

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛТЕХНІЧНИЙ ІНСТИТУТ»**

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

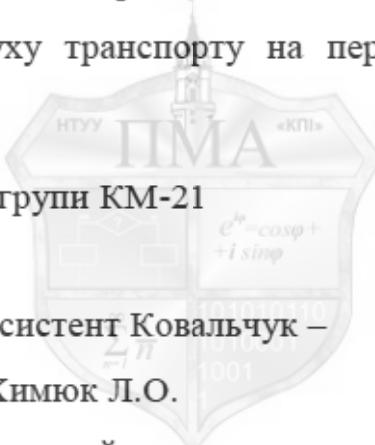
_____ О. Р. Чертов

«____» _____ 2016 р.

**Дипломна робота
на здобуття ступеня бакалавра**

з напрямку підготовки 6.040301 «Прикладна математика»

на тему: Моделювання руху транспорту на перехрестях у задачах оптимізації процесу регулювання



Виконав: студент IV курсу, групи КМ-21

Лимар Борис Олегович

Керівник

асистент Ковальчук —

Химюк Л.О.

Консультант із

старший викладач

нормоконтролю

Мальчиков В. В.

Рецензент

к.т.н., доцент

Тимошенко Ю.О.

Засвідчую, що в цій дипломній роботі
немає запозичень із праць інших авторів
без відповідних посилань.

Студент _____

Київ — 2016

Національний технічний університет України

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2015 р.

ЗАВДАННЯ
на дипломну роботу студента

Лимарю Борису Олеговичу

1. Тема роботи: «Моделювання руху транспорту на перехрестях у задачах оптимізації процесу регулювання»,
керівник роботи Ковальчук – Химюк Людмила Олександрівна, асистент
затверджені наказом по університету від «06» травня 2016 р. № 1499-С.

2. Термін подання студентом роботи: «13» червня 2016 р.

3. Вихідні дані до роботи:

- транспортна мережа Шевченківського району міста Києва (файл з координатами перехресть);

4. Зміст роботи:

- виконати аналіз існуючих методів розв'язання задачі;
- вибрати методи моделювання роботи світлофора та методи оптимізації руху транспорту на перехрестях;
- спроектувати автоматизовану систему моделювання руху транспорту на перехрестях для оптимізації процесу регулювання;
- здійснити програмну реалізацію розробленої системи;
- провести випробування розробленої системи.

5. Перелік ілюстративного матеріалу: блок-схеми розроблених алгоритмів, схема взаємодії модулів системи, знімки екранних форм.
6. Дата видачі завдання: «22» лютого 2016 р.



Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних	12.11.2015	
2	Проведення порівняльного аналізу математичних методів побудови оптимального маршруту	14.12.2015	
3	Проведення порівняльного аналізу математичних методів пошуку максимального потоку транспортної мережі	24.12.2015	
4	Підготовка матеріалів першого розділу роботи	01.02.2016	
5	Розроблення математичного забезпечення для моделювання руху транспорту на перехрестях та оптимізації процесу регулювання	01.03.2016	
6	Підготовка матеріалів другого розділу роботи	15.03.2016	
7	Підготовка матеріалів третього розділу роботи	05.04.2016	
8	Розроблення програмного забезпечення для моделювання руху транспорту на перехрестях та оптимізації процесу регулювання	15.04.2016	
9	Підготовка матеріалів четвертого розділу роботи	03.05.2016	
10	Підготовка матеріалів п'ятого розділу роботи	10.05.2016	
11	Оформлення пояснівальної записки	01.06.2016	

Студент _____

Лимар Б.О.

Керівник роботи _____

Ковальчук – Химюк Л.О.

АНОТАЦІЯ

Дипломну роботу виконано на 75 сторінках, вона містить 2 додатки та перелік посилань на використані джерела з 16 найменувань. У роботі наведено 19 рисунків та 1 таблиця.

Метою даної дипломної роботи є розробка програмного комплексу для моделювання руху транспорту на перехрестях для оптимізації процесу регулювання.

В рамках дипломної роботи розглядаються існуючі системи побудови оптимального маршруту при моделюванні руху транспорту на перехрестях та їх функціональні можливості. В роботі розглядається математична модель транспортної мережі, алгоритми пошуку оптимального маршруту та максимального потоку.

Для розв'язку задач пошуку оптимального шляху та максимального потоку були обрані алгоритми Дейкстри та Форда – Фалкерсона. Для моделювання транспортних потоків використовувався рівномірний закон розподілу.

Було спроектовано та реалізовано автоматизовану систему для моделювання руху транспорту на перехрестях для оптимізації процесу регулювання.

Ключові слова: транспортна мережа, оптимальний маршрут, максимальний потік, алгоритми на графах.

ABSTRACT

The thesis is presented in 75 pages. It contains 2 appendixes and bibliography of 16 references. 19 images and 1 table are given in the thesis.

Software complex for transport move on the crosses modeling and regulation process optimization development is the goal of this work. During work existing systems for optimal root building during transport move on the crosses modeling and their functional abilities were considered. Transport system mathematical model, algorithms of optimal root and maximum flow search were considered in the work.

Deikstra's and Ford-Falkerson's algorithms were chosen for optimal root and maximum flow problems solution. Uniform distribution law was used for transport flows modeling.

Information system for transport move on the crosses for regular process optimization modeling was designed and implemented.

Key words: transport network, optimal root, maximum flow, algorithms on graphs

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	9
Вступ.....	10
1 Постановка задачі.....	11
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	13
2.1 Огляд онлайн сервісів для побудови маршруту	13
2.2 Переваги і недоліки існуючих онлайн сервісів	15
2.3 Класифікація АСУР	16
2.4 Огляд існуючих АСУР	18
2.5 Висновки до розділу.....	21
3 МАТЕМАТИЧНІ МЕТОДИ РОЗВ'ЯЗКУ ПОСТАВЛЕНОЇ ЗАДАЧІ	23
3.1 Опис математичної моделі.....	23
3.2 Алгоритми на графах	25
3.3 Моделювання потоку	28
3.3.1 Рівномірний закон розподілу	28
3.3.2 Гаусовий закон розподілу	29
3.3.3 Пуасонівський закон розподілу.....	29
3.4 Висновки до розділу.....	30
4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	31
4.1 Структура системи	31
4.2 Опис розроблених програмних засобів та структурна схема	32
4.3 Опис розробки і інтерпретація алгоритмів	34
4.4 Опис графічного інтерфейсу	37
4.5 Висновки до розділу.....	41
5 ВИПРОБУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	42
5.1 Контрольні приклади	42
5.2 Аналіз отриманих результатів	46

5.3 Висновки до розділу.....	47
ВИСНОВКИ	48
Перелік посилань	49
Додаток А Лістинги програм.....	51
Додаток Б Ілюстративний матеріал	67



ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АСУ – автоматизована система управління.

АСУР – автоматизована система управління дорожнім рухом.

КУ – координоване управління.



ВСТУП

На сьогоднішній час у світі існує багато проблем пов'язаних з науковим прогресом, одна з них - масова урбанізація суспільства, що, в свою чергу, породжує низку проблем, що потребують ефективного вирішення. Концентрація автомобільного транспорту різко погіршує екологічне та соціальне благополуччя громадян, саме тому дана проблема потребує кваліфікованого підходу щодо методів її усунення.

Існує багато методів для боротьби з такими проблемами. Один з шляхів їх вирішення є створення модифікованих автомобілів в яких знижене вироблення шкідливих газів.

На сьогоднішній день затори складають невід'ємну частину життя кожного водія і мають багато негативних наслідків. Наприклад, збільшення аварійності на дорозі, шуму, зносу автомобіля, порушення роботи екстрених і оперативних служб і т.д.. Тому створення систем для розв'язання даних проблем є важливим на теперішній час[1].

Дана робота присвячена розробці програмного комплексу моделювання руху транспорту на перехрестях для оптимізації процесу регулювання. Актуальність дипломної роботи полягає в спробі зробити оптимізацію регулювання світлофору в залежності від потоку.

1 ПОСТАНОВКА ЗАДАЧІ

Мета роботи – оптимізація процесу регулювання на перехрестях на основі моделювання руху транспорту. Для реалізації мети необхідно спроектувати та реалізувати програмний комплекс.

Для проектування програмного комплексу виділено 2 підзадачі:

- 1) моделювання роботи світлофорів;
- 2) оптимізація руху транспорту на перехрестях.

В першій підзадачі виділені наступні завдання:

- моделювання роботи світлофора з фіксованим часом переключенням;
- моделювання роботи світлофора, що враховує потік машин на перехрестях.

