

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки

(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління

(назва кафедри)

"На правах рукопису"

УДК 004.9+519.87

«До захисту допущено»

В.о.завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ ” 20 19 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю 126 Інформаційні системи та технології

(код та назва спеціальності)

ОПП

Інформаційні управляючі системи та технології

(код та назва спеціалізації)

на тему: Інформаційна система планування обслуговування доріг міста

Виконала: студентка

VI курсу групи ІС-82мп

(шифр групи)

Григорець Ганна Олександрівна

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

доц., к.т.н., доц. Жданова О.Г.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

д.т.н., проф. Томашевський В.М.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

доц., к.т.н., доц. Репнікова Н.Б.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти другий (магістерський) за освітньо-професійною програмою

Спеціальність 126 Інформаційні системи та технології
(код і назва)

ОПП Інформаційні управляючі системи та технології
(код і назва)

ЗАТВЕРДЖУЮ
В.о.завідувача кафедри
О.А.Павлов
(підпис) (ініціали, прізвище)
«__» _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту
Григорець Ганні Олександрівні

(прізвище, ім'я, по батькові)

1. Тема дисертації Інформаційна система планування обслуговування доріг міста

науковий керівник дисертації Жданова Олена Григорівна, к.т.н., доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 27 ” 11 20 19 р. № 4005-с

2. Строк подання студентом дисертації “ 2 ” 12 20 19 р.

3. Об'єкт дослідження Логістичні процеси обслуговування доріг

4. Перелік завдань, які потрібно розробити дослідити предметне середовище та особливості його функціонування; провести аналіз відомих результатів розв'язання задачі маршрутизації транспортних засобів по дугам; сформулювати постановку задачі MM k-CCPP with DC; розробити модифікований алгоритм сформульованої задачі та дослідити його ефективність; розробити інформаційну систему планування обслуговування доріг міста.

5. Орієнтовний перелік ілюстративного матеріалу 1. Схема структурна варіантів використань; 2. Класифікація задач маршрутизації транспортних засобів; 3. Змістовна постановка задачі MM k-CCPP with DC; 4. Математична модель задачі MM k-CCPP with DC; 5. Схема структурна алгоритму розв'язання задачі MM k-CCPP with DC; 6. Схема структурна класів інформаційної системи; 7. Екранні форми.

6. Орієнтовний перелік публікацій Три публікації: одна стаття у науковому виданні, двос тез доповідей на науково-практичних конференціях.

7. Консультанти розділів дисертації

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |

8. Дата видачі завдання “ 2 ” вересня 20 19 р.

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Строк виконання етапів магістерської дисертації | Примітка |
|-------|---|---|----------|
| 1 | <i>Систематизація результатів огляду літератури</i> | 28.10 | |
| 2 | <i>Порівняльний аналіз існуючих методів розв'язання задачі</i> | 03.11 | |
| 3 | <i>Постановка та формалізація математичної моделі задачі</i> | 04.11 | |
| 4 | <i>Модифікація існуючих методів розв'язання задачі</i> | 06.11 | |
| 5 | <i>Розробка інформаційного та програмного забезпечення</i> | 11.11 | |
| 7 | <i>Проведення експериментальних досліджень розробленого алгоритму</i> | 15.11 | |
| 8 | <i>Оформлення документації</i> | 19.11 | |
| 9 | <i>Подання роботи на попередній захист</i> | 20.11 | |
| 10 | <i>Подання роботи на основний захист</i> | 02.12 | |
| | | | |

Студент

_____ (підпис)

Г.О. Григорець

_____ (ініціали, прізвище)

Науковий керівник

_____ (підпис)

О.Г. Жданова

_____ (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 112 с., 12 рис., 20 табл., 1 додаток, 36 джерел.

Актуальність. Проведені у Великобританії дослідження показали, що у вартості продукту, який потрапив до кінцевого споживача, до 70% складають витрати, що так чи інакше пов'язані з логістичними операціями. З огляду на це обслуговування доріг стає необхідним процесом, що спрямований на підтримання належного технічного стану доріг та надання можливості швидкого та безпечного переміщення транспортних засобів задля уникнення аварійних ситуацій і т. ін. Отже, обслуговування доріг потребує якісного планування маршрутів для спеціалізованої дорожньої техніки.

Математичне формулювання цієї задачі відоме як задача маршрутизації по дугам (Arc Routing Problem), що є підмножиною задач маршрутизації транспортних засобів (Vehicle Routing Problem). Існує багато розроблених математичних моделей вищезазначеної задачі, проте врахування додаткових умов часто призводить до створення нової моделі і відповідно пошуку нового або модифікації існуючого методу розв'язання. Робота присвячена дослідженню та розробці методу розв'язання задачі ARP.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів» (№ ДР 0117U000919).

Мета і завдання дослідження. Метою є підвищення якості процесів обслуговування доріг міста за рахунок мінімізації часу проходження транспортними засобами запланованих шляхів сполучень (із врахуванням обмежень на директивні терміни виконання робіт та місткість машин).

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- дослідити предметне середовище та особливості його функціонування;

- провести аналіз відомих результатів розв’язання задачі маршрутизації транспортних засобів по дугам;
- сформулювати постановку задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин;
- розробити модифікований алгоритм сформульованої задачі та дослідити його ефективність;
- розробити інформаційну систему планування обслуговування доріг міста.

Об’єкт дослідження – логістичні процеси обслуговування доріг.

Предмет дослідження – задачі побудови маршрутів транспортних засобів, що орієнтовані на проходження шляхів сполучень та враховують директивні терміни виконання робіт та місткість машин.

Методи дослідження, застосовані в роботі, базуються на методах дослідження операцій, евристичних та метаевристичних алгоритмах.

Наукова новизна отриманих результатів полягає у модифікації та використанні в складі інформаційної системи алгоритму розв’язання задачі маршрутизації по дугам.

Публікації. Матеріали роботи опубліковані у науковому журналі «Paradigm of Knowledge» та тезах доповіді Тринадцятої міжнародної науково-практичної конференції «Математичне та імітаційне моделювання систем. МОДС 2018» та III Всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019).

МАРШРУТИЗАЦІЯ ТРАНСПОРТНИХ ЗАСОБІВ, МАРШРУТИЗАЦІЯ ПО ДУГАМ, VRP, ARP, ПЛАНУВАННЯ, ІНФОРМАЦІЙНА СИСТЕМА, ОБСЛУГОВУВАННЯ ДОРІГ, ОПТИМІЗАЦІЯ, ЗАДАЧА КИТАЙСЬКОГО ЛИСТОНОШІ, ССРР, МІНІМАКС, ДИРЕКТИВНИЙ ТЕРМІН

ABSTRACT

Master dissertation: 112 pp., 12 fig., 20 tab., 1 app., 36 sources.

The relevance. Researches in the UK have shown that up to 70% of the cost of a product that reaches the end consumer are the costs associated with logistics operations in one way or another. In view of this, road maintenance becomes a necessary process aimed to maintain the good condition of roads and enabling the rapid and safe movement of vehicles to avoid emergencies, etc. Therefore, road maintenance requires quality route planning for specialized road technics.

The mathematical model of this problem is known as the Arc Routing Problem, which is a subset of Vehicle Routing Problems. There are many mathematical models developed for the above problem, but taking into account additional conditions often leads to the creation of a new model and, accordingly, finding a new one or modifying an existing method of solving. The work is devoted to researching and developing the algorithm for solving the ARP problem.

Relationship of work with scientific programs, plans, themes. The work was done at the department of computer-aided management and data processing systems of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» within the theme «Effective methods for solving problems of scheduling theory» (№ DR 0117U000919).

Purpose and objects of the research. The goal of the research is to improve the quality of road maintenance processes in the city districts by minimizing the transit time of all routes and taking into account restrictions on car capacity and deadline classes of the work.

The following tasks need be solved to achieve these tasks:

- to investigate the subject environment and peculiarities of its functioning;
- to analyze the known results of solving the arc routing problem;
- create the formulation of the Min-Max Capacitated k -Chinese Postman Problem with Deadline Classes;
- to develop the modified algorithm of the formulated problem and to investigate its effectiveness;

- to develop the information system for city road maintenance planning.

The object of the research – logistic processes of road maintenance.

The subject of the research – vehicle routing problem, which is oriented on the passing of routes by vehicles and takes into account the deadline classes of the work and the capacity of cars.

Methods of the research, used in the paper, are based on operations research methods, heuristic and metaheuristic algorithms.

Scientific novelty of the results is the modification of the algorithm for solving the arc routing problem and the use of this algorithm within the information system.

Publications. The results of the research were published in the scientific journal «Paradigm of Knowledge», at the materials of the XIII International scientific and practical conference «Mathematical Modeling and Simulation of Systems. MODS 2018» and III Ukrainian scientific and practical conference of young scientists and students "Information Systems and Management Technologies" (ISTU-2019).

VEHICLE ROUTING PROBLEM, ARC ROUTING PROBLEM, VRP, ARP, PLANNING, INFORMATION SYSTEM, ROAD MAINTENANCE, OPTIMIZATION, CHINESE POSTMAN PROBLEM, CCPP, MIN-MAX, DEADLINE CLASSES

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ | 10 |
| ВСТУП..... | 11 |
| 1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ОБСЛУГОВУВАННЯ ДОРІГ МІСТА | 13 |
| 1.1 Опис бізнес – процесів | 13 |
| 1.1.1 Опис процесу діяльності..... | 13 |
| 1.1.2 Актори і функції | 17 |
| 1.1.3 Структура бізнес-процесів..... | 18 |
| 1.2 Опис постановки задачі..... | 27 |
| 1.3 Рішення з інформаційного забезпечення..... | 28 |
| Висновки | 31 |
| 2 МОДЕЛІ ТА МЕТОДИ ПЛАНУВАННЯ ОБСЛУГОВУВАННЯ ДОРІГ МІСТА... | 32 |
| 2.1 Аналіз задач маршрутизації транспортних засобів по дугам..... | 32 |
| 2.1.1 Визначення та застосування задачі ARP..... | 32 |
| 2.1.2 Основні поняття..... | 34 |
| 2.1.3 Розвиток та класифікація задач маршрутизації транспортних засобів по дугам | 36 |
| 2.1.4 Задача китайського листоноші (Chinese Postman Problem)..... | 38 |
| 2.1.5 Ієрархічна задача китайського листоноші (Hierarchical Chinese Postman Problem) | 42 |
| 2.1.6 Задача k -китайських листонош (k -Chinese Postman Problem)..... | 44 |
| 2.1.7. Задача мінімаксу k -китайських листонош (Min-Max k -Chinese Postman Problem) | 44 |
| 2.1.8 Задача мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (Min-Max Capacitated k -Chinese Postman Problem with Deadline Classes)..... | 45 |
| 2.2 Змістовна постановка задачі | 45 |
| 2.3 Математична модель | 47 |
| 2.4 Огляд методів розв'язання..... | 50 |

| | |
|--|------------|
| 2.4.1 Алгоритм розв’язання задачі китайського листоноші з пріоритетами (CPP with priorities) | 50 |
| 2.4.2 Алгоритми розв’язання задачі k -китайських листонош (k -CPP) | 51 |
| 2.4.3 Алгоритми розв’язання мінімаксу задачі k -китайських листонош (MM k -CPP) | 53 |
| 2.5 Розробка алгоритму розв’язання MM k -CCPP with DC | 55 |
| 2.5.1 Опис алгоритму | 55 |
| 2.5.2 Алгоритм Дейкстри | 61 |
| 2.6 Результати досліджень ефективності алгоритму | 63 |
| Висновки | 66 |
| 3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ | 67 |
| 3.1 Засоби розробки | 67 |
| 3.2 Архітектура програмного забезпечення | 69 |
| 3.2.1 Структура класів програмного забезпечення | 69 |
| 3.2.2 Специфікація функцій програмного забезпечення | 70 |
| 3.3 Інструкція користувача | 73 |
| Висновки | 81 |
| 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ | 82 |
| 4.1 Опис ідеї стартап-проекту | 82 |
| 4.2 Технологічний аудит ідеї стартап-проекту | 85 |
| 4.3 Аналіз ринкових можливостей запуску стартап-проекту | 86 |
| 4.4 Розроблення ринкової стратегії проекту | 90 |
| 4.5 Розроблення маркетингової програми стартап-проекту | 93 |
| Висновки | 97 |
| ВИСНОВКИ | 98 |
| РЕКОМЕНДАЦІЇ | 100 |
| ПЕРЕЛІК ПОСИЛАНЬ | 101 |
| ДОДАТОК А | 105 |
| Схема структурна варіантів використань | 106 |
| Класифікація задач маршрутизації транспортних засобів | 107 |

| | |
|---|-----|
| Змістовна постановка задачі MM k -CCPP with DC..... | 108 |
| Математична модель задачі MM k -CCPP with DC..... | 109 |
| Схема структурна алгоритму розв'язання задачі MM k -CCPP with DC..... | 110 |
| Схема структурна класів інформаційної системи..... | 111 |
| Екранні форми | 112 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

VRP – англ. Vehicle Routing Problem – задача маршрутизації транспортних засобів.

NRP – англ. Node Routing Problem – задача маршрутизації по вершинам.

ARP – англ. Arc Routing Problem – задача маршрутизації по дугам.

CPP – англ. Chinese Postman Problem – задача китайського листоноші.

CCPP – англ. Capacitated Chinese Postman Problem – задача китайського листоноші з врахуванням місткості.

RPP – англ. Rural Postman Problem – задача сільського листоноші.

MM – англ. Min-Max – умова мінімаксу.

DC – англ. Deadline Classes – класи директивних термінів.

ВСТУП

Задачі оптимізації маршрутів транспортних засобів є ключовими не лише в областях транспортних перевезень і логістики. В якісному плануванні маршрутів зацікавлені як поштові служби, так і муніципальні та приватні підприємства, що займаються вивезенням сміття чи обслуговуванням вулиць міста.

Утримання доріг є трудомістким процесом, що потребує залучання спеціалізованої дорожньої техніки, якою потрібно продуктивно розпоряджатися. Обслуговування доріг включає в себе великий обсяг дорожніх робіт, які спрямовані на огляд, ремонт і облаштування дорожнього покриття за встановленим графіком. Час виконання кожного виду дорожніх робіт різниться та залежить від їх складності. Так у зимовий період обслуговування доріг складається з комплексу робіт, спрямованих на підтримку безпечного і нормального стану дорожнього покриття, своєчасного очищення від льоду та снігу. У випадку надзвичайних ситуацій, таких як заметіль, відновлення нормального транспортного сполучення у найшвидші терміни стає першочерговою задачею. І від якісного її вирішення прямопропорційно залежить зменшення збитків.

В загальному міські вулиці потребують щоденного очищення від сміття, бруду та пилу, ретельного огляду всіх елементів, а також нанесення або відновлення зношеної розмітки. Тож, задачі оптимізації маршрутів транспортних засобів мають величезне практичне значення у сфері муніципального обслуговування. На сьогоднішній день планування обслуговування доріг спеціалізованою дорожньою технікою переважно здійснюється вручну і в першу чергу залежить від знань водіїв та покладається на їх розсуд. Тому задача ефективного і при цьому економічно вигідного визначення маршрутів для транспортних засобів постає актуальною перед відповідальними службами чи підприємствами.

Задача маршрутизації транспортних засобів (VRP) вперше була сформульована у 50-х роках ХХ століття. Такі задачі можна поділити на дві категорії, а саме на задачі маршрутизації по вершинах (NRP) та на задачі маршрутизації по дугам (ARP). Основна різниця між цими задачами стосується того, де розміщені вимоги на обслуговування. Так у NRP вимоги ставляться до вершин графа, а отже, метою є

пройти через всі вершини. У ARP, на відміну від NRP, вимоги ставляться до дуг графу, а мета – пройти всі дуги. Тому NRP і ARP повинні бути представлені різними моделями, а алгоритми для однієї категорії не можуть бути безпосередньо застосовані до іншої [1]. Оскільки у даній роботі розглядаються випадки, коли вимоги відносяться до автомобільних доріг, що можуть біти представлені дугами графу, для дослідження обрано саме ARP.

За роки досліджень, з огляду на складність та комплексність прикладних завдань, виникло багато різновидів задачі маршрутизації транспортних засобів по дугам, починаючи від класичної задачі про китайського листоношу (1962), запропонованої китайським вченим Кваном Мей-Ко (Kwan Mei-Ko) [2]. Над вирішення задачі маршрутизації транспортних засобів працювало багато вчених і особливе місце серед праць посідає книга «Arc Routing», створена під редакцією Моше Дрора (Moshe Dror) [3], що є фундаментальною основою для дослідження задач ARP.

З огляду на те, що задача ARP – NP-складна задача комбінаторної оптимізації, точний розв’язок може бути отриманий лише для прикладів малої розмірності. Тому на практиці широко застосовуються евристичні та метаевристичні методи. Вони хоч і не гарантують оптимальності розв’язку, але за певних умов дозволяють отримати хороші допустимі рішення.

Не заважаючи на існування великої кількості задач класу ARP та методів їх розв’язання, все ще існує ціла низка проблем. Реальні системи представляють собою велику кількість факторів та параметрів, які треба враховувати при створенні математичної моделі розроблюваної системи. Додавання хоча б одного обмеження змінює математичну модель і, як наслідок, алгоритм її розв’язання.

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ОБСЛУГОВУВАННЯ ДОРІГ МІСТА

1.1 Опис бізнес – процесів

1.1.1 Опис процесу діяльності

Дороги – це транспортна інфраструктура, яка відіграє важливу роль у підтримці економіки регіону, тому необхідно виконувати роботи по їхньому обслуговуванню, щоб дороги продовжували оптимально функціонувати. Ефективний принцип утримання доріг з отриманням при цьому економічної вигоди може забезпечити стабільні дорожні умови навіть попри мінімальний бюджет [4].

Обслуговування доріг – це комплекс профілактичних робіт по утриманню доріг, який здійснюється впродовж року в залежності від нагальних потреб, природніх умов, пори року та ін., задля підтримання їхнього транспортно-експлуатаційного стану. Тобто обслуговування доріг є узагальненим поняттям, що включає в себе різні види робіт, які постійно виконуються на дорогах і на які можуть накладатися певні обмеження. Планування обслуговування доріг – важка задача оптимізації, яка пов'язана зі складною маршрутизацією транспортних засобів.

Самі дороги поділяються за їхньою значимістю. Основні транспортні шляхи, аварійні маршрути та головні дороги є більш важливими в процесі надання обслуговування, в той час як житлові та службові мають другорядну важливість та обслуговуються після більш пріоритетних. Також потрібно враховувати ту обставину, що в результаті низки причин на дорогах можуть утворюватися пробки. Автомобільним затором (дорожньої пробкою) є накопичення транспортних засобів на певній ділянці дороги, які рухаються набагато повільніше, ніж звична швидкість на цій дорозі [5]. Це може стати суттєвою перешкодою при обслуговуванні доріг. Існує декілька основних причин утворення дорожніх заторів:

- стоянки автомобілів, які припарковані на узбіччі проїжджої частини;
- перекриття проїжджої частини дороги;
- дорожньо-транспортні пригоди;

- несприятливі умови для руху транспортних засобів (наприклад, погодні умови);
- неузгодженість роботи світлофорів;
- порушення правил перестроювання зі смуги в смугу та ін.

Також причиною заторів можуть стати дорожні роботи, що виконуються в години пік. Тому при складанні плану обслуговування слід враховувати часові обмеження для різних категорій доріг.

Рівень завантаженості дорожнього руху визначається 10-ти бальною шкалою, запропонованою сервісом «Яндекс.Пробки» [6]:

- а) 1 – дороги вільні;
- б) 2 – дороги майже вільні;
- в) 3-4 – місцями ускладнення руху;
- г) 5 – рух щільний;
- д) 6 – рух ускладнений;
- е) 7 – серйозні затори;
- ж) 8 – багатокілометрові затори;
- з) 9 – місто стоїть;
- и) 10 – пішки швидше.

В середньому 1-2 бали відповідають руху вночі або у вихідні, 4-5 балів – руху в робочий час, затори вище 7 балів свідчать про недоцільність поїздки на автотранспорті.

Скарги громадськості щодо якості виконання дорожніх робіт є поширеними. Проведення деякого виду робіт може бути дуже важливим і дорогим питанням у певних місцях та у певний час. Наприклад, останнім часом щорічно кількість снігу значно варіюється в різних регіонах. Отже, попит на прибирання снігу дуже різниться. Висновок полягає в тому, що не можна очікувати використання планів минулого року. Нові ситуації вимагають нових планів, які потребують ефективних інструментів планування [7].

Підсумовуючи все вище наведене щодо складності планування маршрутів можна зазначити, що для виконання планування краще використовувати

автоматизовану систему, якою є інформаційна система планування обслуговування доріг міста, що має досить невеликий час роботи при вирішенні проблеми, враховує головні потреби галузі та забезпечує можливість використання онлайн.

Робота з інформаційною системою планування обслуговування доріг заснована на взаємодії з географічною картою певної ділянки, що потребує обслуговування. В загальному процес діяльності при використанні системи відбувається наступним чином:

- а) користувач вказує на карті місце розташування депо;
- б) користувач обирає на карті дороги, вказує їхні класи пріоритетів та додає їх до загального списку доріг для обслуговування;
- в) користувач вказує кількість та місткість спеціалізованої дорожньої техніки, що буде використовуватися в процесі обслуговування;
- г) користувач вказує вхідну інформацію стосовно запланованих робіт та встановлює часові обмеження на завершення обслуговування для кожного класу пріоритетів;
- д) система намагається скласти план маршрутів з врахуванням введених даних:
 - 1) якщо побудувати допустимі маршрути неможливо, то система повертає користувача на крок г;
 - 2) якщо маршрути були успішно побудовані, то система надає користувачу посилання на план робіт, після чого користувач може переглянути його або одразу перейти на крок е;
- е) система починає симуляцію руху дорожньої техніки на карті відповідно до розроблених маршрутів;
- ж) система періодично оновлює дані про завантаженість руху на дорогах:
 - 1) якщо для певної дороги рівень завантаженості дорожнього руху не перевищує 6 балів, то транспортний засіб продовжує рух по даній дорозі;

- 2) якщо для певної дороги рівень завантаженості дорожнього руху становить 6 і вище балів, то система намагається скорегувати маршрут в обхід даної дороги.

Виключенням з кроку ж2 є дороги для обслуговування, оскільки виконання робіт на цих дорогах має відбутися в будь-якому випадку, тому до них доведеться повертатися пізніше впродовж маршрутів, а це може стати причиною додаткових збитків та порушення виконання складеного плану.

Слід зазначити, що система окремо складає план для кожного району міста, а не для всього міста в цілому. Тобто передбачається, що заздалегідь відомо які дороги повинно обслуговувати те чи інше депо зі спеціалізованою дорожньою технікою, що йому належить (вважається, що в одному районі обслуговування може бути лише одне депо). Також система не розглядає ніякої можливості взаємодії між районами міста. Це в загальному відповідає тому, як на сьогоднішній момент відбуваються ці процеси на практиці, а також тому, як ця проблема розглядається в інших наукових працях та подібних практичних застосуваннях, наприклад, в [8].

При вирішенні даної проблеми планування можливі декілька варіантів встановлення цілі. Один з варіантів полягає в мінімізації часу проходження всіх шляхів сполучень. Проте також важливим аспектом є мінімізація пройденої відстані транспортними засобами. Користувачам системи скоріше за все більш необхідно мінімізувати загальну вартість виконання робіт. А з огляду на екологічну ситуацію непоганий варіант для встановлення цілі – мінімізація забруднення довкілля, причиною якого є рух транспортних засобів. Проте всі ці цілі є досить схожими та пов'язаними між собою. Мінімізація пройденої відстані вірогідно призведе до зниження кількості забруднення, а також зменшить час проходження транспортними засобами шляхів сполучень. Тому в системі за ціль поставлено першу, а саме мінімізацію часу проходження всіх доріг.

1.1.2 Актори і функції

Інформаційна система планування обслуговування доріг міста призначена для використання муніципальними та приватними підприємствами, що виконують певні види робіт на вулицях міста. Використання цієї системи дозволить автоматизувати процеси керування спеціалізованою дорожньою технікою, досягти ефективного розподілу робочого навантаження та зменшити витрати.

Актором інформаційної системи є планувальник робіт. Дії, які може виконувати актор системи, визначено за допомогою UML-діаграми варіантів використання, яка представлена на рисунку А.1.

Планувальником є користувачем системи, який володіє даними, на основі яких відбувається складання плану, а саме:

- вид робіт, що потрібно виконати;
- кількість спеціалізованої дорожньої техніки;
- місткість одиниці спеціалізованої дорожньої техніки;
- перелік вулиць, на яких будуть проводитися роботи;
- інформація про пріоритетність кожної з доріг;
- часові обмеження.

Планувальник вводить інформацію у систему, яка на основі заданих даних складає план обслуговування. Після цього формується документ з запланованими маршрутами для водіїв. Далі водії виконують зазначені роботи відповідно до індивідуально визначеного маршруту. Якщо водій має необхідну інформацію стосовно характеру робіт, то він може виконувати також і функції планувальника.

Розглянемо більш детально прецеденти, наведені на схемі структурній варіантів використання.

Планувальник на початку роботи з системою визначає схему доріг. Для цього йому необхідно визначити місце розташування депо на карті та обрати дороги обслуговування. При доданні дороги необхідно обов'язково вказати клас пріоритетів, до якого ця дорога відноситься. Після додання дороги до схеми в разі необхідності її можна видалити або відредагувати її клас пріоритетів. Сформовану схему можна зберегти у системі для подальшого використання під час наступних планувальних

маршрутів з метою економії часу та спрощення введення даних. Збережені схеми можна видаляти, але не можна редагувати.

Коли схема визначена, планувальник переходить до пошуку маршрутів спеціалізованої дорожньої техніки. Маршрути будуються на заданій схемі доріг. На початку потрібно задати вхідні параметри процесу обслуговування, якими і володіє планувальник. На основі введених даних починається пошук маршрутів. Планувальник може побачити чи можливо побудувати маршрути на заданих параметрах, про що буде свідчити результат роботи системи. Якщо маршрути були побудовані, то планувальник може або просто переглянути загальний час, необхідний для обслуговування всіх доданих доріг з врахуванням заданих обмежень, або переглянути план робіт, що складається з усіх знайдених маршрутів, або почати симуляцію руху транспортних засобів.

1.1.3 Структура бізнес-процесів

В залежності від дій, визначених на схемі структурній варіантів використань, які може виконувати користувач системи, були виділені декілька бізнес-процесів системи. У таблиці 1.1 наведені ці бізнес-процеси інформаційної системи планування обслуговування доріг міста та варіанти використання, що їм відповідають.

