

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ХМЕЛІВСЬКИЙ ЮРІЙ СЕРГІЙОВИЧ

Допускається до захисту:
завідувач кафедри
інформаційних технологій,
д-р техн. наук, доцент
_____ Тетяна НЕСКОРОДЕВА
« ____ » _____ 2022р.

**ДОСЛІДЖЕННЯ ТА ОПТИМІЗАЦІЯ ПРОЦЕСУ ТРАНСПОРТНИХ
ПЕРЕВЕЗЕНЬ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (магістерська) робота

Науковий керівник:

Римар П. В., старший викладач
кафедри інформаційних технологій

Науковий консультант:

Нескородева Т.В.,
д-р техн. наук, доцент

(підпис)

Оцінка: ____ / ____ / ____
(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2022

АНОТАЦІЯ

Хмелівський Ю. С. Дослідження та оптимізація процесу транспортних перевезень. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні технології обробки даних (Data Science)». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

У кваліфікаційній (магістерській) роботі досліджено методи оптимізації транспортної задачі, а також запропоновано інший метод побудови опорного плану транспортної задачі, вдосконалення математичної моделі класичної транспортної задачі, з урахуванням стану дороги і витратами на паливе та розроблено прикладне ПЗ для вирішення поставленої задачі. Робота містить 13 рисунків, 16 таблиць та 40 джерел. Загальний обсяг роботи становить 51 сторінок.

У вступі наведено актуальність теми дослідження. Сформульована мета роботи, визначено об'єкт, предмет та задачі дослідження.

У першому розділі розглянуто постановку задачі роботи, а також проведений огляд існуючих програмних продуктів даної тематики.

Другий розділ присвячений огляду існуючих методів оптимізації транспортних перевезень.

Третій розділ присвячений вибору інструментів та технологій, які використовуються для розробки програмного продукту.

У четвертому розділі мова йде про огляд запропонованого методу оптимізації транспортної задачі та розробка програмного продукту.

У висновку підбитий підсумок виконаної роботи, а також проведений аналіз доцільності запропонованого методу та програмного продукту.

Ключові слова: транспортна задача, транспортні перевезення, оптимізація.

ABSTRACT

Khmelivskiy Yu. S. Research and optimization of the transport process. Specialty 122 "Computer science", educational program "Computer technologies of data processing (Data Science)". Vasyl Stus Donetsk National University, Vinnytsia, 2022.

In the qualification (master's) thesis, the methods of optimizing the transport problem were investigated, as well as another method of building the reference plan of the transport problem, improving the mathematical model of the classical transport problem, taking into account the condition of the road and fuel costs, and developing applied software to solve the problem was proposed. The work contains 13 figures, 16 tables and 40 sources. The total volume of work is 51 pages.

The relevance of the research topic is given in the introduction. The goal of the work is formulated, the object, subject and tasks of the research are defined.

In the first section, the setting of the work task was considered, as well as an overview of existing software products of this topic.

The second section is dedicated to the review of existing methods of optimizing transportation.

The third section is devoted to the selection of tools and technologies used for the development of a software product.

The fourth chapter deals with the review of the proposed method of optimizing the transport problem and the development of the software product.

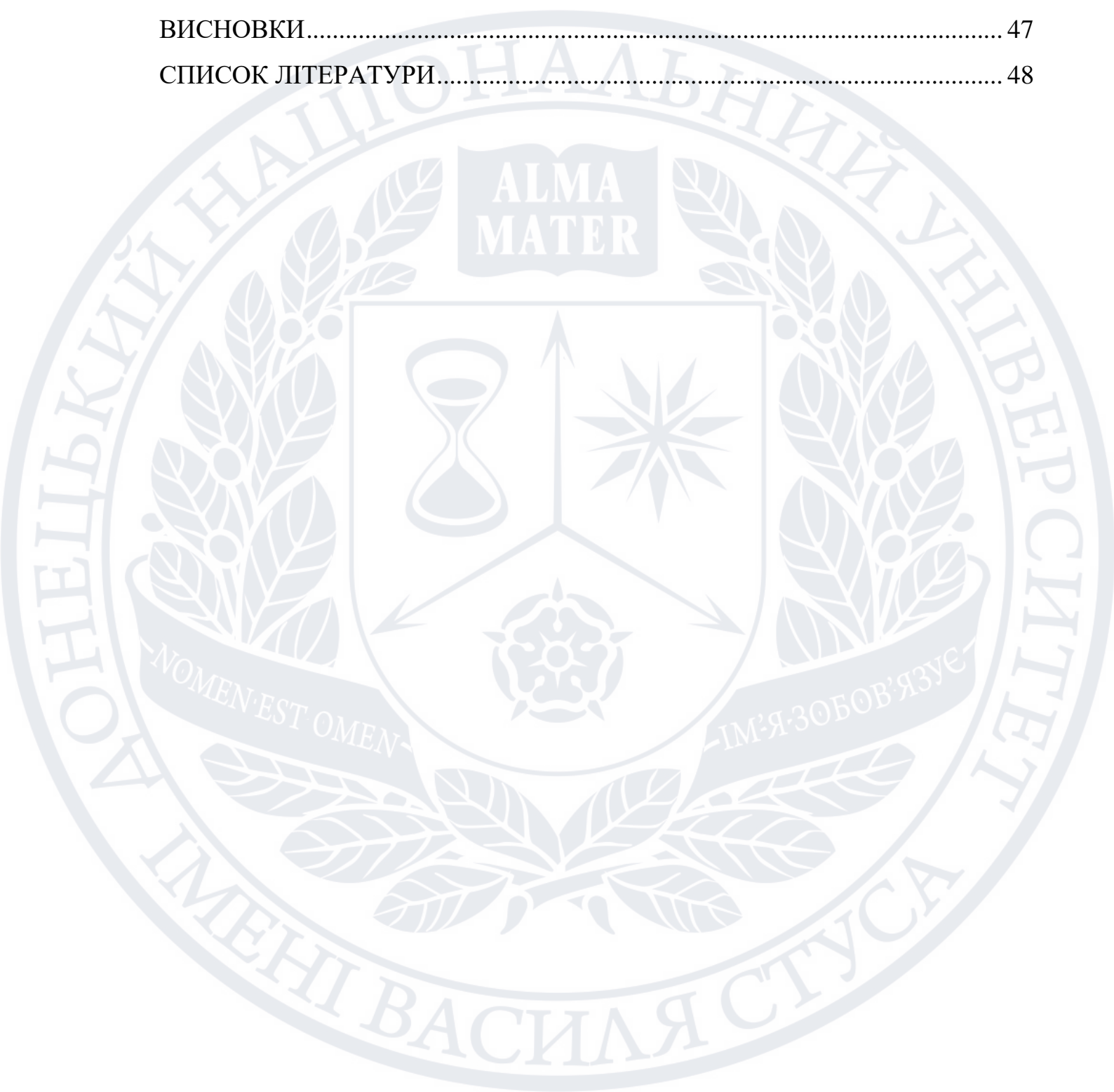
The conclusion sums up the work performed, as well as an analysis of the feasibility of the proposed method and software product.

Keywords: transport task, transport, optimization.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ І ОГЛЯД АНАЛОГІВ ІСНУЮЧИХ ДОДАТКІВ ЗА ДАНОЮ ТЕМАТИКОЮ.....	7
1.1 Опис актуальності дослідження методів оптимізації транспортних перевезень	7
1.2 Постановка задачі	8
1.3 Огляд аналогів.....	12
Висновок до розділу 1	13
РОЗДІЛ 2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ ПЕРЕВЕЗЕНЬ.....	14
2.1 Аналіз існуючих методів оптимізації транспортних перевезень.....	14
2.2.1 Модель транспортної задачі	14
2.2.2 Аналіз методу північно-західного кута	16
2.2.3 Аналіз методу мінімальної вартості	20
2.2.4 Аналіз методу потенціалів	24
Висновок до розділу 2	25
РОЗДІЛ 3 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ	26
3.1 Androin Studio.....	26
3.2 Flutter SDK	28
3.3 Мова програмування Dart	28
3.4 Кросплатформність.....	35
Висновок до розділу 3	37
РОЗДІЛ 4 ОГЛЯД ЗАПРОПОНОВАНОГО МЕТОДУ ОПТИМІЗАЦІЇ ТРАНСПОРТНОГО ПЕРЕВЕЗЕННЯ ТА СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ	38
4.1 Опис запропонованого методу оптимізації транспортної задачі.....	38
4.2 Опис розробленої програми.....	41

4.3 Аналіз практичного використання вдосконаленої моделі транспортної задачі	45
Висновок до розділу 4	46
ВИСНОВКИ.....	47
СПИСОК ЛІТЕРАТУРИ.....	48



ВСТУП

Дана тема є актуальною адже, на сьогоднішній день транспортні перевезення є найбільш перспективним напрямком в діяльності кожного підприємства і є невіддільною частиною економічного розвитку країни. Для сучасного бізнесу є важливим налагодження всіх процесів, від вибору постачальників до доставки товару клієнту. При цьому найважливішою і дуже витратною областю є логістика транспортних перевезень. Вона охоплює собою область від доставки матеріалів на виробництво до розвезення готового товару кінцевим покупцям.

Об'єктом дослідження в рамках кваліфікаційної магістерської роботи є процес оптимізації транспортних перевезень на етапі побудови маршрутів перевезень чи їх корегуванні.

Предметом дослідження є методи оптимізації транспортних перевезень, які використовуються в процесі побудови маршрутів перевезення, та знаходяться в математичних апаратах інформаційних систем.

Метою роботи є дослідження існуючих методів оптимізації транспортних перевезень і вдосконалення математичної моделі транспортної задачі, яка має бути більш ефективною при використанні разом з існуючими методами оптимізації, шляхом введення нових обмежень на якість покриття доріг та інтенсивності транспортного потоку на цих дорогах.

Задачі дослідження полягають у:

- огляді актуальності проблеми транспортних перевезень;
- аналізі існуючих методів оптимізації транспортних перевезень;
- побудові вдосконаленої моделі транспортної задачі;
- розробці програмного продукту для оптимізації транспортних перевезень.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ І ОГЛЯД АНАЛОГІВ ІСНУЮЧИХ ДОДАТКІВ ЗА ДАНОЮ ТЕМАТИКОЮ

1.1 Опис актуальності дослідження методів оптимізації транспортних перевезень

Розвиток інфраструктури та євроінтеграція є вагомими та незмінними пріоритетами Міністерства інфраструктури України, які один без одного неможливі. Тому утворена ефективна платформа для співпраці на регіональному рівні в рамках Транспортної панелі Східного партнерства, основною ціллю якої є сприяння у покращенні транспортних зв'язків між Європейським Союзом та його найближчими сусідами. Транспортні мережі та послуги відіграють ключову роль у покращенні якості життя громадян країни та у зростанні можливостей розвитку промисловості. Тому, транспорт є однією з ключових сфер співпраці між Європою та Україною.