В другій підзадачі виділені наступні завдання:

- знаходження найкоротшого шляху для поїздки;
- моделювання проїзду з врахуванням роботи світлофорів;
- знаходження максимального потоку між початковою вершиною і кінцевою;
- моделювання проїзду з врахуванням часу поїздки.

1.1 Вимоги до програмного комплексу

Програмний комплекс має давати користувачеві наступні функції:

- вибір введення початкової точки, з якої користувач почне свій рух, та кінцевої;
- автоматизовану генерацію перехресть на карті та світлофорів;
- оцінювання похибки комп'ютерних обчислень;

Програмний комплекс має виконуватися в операційних системах Windows 7,8,10 на IBM-сумісному комп'ютері, до складу якого входять:

- x64-сумісний процесор із тактовою частотою, не меншою за 1,5 ГГц;
- оперативна пам'ять обсягом, не меншим за 256 МБ.



2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

На сьогоднішній день існує багато, як офлайн так і онлайн систем , для побудови оптимального маршруту[2].

Серед найбільш поширеніших систем побудови оптимального маршруту можна виділити наступні системи:

- онлайн сервіси та додатки, які можна використовувати через браузер комп'ютера або іншого пристроя з доступом до мережі Інтернет;
- вбудоване програмне забезпечення GPS-навігаторів, що реалізує функцію побудови маршруту;
- програми та додатки для мобільних пристройів;
- програмне забезпечення для використання на стаціонарних персональних комп'ютерах.

2.1 Огляд онлайн сервісів для побудови маршруту

Найбільшою популярністю користуються онлайн сервіси до яких відносяться: Google Maps і Яндекс Карти .

За допомогою сервісу – Яндекс Карти можна шукати за адресами, вулицями міст, регіонами, країнами та організаціями, вимірювати відстані між географічним об'єктами та прокладати автомобільні маршрути. Для певних міст доступний індикатор ситуації на дорогах — сервіс Яндекс.Затори. Рівень завантаженості доріг подано як чотиріколірну графічну та десятибалльну цифрову шкалу. Дані Яндекс заторів можуть враховуватися під час автоматичного прокладання маршрутів. Інформація про дорожні події, що отримується з мобільних Яндекс карт, також впливає на рекомендований маршрут. На картах є інтерактивні схеми транспортних

розв'язок з рекомендаціями, як оптимально ними проїхати. На сервісі можна в реальному часі переглядати зображення з веб-камер[3].

Функції:

- а) показується швидкість, очікуваний час прибуття і відстань до місця прибуття;
- б) програма показує схеми будинків та автоматично перемикається із тривимірного режиму у двовимірний. Доступний вид із супутника;
- в) підтримує векторні карти України;
- г) є пошук потрібних об'єктів поблизу (кінотеатри, ресторани, заправки тощо);
- г) доступна навігація усіма містами України і трасами між Україною, Росією та Білоруссю;
- д) маршрут прокладається через проміжні точки, є голосовий пошук;
- е) має функцію голосових підказок;
- є) у програмі також є розпізнавання голосових команд;
- ж) україномовний інтерфейс;
- з) доступна для смартфонів та планшетів на платформах iOS, Android, Windows Phone.

Google Maps – сервіс корпорації Google Inc. Даний сервіс дозволяє прокладати оптимальний автомобільний маршрут в якості критерію оптимальності використовується відстань між об'єктами та час витрачений на маршрут. На картах сервісу є можливість відображення рівню заторів. В опціях побудови маршруту можна обрати варіанти побудови маршруту без руху по шосе, платних дорогах та поромні переправах. Після побудови маршруту Google Maps дає два маршрути: оптимальний та альтернативний. Для кожного маршруту вказуються відстань та час необхідний для руху[4].

Функції:

- а) інформація про розклади та маршрути руху громадського транспорту;
- б) пошук маршруту за точками;
- в) рівень завантаженості доріг;

- г) показується швидкість, очікуваний час прибуття і відстань до місця прибуття;
- г) програма показує схеми будинків та автоматично перемикається із тривимірного режиму у двовимірний. Доступний вид із супутника;
- д) підтримує векторні карти України;
- е) є пошук потрібних об'єктів поблизу (кінотеатри, ресторани, заправки тощо);
- є) маршрут прокладається через проміжні точки, є голосовий пошук;
- ж) має функцію голосових підказок.;
- з) інформація про маршрути доступна в усіх браузерах, включно з мобільними.

2.2 Переваги і недоліки існуючих онлайн сервісів

Дані онлайн сервіси: Google Maps та Яндекс.Карти, крім довідкової інформації, відкривають широкі рекламні та функціональні можливості. Наприклад, з їх допомогою можна переглядати і розміщувати дані про місцеві компанії, включаючи адресу, контактну інформацію і маршрути проїзду. А за допомогою API, інструмента, дозволяю чого розробникам отримати доступ до частин сервісу карт, їх можна інтегрувати у власні сайти.

Найбільш вагомий аргумент на користь вибору карт Яндекса - їх орієнтація на Росію і Україну, де відмінно промальовані будинки і вулиці кожного міста. Визначення місцезнаходження користувача за ір-адресою (геортаргетінг)[3].

Поки що в Google Maps країни СНД опрацьовані менш детально, але в загальносвітовому масштабі Google - однозначний фаворит.

Більшості людей, виявляється, не так просто зорієнтуватися на карті рідного міста, вони не звичали бачити його в такому ракурсі. У Яндекса існує проект «Народна карта», на якій користувачі самі створюють позначки, орієнтири і

покажчики. На «Народної карті» відзначені зупинки транспорту, кінотеатри, площі, супермаркети, за якими користувачеві часто буває простіше зорієнтуватися.

На основі своєї системи Яндекс дозволяє користувачам створювати власні електронні інтерактивні карти. Схема офісу або підземних комунікацій, план корабля або ринку, навігатор по фотографії з високою роздільною здатністю - трохи зусиль, і все це можливо.

В цілому, у Яндекс.Карт ширший функціонал і API налаштування:

- для більш зручного орієнтування, в кутку екрану можна розташувати міні карту;
- карті можна прикріпити настроюються функціональні кнопки за смаком, такі як «маршрути», вимірювач відстаней «лінійка» та інші;
- присутня можливість малювати динамічні полігони;

Яндекс випустив свій сервіс карт пізніше, ніж Google. Цим і пояснюється краща продуманість API Яндекса, що нагадує оптимізовану версію API Google Maps V2 (навіть імена функцій часто збігаються). Одержані код трохи компактніше і гнучкіше. Деякі функції, наприклад, позиціонування карти після кліка по маркеру в Google Maps треба прописувати вручну, а у Яндекса вони встановлені за замовчуванням. Також Яндекс впровадив в API підтримку популярного фреймворку jQuery[5].

2.3 Класифікація АСУР

АСУР розділяється на наступні види по їх призначенню та ступені технічної оснащеності:

До магістральних АСУ координованого управління (КУ) відносяться:

- безцентрові АСУ КУ;
- централізовані АСУ КУ;

- централізовані інтелектуальні АСУ КУ.

До загальноміських АСУР (ЗАСУР) відносяться:

- спрощені ЗАСУР;

- інтелектуальні ЗАСУР.

Безцентрові АСУ КУ характеризуються тим, що для них відсутня необхідність створення центру управління. Вони виконуються в двох модифікаціях. Синхронізацію роботи контролерів задає один з них, який є головним. Цей контролер, який називається «координатор», пов'язаний лінією зв'язку з кожним із кожним контролером, які залишилися, причому ця лінія може бути або однією для всіх або до неї підключаються контролери паралельно інші контролери (така система називається багато точковою або паралельною), або до кожного контролера прикладена свої лінії зв'язку[6].

Централізовані АСУ КУ характеризуються наявністю центру управління, пов'язаного з контролерами радіальними лініями зв'язку. Як правило, централізовані АСУ КУ характеризуються можливістю здійснювати багато програмне КУ з переключенням програм по часу доби.

Централізовані інтелектуальні АСУ КУ характеризуються тим, що в їх складі на даній транспортній мережі з'являються детектори транспорту, інформація від яких передається по лініям зв'язку в центр управління, в якому встановлюється персональна ЕОМ, яка має можливість змінювати плани координації в залежності від транспортної ситуації на магістралі.

Загальноміські АСУР характеризуються підключенням до центру управління не тільки однієї магістралі, на якій реалізується КУ, а всіх магістралей з КУ. Крім того, подібні системи мають в своєму складі контур диспетчерського управління, який включає в себе підсистему телевізійного надзору за рухом, підсистему відображення інформації про дорожню ситуації і засоби безпосереднього диспетчерського управління світлофорною сигналізацією і знаками, які управляються диспетчерським персоналом центром управління.

