

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ  
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

**Кваліфікаційна робота  
магістра**

на тему ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ  
РІШЕНЬ КОМБІНОВАНОЇ ЗАДАЧІ ТРАНСПОРТНОЇ ЛОГІСТИКИ

Виконав: студент II курсу, групи 8.0512-ек  
спеціальності 051 «Економіка»  
освітньої програми «Економічна кібернетика»  
Зарубін Олексій Станіславович  
Керівник проф. каф. ек. кіб., д.ф.-м.н., професор  
Козін І.В.  
Рецензент доцент каф. ек. кіб., к.е.н., доцент  
Лось В. О.

Запоріжжя  
2023



## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	д.е.н. проф. Козін І.В.	18.09.2023	27.09.2023
Розділ 2	д.е.н. проф. Козін І.В.	28.09.2023	25.10.2023
Розділ 3	д.е.н. проф. Козін І.В.	26.10.2023	22.11.2023

7. Дата видачі завдання 18.09.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Формування мети та завдань кваліфікаційної роботи	01.09–05.09.2022	виконано
2.	Складання плану роботи	06.09–12.09.2022	виконано
3.	Підготовка I розділу	13.09–26.09.2022	виконано
4.	Підготовка II розділу	27.09–17.10.2022	виконано
5.	Підготовка III розділу	18.10–17.11.2022	виконано
6.	Оформлення висновків та роботи	18.11–25.11.2022	виконано
7.	Підготовка до захисту і нормоконтролю	26.11–02.12.2022	виконано

Студент

\_\_\_\_\_

(підпис)

Зарубін О.С.

Керівник роботи (проекту)

\_\_\_\_\_

(підпис)

Козін І.В.

**Нормоконтроль пройдено**

Нормоконтролер

\_\_\_\_\_

Макаренко О.І

## РЕФЕРАТ

Кваліфікаційна робота магістра містить три розділи, 84 с., 13 рис., 9 табл., 57 джерела.

Об'єкт дослідження – задачі транспортної логістики

Предмет дослідження – моделі і методи розв'язання комбінованих задач транспортної логістики.

Мета роботи – інформаційна система підтримки прийняття рішень комбінованих логістичних задач.

Методи дослідження – порівняльний, статистичний, графічний, математичне моделювання, експеримент.

У роботі досліджено важливі аспекти задач транспортної логістики. Зокрема, проведено аналіз класифікації задач транспортної логістики, визначено основні проблеми, що виникають у цьому сегменті. Описано поняття "комбінована логістична задача".

За допомогою порівняльного аналізу розглянуто ефективність різних моделей і методів розв'язання задач транспортної логістики.

За допомогою математичного моделювання було створені моделі транспортних логістичних задач для подальшого аналізу. Досліджено використання методу "Розділяй і володарюй" для вирішення комбінованих задач, шляхом розбиття складних задач на підзадачі.

В результаті аналізу побудовано концептуальну математичну модель з розв'язку комбінованих логістичних задач. На основі моделі розроблено інформаційну систему підтримки прийняття рішень. Ця система має потенціал оптимізувати роботу пошуку оптимального розв'язку логістичних задач.

ІНФОРМАЦІЙНА СИСТЕМА, МАТЕМАТИЧНА МОДЕЛЬ,  
ЛОГІСТИЧНА ЗАДАЧА, КОМБІНОВАНА ЗАДАЧА, ПОШУК  
ОПТИМАЛЬНОГО МАРШРУТУ.

## SUMMARY

The qualification work of the master contains three sections, 84 chapters, 13 figures, 9 tables, 57 sources.

The object of research is the problems of transport logistics

Subject of research - models and methods of solving combined problems of transport logistics.

The purpose of the work is an information system to support decision-making of combined logistics tasks.

Research methods - comparative, statistical, graphic, mathematical modeling, experiment.

The work examines important aspects of transport logistics problems. In particular, an analysis of the classification of transport logistics tasks was carried out, and the main problems arising in this segment were determined. The concept of "combined logistics task" is described.

With the help of comparative analysis, the effectiveness of various models and methods of solving transport logistics problems was considered.

With the help of mathematical modeling, models of transport logistics problems were created for further analysis. The use of the "Divide and Conquer" method for solving combined problems by dividing complex problems into sub-problems has been studied.

As a result of the analysis, a conceptual mathematical model for solving combined logistics problems was built. Based on the model, an information system for decision support was developed. This system has the potential to optimize the search for the optimal solution to logistics problems.

INFORMATION SYSTEM, MATHEMATICAL MODEL, LOGISTICS PROBLEM, COMBINED PROBLEM, SEARCH OF THE OPTIMAL ROUTE.

## ЗМІСТ

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	
РЕФЕРАТ	
SUMMARY	
ВСТУП .....	
РОЗДІЛ 1 ОГЛЯД ЛОГІСТИЧНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМ.....	10
1.1 Класифікація задач логістики .....	10
1.2 Логістичні транспортні задачі додатковими обмеженнями .....	13
1.3 Аналіз існуючих інформаційних систем підтримки прийняття рішень задач комбінованої транспортної логістики.....	
РОЗДІЛ 2 МЕТОДИ, АЛГОРИТМИ РОЗВ’ЯЗКУ ЗАДАЧ ТРАНСПОРТНОЇ ЛОГІСТИКИ, МОДЕЛІ ПРИЙНЯТТЯ РІШЕНЬ .....	38
2.1 Метод «Розділяй і володарюй» для розв’язання складних задач .....	38
2.2. Методи та моделі розв’язку логістичних транспортних задач .....	40
2.3. Моделі транспортних логістичних задач .....	52
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВ’ЯЗКУ СКЛАДНИХ ЗАДАЧ ТРАНСПОРТНОЇ ЛОГІСТИК.....	55
3.1. Побудова моделі розв’язку комбінованих логістичних задач .....	55
3.2. Реалізація інформаційної системи підтримки прийняття рішень.....	56
3.3 Оцінка ефективності та подальшого розвитку інформаційної системи.	65
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	76
ДОДАТОК А Огляд інтерфейсу інформаційної системи PLS .....	81
ДОДАТОК Б Алгоритм роботи інформаційної системи PLS .....	82
ДОДАТОК В Алгоритм роботи інформаційної системи .....	83
ДОДАТОК Г Вхідні данні для аналізу ефективності інформаційної системи	84
ДОДАТОК Д Результат проходження Unichек.....	85

## ВСТУП

Тема вирішення транспортних логістичних задач є надзвичайно актуальною у сучасному світі. Велика кількість наукових конкурсів та робіт, що присвячені цій темі, свідчать про активний інтерес в цьому напрямку.

Логістика, яка включає в себе такі основні елементи, як: пошук оптимального маршруту, розклад маршрутизації транспорту та пакування товару, є ключовим елементом функціонування будь-якого підприємства.

Необхідність вирішення логістичних задач з урахуванням всіх трьох елементів одночасно виникають все частіше. Сучасні логістичні інформаційні системи, в тому числі: WMS, TMS, CRM, SRM. Не можуть задовольнити потреби в комплексних підходах, що дозволять оптимізувати логістичні процеси, враховуючи велику кількість факторів і аспектів.

Об'єктом дослідження виступає логістична задача оптимізації транспортних процесів, яка включає в себе важливі аспекти та вимагає комплексного підходу для ефективного вирішення.

Предметом дослідження є існуючі моделі і методи розв'язання комбінованих задач транспортної логістики.

Мета роботи полягає у розробці інформаційної системи з розв'язку комбінованих логістичних задач.

Для висвітлення даної теми було поставлено наступні задачі:

- дослідити логістичні задачі, дати поняття «комбіновані задачі транспортної логістики»;
- проаналізувати моделі та інформаційні системи з вирішення комбінованих логістичних задач;
- побудувати модель рішень комбінованих логістичних задач на основі методу «Розділяй і володарюй»;
- реалізувати на основі запропонованої моделі інформаційну систему підтримки прийняття рішень комбінованої задачі транспортної логістики.

У роботі використані наступні методи дослідження: порівняльний, за допомогою цього методу було досліджено різні аспекти логістичних задач із застосуванням комбінованих підходів; експериментом було визначено ефективність запропонованої моделі та інформаційної системи; математичне моделювання дозволило формалізувати логістичні задачі та розв'язувати їх за допомогою алгоритмів.

В результаті було отримано наступні наукові результати:

- вперше отримано модель з розв'язку комбінованих логістичних задач на основі методу «Розділяй і володарюй»
- дістало подальший розвиток інформаційна модель з вирішення комбінованих логістичних задач;
- удосконалено визначення термінології задач логістики.



## РОЗДІЛ 1

### ОГЛЯД ЛОГІСТИЧНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

#### 1.1 Класифікація задач логістики

У сучасному науковому контексті, поняття "транспортна логістика" не обмежується єдиним чітким визначенням. Напроти, відзначається значною термінологічною різноманітністю, зумовленою різними підходами та акцентами, вживаними в різних наукових працях. Це обумовлено складністю самої транспортної логістики, яка є суттєвим і розгалуженим компонентом загальної системи логістики.

Транспортна логістика може бути трактована як масштабна галузь логістики, що вивчає різні аспекти планування, управління та оптимізації транспортних процесів у контексті постачання товарів або послуг. Залежно від наукового дослідника і специфіки його досліджень, визначення терміну може варіюватися від акценту на ефективності транспортної інфраструктури до аспектів управління логістичними ланцюгами, охоплюючи весь життєвий цикл транспортних операцій. У таблиці 1.1 наведено визначення транспортної логістики різних авторів.

Таблиця 1.1 - Визначення поняття «транспортна логістика» в інших наукових джерелах

Автор	Визначення
Сокур І.М.	Організація та здійснення процесу доставки товару з мінімальними витратами, максимальною цілісністю у чітко визначений термін [1].
Смирнов І.Г., Косарева Т.В.	Розділ логістики руху (покою) ресурсів, що розглядає управління фізичним переміщенням матеріальних ресурсів у просторі та часі відповідно до інтересів їхніх споживачів [2].
Пономарьова Ю.В	Логістика, яка вирішує комплекс задач, пов'язаних з організацією переміщення вантажів транспортом загального користування [3].

Отже, можна запропонувати наступне визначення: транспортна логістика – це функціональна область логістики, пов'язана з управлінням

процесами переміщення потоків вантажів і пасажирів необхідної кількості, у необхідне місце, оптимальним маршрутом, за необхідний час, з найменшими витратами та відповідним інформаційним забезпеченням.

У сфері транспортної логістики виявляється значна розмаїтість задач, кожна з яких характеризується унікальним набором умов і факторів, обумовлених особливостями сфери діяльності та характеристиками транспортних засобів. Однак, незважаючи на цю різноманітність, усі ці задачі мають загальну основу.

Серед різних варіантів задач логістики, задача маршрутизації транспорту займає центральне положення і виступає як основа для багатьох інших варіантів задач цього типу.

У розгляді стандартної задачі проблеми маршрутизації транспортного засобу припускається наявність однорідного парку транспортних засобів. Зазначені транспортні засоби призначені для обслуговування групи клієнтів, розташованих у межах одного депо. Усі транспортні засоби обладнані фіксованою потужністю, яка підлягає строгому обмеженню, і кожен клієнт має відомий обсяг замовлення, що повинені бути відвантажені. З метою оптимізації ефективності, кожен клієнт повинен бути обслугований за один виїзд транспортного засобу, а кожен транспортний засіб повинен виїхати та повернутися до відповідного депо.

Довжина маршруту підлягає обмеженням, що регулює максимальну відстань, яку може пройти кожен транспортний засіб. Основна мета полягає в розробці оптимальної послідовності доставки для кожного автомобіля з метою задоволення всіх клієнтів та мінімізації загальної пройденої автопарком відстані [4].

Проблема полягає в розробці маршрутів для цих транспортних засобів з урахуванням певних умов і обмежень, а саме:

- а) необхідно задовольняти вимоги всіх клієнтів;
- б) важливо не порушувати вантажопідйомність транспортних засобів;

в) треба враховувати загальний час (або відстань), необхідний кожному транспортному засобу для прокладання свого маршруту.

Загалом ціль рішення можна сформулювати як мінімізацію загальної вартістю доставки, а саме сума витрат, пов'язаних з розміром автопарку та витрати на доставку. Супутні проблеми можуть бути сформульовані наступним чином:

- якщо автопарк складається з одного транспортного засобу досить великої місткості, і якщо обмеження (в) ігнорується, знайти найкоротший тур (за часом або відстань), щоб відвідати всіх клієнтів;
- якщо обмеження (в) ігнорується, знайти найменшу кількість транспортних засобів, які можуть бути необхідними;
- для найменшої можливої кількості транспортних засобів, що відповідають умовам (а), (б) та (в), сплануйте маршрути транспортних засобів таким чином, щоб загальна відстань була зведена до мінімуму;
- якщо ви можете вибрати транспортні засоби різної місткості, знайдіть оптимальне поєднання транспортних засобів, так що загальна вартість доставки зведена до мінімуму;

Ці варіації на одну й ту саму тему підкреслюють численні аспекти керування роботою автопарків, що використовуються для доставки та інших логістичних цілей. [5, 6].

Хоча пошук шляху є фундаментальним завданням у логістиці, сучасні підприємства стикаються з набагато складнішими та різноманітнішими викликами. Логістика, як ключовий елемент успішної діяльності підприємств, є ширшим спектром завдань, ніж просто пошук оптимального маршруту.

Ці завдання можна класифікувати наступним чином:

- пошук шляху: визначає ефективність та швидкість доставки товарів;
- розклад: включає оптимізацію часу і маршрутів, щоб зробити поставки більш ефективними і своєчасними.

-упакування: не лише важливий аспект збереження товарів, а й оптимізація обсягів вантажів для максимального завантаження транспорту.

Сучасні підприємства стикаються з тим, що чисті завдання логістики не завжди відображають їхню реальну мету. Саме тому ми звертаємо увагу на концепцію "комбінованих завдань логістики".

У більшості випадків завдання пошуку шляху, розкладу та упакування переплітаються, створюючи складні, взаємозалежні системи.

В якості прикладу розглянемо задачу з міжнародного конкурсу ROADEF/EURO Challenge 2022 [7], організованим Французьким товариством операційних досліджень (OR) за підтримки прийняття рішень (ROADEF) спільно з Асоціацією європейських товариств операційних досліджень (EURO). Посвячений 3D-завданню оптимізації завантаження вантажівок у співпраці з Групою Renault.

У цій задачі представлені елементи кожного представленого класу. Елементи комбінованої задачі наведено у таблиці 1.2.

Таблиця 1.2 – Елементи комбінованої задачі

Задача	Опис задачі
Пошук шляху	Знаходження оптимального варіанту для розміщення товарів у вантажівках з метою мінімізації кількості використовуваних вантажівок та запасів на заводах.
Пакування	Упаковка набору товарів від постачальників у стопки та подальше упакування цих стопок у вантажівки для ефективної та мінімізації витрат.
Формування розкладу	Створення оптимального розкладу доставки товарів на заводи, враховуючи тимчасове вікно (найраніший час прибуття, найпізніший час прибуття) та вартість.

## 1.2 Логістичні транспортні задачі додатковими обмеженнями

У більшості випадків, виникають ситуації, коли до логістичних задач додаються додаткові умови, які ускладнюють процес прийняття рішень. Ці додаткові умови можуть бути різноманітними: врахування обмежень на

робочий час, необхідність забезпечення стійкості до змін у середовищі, урахування екологічних аспектів, або інші фактори, що впливають на процеси постачання та розподілу.

Задача вибору маршрутизації транспорту з часовими вікнами є розширенням добре відомої задачі маршрутизації транспорту з центральним депо. Мета полягає в тому, щоб спроектувати оптимальний набір маршрутів, який обслуговує всіх клієнтів та задовольняє заданим обмеженням, особливо обмеженням тимчасового вікна.

Вхідні дані у задачі вибору маршрутизації транспорту з часовими вікнами складаються з  $n$  місць,  $n-1$  клієнтів, матриці  $D$ , що задає відстань для переміщення між кожною парою розташування, величини  $q_i$ , як визначає потребу в деякому ресурсі для кожного клієнта  $i$ , та максимальної кількості  $Q$ , ресурсу, який може перевозити транспортне засіб. Крім того, для окремого вузла вказується відрізок часу  $s_i$ , що означає час, необхідний для обслуговування клієнта  $i$ , та тимчасове вікно  $[t_i, T_i]$ , де  $t_i < T_i$  - період протягом якого повинно початися постачання.

Транспортному засобу дозволяється прибути до клієнта до початку тимчасового вікна, але воно має дочекатися «відкриття» вікна, щоб здійснити доставку. Доставка не може розпочатися після закриття тимчасового вікна. Таким чином, завдання реалізації розглядає обмеження тимчасового вікна як жорсткого обмеження. Також, існує обмеження  $V$  на кількість доступних транспортних засобів.

Здійснена рішення для VRPTW складається з набору  $V$  або менше маршрутів, які починаються і закінчуються в депо, таким чином, що кожен клієнт відвівається рівно по одному маршруту протягом зазначеного годинникового вікна, а загальний попит, призначений маршруту, не перевищує місткість транспортного засобу  $Q$ .

Оптимальне рішення для цієї задачі мінімізує загальне комбіновану відстань усіх маршрутів [4, 8].