Таблиця 1.1 – Бізнес-процеси системи згідно варіантів використання

| Бізнес-процес | Варіанти використання |
|----------------------|---|
| Створення схем доріг | Визначення місця розташування депо; Вибір доріг обслуговування; Задання класу пріоритетів дороги обслуговування; Визначення схеми доріг; Редагування класу пріоритетів дороги обслуговування; Видалення дороги обслуговування. |
| Ведення схем доріг | Збереження схеми доріг; Вибір збереженої схеми доріг; Видалення схеми доріг. |

Продовження таблиці 1.1

| Бізнес-процес | Варіанти використання |
|-------------------------------------|---|
| Складання плану робіт | Задання вхідних параметрів обслуговування; Пошук маршрутів транспортних засобів; Перегляд складеного плану робіт. |
| Симуляція руху транспортних засобів | Симуляція руху транспортних засобів. |

Окремо розглянемо послідовність подій, які виконуються користувачем та системою у рамках кожного з бізнес-процесів. Для цього більш детально розглянемо всі варіанти використання. У таблиці 1.2 наведено опис потоку подій бізнес-процесу «Створення схем доріг».

Таблиця 1.2 – Документація потоку подій бізнес-процесу «Створення схем доріг»

| Варіант використання: | Визначення місця розташування депо | | |
|--|--|---|--|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу визначити місце, з якого транспортні засоби будуть розпочинати і завершувати обслуговування | Відкрита географічна карта системи з активними дорогами району міста | 1. Користувач натискає на будь-яке місце на карті. 2. Система позначає розташування депо відповідним маркером. | Виконання варіанту використання «Вибір доріг обслуговування» |
| Варіант використання: | Вибір доріг обслуговування | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу визначити які | Виконання варіанту використання | 1. Користувач натискає на будь-яку активну дорогу на карті. | Виконання варіанту використання |

Продовження таблиці 1.2

| Опис | Передумови | Послідовність подій | Постумови |
|---|---|--|--|
| дороги потребують обслуговування | «Визначення місця розташування депо» | 2. Система виводить спливаюче вікно для вказання класу пріоритетів. | «Задання класу пріоритетів дороги обслуговування» |
| Варіант використання: | Задання класу пріоритетів дороги обслуговування | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу визначати до якого класу пріоритетів відноситься кожна дорога обслуговування | Виконання варіанту використання «Вибір доріг обслуговування» | 1. Користувач вводить чисельне значення класу пріоритетів у поле спливаючого вікна. 2. Користувач натискає на кнопку «Додати дорогу». 3. Система додає дорогу із заданим класом пріоритетів на вкладку «Обрані дороги» інформаційної панелі. 4. Система позначає обрану дорогу синім кольором на карті. | Виконання варіанту використання «Визначення схеми доріг» |
| Варіант використання: | Визначення схеми доріг | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу переглядати визначене депо, додані дороги | Виконання варіантів використання «Визначення місця розташування | 1. Користувач переглядає визначену схему доріг на вкладці «Обрані дороги» інформаційної панелі. | Виконання варіантів використання «Редагування класу пріоритетів дороги |

Продовження таблиці 1.2

| Опис | Передумови | Послідовність подій | Постумови |
|---|---|--|---|
| обслуговування та їх пріоритети | депо», «Вибір доріг обслуговування» та «Задання класу пріоритетів дороги обслуговування» | | обслуговування», «Видалення дороги обслуговування», «Збереження схеми доріг» та «Задання вхідних параметрів обслуговування» |
| Варіант використання: | Редагування класу пріоритетів дороги обслуговування | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу змінити клас пріоритетів для вже обраної дороги обслуговування | Виконання варіанту використання «Визначення схеми доріг» або «Вибір збереженої схеми доріг» | <ol style="list-style-type: none"> 1. Користувач натискає на значок редагування поряд з дорогою. 2. Система виводить поле для введення нового значення класу пріоритетів. 3. Користувач вводить нове числове значення. 4. Користувач натискає на кнопку «Змінити». 5. Система відображає нове значення класу пріоритетів для дороги обслуговування. | |

Продовження таблиці 1.2

| Варіант використання: | Видалення дороги обслуговування | | |
|--|---|--|------------------|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу видалити додану дорогу обслуговування | Виконання варіанту використання «Визначення схеми доріг» або «Вибір збереженої схеми доріг» | <ol style="list-style-type: none"> 1. Користувач натискає на значок видалення поряд з дорогою. 2. Система помічає видалену дорогу початковим сірим кольором. 3. Система видаляє дорогу з переліку доріг обслуговування на вкладці «Обрані дороги» інформаційної панелі. | |

Далі визначимо послідовність подій у рамках бізнес-процесу «Ведення схем доріг». У таблиці 1.3 наведено опис потоку подій бізнес-процесу «Ведення схем доріг».

Таблиця 1.3 – Документація потоку подій бізнес-процесу «Ведення схем доріг»

| Варіант використання: | Збереження схеми доріг | | |
|--|--|--|--|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу зберігати місце розташування депо та дороги обслуговування з їх класами пріоритетів | Виконання варіанту використання «Визначення схеми доріг» | <ol style="list-style-type: none"> 1. Користувач натискає на кнопку «Зберегти схему». 2. Система виводить спливаюче вікно для введення назви схеми. 3. Користувач вводить назву схеми доріг. 4. Користувач натискає кнопку «Зберегти». | Виконання варіантів використання «Вибір збереженої схеми доріг» та «Видалення схеми доріг» |

Продовження таблиці 1.3

| Опис | Передумови | Послідовність подій | Постумови |
|---|--|---|--|
| | | 5. Система додає збережену схему на вкладку «Схеми доріг» інформаційної панелі. | |
| Варіант використання: | Вибір збереженої схеми доріг | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу додати збережену схему доріг на вкладку «Обрані дороги» інформаційної панелі для подальшого планування маршрутів | Виконання варіанту використання «Збереження схеми доріг» | <ol style="list-style-type: none"> 1. Користувач переходить на вкладку «Схеми доріг» інформаційної панелі. 2. Користувач натискає на назву необхідної схеми доріг. 3. Система додає вибрану схему доріг на вкладку «Обрані дороги» інформаційної панелі. 4. Система помічає синім кольором дороги обслуговування, які були присутні в збереженій схемі доріг. | Виконання варіантів використання «Редагування класу пріоритетів дороги обслуговування», «Видалення дороги обслуговування», «Збереження схеми доріг» та «Задання вхідних параметрів обслуговування» |
| Варіант використання: | Видалення схеми доріг | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу видаляти збережену схему доріг | Виконання варіанту використання «Збереження схеми доріг» | 1. Користувач переходить на вкладку «Схеми доріг» інформаційної панелі. | |

Продовження таблиці 1.3

| Опис | Передумови | Послідовність подій | Постумови |
|------|------------|--|-----------|
| | | <p>2. Користувач натискає на значок видалення поряд з назвою схеми доріг.</p> <p>3. Система видаляє схему доріг на вкладці «Схеми доріг» інформаційної панелі.</p> | |

У таблиці 1.4 наведено опис потоку подій бізнес-процесу «Складання плану робіт».

Таблиця 1.4 – Документація потоку подій бізнес-процесу «Складання плану робіт»

| Варіант використання: | Задання вхідних параметрів обслуговування | | |
|---|---|---|--|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу вказувати інформацію щодо виду робіт та значень обмежень | Виконання варіанту використання «Визначення схеми доріг» або «Вибір збереженої схеми доріг» | <p>1. Користувач натискає на кнопку «Розрахувати час».</p> <p>2. Система виводить спливаюче вікно для введення вхідних параметрів.</p> <p>3. Користувач заповнює всі поля.</p> <p>4. Користувач натискає на кнопку «Сформувати план робіт».</p> | Виконання варіанту використання «Пошук маршрутів транспортних засобів» |

Продовження таблиці 1.4

| Варіант використання: | Пошук маршрутів транспортних засобів | | |
|---|---|--|---|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу побудувати маршрути транспортних засобів | Виконання варіанту використання «Задання вхідних параметрів обслуговування» | <p>1. Система запускає алгоритм пошуку маршрутів транспортних засобів.</p> <p>2. Якщо маршрути знайти не можливо, то виводиться повідомлення про те, що потрібно відредагувати вхідні параметри.</p> <p>3. Якщо маршрути знайдені, то виводиться повідомлення про загальний час обслуговування та посилання на сформований план робіт.</p> | Виконання варіантів використання «Перегляд складеного плану робіт» та «Симуляція руху транспортних засобів» |
| Варіант використання: | Перегляд складеного плану робіт | | |
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу переглянути побудовані маршрути транспортних засобів | Виконання варіанту використання «Пошук маршрутів транспортних засобів» | <p>1. Користувач натискає на значок копіювання поряд с посиланням на складений план робіт.</p> <p>2. Користувач відкриває нову вкладку браузера та переходить за скопійованим посиланням.</p> <p>3. Відкривається онлайн excel документ з побудованими маршрутами.</p> | |

У таблиці 1.5 наведено опис потоку подій бізнес-процесу «Симуляція руху транспортних засобів».

Таблиця 1.5 – Документація потоку подій бізнес-процесу «Симуляція руху транспортних засобів»

| Варіант використання: | Симуляція руху транспортних засобів | | |
|---|--|--|------------------|
| Опис | Передумови | Послідовність подій | Постумови |
| Дозволяє користувачу наочно бачити як транспортні засоби рухаються по знайденим маршрутам | Виконання варіанту використання «Пошук маршрутів транспортних засобів» | <ol style="list-style-type: none"> 1. Користувач натискає на кнопку «Почати симуляцію». 2. Система будує граф на карті. 3. Система позначає дуги обслуговування синім кольором на графі. 4. Моделі транспортних засобів починаються рухатись по дугам графа відповідно до знайдених маршрутів. 5. Система позначає дугу обслуговування жовтим кольором, коли транспортний засіб знаходиться в процесі її обслуговування. 6. Система позначає дугу обслуговування зеленим кольором, коли транспортний засіб завершив обслуговування дуги. | |

1.2 Опис постановки задачі

Призначенням є підтримка процесу складання плану проходження доріг району міста обслуговуючими транспортними засобами при очищенні покриття автомобільних доріг від сміття, бруду і сторонніх предметів, мийці та очищенні від пилу, нанесенні та відновленні зношеної розмітки, розчищенні автомобільних доріг від снігу, розподіленні протиожеледних матеріалів (піску, солі).

Метою є підвищення якості процесів обслуговування доріг міста за рахунок мінімізації часу проходження транспортними засобами запланованих шляхів сполучень (із врахуванням обмежень на директивні терміни виконання робіт та місткість машин).

Для реалізації поставленої мети необхідно розробити складові, з яких складається інформаційна система планування обслуговування доріг міста, а саме:

- а) формування схеми доріг району міста для обслуговування:
 - 1) визначення місця розташування вершини-депо;
 - 2) задання доріг для обслуговування;
 - 3) побудова графу.
- б) планування обслуговування доріг транспортними засобами:
 - 1) введення актуальних значень:
 - визначення класу пріоритетів для кожної з доріг;
 - встановлення часу початку виконання робіт;
 - встановлення директивних термінів у відповідність до класів пріоритетів;
 - визначення необхідного виду робіт на дорогах;
 - задання кількості спеціалізованої дорожньої техніки;
 - задання місткості одиниці спеціалізованої дорожньої техніки;
 - 2) складання плану проходження доріг для кожного з транспортних засобів:
 - розрахунок шляхів проходження доріг перед початком виконання робіт з врахуванням інтенсивності дорожнього руху;

- за необхідності, корегування шляхів проходження доріг відповідно до інтенсивності дорожнього руху в процесі виконання робіт;
- 3) формування звіту «Шляхи проходження доріг транспортними засобами».

в) моніторинг виконання плану обслуговування доріг транспортними засобами:

- 1) використання симулятора руху транспортних засобів по дорогам району міста згідно з визначеним планом обслуговування.

1.3 Рішення з інформаційного забезпечення

Вхідні дані надходять з двох джерел, а саме від:

- користувача;
- відкритих джерел географічних даних.

Вхідні дані, які вводить користувач інформаційної системи:

- місце розташування депо;
- дороги обслуговування;
- класи пріоритетів доріг обслуговування;
- назва схеми доріг;
- час початку обслуговування;
- директивні терміни завершення робіт;
- вид робіт;
- кількість спеціалізованої дорожньої техніки;
- місткість одиниці спеціалізованої дорожньої техніки.

Вхідні дані, які надходять з відкритих джерел географічних даних:

- бали завантаженості дорожнього руху;
- координати точок доріг;
- час проходження доріг.

Вихідними даними інформаційної системи планування обслуговування доріг міста є:

- загальний час обслуговування;
- excel документ плану робіт з побудованими маршрутами.

Структура розробленої бази даних інформаційної системи планування обслуговування доріг міста на логічному рівні наведена на рисунку 1.1. Така структура дозволить ефективно керувати даними, що необхідні для роботи з картою і водночас симулятору руху транспортних засобів та загальною інформацією про дороги, види робіт, завантаженість дорожнього руху та ін.



Рисунок 1.1 – Логічна схема бази даних «InformationSystem»

У таблиці 1.6 наведено більш детальний опис таблиць розробленої бази даних.

Таблиця 1.6 – Опис структури бази даних «InformationSystem»

| Назва таблиці | Назва стовпця | Тип даних | Призначення |
|---------------------------------------|---------------|--------------|---|
| Coordinates – таблиця координат точок | Id | int | Ідентифікатор координати |
| | Latitude | double | Значення широти на карті |
| | Longitude | double | Значення довготи на карті |
| Roads – таблиця доріг | Id | int | Ідентифікатор дороги |
| | RoadName | varchar(100) | Назва вулиці |
| | StartPoint | int | Координата початку дороги на карті (ідентифікатор координати) |
| | EndPoint | int | Координата кінця дороги на карті (ідентифікатор координати) |

Продовження таблиці 1.6

| Назва таблиці | Назва стовпця | Тип даних | Призначення |
|--|---------------|--------------|--|
| RoadConnections – таблиця зв'язків між дорогами | Id | int | Ідентифікатор зв'язку між дорогами |
| | RoadFrom | int | Ідентифікатор першої дороги |
| | RoadTo | int | Ідентифікатор другої дороги, що перетинається з першою |
| Jobs – словник видів робіт | Id | int | Ідентифікатор виду роботи |
| | Type | varchar(200) | Назва виду роботи |
| ServiceRoadSpecification – таблиця характеристики обслуговування доріг | Id | int | Ідентифікатор характеристики обслуговування дороги |
| | RoadId | int | Ідентифікатор дороги |
| | JobId | int | Ідентифікатор виду роботи |
| | ServiceTime | time | Час обслуговування дороги для даного виду роботи |
| TransitRoadSpecification – таблиця характеристики транзитних проїздів по дорогам | Id | int | Ідентифікатор характеристики транзитного проїзду |
| | RoadId | int | Ідентифікатор дороги |
| | TransitTime | time | Час транзитного проїзду по дорозі |
| TrafficLevel – словник з інформацією про завантаженість дорожнього руху | Id | int | Ідентифікатор рівня завантаженості руху |
| | Level | int | Рівень завантаженості руху |
| | Multiplier | float | Коефіцієнт інтенсивності руху |
| RoadPattern – таблиця збережених схем доріг | Id | int | Ідентифікатор схеми доріг |
| | PatternNameId | int | Ідентифікатор назви схеми доріг |
| | RoadId | int | Ідентифікатор дороги |
| | Priority | int | Клас пріоритету дороги |

Продовження таблиці 1.6

| Назва таблиці | Назва стовпця | Тип даних | Призначення |
|--|---------------|--------------|---|
| PatternData – таблиця відомостей про збережені схеми доріг | Id | int | Ідентифікатор відомості про схему доріг |
| | PatternName | varchar(200) | Назва схеми доріг |
| | DepotPoint | int | Координата депо на карті (ідентифікатор координати) |

Висновки

Інформаційна система планування обслуговування доріг міста є практичною реалізацією актуальної на даний час задачі оптимізації, оскільки має широке застосування в реальному житті. В деяких випадках навіть незначне покращення маршрутів транспортних засобів може дати значну економію ресурсів.

В цьому розділі було наведено формулювання постановки задачі та мети дослідження, визначено призначення системи та її складові, а саме: побудова схеми доріг району міста для обслуговування, планування обслуговування доріг транспортними засобами, моніторинг виконання плану обслуговування транспортними засобами. Крім того, розглянуто процеси діяльності, відповідно до яких було виділено користувача системи, який має назву планувальника, та визначено функції системи у вигляді UML-діаграми варіантів використання. На основі цієї діаграми було визначено чотири бізнес-процеси системи, а саме: створення схеми доріг, ведення схеми доріг, складання плану робіт та симуляція руху транспортних засобів. У розділі детально розглянуто послідовність подій у межах кожного з бізнес-процесів. Також було визначено джерела надходження даних у інформаційну систему, детально описано вхідні і вихідні дані та розроблено структуру бази даних, яка дозволить зберігати та ефективно керувати цими даними.

2 МОДЕЛІ ТА МЕТОДИ ПЛАНУВАННЯ ОБСЛУГОВУВАННЯ ДОРІГ МІСТА

2.1 Аналіз задач маршрутизації транспортних засобів по дугам

2.1.1 Визначення та застосування задачі ARP

Задачі маршрутизації (Vehicle Routing Problem) широко досліджуються протягом останніх років. Це відбувається в основному тому, що зросла роль ефективного використання парку автомобілів та зросла кількість практичних реалізацій та застосувань. Метою рішення різних видів задач маршрутизації транспортних засобів є складання мінімальних за цінними витратами маршрутів. Приклад задачі VRP зображено на рисунку А.2.

Ці задачі можуть бути поділені на задачі маршрутизації по вершинам (Node Routing Problems) та задачі маршрутизації по дугам (Arc Routing Problems).

У задачах маршрутизації по вершинам вимоги на обслуговування задаються у вершинах графа. Прикладом такої задачі може слугувати комівояжер, який має обійти кожне місто в деякому регіоні.

Задача маршрутизації по дугам має вимоги на обслуговування, що розташовуються на дузі графа. Прикладом такої задачі маршрутизації по дугам може бути листоноша, що має обійти кожен вулицю міста. Тобто задачі маршрутизації по дугам Arc routing problems (ARP) природньо виникають у випадку, коли обслуговування потребують вулиці або споживачі, яким потрібно надати обслуговування, розташовуються вздовж вулиць [3]. Приклад задачі ARP зображено на рисунку А.3.

В області задач маршрутизації по вершинам було зроблено значно більше досліджень в той час, як задачі маршрутизації по дугам отримали порівняно мало уваги, не зважаючи на те, що вони мають застосування до багатьох реальних проблем, таких як прибирання сміття, прибирання вулиць, доставка молока чи пошти, а також контроль за системами постачання (електричні мережі, телефонні лінії чи залізничні

колії). Огляд можливого використання цих задач для вирішення даних проблем опубліковано в [9].

Розглянемо ще декілька більш конкретних прикладів застосування задач маршрутизації по дугам у сучасній практиці.

У 2017 році вчений з кафедри промислової інженерії університету Ататюрка Мустафа Ялмаз (Mustafa Yılmaz) опублікував працю, в якій дослідив застосування задачі китайського листоноші (Chinese Postman Problem) при перевірці стану залізничних колій та безпечних умов для руху поїздів вздовж широкомасштабної залізничної мережі Туреччини [10]. Такі контролюючі роботи зазвичай виконуються декількома спеціалізованими машинами з великим і дорогим обладнанням, тому знаходження оптимального маршруту машин під час перевірок є надзвичайно важливим з точки зору вартості та відстані (щороку на перевірку залізничної мережі витрачаються мільйони доларів). За мету було поставлено знаходження мінімальної загальної відстані шляху вздовж залізничних колій. Застосування цієї моделі у реальній практиці дозволило знизити поточну вартість перевірки на 20,76%.

Хуанфа Чен (Huanfa Chen) та Тао Ченг (Tao Cheng) з університету Лондона у своїй праці продемонстрували можливість побудови маршрутів для патрулювання міста поліцейськими з використанням задачі мінімаксу сільського листоноші (Min-Max Rural Postman Problem) [11]. Ефективне патрулювання вимагає побудови оптимальних маршрутів, оскільки групи патрулів повинні об'їхати закріплені за ними ділянки хоча б один раз, при цьому довжини маршрутів для кожної групи повинні бути приблизно однаковими, щоб уникнути робочого перенавантаження та незадоволеності працівників. Ця проблема є актуальною, бо при існуючій високій щільності ділянок патрульні ресурси залишаються обмеженими.

Проте також відомі застосування задач маршрутизації по дугам в більш нетипових сферах. Так група вчених із Теннессі опублікувала працю, в якій описано розроблений ними метод програмування для зменшення часу роботи інструменту 3D-друку завдяки знаходженню мінімальних шляхів одношарових екструзій в адитивному виробництві [12]. Вони представили графічну модель тривимірного (3D)

процесу друку та показали як вони використали вирішення задачі китайського листоноші (Chinese Postman Problem) для оптимізації руху екструдера на сітці.

2.1.2 Основні поняття

Введемо деякі поняття з теорії графів, які будуть використовуватись при аналізі предметної області в цьому розділі [13].

АРР можна описати за допомогою графа $G = (V, F)$, де V – це множина вершин, $F = E \cup A$, де E – це множина ребер та A – це множина дуг.

Вершина, Вузол – точка, з якої розходяться (або навпаки сходяться) ребра і дуги.

Ребро графа – лінія, що з'єднує одну вершину графа з іншою.

Дуга – таке ребро, для якого одна вершина вважається початком, а інша – кінцем.

Петля – це ребро, що з'єднує вершину саму з собою.

Степінь вершини – кількість ребер графа G , що виходять із однієї вершини.

Повний граф – граф, в якому будь-які його дві вершини v_1, v_2 (різні, якщо не допускаються петлі) з'єднані ребром.

Наведемо одну з класифікацій графів за орієнтованістю.

Неорієнтований граф $G = (V, E)$ складається з множини вершин V та неорієнтованих ребер E . Кожне *ребро* (u, v) з'єднує вершини u та v з V .

Орієнтований граф $G = (V, A)$ складається з множини вершин V та орієнтованих дуг A . *Дуга* (t, h) – це орієнтоване ребро з вершини t до вершини h з множини V . t називається *початком* дуги (t, h) , а h – *кінцем* дуги (t, h) .

Направлений граф – орієнтований граф, у якому дві вершини t і h з'єднуються однією дугою.

Змішаний граф $G = (V, E, A)$ складається з множини вершин V , множини неорієнтованих ребер E та множини орієнтованих дуг A . Орієнтованому ребру (u, v) задається напрямок від u до v .

Вітряний граф (windy graph) $G=(V,E)$ – це неорієнтований граф з двома вартостями на ребро $i, j \in E: c_{ij}$ та c_{ji} , що відповідають напрямку проходження з i до j та з j до i відповідно.

Наведемо основні визначення, що стосуються шляхів.

Маршрут/шлях в графі – така послідовність вершин і ребер $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$, що будь-які дві сусідні вершини v_i і v_{i+1} з'єднані ребром e_{i+1} . Якщо $v_0 = v_k$, то маршрут *замкнений*, інакше *відкритий*.

Ланцюг в графі – маршрут, в якому кожне ребро повторюється не більше одного разу.

Цикл – ланцюг, в якому кінець останнього ребра збігся з початком першого ребра.

Ейлерів ланцюг (або Ейлерів цикл) – ланцюг (цикл), що проходить через кожне ребро графа тільки один раз, при цьому вершини можуть повторюватися.

Довжина шляху – кількість ребер або сума їх довжин (якщо задані), що входять в послідовність, яка задає цей шлях. Для шляху v_1, v_2, \dots, v_n довжина дорівнює $n-1$.

Дві вершини в графі *зв'язні*, якщо існує такий ланцюг, який з'єднує їх.

Зв'язний граф – такий граф, який має ланцюг між будь-якими двома його вершинами.

Підмножина $R \subseteq E \cup A$ ребер та дуг називається *обов'язковою*, тобто такою, що обов'язково має бути обслуженою або покритою транспортним засобом. *Покриваючим* обходом для R є такий обхід, який проходить всі ребра та дуги з R хоча б раз.

Зважений граф – граф, для кожного ребра якого встановлене деяке числове значення u , що називається *вагою ребра*.

Ребра та дуги можуть бути поділені на наступні два базові класи:

- *дуги обслуговування*, для яких задана вартість обслуговування s ;
- *транзитні (прохідні) дуги*, для яких задана вартість транзитного проїзду c .

2.1.3 Розвиток та класифікація задач маршрутизації транспортних засобів по дугам

Розглянемо задачі ARP в історичній перспективі [14].

Вивчення задачі маршрутизації по дугам почалось 26 серпня 1735 року, коли Леонард Ейлер (Leonhard Euler) опублікував вирішення задачі про сім мостів Кенігсберга. Ця задача зараз відома як задача про ейлерів ланцюг і полягає в наступному. *Дано зв'язний граф $G = (V, E)$. Знайти цикл, який обходить кожне ребро рівно один раз, або визначити, що такого циклу не існує.*

Далі у 1962 році китайським математиком Мей-Ку Кван (Mei-Ku Kuan) було об'єднано задачу VRP і задачу маршрутизації по дугам, первісно визначену Ейлером, та розроблену нову модель під назвою задача про китайського листоношу (CPP) [2]. Задачу формально можна описати так: *Дано зв'язаний граф $G = (V, E, C)$, де C – матриця відстаней. Знайти маршрут, який проходить через кожну дугу принаймні один раз і є найкоротшим.*

Це стало поштовхом до наукових проривів для багатьох вчених [15]. У 1974 році Орлов (Orloff) запропонував задачу про сільського листоношу (RPP) [16], яку формально можна викласти так: *Дано неорієнтований граф $G = (V, E, C)$, де C є матрицею вартостей дуг. Знайти маршрут мінімальної вартості, який проходить через кожне ребро в підмножині $R \subseteq E$ принаймні один раз.* Крім того існує багато варіантів даної задачі, наприклад, задача сільського листоноші з директивними термінами (RPP with DC), в якій множина R обов'язкових ребер розділена на кілька підмножин, які в свою чергу визначають порядок обходу.

Задача мінімаксу k -китайських листонош (MM k -CPP) була запропонована Фредріксоном (Frederickson) та ін. в 1978 році [17] і формально може бути викладена наступним чином: *Дано зв'язний неорієнтований граф $G = (V, E, C)$, де C – матриця відстаней зі спеціальним вузлом-депо. Знайти k маршрутів, що починаються і закінчуються в вузлі-депо, так щоб кожне ребро було покритим принаймні одним маршрутом і довжина найдовшого маршруту має бути мінімальною.* Слід зазначити, що ціллю цієї задачі є зведення до мінімуму часу обслуговування, тоді як

більшість інших задач з кількома листоношами намагаються звести до мінімуму сумарну пройдену відстань.

Задача маршрутизації по дугам з врахуванням місткості транспортних засобів (CARP – Capacitated Arc Routing Problem) вперше була запропонована Голденом (Golden) та Вонгом (Wong) в 1981 році [18]. Формально її можна викласти так: *Дано зв'язний неорієнтований граф $G = (V, E, C, D)$, де C – матриця вартості і D – матриця вимог (попиту). Враховуючи кількість ідентичних машин місткістю Q кожна, знайти кількість маршрутів, щоб 1) кожна дуга з позитивним попитом обслуговувалась рівно однією машиною, 2) сума попитів дуг, що обслуговуються кожним транспортним засобом, не перевищувала його місткості Q , і 3) загальна вартість маршрутів була мінімальною.*

Задача китайського листоноші з врахуванням місткості (CCRP) є варіацією CARP, де кожне ребро в графі має строго позитивний попит, і була вперше запропонована Хрістофідесом (Christofides) в 1973 році [19].