Сучасна економічна ситуація, зростання споживчих ринків винесли на поверхню проблеми з логістикою. Нерозвиненість інфраструктури, яка полягає в практично повній відсутності сучасних складських приміщень, орієнтованих на бізнес, як в великих містах, так і в регіонах, і ще важливіше – в нестачі відповідного транспортного парку, як залізничного, так і автомобільного, а також неякісні автодороги, – всі ці проблеми потрібно вирішувати, використовуючи у тому числі з використанням економіко-математичних методів та моделей. Ефективність транспортування, як відомо, визначається трьома факторами: витратами, швидкістю і безперебійністю. Всі ці критерії в автомобільних, залізничних перевезеннях є дуже не стабільні, тому оптимізація перевезень є досить актуальною проблемою.[1]

Логістика – наука, яка вивчає оптимальне управління матеріальними та іншими супутніми потоками в будь-якій галузі людської діяльності. Логістика, в даний час, одна з найбільш актуальних наук і управлінських практик. І не тільки в економіці, але і в інших областях (містобудуванні, інженерії, політиці,

екології та ін.). Логістика займається питаннями оптимізації потоків в роботі системи. Під потоками мається на увазі як і в звичному нам розумінні слова (нафтопроводи, газопроводи, інші комунікації), так і транспортні потоки, інформаційні потоки в мережах та потоки ресурсів на виробництві [4].

Транспортна задача – це задача про оптимальний план перевезень продукту з пунктів наявності в пункти споживання. Її задачею є доставка продукції в певний час і місце при мінімальних сукупних витратах, серед яких трудові, матеріальні і фінансові ресурси. Вона вважається досягнутою, якщо потрібний товар певної якості і в певній кількості доставляється в потрібні час та місце з мінімальними витратами.

Наприклад, якщо транспортна компанія може вчасно бути поінформована про затори на дорозі, їй буде легше заздалегідь скорегувати маршрут своїх машин. Інформованість, економія, розрахунок руху по маршруту і інші технології оптимізації дозволяють доставити вантаж клієнта швидко, вчасно і з максимальною збереженням вантажу.

Оптимізація автотранспортних перевезень – це використання методів і технологій, за допомогою яких можна максимально точно розрахувати час управління маршрутами та витратами, пов'язаними з перевезеннями.

1.2 Постановка задачі

Об'єктом дослідження в рамках кваліфікаційної магістерської роботи є процес оптимізації транспортних перевезень на етапі побудови маршрутів перевезень чи їх корегуванні. Предметом дослідження являються методи оптимізації транспортних перевезень, які використовуються в процесі побудови маршрутів перевезення, та знаходяться в математичних апаратах інформаційних систем. Метою даної роботи є дослідження методів побудови опорного плану перевезень та вдосконалення математичної моделі транспортної задачі, яка має бути більш ефективною при використанні разом з існуючими методами оптимізації, шляхом введення нових обмежень на якість покриття доріг та інтенсивності транспортного потоку на цих дорогах.

Для досягнення мети, необхідно досліджувати наступні питання: – проаналізувати існуючі методи оптимізації транспортних перевезень; – обґрунтувати мету вдосконалення математичної моделі транспортної задачі та сформулювати вимоги до неї; – сформулювати критерії оптимальності до методів оптимізації перевезень; – розробити вдосконалену модель транспортної задачі з урахуванням різноманітних факторів при побудові маршрутів перевезень; – запропонувати метод оптимізації транспортних перевезень, який зменшить число ітерацій для приведення до оптимального плану; – застосувати вдосконалену модель транспортної задачі разом з запропонованим методом оптимізації перевезень на практиці для доведення його переваг над класичною моделлю.

1.2.1 Обґрунтування мети та формування вимог до удосконалення математичної моделі транспортної задачі

Правильно організована система перевезень дає змогу скоротити витрати на утримання автопарку, а також привести до мінімуму ризики псування вантажу та затримок. Саме для цього необхідно постійно оптимізувати систему транспортної логістики, в результаті зменшення витрат без втрати якості перевезення.

Оптимізація транспорту необхідна при неконтрольованому зростанні витрат, а саме:

- витрат на навантаження і розвантаження;
- витрат на доставку в кінцевий та проміжні пункти;
- витрат на паливо і витрат на експлуатацію ТЗ;
- оплати праці водіям, механікам і ін.

Таким чином, при зростанні цих показників при збереженні обсягу і швидкості перевезення необхідно проаналізувати поточну логістичну стратегію і визначити причини і можливі ризики збільшення витрат. Як правило, аналізу піддаються наступні етапи:

- спосіб переміщення вантажу і вибір транспортного засобу;
- маршрути перевезень;
- схема розташування складів компанії;
- вибір логістичних посередників (при необхідності).

1.2.2 Мета вдосконалення математичної моделі

У класичній транспортній задачі маршрут доставки вантажу оцінюється за пройденим транспортним засобом шляху, але при цьому не враховується безліч інших факторів, що впливають на час доставки. Отже, метою даної роботи є вдосконалення математичної моделі транспортної задачі, що дозволяє оптимізувати організацію доставки вантажу з точки зору скорочення затрат на перевезення з урахуванням якості дороги, інтенсивності транспортного потоку і погодних умов.

1.2.3 Вимоги до математичної моделі

Спираючись на результати аналізу існуючих математичних методів та математичну модель класичної транспортної задачі, вимоги до удосконалення наступні:

- вдосконалення математичної моделі класичної транспортної задачі з урахуванням різноманітних факторів, такі як якість покриття та завантаженість доріг, які безпосередньо впливають на процес доставки вантажа до пунктів призначення, і таким чином скоротити витрати на вантажні перевезення;
- застосування вдосконаленої моделі та побудова маршрутів існуючими методами повинна давати результат, який має бути оптимальним з точки зору затрат на перевезення;
- використання методу не повинно займати багато часу.

1.2.4 Формування критеріїв оптимальності до вдосконаленої математичної моделі транспортної задачі

Задля того, щоб мати можливість оцінити та дослідити результати дослідження, а саме застосування удосконаленої математичної моделі з використанням існуючих методів оптимізації вантажних перевезень, були сформовані наступні критерії ефективності, які зображені в таблиці 1.1.

Таблиця 1.1 – Критерії ефективності вдосконаленої математичної моделі транспортної задачі

Критерії	Спосіб обчислення
Вартість вантажного перевезення (для скорегованого маршруту)	$C_{ij} \rightarrow \min$ або $C_{ij}(\text{після корегування}) - C_{ij} \leq 10\%$ де i – постачальник; j – споживач
Мінімум транспортної роботи в тонно-кілометрах	$Z_{\text{після корегування}} = \sum \sum C_{ij} * x_{ij} \rightarrow \min$ або $Z_{\text{після корегування}} - Z \leq 10\%$ де c_{ij} – вартість перевезення одиниці продукції від i -го постачальника до j -го споживача; x_{ij} – кількість продукції, що перевозиться від i -го постачальника до j -го споживача; a_i – запаси продукції i -го постачальника; b_j – попит на продукцію j -го споживача.

Мінімум транспортної роботи в тонно-кілометрах та вартість вантажного перевезення є основними критеріями оптимальності в транспортній задачі. Окрім цього, бувають випадки, коли неможливо перебудувати маршрут більш оптимально, а навпаки – з додатковими витратами, тому що немає інших альтернатив. У цьому випадку, до перелічених вище критеріїв оптимальності слід додати також те, що

перерахована вартість доставки на i -м шляху не повинна перевищувати 10 відсотків від початкової вартості.

1.3 Огляд аналогів

В мережі є велика кількість подібних між собою онлайн калькуляторів для вирішення транспортної задачі. Однак усі вони призначені для теоретичного обрахунку задачі та мають покрокове пояснення ходу розв'язання. А також мають досить застарілий і незручний дизайн. [2]

Math Вища Математика Розв'язання задач та прикладів - OnLine
 Головна | Рішення транспортної задачі | КРОК-1 | КРОК-2 >

Введіть кількість одиниць продукції на складах та потреби у продукції за споживачами:

Склад №	Запас од. продукції	Споживач №	Потреба од. продукції
1	<input type="text"/>	1	<input type="text"/>
2	<input type="text"/>	2	<input type="text"/>
3	<input type="text"/>	3	<input type="text"/>
4	<input type="text"/>		

Заповніть таблицю витрат за перевезення одиниці виробленої продукції зі складу i споживачеві j :

Склад №	Споживачі		
	1	2	3
1	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>

Метод знаходження початкового рішення:

Рисунок 1.1 - Сторінка вводу даних

Завдання:

Є 2 складу, що містять кілька одиниць однотипної продукції (див. таблицю 1), є також 2 споживача, які потребують певної кількості даної продукції (див. таблицю 2). При перевезенні однієї одиниці виробленої продукції зі складу i споживачеві j виникають витрати P_{ij} . Величини витрат наведено у таблиці 3. При перевезенні K одиниць продукції зі складу i споживачеві j сумарні витрати на перевезення становлять $K * P_{ij}$.

Потрібно знайти такий план перевезень, при якому загальні витрати на перевезення всієї продукції, за всіма споживачами, будуть мінімальними.

Таблиця 1

Склад №	Запас од. продукції
1	15
2	25

Таблиця 2

Споживач №	Потреба од. продукції
1	10
2	30

Таблиця 3

Витрати перевезення одиниці виробленої продукції зі складу i споживачеві j

Склад №	Споживачі	
	1	2
1	10	2
2	12	7

Рисунок 1.2 - Опис рішення задачі

Крок:1
Перевірка на збалансованість

Загальна кількість запасів на складах: 40 ; Загальна потреба: 40
Завдання є [збалансоване](#) ([закрито](#)).

Крок: 2
Знаходження початкового рішення. Метод мінімального елемента

Запишемо це завдання у вигляді транспортної таблиці. У верхньому рядку перерахуємо потреби споживачів по порядку номерів. У лівому стовпці перерахуємо наявні запаси на складах. На перетині j -го стовпця та i -го рядка будемо записувати кількість продукції, що поставляється з i -го складу j -го споживача. Поки початкового рішення не знайдено, залишимо ці клітини порожніми. У правому нижньому кутку кожної клітини зеленим кольором відобразимо відповідні витрати.