Задача маршрутизації запасів є однією з найскладніших різновидів задач маршрутизації транспорту, оскільки вона об'єднує в одне завдання як управління запасами, так і рішення щодо маршрутизації.

Вхідні дані для такої задачі складаються з місць для депо та набору з  $n$  клієнтів  $\{1, \dots, n\}$ , часового горизонту, що складається з  $T$  періодів години  $\{1, \dots, T\}$ , та парку з  $M$  транспортних засобів  $\{1, \dots, M\}$ . Депо розташоване на вузлі 0. транспортне засіб має місткість  $Q$ .

Кожний період часу  $t$ ,  $t \in \{0, \dots, T\}$  одиниць ресурсу доступні на складі, і його одиниць споживається клієнтом  $i$ . Кожен вузол  $i \in \{0, \dots, n\}$  починається з початкового рівня інвентаризації, дорівнює  $I_i^0$ . Кожний клієнт  $i \in \{1, \dots, n\}$  повинен підтримувати рівень запасів не менше  $L_i$ , і не більше  $U_i$  та нести витрати за період у розмірі  $h_i$  на кожен одиницю наявних запасів. Для складу не визначено максимальний рівень запасів, але він не може опускатися нижче за 0, а вартість зберігання запасів за період дорівнює  $h_0$ .

Задача маршрутизації запасів є визначення для кожного періоду години  $t$  кількості, яка повинна бути доставлена кожному клієнту  $i$ , та маршрутів, за якими він повинен обслуговувати цих клієнтів. Оптимальним рішенням є те, що мінімізує загальні витрати, гарантуючи, що обмеження щодо місткості транспортних засобів та рівнів запасів вузлів завжди дотримуються. Загальна вартість рішення включає як загальні витрати на зберігання запасів на всіх вузлах так і витрати на маршрути транспортних засобів. Рішення повинно гарантувати, що:

- склад має достатній ресурс для забезпечення поставок у кожному періоді;
- кожен маршрут починається та закінчується в депо з урахуванням обмежень щодо місткості транспортних засобів;
- кожен період години клієнт отримує не більше однієї доставки.

Це так звана політика поповнення максимального рівня запасів, за якою клієнтам може бути доставлено будь-яке кількість за умови, що максимальний рівень запасів  $U_i$  не перевищує мінімальний рівень запасів  $L_i$  [4, 9].

Задача маршрутизації транспорту з роздільними поставками - це пом'якшення задачі маршрутизації транспорту, в якій дозволяється відвідувати кожного клієнта більше одного разу.

Вхідні складаються з місць для депо та набору з  $n$  клієнтів, матриці  $D$ , що задає відстань для проїзду між кожною парою місць розташування, кількості  $q_i$ , яка визначає попит на певний ресурс для кожного клієнта  $i$ , та максимальної кількості ресурсу  $Q$ , що може перевозити транспортне засіб .

Задача маршрутизації транспорту з роздільними поставками скасовує вимогу про те, що кожен клієнт відвівається лише один раз. Таким чином, здійсненим рішенням - набір маршрутів, які починаються і закінчуються в депо разом із сумами  $q_{ik}$  та визначають величину попиту зі сторони клієнта  $i$  на маршрут  $k$ , таким чином, щоб загальна кількість, доставлена  $i$  по всіх маршрутах  $k$  дорівнювала попиту  $i$ , таким чином, щоб загальний транспорт призначений кожному маршруту, не перевищував місткість  $Q$  [4].

Електромобілі є однією з найперспективніших технологій для скорочення викидів парникових газів у логістичних та транспортних системах. Однак вони страждають від ряду технічних обмежень, до яких їх аналоги з двигуном внутрішнього згоряння несприйнятливі, наприклад: тривала година зарядки, обмежена автономність.

Використання електромобілів створює додаткові обмеження в логістичних задачах. Проблему можна сформулювати наступним чином:

Нехай  $I$  – набір клієнтів, яких необхідно обслужити, а  $F$  – набір зарядних станцій, на яких транспортні засоби можуть зупинятися для підзарядки акумуляторів. У кожного клієнта  $i \in I$  є година обслуговування  $g_i$ . Клієнти обслуговуються за допомогою однорідного парку електромобілів. Кожен електромобіль має батарею ємністю  $Q$ .

На початку горизонту планування електромобілі перебувають у єдиному депо, з якого вони виїжджають повністю зарядженими. Депо працює безперервно протягом  $T$ max годин. При переміщенні з одного розташування  $i$  в інше місце  $j$  годину у дорозі  $t_{ij} \geq 0$  та споживання енергії  $e_{ij} \geq 0$ . Через

обмежену ємності акумулятора електромобілям, можливо, доведеться зупинятися біля зарядної станції. Операція зарядки може відбуватися на будь-якій станції. Електромобіль може бути частково зарядженим.

Кожний  $CS_j \in F$  має кусково-лінійну увігнуту функцію зарядки  $\varphi_j(\Delta)$ , яка відображає для розрядженої батареї годину, витрачену  $\Delta$  на заряд при  $j$ , стан заряду автомобіля при виїзді із  $j$ . Наприклад, якщо  $q$  - SoC електромобіля після прибуття в  $j$ , а  $\Delta$  - година зарядки, то SoC електромобіля при виході з  $j$  визначається як  $\varphi_j(\Delta + \varphi_{j-1}(q))$ .

Рішення для такої задачі повинно відповідати умові, що кожен маршрут є енергетично здійсненним, тобто заряд акумулятора електромобіля, коли він прибуває в будь-яке місце знаходиться в діапазоні від 0 до  $Q$ .

Метою є мінімізація часу, необхідного для обслуговування всіх клієнтів, включаючи час на маршрутизацію та зарядку [4].

Завдання дугового трасування визначаються в мережі, що перебуває з вузлів та з'єднань між вузлами. Ці з'єднання називаються дугами, коли вони спрямовані з одного вузла в інший, та ребрами - коли вони ненаправлені.

Категорія завдань дугового трасування традиційно була відокремлена від дослідження маршрутів транспортних засобів через відмінності в математичній формулюванні, а також відмінностей у методології для канонічних варіантів завдань з одним транспортним засобом. Наприклад, завдання китайського листоноші, що може бути вирішене за поліноміальну годину, в той час як завдання комівояжера є NP-складною.

В даний час більшість завдань дугового трасування з кількома транспортними засобами, обмеженнями пропускної здібності та запитами на обслуговування вирішуються за допомогою математичного програмування та метаевристичних методів, які аналогічні тим, які використовуються для них аналогів для маршрутизації транспортних засобів. Це призвело нас до того, що ми інтегрували цей помітний і важливіший варіант у завдання.

Задача дугового трасування тісно пов'язана з характеристиками дорожніх мереж, на яких вони визначені. Насправді, можна стверджувати, що



самі завдання маршрутизації транспортних засобів насамперед визначаються дорожньою мережею, але їх переформування на повний граф з використанням відстаней найкоротшого шляху, зазвичай приховує цей аспект. Варіанти дугового трасування приймають в якості вхідних даних докладну мережу, а не матрицю відстаней.

Задача трасування ємної дуги сформульована для мережі, що складається з вузлів та ненаправлених ребер. Ребро  $i, j$  між вузлами  $i$  та  $j$  може бути використане для переміщення та має вартість  $d_{ij}$  для його обходу. Деякі ребра  $i, j$  вимагають попиту на сервісне кількість  $q_{ij} \geq 0$ . Транспортний засіб має пропускну здатність  $Q$ , яка може обслуговувати на маршруті. Завдання CARP полягає у знаходженні набору маршрутів транспортних засобів, поданих у вигляді послідовності тупикових та обслуговуваних дуг орієнтованих ребер, що починаються і закінчуються в депо, таким чином, що кожне ребро з ненульовим попитом обслуговується одно одним транспортним засобом, кожним транспортним засобом, що не перевищує його межі пропускну здібності  $Q$ , а загальна вартість зводиться до мінімуму.

Задача трасування краю дуги- це завдання дугового трасування, яке виникає у вузлах, ребрах чи дугах. Мережа, яка включає в себе вузли, дуги та ребра, будь-який з яких може мати попит на обслуговування. Усі ці потреби повинні задовольнятися за допомогою здійснюваного набору маршрутів [4].

Перевантаженість є основним фактором у міській логістиці. Проблеми з маршрутизацією, що залежать від часу, враховують зміни в часі в дорозі можуть виникнути внаслідок заторів, погодних умов, перекриття доріг та інших факторів. Проблеми дугового трасування, що залежать від години, не були добре вивчені в літературі, що дещо дивно, оскільки більшість пов'язаних з ними застосувань відбувається у міських умовах, де час у дорозі сильно залежить від заторів.

Залежна від час задача трасування дуги - це варіант трасування дуги, яка використовує вартість подорожі, що залежить від часу. Ця задача позиціонується в мережі, що складається з вузлів та ненаправлених ребер.

Ребро  $(i,j)$  між вузлами  $i$  та  $j$  може бути використане для переміщення. Ребро  $(i,j)$  пов'язане з відстанню  $d_{ij}$ , і для деяких ребер потрібно сервісне кількість  $q_{ij} \geq 0$ . Шкірне транспортне засіб має пропускну здатність  $Q$ , яка обмежує попит, який він може обслуговувати на маршруті. TDCARP полягає в знаходженні набору маршрутів транспортних засобів, поданих у вигляді послідовності тупикових та обслуговуваних дуг орієнтованих ребер, що починаються і закінчуються в депо, таким чином, що кожне ребро з ненульовим попитом обслуговується одно одним транспортним засобом, що обслуговується транспортним засобом, що не перевищує його межі пропускну можливості  $Q$ , а загальна година в дорозі зводиться до мінімуму.

Задача визначається за обрієм планування  $0, N$ . кожне неорієнтоване ребро  $(i,j)$  пов'язане з двома напрямленими дугами  $(i,j)$  і  $(j,i)$ , і кількість години, яка потрібна для проходження дуги, залежить від того, коли вона перетинається в межах горизонту планування. При заданому горизонті планування  $0, N$  ми пов'язуємо з дугою  $(i,j)$  постійну функцію швидкості з  $h_{ij}$  відрізків. Ця функція представляє швидкість руху, тобто відстань, пройдену в одиницю години за цією дугою як функцію години обходу. Відзначимо, що швидкість транспортного засобу може змінюватися під час руху по ланці, заснованій на цій функції [4].

Проблема виклику таксі полягає в тому, що центральний оператор керує однорідним парком транспортних засобів, які обслуговують випадково виникаючі запити на поїздки. При вступі до запиту оператор може назначити його транспортному засобу, відхилити або почекати з ухваленням рішення. Оператор також вибирає, де переміщати транспортні кошти, щоб вони могли краще обслуговувати майбутні запити. Оператор прагне максимізувати прибуток, отриманий за 24-річний період, що розраховується як дохід, отриманий від обслуговування запитів за вирахуванням понесених командування витрат [4].

Задачі розподілу, маршрутизації та складського управління, які є основними складовими логістики, часто потрапляють під клас НП. Це означає,

що для їхнього оптимального вирішення часто потрібно використовувати евристичні методи, які дають приблизні рішення, а не точні. Зокрема, генетичні алгоритми, метаевристика та інші евристичні підходи виявляються досить ефективними в розв'язанні таких задач.

Загалом, класифікація більшості задач логістики як задач класу НП свідчить про необхідність вдосконалення методів оптимізації та розвитку нових підходів для вирішення цих завдань, що є актуальним напрямком досліджень у сфері логістики та оптимізації бізнес-процесів [10].

### 1.3 Аналіз існуючих інформаційних систем підтримки прийняття рішень задач комбінованої транспортної логістики

Згідно А. Ф. Павленко та В. В. Кривещенко [11], логістична інформаційна система — це певним чином організована сукупність взаємопов'язаних засобів обчислювальної техніки, різних довідників і необхідних засобів програмування, що забезпечує вирішення тих або інших функціональних завдань з управління матеріальними потоками. Так само як і будь-яка інша система, інформаційна система складається з впорядковано взаємозалежних елементів та володіє сукупністю інтегрованих якостей.

Декомпозицію інформаційних систем на складові елементи можна здійснювати по-різному. Найчастіше інформаційні системи поділяють на дві підсистеми: функціональну і забезпечувальну.

Функціональна підсистема складається із сукупності розв'язуваних завдань згрупованих за ознакою спільності мети.

Забезпечувальна підсистема у свою чергу включає такі елементи:

- технічне забезпечення, тобто сукупність технічних засобів, які забезпечують обробку і передачу інформаційних потоків;
- інформаційне забезпечення, які містять у собі різні довідники, класифікатори, кодифікатори, засоби формалізованого опису даних;

- математичне забезпечення. Тобто сукупність методів вирішення функціональних завдань. Логістичні інформаційні системи, як правило, є автоматизованими системами управління логістичними процесами. Тобто математичне забезпечення в логістичних інформаційних системах — це комплекс програм і сукупність засобів програмування. Які забезпечують вирішення задач управління матеріальними потоками, обробку текстів.

Інформаційні системи в логістиці створюються з метою управління матеріальними потоками як на макро, так і на мікрорівні.

На рівні окремого підприємства інформаційні системи, у свою чергу, поділяють на три групи:

- планові;
- диспозитивні;
- виконавчі.

Логістичні інформаційні системи, які входять у різні групи, відрізняються як своїми функціональними, так і забезпечувальними підсистемами. Функціональні підсистеми відрізняються складом розв'язуваних завдань. Забезпечувальні підсистеми можуть відрізнитися всіма своїми елементами, тобто технічним, інформаційним і математичним забезпеченням. Зупинимось детальніше на специфіці інформаційних систем.

Планові інформаційні систем створюються на адміністративному рівні управління і служать для прийняття довгострокових рішень стратегічного характеру. Серед розв'язуваних завдань можуть бути такі:

- створення й оптимізація ланок логістичного ланцюжка;
- управління мало змінними даними;
- планування;
- загальне управління запасами;
- управління резервами та інші завдання.

У планових інформаційних системах найвищий рівень стандартизації під час вирішення задач, що дозволяє з найменшими труднощами адаптувати тут стандартне програмне забезпечення.

Диспозитивні інформаційні системи створюються на рівні управління складом і служать для забезпечення налагодженої роботи логістичних систем. Диспозитивні інформаційні системи вирішують наступні завдання:

- детальне управління запасами;
- керування внутрішнім транспортом;
- відбір, облік матеріальних ресурсів згідно до замовлень та їх комплектування

Як недолік диспозитивних інформаційних систем необхідно відмітити неможливість їх комбінування з іншими системами.

Виконавчі інформаційні системи створюються на рівні адміністративного або оперативного управління. Це так званий режим роботи в реальному масштабі часу, який дозволяє отримувати необхідну інформацію про переміщення вантажів у поточний момент часу і вчасно видавати відповідні адміністративні рішення. Як правило в виконавчих інформаційних системах застосовують індивідуальне програмне забезпечення.

Відповідно до концепції логістики інформаційні системи, які належать до різних груп, інтегруються в єдину інформаційну систему. Розрізняють вертикальну і горизонтальну інтеграцію.

Вертикальною інтеграцією вважається зв'язок між плановою, диспозитивною і виконавчою системами за допомогою вертикальних інформаційних потоків.

Горизонтальною інтеграцією вважається зв'язок між окремими комплексами завдань з диспозитивних і виконавчих системах за допомогою горизонтальних інформаційних потоків.

Інформаційні системи виконують роль функціональної технічної інфраструктури, яка забезпечує інтеграцію суб'єктів всього логістичного ланцюжка. Для вирішення зазначених задач розроблено низку інформаційних

систем. Головною особливістю і перевагою сучасних інформаційних систем типу WMS-системи (Warehouse Management Systems — WMS) є забезпечення автоматизації управління всіма бізнес-процесами у режимі реального часу, контроль за виконанням кожної операції, застосування технологій адресного зберігання, застосування штрих кодування планування, виконання вантажно-розвантажувальних операцій, ABC-аналіз, фіксування часу вантажно-розвантажувальних робіт, руху автотранспорту по території складського комплексу, можливість інтеграції з іншими системами, які використовуються підприємством

Класичні облікові складські інформаційні системи (в т. ч. і MRP/ ERP), орієнтовані на облік запасів матеріальних ресурсів у визначених місцях зберігання, що є недостатнім для ефективного управління бізнес-процесами логістичного ланцюга у сфері розподілу, у логістичних структурах типу 3PL і 4PL. Їх місце займають WMS-системи (Warehouse Management Systems — WMS), які забезпечують оперативне управління рухом матеріальних ресурсів, техніки і персоналу складу у режимі реального часу, можливість налагодження гнучкої системи технологій складування (адресне зберігання, проектуємі ячейки, віртуальний склад і т. п.), інтеграція on-line, управління завданнями і аналіз ефективності роботи персоналу, інтеграція з іншими управлінськими інформаційними системами. Це досягається завдяки WMS-системам (RFID — Radio Frequency Identification — Радіочастотна ідентифікація. Система автоматичної ідентифікації товарів за радіомітками; RF/DC — Radio Frequency/Data Communication — Мобільні без провідникові системи передачі даних по радіоканалу; DCC — Data Capture and Collection — Портативні комп'ютери для збору даних сканування міток. Мобільне робоче місце — BT — Bluetooth; WiFi — Wireless Fidelity; WLAN — Wireless Local Area Network; GSM/ GPRS — Безпроводні технології передачі даних і позиціонування, яке підтримується сучасними мобільними комп'ютерами типу Unitech, Intermec і ін.; VDT — Voice Direct Technologies — Технологія і засоби прямого голосового управління; WCS — Warehouse Control

Management — Система контролю товарів. Визначення ваги та габаритів матеріальних ресурсів, які надходять на зберігання/відвантаження CWS — Cubing and Weighing System; Компонент WCS-системи. Автоматичне визначення ваги і габаритних параметрів матеріальних ресурсів) сучасних технологій автоматичної ідентифікації і позиціонування товарів, техніки і операторів складу.