Інтелектуальні ЗАСУР включають в себе потужні керуючі обчислювальні комплекси, які розташовуються в центрі управління рухом і мережу динамічних інформаційних табло, які розташовуються в стратегічних точках дорожньої мережі. Такі системи здійснюють неперервний автоматичний моніторинг транспортних потоків в дорожній мережі і на основі зібраної інформації не тільки дозволяють керуючим обчислювальним комплексам здійснювати автоматичне адаптивне управління рухом, але і забезпечують учасників руху за допомогою динамічних інформаційних табелів інформацією про транспортну ситуацію і тим самим дозволяють розподілити транспортні потоки по мережі.

Інтелектуальні ЗАСУР дозволяють управляти дорожнім рухом на міських магістралях неперервного руху в комплексі з мережевим координованим світлофорним регулюванням. Задача такої системи складається з роботи в трьох напрямках. В одному з них – це координоване управління роботою виїздів на дорогу неперервного руху з цілю забезпечення резерву пропускої здатності на ній, в тому числі забезпечення цієї самої неперервності. Друге направлення – це управління з'їздами на магістралі звичайного типу. Якщо на них в точках з'їздів існує затор, то задача системи – обмежити з'їзд, з тим, щоб черга на ньому не почала блокувати магістраль неперервного руху. Третє направлення – це автоматичне визначення дорожньої транспортної пригоди або затору на магістралі і забезпечення диспетчера інформацією про те що сталося.

2.4 Огляд існуючих АСУР

В сучасних автоматизованих системах управління дорожнього руху, поширених в більшості європейських державах, широко використовується інформація з відеокамер, які входять в склад підсистем відеоконтролю. Отримана від них інформація дозволяє організувати оптимальне управління транспортними

потоками, скоординувати роботу ключових транспортних вузлів міста і т.д. . Перевагою систем відеоконтролю є поєднання числової та візуальної інформації, яка радикально відрізняється від інших систем спостереження. Наприклад, можлива організація моментального зворотнього зв'язку з оператором системи, диспетчера управління при появі якої-небудь позаштатної ситуації для звичайної перевірки системи.

Принцип роботи системи відеоконтролю добре відомий. Над певною ділянкою траси, транспортним вузлом, магістраллю, небезпечною ділянкою дороги на деякій висоті встановлюється відеокамера. Сигнал від неї поступає в модуль обробки відеоінформації. В цьому модулі відбувається виділення рухомих транспортних засобів і визначення інтегральних оцінок. Далі в центрі управління можуть бути отримані як числові дані, для цього достатньо каналу з низькою пропускною здатністю. Так і безпосередньо відеозображення с ділянкою під управлінням[7].

Системи відеоконтролю, орієнтовані на транспорт, надають дані трьох типів:

1. Інформація про трафік для статистичної обробки:

- загальне число виявлених автомобілів;
- швидкість;
- прискорення транспортного потоку;
- занятість полос руху;
- класифікація автомобілів.

2. Інформаціях про пригоди на дорозі:

- висока швидкість, щільність потоку або занятість полос;
- наявність заторів або руху по зустрічній смузі;
- автомобілі, які повільно рухаються або зупинились;
- наявність на дорогах підозрілих предметів.

3. Інформація про наявність/відсутність автомобілів:

- наявність автомобілів, які наближаються;
- наявність автомобілів, які зупинились на перехресті;
- число автомобілів, які проїхали через зони виявлення;

- вимірювання довжини черги.

Останній тип інформації широко застосовується в системах управління світлофорами. Система відеоконтролю інтегрована в модуль управління світлофорами, що дозволяє скоординувати роботу всіх світлофорів перехрестя в якому – небудь напруженому транспортному вузлі.

В багатьох країнах світу чітко налагоджена інформація руху про транспортну ситуацію на напрямках руху, про можливі маршрути об'їзду перезавантажених ділянках, про парковки. На перетині доріг вказуються не тільки дозволені напрямки руху, але і назви районів і вулиць. Для передачі водіям інформації використовуються багатопозиційні дорожні знаки, світлові табло зі змінною інформацією, спеціальні радіо та відеоканали. Наприклад, після включення світлових табло з попередженням про затори, вони усувалися за 20 – 30 хвилин; без табло на це потрібно було 3 – 4 години.

В європейських державах поштовхом до технічної модернізації системи управління і контролю за рухом автотранспорту став досвід Франції.

Управління світлофорами в містах почало проводитись із єдиного центру; на основних трасах були встановлені нові камери пов'язані з радарами, які автоматично засікали перевищення швидкості, фіксували на плівку номер автомобіля, обличчя власника. Ці дані передавались на центральний комп'ютер, який без участі людини виписував штраф господарю автомобіля.

Завдяки цим нововведенням кількість дорожніх транспортних пригод знизилась за два роки на третину. Тим не менше, в інших державах існують свої специфікації особливостей технічної організації дорожнього руху.

У Великій Британії були розроблені «транспортні відеокамери», які мали підвищений рівень безпеки на дорогах за рахунок регулювання швидкості руху. Нові пристрої – це вмонтовані в дорожнє полотно світлові маяки, які за допомогою відеокамери визначають швидкість проїжджаючих автомобілів, знос їх покришок і ідентифікують їх номерні знаки.

Коли швидкість транспорту, який наближається, вимірюна, пристрій починає працювати відповідно до світлофору – світлодіоди подають автомобілістам світлові сигнали від червоного до зеленого. Дані, отримані завдяки маякам, не будуть використовуватись для стягнення штрафу – це система попередження учасників дорожнього руху, а не покарання.

В Японії з 2006 року з'явились «розумні» номери, оснащені вбудованим мікрочіпом, який запам'ятовують та передають інформації про номер автомобіля, його розмірі, місці реєстрації та господаря.

Основна ціль полягає в обмеженні за допомогою сучасних технологій скучення автомобілів в часи пік на центральних магістралях японських міст. Крім оптимізації транспортних потоків система позитивно впливає на стан зовнішнього оточення.

Проблема скучення автомобілів є важливою темою на сьогоднішній день. Першими платити за в'їзд в центр міста почали автомобілісти Сінгапура. В свою чергу Японія стала першою в реалізації «розумних» номерів у світі для розв'язку даної проблеми.

2.5 Висновки до розділу

Сервіси, які були описані в даному розділі займаються пошуком оптимального маршруту на карті з врахуванням також заторів та деталізовано відображають картографічні дані.

Були визначені основні переваги та недоліки цих сервісів: Яндекс Карти та Google Maps. Найбільш популярним сервісом в світі є Google Maps, який більш деталізовано відображає картографічні дані, а Яндекс Карти більш орієнтовані на країни СНД.

Були розглянуті та проаналізовані існуючі АСУР та їх використання і модернізації в різних країнах світу. До недоліків АСУР відносяться застаріле обладнання, неможливість оперативного управління дорожнім рухом і невелике число об'єктів управління, як правило, до 20 перехресть.



3 МАТЕМАТИЧНІ МЕТОДИ РОЗВ'ЯЗКУ ПОСТАВЛЕНОЇ ЗАДАЧІ

3.1 Опис математичної моделі

В якості математичної моделі карти дорожнього руху транспорту було обрано орієнтований граф з невід'ємними дугами.

Граф – це впорядкована пара (V, E) , де V – непуста множина вершин або вузлів, E – множина пар вершин, котрі називають ребрами. Множини V та E вважаються скінченими. Граф може бути як орієнтований – так і неорієнтований[8].

Орієнтований граф – це граф у якого множина ребер E , тобто пар вершин, є впорядкованою. В орієнтованому графі ребра називають орієнтованими або дугами.

Вагова функція – це таке відображення $\omega : E \rightarrow \mathbb{R}$, яке ставить у відповідність ребру або дузі графу деяке дійсне число, яке називається вагою ребра або дуги.

Зважений граф – це граф, у якого кожна дуга або ребро має свою вагу.

Вершина графа a називається досяжною з вершини b , якщо існує така послідовність дуг, яка з'єднує a та b .

Суміжні вершини – це вершини, які з'єднані ребром або дугою .

Транспортна мережа - орієнтований граф, в якому кожне ребро має невід'ємну пропускну здатність.

Пропускна здатність - метрична характеристика, що показує співвідношення граничної кількості одиниць (інформації, предметів, обсягу) в одиницю часу через канал, систему, вузол.

Мережа - зв'язний орієнтований граф без петель.

Потік мережі - деяка функція, яка ставить у відповідність дузі деяке число – вагу дуги.