Класифікація задач маршрутизації транспортних засобів наведена на рисунку А.4. Якщо узагальнити, то ці задачі можна поділити за такими ознаками:

а) за місцем знаходження вимог на обслуговування:

- 1) по вершинам;
- 2) по дугам;

б) за обмеженням по місткості транспортних засобів:

- 1) без врахування місткості;
- 2) з врахуванням місткості;

в) за характеристиками транспортних засобів:

- 1) одна машина;
- 2) кілька машин;
- 3) кілька машин з однаковою місткістю;
- 4) кілька машин з різними місткостями;

г) за часовими вимогами:

- 1) без часових вікон;
- 2) з часовими вікнами;

д) за цільовою функцією:

- 1) з однією цільовою функцією;
- 2) з кількома цільовими функціями;

е) за іншими ознаками.

2.1.4 Задача китайського листоноші (Chinese Postman Problem)

Задача китайського листоноші (CPP) – це класифікація задач маршрутизації по дугам (ARP), що включає знаходження шляху найменшої вартості, що обходить кожен дугу неорієнтованого графу хоча раз. Розглянемо зв'язаний неорієнтований граф $G=(V, E)$, де V – це множина вершин та E – це множина ребер. Всі дуги з вимогами називаються *дугами для обслуговування*. В практичних визначеннях, дуги обслуговування представляють вулиці, які, наприклад, мають поштові скриньки, куди повинні бути рознесені листи. Кожна така дуга буде мати три характеристики [20].

Перша характеристика – це *час проходження* (c_e): час на проходження дуги (тобто час витрачений на те, щоб просто проїхати вздовж вулиці без надання жодного обслуговування на ній).

Другою характеристикою є *час обслуговування* (s_e): час на обслуговування дуги з вимогою (тобто час витрачений на те, щоб проїхати вулицею та надати певне обслуговування, наприклад, час доставки пошти та листів в той час, як листоноша рухається вулицею). Очевидно $s_e \geq c_e$, оскільки час обслуговування має включати також і час проходження.

Також можна виділити таку характеристику як *обсяг* (q_e): обсяг робіт, що мають бути виконані на дузі. Проте вона може не включатись в задачу листоноші, оскільки поштова машина може вмістити величезну кількість листів, але це може бути дуже важливою складовою в інших застосуваннях маршрутизації по дугам, таких як прибирання сміття.

Визначається також *вершина-депо*. У CPP потрібно знайти мінімальну вартість маршруту, що проходить кожне ребро з E точно по одному разу і починається та закінчується у вершині-депо.

Стандартна задача китайського листоноші може бути математично сформульована наступним чином [21]:

Дано $G(V, E, u)$ неорієнтований зважений граф, де V – множина вершин, E – множина ребер, u – визначає додатну вартість проходження ребер.

Нехай x_{ij} – цілочисельна змінна, що позначає кількість проходження ребер (i, j) від i до j , $\delta(i)$ – це множина ребер, що з'єднують вершину i з іншими найближчими вершинами. Тоді математична модель має вигляд:

Цільова функція:

$$\min \sum_{(i,j) \in E} (u_{ij}x_{ij} + u_{ji}x_{ji}) \quad (2.1)$$

Обмеження:

$$x_{ij} + x_{ji} \geq 1, \quad (i, j) \in E, \quad (2.2)$$

$$\sum_{(i,j) \in \delta(i)} (x_{ij} - x_{ji}) = 0 \quad i \in V, \quad (2.3)$$

$$x_{ij}, x_{ji} \in N, \quad (i, j) \in E. \quad (2.4)$$

Розглянемо деякі із існуючих класифікацій СРР. Нижче для прикладу наведено їх описи [22].

Неорієнтована задача китайського листоноші Undirected Chinese Postman Problem (UCPP). Постановкою задачі є зв'язаний неорієнтований граф $G = (V, E)$ та ваги u . Цільова функція – це знаходження замкнутого шляху в G мінімальної вартості, що проходить кожне ребро щонайменше один раз.

Орієнтована задача китайського листоноші Directed Chinese Postman Problem (DCPP). Постановка задачі визначається орієнтованим зв'язаним графом $G = (V, A)$ та вагами u . Знайти орієнтований замкнутий шлях у G мінімальної вартості, що обходить кожну дугу щонайменше раз.

Змішана задача китайського листоноші Mixed Chinese Postman Problem (MCPP). Постановка задачі визначається зв'язаним змішаним графом $G = (V, E, A)$ та вагами u . Цільова функція – знайти змішаний замкнутий шлях в G мінімальної вартості, що проходить кожне ребро щонайменше раз.

Неорієнтована задача сільського листонші Undirected Rural Postman Problem (URPP). Постановка задачі формулюється за допомогою неорієнтованого графу $G = (V, E)$, вагами u та підмножиною обов'язкових ребер $R \subseteq E$. Цільова функція – знайти замкнений шлях в G мінімальної вартості, що проходить по кожному ребру з R щонайменше раз.

Орієнтована задача сільського листонші Directed Rural Postman Problem (DRPP). Постановка задачі формулюється за допомогою зв'язаного орієнтованого графу $G = (V, A)$, ваг u та множини обов'язкових дуг $R \subseteq A$. Цільова функція – знайти орієнтований замкнутий шлях в G мінімальної вартості, що проходить кожену дугу з R хоча б раз.

Змішана задача сільського листонші Mixed Rural Postman Problem (MRPP). Постановка задачі формулюється за допомогою зв'язаного змішаного графу $G = (V, E, A)$, ваг u та множиною обов'язкових ребер $R \subseteq E$. Цільова функція – це знайти змішаний замкнутий шлях G мінімальної вартості, що обходить кожне ребро в R щонайменше раз.

Вітряна задача листонші Windy Postman Problem (WPP). Постановка задачі формулюється зв'язаним неорієнтованим графом $G = (V, E)$, вагами u_1 (проходження ребер від кінця до початку) та u_2 (проходження ребер від початку до кінця). Знайти замкнений шлях у G мінімальної вартості, що проходить кожне ребро щонайменше раз.

Вітряна зважена задача k -сільських листонш Windy Capacitated k -Rural Postman Problem (WC k -RPP). Постановкою задачі є зв'язаний неорієнтований граф $G = (V, E)$, ваги u_1 та u_2 , підмножина обов'язкових ребер $R \subseteq E$, місткість машини Q та окремий вузол-депо. Цільова функція – знайти множину k замкнених маршрутів у G з мінімальною сумарною вартістю, що обходить кожне ребро в R щонайменше раз та де кожне ребро обслуговується тільки одним з k маршрутів, а сума вимог кожного з обов'язкових ребер для кожного маршруту не має перевищувати місткість машини [23].

Узагальнена задача китайського листоноші *Generalized Chinese Postman Problem (GCPP)*. Постановкою задачі є зв'язаний неорієнтований граф $G=(V,E)$ та ваги u . Ребра поділені на підмножини ребер E_1, \dots, E_k . Цільова функція – знайти замкнений шлях у G мінімальної вартості, що проходить хоча б по одному ребру з кожної підмножини $E_i, i=1, \dots, k$.

Неорієнтована зважена задача китайського листоноші *Undirected Capacitated Chinese Postman Problem (UCCPP)*. Постановка задачі формулюється зв'язаним неорієнтованим графом $G=(V,E)$, вагами u , вимогами d , місткістю машини Q та окремим вузлом-депо. Цільова функція – знайти множину замкнутих шляхів у G з мінімальною сумарною вартістю, що задовольняє наступним умовам:

- кожен маршрут починається і закінчується у вузлі-депо;
- кожне ребро покривається хоча б одним маршрутом;
- сума вимог для кожного маршруту не має перевищувати місткість машини.

Змішана задача k -китайських листонош *Mixed k -Chinese Postman Problem (M k -CPP)*. Постановка задачі формулюється зв'язаним змішаним графом $G=(V,E,A)$, вагами u , окремим вузлом-депо та фіксованим числом $k > 1$ листонош. Цільова функція – знайти множину k змішаних замкнених шляхів у G з мінімальною сумарною вартістю, що задовольняє наступні умови:

- кожен маршрут починається і закінчується у вузлі-депо;
- кожне ребро покривається принаймні одним маршрутом.

Задача мінімаксу k -китайських листонош *Min-Max k -Chinese Postman Problem (MM k -CPP)*. Постановка задачі формулюється за допомогою зв'язаного неорієнтованого графа $G=(V,E)$, ваг u , окремого вузла-депо та фіксованої кількості $k > 1$ листонош. Цільова функція – знайти множину k -замкнутих шляхів у G , мінімізуючи вагу найважчого маршруту, що задовольняє наступним умовам:

- кожен обхід починається і закінчується у вузлі-депо;
- кожне ребро включається хоча б до одного маршруту.

Неорієнтована ієрархічна задача китайського листоноші *Undirected Hierarchical Chinese Postman Problem (UHCPP)*. Постановка задачі формулюється за

допомогою зв'язаного неорієнтованого графу $G = (V, E)$ та ваг u . Множина ребер E поділена на кілька класів. Якщо клас ребер E_p передує іншому класу E_q , тоді всі ребра з E_p повинні бути обслужені перед ребрами з E_q . Цільова функція – знайти замкнутий маршрут у G мінімальної вартості, враховуючи зв'язок передування, що проходить кожне ребро щонайменше раз.

Задача сільського листоноші з класами директивних термінів Rural Postman Problem with Deadline Classes (RPP with DC). Постановка задачі формулюється за допомогою зв'язаного неорієнтованого графу $G = (V, E)$, ваг u та окремого вузла-депо. Підмножина $R \subset E$ обов'язкових ребер поділена на кілька класів директивних термінів (дедлайнів) R_1, \dots, R_L . Ребра з R_1 мають бути обслужені до часу T_1 , ребра з R_2 до часу T_2 і т.д. Цільова функція – знайти замкнений маршрут у G мінімальної вартості, що починається і закінчується у вузлі-депо та проходить кожне ребро з R щонайменше раз, зважаючи на обмеження по часу.

Далі більш детально розглянемо деякі класи задач, на основі яких було сформульовано постановку задачі для використання у складі інформаційної системи планування обслуговування доріг міста.

2.1.5 Ієрархічна задача китайського листоноші (Hierarchical Chinese Postman Problem)

Для потреб забезпечення обслуговування доріг місто чи країна зазвичай діляться на кілька зон. Обладнання та команди, що необхідні для виконання робіт розташовуються в одній чи кількох зонах для надання якнайкращого сервісу. Кожна зона складається з мережі доріг та громадських місць, для яких визначений власний пріоритет. Найвищий пріоритет надається діловим районам: головним дорогам для проїзду автомобілів, смугам руху громадського транспорту, школам та районам лікарень. Другий пріоритет надається для обслуговування шляхів, що ведуть до головних доріг, шкіл та лікарень. Третій пріоритет надається вулицям житлових районів, і четвертий пріоритет – громадським парковкам, узбіччям тощо. Обслуговування доріг вищого пріоритету в певній зоні має бути виконано в першу чергу.

Оскільки дороги з однаковим пріоритетом не обов'язково знаходяться поряд один з одним, то в той час, як обслуговуються дороги з вищим пріоритетом, можуть проходитись деякі дороги більш низького пріоритету, щоб дістатись до доріг з вищим. Відстань, що проходить транспортний засіб вздовж вулиць без виконання робіт, відноситься до так званих «порожніх» поїздок. Порожній проїзд може значно сповільнити процес обслуговування, а отже має бути мінімізований. Порожній проїзд також збільшує вартість обслуговування доріг, оскільки він включається у вартість праці, затримку в наданні сервісу тощо. Звідси є можливість зменшити вартість за рахунок зменшення відстаней порожніх проїздів. Завдяки зменшенню порожнього проїзду стає очевидно, що збитки завдані приватній власності, громадськості та загальна вартість процесу обслуговування доріг можуть бути зменшені.

Через представлення кожної вулиці як дуги і кожного перехрестя як вершини кожна зона може бути представлена як граф. Оскільки вулиці мають пріоритети, граф насправді є ієрархічним. Через мінімізацію порожніх проїздів час, необхідний для обслуговування, може бути також мінімізований. Отже, проблема визначення оптимальних маршрутів для спеціалізованих команд обслуговування є НСРР [24]. Таким чином ця задача зводиться до визначення маршруту, що починається в депо і обходить всі дуги, задовольняючи згадану умову передування.

В НСРР множина дуг A поділена на класи згідно з пріоритетами A_1, A_2, \dots, A_k з $A_i \cap A_j = \emptyset$ для $i \neq j$, $\bigcup_{i=1}^k A_i = A$, а класи відповідно $A_1 \prec A_2 \prec \dots \prec A_k$, $k > 1$ [21].

В літературі є два різні формулювання цільової функції для вирішення ієрархічної СРР. Відмінності пов'язані з різною інтерпретацією порядку. Джіані (Ghiani), Імпрота (Improta) (2000) та Дроп (Dror) (1987) визначають цільову функцію як знаходження шляху від депо d , що проходить всі дуги щонайменше раз. Жодна з дуг класу A_j не може бути *пройдена* до проходження всіх дуг з A_i , якщо $A_i \prec A_j$. Альфа (Alfa) і Луї (Liu) (1988), Ейзелт (Eiselt) (1995), Летчфорд (Letchford) і Еглейс (Eglese) (1998) посилаються на різницю між *обслуговуванням* та *проходженням*. Ейзелт та ін. переформулювали НСРР для більш загального випадку. В їхньому формулюванні, дуги можуть бути пройдені в будь-який момент, в той час як

обслуговуватись вони можуть лише відповідно до заданого порядку. Отже, в їхньому формулюванні метою НСРР є визначення шляху мінімальної вартості, що обслуговує всі дуги, при цьому дотримуючись порядку (\prec): жодна з дуг із A_j не може бути обслугована до того часу, коли не будуть обслужені всі дуги з A_i , якщо $A_i \prec A_j$. Однак, дуги можуть бути пройдені в будь-який момент без їхнього обслуговування, тобто мати *порожнє проходження*.

2.1.6 Задача k -китайських листонош (k -Chinese Postman Problem)

В загальному k -СРР можна описати так: $k \geq 2$ листонош мають виконати обслуговування на заданому графі, де кожне ребро має час обслуговування та час транзитного проходження. Завданням є знаходження маршрутів для k листонош з мінімальним загальним часом. На практиці, це часто призводить до незадовільних результатів, оскільки різниця між загальними тривалостями роботи кожного листоноші не береться до уваги.

Є ціла низка реальних задач, пов'язаних з k -СРР. Зазвичай компанія наймає групу людей (а не одну людину), з якої кожний обслуговує деяку підмножину проблеми. До таких сервісів, що вимагають наявності кількох людей в персоналі для обслуговування, належать, наприклад, кур'єрська доставка чи доставка пошти.

k -СРР формулюється наступним чином [25]: $k \geq 2$ листонош на зв'язаному неорієнтованому графі $G=(V, E=S \cup C)$, де v_1 – окрема вершина-депо, час обслуговування s та час порожнього проходження c . Листоноші повинні пройти всі ребра для обслуговування з S , що мають час обслуговування s , таким чином, щоб надати обслуговування кожній з них хоча б раз. Цільова функція – мінімізація сумарної вартості всіх шляхів.

2.1.7. Задача мінімаксу k -китайських листонош (Min-Max k -Chinese Postman Problem)

З економічної точки зору, мінімізація сумарної вартості всіх маршрутів кількох листонош, відповідно до формулювання задачі k -СРР, не є достатньою. В найгіршому випадку, один листоноша буде вимушений обслуговуючи пройти всі ребра, в той час

як інші листоноші будуть лише виходити з депо і повертатися одразу назад. Тому бажано, щоб маршрути для всіх листонош були схожої довжини.

Фредеріксон (Frederickson) та ін. запропонували задачу мінімаксу k -китайських листонош (the Minimum-Maximum (MM) k -CPP). Задача мінімаксу k -китайських листонош може бути описана за допомогою зв'язаного неорієнтованого графу з окремою вершиною-депо v_1 та фіксованої кількості листонош, які мають обійти множину зважених ребер (з визначеним часом обслуговування) як у задачі k -CPP [25]. Але в даному випадку обхід з мінімальною сумарною вартістю всіх маршрутів не є цільовою функцією. Натомість вимагається мінімізувати вартість маршруту, що має максимальну вартість. Як можна побачити, кращий розподіл робочого часу поміж листонош – цільова функція цієї модифікованої задачі k -китайських листонош.

2.1.8 Задача мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (Min-Max Capacitated k -Chinese Postman Problem with Deadline Classes)

Задача мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (MM k -CCPP with DC) виникає у тому випадку, коли множина ребер та дуг $F = E \cup A$ не однорідна, а складається з n підмножин ребер та дуг, що мають різні класи пріоритетів. Тобто $F = F_1 \cup F_2 \cup F_n$, $\forall i, j, i \neq j: F_i \cap F_j = \emptyset$. Кожному класу пріоритетів встановлено у відповідність директивний термін, тому ребра з F_1 мають бути обслужені до часу T_1 , ребра з F_2 до часу T_2 і т.д. Існує $k > 1$ листонош, що обслуговують ребра. Для того, щоб забезпечити рівномірну завантаженість листонош (уникнути «простоїв»), задається додаткова умова зменшення найдовшого маршруту. Крім того, встановлюється обмеження на те, що сума вимог для кожного маршруту не має перевищувати місткості машини Q .

2.2 Змістовна постановка задачі

Інформаційна система планування обслуговування доріг призначена для складання плану проходження доріг районів міста спеціалізованою дорожньою технікою під час проведення робіт на дорогах. З розглянутого вище матеріалу та

спираючись на [9] можна зробити висновок, що саме за допомогою вирішення задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (ММ k -ССРР with DC) можна забезпечити потреби інформаційної системи планування обслуговування доріг міста. Змістовна постановка задачі зображена на рисунку А.5.

Інформаційна система передбачає наявність $k \geq 2$ машин, розташованих в одному депо, з якого вони починають та завершують свою роботу. В одному районі обслуговування може бути лише одне депо. Кожна дорога має свій час обслуговування та час транзитного проходження, причому час обслуговування залежить від виду робіт, що потрібно виконати. Обслуговування машинами кожної з доріг має відбуватися за розкладом з i до конкретно визначеного часу. Кожна машина характеризується певною місткістю. Вважається, що машини, які виконують визначений вид робіт, однакові, тому відповідно мають однакову місткість.

Район міста в системі представляється у вигляді мережі доріг, для яких передбачено визначення пріоритету щодо проведення на них робіт. Таким чином, множина доріг не однорідна, а складається з декількох підмножин, що мають різні класи пріоритетів, яким відповідають директивні терміни. Класи пріоритетів дозволяють співвіднести певні дороги з часовими обмеженнями та задають умову, що дороги з більш важливого класу будуть обслуговуватися в першу чергу. Дороги з нижчим пріоритетом також можуть обслуговуватися під час надання обслуговування дорогам з більш важливого класу, проте це не має порушувати умови директивних термінів.

Завданням є знаходження маршрутів для k одиниць дорожньої техніки, причому таких, щоб розподіл робочого навантаження між ними був приблизно однаковий.

На швидкість переміщення дорожньої техніки впливає рівень завантаженості дорожнього руху. Тож, в разі підвищення інтенсивності дорожнього руху та утворення заторів на певних шляхах, інформаційна система повинна будувати маршрути з врахуванням таких ситуацій задля пришвидшення процесу обслуговування доріг як перед початком виконання робіт, так і корегувати маршрути

під час виконання робіт, якщо виникне така необхідність у зв'язку зі зміною ситуації на дорогах.

2.3 Математична модель

Введемо позначення, які будуть застосовуватись для формулювання математичної постановки задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (ММ k -CCPP with DC). Модель задачі була розроблена на основі математичної постановки, яка описана в [26].

$G = (V, A = S \cup C)$ – граф задачі, де S – множина дуг для обслуговування, C – множина транзитних дуг.

$V = \{v_0, v_1, v_2, \dots\}$ – множина вершин, де $v_0 \in V$ – вершина-депо.

$A = \{a_1, a_2, \dots\}$ – множина дуг, які з'єднують всі вершини з V . Кожній дузі a_i у відповідність встановлені наступні параметри: b_i – вершина, з якої дуга починається, e_i – вершина, в якій дуга закінчується, s_i – час, за який машина обслуговує дугу, та c_i – час транзитного проходження по дузі, r_i – обсяг ресурсу, який необхідний для обслуговування дуги. Обмеження $s_i \geq c_i$ є очевидним.

$D = \{d_1, d_2, \dots\}$ – множина балів, де $d_i \in [1 \dots 10]$ та $d \in Z$, що визначає рівень завантаженості руху по дузі. Для кожної дуги (дороги) $a_i \in A$ встановлюється відповідний їй бал завантаженості дорожнього руху $d_a = d_i$, де $d_i \in D$. Кожний бал завантаженості дорожнього руху d_i характеризується параметром z_i – коефіцієнтом рівня інтенсивності дорожнього руху, який корегує час обслуговування або транзитного проходження дуги a в залежності від балу завантаженості ($s_a^* = s_a z_i$ та $c_a^* = c_a z_i$).

Множина S поділяється на підмножини наступним чином: $S_1 \cup \dots \cup S_n = S$ і $\forall i_1, i_2, i_1 \neq i_2 : S_{i_1} \cap S_{i_2} = \emptyset$.

Підмножині S_i відповідає певний директивний термін T_i , який визначає те, що кожна дуга з множини S_i повинна бути обслужена до встановленого часу T_i .

Вважатимемо, що послідовність директивних термінів задана наступним відношенням: $T_1 < T_2 < \dots < T_n$.

Потрібно знайти P_1, \dots, P_k маршрутів для k одиниць спеціалізованої дорожньої техніки (далі – машин). Маршрут P – це упорядкована множина дуг (доріг) $a_{[1]}, a_{[2]}, \dots, a_{[i]}, \dots, a_{[l]}$, в якій кожна наступна дуга починається з кінця попередньої $e_{[i-1]} = b_{[i]}$, де i відповідає порядковому номеру дуги у маршруті. Також на всі маршрути накладається наступна умова: всі маршрути мають починатись та закінчуватись у вершині-депо v_0 . Таким чином, $b_{[1]} = v_0$ і $e_{[l]} = v_0$.

Щоб спростити подальші математичні визначення, позначимо кожную дугу $a_j \in A$ її порядковим номером j в множині A та маршрути P_i через i .

Визначимо через $x_{ijh} \in \{0,1\}$ маркер обслуговування i -ою машиною (i -ий маршрут) j -ої дуги в позиції маршруту h . Тобто $x_{ijh} = 1$ у тому випадку, коли j -та дуга обслуговується в i -му маршруті на позиції h , в інших випадках $x_{ijh} = 0$. З цього слідує, що $x_{ij} = \sum_h x_{ijh}$ визначає обслуговування j -ої дуги i -ою машиною (коли $x_{ij} = 1$, то j -та дуга обслуговується в i -му маршруті, а коли $x_{ij} = 0$, то навпаки j -та дуга не обслуговується в i -му маршруті). Зважаючи на те, що кожна дуга повинна бути обслужена на одному з маршрутів, при цьому не більше одного разу, то на задачу накладається дане обмеження $\forall j: \sum_i x_{ij} = 1$.

Окрім обслуговування, машини також можуть здійснювати транзитні проходження по дугам. Кількість цих «порожніх» проходжень по дузі не обмежена. Визначимо через $y_{ijh} \in \{0,1\}$ транзитне проходження i -ою машиною j -ої дуги в позиції маршруту h . Тоді, $y_{ij} = \sum_h y_{ijh}$ дорівнює транзитному проїзду по j -ій дузі i -ою машиною.

Одночасне обслуговування і транзитне проходження дуги в одній позиції певного маршруту неможливе. Тому, якщо $x_{ijh} = 1$, то відповідно $y_{ijh} = 0$, і навпаки,

якщо $x_{ijh} = 0$, то $y_{ijh} = 1$. У іншому вигляді ця умова записується таким чином:

$$\forall i, j, h: x_{ijh} + y_{ijh} = 1.$$

Кожна з машин, що обслуговує дуги, характеризується місткістю Q . Зрозуміло, що сума кількості ресурсів, яка необхідна для обслуговування кожного маршруту, не має перевищувати місткості машини. Математично це обмеження можна записати у наступному вигляді: $\forall i: \sum_j r_j x_{ij} \leq Q$.

Так як дуга обслуговується тільки на одному з маршрутів, то для неї можна однозначно встановити i^* – номер маршруту, в якому вона обслуговується, та h^* – позиція (порядковий номер) дуги у маршруті i^* .

З використанням введених позначень визначимо $t_j = \sum_{h=1}^{h^*} (c_j z_j y_{i^*jh} + s_j z_j x_{i^*jh})$ як час завершення обслуговування j -ої дуги. Наведена вище умова є умовою для виконання директивних термінів та може бути описана у такому вигляді: $\forall a_j \in S_i: t_j \leq T_i$.

Час проходження i -ого маршруту визначається так: $W_i = \sum_j (s_j x_{ij} z_j + c_j y_{ij} z_j)$.

Враховуючи сформовані позначення визначимо математичну модель для задачі ММ k -ССРР with DC.

Цільова функція:

$$\max(W_i) \rightarrow \min, i = \overline{1, k} \quad (2.5)$$

Обмеження:

$$\forall j \in A: s_j \geq c_j \quad (2.6)$$

$$\forall P, i = \overline{1, l}: e_{[i-1]} = b_{[i]}, b_{[1]} = v_0, e_{[l]} = v_0 \quad (2.7)$$

$$\forall j: \sum_i x_{ij} = 1 \quad (2.8)$$

$$\forall i, j, h: x_{ijh} + y_{ijh} = 1 \quad (2.9)$$

$$\forall i: \sum_j r_j x_{ij} \leq Q \quad (2.10)$$

$$\forall j \in S_i: t_j \leq T_i \quad (2.11)$$

$$x_{ijh} \in \{0, 1\}, y_{ijh} \in \{0, 1\} \quad (2.12)$$

2.4 Огляд методів розв'язання

Нижче розглянуто деякі з підходів до розв'язання задач маршрутизації по дугам. Майже всі з них – евристичні та метаевристичні, оскільки точні алгоритми дуже рідко дають розв'язок за прийнятний час на великій розмірності задачі.

Точні методи – такі алгоритми перебирають всі чи майже всі розв'язки, поки не буде знайдено оптимальний розв'язок. Відомі застосування методу гілок та меж (Branch and Bound) і методу гілок та відтинань (Branch and Cut). Але в цих роботах час розв'язання задачі досить великий – десятки хвилин, а для великих задач – години. Тому можливість застосування точних методів обмежена, особливо в задачах, в яких планування маршрутів відбувається в режимі реального часу.

Евристичні методи – в цих алгоритмах відбувається відносно обмежений пошук в просторі розв'язків і зазвичай знаходяться хороші розв'язки за прийнятний час. Деякі з алгоритмів поступово будують розв'язок, беручи до уваги загальну вартість.