Вимоги до сучасних інформаційних систем, які забезпечують ефективне управління рухом, зберіганням, транспортування і ін., матеріальних ресурсів у логістичному ланцюжку обумовлені динамічністю розвитку навколишнього середовища. Функціональність інформаційних систем, які забезпечують ефективність логістичної діяльності повинні забезпечувати:

- універсальність до матеріального потоку;
- можливість планування наскрізної поставки через весь логістичний ланцюжок;
- функціональне проектування і оптимізація місць зберігання;
- оперативний контроль і управління у режимі реального часу;
- підтримка роботи з сучасним обладнанням;
- реалізація функцій поповнення запасів, черговість поставок, інвентаризація, робота персоналу;
- можливість інтеграції з іншими інформаційними системами;
- підтримка радіочастотних (RFID), голосових (VDT) технологій управління всіма складськими операціями.

Розглянемо інформаційну систему на основі математичної моделі ланцюга поставок для вантажних перевезень у рамках логістичної системи, яка була описана Ямковим М.В. [12].

Вступ та постановка проблеми: ланцюги поставок, це динамічна і складна система, що включає поведінку кількох учасників. Завдяки реалізації взаємодії між агентами, моделювання реалізується за допомогою інформаційних та матеріальних потоків. Моделювання ланцюга поставок широко застосовується у сфері управління логістичними операціями.

Організація ланцюга поставок є корисним інструментом для спостереження продуктивності загальної логістичної системи. Багато моделей використовують моделювання для прогнозування та дизайну відповідного ланцюга поставок для кожного конкретного підприємства. Крім того, моделювання ланцюга поставок використовується для вивчення економічних явищ в динаміці дії логістичної системи на підприємстві.

Вантажні агенти взаємодіють один з одним через ланцюг поставок, включаючи інформаційні та матеріальні потоки. Запропонована модель імітує діяльність агентів на всьому ланцюгу поставок враховуючи участь у придбанні продукції, виробництві та розподілу. Дії агентів, у тому числі роздрібною торгівлі, оптовиків, виробників, постачальників та перевізників, ґрунтуються на концепції мінімізації загальних витрат та підвищенні рівня продуктивності виробництва. Агент прогнозує попит та розглядає відповідні політики управління запасами, виробництво та розподіл продукції. Політики будуть періодично переглядатися відповідно до прогнозування попиту, щоб сформувати дійсний фактичний попит. З попиту кінцевих споживачів, моделюємо кожен день кількість продуктів, які продаються під час роздрібною торгівлі. На підставі фактичного продажу, агент формує рівень запасів наприкінці кожного дня. Потік інформації фіксується, коли виявлено факт, що рівень запасів досягнув максимальної позиції. Потім агент розміщує замовлення на верхньому рівні ланцюга поставок (відправник вантажу). Після того як відправник вантажу отримав замовлення, потік товару буде запущений; товари завантажуються на вантажівки та передаються замовникам. Під час розподілу продукції, поставки будуть розглядатися відповідно до транспортної політики агента відправника. Потім транспортний агент формує маршрут доставки з мінімізацією часу. Запропонована автором структура моделі наведена на рисунку 1.1.





Рисунок 1.1 - Структура запропонованої моделі [12]

Агенти сортуються за рівнем відповідно до положення у структурі ланцюга поставок. Наприклад, агент може бути 1 роздрібним; агенти 2 та 3 можуть бути виробником та постачальником відповідно. Для кожного агента, процес прийняття рішення складається з шести модулів для роботи з кожним із напрямків логістичної системи:

- модуль продажів;
- модуль виробництва;
- модуль рішення про купівлю;
- модуль інвентаризації;
- модуль транспортування;
- модуль зворотного зв'язку.

Модуль продажів: Математична модель ґрунтується на наступному: попит на товари кінцевих покупців на підприємстві роздрібної торгівлі слідує закону нормального розподілу. Кількість продажів ( $d_i^t$ ) кожного продавця  $i$  та час  $t$ , який, як передбачається слідує нормальному розподілу із середнім значенням продажів  $\bar{d}_i$  і стандартного відхилення  $\varepsilon_i$  моделюється за допомогою концепції Монте Карло.

$$d_i^t = N(\bar{d}_i, \varepsilon_i) \quad (1.1)$$

де  $d_i^t$  – модуль продажу;

$i$  – окремий продавець;

$t$  – час;

$\bar{d}_t$  – середній розподіл продажів;

$\varepsilon_i$  – стандартне відхилення.

Для інших агентів, попит залежить від замовлень, отриманих від агентів на нижньому рівні ланцюжка постачання

Модуль виробництва: Кожна фірма приймає рішення щодо регулювання виробництва, формуючи резерв товару, який буде вироблятися протягом місяця. Середньомісячна сума товарів, вироблених кожною фірмою, оцінюється з використанням методів регресії та залежить від характеристик фірм, площі, робітників, тощо:

$$G_i = f(x_{1i}, x_{2i}, \dots, x_{ki}) \quad (1.2)$$

де  $G_i$  – місячний обсяг виробництва товарів фірми  $i$ ;

$x_{ki}$  – характеристики / показники фірми  $i$ , такі як кількість співробітників, площа, та інші пов'язанні атрибути виробництва.

Модуль рішення про покупку: Фірма може придбати товар із багатьох джерел. Модуль прийняття рішення про купівлю визначає частку товарів, які бути придбані з кожного джерела надходжень. Купівельна фракція будується з трьох частин: розподіл вибору каналу, вибір місця розташування та вибір відправника вантажу.

Щоб вибрати тип промисловості, припустимо, що рішення фірми ґрунтується на привабливості кожного типу промисловості.

Мультиномінальна модель використовується формування поведінки головних змінних, якими в даному дослідженні виступають обсяг виробництва та кількість фірм цього типу промисловості / виробництва

$$P(c) = \frac{\exp(V_c)}{\sum_{c' \in \hat{C}} \exp(V_{c'})} \quad (1.3)$$

$$V_c = f(x_{1c}, x_{2c}, \dots, x_{kc}) \quad (1.4)$$

де  $P(c)$  – частка товару придбана на підприємстві типу  $C$ ;

$V_c$  – функція корисності / привабливості типу промисловості;

$x_{kc}$  – атрибути, що становлять привабливість підприємства типу  $C$ , такі як загальна сума прибутку від загального обсягу виробництва та числа фірм.

Вибір розташування  $P(\gamma|C)$  тобто місце розташування відправників вантажу, які будуть обрані, на основі привабливості зон. Вибір розташування визначається також за допомогою мультиноміальної логістичної моделі. Зональні змінні привабливості (такі як кількість фірм у зоні та рівень загального товарного виробництва в зоні) та зональні змінні опору (наприклад, відстань між зоною замовника та відправника) застосовуються для знаходження рівня привабливості зон:

$$P(\gamma|C) = \frac{\exp(V_\gamma)}{\sum_{\gamma' \in Z} \exp(V_{\gamma'})} \quad (1.5)$$

$$V_\gamma = f(x_{1\gamma}, x_{2\gamma}, \dots, x_{k\gamma}) \quad (1.6)$$

де  $P(\gamma|C)$  – частка товару типу  $C$  знаходиться в зоні  $\gamma$ ;

$V_\gamma$  – функція корисності зони  $\gamma$ ;

$x_{k\gamma}$  – атрибути, що становлять привабливість зони  $\gamma$ , такі як число фірм в зоні, загальне виробництво товару у зоні, відстань між клієнтами та вантажовідправником.

Вибір вантажовласника:  $P(i|C, \gamma)$  Аналогічним чином, відправник вантажу використовується для ідентифікації вантажовласника. Процес вибору рішення реалізується з використанням мультиномінальної логістичної моделі в залежності від привабливості відправника вантажу, тобто якості виробництва фірми:

$$P(i|C, \gamma) = \frac{\exp(V_i)}{\sum_{t \in i} \exp(V_t)} \quad (1.7)$$

$$V_i = f(x_{1i}, x_{2i}, \dots, x_{ki}) \quad (1.8)$$

де  $P(i|C, \gamma)$  – частка придбаного товару типу  $C$ , що знаходить у зоні  $\gamma$ ;

$V_i$  – функція корисності відправника вантажу  $i$ ;

$x_{ki}$  – атрибути привабливості відправника вантажу, залежить від загального обсягу виробництва.

Модуль інвентаризації є головним модулем у вирішенні питання активації інших модулів. Після того, як рівень запасів актуалізує точку замовлення, виробництво та закупівельні модулі активуються на придбання товарів. Аналогічно, коли замовлення розміщується, транспортний модуль запускається на постачання товарів до клієнтів. Кожна фірма приймає рішення щодо інвентаризації, за для постійного контролю рівня запасів на складу та виробництві. Завдання кожної фірми звести до мінімуму загальні витрати, у тому числі витрати на зберігання, замовлення та транспортні витрати. У реальній ситуації є кілька стратегій, пов'язаних з управлінням запасами і всі вони повинні бути розглянуті в аналізі. Однак, у рамках даного дослідження, передбачається, що порядок інвентаризації кожної фірми є фіксованою частотою. Оптимальна частота замовлення формується як:

$$F_{ij} = \sqrt{\frac{Q_{ij}}{\omega * D_{ij}}} \quad (1.9)$$

де  $F_{ij}$  – частота замовлення фірми;

$Q_{ij}$  – щомісячний товар куплений фірмою  $i$  у фірми  $j$ ;

$D_{ij}$  – відстань між фірмою  $i$  та фірмою  $j$ ;

$\omega$  – параметри інвентаризації.

Модуль транспортування: Щоразу, коли інформаційні потоки надходять від клієнтів до відправників вантажу, активується модуль транспортування, щоб розподілити товари від відправників до клієнтів. Модуль транспортування складається з двох модулів: транспортних засобів та маршруту. Коли відправник вантажу доставляє товари для клієнтів, по-перше, він обирає спосіб доставки власноруч або за допомогою експедитора компанії. Після прийняття рішення товари, завантажуються на вантажівки та доставляються до кожного клієнта відповідно до листа маршрутизації транспортного засобу.

Передбачається, що вантажовласник обирає носій та автомобіль враховуючи принцип мінімізації транспортних витрат. Вибір є структурованим у рамках логістичної моделі вибору носія та вибору транспортного засобу. Вибір рішення залежить від характеристик вантажовідправників та клієнтів (наприклад, типу промисловості та чисельності працівників), атрибутів товарів (наприклад, типу товару, розміру партії та частоти відправок) та інших характеристик, пов'язаних із товарами:

$$P_{ij}(LP) = \frac{\exp(V_p)}{1 + \exp(V_B)} * \frac{\exp(V_L + V_p^t * \mu_p)}{1 + \exp(V_L + V_p^t * \mu_p)} \quad (1.10)$$

$$V_p = f(Nc_i, Emp_i, L_{ij}, F_{ij}, TT_{ij}) \quad (1.11)$$

$$V_p^t = \ln[\exp(V_p) + \exp(V_B)] \quad (1.12)$$

де  $P_{ij}(LP)$  – ймовірність вибору великої кількості приватної вантажівки;

$V_p$  – функція корисності приватної вантажівки;

$V_B$  – функція корисності бізнес вантажівки;

$V_L$  – функція корисності великої вантажівки;

$Nc_i$  – число клієнтів для відправника вантажу  $i$ ;

$Emp_i$  – число працівників відправника вантажу  $i$ ;

$F_{ij}$  – частота доставки від відправника вантажу  $i$  до клієнта  $j$ ;

$TT_{ij}$  – час у дорозі від відправника вантажу  $i$  до клієнта  $j$ ;

$\mu_p$  – параметр масштабування для приватної вантажівки.

Автомобіль здійснює рух до клієнтів у відповідності до маршруту. Маршрут доставки зводиться до мінімуму загальний час маршруту, який повинен входити у часові рамки максимальних робочих годин водія вантажівки та обмеженої пропускної здатності. Загальний час у дорозі включає час перебування на місці замовника для паркування, завантаження і розвантаження товару. Математична модель може бути записана у вигляді:

$$TC_i = \sum_{m=1}^n \sum_{t=1}^n ((TT_{t,m} + ST_m) * x_{tm}) \quad (1.13)$$

$$\sum_{m=1}^n x_{tm} = 1 \quad (1.14)$$

$$\sum_{t=1}^n x_{tm} = 1 \quad (1.15)$$

$$\sum_{m=1}^n \sum_{t=1}^n ((TT_{t,m} + ST_m) * x_{tm}) \leq HR_{max} \quad (1.16)$$

$$\sum_{m=1}^n \sum_{t=1}^n (L_{tm} * x_{tm}) \leq WT_{max} \quad (1.17)$$

де  $TC_i$  – загальний час доставки вантажу у відповідність з маршрутом;

$TT_{im}$  – час у дорозі між клієнтом  $l$  і клієнтом  $m$ ;

$ST_m$  – час перебування к клієнта  $m$ ;

$n$  – кількість клієнтів;

$HR_{max}$  – максимальний робочий час водія;

$WT_{max}$  – максимальна пропускна здатність.

Реалізації всіх перерахованих вище стратегій реалізує зворотній зв'язок для агентів. Щоразу, коли фірми отримують замовлення, вони переглядають прогноз попиту та здійснюють рішення про виробництво, закупівлю, інвентаризацію та політику в галузі транспорту. Фірма прогнозує попит із метою перегляду політики інвентаризації. Варто припустити, що фірма використовує середню ковзну як метод прогнозування попиту. Передбачені зміни попиту, частота замовлення також регулюється, щоб задовольнити попит та як наслідок збільшити прибуток та мінімізувати простой [12].

Розглянемо інформаційну систему з розв'язання складних логістичних задач за допомогою мультиштучної інтелектуальної системи, представленою Жюльєном Брамель та Девідом Сімкі-Леві у роботі [13].

Інфраструктура та діаграма даних запропонованого ILS показані на малюнках 1a та 1b відповідно. В основному ILS складається з чотирьох основних модулів, а саме модуля розподіленого вирішення проблем DPSM, модуля визначення конфліктних параметрів CPDM, модуля уточнення рішення SRM і модуля об'єднання рішень SUM.

Розподілений модуль вирішення проблем DPSM — це модуль, який відповідає за надання логістичного рішення за допомогою методів розподіленого вирішення проблем. Він складається з двох підмодулів: модуль виявлення потенційних проблем із якістю RQPIM та модуль агентів розподілених рішень DSAM.

Принцип RQPIM полягає в тому, що він використовує технологію CBR для виявлення можливих проблем якості в робочому процесі логістики та розділення їх на кілька підпроблем на основі різних класів потенційних проблем, які спираються на попередній досвід у подібних випадках. Цей модуль призначає завдання для визначення потенційних проблем, тобто для отримання найбільш схожих випадків у базі випадків. Ці задачі розділені на кілька підпроблем, які базуються на розв'язанні, наданому в найбільш подібному випадку. Потім усі підпроблеми передаються відповідним агентам у DSAM.

iDSAM містить багатоагентну систему, яка реалізує архітектуру зв'язку на дошці. Кожен із агентів у цьому модулі здатний діяти автономно, спільно та ефективно, щоб забезпечити логістичне рішення високої якості та надійності. У розподіленій архітектурі зв'язку з дошкою дані передаються в межах спільноти агентів через публічний простір дошки.

Дошка — це місце, де агенти публікують запити про допомогу, а експерти відповідають на запити, тому агентам не потрібно безпосередньо взаємодіяти один з одним. Взаємодія між агентами та дошками набирається мовою керування знаннями та запитами KQML. Повідомлення для обміну інформацією не залежать від синтаксису й онтології вмісту.