	$b_1=10$	$b_2=30$	
$a_1=15$			
$a_2=25$			

Введемо допоміжні рядок і стовпець, в яких будемо відзначати нерозподілені запаси, що залишилися, і відповідні потреби (залишки). Спочатку їх зміст дорівнює вихідним запасам і потребам, оскільки ще нічого не розподілялося. На малюнку представлені жовтим кольором.

Виберемо клітину в яку розподілятимемо продукцію на наступній ітерації, це клітина містить мінімальне значення витрат (мінімальний елемент). На малюнку як сама клітина, так і відповідні їй залишки відображаються червоним шрифтом.

	$b_1=10$	$b_2=30$	
$a_1=15$		X ₂	15
$a_2=25$			25
	10	30	

Рисунок 1.3 - Опис рішення задачі

a_2	u_2	v_2	w_2
	$v_1=5$	$v_2=0$	

Порядок обчислення потенціалів був наступний:

- 1) Нехай $v_2=0$;
- 2) $u_1 = P_{1,2} - v_2$;
- 3) $u_2 = P_{2,2} - v_2$;
- 4) $v_1 = P_{2,1} - u_2$;

Тепер для всіх вільних клітин робочої матриці витрат обчислимо оцінки S_{ij} за формулою $S_{ij} = P_{ij} - u_i - v_j$ (зелений колір). Кожна така оцінка показує, на скільки зміняться загальні транспортні витрати при завантаженні даної клітини одиницею вантажу. Таким чином, якщо серед оцінок є негативні (витрати зменшуються), то цей план можна поліпшити перемістивши у відповідну клітину деяку кількість продукції. Якщо ж серед оцінок немає негативних – план є оптимальним.

Робоча матриця витрат із заповненими оцінками клітинками представлена нижче.

	b_1	b_2	
a_1	3	2	$u_1=2$
a_2	12	7	$u_2=7$
	$v_1=5$	$v_2=0$	

У наведеної таблиці немає негативних оцінок (план поліпшити не можна), отже досягнуто оптимальне рішення.

	$b_1=10$	$b_2=30$
$a_1=15$		15 2
$a_2=25$	10 12	15 7

Загальні витрати на перевезення всієї продукції для оптимального плану становлять:

$$P_{\text{opt}} = 255$$

Рисунок 1.4 - Опис рішення задачі

Висновок до розділу 1

У даному розділі було розглянуто актуальність теми дослідження методів оптимізації транспортних перевезень та постановку задачі роботи, а також проведений огляд існуючих програмних продуктів даної тематики, були окреслені їх переваги та недоліки. Наступний розділ буде присвячений аналізу існуючих методів вирішення транспортної задачі.

РОЗДІЛ 2

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ ПЕРЕВЕЗЕНЬ

2.1 Аналіз існуючих методів оптимізації транспортних перевезень

У сучасному логістичному менеджменті розроблена методологічна база, яка дозволяє знизити витрати на транспортування вантажів і пасажирів. Рішення в межах даних методів приймаються на основі математичного моделювання. Це можуть бути евристичні моделі або алгоритми для комп'ютерних розрахунків.

Вирішення транспортної задачі починається з пошуку допустимого початкового рішення (опорний план перевезень), щоб всі запаси постачальників були розподілені по споживачам. Допустиме початкове рішення не обов'язково виявляється оптимальним, а метод його знаходження може бути як простим (метод північно-західного кута або аналоги) або більш складним і наближеним до оптимального рішення (мінімальних тарифів, метод Фогеля).

В даній роботі будуть розглянуті та проаналізовані математична модель класичної транспортної та наступні методи оптимізації транспортних перевезень (найбільш поширені та актуальні): метод північно-західного кута, метод мінімальної вартості (мінімальних тарифів) та метод потенціалів для визначення оптимального плану перевезення.

2.2.1 Модель транспортної задачі

Транспортна задача – це специфічна задача лінійного програмування, яка застосовується для визначення найбільш економічного плану перевезення однорідної продукції від постачальників до споживачів [3]. Математична модель транспортної задачі має такий вигляд:

$$Z = \sum_{i=1}^m \sum_{j=1}^n C_{ij} * x_{ij} \rightarrow \min$$

за обмежень:

$$\sum_{j=1}^n x_{ij} = a_i \quad (i = \overline{1, m});$$

$$\sum_{i=1}^m x_{ij} = b_j \quad (j = \overline{1, n});$$

$$x_{ij} \geq 0 \quad (i = \overline{1, m}; j = \overline{1, n}),$$

де x_{ij} – кількість продукції, що перевозиться від i -го постачальника до j -го споживача;

c_{ij} – вартість перевезення одиниці продукції від i -го постачальника до j -го споживача;

a_i – запаси продукції i -го постачальника;

b_j – попит на продукцію j -го споживача.

Транспорту задачу називають збалансованою, або закритою, якщо в ній загальна кількість продукції постачальників дорівнює загальному попиту всіх споживачів, тобто

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

Інакше транспортну задачу називають незбалансованою, або відкритою.

Планом транспортної задачі називають будь-який невід’ємний розв’язок системи обмежень транспортної задачі, який позначають матрицею

$$x = (x_{ij}) \quad (i = \overline{1, m}; j = \overline{1, n}),$$

Оптимальним планом транспортної задачі називають матрицю:

$$X^* = (x_{ij}^*) \quad (i = \overline{1, m}; j = \overline{1, n}),$$

яка задовольняє умови задачі і для якої цільова функція набуває найменшого значення. Початкові дані транспортної задачі записуються у вигляді таблиці, яка наведена на рисунку 1.1.

Таблиця 2.1 – Вигляд початкових даних транспортної задачі

Пункти відправлення	Пункти призначення					Запаси
	B_1	...	B_j	...	B_n	
A_1	C_{11}	...	C_{1j}	...	C_{1n}	a_1
	x_{11}	...	x_{1j}	...	x_{1n}	
...
A_i	C_{i1}	...	C_{ij}	...	C_{in}	a_i
	x_{i1}	...	x_{ij}	...	x_{in}	
...
A_m	C_{m1}	...	C_{mj}	...	C_{in}	a_m
	x_{m1}	...	x_{mj}	...	x_{in}	
Потреби	b_1	...	b_j	...	b_n	

2.2.2 Аналіз методу північно-західного кута

Згідно методу північно-західного кута заповнення таблиці перевезень транспортної задачі починається з лівого верхнього, тобто північно-західного, кута, не враховуючи вартостей перевезень.[4] У клітину записують менше з двох чисел a_1 та b_1 . Далі переходять до наступної клітинки в цьому ж рядку або у стовпчику і заповнюють її і так далі. Закінчують заповнення даної таблиці у правій нижній клітинці. У такий спосіб значення поставок будуть розташовані по діагоналі таблиці.

Для того, щоб закріпити даний алгоритм більш детально розглянемо наступний приклад: на три бази $A_1; A_2; A_3$ поступив товар в кількості 140; 180; 160. Цей вантаж треба перевезти в п'ять пунктів призначення $B_1; B_2; B_3; B_4; B_5$ в кількостях 60; 70; 120; 130; 100.

Тарифи перевезення записані в таблиці, наведеній на рисунку 1.2.

Таблиця 2.2 – Тарифи перевезення

Пункти відправлення	Пункти призначення					Запаси
	B_1	B_2	B_3	B_4	B_5	
A_1	2	3	4	2	4	140
A_2	8	4	1	4	1	180
A_3	9	7	3	7	2	160
Потреби	60	70	120	130	100	480

Для того, щоб знайти план перевезення, спочатку, не враховуючи вартостей перевезення, задовольняємо потреби першого пункту призначення B_1 , використовуючи запаси першого пункту відправлення A_1 . У нашому прикладі потреби в товарі пункту B_1 становлять $b_1 = 60$, а запаси відправника – $a_1 = 140$. Тобто із запасів першого пункту відправлення ми можемо повністю задовільнити потреби першого пункту призначення. Тому у клітинку A_1B_1 записуємо менше із значень a_1, b_1 , тобто 60. На рисунку 1.3 наведено результат першого кроку побудови плану перевезення (задоволення потреб першого пункту призначення B_1):

Таблиця 2.3 – Задоволення потреб першого пункту призначення B_1

Пункти відправлення	Пункти призначення					Запаси
	B_1	B_2	B_3	B_4	B_5	
A_1	2	3	4	2	4	140
	60					
A_2	8	4	1	4	1	180
A_3	9	7	3	7	2	160

Потреби	60	70	120	130	100	480
---------	----	----	-----	-----	-----	-----

Тепер переходимо до задоволення потреб другого пункту призначення – B_2 (рисунок 1.4), потреби якого становлять $b_2 = 70$. Після задоволення потреб пункту A_1 , залишок запасів першого пункту відправлення становить $140 - 60 = 80$ (цього достатньо, щоб задовільнити 17 потреби і другого пункту призначення). Тому записуємо в клітинку A_1B_2 значення 70 і переходимо до задоволення потреб пункту B_3 .

Таблиця 2.4 – Задоволення потреб другого пункту призначення B_2

Пункти відправлення	Пункти призначення					Запаси
	B_1	B_2	B_3	B_4	B_5	
A_1	2	3	4	2	4	140
	60	70				
A_2	8	4	1	4	1	180
A_3	9	7	3	7	2	160
Потреби	60	70	120	130	100	480

Залишок запасів у першого пункту призначення, після задоволення потреб пунктів призначення B_1 і B_2 , становить $140 - 60 - 70 = 10$. Тому третьому споживачеві від першого відправника можемо перевезти лише 10 одиниць продукції (рисунок 1.5). Отже, в клітинку A_1B_3 помістимо число 10.

Таблиця 2.5 – Задоволення потреб третього пункту призначення B_3

Пункти відправлення	Пункти призначення					Запаси
	B_1	B_2	B_3	B_4	B_5	
A_1	2	3	4	2	4	140

	60	70	10			
A_2	8	4	1	4	1	180
A_3	9	7	3	7	2	160
Потреби	60	70	120	130	100	480

Після цього, оскільки запаси першого відправника повністю вичерпані, переходимо до використання запасів наступного постачальника A_2 . Його запаси рівні $a_2 = 180$. А незадоволені потреби третього пункту призначення $120 - 10 = 110$. Тому в клітинку A_2B_3 записуємо число 110, і третій споживач, у такий спосіб, також отримав необхідну кількість продукції. Подальші кроки задоволення потреб пунктів призначення виглядають так само, тому остаточний результат буде наведений на рисунку 1.6.