Метаевристики – в таких методах ретельно досліджуються найбільш перспективні частини простору розв'язків. Якість отриманих розв'язків зазвичай вища, ніж в отриманих класичними евристичними методами. Саме метаевристичні методи є найбільш перспективними методами розв'язання складних задач типу ARP.

2.4.1 Алгоритм розв'язання задачі китайського листоноші з пріоритетами (CPP with priorities)

Відомі методи розв'язання задачі китайського листоноші з пріоритетами (PCCPP – Priority Constrained Chinese Postman Problem): алгоритм Флері (Fleury) для побудови ланцюга Ейлера та алгоритм Дейкстри (Dijkstra) для знаходження найкоротшого маршруту [22]. Коротко процедуру алгоритму можна описати наступним чином. По-перше, як і для звичайної задачі про китайського листоношу, визначається множина непарних і до початкового графу додаються найкоротші шляхи між парами вершин з дублюванням ребер за необхідності для отримання графа з парними вершинами. Позначимо отриманий граф через G_0 . На другій фазі будується ланцюг Ейлера на новому графі. Спочатку будується маршрут P_1 з депо до першої

пріоритетної вершини v_1 відповідно до наступного правила: це один з маршрутів з v_0 до v_1 , що є найкоротшим за умови, що G_0 без P_1 залишається зв'язаним. Позначимо $G_1 = G_0 - P_1$. Тоді маршрут P_2 обирається так, щоб найкоротший маршрут з v_1 до v_2 був таким, щоб граф $G_2 = G_1 - P_2$ залишався зв'язаним. Якщо вершина v_2 вже була відвідана, то P_2 є порожнім. Таким чином всі пріоритетні вершини будуть відвідані і, нарешті, маршрут до v_0 будується так, щоб обійти всі ребра з G_k .

Використовуючи таку побудову, буде знайдено оптимальне вирішення даної задачі.

2.4.2 Алгоритми розв'язання задачі k -китайських листонош (k -СРР)

2.4.2.1 Алгоритм планування Longest Processing Time (LPT)

Алгоритм планування LPT був запропонований і проаналізований Грехемом (Graham). Ціллю є визначення r незалежних задач для n однакових блоків обробки таких, щоб період виготовлення ω (сумарний час, необхідний для закінчення всіх завдань) був мінімальним [25].

Схема LPT спочатку сортує завдання у порядку спадання часу їхнього виробництва. Тоді кожне завдання призначається до оброблювального блоку, що має найменшу завантаженість.

Пропонується планувати обслуговування ребер листоношою, використовуючи LPT, при застосуванні цієї евристики до вирішення k -СРР. В LPT вартість ребра визначається не лише вартістю проходження ребра, але також і вартістю проходження від та до вершини-депо, тобто є вартістю найкоротшого шляху.

Звідси, LPT вартість ребра $s \in S$ визначається наступним чином:

$$LPT \text{ cost}(s) = \text{cost}(SP(v_1, v_i)) + s(v_i, v_j) + \text{cost}(SP(v_j, v_1)) \quad (2.13)$$

Сумарна вартість маршруту для кожного листоноші може бути легко обчислена як сума LPT вартостей ребер, що він має обслужити.

Алгоритм LPT планування забезпечує те, що всі листоноші мають маршрути схожої довжини. Однак після обслуговування кожного ребра листоноша повертається до депо замість того, щоб використати (можливий) короткий шлях до наступного

ребра-обслуговування. Тому обчислення вартості поточного маршруту в LPT алгоритмі має бути модифікованим так, щоб відображати такий короткий шлях. Також, якщо листоноша не повертається в депо щоразу, порядок, в якому обслуговуються ребра, має визначальний вплив на вартість маршруту і має бути прийнятий до уваги.

2.4.2.2 Метод гілок та меж (Branch and Bound)

Для розв'язання k -CSP може бути використано метод гілок та меж. Формулювання задачі складається з нижньої межі, верхньої межі та призначення ребер обслуговування відповідним листоношам. Перші ребра обслуговування зафіксовані, тобто їхнє призначення залишається незмінним для всіх наступних підзадач. На першому етапі лише ребро s_1 зафіксоване за листоношою номер один. Всі варіанти розв'язання задачі визначають дерево гілок та меж. На кожній ітерації поточне рішення з найменшою нижньою межею обробляється наступним чином. Спочатку ребро розгалуження s_{f+1} визначається з відсортованої множини ребер обслуговування S . Після чого створюється $\varphi = \min(f+1, k)$ підзадач за рахунок призначення ребра розгалуження s_{f+1} листоноші i , де $i = 1, \dots, \varphi$ і позначення його як зафіксованого. Ребра, що залишились, перегруповуються з використанням LPT і розраховуються результуючі нижні та верхні межі. В решті решт, підзадачі додаються до дерева відповідно до значення їхньої нижньої межі.

2.4.2.3 Генетичний алгоритм

Генетичний алгоритм – це метаевристичний алгоритм, заснований на популяціях, які аналогічні до тих, що виникають у природному середовищі. В ході алгоритму формується група популяцій (розв'язків), що складається із незалежних один від одного варіантів розв'язання, для яких обчислюється фітнес-функція та відбувається відбір найкращих розв'язків. Завдяки цьому оцінюється одночасно декілька точок із множини розв'язків і це дозволяє в кінцевому результаті досягти майже оптимального рішення.

В умовах розв'язання k -СРР цільова функція цієї задачі перетворюється на фітнес-функцію генетичного алгоритму, що дозволяє мінімізувати кількість транзитних проходжень. Хромосоми з кращої популяції помножуються, а з гіршої – видаляються. Новий нащадок створюється з вибраних випадковим чином батьківських хромосом, для якого розраховується значення фітнес-функції. В ході алгоритму маршрути вдосконалюються шляхом заміни транзитної дуги на обслуговуючу, що були отримані на попередньому кроці. Це відповідає процесу мутації генетичного алгоритму. Якщо дуги були включені до іншого шляху, то вони видаляються із попереднього по принципу хромосом із найгіршої популяції [27].

2.4.3 Алгоритми розв'язання мінімаксу задачі k -китайських листонош (ММ k -СРР)

2.4.3.1 Алгоритм Фредеріксона

Стратегією евристики Фредеріксона-Хетча-Кіма (Frederickson-Hecht-Kim ФНК) є визначення єдиного маршруту (маршрут для одного листоноші), що покриває всі ребра $e \in E$, з подальшим поділом маршруту на k частин.

Знаходження маршруту одного листоноші – це добре відома задача про китайського листоношу (СРР), що може бути вирішена за допомогою алгоритму Едмундса (Edmonds) та Джонсона (Johnson).

Маршрут одного листоноші R , обчислений на першому кроці, може бути поділений наступним чином: спочатку визначаються $k-1$ так званих *вершин поділу* на R так, щоб вони позначали частини маршруту R , що мають приблизно однакову довжину. Потім k маршрутів будуються шляхом з'єднання цих частин початкового маршруту з вершиною-депо шляхами найменшої довжини [17].

2.4.3.2 Алгоритм приросту злиттям

Ідея алгоритму приблизно наступна. Будується замкнутий шлях C_e для кожного ребра $e = \{v_i, v_j\} \in E$, що складається з ребер найкоротшого шляху між вершиною-депо v_1 та v_i , власне ребра e та ребер з найкоротшого шляху між v_j та v_1 , тобто $C_e = (SP(v_1, v_i), e, SP(v_j, v_1))$. Потім два замкнуті шляхи зливаються для отримання

невеликої і збалансованої ваги маршруту, поки не було отримано k маршрутів. Деталізований опис алгоритму наведено нижче [28].

Алгоритм: Приріст злиттям

Крок 1. Відсортувати ребра e у порядку спадання їхньої ваги $w(C_e)$.

Крок 2. В порядку спадання $w(C_e)$ для кожного $e = \{v_i, v_j\} \in E$ створити замкнутий шлях $C_e = (SP(v_1, v_i), e, SP(v_j, v_1))$, якщо e ще не покрито вже існуючим маршрутом. Нехай $C = (C_1, \dots, C_m)$ результуюча множина маршрутів. Зверніть увагу, що маршрути відсортовані за їхніми довжинами, тобто $w(C_1) \geq w(C_2) \geq \dots \geq w(C_m)$.

Якщо $m \leq k$, то отримали оптимальний маршрут для k листонош і зупиняємо подальші обчислення.

Якщо $m < k$, то додаємо $k - m$ «фіктивних» маршрутів до C , кожен з яких складається з ребер, що мають вагу, яка є вдвічі меншою за початкову.

Крок 3. Поки $|C| > k$ з'єднуємо маршрут P_{k+1} з маршрутом з послідовності C_1, \dots, C_k так, щоб вага об'єднаного маршруту була мінімальною.

2.4.3.3 Алгоритм кластеризації

Цей алгоритм базується на підході спочатку кластеризація – потім маршрутизація. На першому етапі ділимо множину ребер E на k кластерів і на другому етапі будуємо маршрути для кожного кластера [28].

Повертаючись до другого етапу алгоритму приросту злиттям, було помічено, що для деяких ребер e потрібно створити маршрут C_e , а для інших ребер ні, оскільки вони вже присутні у існуючих маршрутах. Позначимо перший тип ребер як *критичні ребра*, оскільки необхідно врахувати їх обслуговування, в той час як другий тип ребер можна класифікувати як *некритичні*, оскільки вони входять до складу найкоротших маршрутів, що з'єднують критичні ребра та вершину-депо.

Базуючись на цьому спостереженні спочатку виконується k -кластеризація критичних ребер на множини ребер F_1, \dots, F_k . Після цього кожен кластер F_i доповнюється ребрами з найкоротших маршрутів, що з'єднують критичні ребра і

депо. Етап маршрутизації складається з обчислення оптимального маршруту для одного листоноші для кожного підграфа, представленого множиною F_i .

2.5 Розробка алгоритму розв'язання ММ k -ССРР with DC

Процес знаходження k маршрутів для задачі мінімаксу k -китайських листонош з врахуванням директивних термінів та місткості схематично може бути представлений етапами, що описані нижче [29].

Етап 1. Знаходження «простих» маршрутів відповідно до кількості дуг, де «простий» маршрут представляє собою найкоротший шлях від вершини-депо до однієї з дуг обслуговування, власне дуги обслуговування та найкоротший шлях від дуги назад до вершини-депо, а довжини дуг є часом транзитного проходження. Найкоротші шляхи між вершинами знаходяться за допомогою алгоритму Дейкстри.

Етап 2. Злиття маршрутів для отримання допустимого розв'язку задачі, тобто побудова k маршрутів з дотриманням обмежень на директивні терміни виконання робіт та місткість машин.

Етап 3. Об'єднання маршрутів так, щоб при цьому час проходження об'єднаного маршруту був мінімальним та виконувались обмеження на директивні терміни виконання робіт і місткість машин. Завдяки цьому покращується розв'язок задачі через врівноваження найкоротших і найдовших маршрутів.

У наведеному далі описі алгоритму кожен з етапів розглянуто детальніше. Отриманий в результаті розв'язок може бути використаний в прикладних задачах, таких як інформаційна система планування обслуговування доріг, в такому вигляді, як наведено. Блок-схема алгоритму розв'язання задачі зображена на рисунку А.6.

2.5.1 Опис алгоритму

Розглянемо детальніше алгоритм розв'язання задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (ММ k -ССРР with DC). Алгоритм розв'язання задачі був розроблений шляхом модифікації алгоритму побудови допустимих розв'язків, що наведений у [30].

Вхідні дані:

- $G = (V, A = S \cup C)$ – граф задачі;
- $V = \{v_0, v_1, v_2, \dots\}$ – множина вершин, де $v_0 \in V$ – вершина-депо;
- $A = \{a_1, a_2, \dots\}$ – множина всіх дуг, яка з'єднує між собою вершини з V ;
- S – множина дуг для обслуговування;
- C – множина транзитних дуг.

Введемо для спрощення деякі умовні позначення, які будуть надалі використовуватися у алгоритмі:

- b_i – вершина, де починається дуга;
- e_i – вершина, де закінчується дуга;
- s_i – час обслуговування;
- c_i – час транзитного проходження;
- z_i – коефіцієнт інтенсивності руху на дузі;
- r_i – обсяг ресурсу, який необхідний для обслуговування дуги a_i ;
- t_i – час завершення обслуговування дуги a_i ;
- T_i – визначений директивний термін ($t_i \leq T_i$);
- Q – місткість однієї машини.

Вихідні дані: $l \leq k$ допустимих маршрутів, де l – кількість маршрутів, отриманих в результаті роботи алгоритму, якщо можливо знайти допустимий розв'язок, у якому час обслуговування кожної дуги відповідає директивним термінам виконання робіт та обсяг необхідних ресурсів для кожного з маршрутів не перевищує місткості машини.

Крок 0. Визначаємо рівень завантаженості дорожнього руху d_a для всіх дуг $a_i \in A$. Кожний бал d_a характеризується параметром z_i – коефіцієнтом інтенсивності дорожнього руху ($z_i \geq 1$). Час обслуговування дуги s_j та час транзитного проїзду c_j залежать від рівня завантаженості дорожнього руху та корегується за допомогою даного коефіцієнту z_j на наступних кроках алгоритму.

Крок 1. Для кожної дуги a будемо замкнутий шлях P_a , який складається з найкоротшого шляху від вершини-депо v_0 до b , дуги a та найкоротшого шляху від e до v_0 , тобто $P_a = (SP(v_0, b), a, SP(e, v_0))$. Щоб знайти найкоротший шлях $SP(v_i, v_j)$, $\forall v_i, v_j \in V$, використовується час транзитного проходження c_a по дугам, які входять до маршруту.

Після побудови таких маршрутів для кожної дуги та маршруту P_a визначаємо такі величини:

- $w(P_a) = \sum_{a \in SP(v_0, b)} c_a z_a + s_a z_a + \sum_{a \in SP(e, v_0)} c_a z_a$ – час проходження маршруту P_a ;
- $t_a = \sum_{a \in SP(v_0, b)} c_a z_a + s_a z_a$ – час завершення обслуговування дуги a ;
- $\Delta_a = T_a - t_a$ – залишок часу після обслуговування дуги a ;
- $q(P_a) = \sum_{a \in S} r_a$ – необхідний обсяг ресурсу для обслуговування маршруту P_a .

Якщо дуга a є транзитною дугою у маршруті, тоді $r_a = 0$.

Якщо існує така дуга a , обслуговування якої не вкладається в допустимий термін $\exists \Delta_a < 0$ або для обслуговування якої недостатньо ресурсів $\exists q(P_a) > Q$, то закінчуємо обчислення, оскільки допустимий розв'язок знайти неможливо.

Якщо $|P| \leq k$, то знайдені шляхи є оптимальним розв'язком задачі, тому припиняємо обчислення.

Крок 2. Використовуємо отримані «прості» маршрути для їх злиття у початкові маршрути.

Крок 2.1. Обираємо дугу a , для якої значення $\Delta_a > 0$ є найменшим.

Крок 2.2. Обираємо серед транзитних дуг маршруту дугу $\hat{a} \in SP(v_0, b)$ найближчу з тих, що з'єднують v_0 з дугою a так, щоб включити її в маршрут як дугу-обслуговування замість транзитної, при цьому виконуючи умови стосовно директивних термінів ($s_a - c_a \leq \Delta_a$) та необхідних ресурсів ($q(P_a) + r_{\hat{a}} \leq Q$). Дуга \hat{a} повинна входити як дуга-обслуговування до «простого» маршруту і тільки до нього.

Крок 2.3. Транзитивну дугу \hat{a} замінюємо на дугу обслуговування у маршруті P_a , після чого видаляємо «простий» маршрут P_a . Перераховуємо значення $w(P_a)$ та $q(P_a)$ для створеного нового маршруту, а також всі значення $t_a, t_a, \Delta_a, \Delta_a$ для переглянутих дуг a, \hat{a} .

Якщо $|P| \leq k$, то завершуємо роботу алгоритму, адже отримали допустимий розв'язок задачі.

Перебираємо всі транзитні дуги $\hat{a} \in SP(v_0, b)$, починаючи знову з кроку 2.2.

Крок 2.4. Завершуємо роботу з дугою a з кроку 2.1 і повторюємо всі етапи кроку 2, поки не переглянемо всі інші дуги.

Крок 3. Нехай $P = \{P_1, \dots, P_m\}$ – це результуюча множина маршрутів, отримана після виконання всіх етапів кроку 2. Відсортуємо маршрути в порядку зменшення часу їхнього проходження: $w(P_1) \geq w(P_2) \geq \dots \geq w(P_m)$.

Переглядаємо всі маршрути у множині P , починаючи з кінця, тобто з маршруту найменшої вартості або іншими словами маршруту з найкоротшим часом проходження.

Крок 3.1. Нехай P_i – обраний поточний маршрут. Знаходимо серед маршрутів P_1, P_2, \dots, P_{i-1} такий маршрут, який буде містити всі дуги маршруту P_i . Цей маршрут позначимо як P' .

Якщо такий маршрут P' існує, то переходимо на наступний крок. Інакше вибираємо наступний маршрут та знову повторюємо цю процедуру.

Якщо переглянуто всі маршрути, то переходимо на крок 4.

Крок 3.2. На основі маршруту P' створюємо новий маршрут, де дуга a буде дугою-обслуговування в тому випадку, коли це дуга обслуговування хоча б в одному з розглянутих маршрутів P' або P_i , а інакше a буде транзитною дугою.

Перераховуємо всі значення для результуючого маршруту ($w(P_a)$, $q(P_a)$, t_a та Δ_a). Якщо всі значення $\Delta_a \geq 0$ та $q(P_a) \leq Q$, то замінюємо маршрут P' з

кроку 3.1 на маршрут, побудований на даному кроці. Далі переходимо на наступний крок. Інакше обираємо попередній маршрут P_{i-1} і знову переходимо на крок 3.1.

Крок 3.3. Маршрут P_i видаляємо з підмножини маршрутів $\{P_1, \dots, P_{i-1}\}$ та відсортовуємо її.

Якщо $|P| \leq k$ завершуємо роботу алгоритму, оскільки отримали допустимі маршрути для k машин. Інакше знову повертаємось на крок 3.1.

Крок 4. Нехай $P = \{P_1, \dots, P_m\}$ – це результуюча множина маршрутів, отримана після виконання всіх етапів кроку 3. Відсортуємо маршрути в порядку зменшення часу їхнього проходження, тобто $w(P_1) \geq w(P_2) \geq \dots \geq w(P_m)$.

Об'єднуємо маршрут P_{k+1} з іншим маршрутом з послідовності P_1, \dots, P_k поки виконується $|P| > k$ таким чином, щоб час проходження об'єднаного маршруту був мінімальним та при цьому виконувалась умова директивних термінів виконання робіт і не перевищувався обсяг наявних ресурсів. Для досягнення цього переходимо до наступних кроків.

Крок 4.1. Серед маршрутів з підмножини $WP = \{P_1, \dots, P_k\}$ обираємо такий маршрут P_i , для якого значення $w(P_{k+1}) + w(P_i)$ буде найменшим.

Розраховуємо значення $q(P_{k+1})$ та $q(P_i)$ для цих маршрутів. Якщо $q(P_{k+1}) + q(P_i) \leq Q$, то переходимо на наступний крок. Інакше переходимо на крок 4.4.

Крок 4.2. Від вершини v_i , в якій закінчується остання дуга обслуговування із маршруту P_{k+1} , до вершини v_j , з якої починається перша дуга обслуговування із маршруту P_i , знаходимо найкоротший шлях $SP(v_i, v_j)$.

Крок 4.3. Створюємо новий маршрут, який складається із частини маршруту P_{k+1} до вершини v_i , найкоротшого шляху $SP(v_i, v_j)$ та частини маршруту P_i від вершини v_j .

Перераховуємо всі значення для результуючого маршруту ($w(P_a)$, $q(P_a)$, t_a та Δ_a).

Якщо всі значення $\Delta_a \geq 0$, то замінюємо маршрут P_i на відповідне злиття, видаляємо маршрут P_{k+1} та переходимо до кроку 4.8. Інакше переходимо до кроку 4.5.

Крок 4.4. Зливаємо маршрути P_{k+1} і P_i у їх початковому вигляді без змін. Обчислюємо значення t_a та Δ_a для всіх дуг обслуговування створеного маршруту.

Якщо всі значення $\Delta_a \geq 0$, то замінюємо маршрут P_i на маршрут, побудований на даному кроці, перераховуємо вагу маршруту $w(P_a)$, необхідний обсяг ресурсів $q(P_a)$ та переходимо на крок 4.8.

Інакше якщо є хоча б одне значення $\Delta_a < 0$, то такий маршрут зробити неможливо. Переходимо на крок 4.7.

Крок 4.5. Від вершини v_i , в якій закінчується остання дуга обслуговування із маршруту P_i , до вершини v_j , з якої починається перша дуга обслуговування із маршруту P_{k+1} , знаходимо найкоротший шлях $SP(v_i, v_j)$.

Крок 4.6. Створюємо новий маршрут, який складається із частини маршруту P_i до вершини v_i , найкоротшого шляху $SP(v_i, v_j)$ та частини маршруту P_{k+1} від вершини v_j .

Перераховуємо всі значення для результуючого маршруту ($w(P_a)$, $q(P_a)$, t_a та Δ_a).

Якщо всі значення $\Delta_a \geq 0$, то замінюємо маршрут P_i на побудований маршрут, видаляємо маршрут P_{k+1} та переходимо до кроку 4.8. Інакше якщо хоча б одне зі значень $\Delta_a < 0$, то такий маршрут злити неможливо. В цьому випадку виключаємо маршрут P_i з множини маршрутів WP . Якщо $|WP| > 0$, повертаємось знову на початок до кроку 4.1, інакше припиняємо обчислення, оскільки допустимий розв'язок знайти неможливо.

Крок 4.7. Зливаємо маршрути P_i і P_{k+1} у їх початковому вигляді без змін. Обчислюємо значення t_a та Δ_a для всіх дуг обслуговування створеного маршруту.

Якщо всі значення $\Delta_a \geq 0$, то замінюємо маршрут P_i на маршрут, побудований на даному кроці, перераховуємо вагу маршруту $w(P_a)$, необхідний обсяг ресурсів $q(P_a)$ та переходимо на крок 4.8.

Інакше якщо хоча б одне зі значень $\Delta_a < 0$, то такий маршрут злити неможливо. В цьому випадку виключаємо маршрут P_i з множини маршрутів WP . Якщо $|WP| > 0$, повертаємось знову на початок до кроку 4.1, інакше припиняємо обчислення, оскільки допустимий розв'язок знайти неможливо.

Крок 4.8. Відсортовуємо множину маршрутів P в порядку зменшення часу їхнього проходження.

Складність розробленого алгоритму складає $O(n^3m^3)$, де n – кількість маршрутів, m – кількість дуг. Проте зазвичай на практиці значення n значно менше за m , таким чином, алгоритм за складністю можна вважати кубічним.

2.5.2 Алгоритм Дейкстри

На кроці 1 алгоритму побудови допустимого розв'язку (підрозділ 2.5.1) для знаходження найкоротших шляхів між двома вершинами використовується алгоритм Дейкстри.

Також даний алгоритм використовується після завершення роботи алгоритму, описаного у підрозділі 2.5.1, в разі значного сповільнення руху вздовж побудованих маршрутів, тобто коли рівень завантаженості руху на певній дузі дорівнює 6 та більше балів ($d_i \geq 6$). В цьому випадку побудовані маршрути корегуються для обходження такої дуги у графі. Інтенсивність дорожнього руху після початку обслуговування визначається для п'яти передуючих дуг. Оскільки інтенсивність дорожнього руху змінюється з плином часу, то це дозволяє уникати зайвих корегувань маршрутів у тих частинах, що не будуть досягнуті транспортним засобом у найближчий час.

Принцип, за яким відбувається корегування маршрутів, наступний:

- а) від поточного місцезнаходження транспортного засобу на дузі з вершиною v_i до дуги з вершиною v_j , для якої $d < 6$ (якщо між цими дугами немає дуг обслуговування);
- б) від поточного місцезнаходження транспортного засобу на дузі з вершиною v_i до наступної дуги-обслуговування з вершиною v_j (якщо для наступної дуги обслуговування $d_i \geq 6$).

Якщо час обслуговування скорегованого маршруту більше або дорівнює часу обслуговування початкового маршруту, що включає дугу з ускладненим дорожнім рухом, або якщо дузі з ускладненим рухом необхідно надати обслуговування, то маршрут залишається незмінним [29].

Нижче наведено опис алгоритму Дейкстри [31].

Дано зважений орієнтований граф $G(V, A)$ без петель та дуг з від'ємними вагами. Знайти найкоротший шлях від деякої вершини v_1 до вершини v .

Введемо наступні позначення: U – множина відвіданих вершин, $d[v]$ – по завершенню роботи алгоритму дорівнює довжині найкоротшого шляху від вершини v_1 до вершини v , $p[v]$ – містить найкоротший шлях від v_1 до вершини v .

Крок 0. Відстань до початкової вершини прирівнюємо до нуля $d[v_1]=0$, а всі інші відстані заповнюємо великим додатнім числом (більшим максимально можливого шляху в графі).

Крок 1. Поки вершина v не належить множині відвіданих вершин U .

Крок 1.1. Знаходимо вершину v , що не належить множині U з мінімальним значенням відстані $d[v]$.

Крок 1.2. Додаємо вершину v до множини відвіданих вершин U .

Крок 1.3. Перевіряємо сусідні вершини до вершини v . Нехай u – сусідня вершина до вершини v . Якщо значення відстані $d[u]$ більше, ніж сума відстані до поточної вершини $d[v]$ та вартості ребра $w(v,u)$, то замінюємо значення $d[u]$ на цю суму і записуємо до найкоротшого шляху $p[u]$ значення $p[v]$ та u .

2.6 Результати досліджень ефективності алгоритму

Для перевірки ефективності роботи розробленого алгоритму розв'язання задачі ММ k -ССРР with DC проведемо декілька експериментів.

Експеримент №1. Виявимо залежність часу обслуговування доріг від кількості машин. Для експерименту були використані наступні вхідні дані: місце розташування депо, 30 доріг обслуговування та їх розподіл за пріоритетністю на 5 класів, вид робіт (в даному випадку обрано розчищення доріг від снігу), час початку робіт 08:00 годин та директивні терміни 09:00, 10:15, 10:45, 11:15, 11:30 годин відповідно до спадання важливості класів пріоритетів. Завантаженість дорожнього руху в даному експерименті є сталою та дорівнює 4 балам. З використанням інформаційної системи було проведено симуляцію процесу обслуговування доріг на однакових вхідних параметрах, описаних вище, але з використанням різної кількості машин. Результати експерименту наведено у таблиці 2.1.