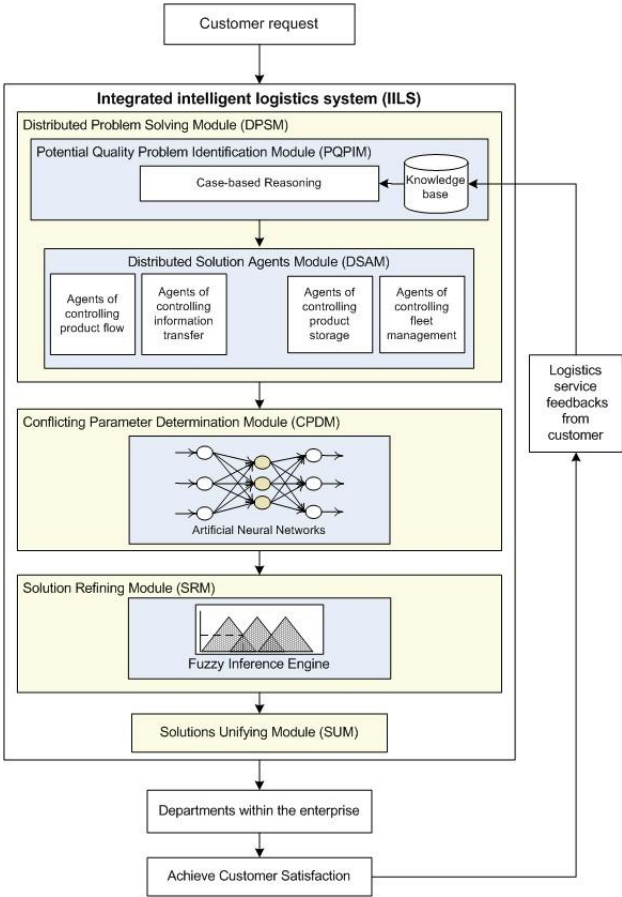
CPDM — це модуль, який гарантує, що якість рішень, створених DSAM, відповідає стандартам. Оскільки в DSAM використовується розподілене вирішення проблем, згенеровані локальні рішення не розглядаються глобально. Це означає, що локальні рішення співіснують із суперечливими параметрами, що може спричинити розбіжності та динамічні проблеми в робочому процесі логістики. Таким чином, CPDM розроблено для визначення наявності суперечливих параметрів у запропонованому логістичному рішенні. Технологія ШНМ вбудована в CPDM для ідентифікації конфлікуючих параметрів. Крім того, у цьому модулі застосовано регресію багатошарового перцептрона MLP із алгоритмом зворотного поширення.

SRM використовується для вдосконалення рішення, яке містить суперечливі параметри, визначені в CPDM. SRM використовує нечітку логіку для налаштування деяких параметрів, щоб усунути конфлікти між локальними рішеннями. Ці параметри є вхідними значеннями системи, тому вхідні значення функції приналежності можна отримати за допомогою фазифікації. Тоді вихідні значення членства можуть бути отримані за допомогою набору правил IF-THEN. На етапі дефазифікації вихідне значення системи може бути отримано за допомогою вибраного методу дефазифікації. Нарешті, ще один модуль під назвою SUM об'єднає всі рішення, розроблені різними агентами, і розповсюдить їх у відповідних відділах підприємства.



Хоча логістична система зворотного зв'язку не включена в цю структуру, це канал, за допомогою якого клієнти можуть надавати відгуки та коментарі щодо послуг постачальникам логістичних послуг. Клієнти можуть оцінювати логістичні рішення та скаржитися на ці рішення. Найбільш очевидною функцією є допомога у виявленні повторюваних проблем із обслуговуванням, які не зберігаються в базі знань. Нова проблема послуги буде сформульована для нового випадку, який буде проіндексовано та переміщено в базу знань.

Ця інфраструктура дозволяє краще використовувати функції розподіленого вирішення проблем завдяки колективним зусиллям кількох спеціалістів, які вирішують проблеми, щоб об'єднати свої знання, інформацію та можливості, щоб знайти рішення проблем із якістю, кожен з яких неможливо вирішити окремо. Інфраструктура інформаційної системи наведена на рисунку 1.2.



Рисуюнок 1.2 - Інфраструктура інформаційної системи CPDM [13]

Після того, як CPDM надасть серію суперечливих параметрів, як конфліктуючі, так і кореляційні параметри будуть передані до SRM для налаштування. Цей модуль забезпечує підтримку прийняття рішень щодо оптимізації продуктивності, щоб переконатися, що вона прийнятна з бажаними вимірюваннями. Крім того, він уточнює рішення шляхом коригування деяких параметрів для усунення конфліктів між локальними рішеннями в багатозначній нечіткій логіки. Механізм уточнення рішень вбудовано в цей модуль, який складається з нечіткого набору, нечіткого правила, нечіткого висновку та бази знань. База знань містить знання предметної області, які включають низку правил, що зберігаються в об'єктно-орієнтованій структурі. Він використовується для якісного вирішення проблем і корекційних дій, коли виникає проблема продуктивності. Розуміючи правила вимог замовника та застосовуючи їх для перетворення байтів даних в інформацію, можна досягти рішення покращеної якості. Як правило, кожне правило визначає відношення, рекомендацію, директиву, стратегію або евристику та має структуру ЯКЩО умова ТО дія. Коли умовна частина правила задовольняється, правило спрацьовує, а діюча частина виконується.

Фазифікація є першим кроком, який необхідно виконати в процесі нечіткого висновку. Після отримання чітких вхідних значень із запропонованого рішення продуктивності в PDM, вони будуть розмиті для всіх визначених функцій належності. Правило буде оцінено шляхом отримання нечітких вхідних даних і застосування їх до нечітких правил, що зберігаються в базі знань. Нечіткі правила зберігаються та визначаються як умовні оператори у формі IF-THEN, напр. ЯКЩО площа розміщення середня І кількість вхідних матеріалів велика, ТОДІ час крос-докінгу трохи довший. Якщо задане нечітке правило має кілька попередніх, нечіткий оператор AND або OR буде використано для отримання єдиного числа, яке представляє результат оцінки попереднього. Це число потім застосовується до послідовної функції належності.

Агрегація буде виконана для уніфікації результатів усіх правил. Функції належності всіх наслідків правил попередньо обрізаються або масштабуються та об'єднуються в один нечіткий набір. Як наслідок, входом процесу агрегування є список обрізаних або масштабованих послідовних функцій належності, а виходом є один нечіткий набір кожної вихідної змінної

Дефазифікація, яка є останнім кроком у нечіткому логічному висновку, є операцією декодування, яка створює єдине чітке значення на виході. Не існує унікального способу виконання дефазифікації. Центр ваги COG є найпопулярнішим методом, який використовується в цьому дослідженні. Теоретично COG обчислюється за континуумом точок у функції приналежності агрегованого виходу, але на практиці розумну оцінку можна отримати, обчисливши її за вибіркою точок, як показано: (Negnevitsky, 2005).ii

Рисунок 1.3 ілюструє приклад процесу SUM, який є модулем для об'єднання локальних рішень, які надсилаються з PQPIM і SRM.

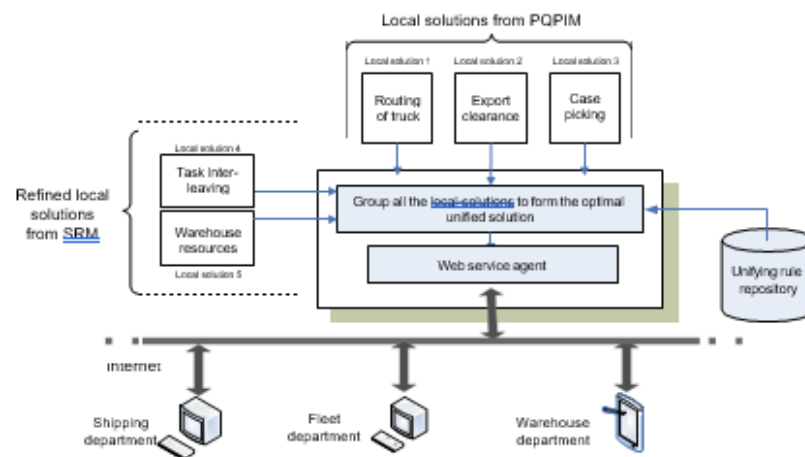


Рисунок 1.3 - Приклад процесу SUM в інформаційні системі CPDM [13]

Зокрема, локальні рішення від SRM є вдосконаленими рішеннями, а параметри налаштовані для запобігання конфліктним діям, що відбуваються в логістичній операції. Ці локальні рішення приходять до набору уніфікованих рішень для розв'язання вихідної загальної проблеми якості та уніфікуються відповідно до правил уніфікації рішень, які зберігаються в репозиторії

уніфікуючих правил. Цей репозиторій містить формат для уніфікації рішення в найбільш відповідній послідовності та комбінації. Послідовність впливає на тип дії, яка буде ініційована в певний час, і деякі дії виконуватимуться одночасно в різних відділах, щоб досягти найкращої продуктивності служби. Потім уніфіковане рішення буде перетворено у формат XML і надіслано до відповідних відділів через агента веб-сервісу [13].

## РОЗДІЛ 2

### МЕТОДИ, АЛГОРИТМИ РОЗВ'ЯЗКУ ЗАДАЧ ТРАНСПОРТНОЇ ЛОГІСТИКИ, МОДЕЛІ ПРИЙНЯТТЯ РІШЕНЬ

#### 2.1 Метод «Розділяй і володарюй» для розв'язання складних задач

"Розділяй і володарюй" - це метод вирішення складних задач, запропонований Гай Юлієм Цезарем, відомим римським полководцем і стратегом. Цей принцип визначається тим, що розділення великої проблеми на менші, більш керовані частини робить управління і вирішення завдань більш ефективними. Цезар вперше використовував цей підхід під час Галльських війн, коли він ділив свою армію на менші загони для ефективного контролю над великим територіальним обсягом.

Цей метод широко застосовується не лише в військовій стратегії, але й у багатьох інших галузях, таких як управління проектами, бізнес-стратегія, наукові дослідження та інші. Ідея полягає в тому, щоб поділити велику задачу на менші, більш керовані частини, щоб краще контролювати процес і досягати поставлених цілей.

Метод "Розділяй і володарюй" базується на кількох ключових ідей та концепціях, які роблять його ефективним для розв'язання складних задач. Основні ідеї та концепції методу:

- центральною ідеєю є розбиття складної задачі на менші, більш керовані підзадачі. Декомпозиція дозволяє розглядати проблему на рівні її складових частин, що полегшує розуміння та розв'язання;
- кожна підзадача повинна бути якнайбільш незалежною від інших. Це дозволяє розв'язувати їх паралельно та уникати взаємних залежностей;
- задачі можуть бути додатково розбиті на ще менші частини, і так далі, створюючи ітераційну структуру, де кожна ітерація призначена для вирішення конкретної частини завдання;

- результати вирішення підзадач збираються та об'єднуються для отримання рішення загальної проблеми.

Основні етапи методу "Розділяй і володарюй":

- спочатку велика задача розбивається на менші, більш прості підзадачі. Це може бути повторенням процесу розбиття до того моменту, коли підзадачі стають достатньо простими для ефективного розв'язання;
- кожна з отриманих підзадач розв'язується окремо. Цей етап визначається самостійністю і простотою розв'язання окремих підзадач;
- розв'язання отриманих підзадач об'єднується для формування відповіді для вихідної проблеми. Іноді цей етап також називають "конкатенацією" розв'язків [14, 15].

Цей метод широко використовується в алгоритмах і програмуванні, а також у великій кількості областей, включаючи обчислювальну математику, сортування, пошук, оптимізацію завдань та інші. Він сприяє збільшенню ефективності та простоти вирішення складних завдань, зокрема тих, які можуть бути легко поділені на менші частини.

Метод "Розділяй і володарюй" має кілька переваг і особливостей:

- даний метод може значно полегшити вирішення складних завдань, роблячи їх меншими та більш керованими;
- цей підхід особливо ефективний при масштабуванні, коли вихідні дані стають дуже великими. Розділення проблеми на менші частини дозволяє обробляти їх незалежно, що призводить до зменшення часу виконання;
- в реалізації алгоритмів принцип "Розділяй і володарюй" може бути відносно простим, особливо коли відомі ефективні методи розділення і об'єднання;
- оскільки підзадачі вирішуються незалежно одна від одної, цей метод легко піддається паралельному виконанню на багатоядерних або розподілених системах [15].

## 2.2. Методи та моделі розв'язку логістичних транспортних задач

Алгоритми пошуку маршрутів базуються на алгоритмах графового пошуку та аналізують шляхи між вузлами, виходячи з одного вузла та просуваючись по зв'язках до досягнення пункту призначення. Ці алгоритми застосовуються для визначення оптимальних маршрутів у графі для різних завдань, таких як логістичне планування або IP-маршрутизація.

Задача полягає в знаходженні найкоротшого шляху між вихідною вершиною  $s$  та кінцевою вершиною  $d$ , де  $d \in V$ , у напівнескінченному графі, представленому множиною вершин  $V$ , множиною вагованих ребер  $E$ , та вагами  $w(e)$  для кожного ребра  $e \in E$ . У вхідних даних до алгоритму найкоротшого шляху використовується граф  $G$ , що складається з множини вершин  $V$  та множини ребер  $E$ , де  $G = (V, E)$ . Ребра можуть бути орієнтованими або неорієнтованими, і мають явно визначені ваги, де вага позначається як  $w(e)$ , де  $e \in E$ , або неваговані, де неявна вага вважається рівною 1 [6].

З урахуванням графа  $G = (V, E)$  та вибору джерела  $s \in V$  обчисліть всі відстані  $\delta(s, v)$ , де  $v \in V$ . Простий випадок для пошуку найкоротших шляхів з одного джерела (SSSP) передбачає відсутність напрямку в графі.

У роботі Кормен та інших [16] вказується, що можна використовувати пошук в ширину, розпочинаючи з кореневої вершини та аналізуючи всі прилеглі вершини. Для кожної сусідньої вершини він перевіряє невідвідані вершини, доки не знайдено шлях з мінімальною кількістю ребер від джерела до цільової вершини.

Алгоритм Дейкстри [17] розв'язує задачу пошуку найкоротшого шляху з одного джерела (SSSP) до всіх інших вершин графа. Цей алгоритм призначений для використання в орієнтованих графах з невід'ємними вагами ребер. Алгоритм розрізняє два типи вершин: розв'язані, нерозв'язані.

Спочатку алгоритм визначає вихідну вершину як вже оброблену і оцінює всі інші ребра пов'язані з невирішеними вершинами щодо

найкоротших шляхів до кінцевої точки. Цей алгоритм визначає найкоротше ребро та додає відповідну вершину до списку оброблених вершин.

Алгоритм продовжує ітерації до тих пір, поки всі вершини не будуть оброблені. Часова складність алгоритму Дейкстри становить  $O(n^2)$ , де  $n$  - кількість вершин, оскільки він перевіряє всі ребра. Це особливо ефективно в ситуаціях, коли ваги на деяких ребрах є великими. Однак важливо зауважити, що алгоритм працює лише з ребрами, де ваги є не від'ємними, і він призначений для статичних графів.

Алгоритм Дейкстри використовує метод грубої сили для пошуку оптимального найкоротшого шляху, і тому він є представником жадібних алгоритмів. Його процедура послідовного покращення ґрунтується на принципі оптимальності, введеному алгоритмом Беллмана-Форда [18].

Алгоритм Дейкстри має здатність вирішувати рівняння динамічного програмування за допомогою методу, що відомий як метод досягнення [19, 20, 21]. Динамічне програмування, як метод, надає перевагу, оскільки воно дозволяє уникнути процесу грубого пошуку шляхом вирішення підзавдань. Основна перевага полягає у здатності розглядати експоненційно великий набір рішень, уникаючи при цьому явного вивчення всіх можливих варіантів.

Жадібні та динамічні версії алгоритму Дейкстри, з точки зору пошуку оптимального рішення, мають схожі характеристики. Вони спрямовані на досягнення оптимального рішення. Втім, їхня відмінність полягає в можливості обрати різні шляхи для досягнення цього оптимуму. Різні рішення можуть бути досягнуті завдяки варіативності шляхів, які ці алгоритми враховують у процесі пошуку.

Фредман та Тар'ян [22] вдосконалили алгоритм Дейкстри, використовуючи структуру даних, відому як купа Фібоначчі (F-купа). Ця реалізація дозволяє досягти часу виконання  $O(n \log n + m)$ , оскільки загальний час, витрачений на операції з купою, становить  $O(n \log n + m)$ , і вартість інших операцій складає  $O(n + m)$ .



Фредман та Віллард [23, 24] представили розширення алгоритму Дейкстри за допомогою структури, відомої як AF-Heap, яке забезпечує час виконання  $O(n + m \log n / \log \log n)$ . AF-Heap дозволяє сталий час для більшості операцій з купою та  $O(\log n / \log \log n)$  для операції видалення.

Дрісколл та Габоу [25] пропонують ослаблену купу Фібоначчі, яка представляє собою біноміальну чергу, що дозволяє порушувати порядок купи. Цей алгоритм є паралельною реалізацією алгоритму Дейкстри, з оптимізацією реалізації порядку пріоритетів.

Боас [26] та його співавтори використовують стратифіковане двійкове дерево для реалізації своїх алгоритмів. Запропонована методика надає змогу ефективно опрацьовувати пріоритетні черги у режимі онлайн. Цей алгоритм має складність обчислень  $O(\log \log n)$  та вимоги до пам'яті  $O(n \log \log n)$  [27].

Дослідження, проведене Торуп [28], вказує на існування аналогії між задачею пошуку найкоротших шляхів (SSSP) та проблемою сортування, де завдання SSSP не є більш складним, ніж сортування ваг ребер.

Торуп подає опис черги з пріоритетами, зазначаючи складність операцій як  $O(\log \log n)$  та складність  $O(m \log \log n)$  для SSSP. У своєму дослідженні він детально розглядає використання черги з пріоритетами при наявності пам'яті з довільним розміром слова.

Проводячи аналогію з вищевикладеним, Хан [29] представляє детермінований алгоритм сортування цілих чисел у лінійному просторі, забезпечуючи часову складність  $O(m \log \log n \log \log n)$  для задачі пошуку найкоротших шляхів (SSSP). Підхід Хана вказує на те, що можна вирішувати проблему сортування будь-якої величини, замінивши її на сортування дуже малих цілих чисел.