Переходимо до задоволення потреб наступного споживача, а саме B_4 . У результаті часткового використання запасів другого пункту відправлення його залишок продукції становить $180 - 110 = 70$. Отже від другого пункту відправлення до четвертого пункту призначення можна перевезти лише 70 одиниць продукції. Тому клітинка A_2B_4 міститиме число 70, і цим запаси постачальника A_2 будуть також повністю вичерпані.

Переходимо до використання запасів останнього пункту відправлення A_3 . Залишок потреб четвертого пункту становить $130 - 70 = 60$. Для їх задоволення скористаємось запасами відправника A_3 . У клітинку A_3B_4 записуємо число 60 і потреби четвертого пункту також повністю задоволені.

Переходимо до останнього споживача B_5 з потребами $b_5 = 100$, які повністю задовольняються за рахунок залишку третього відправника $160 - 60 = 100$, тобто у клітинку A_3B_5 записуємо число 100. Таким чином ми отримали кінцеву таблицю, у заповнених клітинках якої містяться числа, які означають можливий план перевезення продукції з загальною вартістю $F = 60 * 2 + 70 *$

$3 + 10 * 4 + 110 * 1 + 70 * 4 + 60 * 7 + 100 * 2 = 1380$ умовних одиниць. Результат побудови опорного плану перевезення за методом північно-західного кута наведений у рисунку 1.6.

Таблиця 2.6 – Опорний план перевезення

Пункти відправлення	Пункти призначення					Запаси
	B ₁	B ₂	B ₃	B ₄	B ₅	
A ₁	2	3	4	2	4	140
	60	70	10			
A ₂	8	4	1	4	1	180
			110	70		
A ₃	9	7	3	7	2	160
				60	100	
Потреби	60	70	120	130	100	480

2.2.3 Аналіз методу мінімальної вартості

Суть методу мінімального елемента полягає в наступному: на кожному кроці заповнюють клітинку таблиці, яка має найменшу вартість перевезення одиниці продукції. Такі дії повторюють до тих пір, поки не буде розподілено всю продукцію між пунктами відправлення і пунктами призначення. [5]

Складемо розв'язок наступної задачі з допомогою цього методу. Приклад: На три бази A₁; A₂; A₃ поступив товар в кількості 160; 140 і 170 одиниць відповідно. Цей вантаж потрібно перевезти в чотири пункти призначення B₁; B₂; B₃; B₄, потреби яких становлять 120; 50; 190; 110. Тарифи перевезення записані в таблиці, яка представлена на рисунку 1.7.

Таблиця 2.7 – Тарифи перевезення

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_4	B_5	
A_1	7	8	1	2	160
A_2	4	5	9	8	140
A_3	9	2	3	6	170
Потреби	120	50	190	110	470

Найменшу вартість має перевезення, яке здійснюється з A_1 в B_3 , ціна перевезення одиниці продукції якого становить 1-ну умовну одиницю. Заповнимо дану клітинку. Оскільки відправник A_1 має в запасі 160 одиниць продукції, а пункт призначення B_3 потребує – 190, то від першого відправника третьому споживачеві можна перевезти лише 160 одиниць продукції. І таким чином запаси першого пункту відправлення повністю вичерпані (перший рядок викреслюємо з розгляду, рисунок 1.8).

Таблиця 2.8 – Часткове задоволення потреб пункту призначення B_3

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_4	B_5	
A_1	7	8	1	2	160
			160		
A_2	4	5	9	8	140
A_3	9	2	3	6	170
Потреби	120	50	190	110	470

З клітинок, що залишилися вибираємо ту, в якій знаходиться маршрут з мінімальною вартістю перевезення. Таких клітинок у нас дві: $A3B2$ і $A1B4$. Виходячи з того, що клітинка $A1B4$ знаходиться в першому рядку, а його ми на попередньому кроці викреслили з розгляду, то будемо заповнювати клітинку $A3B2$. Обсяг запасів пункту відправлення рівні $a_3 = 170$, а потреби $b_2 = 50$, тому, за рахунок запасів третього відправника, потреби другого споживача задовольняються в повному обсязі (стовбець під номером два викреслюється з розгляду) і в клітинку $A3B2$ записуємо число 50. В результаті отримуємо таблицю, яка наведена на рисунку 1.9.

Таблиця 2.9 – Задоволення потреб пункту призначення $B2$

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_4	B_5	
A_1	7	8	1	2	160
			160		
A_2	4	5	9	8	140
A_3	9	2	3	6	170
		50			
Потреби	120	50	190	110	470

Знову вибираємо клітинку (серед тих що залишилися незаповненими) з найменшою вартістю перевезень. Такою клітинкою буде $A3B3$. Виходячи з того, що запаси третього пункту відправлення становлять $a_3 = 170 - 50 = 120$, а потреби третього пункту призначення рівні $b_3 = 190 - 160 = 30$, то 21 ставимо в клітинку $A3B3$ значення 30. І таким чином потреби 3-го пункту призначення задоволені, а стовбець в якому знаходиться даний пункт викреслюємо з розгляду (рисунок 1.10).

Таблиця 2.10 – Задоволення потреб пункту призначення B_3

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_4	B_5	
A_1	7	8	1	2	160
			160		
A_2	4	5	9	8	140
A_3	9	2	3	6	170
		50	30		
Потреби	120	50	190	110	470

Продовжуючи даний процес до тих пір, поки усі запаси не будуть вичерпані, а потреби — задоволеними, ми отримаємо таблицю (рисунок 1.11), у заповнених клітинках якої містяться числа, які означають можливий план перевезення продукції із загальною вартістю $F = 120 * 4 + 50 * 2 + 160 * 1 + 3 * 30 + 20 * 8 + 90 * 6 = 1530$.

Таблиця 2.11 – Опорний план перевезення за методом мінімальної вартості

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_4	B_5	
A_1	7	8	1	2	160
			160		
A_2	4	5	9	8	140
	120			20	
A_3	9	2	3	6	170
		50	30	90	
Потреби	120	50	190	110	470

2.2.4 Аналіз методу потенціалів

Для знаходження оптимального плану транспортної задачі необхідно спочатку визначити опорний план перевезень з допомогою методу північно-західного кута чи методу мінімального елемента, які було розглянуто у попередніх підрозділах.

Теорема: Якщо X^* – деякий опорний план транспортної задачі, для якого виконуються наступні обмеження, а саме:

$$\beta_j + \alpha_i = C_{ij} \text{ для } x_{ij} > 0$$

$$\beta_j + \alpha_i \leq C_{ij} \text{ для } x_{ij} = 0$$

тоді даний опорний план є оптимальним. α_i ; β_j – називаються потенціалами пунктів відправлення та пунктів призначення.

Перевірку опорного плану на оптимальність здійснюють за допомогою методу потенціалів.

Алгоритм даного методу полягає в наступному:

– знаходимо потенціали пунктів відправлення (α_i) і пунктів призначення (β_j). Для цього, для кожної зайнятої клітинки будемо лінійне рівняння виду $\beta_j + \alpha_i = C_{ij}$. В результаті отримуємо систему лінійних рівнянь, розв'язавши яку знайдемо шукані потенціали. Зауваження: Зайнятих клітинок завжди є $n+m-1$. Значить ми отримаємо $n+m-1$ рівняння з $n+m$ невідомими. Щоб розв'язати дану систему покладемо $\alpha_1 = 0$;

– для усіх вільних клітинок знаходимо значення $\alpha_{ij} = C_{ij} - \beta_j - \alpha_i$. Якщо всі елементи α_{ij} будуть додатними, то побудований опорний план є оптимальним. Якщо ж серед значень α_{ij} будуть від'ємні елементи, то це означає, що побудований опорний план не являється оптимальним і є можливість переходу до іншого опорного плану.

Алгоритм переходу до іншого опорного плану наступний:

– серед усіх від'ємних α_{ij} знаходимо максимальне по модулю значення.

Це означає, що дана клітинка повинна бути зайнятою;

- для клітинки в якій знаходиться максимальне по модулю значення будуємо цикл перерахунку;
- починаючи з клітинки, де знаходиться α_{ij} , кожна вершину циклу позначаємо знаком плюс і мінус, чергуючи їх послідовно;
- серед значень клітинок позначених мінусом вибираємо мінімальне значення;
- до клітинок позначених знаком плюс додаємо це (мінімальне) значення, а від клітинок позначених знаком мінус
- віднімаємо мінімальне значення.

Клітинка де знаходилось мінімальне значення стає вільною. В результаті отримуємо новий опорний план, який знову потрібно перевірити на оптимальність. [6]

Висновок до розділу 2

У даному розділі було розглянуто існуючі методи оптимізації транспортних перевезень. Наступний розділ буде присвячений вибору інструментів для розробки.

РОЗДІЛ 3

ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Androin Studio

Android Studio – це інтегроване середовище розробки (IDE) призначене для роботи з платформою Android. Android Studio заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, що є офіційним засобом розробки Android додатків[7].

Ця програма для розробки мобільних додатків підтримує кілька мов програмування, серед яких Kotlin, Java, Dart, в наявності вбудований емулятор і велика бібліотека з різними компонентами і шаблонами, які істотно полегшують і прискорюють процес розробки програмних продуктів.

В IDE можна створювати додатки для останньої версії Android, а створену програму можна одразу ж перевірити на помилки, протестувати різноманітними інструментами всі елементи програми і заздалегідь знайти всі можливі баги в її роботі. Вбудований емулятор дозволяє проводити тести коректності і продуктивності роботи додатків, які розробляються на різних системах і, за необхідності, здійснювати оптимізацію. Також емулятор дозволяє проводити тестування на пристроях, екрани яких мають різне співвідношення сторін. Особливо актуальною ця функція стала після входу в тренди смартфонів, в яких встановлені екрани зі співвідношенням сторін 18:9. Відмінна особливість емулятора – перегляд приблизних показників продуктивності при запуску програми на найпопулярніших пристроях.

Android Studio є привабливим для розробників-початківці, оскільки це середовище більш зрозуміле і в ньому можна працювати навіть без особливого досвіду в програмуванні. Локалізація додатків стає значно простіше з функцією SDK, яка також входить до переліку переваг Android Studio.[8]

Отже, у підсумку, перевагами і особливостями Android Studio будуть наступні пункти:

а) середовище розробки підтримує роботу з декількома мовами програмування, до яких відносяться найпопулярніші – Java, Kotlin.

