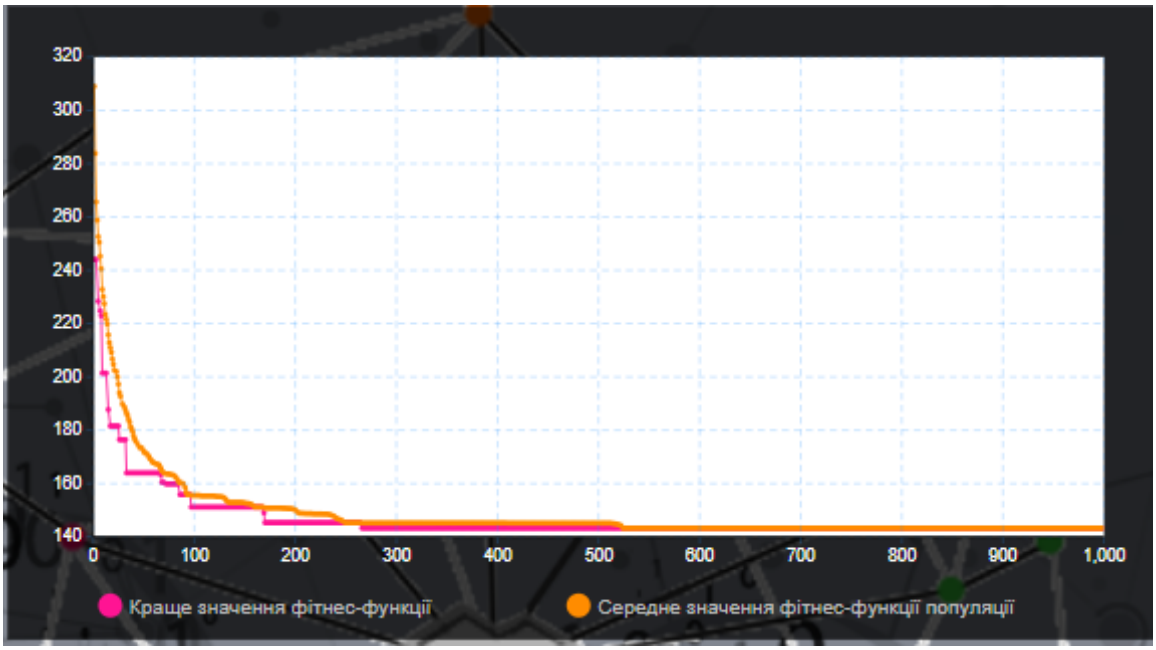

Рисунок 3.20 – Результуючий графік кращого та середнього значення фітнес-функції при частці схрещування 0,8

Змінюємо частку схрещення з 0,8 на 0,5 та проводимо моделювання. На рисунку 3.21 представлений результуючий графік кращого та середнього значення фітнес-функції для значення частки схрещення 0,5. Бачимо, що така зміна призводить до зменшення сходимості алгоритму.

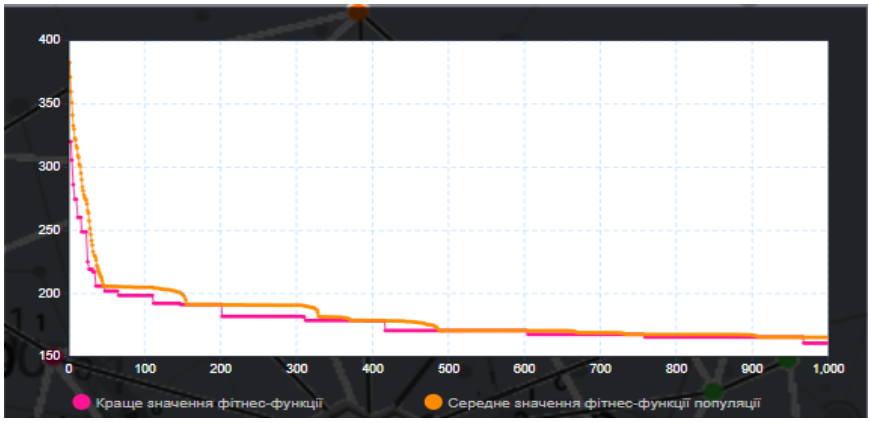

Рисунок 3.21 – Результуючий графік кращого та середнього значення фітнес-функції при частки схрещування 0,5

3.4 Опис експериментів

Збільшимо кількість магазинів для відвідування (рис. 3.22).