Торуп [28] вводить детермінований алгоритм, що працює у лінійному просторі та часі, створюючи ієрархічну структуру групування, яка уникає операції сортування. Ця структура групування є динамічним набором, куди елемент може бути вставлений чи видалений. Елементи з сегментів можуть

бути вибрані довільним чином, наприклад, за допомогою двозв'язного списку. Алгоритм Торупа працює через обхід дерева компонентів.

Хагеруп вдосконалює алгоритм Торупа, досягаючи часової складності  $O(n+m \log w)$ , де  $w$  - ширина машинного слова. Це досягається за допомогою детермінованого лінійного алгоритму часу та простору [30].

Беллман, Форд і Мур [31] представили алгоритм пошуку найкоротших шляхів (SSSP), який відрізняється від алгоритму Дейкстри тим, що може обробляти негативні ваги. По суті, цей метод працює аналогічно алгоритму Дейкстри, спрямовуючи свої зусилля на визначення найкоротшого шляху. Однак відмінність полягає в тому, що замість вибору лише тих сусідніх ребер, які мають мінімальну відстань, він розглядає всі сусідні ребра. Для цього він виконує  $n-1$  циклів, щоб гарантувати розповсюдження всіх змін у графі.

Хоча цей метод надає швидше рішення порівняно із алгоритмом Беллмана-Форда, але він не може виявляти негативні цикли або працювати з негативними вагами. В існуванні негативного циклу полягає той факт, що не існує однозначного найкоротшого шляху, оскільки цикл спричиняє зменшення загальної ваги через повторні ітерації. Алгоритм Беллмана-Форда має часову складність  $O(nm)$ , і його перевагами є здатність працювати з негативними вагами та виявлення негативних циклів. Однак його недоліком є можливо менша швидкість в порівнянні з алгоритмом Дейкстри. Крім того, алгоритм Беллмана-Форда не завершується, коли ітерації більше не впливають на ваги графа.

Короп [32] розглядає проблему визначення наявності негативного циклу у графі та пропонує концепцію "мінімального середнього циклу". Він стверджує, що виявлення мінімального середнього циклу аналогічно виявленню негативного циклу.

Єн [33] представляє дві модифікації для покращення продуктивності порівняно з алгоритмами Беллмана-Форда і Мура. Перша модифікація передбачає розслаблення ребер. Ребро розслаблюється, якщо значення вершини змінюється. Друга модифікація включає розділ ребер на основі

лінійного порядку за всіма вершинами. Набір ребер потім розбивається на одне або кілька підмножин, після чого проводиться порівняння двох наборів за визначеною схемою.

Зазначимо, що Баністер і Елпштейн [34] запропонували використання випадкового порядку замість довільного. Це призвело до зменшення кількості ітерацій для обох підмножин.

При розгляді графа  $G = (V, E)$  обчисліть всі відстані між вихідною вершиною та точкою призначення  $v$ , використовуючи різні елементи множини  $V$ . Найзагальнішим випадком усіх найкоротших шляхів (APSP) є граф з невід'ємними вагами ребер. У цьому випадку алгоритм Дейкстри може бути застосований окремо для кожної вершини графа. Тимчасова складність цього підходу становить  $O(mn + n^2 \log n)$  [35]. За час проведення досліджень було розроблено значну кількість алгоритмів, спрямованих на опрацювання реальних ваг ребер у випадку завдання пошуку найкоротших шляхів для всіх можливих пар вершин.

Алгоритм Флойда-Уоршалла [36] призначений для знаходження всіх можливих найкоротших шляхів (APSP) у ваговому графі, що містить як ребра з позитивними, так і з негативними вагами. Цей алгоритм виявляє наявність циклів із негативною вагою, хоча не здатний їх усунути. Складність алгоритму Флойда-Уоршалла становить  $O(n^3)$ , де  $n$  - кількість вершин. Виявлення циклів із негативною вагою відбувається при дослідженні матриці шляхів.

Недоліком алгоритму Флойда-Уоршалла є його неспроможність точно визначити найкоротші шляхи між парами вершин, оскільки він не зберігає проміжні вершини під час обчислень. Однак цю інформацію можна зберегти за допомогою простих оновлень під час виконання алгоритму, що призводить до просторової складності  $O(n^3)$  або навіть  $O(n^2)$  при використанні одного масиву зміщень.

Однією з сильних сторін алгоритму є його здатність обробляти ребра з негативними вагами та виявляти цикли із негативними вагами.

Проте основним недоліком є складність синхронізації при використанні алгоритму Дейкстри на всіх вершинах для перетворення його з проблеми найкоротших шляхів з однієї вершини (SSSP) у проблему найкоротших шляхів між всіма парами вершин (APSP). Ця часова складність може бути нижчою за  $O(n^3)$ , лише якщо  $m < n^2$  (з умови розрідженого графа). У численних дослідженнях були запропоновані оптимізовані варіанти алгоритму з найкращою годинною продуктивністю порівняно з алгоритмом Флойда-Уоршалла для графів із великим числом ребер.

Значний прогрес був досягнутий завдяки інноваційному підходу, представленому Фредманом [37], що базується на матрично-орієнтованій концепції. Цей підхід виходить з теореми, розробленої Ахо і Хопкрофтом [38], яка стверджує, що складність множення матриці  $N \times N$  за допомогою методу *amin/plus* аналогічна складності множення найкоротших шляхів. Відзначено, що для вирішення задачі знаходження найкоротших шляхів для всіх пар (APSP) достатньо  $O(N^{5/2})$  порівнянь. Результатом є алгоритм із часовою складністю  $O(n^3(\log \log n)/\log n^{1/3})$ .

Найбільш значущий внесок у досягнення цілі зроблено Ханом та Такаока [39], які демонструють зниження коефіцієнта  $O(\log \log n)^2$  у порівнянні з іншими дослідженнями. Їхній підхід акцентується на обчисленні творів відстані. Згодом,  $n \times n$  матриця  $J$  розбивається на  $m$  підматриць, кожна з яких має розміри  $n \times n/m$ , де  $m$  визначається на підставі конкретного критерію. Алгоритм виконує послідовні маніпуляції з матрицею, включаючи побудову індексів, кодування та етапи розділу, досягаючи запропонованої межі. Найкраща часова складність для ребра з невід'ємною вагою =  $O(n^2 \log n)$ .

Спочатку алгоритм впорядковує всі списки суміжності в порядку зростання ваги. Потім він виконує обчислення найкоротших шляхів з вагою  $p$  разів і продовжує ітерації. На першому етапі він використовує концепцію потенціалу по ребрах вершин і вибирає та позначає ребро з мінімальним потенціалом. Потенціал, визначений з потенційної моделі, представляє собою розподіл ймовірностей на повних орієнтованих графах з довільною довжиною

ребер, що не містять негативних циклів. Алгоритм діє у двох основних етапах, кожен з яких має визначений інваріант, і має складність  $O(n^2 \log n)$ .

Краще позитивне ціле число  $i$ . Складність ваги ребра оцінюється за формулою  $O(n^\omega + c)$ , де  $\omega < 2,575$  - показник ступеня, запропонований Копперсмітом [33]. Пропонований алгоритм забезпечує гладкий перехід між найшвидшим точним та наближеним алгоритмами пошуку найкоротших шляхів з лінійною частотою помилок. Фокус алгоритму на орієнтованих графах з невеликими позитивними цілочисленими вагами дозволяє отримати адитивні апроксимації. Важливо відзначити, що ці апроксимації є поліноміальними, враховуючи фактичні відстані між парами вершин.

Алгоритм орієнтованого пошуку найкоротшого шляху ґрунтується на додаванні анотацій до вершин або ребер графа, що містять додаткову інформацію. Ця інформація дозволяє алгоритму визначити, яку частину графа слід видаляти у просторі пошуку.

Харт та інші автори [40] пропонують простий орієнтований алгоритм, відомий як  $A^*$ . Цей алгоритм використовує евристичний підхід до пошуку найкоротшого шляху. На відміну від алгоритму Дейкстри,  $A^*$  є інформованим алгоритмом, який шукає маршрути, що ведуть до кінцевої мети.  $A^*$  - це оптимальний жадібний алгоритм з принципом "найкращий перший".

Відмінність  $A^*$  від інших алгоритмів полягає у його здатності враховувати пройдену відстань.  $A^*$  завжди знаходить найкоротший шлях, якщо використовується допустима евристична функція. Сильною стороною алгоритму є його ефективність в порівнянні з Дейкстрою, оскільки він досліджує менше вершин. З іншого боку, якщо  $A^*$  не використовує ефективну евристичну функцію, він може не забезпечити найкоротший шлях. Деякі модифікації алгоритму  $A^*$  використовують орієнтацію та інші стратегії для досягнення кращої продуктивності в різних умовах налаштування.

Голдберг та Вернек [41] розробили етап передпроцесінгу, на якому спочатку обирається кілька орієнтирів, а потім розраховується найкоротший шлях, який зберігається між вершинами цих орієнтирів. Вони пропонують

метод нижньої межі з постійним часом, який використовує розраховані відстані, додатково до якості нерівності трикутника. Цей метод нижньої межі базується на алгоритмі  $A^*$ , обраному орієнтиру та нерівності трикутника.

Гутман [42] вносить внесок у вирішення проблеми, використовуючи концепцію досяжності. Метод Гутмана ґрунтується на збереженні значень площі та евклідових координат усіх вершин. Основною перевагою підходу Гутмана є його можливість комбінування з алгоритмом  $A^*$ .

У порівнянні з роботою Голдберга та Вернека, підхід Гутмана переважає їхній метод при використанні одного орієнтира, але виявляється менш ефективним у випадку шістнадцяти орієнтирів. З іншого боку, підхід Гутмана ґрунтується на припущеннях, специфічних для предметної області, і має більшу складність передпроцесінгу, що обумовлює його обмежену застосовність в динамічних умовах.

Потаміас та ін. [43] представляють наближений метод, базований на орієнтирах, для оцінки відстані між точками у великих мережах. Вони теоретично доводять NP-складність даної проблеми та пропонують евристичні рішення. Зокрема, вони виводять інтелектуальний метод вибору орієнтирів, який може забезпечити більш високу точність та охоплювати у 250 разів менше місця, ніж випадковий вибір орієнтирів. Серед розглянутих стратегій, центральність виявляється більш надійною, ніж стратегія ступеня. Додатково, стратегії, що ґрунтуються на секціонуванні, наприклад, Border/P, продемонстрували найкращі обчислювальні витрати на різних наборах даних.

Клейнберг та ін. [44] пропонують алгоритм із доведеними гарантіями продуктивності для триангуляції та вбудовування на основі маяків. Алгоритми, що базуються на маяках, в основному використовуються для триангуляції, використовуючи нерівність трикутника для визначення невимірних відстаней. Автори вказують, що мультиплікативна помилка  $1+\delta$  на частці відстаней може бути досягнута через реконструкцію на основі триангуляції при постійній кількості маяків. Алгоритм також забезпечує постійне спотворення на величину  $1 - \delta$  відстаней.

Мітки ребер - це підхід, що базується на попередньому обчисленні інформації для конкретного ребра "e" та набору вершин "M". Надмножина  $M(e)$  включає всі вершини найкоротшого шляху, який починається з ребра "e". Граф спочатку поділяється на набір областей однакового розміру, а також визначається заздалегідь обчислений набір граничних вершин.

Для обчислення прапорів ребер виконується обчислення найкоротших шляхів з однієї чи кількох граничних вершин в регіонах. Різні дослідження пропонують алгоритм для розріджених орієнтованих графів з невід'ємними вагами ребер, який відомий як "дуга-прапор".

Підхід "дуга-прапор" передбачає попередню обробку графа для створення інформації, яка прискорює запити найкоротшого шляху, розділяючи граф на області та визначаючи, чи містить дуга найкоротший шлях в конкретній області. З використанням відповідної схеми поділу та двоспрямованого пошуку, метод "дуга-прапор" може працювати у 500 разів швидше, ніж стандартний алгоритм Дейкстри на великих графах.

Шилінг та ін. [45] представляють додаткові удосконалення, використовуючи пошук одного разу для кінцевого регіону. За їхнім методом вдається забезпечити прискорення більше, ніж на 1470 разів у підмережі з 1 мільйоном вершин. Голдберг і Вернек [41] впроваджують алгоритм пошуку орієнтирів (ALT) на основі  $A^*$ , який використовує нерівність трикутника. Вони доводять, що попереднє обчислення відстаней до набору орієнтирів може значно обмежити обчислювальні витрати на пошук найкоротшого шляху. У середньому вони пропонують використовувати 20 орієнтирів, які розташовані рівномірно за кутами графа.

В свою чергу, цей підхід призводить до прискорення планування маршруту. Бауер та ін. [46] розглядають систематичне поєднання методів прискорення, які були запропоновані для алгоритму Дейкстри, зокрема, додаючи цілеспрямовані стратегії до ієрархічних підходів. Вони представляють узагальнений метод, який демонструє, як можна досягти покращення продуктивності шляхом комбінації різних підходів. Результати

вказують, що Highway Vertex Routing та Arc-Flags забезпечують найкраще прискорення при збереженні прийнятних витрат на попередню обробку. Також, вони впроваджують ієрархічний алгоритм пошуку орієнтирів (ALT) з використанням  $A^*$  для щільних графів.

Делінг та ін. [47] представляють алгоритм, відомий як кільцевий маршрутизатор громадського транспорту (RAPTOR). Відмінною особливістю RAPTOR є його незалежність від алгоритму Дейкстри, оскільки він перевіряє кожен маршрут у графі не більше одного разу. RAPTOR працює в динамічних сценаріях і може легко розширюватись для обробки гнучких графіків відправлень, наприклад, гнучких годин відправлення. Бауер і Делінг використовують ієрархічні методи для розширення підходу до граничних прапорів, зокрема, використовуючи ієрархії стискання під час попередньої обробки, що дозволяє усунути основний недолік обробки граничних прапорів. Запропонована робота відома як SHARC (Shortcuts + Arc-Flags).

Ключовою особливістю SHARC є можливість встановлення неоптимальних прапорів ребер для більшості ребер, що спрощує попередню обробку, зосереджуючи увагу лише на ключових ребрах. Ще однією важливою особливістю SHARC є неявне включення ієрархічних аспектів. SHARC також розширює підхід Мерінга та інших, використовуючи прапори ребер. Усе це спрямовано на забезпечення швидкого однонапрявленого алгоритму для оптимального пошуку маршрутів.

Алгоритми ієрархічного найкоротшого шляху пов'язані з створенням багаторівневої ієрархії вершин на етапі попередньої обробки. Ієрархічна структура помітна в таких областях, як дорожні мережі, де вона виявляє ієрархічні властивості, наприклад, упорядковує важливі вулиці, автомагістралі та міські вулиці [49].

Загалом методи, що використовують ієрархії стиснення, забезпечують низьку просторову складність. Ієрархії стиснення містять безліч варіантів, таких як методи, засновані на охопленні, ієрархії шосе та маршрутизація



вершин. З іншого боку, маршрутизація транзитних вершин та мітки концентраторів забезпечують швидке виконання запиту.

Ієрархії шосе фіксують властивості, засновані на ребрах. Наприклад, краю шосе краще уявляти найкоротші шляхи, хоча вони можуть і не розташовуватися між вихідною та кінцевою вершинами. Алгоритм генерує ієрархію графів, яка забезпечує швидку годину виконання запиту з гарантіями коректності.

Сандерс та Шультес [50] пропонують алгоритм астатичних ненаправлених ієрархій шосе, заснований на понятті правильного визначення локального пошуку та мережі автомагістралей. Вони визначають локальний пошук як такий, який відвідує  $N$  (параметр налаштування) найближчих вершин від джерела або цілі. Край шосе створюється, якщо він лежить на шляху від вихідної вершини до цільової вершини, причому це ребро не знаходиться в межах найближчих вершин від джерела або призначення.

Стискаюча ієрархія має рівень для кожної вершини, що досягає  $n$  рівнів. Ієрархічні моделі можуть покращити продуктивність запитів, оскільки пошук може проводитись лише за графом. Це зменшило складність простору, оскільки ребра зберігаються лише в них нижніх кінцевих точках.

Гейсбергер та ін. ал. [51] пропонують стискаючі ієрархії, в яких вершини спочатку упорядковуються за важливістю, а потім ієрархія генерується шляхом ітеративного стиснення найменш важливих вершин. Стиснення – це процес заміни найкоротших шляхів, що проходять через вершину, на те, що вони називають ярликами. Вони пропонують ієрархічний алгоритм, який використовує двонаправлений метод пошуку найкоротшого шляху.

Батц та ін. [52] пропонують версію алгоритму, що залежить від часу. Він стосується залежати від часу дорожніх мереж, пропонуючи швидкий та точний алгоритм планування маршруту. Проблема, з якою він стикається, — це просторова складність. Вони вирішують цю проблему, використовуючи апроксимації шматково-лінійних функцій, які призводять до значного

скорочення простору при збереженні коректності. ярликів, за якими можна, можливо здійснювати пошук.

Гейсбергер та ін. [53] розробили алгоритм, що базується на скорочуються ієрархіях, для розрахунку найкоротших шляхів на основі континентів. Етап попередньої обробки спирається на ієрархічні властивості дорожніх мереж для найкоротших ребер. Вони використовують модифіковану версію алгоритму «Дейкстри», який відвідує лише кілька сотень вершин.