б) редактор коду, з яким зручно працювати;

в) дозволяє розробляти додатки не тільки для смартфонів / планшетів, а й для портативних ПК, мобільних пристроїв з незвичайним співвідношенням сторін екрану, приставок для телевізорів Android TV, пристроїв Android Wear,;

г) тестування коректності роботи нових ігор, утиліт, їх продуктивності в тій чи іншій системі відбувається безпосередньо в емуляторі;

д) рефакторинг вже готового коду;

е) розробка програми для останньої версії операційної системи Android;

ж) попередня перевірка вже створеного додатку на предмет помилок в ньому;

з) великий набір засобів інструментів для тестування кожного елемента програми, ігри;

и) для недосвідчених / початківців розробників спеціально створено посібник з використання Android Studio, який розміщено на офіційному сайті;

к) базування на Gradle;

л) шаблони для створення поширених Android дизайнів та компонентів.

Недоліки

Android Studio є практично ідеальним середовищем для розробки мобільних додатків під Android, але для комфортної роботи в IDE знадобиться потужний комп'ютер. Сучасні комп'ютери часто повністю відповідають всім необхідним вимогам, а ось на старих машинах – можливі проблеми. Незважаючи на наявність вбудованого Android-емюлятора в самому середовищі розробки, з тестуванням додатків можуть виникнути

труднощі. Запуск емулятора потребує досить значної за продуктивністю апаратної основи ПК, на якому буде проводитися тестування. Також недоліком є неможливість написати серверні проекти на мові Java для ПК та Android пристроїв [9].

3.2 Flutter SDK

Flutter - це повноцінний SDK. В основі його лежить мова програмування DART – який у свою чергу базується на відомій Java. Тому знаючи Java або C# – можна легко перейти програмувати на Dart.[10]

Flutter – SDK для створення високоякісних інтуїтивних GUI на платформах iOS та Android. Flutter легко приєднується до поточних розробок та використовується розробниками по всьому світу. А найголовніше він має відкритий код і повністю безкоштовний.

Розглядаючи цю заяву в контексті можна зробити найпростіший висновок. Компанія Google представляє новий продукт на основі власних розробок. Завдяки шліфуванню коду Dart та людської інтерпретації у віртуальну машину Java з подальшою інтеграцією в Осі Android та iOS він дозволяє дійсно скоротити час розробки складних інтерфейсів у рази. Це підтверджує як мінімум одна серйозна компанія, що на повну катушку використовує ПЗ від Alphabet inc. – а саме їхній головний конкурент на ринках Азії Alibaba Corp.

3.3 Мова програмування Dart

Dart є оптимізованою для клієнта мовою програмування для програмування на декількох платформах. Він розроблений Google і використовується для створення мобільних, десктопних, серверних та веб-додатків.[11]

Dart – це об'єктно-орієнтований, на основі класу, сміття зібране, C-стиль синтаксис.[12] Він підтримує інтерфейси, міксини, абстрактні класи, рейфікацію, дженерики, і висновок типу.[13]

Не можна не звернути увагу на найголовнішу складову Flutter – а саме мову програмування DART. На вигляд ми маємо комбінацію між Java і Javascript. Погано це чи добре?

З одного боку - використання DART обмежує вас у методах розробки GUI - наприклад ви не зможете повноцінно використовувати сторонні бібліотеки в тому ключі як би Вам хотілося. З іншого боку, Dart не може надати повноцінного функціоналу для створення нативних інтерфейсів.

Натомість нам пропонують:

- Набір бібліотек для швидкого розробки програм.
- Обрізки Java машини.
- Просту інтеграцію з iOS та Android без "велосипедобудування".

На виході маємо конструктор леґо для створення інтерфейсів, із зручною машиною інтерпретації для впровадження у вихідних код основного ПЗ.

Що стосується продуктивності мови - то, звичайно, вона значно поступається коду написаному на Java. Так, навіть на досить стародавніх пристроях (HTC з андроїд 4.1. під 1 ррц ядро та 512 ОЗУ). Складний адаптивний GUI видає цілком прийнятну продуктивність, а у випадку з програмуванням на iOS внутрішня оптимізація інтерпретатора дозволяє не бачити різниці навіть при сильному завантаженні центрального ядра.

Насамкінець дві речі, які важливі для програмістів. З кожним новим оновленням Flutter підрубує все більше кросплатформеності для програмістів.

Друга річ – це документація. Саме за рахунок документації можна розібратися в програмуванні на Dart навіть якщо до цього ви використовували тільки графічні інструменти Unity для створення іграшок під Android.

Звичайно flutter буде незвичним для користувачів «шарпів» та класичної «ява машини». Але це не означає, що вам доведеться виходити із зони комфорту. Освоївши невеликі відмінності синтаксису можна незабаром переконатися, що розробка інтерфейсу користувача скоротиться в кілька разів у порівнянні з іншими методами.

Основні плюси мови Dart

Ось короткий список функцій Dart, які разом роблять його незамінним для Flutter:

1. Dart – це AOT (Ahead Of Time) , скомпільований у швидкий, передбачуваний та нативний код, що дозволяє писати майже весь Flutter на Dart. Це не тільки робить Flutter швидким, але й дозволяє налаштувати практично все (включаючи усі віджети).
2. Dart також може бути скомпільований JIT (Just In Time) для виключно швидких циклів розробки та революційного робочого процесу (включаючи популярне у Flutter гаряче перезавантаження зі збереженням стану за частки секунди).
3. Dart спрощує створення плавних анімацій та переходів зі швидкістю 60 кадрів за секунду. Dart може виконувати аналіз та складання сміття (Garbage Collector) без блокувань. Як і JavaScript, Dart уникає попереджувального планування і пам'яті, що розділяється (і, отже, блокувань). Оскільки програми Flutter скомпільовані в нативний код, їм не потрібний повільний міст між областями (наприклад, JavaScript в нативний код). Вони також запускаються набагато швидше.
4. Dart дозволяє Flutter уникнути необхідності в окремій мові декларативного компонування, такому як JSX або XML, або окремих конструкторів візуальних інтерфейсів, оскільки декларативне програмне компонування Dart легко читати та візуалізувати. А з усім компонуванням однією мовою і в одному місці Flutter легко надає розширені інструменти, які роблять компонування простіше простого.
5. Розробники виявили, що Dart особливо легко освоїти, оскільки має функції, знайомі користувачам як статичних, і динамічних мов.

Не всі ці функції є унікальними для Dart, але їх комбінація забезпечує найкращі результати, які й роблять Dart унікально потужним інструментом для реалізації Flutter. Настільки важко уявити, що Flutter може бути таким потужним без Dart.

Компіляція та виконання

До роботи над мовою Dart, його члени команди проробили новаторську роботу над передовими компіляторами та віртуальними машинами як для динамічних мов (таких як V8 для JavaScript і Strongtalk для Smalltalk), так і для статичних мов (таких як компілятор Hotspot для Java). Вони використали цей досвід, щоб зробити Dart надзвичайно гнучким у плані компіляції та виконання.

Dart - одна з дуже небагатьох мов (і, можливо, єдина "основна" мова), яка добре підходить для компіляції як АОТ, так і JIT. Підтримка обох видів компіляції дає істотні переваги Dart та, особливо, Flutter.

Компіляція JIT використовується під час розробки з особливо швидкого компілятора. Потім, коли програма готова до випуску, вона компілюється АОТ. Отже, за допомогою передових інструментів та компіляторів Dart може надати найкраще з обох світів: надзвичайно швидкі цикли розробки та швидкий час виконання та запуску.

Гнучкість Dart у компіляції та виконанні на цьому не закінчується. Наприклад, Dart може скомпілювати JavaScript , щоб він міг виконуватися браузерами. Це дозволяє повторно використовувати код між мобільними програмами та веб-додатками. Розробники повідомляють про 70% повторного використання коду між мобільними та веб-додатками. Dart також можна використовувати на сервері або шляхом компіляції у власний код, або шляхом компіляції JavaScript і використання його з Node.js .

Нарешті, Dart також надає автономну віртуальну машину, яка використовує саму мову Dart як свою проміжну мову (по суті, діючи як інтерпретатор).

Dart можна ефективно компілювати AOT чи JIT, інтерпретувати чи транслювати іншими мовами. Компіляція та виконання Dart не тільки надзвичайно гнучкі, а й особливо швидкі.

Однією з найпопулярніших особливостей Flutter є надзвичайно швидке гаряче перезавантаження. Під час розробки Flutter використовує компілятор JIT, який може перезавантажити та продовжити виконання коду, як правило, менш ніж за секунду. Стан програми зберігається при перезавантаженні, коли це можливо, тому програма може продовжувати роботу з того місця, де вона була зупинена.

Важко оцінити, наскільки важливою може бути дійсно швидке і надійне гаряче перезавантаження під час розробки, якщо ви не випробували її на собі. Розробники повідомляють, що це змінює те, як вони створюють свої програми, описуючи, що це схоже на те, ніби вони малюють свою програму наживо.

Гаряче перезавантаження Flutter значно полегшує реалізацію нових ідей або експериментування з альтернативами, забезпечуючи величезний імпульс творчості.

Робота з плавністю

Швидкий додаток - це добре, але плавне - ще краще. Навіть надшвидка анімація виглядатиме погано, якщо вона смикається. Тим не менш, запобігти ривку може бути складно, тому що існує дуже багато різних причин. Dart має ряд функцій, що дозволяють уникнути багатьох поширених речей, що викликають зависання.

Звичайно, (як і в будь-якій мові) на Flutter все ще можна написати смиканий додаток. Dart допомагає, будучи більш передбачуваним і надаючи розробнику більший контроль над плавністю роботи своєї програми, спрощуючи забезпечення найкращого досвіду користувача, без винятку.

Працюючи зі швидкістю 60 кадрів в секунду, інтерфейси користувача, створені за допомогою Flutter, працюють набагато краще, ніж створені за

допомогою інших кросплатформних середовищ розробки. І не тільки краще, ніж кросплатформні додатки, але й не гірше за кращі нативні додатки.

У скомпільованого коду AOT є ще більша перевага, що полягає в тому, що він дозволяє уникнути "моста JavaScript". Коли динамічні мови (такі як JavaScript) повинні взаємодіяти з власним кодом на платформі, вони повинні взаємодіяти через міст, що викликає перемикавання контексту, що має зберігати особливо велику кількість станів (можливо, у вторинному сховищі). Ці перемикавання контексту завдають подвійного удару по продуктивності, оскільки вони уповільнюють роботу, а й можуть викликати серйозні збої.