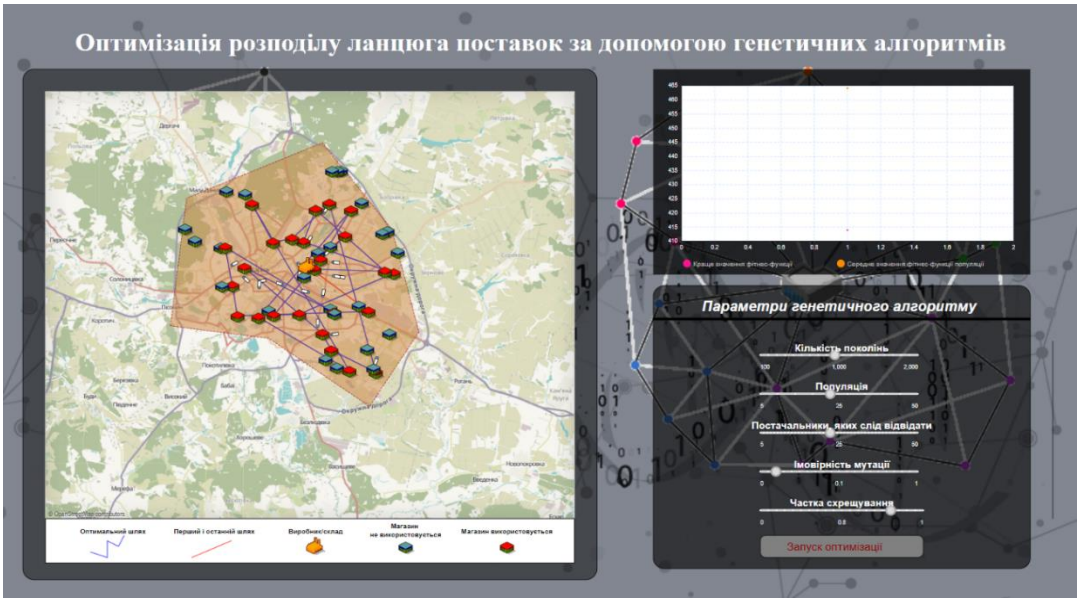

Рисунок 3.22 – Збільшуємо кількість магазинів для відвідування

Результат моделювання представлений на рисунку 3.23. Моделювання пошуку маршруту для 25 магазинів потребувало вдвічі більшого часу.