У багаторівневому накладеному графі, якщо набір вершин лежить на певному рівні, найкоротші шляхи на цьому рівні не використовують вершини верхніх рівнів. У свою чергу, цей метод залежить від правильного вибору вершин, які будуть діяти як орієнтири на більш високих рівнях.

Шульц та ін. [54] пропонують багаторівневий метод декомпозиції на основі графів, спрямований на скорочення простору. Цей метод попередньо обчислює найкоротші шляхи та замінює ваги окремих ребер вагами, рівними довжині найкоротшого шляху. В результаті виходить підграф меншого розміру. за розміром проти вихідним графом. відстані підграфа між набором вершин такі ж, як відстань графа найкоротшого шляху тим самим набором вершин у вихідному графі.

Хольцер та ін. [55] вводять кілька критеріїв вибору вершин у накладених графах. До них належать критерії для визначення репрезентативної підмножини вихідного графа. Вони досліджують ефективність критеріїв у багаторівневих накладених граф і прискорення, що досягається при обчисленні найкоротшого шляху.

Маршрутизація транзитних вершин попередньо розраховує найкоротші шляхи до всіх орієнтирів, зазначених у графі, і назад. Алгоритм вимагає великої попередньої обробки, але має дуже швидку годину запиту, оскільки вимагає обмеженого кількості пошуків між орієнтирами, розташованими в різних місцях.

Баст та ін. [56] пропонують маршрутизацію транзитних вершин. Вони припускають, що вертикального та горизонтального сканування достатньо для

обчислення набору транзитних вершин. Вони також ілюструють деякі методи, що дозволяють зробити цей підхід більш економічним.

Моделювання дорожніх мереж у вигляді низькорозмірного графа - це метод, який використовується для обчислення найкоротших шляхів. Одним з методів, що використовують для такого моделювання, є процес маркування.

Алгоритми маркування були введені в область розподілених обчислень [57]. На етапі попередньої обробки кожною вершиною обчислюється та присвоюється пряма мітка та зворотна мітка. Пряма мітка включає набір вершин  $w$ , де шкiрна вершина містить обчислене відстань  $\text{dist}(v, w)$  від  $v$ . Зворотна мітка складається з набору вершин  $u$ , де шкiрна вершина містить обчислене відстань  $\text{dist}(u, v)$  до  $v$ . Ці мітки пізніше використовують на етапі запиту для визначення вершин, які мінімізують відстань від джерела до пункту призначення. Мітку можна сприймати як набір вузлів, які вершина  $v$  має пряме з'єднання  $s$ . Алгоритм маркування гарантує, що будь-які дві вершини мають один загальний концентратор при обчисленні найкоротшого шляху. Маркування концентратора починається з попередньої обробки вершин, де для кожної вершини  $v$  попередньо обчислюється відстань до набору орієнтирів  $L(v)$  у мітці вершини.

Алгоритм запиту працює швидко, поки кількість орієнтирів вихідної та цільової вершин невелика. мітки послідовно дозволяють алгоритму демонструвати гарну локальність.

### 2.3. Моделі транспортних логістичних задач

Розглянемо приклад класичного формулювання транспортної задачі: нехай маємо  $m$  пунктів відправлення вантажів за формулою (2.1), на яких зосереджені запаси якого-небудь однорідного вантажу за формулою (2.2) відповідно. Сумарний запас у постачальників дорівнює сумі  $a_m$ .

$$A_1, A_2, A_3, \dots, A_m \quad (2.1)$$

$$a_1, a_2, a_3, \dots, a_m \quad (2.2)$$

де  $A_m$  – пункт відправлення;

$a_m$  – запас вантажу.

Крім того, за формулою (2.3) є  $n$  пунктів споживачів, які подали заявки на поставку вантажу в обсягах за формулою (2.4). Сумарна величина заявок становить у дорівнює сумі  $b_n$ .

$$B_1, B_2, B_3, \dots, B_n \quad (2.3)$$

$$b_1, b_2, b_3, \dots, b_n \quad (2.4)$$

де  $B_n$  – пункт споживача;

$b_n$  – величина заявок.

Вартість перевезення однієї одиниці вантажу від постачальника  $A_i$  до споживача  $B_j$  подані за формулою (2.5) як елементи матриці  $C$ .

$$C = \begin{pmatrix} C_{11} & C_{12} & C_{1n} \\ C_{21} & C_{22} & C_{2n} \\ C_{m1} & C_{m2} & C_{mn} \end{pmatrix} \quad (2.5)$$

де  $C$  – транспортні тарифи.

Тоді, транспортна задача формулюється наступним чином: необхідно скласти оптимальний план перевезень, тобто знайти такі значення об'єму перевезень за формулою (2.6), щоб вивести всі вантажі від постачальників, задовольнити заявки всіх споживачів і забезпечити мінімальні транспортні витрати на перевезення вантажу.

$$x_{ij} = (i = \overline{1, m}; j = \overline{1, n}) \quad (2.6)$$

де  $x_{ij}$  – об'єму перевезень від постачальника  $i$  до споживача  $j$ .

Математична модель транспортної задачі полягає в мінімізації функції за формулою (2.7), при умовах (2.7, 2.8, 2.9)

$$F = \sum_{i=1}^m \sum_{j=1}^n c_{ij} * x_{ij} \rightarrow \min \quad (2.7)$$

$$\sum_{j=1}^n x_{ij} = a_j, i = \overline{1, m} \quad (2.8)$$

$$\sum_{i=1}^m x_{ij} = b_j, i = \overline{1, n} \quad (2.9)$$

$$x_{ij} \geq 0, i = \overline{1, m}; j = \overline{1, n} \quad (2.10)$$

Будь-яке невід'ємне рішення системи лінійних рівнянь за формулою (2.8, 2.9), називається планом перевезень транспортної задачі.

План перевезень, що має не більше  $m + n - 1$  відмінних від нуля змінних  $x_{ij}$ , називається опорним.

Якщо число відмінних від нуля компонент в опорному плані, в точності, дорівнює  $m + n - 1$ , то план перевезень називається не виродженим, якщо менше цього числа, то виродженим.

План перевезень при якому функція за формулою (2.7) приймає своє мінімальне значення, називається оптимальним планом транспортної задачі.

Розглянемо транспортну задачу за критерієм часу: як і для класичної транспортної задачі, маємо  $m$  постачальників із запасами однорідного вантажу в кількості  $a_m$  та  $n$  споживачів, яким цей вантаж треба доставити в обсязі  $b_n$ .

Для доставки однієї одиниці вантажу з кожного пункту відправлення  $A$ , - в кожен пункт прибуття  $B$  необхідно витрати часу розміром  $t_B$ .

Необхідно знайти оптимальний план перевезень за формулою (2.8, 2.9). Крім того, час  $t_B$ , який витратиметься на всі перевезення, був би мінімальним.

### РОЗДІЛ 3

## РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВ'ЯЗКУ СКЛАДНИХ ЗАДАЧ ТРАНСПОРТНОЇ ЛОГІСТИК

### 3.1. Побудова моделі розв'язку комбінованих логістичних задач

Згідно методу “Розділяй і Володарюй”, вирішення задачі полягає у розбитті її на більш прості частини - підзадачі, та поєднання отриманих рішень з кожної підзадачі в єдине рішення для вихідної задачі.

Отже, нехай маємо вихідну комбіновану логістичну задачу  $N$ , яку хочемо вирішити. Використовуючи метод “Розділяй і Володарюй” розбиваємо вихідну задачу на підзадачі, алгоритми  $X_1$ ,  $X_2$ , де  $X_k$  - алгоритм вирішення окремої підзадачі. Тоді:

$$N = F(x_1, x_2) \quad (3.1)$$

де  $F$  - функція, яка поєднує частини  $X_k$ .

Розділяючи задачі на окремі підзадачі та поєднуючи результат, ми отримуємо бажаний алгоритм з розв'язку задачі. Однак при комбінуванні двох або більше алгоритмів, виникає питання щодо порядку їх об'єднання. Розглянемо приклад:

$$N_1 = F(x_1, x_2) \quad (3.2)$$

$$N_2 = F(x_2, x_1) \quad (3.3)$$

Отже, маємо два різних підходи к вирішенню однієї і тієї задачі. Загальну кількість унікальних підходів дорівнює факторіалу кількості  $X$ . При комбінування алгоритмів отримаємо матрицю можливих комбінувань.

$$N_k = \begin{bmatrix} 1 & x_1x_2 \\ x_2x_1 & 1 \end{bmatrix} \quad (3.4)$$

Оцінити кожний з підходів та обрати найкращий можемо тільки за допомогою показника ефективності -  $R_k$ .

$$N = \text{MAX}(R_1, R_2) \quad (3.5)$$

$$R_1 = F(x_1, x_2) \quad (3.6)$$

$$R_2 = F(x_2, x_1) \quad (3.7)$$

Показник ефективності  $R_k$  є змінною і залежить від поставлених цілей задачі. Він відображає загальну ефективність алгоритмів. Наприклад, до цього показника можуть входити: кількість інтеграцій, загальний час роботи алгоритму, кількість використаних транспортних засобів і тому подібне.

### 3.2. Реалізація інформаційної системи підтримки прийняття рішень.

Метою інформаційної системи, є надання користувачеві інформації щодо найбільш оптимальних комбінацій алгоритмів для вирішення задачі комбінованої логістики.

Роздивимося потреби користувачів інформаційної системи:

- логістичні менеджери потребують інструмент для швидкого та точного вибору оптимальних комбінацій алгоритмів для задач логістики;
- аналітики бажають доступ до статистичної інформації та аналізу ефективності використання різних алгоритмів;
- стейкхолдери мають інтерес до можливості генерації звітів для подальшого узагальнення та прийняття рішень.

На основні потреб користувачів, можемо виділити основні функціональні вимоги:

- забезпечити можливість завантаження вхідних даних, що включають параметри завдань комбінованої логістики: обсяг вантажу, відстань перевезення, час доставки, тощо;
- створити базу даних алгоритмів, яка включатиме різні види алгоритмів для розв'язання завдань комбінованої логістики: генетичні алгоритми, алгоритми імітації відпалу, алгоритми оптимізації маршрутів тощо;
- розробити модуль для аналізу та порівняння різних комбінацій алгоритмів на основі вхідних даних та параметрів користувача;
- розробити зручний графічний інтерфейс для користувача, який дозволить введення даних, вибір алгоритмів та отримання результатів у зручному вигляді;
- надати користувачеві рекомендації щодо найбільш оптимальних комбінацій алгоритмів для конкретного завдання комбінованої логістики.

Алгоритм роботи користувача в інформаційній системі наведено на рисунок 3.1.

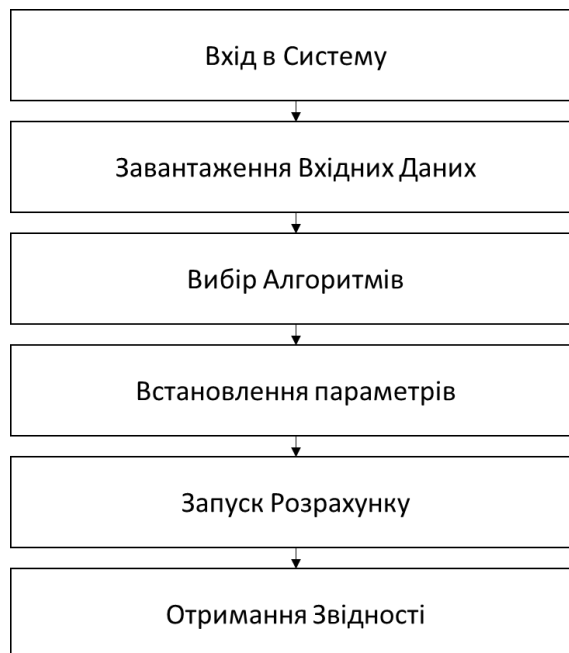


Рисунок 3.1 – Алгоритм роботи користувача



У таблиці 3.1 наведені технічні вимоги щодо інформаційної системи.

Таблиця 3.1 – Класифікація технічних вимог

Аспект	Вимоги
Комбінування	<ul style="list-style-type: none"> <li>- автоматичне комбінування текстових алгоритмів: Система має бути здатна автоматично поєднувати текстові алгоритми, надані користувачем, з урахуванням особливостей завдання.</li> <li>- виявлення зв'язків: При комбінації алгоритмів система має визначати зв'язок між ними, враховуючи контекст завдання.</li> <li>- автоматичне проставлення зв'язків: Система повинна автоматично проставляти зв'язки між комбінованими алгоритмами.</li> </ul>
Перетворення	<ul style="list-style-type: none"> <li>- перетворення алгоритму на код: Система має бути здатна перетворювати текстові алгоритми на виконуваний код.</li> <li>- запуск у тестовому вигляді: Перетворений код має бути запускається у тестовому режимі для перевірки його працездатності.</li> <li>- обробка помилок: Система повинна передбачати обробку можливих помилок під час перетворення та запуску алгоритмів.</li> </ul>
Результат	<ul style="list-style-type: none"> <li>- цифровий висновок результату: Система повинна виводити результат виконання алгоритмів у цифровому форматі, якщо це стосується задачі.</li> <li>- текстовий висновок результату: Результат має бути представлений у текстовому вигляді розуміння користувача.</li> </ul>

Інструменти реалізації наведені у таблиці 3.2.

Таблиця 3.2 – Інструмент реалізації інформаційної системи

Інструмент	Завдання
SQL Server	<ul style="list-style-type: none"> <li>- використання SQL Server для зберігання та управління базою даних алгоритмів.</li> <li>- створення таблиць для різних видів алгоритмів, їх параметрів та результатів використання.</li> <li>- забезпечення ефективного доступу до даних для аналізу та вибору оптимальних комбінацій.</li> </ul>
Python	<ul style="list-style-type: none"> <li>- використання мови програмування Python для розробки інформаційної системи.</li> <li>- використання Python для реалізації модулів аналізу вхідних даних, комбінування алгоритмів, графічного інтерфейсу та інших функціональностей системи.</li> </ul>
Текстові моделі конструктори	<ul style="list-style-type: none"> <li>- використання текстових нейронних мереж для автоматичного комбінування алгоритмів.</li> <li>- визначення зв'язків між алгоритмами враховуючи особливості задачі комбінованої логістики.</li> <li>- автоматичне проставлення зв'язків між комбінуваннями алгоритмів.</li> </ul>

## Продовження таблиці 3.2

Нейроні мережі	<ul style="list-style-type: none"> <li>- використання нейронних мереж для перетворення текстових алгоритмів у програмний код.</li> <li>- забезпечення можливості тестового запуску перетвореного коду для перевірки його працездатності.</li> <li>- обробка можливих помилок під час перетворення та запуску алгоритмів.</li> </ul>
----------------	---

Проаналізувавши особливості побудови інформаційної системи, можемо виділити основні її компоненти:

а) база даних:

- 1) включає в себе інформацію про різні алгоритми, їх характеристики, параметри та інші важливі дані;
- 2) забезпечує можливість зберігання та оновлення даних щодо нових алгоритмів та їх параметрів.

б) модуль аналізу та вибору алгоритмів:

- 1) використовує вхідні дані та базу алгоритмів для аналізу та порівняння різних комбінацій алгоритмів;
- 2) генерує результати, які вказують на оптимальні комбінації алгоритмів для конкретних задач логістики.

в) графічний інтерфейс:

- 1) забезпечує зручний спосіб введення даних користувачем;
- 2) дозволяє вибирати алгоритми та переглядати результати в зручному графічному вигляді.

г) генератор звітності:

- 1) надає користувачеві рекомендації щодо найбільш оптимальних комбінацій алгоритмів для вирішення конкретної задачі логістики;
- 2) забезпечує моніторинг роботи системи та збір статистики щодо використання алгоритмів;
- 3) генерує звіти про ефективність використання алгоритмів та часові показники.

Архітектуру системи представлено на рисунку 3.2.

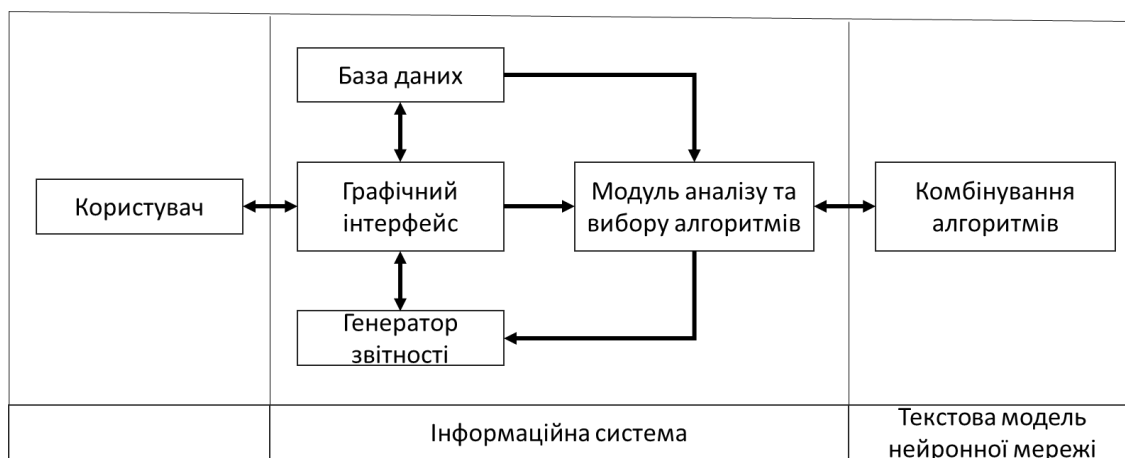


Рисунок 3.2 – Архітектура інформаційної система

На рисунку 3.3 зображено інтерфейс головного вікна.