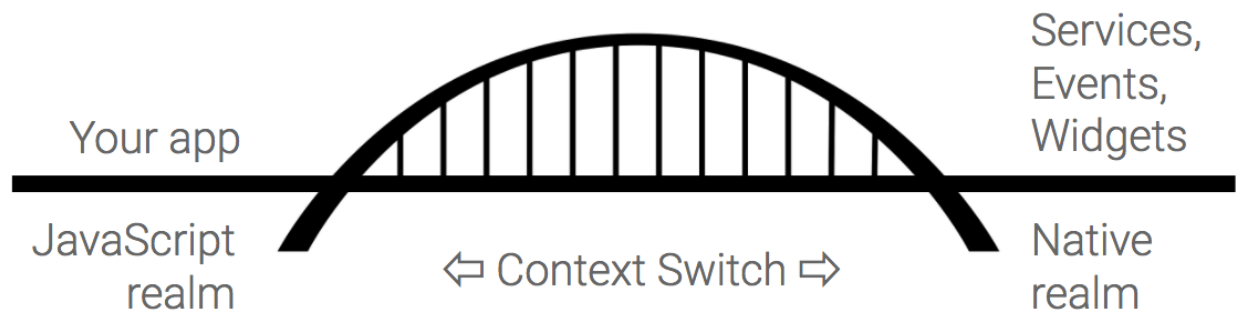


Рисунок 3.1 – «Міст JavaScript»

Навіть скомпільованому коду може бути потрібний інтерфейс для взаємодії з кодом платформи, і це також можна назвати мостом, але зазвичай він на кілька порядків швидше, ніж міст, необхідний динамічною мовою. Крім того, оскільки Dart дозволяє переміщати такі речі, як віджети, додаток, необхідність переходу через міст зменшується.

Запобіжне планування, розподіл часу та загальні ресурси

Більшість комп'ютерних мов, які підтримують кілька паралельних потоків виконання (включаючи Java, Kotlin, Objective-C і Swift), використовують витіснення перемикавання між потоками. Кожному потоку виділяється "шматочок" часу виконання, і якщо він перевищує виділений час, потік переривається з допомогою перемикавання контексту. Однак якщо

витіснення відбувається при оновленні ресурсу, що спільно використовується потоками (наприклад, пам'яті), це викликає стан гонки.

Стан гонки— це подвійний удар по продуктивності, тому що вони можуть спричинити серйозні помилки, включаючи збій вашої програми та втрату даних, і їх особливо важко знайти та виправити, тому що вони залежать від відносного часу незалежних потоків. Занадто часто умови стану перегонів перестають виявлятися, коли ви запускаєте програму у відладчику.

Типовий спосіб виправити стан гонки - захистити загальний ресурс за допомогою блокування, що перешкоджає виконанню інших потоків, але самі по собі блокування можуть викликати зависання або навіть серйозніші проблеми (включаючи взаємоблокування та голодування).

Dart по-іншому підійшов до цієї проблеми. Потоки в Dart, які називаються ізолятами, не використовують спільну пам'ять, що усуває необхідність у більшості блокувань. Ізоляти спілкуються, передаючи повідомлення каналами, що схоже на акторів в Erlang або web-workers в JavaScript.

Dart, як і JavaScript, є однопотоким, що означає, що він взагалі не допускає витіснення. Натомість потоки явно поступаються (використовуючи `async/await`, `Futures` або `Streams`). Це дає розробнику більше контролю за виконанням. Однопотоківість допомагає розробнику гарантувати, що критично важливі функції (включаючи анімацію та переходи) виконуються повністю, без витіснення. Це часто є великою перевагою не тільки для інтерфейсів користувача, але і для іншого клієнт-серверного коду.

Звичайно, якщо розробник забуває передати керування, це може призвести до затримки виконання іншого коду. Однак ми виявили, що забування `yield` зазвичай набагато легше знайти і виправити, ніж забування `lock` (оскільки важко знайти умови стану гонки).

Ще одна серйозна причина сміття – збір сміття (Garbage Collection). Справді, це лише окремий випадок доступу до спільного ресурсу (пам'яті),

який у багатьох мовах вимагає використання блокувань. Але блокування можуть перешкодити запуску всієї програми під час збирання вільної пам'яті. Проте Dart майже завжди може виконувати складання сміття без блокувань.

Dart використовує вдосконалену схему складання сміття і розподілу, яка особливо швидко розподіляє безліч недовговічних об'єктів (ідеально підходить для реактивних інтерфейсів, таких як Flutter, які перебудовують незмінне дерево уявлення для кожного кадру). Dart може виділяти об'єкт одним використанням покажчика (блокування не потрібне). Знову ж таки, це призводить до плавної прокрутки та анімації без ривків. [14]

Висновок

Перелічені причини звичайно суб'єктивні, але засновані на тому, як розвивається Flutter і Dart. Істину знають лише у Google.

Dart дозволяє Flutter бути саме таким, яким він є – крутим фреймворком для створення якісних кросплатформних додатків. У Dart склалися разом всі чинники: він цілком сучасний і має всі необхідні властивості, простий в освоєнні, належить Google. Ідеальний кандидат для кросплатформного фреймворку, який змінить світ.

3.4 Кросплатформність

Функціональність мобільних додатків є вкрай різноманітною: від ігор та сервісів з надання послуг, до офісних програм та фітнес-трекерів. Однак всі додатки можна розділити на 2 категорії: нативні та кросплатформні. [15]

Нативні додатки

Це додатки, розроблені для конкретної платформи (Android, IOS, Windows та інші), з урахуванням особливостей даної платформи та доступом до всіх її ресурсів.

Переваги нативних додатків:

- Просте опанування. Інтерфейс користувача у нативних додатків наслідують ідеологію дизайну, що закладена у конкретну операційну систему. Тому опанувати такий додаток не складно, а його використання не буде викликати дисонансу з вже існуючим досвідом користувача.
- Висока продуктивність. Нативні додатки оптимізовано під певну мобільну операційну систему, в наслідок чого вони працюють швидко і дуже стабільно. Окрім того, дані таких додатків зберігаються здебільшого на пристрої, що також збільшує швидкість їх роботи та зменшує залежність від Інтернету.
- Широкі можливості. Такий тип додатків має доступ до всіх можливостей ОС та компонентів пристрою: GPS, камери, календаря, адресної книги та іншого. Завдяки цьому, нативні додатки мають широку функціональність та легко інтегруються між собою.

Кросплатформні додатки

Це універсальні додатки, які створені для багатьох платформ відразу і мають таку ж саму функціональність незалежно від самої платформи. Можна провести аналогію, що кросплатформний додаток – це як веб-сайт у звичному форматі мобільного додатку.

Переваги кросплатформних додатків:

- Універсальність. Кросплатформні додатки створюються для кількох платформ одразу, після чого відносно просто адаптуються під кожен. Сюди входить не тільки версія для Android чи iOS, а також десктопні та веб-версії. Завдяки цьому, кросплатформні додатки мають значно більшу аудиторію користувачів.
- Дешевша вартість розробки. Цей тип додатків простіший та швидший у розробці, тому їх початкова собівартість нижча, ніж у нативних додатків. Тому для компаній, що мають обмежений

стартовий бюджет, кросплатформні додатки є більш привабливим рішенням.

- Швидший вихід на ринок. Простіша розробка кросплатформних додатків надає ще одну перевагу – менший термін розробки, що дозволяє відносно швидко випустити продукт і отримати перший прибуток. Якщо компанія не має можливості витратити багато часу на розробку, то такий тип додатків є доцільним рішенням.

Висновок до розділу 3

У даному розділі наведені характеристики інструментів, що були використані під час розробки кросплатформного додатку. У наступному розділі буде розглядатися сам процес розробки.

РОЗДІЛ 4

ОГЛЯД ЗАПРОПОНОВАНОГО МЕТОДУ ОПТИМІЗАЦІЇ ТРАНСПОРТНОГО ПЕРЕВЕЗЕННЯ ТА СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ

Для демонстрації роботи оптимізації транспортних перевезень з використанням вдосконаленої моделі було розроблено програмне забезпечення з запропонованим вдосконаленим методом оптимізації транспортної задачі. Це не є повноцінною інформаційною системою, а тільки фрагмент автоматизації поставленої у рамках даної роботи задачі.

4.1 Опис запропонованого методу оптимізації транспортної задачі

Ідея методу полягає в тому, що в таблиці перевезень транспортної задачі спершу заповнюються клітинки з мінімальним тарифом у кожному стовпчику. В ці клітинки вноситься уся потреба для відповідного пункту призначення. Далі з рядків, які переповнені, тобто перевищує запас, потрібно перемістити у рядок, в якому недобір по найменшій різниці тарифів стільки одиницю продукції, скільки потрібно, щоб сума в рядку дорівнювала його запасу. Повторювати цей алгоритм доти, доки сума одиниць продукції не буде дорівнювати його запасу.

Для того, щоб закріпити даний алгоритм більш детально розглянемо наступний приклад: на три бази A_1 ; A_2 ; A_3 поступив товар в кількості 300; 280; 240. Цей вантаж треба перевезти в чотири пункти призначення B_1 ; B_2 ; B_3 ; B_4 ; в кількостях 200; 190; 220; 210.

Тарифи перевезення записані в таблиці, наведеній на рисунку 4.1.

Таблиця 4.1 – Тарифи перевезення

Пункти відправлення	Пункти призначення				Запаси
	B_1	B_2	B_3	B_4	

A ₁	5	6	8	4	300
A ₂	9	7	10	11	280
A ₃	14	4	6	12	240
Потреби	200	190	220	210	820

Проставляємо в клітинки з мінімальним тарифом кожного стовпця усю кількість одиниць продукції. Тобто в A₁V₁ з тарифом 5 поміщаємо 200 одиниць, в A₃V₂ з тарифом 4 поміщаємо 190 одиниць, в A₃V₃ з тарифом 6 поміщаємо 220 одиниць, в A₁V₄ з тарифом 4 поміщаємо 210 одиниць.