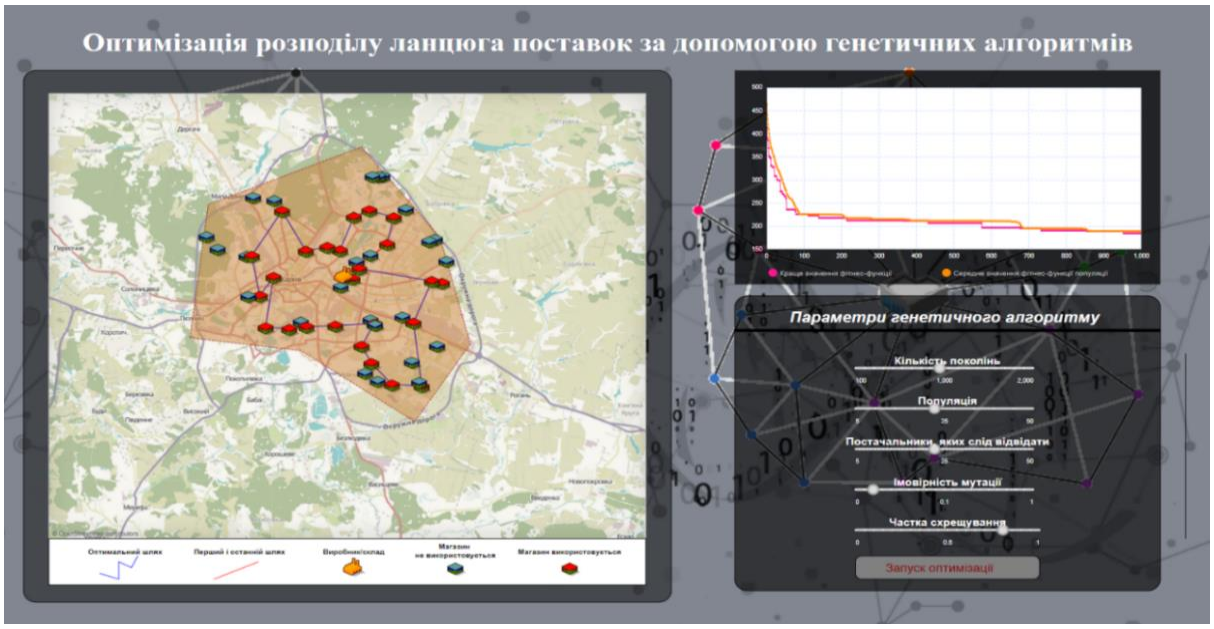

Рисунок 3.23 – Маршрут та графік зміни фітнес-функції для збільшеної кількості магазинів для відвідування

На рисунку 3.24 обираємо всі магазини для відвідування.

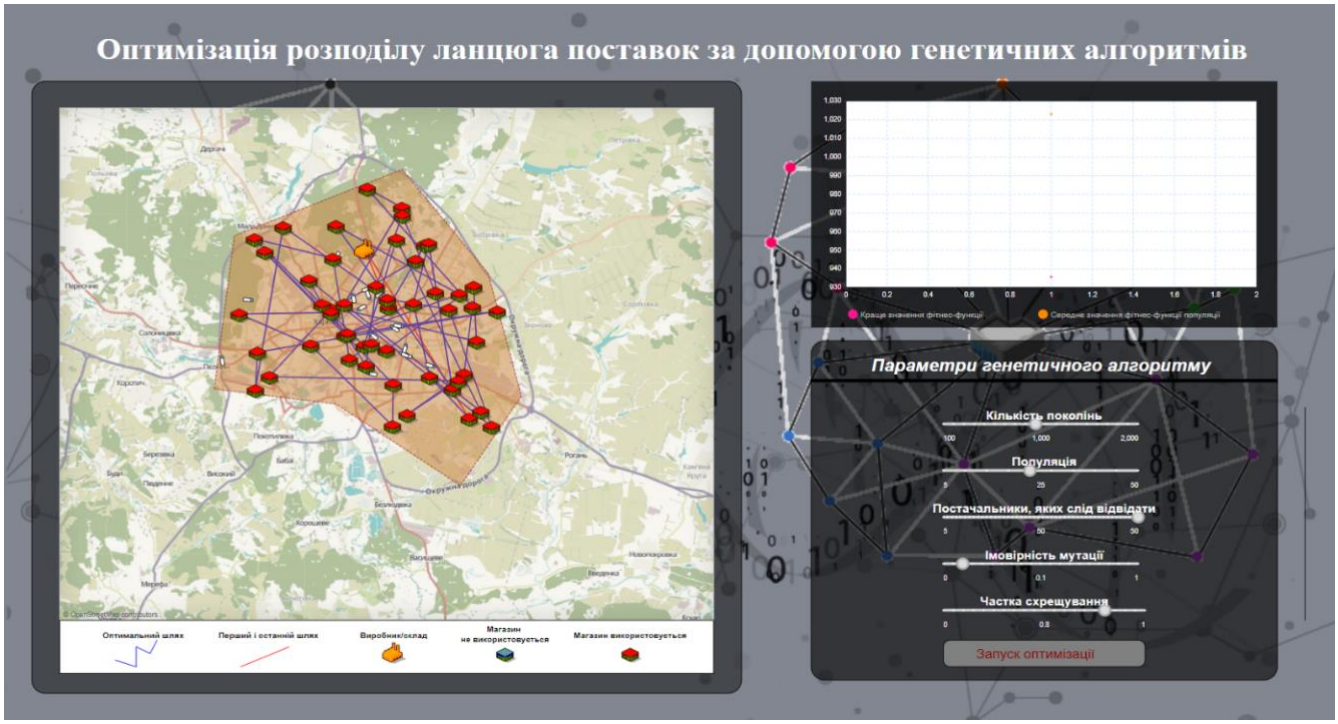

Рисунок 3.24 – Обираємо всі магазини для відвідування

Результат моделювання представлений на рисунку 3.25. Моделювання потребувало часу, що в 6 разів більше ніж при моделювання з параметром кількості магазинів 15 (табл. 3.1).