Мережу автомобільних доріг, ліній комунікацій можна представити будь-яким графом, який необхідно буде використовувати в комп'ютерній програмі. При реалізації постає проблема збереження інформації про цей граф у пам'яті комп'ютера. Вибір структури даних для збереження графа в пам'яті має визначальне значення у процесі розробки ефективних алгоритмів.

Граф може бути представлений декількома способами. Наприклад, матрицею суміжності, матрицею інцедентності, списком суміжності, списком ребер [9].

Матриця суміжності. Таблиця, де як стовбці, так і рядки відповідають вершинам графу. В кожній комірці цієї матриці записується число, яке визначає наявність зв'язку від вершини – рядка до вершини – стовпчика або навпаки. Недоліком даного представлення є вимоги до пам'яті, прямо пропорційні квадрату кількості вершин.

Матриця інцедентності. Таблиця, де рядки відповідають вершинам графу, а стовпчики відповідають зв'язкам (ребрам) графу. В комірку матриці на перетині рядку i зі стовпчиком j записуються:

- 1, у випадку, якщо j «входить» із вершини i ;
- 1, якщо зв'язок «входить» у вершину;
- 0, у всіх інших випадках, тобто якщо зв'язок є петлею або зв'язок не інцендентний вершині.

Даний спосіб представлення є найбільш ємним для збереження, тому застосовується дуже рідко, лише в особливих випадках. Наприклад, для швидкого пошуку знаходження циклів в графі.

Список суміжності. Список, де кожній вершині графу відповідає рядок, в якому зберігається список суміжних вершин. Така структура даних не є таблицею в звичайному розумінні, а представляє список зі списків.

Розмір займаної пам'яті: $O(|V| + |E|)$.

Це найбільш зручний спосіб для представлення розріджених графів, а також при реалізації базових алгоритмів обходу графа в ширину або глибину, де потрібно швидко отримувати «сусідів» поточної вершини, яка переглядається.

Список ребер. Список, де кожному ребру графу відповідає рядок, в якому зберігається дві вершини, інцедентні ребр. Розмір займаної пам'яті: $O(|E|)$. Це найбільш компактний спосіб представлення графів, тому часто застосовується для зовнішнього збереження чи обміну даними.

З вище представлених способів була обрана матриця суміжності для представлення графу за рахунок простоти реалізації. Пропускні здатності графу також представлені у вигляді матриці.

3.2 Алгоритми на графах

Оптимальні маршрути і їх довжину між початковою вершиною і всіма іншими вершинами знаходять з допомогою алгоритмів Дейкстри і Белмана – Форда.

Алгоритм Белмана – Форда працює із дугами , які мають від'ємну вагу. Це означає, що найкоротшого шляху немає. Таким чином, в графі, який містить цикл з від'ємною сумарною вагою, існує короткий шлях від однієї вершини цього циклу до іншої.

Складність алгоритму Белмана – Форда дорівнює $O(n * m)$ порівняно з алгоритмом Дейкстри – $O(n^2 + m)$, де m – кількість ребер, n – кількість вершин.

Число $m = \frac{n(n-1)}{2}$, тому складність алгоритму Белмана – Форда є кубічною. Час роботи алгоритму Дейкстри залежить від реалізації не спадної черги з пріоритетами і складає $O(n * \lg(n) + m)$, якщо неспадна черга з пріоритетами реалізується за допомогою фібоначчевої піраміди[8]. Реалізація фібоначчевої піраміди дозволяє досягнути меншої складності алгоритму Дейкстри ніж Белмана – Форда .

Таким чином, для реалізації пошуку оптимального маршруту було обрано алгоритм Дейкстри.

Для розв'язання задачі про максимальний потік застосовується метод Форда – Фалкерсона[10]. Він базується на трьох важливих ідеях: залишкові мережі, збільшуочи шляхи і розрізи. Метод застосовується в багатьох потокових алгоритмах і задачах.

Під час роботи алгоритму спочатку знаходитьсья будь – який збільшуваний шлях. Збільшується потік по ребрах даного шляху на мінімальну величину із залишкових пропускних здатностей.

При невдалому виборі методу пошук максимального потоку може не закінчитись: величина потоку буде послідовно збільшуватись, але необов'язково буде сходитись до максимального значення потоку. Тому може бути таке, що алгоритм працюватиме нескінченно, якщо значення пропускної здатності ребер є ірраціональними числами.

Алгоритмічна складність алгоритму складає $O(Ef)$, де E – кількість ребер у графі, а f – максимальний потік. На кожному кроці алгоритму додається потік збільшуваного шляху до вже існуючого потоку. Якщо пропускна здатність всіх ребер – цілі числа, то можна довести методом математичної індукції, що і потоки через всі ребра будуть завжди цілими числами. Відповідно, на кожному кроці алгоритм збільшує потік хоча б на одиницю. Отже, він зійдеться не більше ніж за $O(f)$ кроків. Також можна виконати кожен крок за час $O(E)$, тоді загальний час алгоритму обмежений $O(Ef)$.

Задача про максимальний потік можна представити як задачу лінійного програмування. Змінними є потоки на ребрах, а обмеженнями - збереження потоку та обмеження пропускної здатності. Складність методу залежить від конкретного алгоритму. Так для симплекс – методу складність є поліноміальною.

Загальною (стандартною) задачею лінійного програмування називається задача знаходження мінімуму лінійної цільової функції вигляду:

$$f(x) = \sum_{j=1}^n c_j x_j$$

Задача, в якій фігурують обмеження в формі нерівностей, називається основною задачею лінійного програмування[11]:

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\geq b_i \quad (i = 1, 2, \dots, m) \\ x_j &\geq 0 \quad (j = 1, 2, \dots, n) \end{aligned}$$

Алгоритм Едмондса – Карпа також вирішує задачу знаходження максимального потоку в транспортній мережі. Даний алгоритм представляє частковий випадок методу Форда – Фалкерсона із алгоритмічною складністю $O(VE^2)$, E – кількість ребер в графі, а V – кількість вершин[10]. Різниця із звичайним методом Форда – Фалкерсона полягає в тому полягає в тому, що на кожному кроці вибирається найкоротший доповнюючий шлях з s – початкова вершина в t – кінцева вершина, в залишковій мережі (вражаючи, що кожне ребро має одиничну довжину). Найкоротший шлях знаходиться методом пошуку в ширину.

Ще одним алгоритмом за допомогою якого можна знайти максимальний потік в транспортній мережі є алгоритм Дініца[12] .

Так даний алгоритм є удосконаленням алгоритму Едмондса – Карпа. На кожній ітерації, використовуючи пошук в ширину, визначається відстань від точки джерела до всіх вершин в залишковій мережі. Будується граф, що містить тільки такі ребра залишкової мережі, на яких ця відстань збільшується на 1. Виключаються з графу всі тупикові вершини с інцендентними їм ребрами, поки всі вершини не стануть нетупиковими.

Тупиковою називається вершина, крім джерела і стоку, в яку не входить ні одне ребро чи з якою не виходить ні одного ребра.

На отриманому графі шукається найкоротший збільшуваний шлях. Цим шляхом буде будь – який шлях із s в t . Виключається із залишкової мережі ребро з мінімальною пропускною здатністю, знову виключаються тупикові вершини, і алгоритм працює до тих пір поки є збільшуваний шлях.

Тимчасова складність алгоритму складає $O(V^2E)$. В мережах з одиничними пропускними здатностями існує більш потужна оцінка тимчасової складності: $O(EV)$

Можна показати, що кожен раз кількість ребер в блокуючому потоці збільшується хоча б на 1, тому в алгоритмі не більше ніж $n-1$ блокуючих потоків, де n – кількість вершин в мережі. Допоміжна мережа G_L може бути побудована обходом в ширину за час $O(E)$, а блокуючий потік на кожному рівні графу може бути знайдений за час $O(VE)$. Тому час роботи алгоритму Дініца складає $O(V^2E)$.

Використовуючи структури даних, які називаються динамічні дерева, можна знаходити блокуючий потік на кожній фазі за час $O(E \log V)$, тоді час роботи алгоритму Дініца може бути покращено до $O(VE \log V)$ [13].

3.3 Моделювання потоку

Для моделювання потоків машин можна використовувати різні закони розподілу: рівномірний, гаусівський та пуассонівський.

3.3.1 Рівномірний закон розподілу

Кажуть, що випадкова величина має неперервний рівномірний розподіл на відрізку $[a,b]$, де $a, b \in R$, якщо щільність $f_x(x)$ має вигляд[14]:

$$f_x = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

При моделюванні потоку генерується число в межах від 0 до 1. Для зручності число від (0,1) позначимо як α , яке моделює функція rand.