Таблиця 4.2 – Заповнення клітинок з мінімальним тарифом кожного стовпця

Пункти відправлення	Пункти призначення				Запаси
	V ₁	V ₂	V ₃	V ₄	
A ₁	5	6	8	4	300
	200			210	
A ₂	9	7	10	11	280
A ₃	14	4	6	12	240
		190	220		
Потреби	200	190	220	210	820

Бачимо, що сума одиниць продукції в клітинках складах A₁ та A₃ перевищують значення запасу у відповідних складах, тому якусь кількість одиниць продукції в певному пункті призначені потрібно перемістити на склад A₂. Визначаємо мінімальну різницю між тарифами для кожного пункту

призначення: $V_1 - 9-5=4$; $V_2 - 7-4=3$; $V_3 - 10-6=4$; $V_4 - 11-4=7$. Отже, найменша різниця тарифів у V_2 , тому одиниці продукції слід перемістити з клітинки A_3V_2 в клітинку A_2V_2 у розмірі 170 одиниць, так сума одиниць продукції в клітинках складах A_3 дорівнює його запасу 240.

Таблиця 4.3 – Оптимізація плану перевезень

Пункти відправлення	Пункти призначення				Запаси
	V_1	V_2	V_3	V_4	
A_1	5	6	8	4	300
	200			210	
A_2	9	7	10	11	280
		170			
A_3	14	4	6	12	240
		20	220		
Потреби	200	190	220	210	820

Далі потрібно одиниці продукції перемістити з складу A_1 в склад A_2 . Визначаємо мінімальну різницю між тарифами для пунктів призначення V_1 та V_4 , оскільки лише там є що перемістити з складу A_1 в склад A_2 : $V_1 - 9-5=4$; $V_4 - 11-4=7$. Отже, найменша різниця тарифів у V_1 , тому одиниці продукції слід перемістити з клітинки A_1V_1 в клітинку A_2V_1 у розмірі 110 одиниць, так сума одиниць продукції в клітинках складах A_1 та A_2 дорівнюють їх запасу 300 та 280 відповідно.

Оптимізований план перевезень продемонстрований у таблиці 4.4.

Даний метод при перевірці на оптимальність методом потенціалів майже завжди показує, що план оптимальний, що зменшує кількість ітерацій та швидкість обчислення.

Таблиця 4.4 – Оптимізований план перевезень

Пункти відправлення	Пункти призначення				Запаси
	B ₁	B ₂	B ₃	B ₄	
A ₁	5	6	8	4	300
	90			210	
A ₂	9	7	10	11	280
	110	170			
A ₃	14	4	6	12	240
		20	220		
Потреби	200	190	220	210	820

4.2 Опис розробленої програми

Розроблене ПЗ є демонстраційним додатком для оптимізації транспортних перевезень при побудові плану перевезень чи його корегуванні.

Робота програми розпочинається з форми авторизації (Рис 4.1 та Рис 4.2).

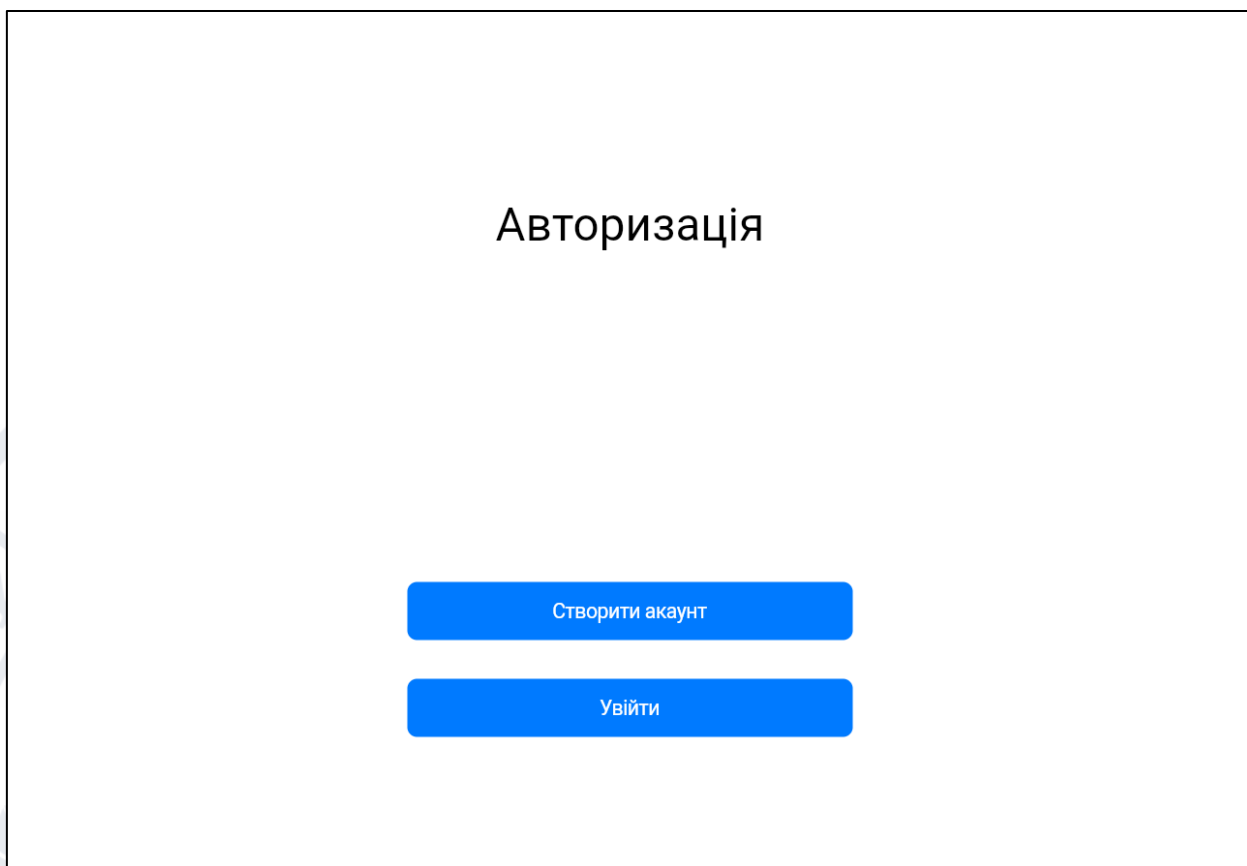


Рисунок 4.1 - Стартова сторінка

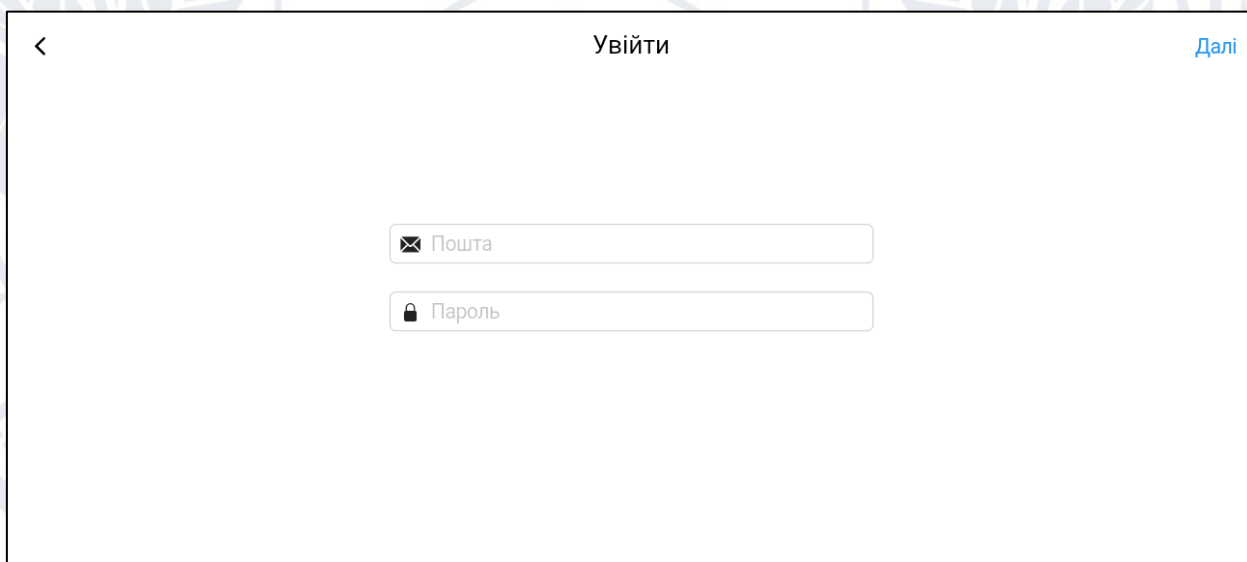


Рисунок 4.2 – Веб-інтерфейс авторизації

Після успішної авторизації користувач програми має змогу побудувати план перевезень.

На рисунку 4.3 наведено форма для вводу розміру матриці тарифів.
На рисунку 4.4 наведено форму для побудови нового плану перевезень

класичному для транспортної задачі вигляді – заповнення запасів постачальників, потреб споживачів та вартості перевезень відповідно (матриця тарифів).

Transport

Розмір

3 4 **Застосувати**

Рисунок 4.3 - Розмір матриці тарифів

Transport

Розмір

3 4 **Застосувати**

	Споживач 1	Споживач 2	Споживач 3	Споживач 4	Запаси
Постачальник 1	5	6	8	4	300
Постачальник 2	9	7	10	11	280
Постачальник 3	14	4	6	12	240
Потреби	200	190	220	210	

Розрахувати

Рисунок 4.4 - Веб-інтерфейс побудови плану перевезень

Веб-інтерфейс результату побудови плану перевезень наведений на рисунку 4.5 і являє собою таблицю з детальною інформацією щодо вантажного перевезення, а також відображає актуальний стан маршруту.

Результати							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Постачальник 1	Споживач 1	90	5	149	3.3	450	Хороший
Постачальник 1	Споживач 4	210	4	115	11.3	840	Хороший
Постачальник 2	Споживач 1	110	9	65	1.8	990	Хороший
Постачальник 2	Споживач 2	170	7	66	12.5	1190	Хороший
Постачальник 3	Споживач 2	20	4	2	7.6	80	Жахливий
Постачальник 3	Споживач 3	220	6	93	6.9	1320	Задовільний
Всього: 6565							

Рисунок 4.5 - Веб-інтерфейс побудованого плану перевезень. Якщо на певному маршруті є певні проблеми (погіршення погодних умов, ремонт доріг тощо), користувачу одразу пропонується скорегувати маршрут, виконавши пошук альтернативного шляху (рисунок 4.6).