Таблиця 2.1 – Час обслуговування доріг при різній кількості машин

| | М-т. № 1 | М-т. № 2 | М-т. № 3 | М-т. № 4 | М-т. № 5 | М-т. № 6 |
|-----------------|-----------------|----------|----------|-----------|-----------|---------------|
| 6 машин | Розв'язку немає | | | | | |
| 7 машин | 251 хв. | 147 хв. | 138 хв. | 135 хв. | 129 хв. | 125 хв. |
| 8 машин | 179 хв. | 147 хв. | 136 хв. | 135 хв. | 129 хв. | 127 хв. |
| 9 машин | 179 хв. | 135 хв. | 134 хв. | 128 хв. | 125 хв. | 120 хв. |
| 10 машин | 179 хв. | 134 хв. | 128 хв. | 124 хв. | 120 хв. | 108 хв. |
| 11 машин | 179 хв. | 128 хв. | 125 хв. | 120 хв. | 108 хв. | 95 хв. |
| | М-т. № 7 | М-т. № 8 | М-т. № 9 | М-т. № 10 | М-т. № 11 | Загальний час |
| 7 машин | 120 хв. | – | – | – | – | 1045 хв. |
| 8 машин | 119 хв. | 172 хв. | – | – | – | 1144 хв. |
| 9 машин | 108 хв. | 95 хв. | 146 хв. | – | – | 1170 хв. |
| 10 машин | 95 хв. | 85 хв. | 83 хв. | 135 хв. | – | 1191 хв. |
| 11 машин | 85 хв. | 83 хв. | 73 хв. | 69 хв. | 134 хв. | 1199 хв. |

Як можна побачити з таблиці 2.1, на заданих вхідних параметрах найменший загальний час обслуговування досягається при використанні 7 машин. Проте кращий

розподіл робочого навантаження між машинами досягається при використанні 8 машин, коли час проходження кожного з маршрутів приблизно однаковий. При подальшому збільшенні кількості машин загальний час обслуговування постійно зростає, хоча час проходження кожного з маршрутів зменшується. Це зумовлено тим, що хоч кожна з машин і виконує менше робіт на дорогах, проте час транзитних проходжень зростає, оскільки збільшується кількість машин, яким потрібно спершу дістатися своїх дуг обслуговування, а потім повернутися в депо. На даних вхідних параметрах оптимальне використання 8 машин. Хоча загальний час в цьому випадку трохи більший, ніж при використанні 7 машин, але час роботи у машин більш рівномірно розподілений, тому це дозволить ефективніше розпоряджатися ресурсами та зменшити зношування техніки.

Експеримент №2. Виявимо залежність часу обслуговування від завантаженості дорожнього руху. Для експерименту були використані наступні вхідні дані: місце розташування депо, 15 доріг обслуговування та їх розподіл за пріоритетністю на 5 класів, 5 машини, вид робіт (в даному випадку обрано очищення від бруду), час початку робіт 7:00 годин та директивні терміни 9:40, 11:15, 14:50, 16:20, 18:40 годин відповідно до спадання важливості класів пріоритетів. Симуляцію процесу обслуговування доріг було проведено при наступних максимально можливих значеннях завантаженості дорожнього руху: 6 балів, тобто коли рух ускладнений, 7 балів, що відповідає серйозним заторам, 8 балів, коли утворюються багатокілометрові затори, 9 балів, що відповідає ситуації, коли місто стоїть. Результати експерименту наведено у таблиці 2.2.

Таблиця 2.2 – Час обслуговування при різній завантаженості дорожнього руху

| Максимально можливе завантаження руху | Час обслуговування без корегування маршрутів | Час обслуговування з корегуванням маршрутів |
|--|---|--|
| 6 балів | 728 хв. | 717 хв. |
| 7 балів | 813 хв. | 801 хв. |
| 8 балів | 935 хв. | 862 хв. |
| 9 балів | 952 хв. | 941 хв. |

Як бачимо з таблиці 2.2, завдяки корегуванню маршрутів вдалося зменшити загальний час обслуговування, тобто можна зробити висновок, що врахування інтенсивності дорожнього руху у системі є доцільним. Оскільки наразі рівень завантаженості руху генерується випадковим чином та при кожному запуску системи утворюються нові значення, тому і отримане зменшення часу є приблизним. Незначна різниця у часі може бути спричинена тим, що ускладнений рух може виникати або на дорогах обслуговування, через що скорегувати маршрути неможливо, або на дорогах, які не проходять транспортні засоби.

Експеримент №3. Перевіримо швидкодію алгоритму. Для експерименту були задані наступні вхідні дані: місце розташування депо, розподіл доріг обслуговування за пріоритетністю на 5 класів, 10 машин, вид робіт (в даному випадку обрано збір сміття), час початку робіт 6:30 годин та директивні терміни 08:30, 10:30, 11:30, 11:55, 12:30 годин відповідно до спадання важливості класів пріоритетів. Максимальне значення завантаженості дорожнього руху в даному експерименті дорівнює 8 балам. Симуляцію процесу обслуговування було проведено на різній кількості обраних доріг обслуговування. Результати експерименту наведено у таблиці 2.3.

Таблиця 2.3 – Час роботи алгоритму при різній кількості доріг обслуговування

| Кількість дуг обслуговування | Час роботи алгоритму |
|-------------------------------------|-----------------------------|
| 15 | 0,0116 с |
| 30 | 0,0225 с |
| 40 | 0,0254 с |
| 50 | 0,0311 с |
| 60 | 0,0420 с |

Як бачимо з таблиці 2.3, алгоритм працює досить швидко на різній кількості вхідних параметрів. При збільшенні кількості доріг обслуговування час роботи алгоритму збільшується несуттєво та є прийнятним для роботи інформаційної системи планування обслуговування доріг.

Додаткові експерименти з інформаційною системою планування обслуговування доріг міста наведено у [32].

Висновки

В даному розділі було наведено огляд існуючих типів задач маршрутизації транспортних засобів по дугам. Зокрема в огляді та в подальшому аналізі основна увага була приділена типам задач про китайського листоношу. В процесі аналізу було знайдено такий тип задачі, який найкраще підходить для реалізації потреб інформаційної системи планування обслуговування доріг міста: задача мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин (MM k -CCPP with DC).

Також в розділі наведено змістовну постановку досліджуваної задачі та розроблено відповідну їй математичну модель. Були розглянуті методи розв'язання суміжних задач до досліджуваної. На базі розглянутих алгоритмів відбувалась розробка алгоритму розв'язання досліджуваної задачі MM k -CCPP with DC.

В розділі було описано результати проведення досліджень та зроблено аналіз отриманих результатів розробленого алгоритму. З проведених експериментів можна зробити висновок про те, що при побудові маршрутів транспортних засобів дотримується умова мінімаксу задачі, завдяки чому досягається більш рівномірний розподіл робочого навантаження між машинами, а також про те, що корегування маршрутів дозволяє зменшити загальний час обслуговування у деяких випадках. Час роботи алгоритму є швидким, тому використання на практиці інформаційної системи планування обслуговування доріг є доцільним.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

В якості мови розробки програмного забезпечення для реалізації інформаційної системи планування обслуговування доріг та алгоритму знаходження допустимого розв'язку мінімаксної задачі k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин було обрано мову програмування JavaScript, так як вона є найзручнішим рішенням для реалізації веб-застосунків. Для розробки серверної частини була обрана мова програмування Node.js. Як інструмент для розробки інтерфейсу користувача було вирішено використовувати бібліотеку React.js.

Цей вибір був здійснений виходячи з наступних міркувань:

- JavaScript може виконуватися як в браузері, так і на сервері;
- JavaScript має дуже високу швидкість роботи;
- JavaScript підтримується усіма сучасними браузерами;
- JavaScript має максимально зрозумілий синтаксис;
- Node.js забезпечує узгоджену поведінку коду;
- Node.js має автоматичне і високопродуктивне управління пам'яттю;
- Node.js, як і JavaScript, використовує асинхронну поведінку, що значно пришвидшує час виконання та обробки запитів;
- Node.js сервіси легко інтегруються у інші системи;
- React.js підвищує продуктивність високонавантажених веб-застосунків;
- React.js швидко працює, бо виконує оптимізацію при компіляції коду;
- React.js має безпеку типів, більшість помилок можуть бути виявлені під час компіляції.

У системі використано карту OpenStreetMap [33], для відображення якої застосовується бібліотека Leaflet [34]. OpenStreetMap є зручним ресурсом, оскільки має відкриті дані: їх можна використовувати для будь-яких цілей, якщо вказуються авторські права OpenStreetMap і його учасників. Карти охоплюють поверхню всієї земної кулі. Їх використання можливе через сам сайт OpenStreetMap та через його

API. Leaflet є сучасним відкритим проектом, що написаний на мові JavaScript, для відображення інтернет-карт. Ця бібліотека однаково добре працює і на десктопних браузерях, і на мобільних пристроях, є дуже швидкою та зрозумілою. Функціональність бібліотеки може бути розширена за допомогою добре задокументованого і зручного API, простого вихідного коду.

За допомогою Leaflet-бібліотеки з сервера OpenStreetMap завантажується легкомасштабуєме векторне зображення географічної карти. Навігація по карті здійснюється стандартними графічними веб-інтерфейсами. Відображення карти складається із маленьких фрагментів, поверх яких Leaflet дозволяє ще накладати додаткові шари. Шари можуть бути інтерактивними, наприклад, відображати підказку при кліку по маркеру.

Було обрано Mapbox Directions API [35] для створення маршрутизації транспортних засобів по карті та Google Sheets API [36] для формування онлайн документів з побудованими маршрутами. Mapbox Directions API дозволяє будувати маршрути руху транспортних засобів, пішої ходьби та їзди на велосипеді з урахуванням дорожнього трафіку та дорожньо-транспортних пригод. Google Sheets API надає повний доступ до електронних таблиць, дозволяє програмно зчитувати та записувати дані, форматовувати текст та багато іншого.

Для створення та керування базою даних обрано реляційну систему управління базами даних MySQL. MySQL є найбільш пристосованою для використання в середовищі веб-застосувань СУБД, бо є високоефективною та при цьому володіє високою надійністю. MySQL підтримує SQL – мову структурованих запитів.

Основні переваги MySQL:

- багатопоточність, підтримка декількох одночасних запитів;
- швидка робота, масштабованість;
- високий рівень безпеки;
- стабільність, важко вивести з ладу;
- безкоштовна в більшості випадків.

3.2 Архітектура програмного забезпечення

3.2.1 Структура класів програмного забезпечення

Детальна схема структурна класів інформаційної системи планування обслуговування доріг міста представлена на рисунку А.7. Далі коротко розглянемо класи, що представлені на цій схемі.

Основним класом інформаційної системи є клас Application, який контролює роботу всіх інших класів в системі, а також містить в собі сховище даних, яке ними використовується. Також цей клас відповідає за відображення та оновлення компонентів системи.

Клас Map займається відображенням географічної карти та активних доріг на ній, а також дозволяє обирати на карті дороги обслуговування. Цей клас пов'язаний із класом Route, що дозволяє будувати дороги по точках на карті. Точки задаються за допомогою класу Waypoint, який в свою чергу пов'язаний з класом LatLng, через який визначаються географічні координати (широта і довгота) цих точок. При цьому зберігаються не лише точки початку та кінця дороги, але і проміжні точки вздовж дороги для того, щоб в точності відтворити структуру дороги з усіма її вигинами.

Клас Graph відповідає за побудову графу задачі MM *k*-CCPP with DC та також має зв'язок із класом Map, оскільки вершини графа задаються по точкам доріг на карті, а саме по географічним координатам перетинів всіх доріг. Клас Arc створює дуги графа, з'єднуючи між собою точки вершин. Ці точки зберігаються у класі Point. Також клас Graph має зв'язок із класом AddRoutePopup, що відображає спливаюче вікно для вказання класу пріоритетів, до якого відноситься дорога обслуговування.

Клас Menu, що відповідає за відображення бокової інформаційної панелі меню з її вкладками, має зв'язок безпосередньо з класом Application. Цей клас пов'язується з чотирма іншими: ChoosenRoad, RoadSchema, CalculateDialog та SaveRoadSchemaDialog. Клас ChoosenRoad відповідає за вкладку меню «Обрані дороги» з переліком доданих доріг обслуговування, а клас RoadSchema за вкладку «Схеми доріг». Таким чином, клас SaveRoadSchemaDialog дозволяє зберігати схеми,

а клас CalculateDialog викликає спливаюче вікно для задання вхідних параметрів роботи алгоритму.

Класи SuccessDialog та ErrorDialog відповідно відображають повідомлення про успішне завершення роботи алгоритму та про те, що розв'язок знайти неможливо. Також клас SuccessDialog відповідає за початок симуляції руху транспортних засобів по графу.

3.2.2 Специфікація функцій програмного забезпечення

Так як кількість функцій програмного забезпечення досить велика, то у таблиці 3.1 наведено опис тільки основних функцій інформаційної системи планування обслуговування доріг міста.

Таблиця 3.1 – Специфікація функцій інформаційної системи планування обслуговування доріг

| Назва функції | Вхідні параметри | Опис функції |
|---------------------------|---|--|
| path | begin – початкова вершина, end – кінцева вершина | Реалізує алгоритм Дейкстри |
| simpleRoute | begin – початкова вершина, end – кінцева вершина | Формує маршрути на першому кроці алгоритму |
| routeBetweenVertex | begin – початкова вершина, end – кінцева вершина | Формує шлях між двома маршрутами на четвертому кроці алгоритму |
| kCCPP | graph – граф задачі, arcs – дуги для обслуговування, amountOfCar – кількість машин, Q – місткість однієї машини | Реалізує алгоритм розв'язання задачі MM k -CCPP with DC |
| getRouteIndexByServiceArc | routes – всі маршрути, arc – дуга з маршруту | Знаходить транзитні дуги, які можна включити в маршрут як дуги обслуговування на другому кроці алгоритму |

Продовження таблиці 3.1

| Назва функції | Вхідні параметри | Опис функції |
|-----------------------|--|--|
| isMergePossible | destRoute – маршрут для злиття, sourceRoute – початковий маршрут | Перевіряє чи задовольняють побудовані маршрути обмеженням задачі |
| getServiceArcIndex | paths – маршрут | Знаходить індекс дуги обслуговування у маршруті |
| recalculatePathParams | routes – всі маршрути | Перераховує значення обмежень для маршрутів |
| isPathConsistAllArcs | originRoute – маршрут, який включає дуги іншого, route – маршрут, який включається в інший маршрут | Перевіряє на третьому кроці алгоритму чи маршрут включає в себе всі дуги іншого маршруту |
| buildNewRoute | originRoute – маршрут, який включає дуги іншого, route – маршрут, який включається в інший маршрут | Будує новий маршрут на третьому кроці алгоритму шляхом включення одного маршруту в інший |
| isPossibleRoute | pathList – всі дуги маршрути, ignore – параметр, що вказує враховувати місткість машини чи ні | Перевіряє чи побудований на третьому кроці алгоритму маршрут задовольняє обмеженням задачі |
| connectPathList | pathListA – перший маршрут, pathListB – другий маршрут | Будує новий маршрут на четвертому кроці алгоритму шляхом об'єднання двох маршрутів у їх початковому вигляді без змін |
| combineThreePath | a – частина першого маршруту, b – шлях між дугами обслуговування | Об'єднує в один маршрут частину першого маршруту без частини повернення в депо зі |

Продовження таблиці 3.1

| Назва функції | Вхідні параметри | Опис функції |
|-------------------|--|---|
| | обох маршрутів, s – частина другого маршруту | шляхом від останньої дуги обслуговування першого маршруту до першої дуги обслуговування другого маршруту та частину другого маршрут без частини відправлення з депо |
| render | – | Відображає компонент системи |
| fetchRoads | – | Отримує список всіх доріг |
| resolveKCCPP | – | Отримує результат розв'язання задачі ММ k -ССРР with DC |
| createGraph | map – карта | Будує граф задачі на карті |
| removeGraph | map – карта | Видаляє граф задачі з карти |
| buildAllRoutes | routes – всі дороги, map – карта | Будує всі дороги на карті |
| buildRoute | map – карта, id – ідентифікатор дороги | Відображає дорогу на карті як активний елемент |
| buildServiceRoute | route – дорога обслуговування, map – карта | Позначає дорогу як дорогу обслуговування |
| handleAnimateCar | pathList – всі дуги маршруту | Симулює рух транспортних засобів по графу |
| setPriorityVal | priority – клас пріоритетів | Встановлює клас пріоритетів для дуги обслуговування |
| buildArc | arc – дуга | Будує дугу графу |
| setStyle | object – об'єкт стилів | Змінює стиль дуг графу під час симуляцію руху транспортних засобів |

Продовження таблиці 3.1

| Назва функції | Вхідні параметри | Опис функції |
|-----------------|------------------|---|
| removeGraph | – | Видаляє граф з карти |
| getPoints | – | Отримує точки початку і кінця дуги графу |
| simulate | – | Запускає симуляцію транспортних засобів після натискання на кнопку «Почати симуляцію» |
| recalculatePath | – | Перераховує маршрути в разі підвищення інтенсивності дорожнього руху |

3.3 Інструкція користувача

В даному розділі наведено інструкцію користувача щодо використання інформаційної системи планування обслуговування доріг міста.

Для початку роботи з інформаційною системою перейдіть за посиланням <https://kccpp.herokuapp.com>. На сторінці відображається карта, де виділено адміністративний район обслуговування, для якого при подальшій роботі буде складатися план робіт, та всі доступні дороги, що йому належать (рисунок А.8). Наразі в інформаційній системі демонструється Солом'янський район міста Києва, тобто складання плану обслуговування можливе лише для доріг цього району. Це зроблено для того, щоб показати можливості реального впровадження методу розв'язання задачі маршрутизації по дугам на практиці. В подальшому система може бути розширена для маршрутизації транспортних засобів по решті районів Києва чи будь-якого іншого міста.

Першим кроком у процесі складанні плану обслуговування є вибір місця розташування депо, з якого транспортні засоби будуть починати свої маршрути. Для цього необхідно просто натиснути на відповідне місце на карті. Депо буде позначено

маркером (рисунок 3.1). Для одного району обслуговування можливо визначити лише одне депо.

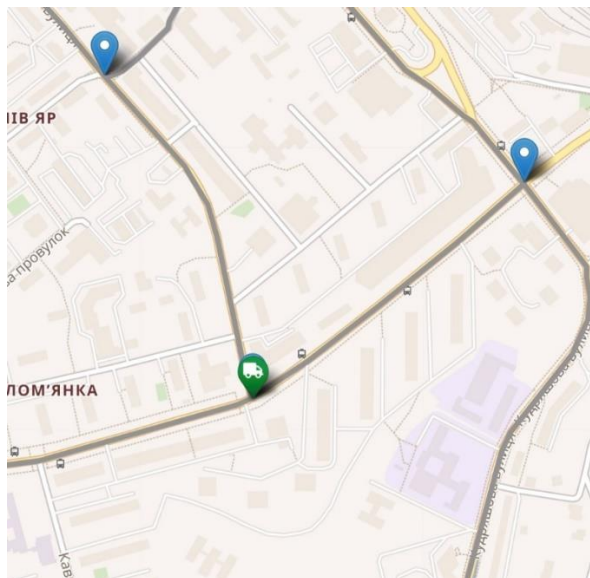


Рисунок 3.1 – Визначення розташування депо

Після визначення точки депо користувачу потрібно обрати принаймні одну дорогу, якій необхідно надати обслуговування. Для цього треба натиснути на потрібну дорогу. З'явиться спливаюче вікно, в якому необхідно вказати до якого класу пріоритетів відноситься обрана дорога (рисунок 3.2). Це поле є обов'язковим для заповнення. Після введення значення пріоритету для дороги, потрібно натиснути на кнопку «Додати дорогу». Зліва на інформаційній панелі у розділі «Обрані дороги» з'явиться назва доданої дороги та вказаний клас пріоритетів (рисунок 3.3). Користувач повторює цю послідовність дій доки не вибере усі необхідні для обслуговування дороги.

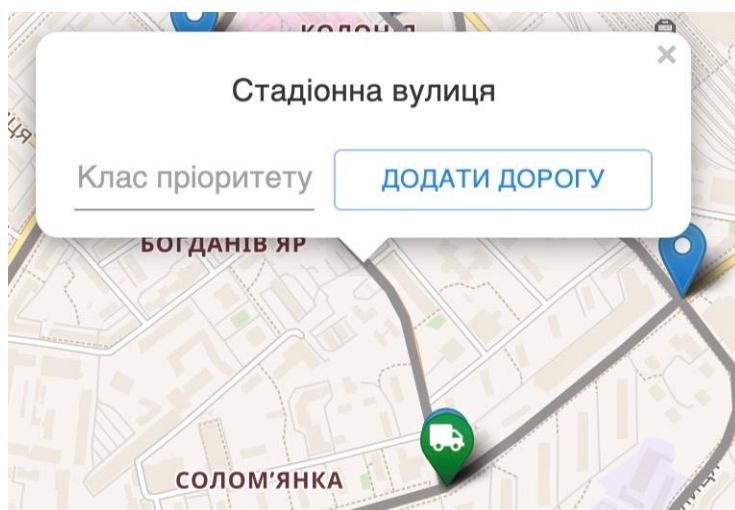


Рисунок 3.2 – Додання дороги

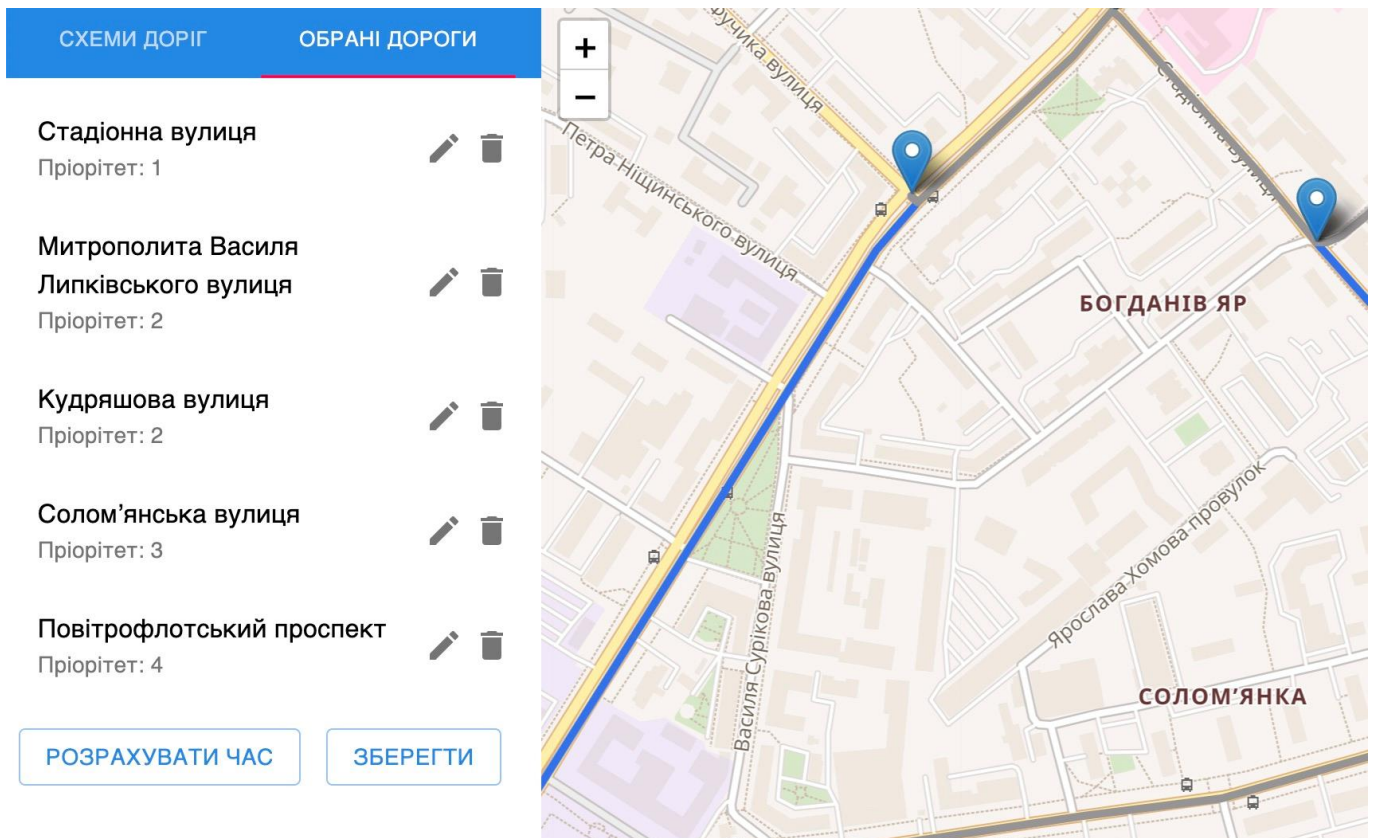


Рисунок 3.3 – Перелік усіх обраних доріг

Для того, щоб відредагувати клас пріоритету вже доданої дороги, треба натиснути на значок редагування, що відображається поряд з назвою вулиці. З'явиться текстове поле, як на рисунку 3.4, для введення нового значення пріоритету. Також за необхідності можна видалити додану дорогу обслуговування, натиснувши на значок видалення, який теж показано на рисунку 3.4.

Для зручності в системі існує можливість зберігати створені схеми доріг. Схемою доріг є послідовно вказані місце розташування депо та дороги обслуговування з відповідними їм класами пріоритетів. Ці схеми можна використовувати в наступні рази користування інформаційною системою, при цьому перелік доріг та їх пріоритетність може бути змінена, але без збереження таких змін в існуючій схемі (для збереження змін необхідно створити нову схему). Функція збереження схеми доступна при виборі 5 та більше доріг, оскільки задання невеликої кількості доріг обслуговування не є складним та витратним по часу процесом.

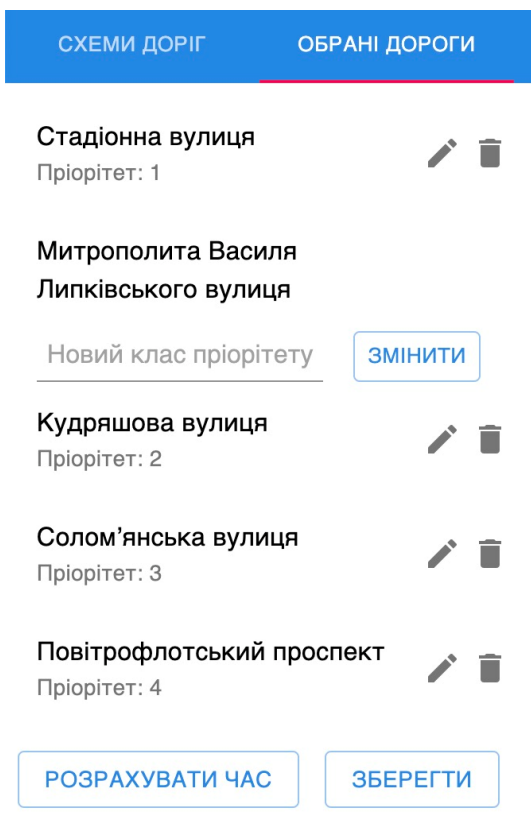


Рисунок 3.4 – Редагування класу пріоритетів

Після того, як було визначено точку депо та вказано хоча б 5 доріг обслуговування, для створення нової схеми треба натиснути на кнопку «Зберегти», що зображена на рисунку 3.3. Далі у виведеному вікні необхідно вказати назву схеми (нові назви не мають співпадати зі вже існуючими) та натиснути на кнопку «Зберегти» (рисунок 3.5). Буде виведено повідомлення про те, що схема була створена (рисунок 3.6), після цього користувач зможе продовжити виконувати планування.