3.3.2 Гаусовий закон розподілу

Нормальний розподіл (розподіл Гаусса) – розподіл ймовірностей випадкової величини, що характеризується густинною ймовірності:

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

де μ - математичне очікування, σ^2 - дисперсія випадкової величини. Параметр σ також відомий, як стандартний відхил. Розподіл із $\mu = 0$ та $\sigma^2 = 1$ називають стандартним нормальним розподілом [14].

3.3.3 Пуасонівський закон розподілу

Процеси в системі обслуговування (надходження автомобілів на вантаження-вивантаження, на заправку паливом, на станцію технічного обслуговування; заявки на таксі; підхід пасажирів на зупинки транспорту та інші) в переважній більшості випадків розподіляються згідно ймовірнісним законам Пуассона. Такі системи

прийнято називати пусонівськими. Аналітичні залежності, що описують зазначені ймовірні закони, в загальному випадку є аналітичними моделями цих процесів.

Потік машин може моделюватись за законом Пуассона. Ймовірність подій, розподілених за цим законом має наступний вигляд:

$$P(m, a) = \frac{a^m}{m!} e^{-a} = \frac{(t\lambda)^m}{m!} e^{-\lambda t}$$

де $P(m, a)$ – ймовірність появи події при заданому значенні параметра a дорівнює m раз; $a = \lambda t$ – математичне очікування (середнє число подій за даний відрізок часу); m – випадкова величина, число подій за даний відрізок часу; t – відрізок часу, за який розглядається поведінка випадкової величини; $e = 2,71 \dots$ – основа натурального логарифма [15].

3.4 Висновки до розділу

Математичною моделлю карти було обрано зважений орієнтований граф. Для задачі пошуку найкоротшого маршруту було обрано алгоритм Дейкстри оскільки в даному випадку ребра графу не будуть приймати від'ємні значення. Для задачі пошуку максимального потоку на графі було обрано метод Форда – Фалкерсона без часткових випадків оскільки ребра графу не будуть мати одиничну довжину. Також було обрано рівномірний закон розподілу для моделювання потоку на ребрах графа. Перевагою даного закону розподілу є те, що для його лише один параметр – це пропускну спроможність дороги в порівнянні з іншими законами розподілу.

4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Структура системи

Програмна реалізація системи виконана на мові програмування Java восьмої версії в програмному середовищі NetBeans IDE 8.1. Для написання програмного інтерфейсу використовувався фреймворк Swing.

Проектування системи виконувався згідно парадигм об'єктно – орієнтованого програмування. Використовувався шаблон проектування MVC – Model View Controller (Модель – Представлення – Контролер). На рисунку 4.1 наведено схема взаємодії даних програми з діями користувача.

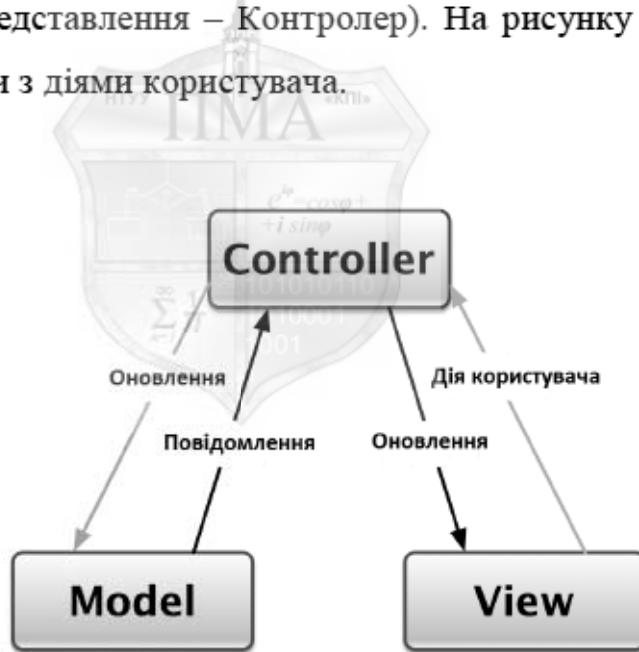


Рисунок 4.1 – Шаблон MVC

Модуль «View» (Представлення) відображає користувачу дані в доступній формі і забезпечує графічний інтерфейс при роботі з програмою. Модуль забезпечує наступні функції:

- відображення результату роботи програми;
- відображення вихідних даних.

- введення користувачем початкової точки маршруту та кінцевої.

Модуль «Model» (Модель) зберігає в собі всю логіку програми. До логіки програми відносяться математичні алгоритми Дейкстри та Форда – Фалкерсона. На даний модуль покладаються наступні функції:

- обчислення найкоротшого шляху за критерієм відстані;
- обчислення найкоротшого шляху за критерієм часу;
- знаходження максимального потоку між початкової та кінцевою вершинами;

Модуль «Controller» (Контролер) керує обміном даними між модулем «Модель» та «View» та забезпечує зв’язок між користувачем і системою: контролює введення даних користувачем і використовує модель і представлення для реалізації необхідної реакції.



4.2 Опис розроблених програмних засобів та структурна схема

Для представлення графу використовувались наступні класи: `Vertice` і `Distation`.

`Vertice`. Об’єкти даного класу описують вершини графу. В даному класі основними атрибутами є точки розміщення вершин на карті – це координати `x` та `y`, і відповідно, це `id` кожної вершини.

`Distation`. Об’єкти даного класу описують кожне ребро графа з його властивостями. Так кожне ребро має атрибут `шлях` – пара суміжних вершин графу, `напрямок` – даний атрибут необхідний для задання руху машини по ребру, `довжина ребра` – числове значення, яке представляє собою модуль вектору, `потік ребра` – числове значення, яке представляє пропускну спроможність для проїзду транспорту по ребру.

Алгоритми на графах представлені в попередньому розділі. Для їх реалізації використовувались класи Deikstra і FordFal.

Deikstra. Представляє собою реалізацію алгоритму Дейкстри та повертає найкоротший шлях між двома вершинами;

FordFal. Представляє собою реалізацію алгоритму Форда – Фалкерсона та повертає максимальний потік між вершинами.

Для реалізації регулювання світлофорами було створено клас Lights. Кожен світлофор може приймати наступні значення у вигляді кольорів: червоний, зелений, жовтий. Даний клас містить наступні атрибути:

- атрибут color приймає кольорів значення світлофору;

- атрибут lastColor – містить останнє значення світлофору. Він необхідни для реалізації перемикання кольорів. Атрибути x,y призначені для розміщення світлофорів на карті;

- атрибут ellipse описує фізичне розташування та розміри світлофору.

Світлофор - овал з наступними параметрами (координата x, координата y, ширина, висота).

Клас Car містить опис основних характеристик машини, яка моделюється на карті разом зі світлофорами. Даний клас має наступні атрибути:

- url. Даний атрибут необхідний для того, щоб можна було зчитати розміщення картинки на комп’ютері, і відповідно завантажити його в програму для відображення на карті;

- carX та carY. Атрибути необхідні для задання початкового розміщення машини на карті. За замовчуванням машину можна розміщувати лише від початкової вершини з якої почнатиметься рух;

- image. Атрибут необхідний для завантаження моделі машини на карту.

На рисунку 4.2 зображена діаграма схема класів описаних вище. Так класи Vertice і Distation об’єднуються в інтерфейс Graph і дозволяють працювати з графом класу Map через об’єкти даних класів.



Рисунок 4.2 – Діаграма класів

4.3 Опис розробки і інтерпретація алгоритмів

Для розробки алгоритмів Дейкстри та Форда – Фалкерсона були використані класи Deikstra та Ford – Falkerson. Блок – схеми даних алгоритмів наведені на рисунках 4.3 та 4.4

Робота алгоритму Дейкстри полягає в пошуку оптимального маршруту за двома критеріями: критерій мінімальної відстані та мінімального часу.

Пошук за критерієм мінімальної відстані підлягає класичній реалізації алгоритму Дейкстри: з файлу з даними графу отримується матриця суміжності та

матриця довжин сусідніх ребер. Після отримання цих даних відбувається пошук шляху за класичним алгоритмом Дейкстри.

При пошуку мінімального часу алгоритм Дейкстри інтерпретується по – іншому. З файлу з даними графу отримується матриця суміжних вершин та матриця з потоками машин кожного ребра. Кожна машина на ребрі, в залежності від потоку, має свою рівномірну швидкість. Відповідно час проїзду через кожне ребро розраховується за формулою $t = s/v$, де t – час проїзду машиною через ребро, s – довжина ребра, v – швидкість машини на ребрі.