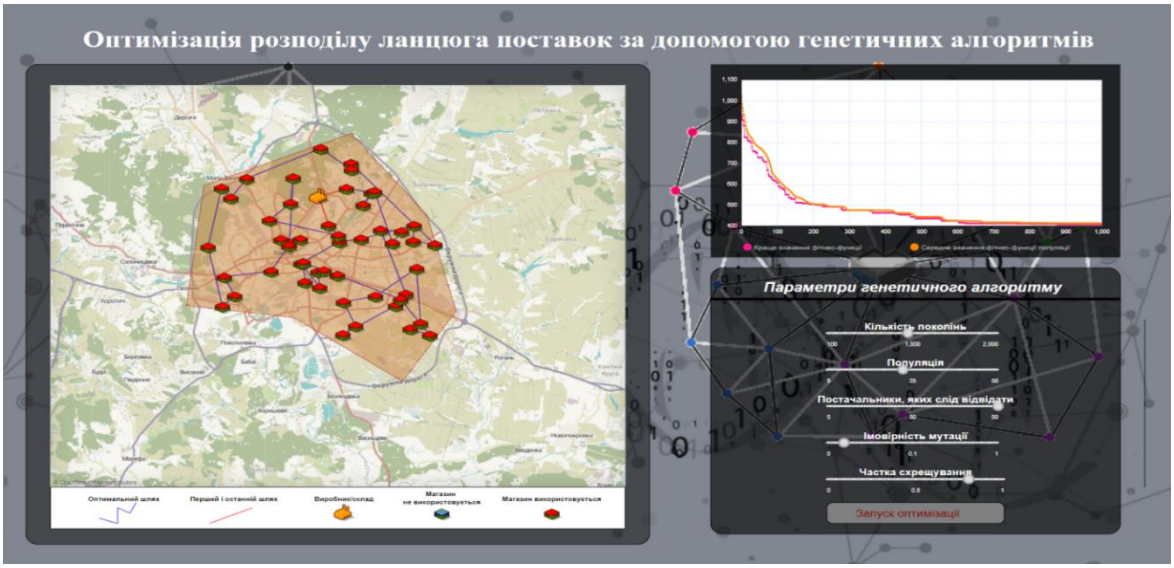

Рисунок 3.25 – Маршрут та графік зміни фітнес-функції для макисмальної кількості магазинів для відвідування

Таблиця 3.1 – Експерименти для різної кількості магазинів для відвідування

№	Кількість поколінь	Популяція	Магазини, які слід відвідати	Імовірність мутації	Частка схрещування	Покоління де було знайдено останнє покращення фітнес-функції	Загальний час оптимізації
1	1000	25	15	0,1	0,8	224	5 хв 5 с
2	1000	25	25	0,1	0,8	953	9 хв 10 с
3	1000	25	50	0,1	0,8	916	36 хв 30 с

Експерименти з варіюванням параметру імовірності мутації наведено у таблиці 3.2. Значного покращення в ефективності не спостерігається.

Таблиця 3.2 – Експерименти з варіюванням імовірності мутації

№	Кількість поколінь	Популяція	Магазини, які слід відвідати	Імовірність мутації	Частка схрещування	Покоління де було знайдено останнє покращення фітнес-функції	Загальний час оптимізації
1	1000	25	15	0,1	0,8	224	5 хв 5 с
2	1000	25	15	0,2	0,8	210	5 хв 20 с
3	1000	25	15	0,3	0,8	454	6 хв
4	1000	25	15	0,4	0,8	490	5 хв 40 с
5	1000	25	15	0,5	0,8	361	5 хв 50 с

Експерименти з варіюванням параметру частки схрещування наведено у таблиці 3.3. Результати демонструють погіршення сходимості алгоритму.

Таблиця 3.3 – Експерименти з варіюванням частки схрещування

№	Кількість поколінь	Популяція	Магазини, які слід відвідати	Імовірність мутації	Частка схрещування	Покоління де було знайдено останнє покращення фітнес-функції	Загальний час оптимізації
1	1000	25	15	0,1	0,8	224	5 хв 5 с
2	1000	25	15	0,1	0,5	883	5 хв
3	1000	25	15	0,1	0,3	980	4 хв 55 с

ВИСНОВКИ

Розроблено імітаційну модель в середовищі Anylogic для оптимізації транспортних перевезень в мережі продуктових магазинів міста Харкова на основі генетичних алгоритмів.