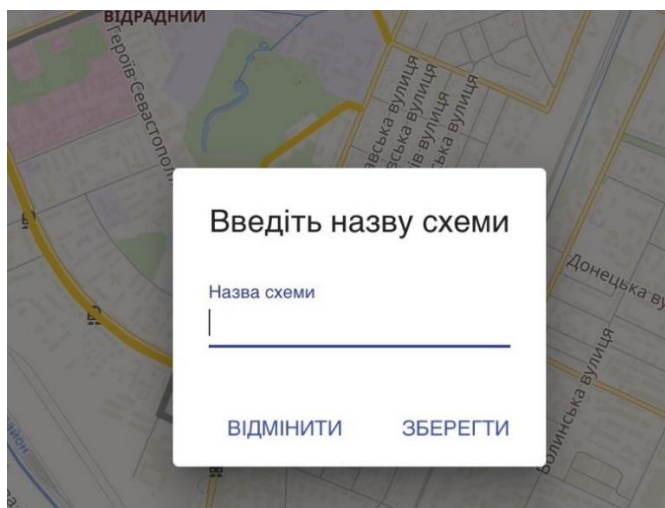


Рисунок 3.5 – Створення нової схеми доріг

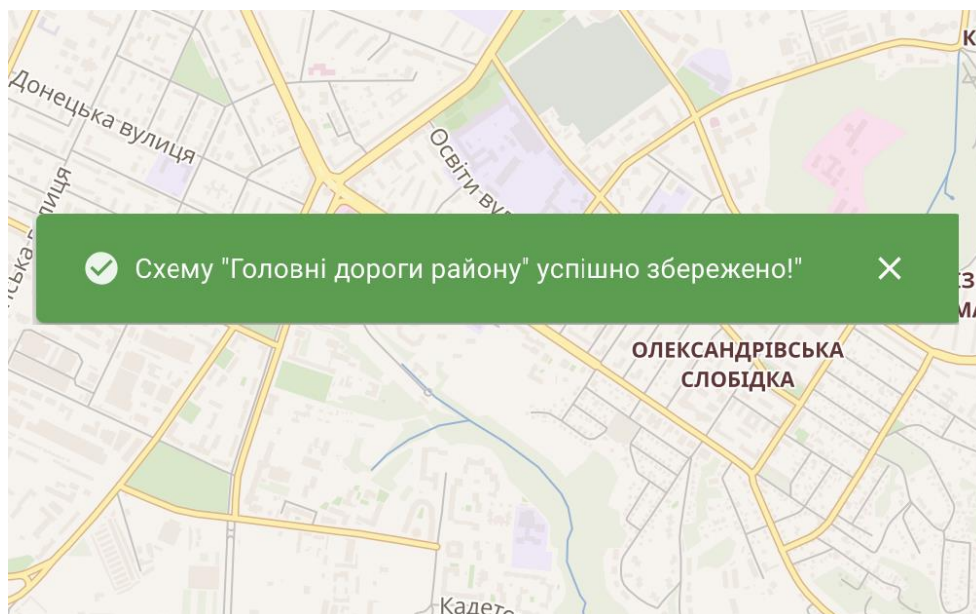


Рисунок 3.6 – Повідомлення про успішне збереження схеми доріг

Всі створені схеми можна переглянути на інформаційній панелі у розділі «Схеми доріг» (рисунок 3.7). Також є можливість видаляти будь-які схеми доріг.

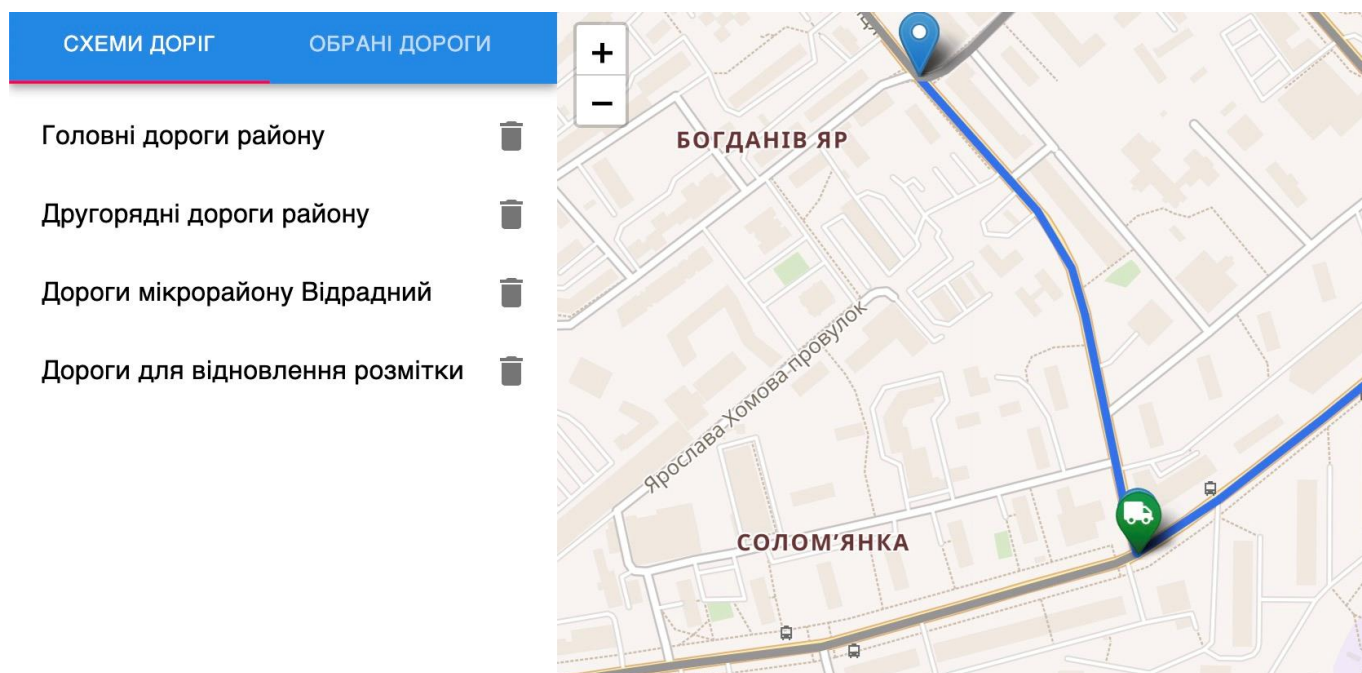


Рисунок 3.7 – Перелік схем доріг

Повернемося до розгляду послідовності дій при складанні плану маршрутів. Після того, як було визначено депо та обрано необхідні дороги обслуговування, користувачу треба натиснути на кнопку «Розрахувати час», яка зображена на рисунку 3.3, щоб завершити визначення вхідних даних. На рисунку 3.8 зображено спливаюче вікно, в якому необхідно вказати наступну інформацію: вид робіт, що буде виконуватися на заданих дорогах, кількість доступної дорожньої техніки, місткість

однієї одиниці дорожньої техніки, запланований час початку виконання обслуговування, а також час, до якого повинні бути обслуговані всі дороги з кожного із визначених класів пріоритетів (тобто вказати директивні терміни).

Коли були заповнені всі поля, що зображені на рисунку 3.8, потрібно натиснути на кнопку «Сформувати план робіт» для пошуку маршрутів транспортних засобів. Якщо відповідно до введених даних побудувати маршрути неможливо, то виводиться повідомлення, як показано на рисунку 3.9, та користувач має змогу завершити планування або відредагувати вхідні параметри у вікні, яке зображене на рисунку 3.8.

| | |
|---|-------------------------------|
| Види робіт Збір сміття | |
| Кількість машин | Місткість однієї машини |
| Час початку робіт 07:30 | |
| Пріоритет 1 - (1 дорога) | Час закінчення робіт 08:00 |
| Пріоритет 2 - (2 дороги) | Час закінчення робіт 08:55 |
| Пріоритет 3 - (1 дорога) | Час закінчення робіт 09:20 |
| Пріоритет 4 - (1 дорога) | Час закінчення робіт 09:45 |
| ВІДМІНИТИ СФОРМУВАТИ ПЛАН РОБІТ | |

Рисунок 3.8 – Визначення даних для виконання планування

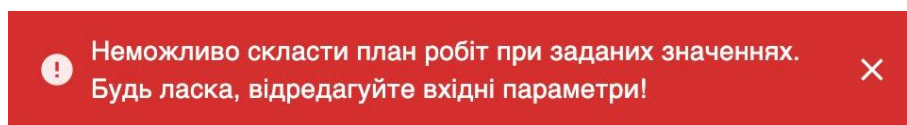


Рисунок 3.9 – Повідомлення про відсутність допустимих маршрутів

Якщо маршрути, що відповідають заданим обмеженням, були побудовані, то виводиться повідомлення про успішне складання плану обслуговування (рисунок 3.10). В разі, коли користувач лише оцінює можливість виконання тих чи інших робіт, він може завершити планування після отримання результату, натиснувши на відповідну кнопку.

Якщо ж користувач хоче переглянути процес виконання обслуговування одразу, то потрібно натиснути на кнопку «Почати симуляцію», яка наведена на рисунку 3.10.

План виконання робіт сформовано успішно

Запланований загальний час обслуговування: 20 хв

<https://docs.google.com/spreadsheets/d/1s>



ЗАВЕРШИТИ ПОЧАТИ СИМУЛЯЦІЮ

Рисунок 3.10 – Повідомлення про успішну побудову допустимих маршрутів

На рисунку 3.10 показано, що користувачу надається посилання на складений план робіт у вигляді онлайн excel документу. Користувачу потрібно натиснути на значок поряд з посиланням, щоб скопіювати його та відкрити документ за цим посиланням у новій вкладці браузера. Приклад такого документу зображено на рисунку 3.11.

Після натискання на кнопку «Почати симуляцію», у системі починає відбуватися симуляція руху транспортних засобів, що представлена на рисунку А.9, відповідно до складених маршрутів. Для цього будується граф, в якому дуги – це всі дороги, а вершини – це точки перетину доріг. Депо також є вершиною графу, з якої моделі транспортних засобів починаються свій рух по графу.

| | A | B | C |
|----|--|--|---|
| 1 | Вид робіт | Час початку робіт | |
| 2 | Очищення від бруду | 7:35 | |
| 3 | | | |
| 4 | Назва вулиці | Запланований час завершення робіт | |
| 5 | Митрополита Василя Липківського вулиця | | |
| 6 | Стадіонна вулиця | | |
| 7 | Повітрофлотський проспект | 7:44 | |
| 8 | Івана Огієнка вулиця | | |
| 9 | Кудряшова вулиця | | |
| 10 | Генерала Шаповала вулиця | 8:03 | |
| 11 | Солом'янська вулиця | | |
| 12 | | | |

Рисунок 3.11 – Документ з запланованими роботами

Системні позначення доріг відповідно до їх кольору:

- сірий (на етапі вибору доріг для обслуговування): дорога доступна для вибору;
- синій (на етапі вибору доріг та на етапі моделювання руху транспортних засобів): дорога обрана для обслуговування;
- жовтий (на етапі моделювання руху транспортних засобів): дорога під час процесу обслуговування;
- зелений (на етапі моделювання руху транспортних засобів): дорога з завершеним обслуговуванням.

Якщо в процесі виконання обслуговування на якійсь із доріг рівень інтенсивності дорожнього руху підвищується вище заданої норми, яка дорівнює б балам, то система починає шукати можливості корегування шляхів, до яких входить ця дуга. Якщо шляхи були скореговані, то документ з планом робіт автоматично оновлюється. Їхня структура залишається такою, як зображено на рисунку 3.11. Моделі транспортних засобів на цих шляхах змінюють свій рух на графі відповідно до скоригованих для них маршрутів. Інші – дотримуються початкових планів.

Висновки

В даному розділі було описано засоби розробки, що були використані під час реалізації інформаційної системи планування обслуговування доріг міста, що реалізує алгоритм знаходження допустимого розв'язку задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин. Було аргументовано вибір мови програмування JavaScript та засобів для роботи з інтерактивною картою, таких як OpenStreetMap, Leaflet, Mapbox Directions API. Для роботи з онлайн документами було вирішено використовувати Google Sheets API. Також було описано переваги обраної системи управління базами даних MySQL для зберігання вхідних та вихідних даних.

Архітектура програмного забезпечення в даному розділі була представлена за допомогою діаграми класів. Було описано призначення класів інформаційної системи та специфікацією функцій цих класів, що реалізують розроблене програмне забезпечення.

Також було наведено докладний опис роботи з програмою у вигляді покрокової інструкції користувача, що охоплює основні процеси роботи з інформаційною системою планування обслуговування доріг міста.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї стартап-проекту

Аналіз змісту ідеї та загальних напрямків використання стартап-проекту, а саме інформаційної системи планування обслуговування доріг міста, наведено у таблиці 4.1.

Таблиця 4.1 – Зміст ідеї стартап-проекту

| Ідея | Можливі застосування | Вигоди для користувачів |
|---|---|---|
| Автоматизація процесу планування обслуговування доріг міста, який складається з побудови маршрутів для таких видів робіт: | 1. Очищення дорожнього покриття від сміття, бруду, сторонніх предметів. | 1. Забезпечення нормального руху транспортних засобів. |
| | 2. Мийка дорожнього покриття та очищення від пилу. | 2. Підтримання стану дорожнього покриття. |
| | 3. Нанесення та відновлення зношеної розмітки. | 3. Уникнення аварійних ситуацій. |
| | 4. Розчищення автомобільних доріг від снігу. | 4. Відновлення транспортного руху в найкоротші терміни. |
| | 5. Розподілення протижеледних матеріалів. | 5. Надання можливості безпечного руху транспорту. |

Слід зазначити, що в загальному застосування інформаційної системи планування обслуговування доріг міста в усіх напрямках забезпечує для користувачів більш ефективно та економічно вигідне планування процесів, що пов'язані з різними видами робіт на дорогах.

Після огляду існуючих аналогів для подальшого порівняння було обрано Ecofleet, Maanteeamet та Транснавігація.

Ecofleet – це міжнародна компанія з розробки програмного забезпечення, що спеціалізується на управлінні автомобільним парком і робочою командою на базі GPS. Ecolleet надає можливість оптимізації, обліку та документації роботи, а також мінімізації управлінського навантаження.

Maanteeamet – це електронна служба управління дорожнім рухом, що забезпечує літнє та зимове утримання державних магістралей з використанням дорожніх метеостанцій та камер, а також GPS-системи моніторингу.

Транснавігація – автоматизована система навігаційного диспетчерського контролю за роботою машин і механізмів для утриманні автомобільних доріг. Система забезпечує контроль за використанням дорожньої техніки, облік і аналіз виконання планів по обслуговуванню доріг і магістралей, регулювання відхилень, що виникають при виконанні заданих робіт.

Визначимо відмінності інформаційної системи планування обслуговування доріг міста від конкурентів та її техніко-економічні переваги і недоліки. В таблиці 4.2 проаналізовано чим відрізняється стартап-проект від існуючих аналогів.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| № п/ п | Техніко-економічні характеристики ідеї | Потенційні товари/концепції конкурентів | | | | W ⁵ | N ⁶ | S ⁷ |
|--------------|--|--|---|---|---|----------------|----------------|----------------|
| | | 1 | 2 | 3 | 4 | | | |
| 1. | Доступність в Україні | + | - | - | - | | | S |
| 2. | Обслуговування міських вулиць | + | + | + | - | | N | |
| 3. | Обслуговування магістралей країни | - | + | + | + | W | | |
| 4. | Динамічне задання депо і доріг | + | - | - | - | | | S |
| 5. | Зимове утримання доріг | + | + | - | + | | N | |
| 6. | Роботи по розмітці доріг | + | - | - | + | | N | |
| 7. | Мийка та підмітання дорожнього покриття | + | + | - | + | | N | |
| 8. | Розрахунок найкращого маршруту | + | - | - | - | | | S |
| 9. | Врахування директивних термінів | + | - | - | - | | | S |

Продовження таблиці 4.2

| № п/ п | Техніко-економічні характеристики ідеї | Потенційні товари/концепції конкурентів | | | | W ⁵ | N ⁶ | S ⁷ |
|--------------|---|--|---|---|---|----------------|----------------|----------------|
| | | 1 | 2 | 3 | 4 | | | |
| 10. | Врахування пріоритетності доріг | + | - | - | - | | | S |
| 11. | Відображення фактичного місцезнаходження транспортних засобів | - | + | + | + | W | | |
| 12. | Врахування дорожніх заторів в реальному часі | + | - | - | - | | | S |
| 13. | Документування плану виконання робіт | + | + | + | + | | N | |
| 14. | Підрахунок вартості виконаних робіт | + | + | - | + | | N | |
| 15. | Формування звітності по завершенню виконання робіт | - | + | + | + | W | | |

Позначення до таблиці 4.2:

- а) 1 – Інформаційна система планування обслуговування доріг міста;
- б) 2 – Ecofleet;
- в) 3 – Транснавігація;
- г) 4 – Maanteeamet;
- д) 5 – Слабка сторона;
- е) 6 – Нейтральна сторона;
- ж) 7 – Сильна сторона.

4.2 Технологічний аудит ідеї стартап-проекту

Визначимо перелік технологій, за допомогою яких можна реалізувати ідею стартап-проекту (таблиця 4.3).

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

| Ідея проекту | № п/п | Технології її реалізації | Наявність технологій | Доступність технологій |
|--|-------|--|----------------------|------------------------|
| Автоматизація процесу планування обслуговування доріг міста. | 1. | JavaScript – мова програмування. | Наявна | Доступна |
| | 2. | Node.js – серверна платформа. | Наявна | Доступна |
| | 3. | React.js – бібліотека для розробки інтерфейсів користувача. | Наявна | Доступна |
| | 4. | Leaflet – бібліотека для роботи з інтерактивною картою. | Наявна | Доступна |
| | 5. | MySQL – реляційна система управління базами даних. | Наявна | Доступна |
| | 6. | Алгоритм побудови маршрутів. | Необхідно розробити | Доступна |
| | 7. | OpenStreetMap – відкрита географічна карта світу. | Наявна | Доступна |
| | 8. | Mapbox Directions API – API для створення маршрутизації по карті від Mapbox. | Наявна | Доступна |
| | 9. | GoogleMaps API – картографічний сервіс. | Наявна | Недоступна |
| | 10. | Google Directions API – API для створення маршрутизації по карті від Google. | Наявна | Недоступна |
| | 11. | Google Sheets API – онлайн-сервіс для роботи з таблицями Excel. | Наявна | Доступна |

Обрані технології для реалізації ідеї проекту: JavaScript, Node.js, React.js, Leaflet, MySQL, Алгоритм побудови маршрутів, OpenStreetMap, Mapbox Directions API, Google Sheets API.

Такі технології, як GoogleMaps API та Google Directions API, недоступні для даного стартап-проекту через їхнє платне використання.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначимо потенційні групи майбутніх користувачів інформаційної системи планування обслуговування доріг міста. У таблиці 4.4. описано потреби кожної з можливих груп.

Таблиця 4.4 – Характеристика потенційних користувачів стартап-проекту

| № п/п | Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|-------|--|---|--|---|
| 1. | Підвищення ефективності обслуговування доріг | 1. Державні підприємства («Укравтодор») 2. Муніципальні підприємства («Київавтодор») 3. Приватні підприємства | На сьогоднішній день в Україні серед підприємств, що займаються обслуговуванням доріг, переважають державні та муніципальні підприємства. Така ситуація склалася через грошові проблеми оплати праці підрядників: приватний бізнес, на відміну від держпідприємств, не готовий до затримки оплати. | – невисока вартість товару; – зручність та простота у використанні; – ефективність впровадження; – швидкість роботи; – доступність у будь-який час. |
| 2. | Зменшення витрат | | | |
| 3. | Зменшення пройденої відстані | | | |
| 4. | Зниження часу виконання дорожніх робіт | | | |
| 5. | Враховання часових обмежень | | | |

Далі необхідно розглянути фактори, що будуть сприяти впровадженню стартап-проекту на потенційний ринок, та фактори, що можуть цьому перешкодити.

Фактори можливостей:

- розширення мережі доріг;

- збільшення кількості транспортних засобів на дорогах;
- необхідність підвищення якості та швидкості виконання робіт;
- економічна доцільність;
- вилучення людського фактору та упередженості через автоматизацію процесів;
- потреба у зменшенні управлінського навантаження.

Фактори загроз:

- сталі процеси та небажання вводити інновації;
- відсутність фінансування у підприємства;
- невелика ділянка чи обсяг робіт, що виконує підприємство.

Як можна побачити, наразі склалися сприятливі можливості для виведення на ринок інформаційної системи планування обслуговування доріг міста. В такому разі, треба дослідити умови конкуренції на ринку (таблиця 4.5).

Таблиця 4.5 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства |
|--|---|--|
| 1. Тип конкуренції: монополія | З проаналізованих раніше існуючих аналогів, можна зробити висновок, що в Україні немає іншої подібної системи, яку могли б використовувати підприємства як альтернативу до стартап-проекту, що розробляється. | Стартап-проект не має близьких конкурентів, тому це спрощує його вихід на ринок. |
| 2. Рівень конкурентної боротьби: національна | Може використовуватися підприємствами з будь-яких міст України. | На даному етапі демонструється робота системи на певному районі міста Києва. В подальшому потрібно |

Продовження таблиці 4.5

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства |
|--|--|--|
| | | розширювати зону впровадження системи. |
| 3. Галузева ознака: міжгалузева | Система є програмним продуктом, призначеним для використання у галузі обслуговування доріг. | Потрібно ретельно дослідити галузі, що пов'язані зі стартап-проектом. |
| 4. Конкуренція за видами товарів: товарно-видова | Замість системи, що розробляється, можливо використовувати тільки систему того ж самого призначення, що буде виконувати схожі функції. | При розробці системи потрібно врахувати якомога більше особливостей сфери діяльності та додати функції, яких немає у систем схожого призначення. |
| 5. Характер конкурентних переваг: нецінова | Конкуренція відбувається в неціновому форматі за рахунок функціональних можливостей та ефективності використання. | Потрібно підвищувати якість та надійність роботи системи, а також перелік наданих можливостей. |
| 6. Інтенсивність: не марочна | Оскільки це стартап, то проект є новим та не має відомої торговельної марки. | Потрібно займатися розкруткою стартап-проекту, щоб зацікавити цільову аудиторію. |

З огляду на поточну конкурентну ситуацію можна зробити висновок, що інформаційна система планування обслуговування доріг міста може бути конкурентоспроможним та затребуваним товаром поміж своєї цільової аудиторії.

На основі попередньо опрацьованої інформації проведемо аналіз факторів конкурентоспроможності інформаційної системи планування обслуговування доріг міста (таблиця 4.6).

Таблиця 4.6 – Обґрунтування факторів конкурентоспроможності

| № п/п | Фактор конкурентоспроможності | Обґрунтування |
|-------|---|--|
| 1. | Доступність системи на території України | Відсутність подібних аналогів на території України. |
| 2. | Зручність, простота і наочність у використанні | Не потребує додаткового навчання та залучення нових робітників. |
| 3. | Якісне планування маршрутів | Мінімізація часу проходження транспортними засобами всіх доріг. |
| 4. | Врахування часових обмежень | Директивні терміни є природними обмеженнями в реальних умовах, проте більшість програмних застосувань не мають можливості враховувати такі обмеження. |
| 5. | Поділ доріг згідно їх пріоритетів | Зазвичай райони міста поділяються на декілька зон відповідно до їх значущості для надання якнайкращого сервісу. |
| 6. | Врахування інтенсивності дорожнього руху в реальному часі | Ефективніше планування маршрутів з врахуванням поточної ситуації на дорогах або швидке корегування маршруту в разі непередбачуваних ситуацій (аварія, затор тощо). |

Завершальним етапом аналізу ринкових можливостей запуску стартап-проекту є SWOT-аналіз, що наведено у таблиці 4.7.

Таблиця 4.7 – SWOT-аналіз стартап-проекту

| | |
|--|---|
| <p><i>Сильні сторони:</i></p> <ul style="list-style-type: none"> – доступність системи; – зручність, простота і наочність у використанні; – якісне планування маршрутів; – врахування часових обмежень; – ранжування доріг згідно їх пріоритетів; | <p><i>Слабкі сторони:</i></p> <ul style="list-style-type: none"> – немає можливості відслідковувати поточне місцезнаходження транспортних засобів; – обмеженість у використанні лише на міських дорогах; – відсутність звітності та аналізу; |
|--|---|

Продовження таблиці 4.7

| | |
|---|--|
| <ul style="list-style-type: none"> – врахування інтенсивності дорожнього руху. <p><i>Можливості:</i></p> <ul style="list-style-type: none"> – збільшення обсягів робіт, що виконуються підприємством; – зниження складності планування маршрутів; – уникнення збитків на підприємстві через некоректне планування маршрутів; – запобігання допущення помилок працівниками, що займаються складанням планів маршрутів; – відсутність потреби у додаткових фахівцях, що займаються плануванням маршрутів. | <ul style="list-style-type: none"> – новий проект, що не має відомої марки. <p><i>Загрози:</i></p> <ul style="list-style-type: none"> – сформовані процеси планування на підприємстві не підлягають внесенню змін; – нестача коштів на підприємстві для виконання робіт; – використання додаткового інструменту планування не має сенсу через невеликі обсяги робіт та/або ділянки обслуговування. |
|---|--|

4.4 Розроблення ринкової стратегії проекту

Для розроблення ефективної ринкової стратегії стартап-проекту перш за все визначимо характеристики цільових груп можливих споживачів (таблиця 4.8). Потенційними клієнтами для інформаційної системи планування обслуговування доріг міста є підприємства, що займаються виконанням різних видів робіт на міських вулицях.

Таблиця 4.8 – Вибір цільових груп потенційних споживачів

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|----------|---|--|--|--|--|
| 1. | Державні підприємства | Готовність до сприйняття нижче середнього. | Відповідальні за обслуговування доріг лише державного значення (за межами населених пунктів). На теперішній момент система призначена для використання на міських дорогах. Проте у веденні держпідприємств всі рівні дорожніх організацій країни, тому можливий низький рівень попиту. | Висока інтенсивність. | Складний вхід у сегмент через високий рівень значущості підприємств. |
| 2. | Муніципальні підприємства | Готовність до сприйняття нижче середнього, | Виконують 80% робіт по обслуговуванню доріг місцевого | Висока інтенсивність. | Вхід у сегмент середньої важкості через те, що на цих |

Продовження таблиці 4.8

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|----------|---|---|--|--|---|
| | | оскільки рідко впроваджують нові підходи. | значення (дороги вулиць міст та ін. населених пунктів), тому рівень попиту може бути досить високим. | | підприємствах довготривало встановлені процеси роботи, до яких не вносяться зміни. |
| 3. | Приватні підприємства | Висока готовність до впровадження нових рішень, бо прагнуть працювати краще та ефективніше. | Виконують 20% робіт по обслуговуванню доріг місцевого значення, тому рівень попиту є середнім. | Висока інтенсивність. | Досить простий вхід у сегмент, бо такі підприємства зазвичай прагнуть отримати якнайбільший прибуток. |

Обрано наступні цільові групи: муніципальні та приватні підприємства через досить високий рівень попиту, простоту входу у сегмент та готовність до сприйняття продукту.