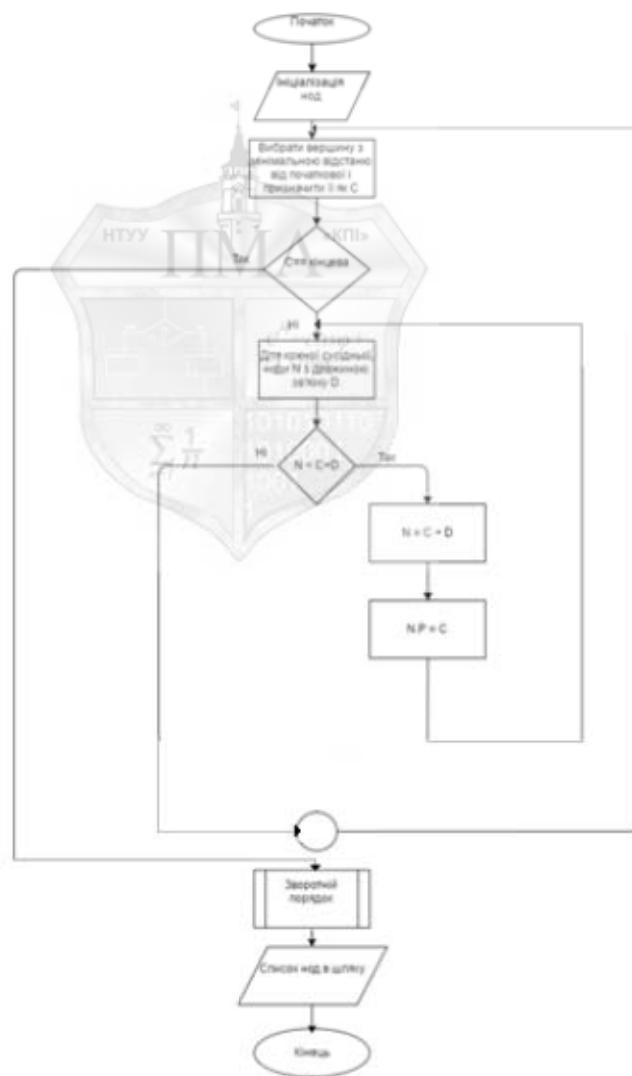


Рисунок 4.3 – Блок - схема алгоритму Дейкстри

Алгоритм Форда – Фалкерсона знаходить максимальний потік між початковою і кінцевою вершинами маршруту. Даний алгоритм необхідний для реалізації задачі моделювання світлофорів за нефіксований час.

На кожному ребрі моделюється потік машин за рівномірним законом розподілу від 0 до a , де a – це пропускна здатність ребра. Після цього спрацьовує алгоритму Форда – Фалкерсона і за допомогою нього буде отримано максимальний потік між вершинами. Прокладається, між початковою та кінцевою вершинами, які задає користувач, оптимальний маршрут або за критерієм мінімальної відстані або за критерієм мінімального часу. На шляху зчитується чи є на ньому світлофори з відеокамерами. Якщо світлофори є, то починається оптимізація роботи світлофора. Зчитуються потоки, які були змодельовані за рівномірним законом та обраховується відсоток завантаженості дороги на перехресті за формулою $p_i = \frac{\max(f_i) - model(f_i)}{\max(f_i)}$, де p_i – відсоткове відношення завантаження на перехресті, $\max(f_i)$ – максимальний потік, $model(f_i)$ – змодельований потік на ребрі за рівномірним законом розподілу.

На перехресті отримано два числа, де перше число означає відсоток завантаженості однієї сторони, а друге означає число означає відсоток завантаженості протилежної сторони. Ці числа порівнюються і в напрямку більшого завантаження починає горіти зелене світло світлофора.

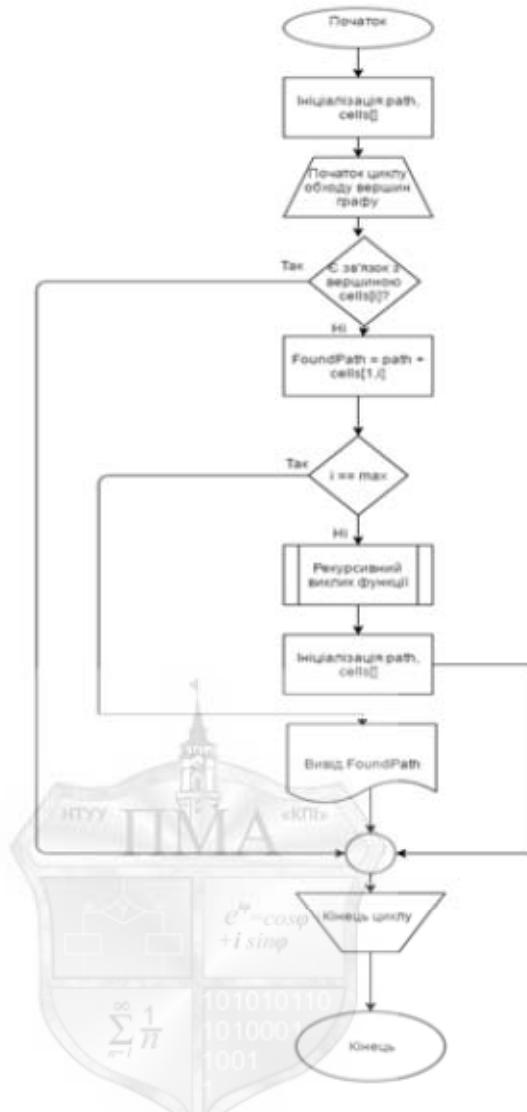


Рисунок 4.4 – Блок - схема алгоритму Форда - Фалкерсона

4.4 Опис графічного інтерфейсу

Графічний інтерфейс програми наведений на рисунках 4.5, 4.6, 4.7. Так на рисунку 4.5 зображена основна форма: відображається лише карта з вершинами. Програма працює в декількох режимах. Користувач може змоделювати декілька варіантів роботи світлофорів: регулювання світлофорів з фіксованим та нефіксованим часом. При роботі з фіксованим часом світлофори запускаються в

випадковий момент часу та кольори кожного світлофора горять фіксований інтервал.

Користувач може вибрати декілька варіантів демонстрації роботи світлофорів з фіксованим часом. Він може вибрати декілька критеріїв для прокладання маршруту від початкової до кінцевої вершини. Користувачу пропонуються наступні критерії зображені на рисунку 4.5:

- критерій мінімальної відстані;
- критерій мінімального часу.

Оптимальний маршрут за цими критеріями знаходить алгоритм Дейкстри.



Рисунок 4.5 – Головна форма програми

На рисунку 4.6 та 4.7 відображаються форми за допомогою яких користувач може вибрати початкову вершину і кінцеву. Після вибраних вершин він натисне кнопку «Прокласти маршрут» після чого йому на екран виведеться форма на рисунку 4.8 із зmodeльованими світлофорами, прокладеним маршрутом та буде відображати машину, яка рухатиметься оптимальним шляхом. Також користувачу на екран виведуться результати роботи алгоритму в числових значеннях зображені на рисунку 4.8. До них відноситься час поїздки користувача на автомобілі в

секундах, відстань, яку він має проїхати в метрах, похибка результатів, та сам маршрут у наступному вигляді «1-2-5..10».

На рисунку 4.9 зображений результат моделювання роботи світлофорів. Користувачу зображується машина на формі, яка починає свій шлях від початкової вершини до кінцевої. Дуги графу, як видно на рисунку, зафарбовані в різні кольори. Так червоний колір означає, що дуга завантажена на 75% потоком від пропускної здатності, жовтий колір означає, що дуга завантажена від 45 до 75 відсотків від пропускної здатності, а зелений колір означає, що дуга завантажена менше ніж на 45 відсотків.