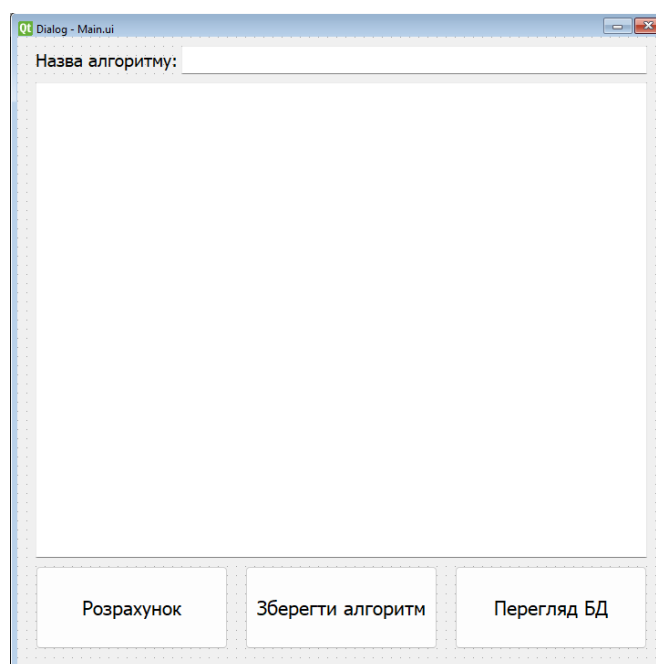


Рисунок 3.3 – Інтерфейс головного вікна

У таблиці 3.3 наведено опис елементів інтерфейсу головного вікна.

Таблиця 3.3 – Опис елементів інтерфейсу

Назва	Функція	Опис
Кнопка "Зберегти алгоритм"	Запис алгоритму в базу даних	При натисканні на цю кнопку введений користувачем алгоритм з поля вводу буде збережений в базі даних для подальшого використання. Ця функція дозволяє зберегти створений алгоритм і використовувати його в майбутніх сеансах роботи з програмою.
Поле "Назва алгоритму"	Визначення назви алгоритму.	Користувач вводить назву алгоритму у це поле. Назва служить для ідентифікації і визначення алгоритмів у базі даних. Це допомагає з легкістю знаходити і вибирати конкретний алгоритм під час подальшої роботи.
Поле вводу алгоритму	Введення текстового опису алгоритму	В цьому полі користувач вписує текстовий опис алгоритму. Це може бути послідовність інструкцій, код, або будь-який інший формат, який описує виконання конкретного алгоритму. Введений текст буде збережений в базі даних після натискання кнопки "Зберегти алгоритм".
Кнопка "Перегляд БД"	Відкриває вікно з записами в базі даних збережених алгоритмів.	При натисканні цієї кнопки відкривається вікно, в якому користувач може переглядати всі збережені в базі даних алгоритми. Це надає можливість легко знаходити, редагувати чи видаляти раніше збережені алгоритми.
Кнопка "Розрахунок"	Відкриває вікно з параметрами комбінування алгоритмів	Натискання цієї кнопки відкриває нове вікно з параметрами, які дозволяють користувачеві комбінувати збережені алгоритми. Це може включати в себе вибір конкретних алгоритмів для використання, встановлення параметрів виконання та інші налаштування, необхідні для старту алгоритмів або комбінації алгоритмів.

Інтерфейс вікна перегляду бази даних представлено на рисунку 3.4.

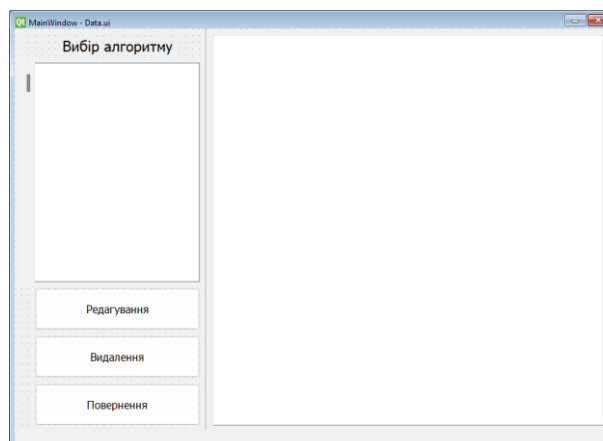


Рисунок 3.4 - Інтерфейс вікна перегляду бази даних

У таблиці 3.4 наведено опис елементів інтерфейсу вікна перегляду бази даних.

Таблиця 3.4 – Опис елементів інтерфейсу

Назва	Функція	Опис
Список алгоритмів	Вибір алгоритмів для перегляду	Цей список містить назви всіх збережених алгоритмів у базі даних. Користувач може вибрати конкретний алгоритм для подальшого перегляду, редагування або видалення.
Поле змісту алгоритму	Перегляд алгоритма без можливості редагування	У цьому полі користувач може переглядати текстовий опис вибраного алгоритму. Однак воно не дозволяє вносити зміни у вміст алгоритму.
Кнопка "Редагування"	Відкриває вікно з можливістю редагування алгоритму	Натискання на цю кнопку відкриває вікно, де користувач може внести зміни у вибраний алгоритм. Вікно містить дві функціональні кнопки: "Запис" для збереження внесених змін і "Відміна" для скасування редагування і повернення до попереднього стану алгоритму.
Кнопка "Видалення"	Видаляє алгоритм з бази даних	При натисканні цієї кнопки вибраний алгоритм буде вилучений із бази даних. Користувач отримує попередження або підтвердження операції видалення залежно від дизайну інтерфейсу.
Кнопка "Повернення"	Повертає до головного інтерфейсу	Натискання на цю кнопку закриває поточне вікно та повертає користувача до головного інтерфейсу програми. Це дозволяє легко переходити між різними функціональними частинами програми.

Інтерфейс вікна з розрахунковими налаштуваннями представлено на рисунку 3.5.

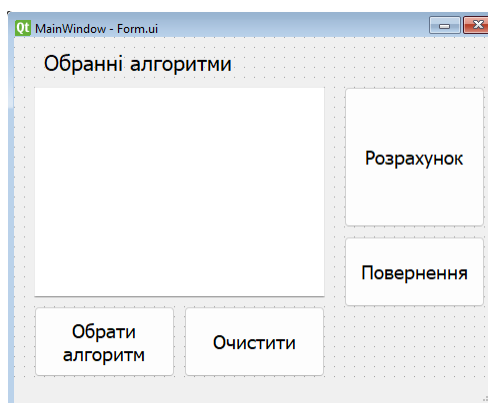


Рисунок 3.5 - Інтерфейс вікна з розрахунковими налаштуваннями.

У таблиці 3.5 наведено опис елементів інтерфейсу вікна з розрахунковими налаштуваннями

Таблиця 3.5 – Опис елементів інтерфейсу

Назва	Функція	Опис
Кнопка "Обрати алгоритм"	Вибір алгоритму для комбінування	При натисканні цієї кнопки користувач отримує доступ до списку всіх збережених алгоритмів, з яких він може обрати ті, які бажає використовувати для комбінування. Це важлива функція для налаштування параметрів комбінації алгоритмів.
Поле "Список алгоритмів"	Перечень обраних алгоритмів для комбінування	У цьому полі відображаються назви алгоритмів, які обрані для подальшої комбінації. Користувач може переглядати і відредагувати цей список з метою належного налаштування комбінацій.
Кнопка "Очистити"	Очищення вибору алгоритмів	Натискання на цю кнопку забезпечує можливість очистити вибір обраних алгоритмів, які були додані у поле "Список алгоритмів". Це дозволяє користувачеві змінювати свій вибір та ефективно налаштувати комбінації.
Кнопка "Розрахунок"	Починає ітерацію розрахунку результатів комбінації алгоритмів	При натисканні цієї кнопки програма розпочинає процес обчислення результатів комбінації обраних алгоритмів. Це може включати в себе виконання алгоритмів в заданому порядку, обчислення результатів та відображення отриманих даних.
Кнопка "Повернення"	Повертає до головного інтерфейсу	Натискання на цю кнопку закриває поточне вікно і повертає користувача до головного інтерфейсу програми. Це дозволяє зручно переміщатися між різними функціональними частинами програми.

Більш детальні алгоритми роботи інформаційної системи наведені у додатку В.

Роздивимося особливості комбінування алгоритмів комбінування алгоритмів за допомогою текстових нейронних мереж — це процес створення більш потужного та ефективного алгоритму, який базується на використанні кількох текстових моделей. Одним з підходів до цього завдання є використання текстових конструкторів.

Текстові конструктори представляють собою інструменти, які дозволяють об'єднувати та змішувати різні моделі та їхні алгоритми для досягнення кращих результатів. Процес комбінування включає кілька етапів:

- вибір різних моделей для комбінування може бути здійснений на основі їхніх сильних сторін та придатності до конкретного завдання.
- токенизація - це процес розбиття тексту на "токени" або найменші змістовні одиниці, такі як слова або підрядки. Це ключовий етап при обробці текстової інформації, оскільки нейронні мережі опрацьовують дані у вигляді токенів. Різні алгоритми можуть використовувати різні методи токенизації, і комбінування моделей може включати в себе використання різних підходів до цього етапу.
- вихідні результати від різних моделей об'єднуються для отримання кінцевого результату. Це може бути здійснено шляхом вагового об'єднання, де кожен вихід моделі має свою вагу, або іншими методами, які залежать від конкретного завдання.
- процес комбінування моделей також може включати настройку параметрів для оптимізації сумісності між різними частинами системи.
- кінцевий об'єднаний алгоритм оцінюється за допомогою тестового набору даних, і процес комбінування може бути вдосконалений за допомогою зміни архітектури чи параметрів.

Деякі додаткові аспекти, які можуть бути важливими при комбінуванні алгоритмів за допомогою текстових нейронних мереж:

- ансамблі - це підхід, при якому кілька моделей навчаються на одних і тих самих даних, і їхні прогнози об'єднуються. Це може бути використано для створення більш стійкого та точного алгоритму.
- якщо моделі використовуються в режимі онлайн, процес навчання може бути динамічним. Алгоритм може навчатися на нових даних та адаптуватися до змінюючогося середовища.
- різні моделі можуть бути призначені для різних аспектів завдання. Наприклад, одна модель може спеціалізуватися на розпізнаванні

сутностей, інша - на аналізі настроїв, а третя - на виокремленні ключової інформації. Потім їхні вихідні результати можуть бути об'єднані для отримання комплексного аналізу.

- у текстових завданнях важливо враховувати контекст, особливо якщо використовуються різні моделі для обробки різних частин тексту. Можливість передавати та використовувати інформацію про контекст може значно поліпшити результати.
- комбінація текстових моделей може бути ефективною, коли її поєднують з іншими технологіями, такими як обробка зображень, звуку чи геоданих, для створення комплексного рішення.

### 3.3 Оцінка ефективності та подальшого розвитку інформаційної системи

Розглянемо конкретний приклад застосування інформаційної системи на підприємства "Еліз". Зазначимо, що до цього на підприємстві не було єдиного логістичного рішення, що могло призвести до неефективного управління та неоптимального використання ресурсів.

Логістична задача на підприємстві «Еліз» включає в себе ефективне управління рухом і виробництвом продукції через різні ділянки. Враховуючи 7 ділянок виробництва, а саме: намотувальний, слюсарний, магнітопровід, монтажний, складальний, випробування і склад готової продукції, ця задача стає складною і важливою для забезпечення ефективності виробництва.

Основні аспекти логістичної задачі на підприємстві «Еліз»:

- кожна продукція має свій унікальний ланцюжок виробництва, який включає ділянки виробництва в порядку, необхідному для завершення виробничого процесу.
- кожен етап виробництва займає певний час  $t$ , і ефективне розподілення часу між різними етапами важливо для підтримки оптимального виробництва.



- кількість шляхів обмежена, і кожен шлях може мати тільки одного навантажувача одночасно. Це важливо для уникнення конфліктів та максимізації потоку продукції.
- після завершення виробництва на одному етапі, продукцію потрібно перевести на інший. Це вимагає оптимального планування і координації транспортних засобів для забезпечення ефективності перевезення.
- тільки продукція, що має вагу більше визначеної, потребує перевезення між ділянками виробництва. Це важливо для оптимізації використання транспортних засобів і ресурсів.
- основна мета логістичної задачі - максимізувати ефективність виробництва, знизити час виробництва та забезпечити найбільш оптимальний розподіл ресурсів для кожного етапу.
- на останньому етапі, продукція готова для відвантаження, і важливо забезпечити ефективне управління складом готової продукції та її доставку з підприємства.
- важливим аспектом логістичної задачі є розробка ефективного плану виробництва і координація всіх етапів. Планування повинно враховувати обмеження часу, обмеження шляхів, доступність робочої сили і ресурсів.

Загальна мета - забезпечити ефективне функціонування всіх етапів виробництва, мінімізувати затримки, забезпечити оптимальний використання ресурсів та забезпечити вчасну поставку готової продукції клієнтам.

Отже, розділимо завдання яке представлено у додатку Г на підзадачі, згідно методу «Розділяю і володарюю». Отриманий результат наведено у таблиці 3.6.

Таблиця 3.6 – Розподіл завдань на підзадачі

Завдання	Підзадача	Вибір алгоритму
<p>На підприємстві "Еліз" необхідно оптимізувати внутрішні маршрути для переміщення сировини, напівфабрикатів та готової продукції між різними цехами. Мета – скоротити час переміщення, зменшити витрати на паливо та забезпечити більш ефективне використання ресурсів підприємства. Розв'язання завдання має враховувати особливості кожного цеху, обмеження швидкості переміщення та безпеки.</p>	<p>Оптимізація маршрутів усередині підприємства „Еліз” для переміщення матеріалів між цехами.</p>	<p>Метод гілок та кордонів</p>
<p>На підприємстві "Еліз" існує необхідність ефективної організації роботи вантажних машин для забезпечення безперебійного процесу завантаження та розвантаження вантажів. Завдання полягає у створенні оптимального розкладу роботи навантажувачів, враховуючи обсяги вантажів, їх пріоритети, доступність вантажної техніки та трудових ресурсів. Розклад має бути гнучким та адаптивним до змін у виробничому процесі, таких як термінові замовлення чи технічні проблеми.</p>	<p>Розробка розкладу роботи вантажних машин на підприємстві "Еліз".</p>	<p>Побудова розкладу за нижньою оцінкою</p>

Розглянемо методи вирішення кожної з підзадач:

- метод гілок та кордонів: Цей метод можна використовувати для пошуку оптимального маршруту між цехами. Ідея полягає в розбитті завдання на підзавдання (гілки) та виключення деяких варіантів маршрутів за допомогою встановлення верхніх та нижніх меж для цільової функції (у даному випадку часу переміщення). На кожному етапі алгоритм вибирає галузь із найбільшим потенціалом оптимізації, що дозволяє звужити пошук та сфокусуватися на найбільш перспективних варіантах маршрутів;
- побудова розкладу за нижньою оцінкою: Цей метод заснований на створенні початкового приблизного розкладу та його подальшої оптимізації. У першому етапі формується нижня оцінка ефективності розкладу, враховуючи основні параметри (обсяги вантажів, пріоритети

тощо.). Потім використовуються методи оптимізації, такі як генетичні алгоритми або алгоритми імітації відпалу, щоб покращити розклад, прагнучи максимізації його ефективності щодо цієї оцінки. Цей процес повторюється до досягнення задовільного рішення.

Алгоритм методу гілок та кордонів складається з кількох кроків. Давайте розглянемо його покроковий опис:

а) ініціалізація

- 1) визначте цільову функцію, яку потрібно оптимізувати (наприклад, час переміщення між цехами);
- 2) розбийте завдання на підзавдання, створивши початкові гілки (можливі маршрути) на дослідження.

б) розгалуження

- 1) виберіть одну із гілок (підзавдань) для подальшого дослідження;
- 2) розбийте вибрану гілку на підзавдання, створивши додаткові гілки.

в) оцінка кордонів

- 1) для кожної гілки оцініть верхню та нижню межі цільової функції;
- 2) якщо верхня межа гілки менша або дорівнює поточному кращому рішенню, припиніть дослідження цієї гілки;
- 3) якщо нижня межа гілки більша за поточну верхню межу, припиніть дослідження цієї гілки (відсікання).

г) перевірка критерію зупинки

- 1) перевірте виконання критеріїв зупинки (наприклад, досягнення певної точності або досягнення максимальної кількості ітерацій);
- 2) якщо критерії зупинки виконані, завершіть алгоритм. Інакше перейдіть до наступного кроку.

д) вибір гілки для подальшого дослідження

1) виберіть наступну галузь дослідження. Це може бути гілка з найкращим потенціалом оптимізації (наприклад, найбільшою різницею між верхньою та нижньою межами).

е) повторення

1) повторюйте кроки з 2 по 5, доки не буде досягнуто оптимального рішення або виконано критерії зупинки.