При виконанні роботи було виконано наступні задачі:

– проведено порівняльний аналіз методів реалізації транспортної задачі. Встановлено, що використання еволюційних алгоритмів, до яких відноситься й генетичний алгоритм, дозволяють знайти адекватні за часом виконання рішення для задач з великою кількістю клієнтів;

– здійснено детальний аналіз складових генетичного алгоритму;

– побудовано генетичні оператори, що відображають специфіку розв’язуваної задачі ланцюга постачання у мережі магазинів;

– реалізовано імітаційну модель для оптимізації транспортних перевезень в мережі магазинів міста на основі генетичних алгоритмів;

– проведено експерименти.

Розглянута система дозволяє тестувати алгоритм з різними параметрами кількості популяцій, частки схрещування, імовірності мутації. Встановлено суттєвий вплив розміру початкової популяції на якість кінцевого рішення. Отримані експериментальні результати показують, що розроблений алгоритм впорався з вирішенням поставленої задачі маршрутизації транспортних засобів в ланцюгу постачання товарів в мережі магазинів/дистриб’юторів.

Використано вбудовані сервіси ГІС в системі AnyLogic на базі OpenStreetMap, тому на відміну від багатьох інших проєктів, в цьому будуть маршрути на реальній дорожній інфраструктурі міста. Це важлива відмінність.

Магазини в межах означеного регіону також формуються випадковим чином відповідно налаштованому параметру кількості. Знов таки, якщо потрібно можемо вказати статичну структуру розподільної мережі або зчитувати їх наприклад з бази даних. При цьому це можна зробити наступним чином: здійснюється надсилання адресу на онлайн-сервер OpenStreetMap, отримання

відповідних координат із сервера та розміщення магазинів на карті в точках із отриманими координатами.

Було отримано базову модель, яка є достатньо гнучка, щоб в подальшому можливо було деталізувати та дообробляти у різних напрямах. Навіть перейти до моделювання мультимодальних перевезень в межах країни або врахувати особливості військової логістики та транспортування вантажів в зону бойових дій, що зараз є одним з актуальних та важливих завдань.

Результати роботи апробовано у вигляді тез доповіді під час XXVII Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [31].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko, I. S., & Kramarenko, O. O. (2019). Software determination of the optimal route by geoinformation technologies. Radio Electronics, Computer Science, Control, (3), 131-142.

2. Творошенко, І. С., Мгеброва, В. Р., & Білий, В. В. (2016). Практичні аспекти створення вихідної інформації для проведення геоінформаційного аналізу у сфері управління нерухомістю.

3. Творошенко І.С., Крамаренко О.О. (2017). Особливості застосування геоінформаційних технологій під час розробки мережі просторових об'єктів оперативного пожежогасіння в місті Харкові. Геоінформаційні технології у територіальному управлінні та експертних дослідженнях: правові, організаційні, технічні проблеми: тези доповідей IV Міжнародної науковопрактичної конференції. С. 159–163.

4. Творошенко, І. С., & Табашник, В. А. (2018). Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків.

5. Творошенко, І. С., & Подласенко, Є. П. (2019). Дослідження методу розпізнавання геоінформаційних ситуацій в системах моніторингу територій.

6. Li, S. B., Wang, G. M., Wang, T., & Ren, H. L. (2017). Research on the method of traffic organization and optimization based on dynamic traffic flow model. *Discrete Dynamics in Nature and Society*.

7. Wei, Y., Chen, F., & Wang, H. (2018). Inventory and production dynamics in a discrete-time vendor-managed inventory supply chain system. *Discrete Dynamics in Nature and Society*.

8. Crainic, T. G., Perboli, G., & Rosano, M. (2018). Simulation of intermodal freight transportation systems: a taxonomy. *European Journal of Operational Research*, 270(2), 401-418.

9. Noorizadegan, M., & Chen, B. (2018). Vehicle routing with probabilistic capacity constraints. *European Journal of Operational Research*, 270(2), 544-555.

10. Hu, W., Wang, H., Qiu, Z., Yan, L., Nie, C., & Du, B. (2018). An urban traffic simulation model for traffic congestion predicting and avoiding. *Neural Computing and Applications*, 30, 1769-1781.