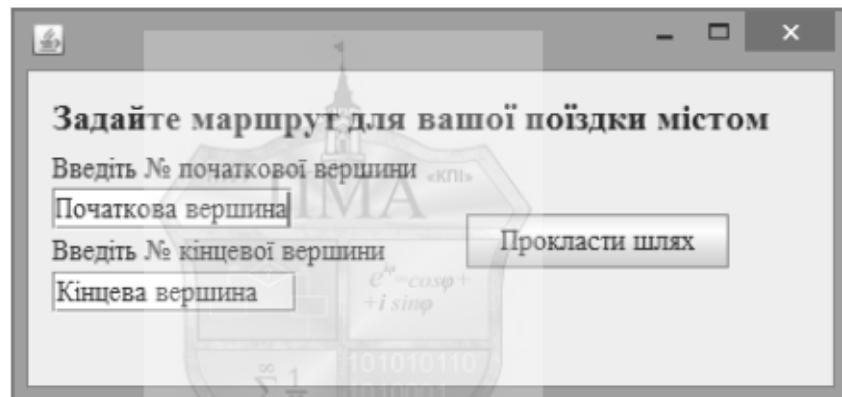


Рисунок 4.6 – Форма «Критерій мінімальної відстані»

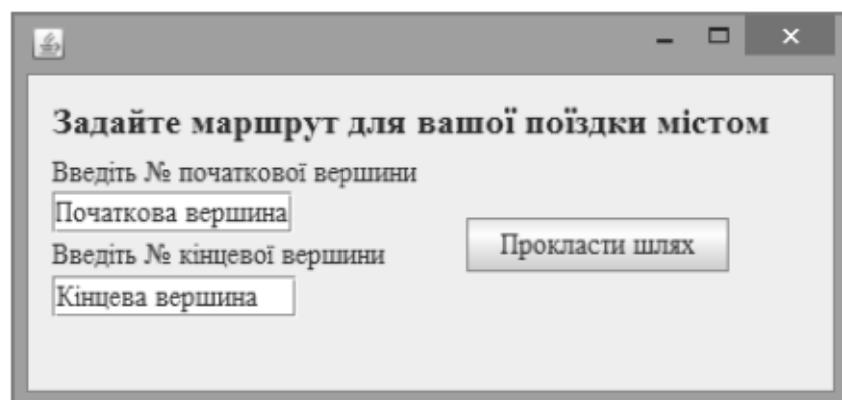


Рисунок 4.7 – Форма «Критерій мінімального часу»

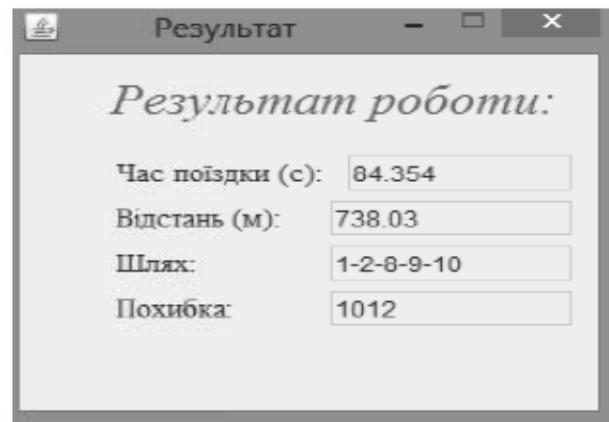


Рисунок 4.8 – Форма «Результат»

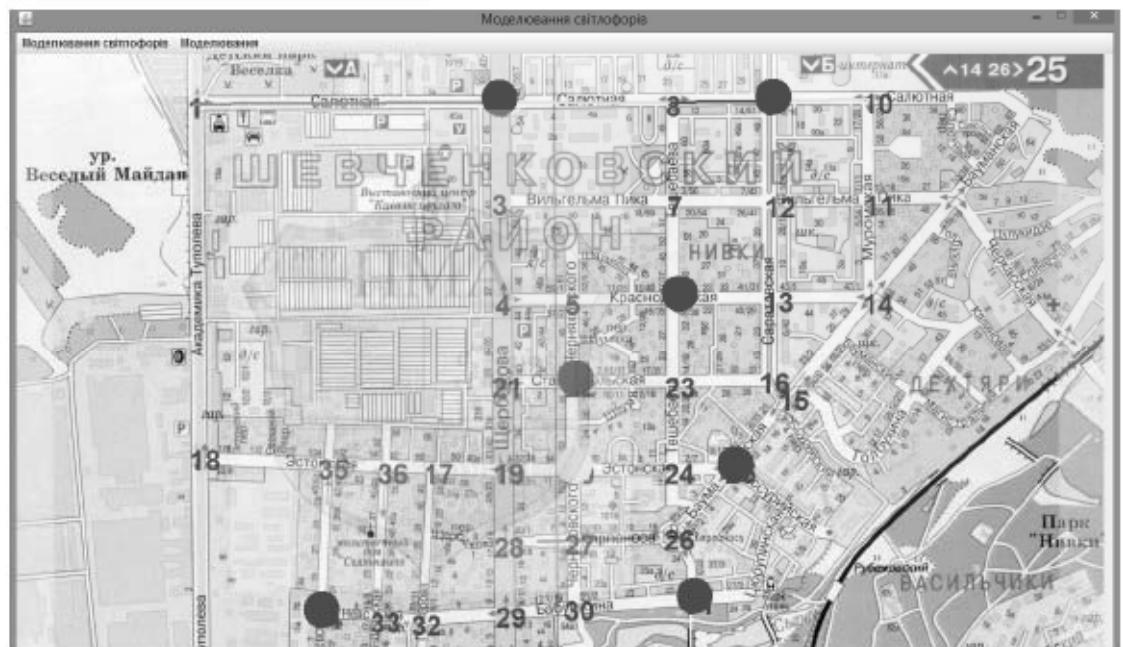


Рисунок 4.9 – Форма «Результат моделювання роботи світлофорів»

4.5 Висновки до розділу

Логіка роботи програмного забезпечення спроектована за допомогою шаблону MVC – Модель – Представлення – Контролер. Було спроектовано автоматизовано систему та наведено структурну схему класів з їх описом. Для роботи з алгоритмами та математичною моделлю та також реалізовані алгоритми Дейкстри та Форда – Фалкерсона з наведеними блок-схемами. Також був спроектований графічний інтерфейс користувача.



5 ВИПРОБУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розроблений програмний комплекс випробуваний з декількома точками. Для наглядного прикладу кожен світлофор було запущено в різний час та його робота виконується за власним фіксованим та нефіксованим часом.

5.1 Контрольні приклади

При запуску програми користувачу відкривається форма, яка зображена на рисунку 5.1. Користувач має вибрати маршрут. Наприклад, від першої точки до 20 – і.



Рисунок 5.1 – Контрольний приклад 1. Вибір маршруту

Натиснувши в меню «Моделювання світлофорів» користувач може вибрати пункт підменю з назвою «Робота з фіксованим часом» і відповідно до цього обрати, що йому важливіше доїхати в кінцеву точку за найменший час, чи за найкоротшу відстань.



Рисунок 5.2 - Контрольний приклад 1. Вибір критерію маршруту

Спочатку нехай користувач обере найкоротший маршрут за критерієм мінімальної відстані і введе вершини від яких буде прокладений маршрут як зображеного на рисунку 5.3.

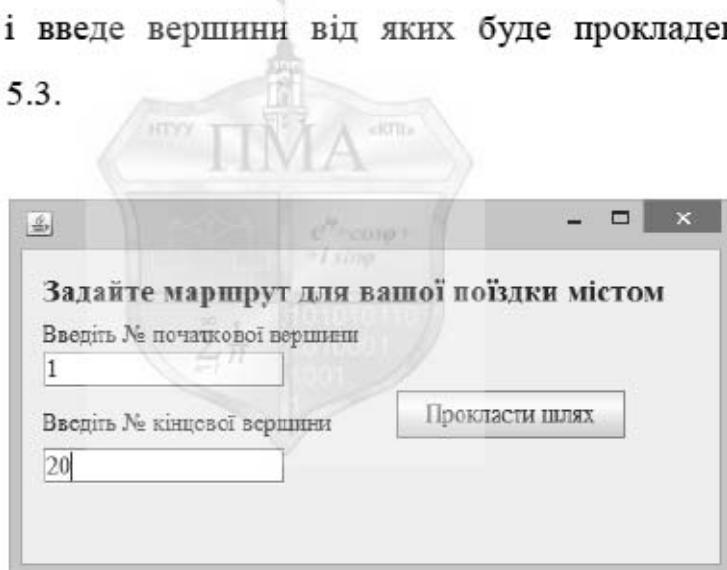


Рисунок 5.3 – Контрольний приклад №1. Задання вершин.

Користувач в формах введення даних не зможе ввести невалідні дані. До таких даних відносяться: букви, не існуючі точки на карті.

Після натиснення кнопки «Прокласти маршрут» буде розпочато моделювання світлофорів та відриється форма як на рисунку 5.4. Машина починає рух з вершини 1 до вершини 20. Також на рисунку 5.5 користувач надається інформація щодо поїздки автомобіля – це час поїздки, відстань між введеними точками, шлях та похибка обчислень.