Обрана стратегія охоплення ринку – стратегія диференційованого маркетингу, а також визначена базова стратегія розвитку як стратегія спеціалізації, що не передбачає охоплення всього ринку, а навпаки зосередження на конкретних цільових сегментах та задоволенні їхніх потреб.

Далі визначимо базову стратегію конкурентної поведінки. Стартап-проект буде залучати до своєї інформаційної системи нових користувачів, а не переманювати існуючих у конкурентів. Хоча основні характеристики стартап-проекту будуть відповідати реалізованим можливостям у конкурентів, все ж таки у системі буде розроблено ряд інноваційних рішень, які зроблять її відмінною від аналогів. З огляду на це, а також на те, що проект буде новатором на ринку, можна зробити висновок, що кращою стратегією конкурентної поведінки буде стратегія заняття конкурентної ніші. В цьому разі обирається декілька ринкових сегментів та формується прихильність споживачів, розвиваються відмінні властивості системи, що відповідають потребам та попиту на ринку.

На основі вже визначених рішень щодо ринкової поведінки сформуємо комплекс ключових асоціацій, що дозволять споживачам ідентифікувати позицію стартап-проекту:

- якісне планування маршрутів для обслуговування міських доріг завдяки мінімізації часу проходження транспортними засобами шляхів сполучень;
- врахування суттєвих обмежень щодо директивних термінів виконання робіт та місткості машин;
- підвищення ефективності планування через корегування маршрутів відповідно до інтенсивності дорожнього руху в реальному часі.

4.5 Розроблення маркетингової програми стартап-проекту

У таблиці 4.9 описано маркетингову концепцію стартап-проекту.

Таблиця 4.9 – Визначення ключових переваг концепції потенційного товару

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами |
|-------|--|---|--|
| 1. | Виконання якісного обслуговування доріг. | Підвищення ефективності обслуговування доріг. | Корегування маршрутів в реальному часі у разі заторів на дорогах, аварій тощо. |

Продовження таблиці 4.9

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами |
|----------|---|---|---|
| 2. | Підрахунок вартості виконаних робіт. | Зменшення витрат та уникнення збитків. | Мінімізація вартості обслуговування в результаті роботи алгоритму. |
| 3. | Підрахунок пройденної відстані. | Зменшення пройденної відстані. | Мінімізація пройденної відстані обслуговування в результаті роботи алгоритму. |
| 4. | Підрахунок часу виконання робіт. | Зниження часу виконання дорожніх робіт. | Мінімізація часу обслуговування в результаті роботи алгоритму. |
| 5. | Врахування обмежень, що виникають під час обслуговування доріг. | Врахування часових обмежень та місткості машин. | Можливість розділяти дороги за класами пріоритетів та встановлення їм у відповідність директивних термінів, врахування потреби поновлювати ресурси. |
| 6. | Пришвидшення процесу планування маршрутів. | Зниження складності планування маршрутів. | Автоматизованість процесу планування, а також зручність, простота і наочність у використанні системи. |

Стартап-проект буде захищено від копіювання через захист прав на інтелектуальну власність. А далі розробимо концепцію маркетингових комунікацій (таблиця 4.10).

Таблиця 4.10 – Концепція маркетингових комунікацій

| № | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|----|--|--|--|---|--|
| 1. | Щоденне планування маршрутів. | – електрона пошта; – телефонний зв'язок; – особисті зустрічі; – презентації на тематичних | Якісне та швидке складання планів маршрутів. | Надання інформації щодо існування інструменту для планування маршрутів. | Зручність, швидкість і простота системи у використанні. |
| 2. | Комунікація диспетчера з водіями транспортних засобів для видачі планів. | заходах, конференціях тощо. | Полегшення керування персоналом. | Переконання у полегшенні праці робітників та уникнення людського фактору. | Передача та оновлення планів у онлайн режимі, відсутність потреби у додаткових фахівцях. |
| 3. | Повільна модернізація процесів роботи. | | Автоматизація процесу планування маршрутів. | Схиляння до рішення про покупку. | Переконання у необхідності впровадження нових підходів до роботи. |

Продовження таблиці 4.10

| № | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|----|--|--|--|--|--|
| 4. | Отримання більшого прибутку. | | Мінімізація вартості обслуговування. | Інформування про ефективність використання розробленого алгоритму. | Демонстрація переваг алгоритму. |
| 5. | Покращення обслуговування доріг міста. | | Врахування обмежень. | Нагадування про існування у реальній практиці природніх обмежень. | Можливість розподілу доріг за класами пріоритетів та встановлення відповідних їм директивних термінів. |
| 6. | Пришвидшення обслуговування доріг міста. | | Мінімізація часу обслуговування та пройденої відстані. | Інформування про ефективність використання розробленого алгоритму. | Демонстрація роботи алгоритму. |

Продовження таблиці 4.10

| № | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|----|---|--|---|--|--|
| 7. | Прийняття рішень щодо зміни маршрутів у разі непередбачуваних ситуацій (затори, аварії тощо). | | Корегування маршрутів в реальному часі у випадку непередбачуваних ситуацій. | Надання інформації про врахування у системі інтенсивності дорожнього руху. | В разі зміни маршруту водії можуть бачити оновлений план якнайшвидше онлайн. |

Висновки

Після проведеного аналізу було визначено, що є можливість ринкової реалізації стартап-проекту. Як показав аналіз цільових груп потенційних споживачів найбільший рівень попиту присутній у двох сегментах ринку: муніципальні підприємства та приватні підприємства, що займаються обслуговуванням доріг. Саме тому потрібно використовувати стратегію зосередження на конкретних цільових групах та задоволенні їхніх потреб. Інформаційна система планування обслуговування доріг міста є конкурентоспроможним проектом та має перспективи для впровадження, хоча в деяких групах потенційних клієнтів наявний досить високий бар'єр входу у сегмент. З огляду на те, що проект буде новатором на ринку, а також, що склалися сприятливі можливості для виведення на ринок даної системи, можна зробити висновок про доцільність подальшої імплементації проекту.

ВИСНОВКИ

Магістерська дисертація присвячена розробці інформаційної системи планування обслуговування доріг міста та впровадженню у систему різновиду задач маршрутизації по дугам – мінімаксної задачі k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин. Метою є підвищення якості процесів обслуговування доріг районів міста за рахунок мінімізації часу проходження транспортними засобами всіх шляхів сполучень та врахування обмежень на директивні терміни виконання робіт та місткість машин.

У першому розділі було проведено ґрунтовне дослідження предметного середовища та процесів діяльності, прийнято рішення з інформаційного забезпечення потреб системи. Визначено особливості функціонування сфери застосування інформаційної системи планування обслуговування доріг. Також описані призначення, мета та задачі, які необхідно вирішити для досягнення поставленої мети.

Другий розділ присвячено аналізу задач маршрутизації транспортних засобів, що орієнтовані на проходження шляхів сполучень (ARP). Було акцентовано увагу на типах задач про китайського листоношу. В процесі аналізу було знайдено випадок, який найкраще задовольняє потреби інформаційної системи планування обслуговування доріг – мінімаксна задача k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин. Тому було представлено змістовну постановку та математичну модель задачі MM k -CCPP with DC. Здійснено аналіз методів розв'язання споріднених задач до досліджуваної. Розглянуті у розділі алгоритми були взяті за основу при розробці алгоритму побудови допустимого розв'язку. Також розділ містить загальну схему методу розв'язання задачі та розроблений алгоритм знаходження розв'язку. Особливістю алгоритму побудови допустимого розв'язку є обов'язковість виконання умов директивних термінів та місткості машин на всіх кроках алгоритму. Для побудови простих маршрутів та їх корегування у разі підвищення інтенсивності руху використано алгоритм Дейкстри. У розділі описано проведення досліджень та аналіз отриманих результатів.

Третій розділ описує програмну реалізацію інформаційної системи планування обслуговування доріг з використанням розробленого алгоритму. Для розробки програмного продукту була обрана мова програмування JavaScript та використано такі технології як відкриту географічну карту OpenStreetMap, бібліотеку для роботи з картою Leaflet, Mapbox Directions API для створення маршрутизації та Google Sheets API для формування онлайн документів. Цей розділ містить опис архітектури програмного забезпечення, що включає опис класів системи та специфікацію функцій. Також було наведено докладний опис роботи з програмним продуктом у вигляді покрокової інструкції користувача.

В четвертому розділі було виконано один з етапів розроблення стартап-проекту, яким є інформаційна система планування обслуговування доріг, а саме маркетинговий аналіз. Було оцінено можливості стартап-проекту та зроблено висновок про доцільність подальшої імплементації проекту.

Під час роботи над магістерською дисертацією:

- в результаті дослідження предметного середовища виявлено необхідність врахування обмежень на директивні терміни виконання робіт та місткість машин;
- на основі аналізу задач маршрутизації по дугам обрано тип задач про китайського листоношу для подальшого дослідження;
- сформульовано математичну модель задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин, яка враховує необхідні особливості функціонування предметного середовища;
- модифіковано алгоритм розв’язання сформульованої задачі для врахування додаткових обмежень;
- проведено серію експериментів для дослідження ефективності модифікованого алгоритму, яка показала його високу швидкість при різних вхідних параметрах;
- розроблено інформаційну систему, в якій використано модифікований алгоритм.

РЕКОМЕНДАЦІЇ

Інформаційна система планування обслуговування доріг має перспективи для подальших удосконалень. Зокрема, для кращої пристосованості системи до реально існуючих процесів необхідно додати до системи підрахування кількості смуг дорожнього руху на кожній з доріг. Таким чином, граф поточної задачі перетвориться на мультиграф.

Подальші дослідження можливі і для задачі мінімаксу k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин. В даній роботі задача була поставлена так, що директивні терміни або виконуються, або розв'язок є недопустимим. Але замість таких жорстких умов виконання директивних термінів можна накладати штрафи за невиконання директивних термінів. Крім того, можна модифікувати розроблений алгоритм побудови допустимих розв'язків для того, щоб збільшити його ефективність.

Час виконання кожного виду робіт взятий приблизно, виходячи із середньої швидкості. У майбутній перспективі необхідно обладнати кожен одиницю спеціалізованої дорожньої техніки GPS для вдосконалення роботи інформаційної системи. Це дозволить отримувати потрібні дані, з яких можна вираховувати час обслуговування кожної дороги для всіх видів робіт, тобто в результаті буде досягнуто більшої точності роботи алгоритму. Також аналіз по GPS дозволить відображати у системі пересування транспортних засобів на карті відповідно до їх реального положення для того, щоб можна були побачити поточний прогрес під час виконання обслуговування.

Таким чином, в даній роботі було передбачено основні, але не всі особливості процесу обслуговування доріг при розробці інформаційної системи планування та покрито деяку частину можливих досліджень мінімаксної задачі k -китайських листонош з врахуванням директивних термінів виконання робіт та місткості машин.

ПЕРЕЛІК ПОСИЛАНЬ

1. Huanfa Chen. A Balanced Route Design for Min-Max Multiple Depot Rural Postman Problem (MMMDRPP): a police patrolling case [Текст] / Huanfa Chen, Tao Cheng & John Shawe-Taylor // International Journal of Geographical Information Science, Vol. 32. – 2018. – № 1. – pp. 169-190.
2. Mei-Ko Kwan. Graphic programming using odd or even points [Текст] // Chinese Mathematics 1. – 1962. – pp. 273-277.
3. Arc routing: theory, solutions, and applications [Текст] / edited by Moshe Dror // Kluwer Academic Publishers, New York. – 2000.
4. Setianingsih, A. I. Road Maintenance and Rehabilitation Program Using Functional and Structural Assessment [Текст] / A. I. Setianingsih, S. Sangaji, A. Setyawan // IOP Conference Series: Materials Science and Engineering 176. – 2017.
5. Дорожний затор – Вікіпедія [Електронний ресурс] // Режим доступу: http://ru.wikipedia.org/wiki/Дорожний_затор
6. Яндекс.Пробки – Вікіпедія [Електронний ресурс] // Режим доступу: <https://ru.wikipedia.org/wiki/Яндекс.Пробки>
7. Kaj Holmberg. Heuristics for the weighted k -rural postman problem with applications to urban snow removal [Текст] // Journal on Vehicle Routing Algorithms 1. – 2018. – pp. 105-119.
8. Kaj Holmberg. The (Over) Zealous Snow Remover Problem [Текст] // Transportation Science, Vol. 53. – 2019. – № 3. – pp. 623-656.
9. Григорець, Г.О. Застосування задачі маршрутизації транспортних засобів по дугах для оптимізації процесів обслуговування доріг міста [Текст] матеріали Тринадцятої міжнародної науково-практичної конференції «Математичне та імітаційне моделювання систем. МОДС 2018»: тези / Г.О. Григорець // 392 с. – Чернігів : ЧНТУ, 2018. – 103-106 с.
10. Mustafa Yılmaz. Chinese postman problem approach for a large-scale conventional rail network in Turkey [Текст] / Mustafa Yılmaz, Merve Kayacı Çodur, Hamid Yılmaz // Tehnički vjesnik 24. – 2017. – pp. 1471-1477.

11. Huanfa Chen. Modelling Police Patrol Routing as Min-Max Postmen Problems [Текст] / Huanfa Chen, Tao Cheng // 25th Geographical Information Science Research UK Conference. – 2016.
12. Gregory Dreifus. Path Optimization Along Lattices in Additive Manufacturing Using the Chinese Postman Problem [Текст] / Gregory Dreifus, Kyle Goodrick, Scott Giles, Milan Patel, Reed Matthew Foster, Cody Williams, John Lindahl, Brian Post, Alex Roschli, Lonnie Love, Vlastimil Kunc // 3D printing and additive manufacturing 4. – 2017. – № 2. – pp. 98-104.
13. Определения и простейшие свойства графов – Yaklass [Электронный ресурс] // [Режим доступа]: https://www.yaklass.ru/Определения_и_свойства_графов
14. Sanne Wøhlk. A Decade of Capacitated Arc Routing [Текст] // B. Golden et al. (eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. – 2008. – pp. 29-48.
15. Mario Galić. Simulation Model for Scenario Optimization of the Ready-Mix Concrete Delivery Problem [Текст] / Mario Galić, Ivan Kraus // Journal of civil engineering, Vol. 11. – 2016. – № 2. – pp. 7-18.
16. Orloff, C. A fundamental problem in vehicle routing. [Текст] // Networks 4. – 1975. – № 1. – pp. 35-64.
17. Frederickson, G. N. Approximation algorithms for some routing problems. [Текст] / G. N. Frederickson, M. S. Hecht, C. E. Kim // In 17th Annual Symposium on Foundations of Computer Science. – 1978. – pp. 216-227.
18. Golden, B. L. Capacitated arc routing problems. [Текст] / B. L. Golden, R. T. Wong // Networks 11. – 1981. – № 3. – pp. 305-315.
19. Christofides, N. The optimum traversal of a graph. [Текст] // The International Journal of Management Science 1. – 1973. – № 6. – pp. 719-732.
20. Patrick Breslin. The capacitated arc routing problem: Lower bounds [Текст] / Patrick Breslin, Antony Keane // 1997.
21. Peter Korteweg. On the Hierarchical Chinese Postman Problem with linear ordered classes [Текст] / Peter Korteweg, Ton Volgenant // European Journal of Operational Research 169. – 2006. – pp. 41-52.

22. Tomaz Kramerger. Priority Constrained Chinese Postman Problem [Текст] / Tomaz Kramerger, Janez Zerovnik // *Logistics & Sustainable Transport*, Vol. 1. – 2007. – № 1.
23. Masoud Rabbani. Capacitated Windy Rural Postman Problem with several vehicles: A hybrid multi-objective simulated annealing algorithm [Текст] / Masoud Rabbani, Safoura Famil Alamdara, Hamed Farrokhi-Asl // *International Journal of Supply and Operations Management*, Vol. 2. – 2016. – № 4. – pp. 1003-1020.
24. Purushothaman Damodaran. Lower Bounds for Hierarchical Chinese Postman Problem [Текст] / Purushothaman Damodaran, Murali Krishnamurthi, Krishnaswami Shihari // *International Journal of Industrial Engineering* 15. – 2008. – № I. – pp. 36-44.
25. Andre Osterhues. On variants of the k -Chinese Postman Problem [Текст] / Andre Osterhues, Frank Mariak // *Operations Research und Wirtschaftsinformatik*. – 2005. – № 30.
26. Невмержицька, С.І. Математична модель мінімаксної задачі k -китайських листонош з врахуванням директивних термінів [Текст] матеріали X Міжнародної наукової конференції ISDMCI'2014 «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту»: тезиси / С.І. Невмержицька, О.Г. Жданова / [редактор В.І. Литвиненко]. – 382 с. – Херсон: ХНТУ, 2014. – 217-219 с.
27. İbrahim Zeki Akzur. Using Genetic Algorithm for Winter Maintenance Operations: Multi Depot k -Chinese Postman Problem [Текст] / İbrahim Zeki Akzur, Timur Keskindürk, Çağatay Kalkanç // *Emerging Markets Journal*, Vol. 5. – 2015. – № 1. – pp. 50-59.
28. Dino Ahr. New Heuristics and Lower Bounds for the Min-Max k -Chinese Postman Problem [Текст] / Dino Ahr, Gerhard Reinelt // *Lecture Notes in Computer Science*, Vol. 2461. – 2002. – pp. 64-74.
29. Григорець, Г. О. Задача транспортної маршрутизації у складі інформаційної системи планування обслуговування доріг міста [Текст] / Г. О. Григорець, О. Г. Жданова // *Paradigm of Knowledge* 4. – 2019. – № 36. – 5-24 с.
30. Невмержицька, С.І. Побудова допустимого розв'язку мінімаксної задачі k -китайських листонош з врахуванням директивних термінів [Текст] матеріали V

Всеукраїнській заочній науково-практичній конференції СІТ-2014 «Сучасні інформаційні технології»: тези / С.І. Невмержицька, О.Г. Жданова // 71 с. – В надзаг.: Міністерство освіти та науки України, Національний технічний університет України «КПІ», факультет інформатики та обчислювальної техніки: Київ, 2014. – 16-20 с.

31. Dijkstra's Algorithm, in Simple Steps [Електронний ресурс] // Режим доступу: <https://www.instructables.com/id/Dijkstras-Algorithm-in-Simple-Steps/>

32. Григорець, Г.О. Інформаційна система планування обслуговування доріг міста з використанням задачі маршрутизації по дугам [Текст] матеріали Всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019): тези / Г.О. Григорець, Жданова О.Г. // 222 с. – Київ: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р. – 12-17 с.

33. OpenStreetMap Україна [Електронний ресурс] // Режим доступу: <http://openstreetmap.org.ua>

34. Leaflet [Електронний ресурс] // Режим доступу: <https://leafletjs.com/>

35. Mapbox API Documentation – Mapbox [Електронний ресурс] // Режим доступу: <https://www.mapbox.com/api-documentation/#directions>

36. Google Sheets API – Google Sheets [Електронний ресурс] // Режим доступу: <https://developers.google.com/sheets/api/>

ДОДАТОК А

Графічні матеріали

Схема структурна варіантів використання

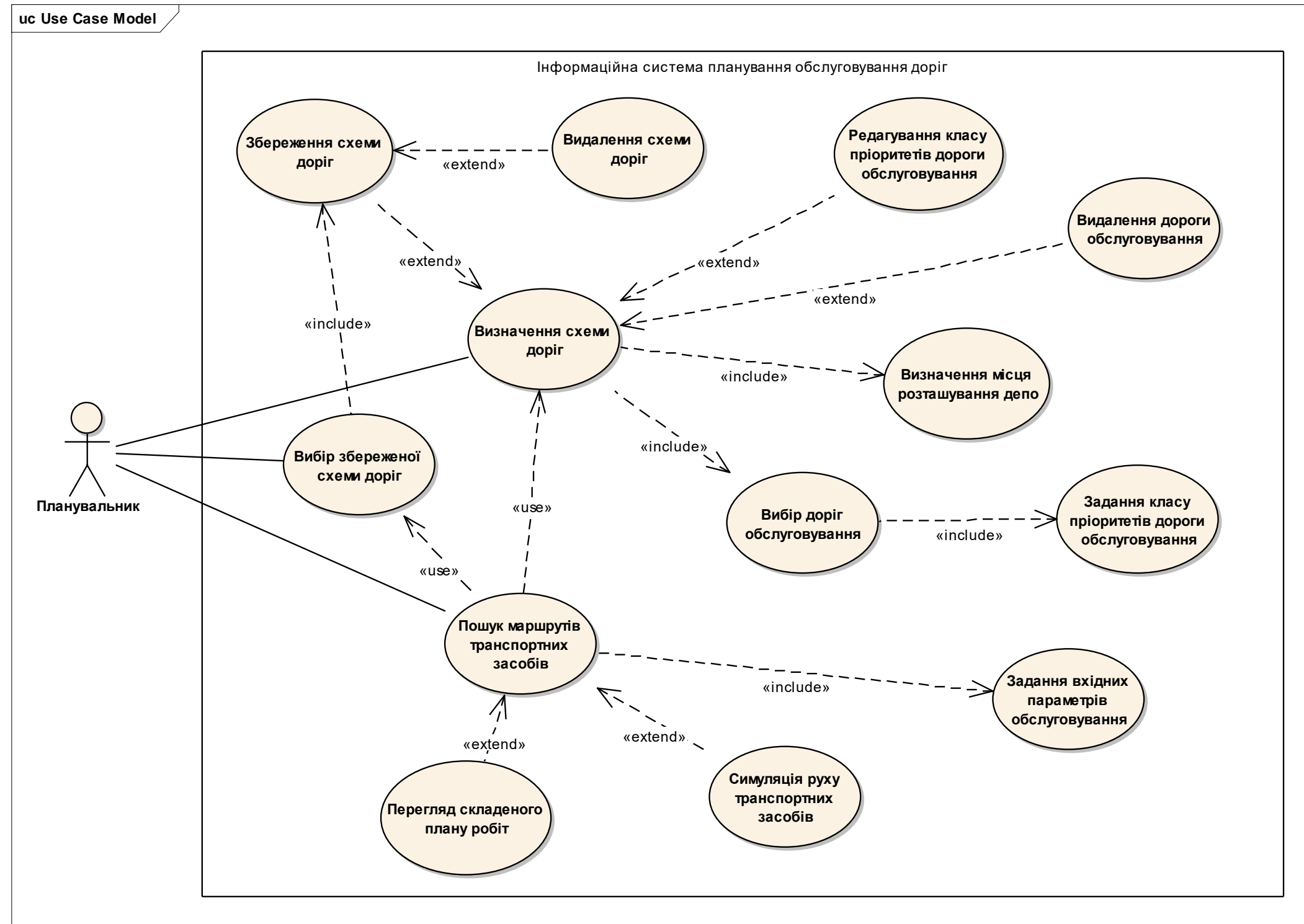


Рисунок А.1 – Схема структурна варіантів використання

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант

Григорєць Г.О.

Керівник

Жданова О.Г.