11. Mulloorakam, A. T., & Nidhiry, N. M. (2019). Combined objective optimization for vehicle routing using genetic algorithm. *Materials Today: Proceedings*, 11, 891-902.

12. Soroush Fatemi-Anaraki, Mahdi Mokhtarzadeh, Masoud Rabbani, Dorsa Abdolhamidi. (2022) A hybrid of K-means and genetic algorithm to solve a bi-objective green delivery and pick-up problem. *Journal of Industrial and Production Engineering* 39:2, 146-157.

13. Бондаренко П. В. (2022). Підвищення ефективності системи планування та оптимізації транспортних перевезень на основі генетичного методу. Державний університет телекомунікацій. Магістерська робота. С. 29-30.

14. Schermer, D., Moeini, M., & Wendt, O. (2019). A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, *109*, 134-158.

15. Lee, C. (2021). An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. *Journal of the Operational Research Society*, *72*(7), 1461-1485.

16. Zhou, X. C., YIv, C., & He, H. (2020). Multi-site green vehicle path model and optimization algorithm considering time-varying speeds. *Control and Decision*, *37*(2), 473-482.

17. XueJing, J. I., & Xu, Y. O. N. G. (2019). Application of Genetic Algorithm in Logistics Path Optimization. *Academic Journal of Computing & Information Science*, 2(1).

18. Saxena, R., Jain, M., Malhotra, K., & Vasa, K. D. (2020). An optimized openmp-based genetic algorithm solution to vehicle routing problem. In Smart Computing Paradigms: New Progresses and Challenges: Proceedings of ICACNI 2018, Volume 2 (pp. 237-245). Springer Singapore.

19. Sbai, I., & Krichen, S. (2020). A real-time decision support system for big data analytic: A case of dynamic vehicle routing problems. *Procedia Computer Science*, 176, 938-947.

20. Ongcunaruk, W., Ongkunaruk, P., & Janssens, G. K. (2021). Genetic algorithm for a delivery problem with mixed time windows. *Computers & Industrial Engineering*, 159, 107478.

21. Euchi, J., & Sadok, A. (2021). Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication*, 44, 101236.

22. Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., & Banaszak, Z. (2021). Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. Neurocomputing, 423, 670-678.

23. Iswari, T., & Asih, A. M. S. (2018, April). Comparing genetic algorithm and particle swarm optimization for solving capacitated vehicle routing problem. In IOP Conference Series: *Materials Science and Engineering* (Vol. 337, No. 1, p. 012004). IOP Publishing.

24. Ochelska-Mierzejewska, J., Poniszewska-Marańda, A., & Marańda, W. (2021). Selected Genetic Algorithms for Vehicle Routing Problem Solving. Electronics, 10(24), 3147. *MDPI AG*.

25. Ibrahim, Muhammad Faisal & Putri, M.M & Farista, D & Utama, Dana. (2021). An Improved Genetic Algorithm for Vehicle Routing Problem Pick-up and Delivery with Time Windows. *Jurnal Teknik Industri*. 22. 1-17.

26. Altinoz, M., Altinoz, O.T. (2023). Multiobjective problem modeling of the capacitated vehicle routing problem with urgency in a pandemic period. *Neural Comput & Applic* 35, 3865–3882.

27. Becerra-Rozas, M., Cisternas-Caneo, F., Crawford, B., Soto, R., García, J., Astorga, G., & Palma, W. (2022). Embedded Learning Approaches in the Whale Optimizer to Solve Coverage Combinatorial Problems. *Mathematics*, 10(23), 4529. MDPI AG.

28. Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist?. *Swarm and Evolutionary Computation*, *54*, 100671.

29. Fernández Gil A., Lalla-Ruiz E., Sánchez M.G., Castro C. (2022). A Review of Heuristics and Hybrid Methods for Green Vehicle Routing Problems considering Emissions. *Journal of Advanced Transportation*, vol. 2022, Article ID 5714991

30. Chen, C.-M., Lv, S., Ning, J., & Wu, J. M.-T. (2023). A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem. *Symmetry*, 15(1), 124. MDPI AG.

31. Ясько О. С. (2023) Алгоритм мурашиної колонії для вирішення транспортних задач із виходом з локальних мінімумів. 27-ий міжнародний молодіжний форум «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ», С. 222-223.