Рисунок 5.4 – Контрольний приклад №1. Рух машини по прокладеному шляху

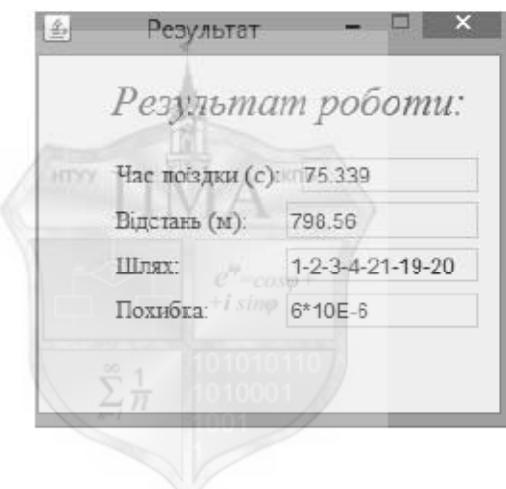


Рисунок 5.5 – Контрольний приклад №1. Числові результатати

Для контрольного прикладу №2 користувач також може прокласти маршрут від 1-ї вершини до 20-ї за критерієм мінімального часу як зображено на рисунку 5.6

Пошук шляху за критерієм часу

Введіть № початкової вершини

Введіть № кінцевої вершини

Рисунок 5.6 Контрольний приклад №2. Введення вершин

Після введених даних для користувача відкриється форма з результатом моделювання як зображене на рисунку 5.7 та числовий результат на рисунку 5.8.



Рисунок 5.7 – Контрольний приклад №2. Результат моделювання

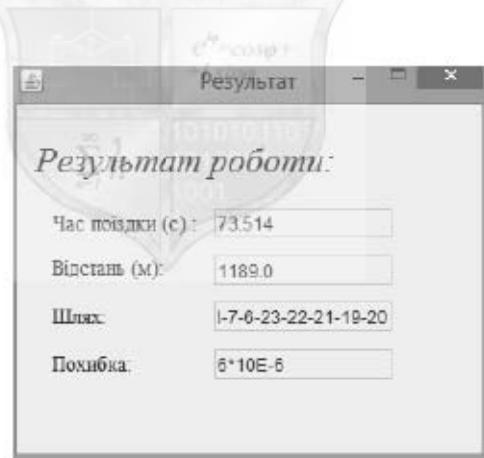


Рисунок 5.8 – Контрольний приклад №2. Числовий результат

Для контрольного прикладу №3 світлофори будуть перемикатись за нефіксований час. Користувач прокладає маршрут від 1-ї точки до 20-ї за критерієм часу після чого відбувається процес моделювання проїзду (рисунок 5.9). Під час руху автомобіля відбувається оптимізація роботи світлофорів, які розташовані в точках 22 та 20. Числові результати оптимізації представлені на рисунку 5.10.



Рисунок 5.9 – Контрольний приклад №3. Результат моделювання

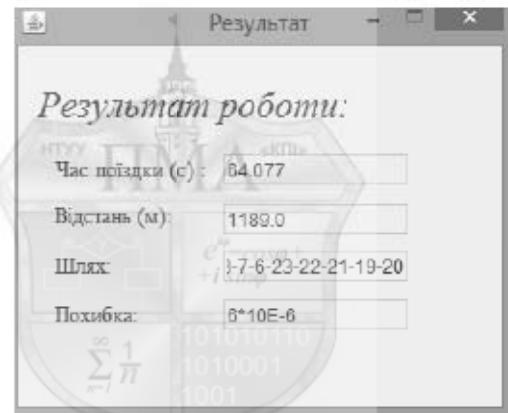


Рисунок 5.10 – Контрольний приклад №3. Числовий результат

5.2 Аналіз отриманих результатів

В таблиці 5.1 містяться результати роботи контрольних прикладів.

Таблиця 5.1 – Результати роботи контрольний прикладів

Робота світлофорів	Фіксований час		Нефіксований час
Критерій	Мінімальної відстані	Мінімального часу	Мінімального часу
Час поїздки(с)	75.339	73.514	64.077
Відстань(м)	798.56	1189.0	1189.0
Маршрут	1-2-3-4-21-19-20	1-2-3-7-6-23-22-21-19-20	1-2-3-7-6-23-22-21-19-20
Похибка	$6 \cdot 10^{-6}$	$6 \cdot 10^{-6}$	$6 \cdot 10^{-6}$

За цими результатами оптимізація роботи світлофорів дозволяє зменшити час поїздки на 11 секунд в порівнянні з критерієм мінімальної відстані та 8.5 секунд в порівнянні з критерієм мінімального часу. Похибка комп'ютерних обчислень у всіх результатах складає $6 * 10^{-6}$.

5.3 Висновки до розділу

Результати випробування показали, що розроблений програмне забезпечення має високу точність та коректно відображає результати роботи програми в залежності від задачі. Були випробувані контрольні приклади №1, №2, №3. В результаті за допомогою оптимізації світлофорів час поїздки з точки 1 в точку 20 складає 64 секунди, що в середньому на 10 секунд менше ніж при роботі світлофора з фіксованим часом.

ВИСНОВКИ

У дипломній роботі проведено огляд існуючих методів розв'язання задачі та вибрані методи моделювання роботи світлофора та методи оптимізації руху транспорту на перехрестях. Для вирішення задачі пошуку оптимального маршруту було обрано алгоритм Дейкстри, а для задачі пошуку максимального потоку було обрано алгоритм Форда – Фалкерсона. Для моделювання потоку на ребрах використовувався рівномірний закон розподілу.

Був спроектований та реалізований програмний комплекс для моделювання руху транспорту на перехрестях для оптимізації процесу регулювання, який дозволяє зменшити час поїздки транспорту при інтенсивному потоці за рахунок оптимізації роботи світлофорів на перехрестях.

Реалізація програмного комплексу та її алгоритмічної частини були розроблені за допомогою мови програмування Java та фреймворк Swing.

Результати випробування показали, що розроблений програмний комплекс має високу точність та коректно відображає результати роботи програми в залежності від задачі. Були випробувані контрольні приклади №1, №2, №3. В результаті за допомогою оптимізації світлофорів час поїздки з точки 1 в точку 20 складає 64 секунди, що в середньому на 10 секунд менше ніж при роботі світлофора з фіксованим часом.

Основні ідеї та результати дипломної роботи були представлені на науковій конференції «Прикладна математика та комп’ютинг», квітень 2016 рік[16].

ПЕРЕЛІК ПОСИЛАНЬ

1. Питер Ньюман Устойчивость и города: преодоление от автомобильной зависимости / Питер Ньюман, Джейфри Кенворті — Island Press, Вашингтон, 1999 — 305 с.
2. Анна Нагурней Сети экологически чистого транспорта / Анна Нагурней — Island Press, Англия, 2000 — 283 с.
3. Офіційна документація Google Maps. Створення маршруту та планування поїздок [Електронний ресурс] Режим доступу:
<https://support.google.com/maps/#topic=3292869>
4. Офіційна документація Яндекс.Карти. API для разработчиков [Електронний ресурс] Режим доступу: <https://tech.yandex.ru/maps/>
5. Таран Т.А. Основы дискретной математики / Таран Т.А. — Киев: «Просвіта», 2003. — 288 с.
6. Уилсон Р. Введение в теорию графов / Уилсон Р. — М.: Мир, 1977. — 208с.
7. Петров В.В. Автоматизированные системы управления дорожным движением в городах / В. В. Петров — Омск: Изд-во СибАДИ, 2007. — 104с.
8. Алгоритмы: построение и анализ / [Кормен Т., Томас Х. и др.]. — М. : ООО «И. Д. Вильямс», 2013. — 1328 с.
9. Гасс С. Линейное программирование / Гасс С. — М.: Физико-математическая литература, 1961. — 300 с.
10. Yefim Dinitz Dinitz' Algorithm: The Original Version and Even's Version / Yefim Dinitz — Springer, 2006. — Р. 218–240. — ISBN 978-3540328803.
11. Пряха Б. Г. Про точність геодезичних вимірювань / Пряха Б. Г., Білецький Я. В. — Л.: Видавництво Національного університету "Львівська політехніка", 2003. — С. 43-49.
12. Кингман Дж. Пуассоновские процессы / Кингман Дж. — М.: МЦНМО, 2007. — 136 с

13. Вентцель Е. С. Теория вероятностей / Вентцель Е. С. — М.: «Академия», 2005. — 576 с.
14. Справочник по теории вероятностей и математической статистике / [Королюк В.С., Портенко Н.И., Скороход А.В., Турбин А.Ф.] — М.: Наука, 1985. — 640 с.
15. Введение в математическое моделирование транспортных потоков / [Гасников А.В., Кленов С.Л., Нурминский Е.А., Холодов Я.А. и др.] — М.: МФТИ, 2010. — 362 с. — ISBN 978-5-7417-0334-2
16. Прикладна математика та комп'ютинг. ПМК, 2016 : восьма наук. конф. магістрантів та аспірантів, Київ, 20—22 квіт. 2016 р. : зб. тез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2016. — 295—300 с.