Класифікація задач маршрутизації транспортних засобів

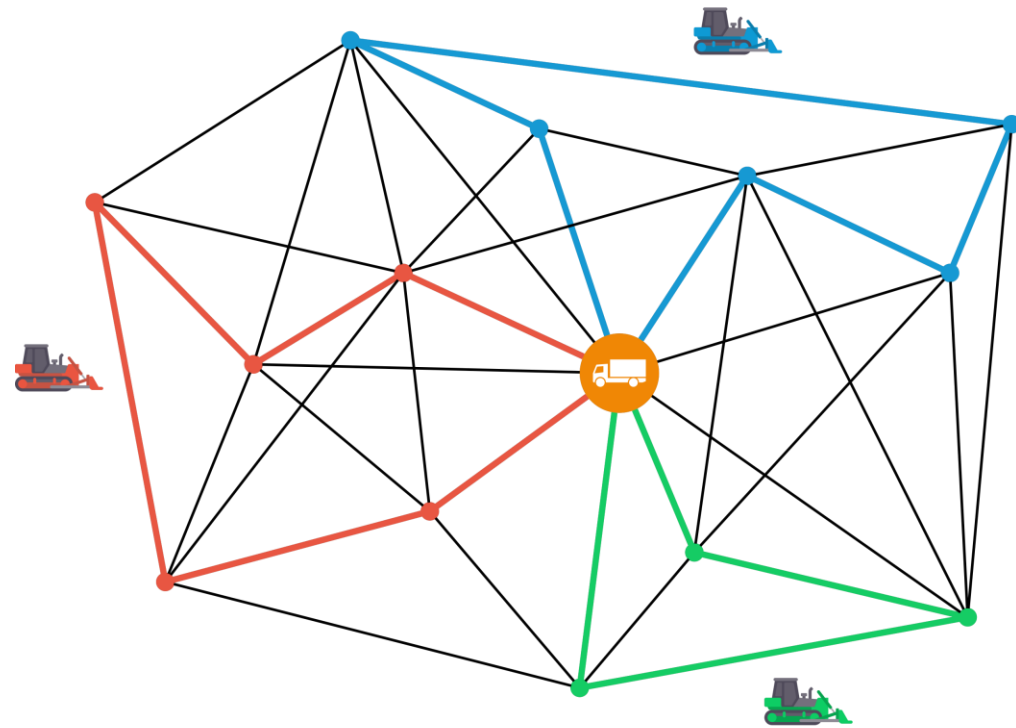


Рисунок А.2 – Зображення задачі VRP

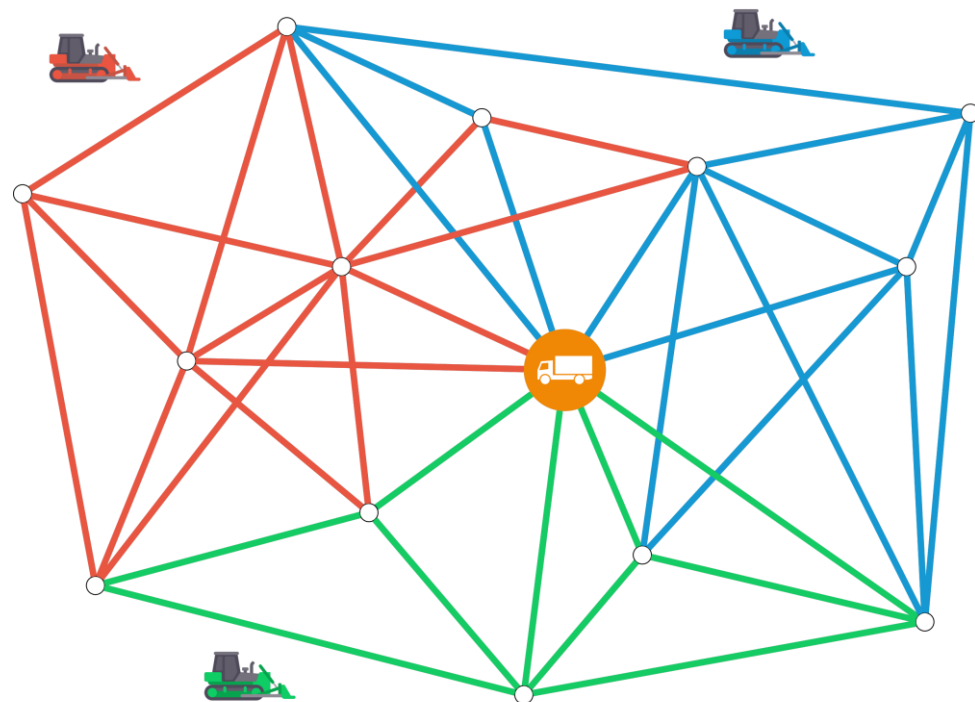


Рисунок А.3 – Зображення задачі ARP

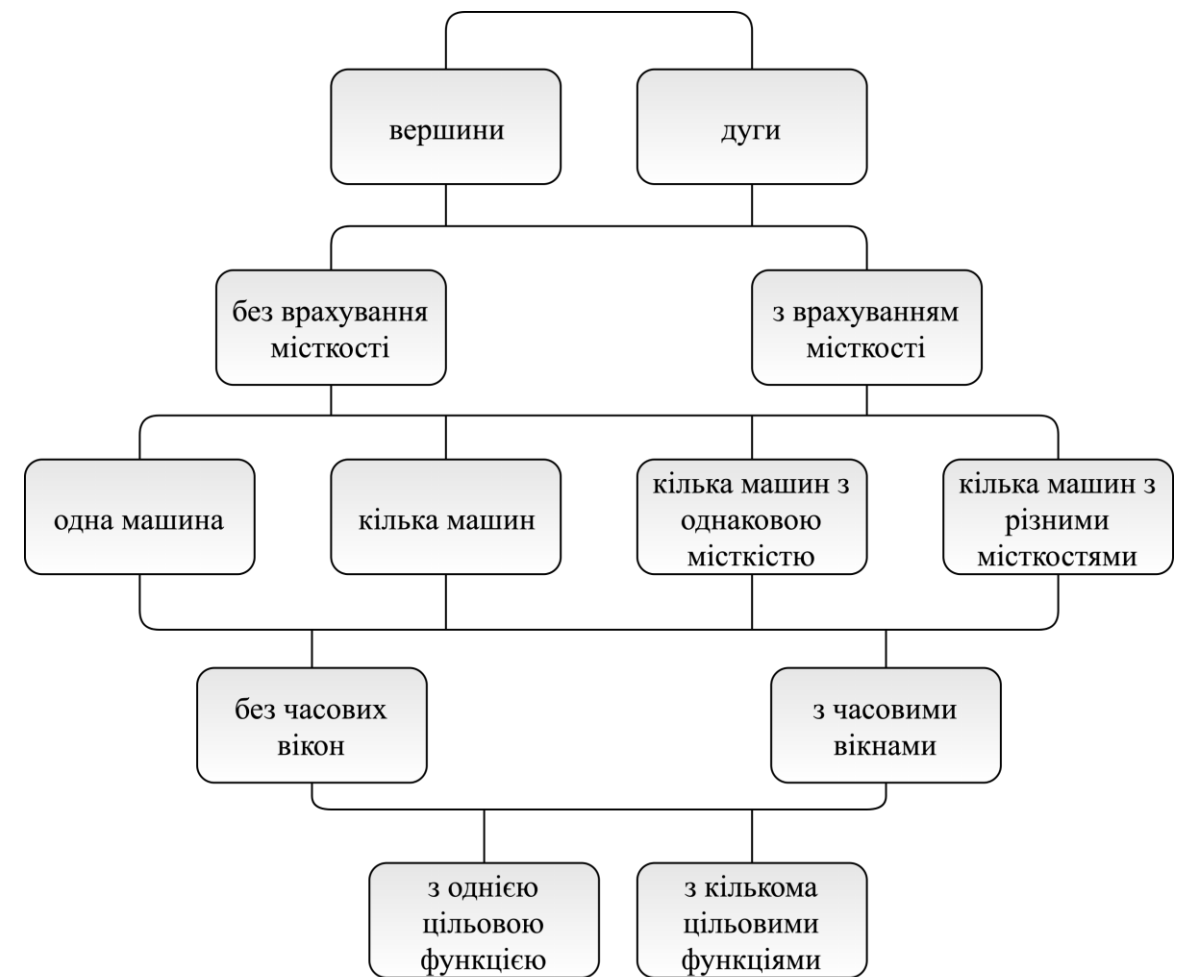


Рисунок А.4 – Схема класифікації задач VRP

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант

Григорєць Г.О.

Керівник

Жданова О.Г.

Змістовна постановка задачі MM k -CCPP with DC

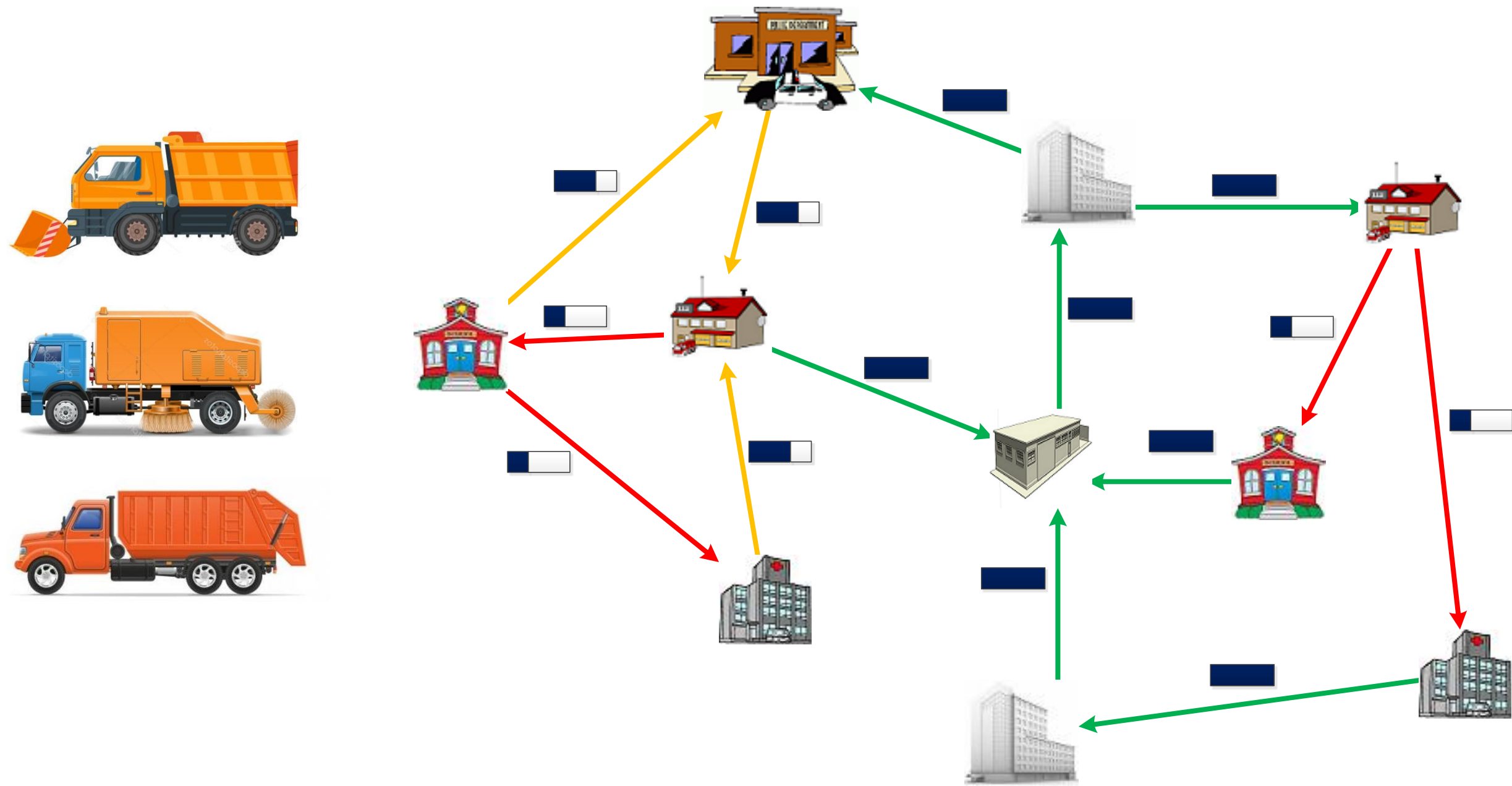


Рисунок А.5 – Зображення змістовної постановки задачі MM k -CCPP with DC

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант

Григорєць Г.О.

Керівник

Жданова О.Г.

Математична модель задачі ММ k -ССРР with DC

Дано:

$G = (V, A = S \cup C)$ – граф задачі;

$V = \{v_0, v_1, v_2, \dots\}$ – множина вершин, де $v_0 \in V$ – вершина-депо;

$A = \{a_1, a_2, \dots\}$ – множина всіх дуг, яка з'єднує між собою вершини з V ;

S – множина дуг для обслуговування;

C – множина транзитних дуг,

Q – місткість однієї машини.

$S_1 \cup \dots \cup S_n = S, \forall i_1, i_2, i_1 \neq i_2 : S_{i_1} \cap S_{i_2} = \emptyset$ – поділ дуг обслуговування на класи пріоритетів

⇓

$T_1 < T_2 < \dots < T_n$ – директивні терміни виконання робіт

Знайти:

P_1, \dots, P_k – маршрути для k машин.

Змінні:

$x_{ijh} \in \{0, 1\}$ – обслуговування i -ою машиною j -ої дуги в позиції маршруту h ;

$x_{ij} = \sum_h x_{ijh}$ – обслуговування j -ої дуги i -ою машиною;

$y_{ijh} \in \{0, 1\}$ – транзитне проходження i -ою машиною j -ої дуги в позиції маршруту h ;

$y_{ij} = \sum_h y_{ijh}$ – транзитний проїзд по j -ій дузі i -ою машиною;

i^* – номер маршруту, в якому обслуговується дуга a_i ;

h^* – позиція дуги $a_i \in S$ у маршруті i^* ;

$t_j = \sum_{h=1}^{h^*} (c_j z_j y_{i^*jh} + s_j z_j x_{i^*jh})$ – час завершення обслуговування j -ої дуги;

$W_i = \sum_j (s_j x_{ij} z_j + c_j y_{ij} z_j)$ – час проходження i -ого маршруту.

Характеристики дуги a_i :

b_i – вершина, де починається дуга;

e_i – вершина, де закінчується дуга;

s_i – час обслуговування дуги;

c_i – час транзитного проходження по д $\Rightarrow s_i \geq c_i$

z_i – коефіцієнт інтенсивності руху на дузі;

r_i – обсяг ресурсу, який необхідний для обслуговування дуги.

Цільова функція:

$$\max(W_i) \rightarrow \min, i = \overline{1, k}$$

Обмеження:

$$\forall j \in A : s_j \geq c_j$$

$$\forall P, i = \overline{1, l} : e_{[i-1]} = b_{[i]}, b_{[1]} = v_0, e_{[l]} = v_0$$

$$\forall j : \sum_i x_{ij} = 1$$

$$\forall i, j, h : x_{ijh} + y_{ijh} = 1$$

$$\forall i : \sum_j r_j x_{ij} \leq Q$$

$$\forall j \in S_i : t_j \leq T_i$$

$$x_{ijh} \in \{0, 1\}, y_{ijh} \in \{0, 1\}$$

Демонстраційний плакат до магістерської дисертації

на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант

Григорєць Г.О.

Керівник

Жданова О.Г.

Схема структурна алгоритму розв'язання задачі MM k -CCPP with DC

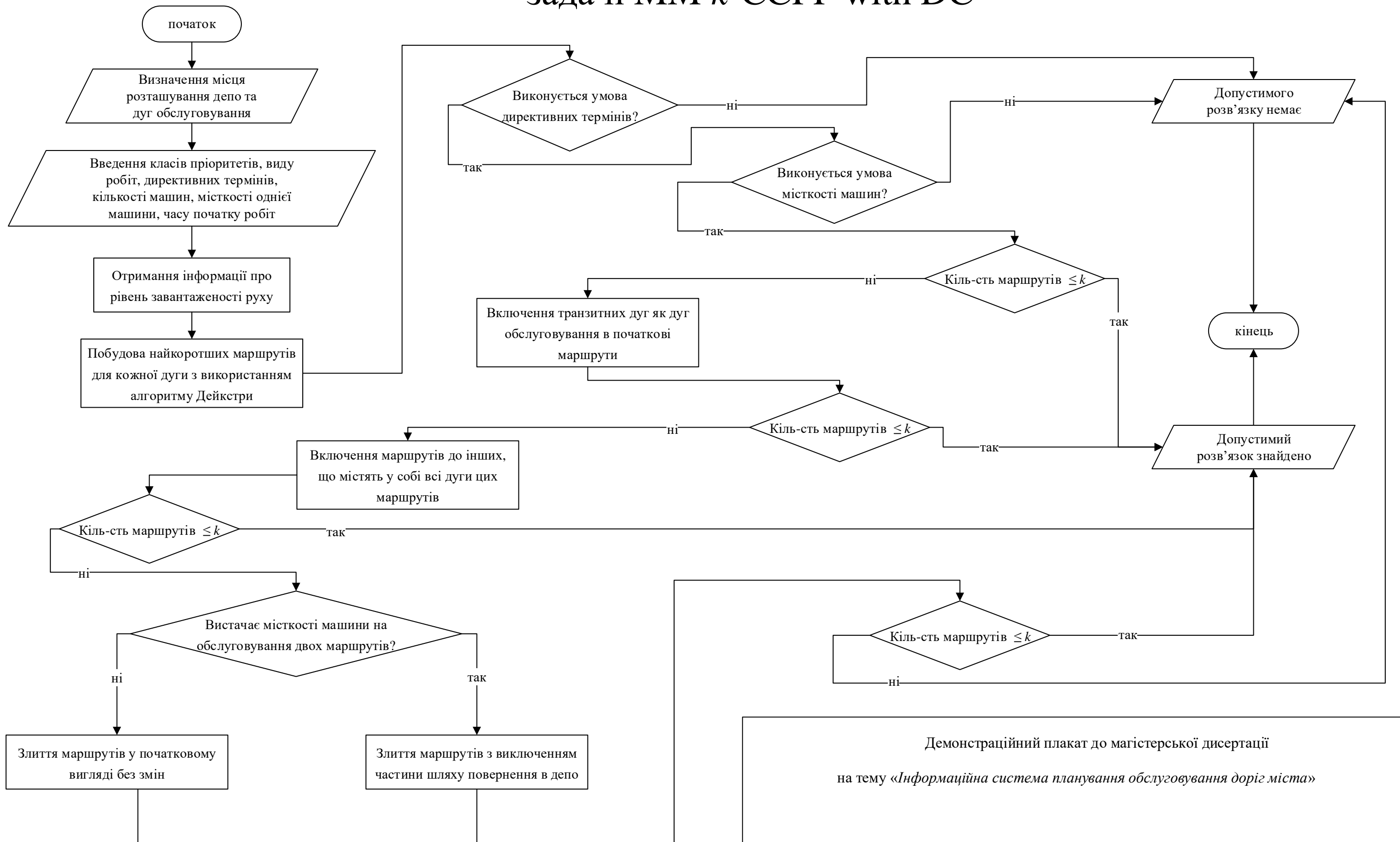


Рисунок А.6 – Схема структурна алгоритму розв'язання задачі MM k -CCPP with DC

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант
Керівник

Григорєць Г.О.
Жданова О.Г.

Схема структурна класів інформаційної системи

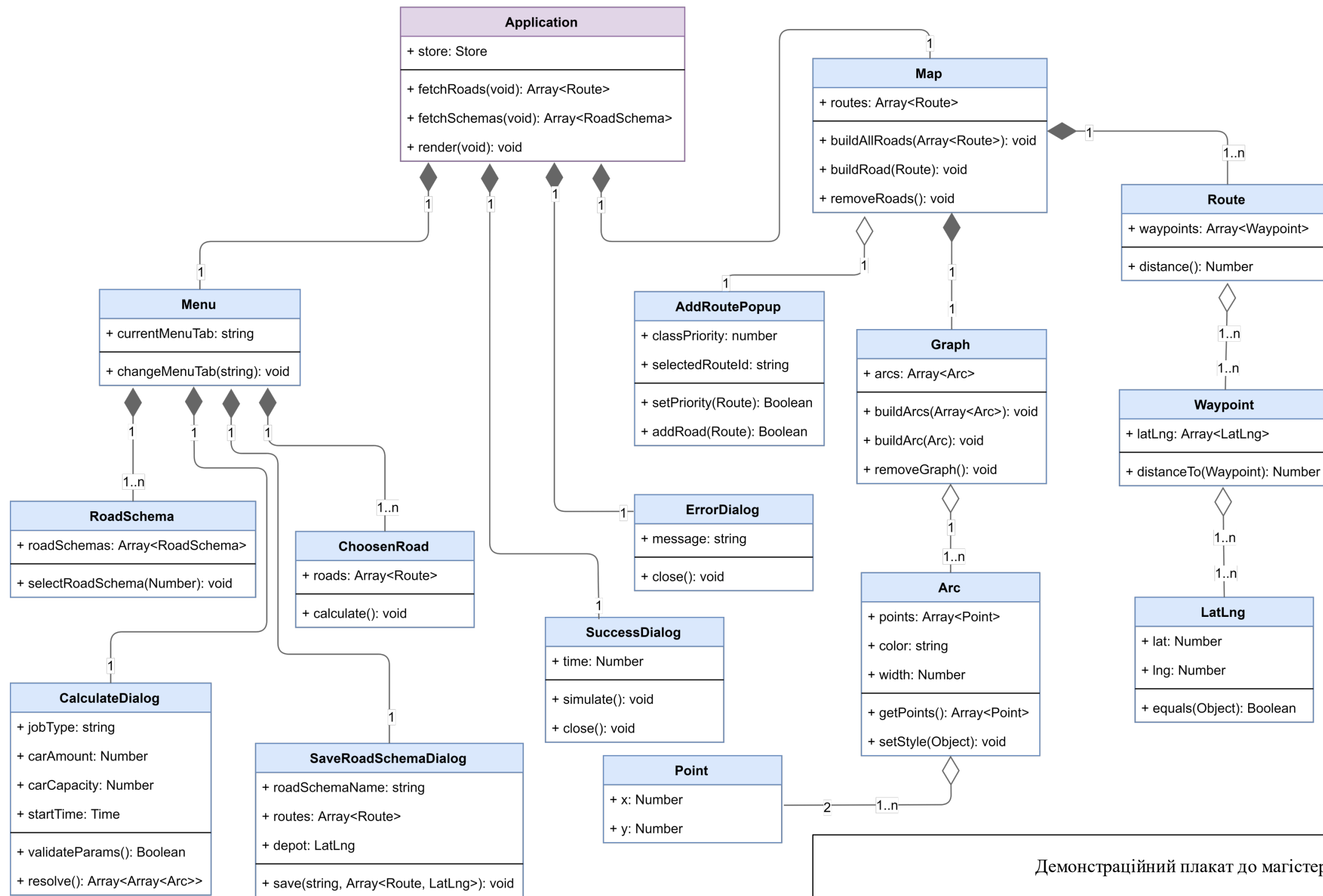


Рисунок А.7 – Схема структурна класів інформаційної системи планування обслуговування доріг міста

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант
Керівник

Григорєць Г.О.
Жданова О.Г.

Екранні форми

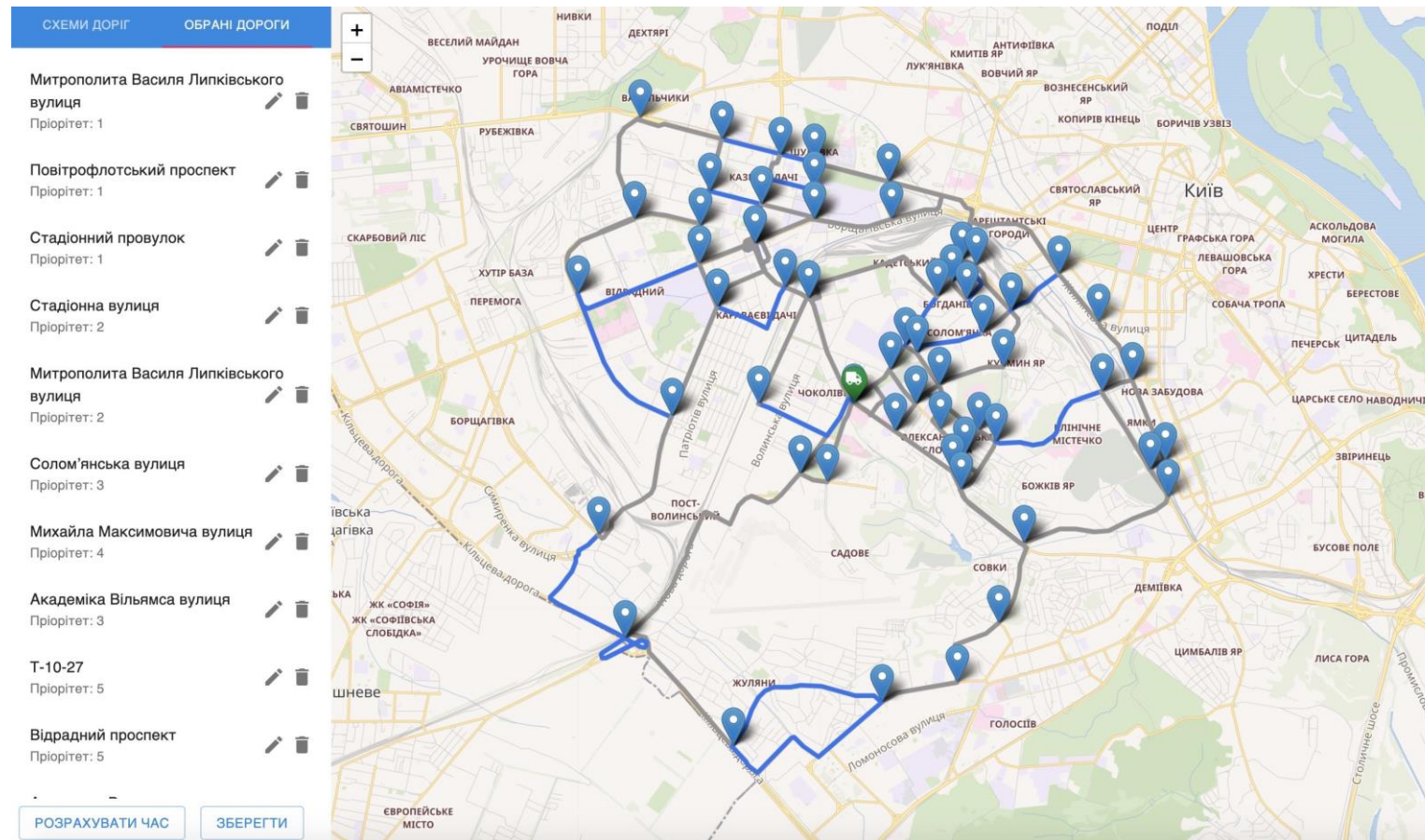


Рисунок А.8 – Район обслуговування з активними дорогами

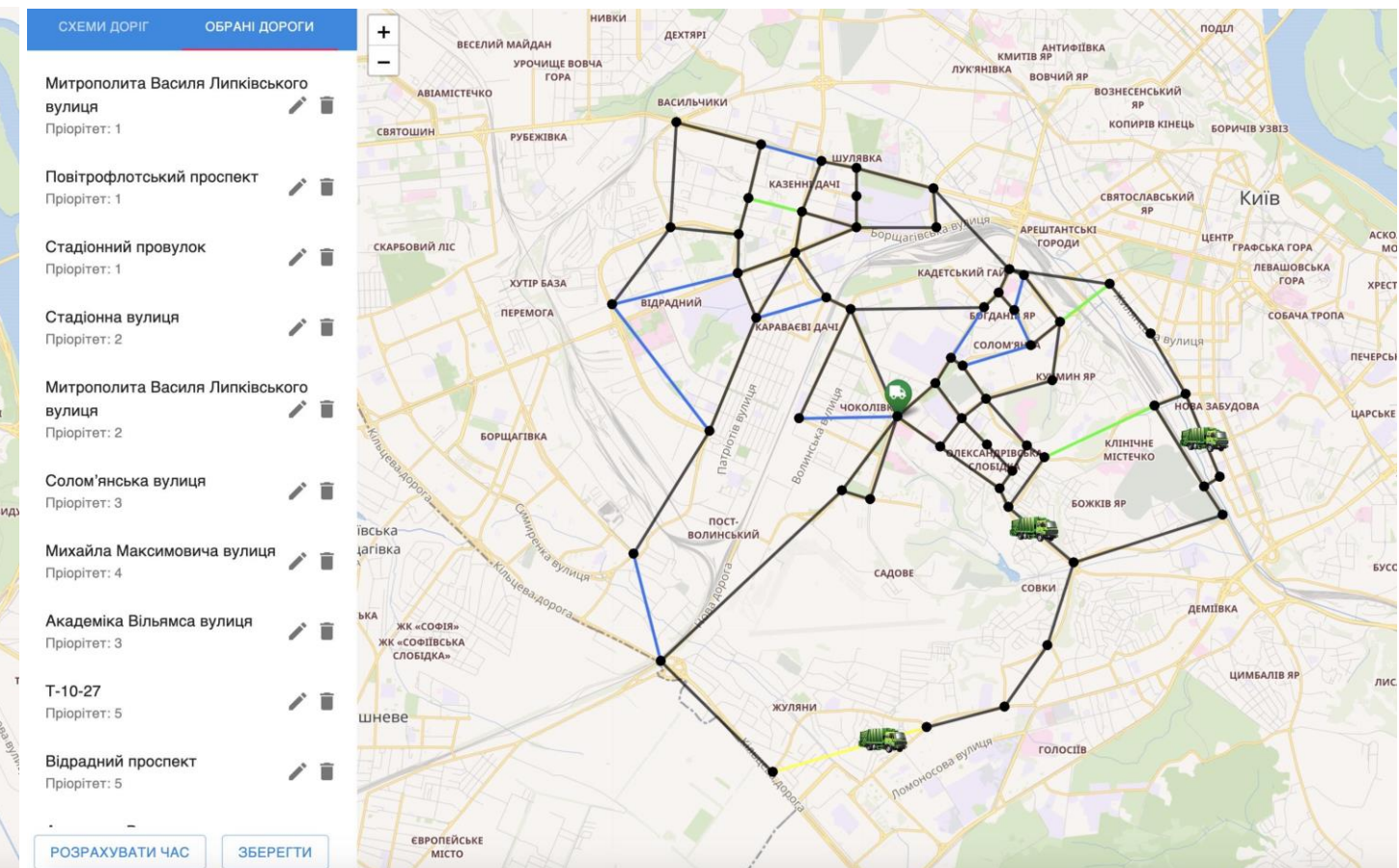


Рисунок А.9 – Симуляція руху транспортних засобів

Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна система планування обслуговування доріг міста»

Магістрант

Григорєць Г.О.

Керівник

Жданова О.Г.