Результати							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Змінити параметри							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Постачальник 3	Споживач 2	20	<input type="text" value="4"/>	<input type="text" value="2"/>	<input type="text" value="7.6"/>	<input type="text" value="80"/>	Жахливий
Скасувати Застосувати							

Рисунок 4.6 - Корегування маршруту

Результати							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Змінити параметри							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Постачальник 3	Споживач 2	20	<input type="text" value="5"/>	<input type="text" value="3"/>	<input type="text" value="4.5"/>	<input type="text" value="100"/>	Хороший
Скасувати Застосувати							

Рисунок 4.7 - Корегування маршруту

Відкоригувавши маршрут, бачимо що сумарна вартість перевезення зменшилась на 156,7 одиниць продукції. Хоч і на одному з маршрутів збільшився тариф перевезення, проте вдалося зекономити на паливі за рахунок кращого стану доріг.

Результати							
Постачальник	Споживач	Кількість товару	Тариф перевезення	Відстань, км	Витрати палива л/100км	Вартість перевезення	Стан маршруту
Постачальник 1	Споживач 1	90	5	149	3.3	450	Хороший
Постачальник 1	Споживач 4	210	4	115	11.3	840	Хороший
Постачальник 2	Споживач 1	110	9	65	1.8	990	Хороший
Постачальник 2	Споживач 2	170	7	66	12.5	1190	Хороший
Постачальник 3	Споживач 2	20	5	3	4.5	100	Хороший
Постачальник 3	Споживач 3	220	6	100	2.9	1320	Хороший
Всього: 6408							

Рисунок 4.8 - Відкоригований маршрут

4.3 Аналіз практичного використання вдосконаленої моделі транспортної задачі

Доведення отриманих результатів на практиці – один серед найважливіших етапів будь-якої наукової роботи. Для демонстрації роботи існуючих методів з використанням удосконаленої моделі транспортної задачі було розроблене демонстраційне ПЗ, яке дозволяє виконати усі етапи оптимізації транспортних перевезень на етапі побудови їх маршрутів або оперативному корегуванні.

Перевагами реалізації даного програмного забезпечення є:

- простота використання. ПЗ має дуже простий і зрозумілий інтерфейс користувача;
- результати корегування плану перевезень порівнюються згідно визначеним критеріям ефективності;
- на відміну від існуючих систем для транспортної логістики, розроблене ПЗ є дуже простим – воно не потребує використання значних ресурсів та легко інтегрується до будь-якої іншої інформаційної системи з мінімальними затратами;
- результати, отримані за допомогою розробленого ПЗ, можуть бути перевірені та підтверджені ручним обчисленням.

Враховуючи те, що розроблене ПЗ є демонстраційним, та як правило, має деякі обмеження, то серед недоліків можна відмітити лише те, що вхідні дані щодо запасів та потреб постачальників і споживачів відповідно, потрібно вводити вручну і для великого замовлення, це може

бути не зовсім ефективно з точки зору трудовитрат.

Як висновок можна сказати те, що розроблене ПЗ дозволяє вирішити поставлену у рамках даної кваліфікаційної магістерської роботи завдання, а саме оптимізація транспортних перевезень при побудові/корегуванні плану перевезень. Остаточний план перевезень, зображений на рисунку 4.5, повністю співпадає з результатами, які було отримано при ручному обчисленні у розділі 4.1.

Висновок до розділу 4

У даному розділі було розглянуто запропонований метод оптимізації транспортної задачі та розробку додатку. Наведені екрани самого додатку та їх опис.

ВИСНОВКИ

В ході виконання практичної підготовки було проведено дослідження методів оптимізації транспортних перевезень.

Було обґрунтовано і доведено актуальність дослідження існуючих методів оптимізації транспортних перевезень, обґрунтовано мету щодо вдосконалення математичної моделі транспортної задачі, яка може бути використана із вже існуючими методами, а також було сформульовано постановку задачі та оглянуті програми аналогів.

В результаті досліджень було розроблено вдосконалену математичну модель транспортної задачі, нові обмеження якої потрібні для вирішення задачі побудови плану перевезень чи для його оперативного корегування, враховуючи такі фактори, як актуальний стан якості дорожнього покриття маршрутів, а також пропускна здатність цих доріг. Було описано етапи оптимізації транспортних перевезень при побудові опорного плану перевезень, а також розроблено алгоритм реалізації вдосконаленої моделі із вдосконаленим методом оптимізації транспортних перевезень.

Для демонстрації побудови плану транспортних перевезень з використанням вдосконаленої моделі, було розроблено відповідне програмне забезпечення.

Розроблена в ході даної роботи модель транспортної задачі не є досконалою та потребує доробок, тому у майбутньому планується продовження досліджень у напрямку оптимізації транспортних перевезень на етапі побудови опорного плану чи його оперативному корегуванні.

Отже, всі поставлені задачі було виконано в повному об'ємі та результат доведений на практиці. Також було отримано та засвоєно нові знання з розробки кросплатформних додатків на Flutter з допомогою мови програмування Dart.

СПИСОК ЛІТЕРАТУРИ

1. Тенденції розвитку транспорту і зв'язку [Електронний ресурс] – Режим доступу до ресурсу: <http://buklib.net/books/33285/>
2. Решение транспортной задачи - OnLine [Електронний ресурс] – Режим доступу до ресурсу: http://www.math-pr.com/tzd_1.php
3. Транспортна задача лінійного програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathros.net.ua/transportna-zadacha-matematychna-postanovka-zadachi.html>
4. Метод північно-західного кута [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mathros.net.ua/metod-pivnichno-zahidnogo-kuta.html>
5. Метод мінімального елемента [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mathros.net.ua/metod-minimalnogo-elementa.html>
6. Метод потенціалів [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mathros.net.ua/znahodzhennja-optymalnogo-planu-transportnoi-zadachi-metodom-potencialiv.html>
7. Android Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/studio>
8. Android Studio and IntelliJ [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.flutter.dev/development/tools/android-studio>
9. The Major Advantages of Android Studio App Development [Електронний ресурс] – Режим доступу до ресурсу: <https://www.indianappdevelopers.com/blog/advantages-of-android-studio-app-development/>
10. Flutter – корисна річ чи черговий хайп? [Електронний ресурс] – Режим доступу до ресурсу: <https://web-academy.ua/blog/junior/flutter>
11. Dart [Електронний ресурс] – Режим доступу до ресурсу: <https://dart.dev/>
12. A tour of the Dart language [Електронний ресурс] – Режим доступу до ресурсу: <https://dart.dev/guides/language/language-tour#important-concepts>
13. The Dart type system [Електронний ресурс] – Режим доступу до ресурсу: <https://dart.dev/guides/language/type-system>

14. Що таке Dart? [Електронний ресурс] – Режим доступу до ресурсу: <https://edu.gss.guru/dart/>
15. Типи мобільних додатків [Електронний ресурс] – Режим доступу до ресурсу: <https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types>
16. Козаченко Д. М., Вернигора Р. В., Малашкін В. В. Основи дослідження операцій у транспортних системах: приклади та задачі: навч. посіб. Дніпро : ДНУЗТ, 2015. 280 с.
17. Кучма М.І. Математичне програмування: приклади і задачі: навч. посіб. — Львів: Новий Світ — 2020. — 344 с.
18. Транспортна задача [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5392511/>
19. Транспортна задача [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/52016/ekonomika/transportna_zadacha
20. Транспортна задача лінійного програмування – Математика [Електронний ресурс] – Режим доступу до ресурсу: http://8ref.com/6/referat_6603.html
21. Побудова опорного плану транспортної задачі методом Фогеля [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathros.net.ua/pobudova-opornogo-planu-transportnoi-zadachi-metodom-fogelja.html>
22. Знаходження початкового опорного плану транспортної задачі методом плаваючих зон [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathros.net.ua/znahodzhennja-pochatkovogo-opornogo-planu-transportnoi-zadachi-metodom-plavajuchyh-zon.html>
23. Побудова опорного плану транспортної задачі методом подвійної переваги [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathros.net.ua/pobudova-opornogo-planu-transportnoi-zadachi-metodom-podvijnoi-perevagy.html>
24. Is the Transportation Problem? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.imsl.com/blog/solving-transportation-problem>

25. Transportation Problem Explained and how to solve it? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mygreatlearning.com/blog/transportation-problem-explained/>
26. 5 Transportation Issues And Challenges Facing US Cities And Their Solutions [Электронный ресурс] – Режим доступа до ресурсу: <https://www.remix.com/blog/5-transportation-challenges-facing-us-cities>
27. Transport problem [Электронный ресурс] – Режим доступа до ресурсу: https://encyclopediaofmath.org/wiki/Transport_problem
28. Dart Programming Tutorial [Электронный ресурс] – Режим доступа до ресурсу: https://www.tutorialspoint.com/dart_programming/index.htm
29. What is Dart Programming [Электронный ресурс] – Режим доступа до ресурсу: <https://www.javatpoint.com/flutter-dart-programming>
30. Dart [Электронный ресурс] – Режим доступа до ресурсу: <https://developers.google.com/learn/topics/dart>
31. ФРЕЙМБОРК FLUTTER [Электронный ресурс] – Режим доступа до ресурсу: <https://avada-media.ua/ua/services/flutter/>
32. Announcing Flutter Forward [Электронный ресурс] – Режим доступа до ресурсу: <https://medium.com/flutter/announcing-flutter-forward-666f9a506cc6>
33. Flutter & Dart - The Complete Guide [2023 Edition] [Электронный ресурс] – Режим доступа до ресурсу: <https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/>
34. The Good and the Bad of Flutter App Development [Электронный ресурс] – Режим доступа до ресурсу: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-flutter-app-development/>
35. Widget catalog [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.flutter.dev/development/ui/widgets>
36. Add Firebase to your Flutter app [Электронный ресурс] – Режим доступа до ресурсу: <https://firebase.google.com/docs/flutter/setup?platform=ios>

37. Flutter Clean Architecture Series — Part 1 [Електронний ресурс] – Режим доступу до ресурсу: <https://devmuaz.medium.com/flutter-clean-architecture-series-part-1-d2d4c2e75c47>
38. Futures – Isolates – Event Loop [Електронний ресурс] – Режим доступу до ресурсу: <https://www.didierboelens.com/2019/01/futures-isolates-event-loop/>
39. Flutter vs. React Native: Which is the right cross-platform framework for you? [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.blog/2022/10/31/comparing-frameworks-for-cross-platform-apps-flutter-vs-react-native/>
40. What the Flutter [Електронний ресурс] – Режим доступу до ресурсу: <https://flutter.wtf/>