Алгоритм побудови розкладу за нижньою оцінкою включає кілька етапів. Давайте розглянемо покроковий опис:

а) ініціалізація

1) визначте параметри завдання, такі як обсяги вантажів, пріоритети, доступність ресурсів (вантажних машин) та інші фактори, що впливають на формування розкладу.

б) формування початкового розкладу

1) створіть початковий наближений розклад, враховуючи наявні обмеження та вимоги;

2) визначте часові інтервали для кожного завдання (наприклад, завантаження/розвантаження вантажів).

в) оцінка нижньої оцінки ефективності

1) оцініть нижню оцінку ефективності поточного розкладу. Це може включати підсумовування часу виконання кожного завдання, облік пріоритетів, та інші критерії.

г) оптимізація розкладу

1) застосуйте методи оптимізації, такі як генетичні алгоритми, алгоритми імітації відпалу або локальні пошукові методи для покращення поточного розкладу;

2) у процесі оптимізації враховуйте обмеження та вимоги, а також намагайтеся максимізувати нижню оцінку ефективності.

д) перевірка критеріїв зупинки

1) перевірте виконання критеріїв зупинки, таких як досягнення певної точності, задовільного рівня ефективності розкладу або досягнення максимальної кількості ітерацій.

е) повторення

1) повторюйте кроки з 3 по 5, доки не буде досягнуто задовільного розкладу або не буде виконано критерії зупинки.

У такому випадку отримаємо два варіанти алгоритму ( $X_1$ ,  $X_2$ ).

Роздивимося комбінацію  $X_1$ :

а) ініціалізація

1) визначте цільову функцію для оптимізації (наприклад, час переміщення між цехами) та параметри завдання (обсяги вантажів, пріоритети, доступність ресурсів тощо).

б) метод гілок та кордонів - Розгалуження та Оцінка кордонів

1) розбийте завдання на підзавдання, створивши початкові гілки (можливі маршрути) на дослідження;

2) виберіть галузь дослідження;

3) розбийте вибрану гілку на підзавдання;

4) оцініть верхню та нижню межі цільової функції для кожної гілки.

в) побудова розкладу за нижньою оцінкою - Формування початкового розкладу

1) створіть початковий наближений розклад, враховуючи наявні обмеження та вимоги;

2) визначте часові інтервали для кожного завдання.

г) побудова розкладу за нижньою оцінкою - Оцінка нижньої оцінки ефективності

1) оцініть нижню оцінку ефективності поточного розкладу.

д) метод гілок та кордонів - Відсікання та вибір наступної гілки

1) якщо верхня межа гілки менша або дорівнює поточному кращому рішенню, припиніть дослідження цієї гілки;

- 2) якщо нижня межа гілки більша за поточну верхню межу, припиніть дослідження цієї гілки (відсікання);
  - 3) виберіть наступну галузь для дослідження (наприклад, галузь із найкращим потенціалом оптимізації).
- е) побудова розкладу за нижньою оцінкою - Оптимізація розкладу
- 1) застосуйте методи оптимізації для покращення поточного розкладу;
  - 2) врахуйте обмеження та вимоги, прагнете максимізації нижньої оцінки ефективності.
- ж) перевірка критеріїв зупинки
- 1) перевірте виконання критеріїв зупинки, таких як досягнення певної точності або задовільного рівня ефективності розкладу.
- з) повторення
- 1) повторюйте кроки з 2 по 7, доки не буде досягнуто задовільного розкладу або не буде виконано критерії зупинки.

Роздивимося комбінацію X2:

- а) ініціалізація
- 1) визначте цільову функцію для оптимізації та параметрів завдання.
- б) побудова розкладу за нижньою оцінкою - Формування початкового розкладу
- 1) створіть початковий наближений розклад, враховуючи наявні обмеження та вимоги;
  - 2) визначте часові інтервали для кожного завдання.
- в) побудова розкладу за нижньою оцінкою - Оцінка нижньої оцінки ефективності
- 1) оцініть нижню оцінку ефективності поточного розкладу.
- г) метод гілок та кордонів - Розгалуження та Оцінка кордонів
- 1) розбийте завдання на підзавдання, створивши початкові гілки (можливі маршрути) на дослідження;

- 2) виберіть галузь дослідження;
  - 3) розбийте вибрану гілку на підзавдання;
  - 4) оцініть верхню та нижню межі цільової функції для кожної гілки.
- д) побудова розкладу за нижньою оцінкою - Оптимізація розкладу
- 1) застосуйте методи оптимізації для покращення поточного розкладу;
  - 2) врахуйте обмеження та вимоги, прагнете максимізації нижньої оцінки ефективності.
- е) метод гілок та кордонів - Відсікання та вибір наступної гілки
- 1) якщо верхня межа гілки менша або дорівнює поточному кращому рішенню, припиніть дослідження цієї гілки (відсікання);
  - 2) якщо нижня межа гілки більша за поточну верхню межу, припиніть дослідження цієї гілки (відсікання);
  - 3) виберіть наступну галузь для дослідження (наприклад, галузь із найкращим потенціалом оптимізації).
- ж) перевірка критеріїв зупинки
- 1) перевірте виконання критеріїв зупинки, таких як досягнення певної точності або задовільного рівня ефективності розкладу.
- з) повторення
- 1) повторюйте кроки з 2 по 7, доки не буде досягнуто задовільного розкладу або не буде виконано критерії зупинки.

Перетворимо кожний з представлених алгоритмів у програмний код, запустимо його на віртуальній машині. На основі вхідних даних представлених додатку Г - запишемо результати у таблицю 3.7.

Таблиця 3.7 – Порівняння результатів роботи алгоритмів

Задача	Кількість ітерацій	Час пошуку рішення	Сума результатів
X1	46 234	3 хв. 46 сек.	59
X2	24 585	1 хв. 29 сек.	13

Отже, на підставі вищезазначеного можемо здійснити висновок, що в умовах комбінації алгоритмів "Метод гілок та кордонів" та "Побудова розкладу за нижньою оцінкою", ефективніше використовувати стратегію, за якою спочатку проводиться обчислення маршруту.

Слід звернути увагу на недосконалість системи, що виявляється через низку виявлених недоліків:

- різноманітність алгоритмів: слід зазначити, що завдання логістики не мають єдиного оптимального алгоритму. У розділі 2, було розглянуто понад двадцять унікальних алгоритмів з пошуку оптимального маршруту. Кожен алгоритм має свій власний результат, який залежить від вхідних даних та умов завдання. У представленій інформаційній системі, використовується одна комбінація алгоритмів, що обмежує різноманіття варіантів розв'язання задачі;
- комбінування алгоритмів є складнішим процесом, ніж проста зміна порядку їх застосування. Ефективність такої комбінації залежить від правильного встановлення взаємозв'язків між алгоритмами. Представлена інформаційна система не завжди враховує оптимальні взаємозв'язки між алгоритмами. Комбінування однієї і тієї задачі може мати різний вигляд.



## ВИСНОВКИ

В даній науковій роботі, на основі проведених досліджень і розробок, вдалося вирішити проблеми комбінованих логістичних задач. Зокрема, було здійснено аналіз сучасних логістичних задач, що виникають в різних галузях діяльності підприємств. Визначено та описано термін "комбінована задача транспортної логістики", а також висвітлено основні елементи логістичних завдань.

На наступних етапах дослідження був проведений аналіз існуючих моделей та інформаційних систем, спрямованих на розв'язання комбінованих логістичних задач. Детально розглянуто дві інформаційні системи, які вже існують в цій сфері.

Засновуючись на вивченні існуючих методів і алгоритмів розв'язання логістичних задач, використовуючи метод "Розділай і володарюй", була розроблена модель розв'язання комбінованих логістичних задач.

Далі, на основі розробленої моделі, була створена інформаційна система підтримки прийняття рішень для вирішення комбінованих задач транспортної логістики.

Важливим кроком у дослідженні була експериментальна перевірка результатів. Ефективність та придатність запропонованої моделі рішень та інформаційної системи були підтвержені через проведення експериментів.

Додатково, у продовженні дослідження можна розглянути кілька напрямків для подальших вдосконалення:

- подальший аналіз та оптимізація використовуваних алгоритмів може покращити швидкодію та точність розробленої моделі. Врахування новітніх алгоритмів машинного навчання та оптимізаційних підходів може сприяти підвищенню продуктивності.
- розгляд можливості розширення функціональності інформаційної системи для вирішення більш широкого спектру задач логістики. Це може включати у себе інтеграцію з іншими інформаційними

системами, розширення варіантів аналізу даних та підтримку нових видів логістичних викликів.

- розгляд можливості автоматичної адаптації моделі до змінних умов ринку та логістичних процесів. Це може включати у себе розробку алгоритмів, які автоматично адаптуються до змін у попиті, постачанні, та інших факторів.
- забезпечення високого рівня безпеки та конфіденційності даних у системі є критичним аспектом. Розгляд можливостей для вдосконалення заходів безпеки та врахування сучасних стандартів в цій області.
- проведення пілотних проєктів для розгортання розробленої інформаційної системи в реальних умовах діяльності підприємств дозволить оцінити її реальну продуктивність та виявити можливі обмеження.
- розгляд можливостей врахування впливу запропонованої моделі та інформаційної системи на сталий розвиток підприємств. Оцінка їх впливу на зменшення витрат, оптимізацію ресурсів та екологічні параметри може бути важливою для бізнес-спільноти.

Загальний висновок полягає в тому, що розроблені модель та інформаційна система успішно вирішують комбіновані логістичні задачі, а їх ефективність була підтверджена експериментально. Це може мати важливе значення для підприємств та організацій, які стикаються з подібними логістичними викликами.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сокур І. М. Транспортна логістика : навч. посіб. Київ : ЦУЛ, 2022. 220 с.
2. Смирнов І. Г. Транспортна логістика: навч. посіб. Київ : ЦУЛ, 2008. 224 с.
3. Пономарьова Ю.В. Логістика : навч. посіб. Київ. : ЦУЛ, 2005. 328 с.
4. Implementation Challenge: Vehicle Routing. URL: <http://dimacs.rutgers.edu/programs/challenge/vrp/>
5. B. Golden, E. Wasil, J. Kelly, and I. Chao. The impact of metaheuristics on solving the vehicle routing problem. *Fleet Management and Logistics*. New York, NY : Springer, 1998. Pp. 33–56.
6. N. Christofides, S. Eilon. Algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*. 1969. Pp. 309-318.
7. ROADEF. URL: <https://www.roadef.org/challenge/2022/en/index.php>
8. J. Homberger, H. Gehring. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *Information Systems and Operational Research* №37. 1999. Pp. 297–318.
9. M. Grazia Speranza, L. Bertazzi. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*. 2012. Pp. 307-326.
10. Michael R. Garey, David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY: W. H. Freeman & Co. 1979. Page 338.
11. Павленко А. Ф., В. В. Кривещенко. Логістичні інформаційні системи. 2011. 291-299 с.
12. Ямковой М.В. Математичне моделювання ланцюга поставок для вантажних перевезень у рамках логістичної системи. *Комп'ютерно-інтегровані технології, Луцьк 2023. Випуск № 51. 2023. 75-19 с.*

13. Julien Bramel, David Simchi-Levi. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*. New York, NY: Springer. 2014. Page 447.
14. Зварич М. В. Огляд парадигми розробки алгоритму «розділяй і володарюй». *Актуальні задачі сучасних технологій*. Тернопіль 27-28 листопада 2019 року. С. 37.
15. Douglas R. Smith. The design of divide and conquer algorithms. *Science of Computer Programming №5*. 1985. Pp. 37-58.
16. D. Coppersmith, S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*. 1990. Pp. 251 – 280.
17. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959. Pp. 269–271.
18. R. Bellman. *Dynamic Programming*. Princeton, NJ: Princeton University Press. 2010. P. 392.
19. E. V. Denardo. *Dynamic Programming: Models and Applications*. Upper Saddle River, NJ: Dover Publications. 2003. P. 240.
20. M. Sniedovich. Dijkstra's algorithm revisited : the dynamic programming connexion. *Journal of Control and Cybernetics*. 2006. Pp. 599–620.
21. M. Sniedovich. *Dynamic Programming: Foundations and Principles*. Boca Raton, FL: CRC Press. 2010. P. 616.
22. M. Fredman, R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*. 1987. Pp. 338–346.
23. M. Fredman, D. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Proceedings 31st Annual Symposium on Foundations of Computer Science*. 1990. Pp. 719–725.
24. M. Fredman, D. Willard. Surpassing the information theoretic bound with fusion trees. *Journal of computer and system sciences*. 1993. Pp. 424–436.
25. D. Dor, S. Halperin, U. Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*. 2000. Pp. 124–131.

26. S. Cabello. Many distances in planar graphs. *Algorithmica*. 2012. Page 364-381.
27. E. Boas, R. Kaas, E. Zijlstra. Design and implementation of an efficient priorityqueue. *Mathematical Systems Theory*. 1976. Pp. 99–127.
28. E. Boas. Preserving order in a forest in less than logarithmic time. Amsterdam: University of Amsterdam. 1975. P. 84.
29. M. Thorup. On RAM priority. *SODA*. 1996. Pp. 59–67.
30. T. Hagerup. Improved shortest paths on the word RAM. *Automata, Languages and Program-ming*. 2000. Pp. 61–72.
31. R. Bellman. On a routing problem, quarterly of Applied Mathematics. Princeton, NJ: Princeton University Press. 1958. Page 90.
32. R. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*. 2000. Pp. 309–311.
33. J. Y. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*. 1970. Pp. 526–530.
34. M. J. Bannister, D. Eppstein. Randomized Speedup of the Bellman-Ford Algorithm. California: University of California. 2011. Page 12.
35. D. Karger, D. Koller, S. Phillips. Finding the hidden path: time bounds for all-pair shortest paths. *Proceedings 32nd Annual Symposium of Foundations of Computer Science*. 1993. Pp. 560–568.
36. S. Warshall. A theorem on boolean matrices. *Journal of the ACM*. 1962. Page 32-39.
37. M. Fredman. New bounds on the complexity of the shortest path problem. *SIAM*. 1976. Pp. 83–89.
38. A. V. Aho, J. E. Hopcroft. The Design and Analysis of Computer Algorithms. Boston, MA: Addison-Wesley Longman Publishing. 1974. Page 480.
39. Y. Han, T. Takaoka. An  $o(n^3 \log \log n / \log^2 n)$  time algorithm for all pairs shortest paths. *Proceedings of the 13th Scandinavian conference on Algorithm Theory*. 2012. Pp. 131–141.

40. P. Hart, N. Nilsson, B. Raphael. Formal basis for the heuristic determination of minimumcost paths. *Systems Science and Cybernetics*. 1968. Pp. 100–107.
41. A. Goldberg, R. Werneck. Computing Point-to-Point Shortest Paths from External Mem-ory. Vancouver: ALENEX/ANALCO. 2005. Page 22.
42. R. Gutman. Reach-Based Routing: A New Approach to Shortest Path. Algorithms Optimizedfor Road Networks. New Orleans: ALENEX/ANALCO. 2004. Page 12.
43. M. Potamias, F. Bonchi, C. Castillo, A. Gionis. Fast shortest path distance estimationin large networks. *Proceeding of the 18th ACM conference on Information and knowledgemanagement*. 2009. Pp. 867-870.
44. J. Kleinberg, A. Slivkins, T. Wexler. Triangulation and Embedding Using Small Sets of Beacons. *45th Annual IEEE Symposium on Foundations of Computer Science*. 2004. Pp. 444–453.
45. E. K. Schilling, R. H. Mohring. Fast point-to-point shortest path computationswith arc-flags. *DIMACS Challenge*. 2006.
46. R. Bauer, D. Delling, P. Sanders, D. Schieferdecker, D. Schultes, D. Wagner. Combin-ing hierarchical and goal-directed speed-up techniques for dijkstra’s algorithm. *Journal of Experimental Algorithmics*. 2010. Pp. 303–318.
47. D. Delling, T. Pajor, R. Werneck. Round-based public transit routing. *Proceedings of the Fourteenth Workshop on Algorithm Engineering and Experiments*. 2012. Pp. 130–140.
48. J. Maue, P. Sanders, D. Matijevic. Goal-directed shortest-path queries using precomputedcluster distances. *Journal of Experimental Algorithmics*. 2009. Pp. 14–29.
49. D. Schultes. Fast and exact shortest path queries using highway hierarchies. Saarlandes: Universit“at des Saarlandes. 2005. Page 67.
50. P. Sanders, D. Schultes. Highway hierarchies hasten exact shortest path queries. *Algorithms*. 2005, Pp. 568–579.

51. R. Geisberger, P. Sanders, D. Schultes, D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. *Experimental Algorithms*. 2008. Pp. 319–333.
52. G. Batz, R. Geisberger, S. Neubauer, P. Sanders. Time-dependent contraction hierarchies and approximation. *Experimental Algorithms*. 2010. Pp. 166–177.
53. R. Geisberger, P. Sanders, D. Schultes, C. Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*. 2012. Pp. 388–404.
54. F. Schulz, D. Wagner, C. Zaroliagis. Using multi-level graphs for timetable information in railway systems. *Algorithm Engineering and Experiments*. 2002. Pp. 43–59.
55. M. Holzer, F. Schulz, D. Wagner. Engineering multilevel overlay graphs for shortest-path queries. *Journal of Experimental Algorithmics*. 2009. Pp. 173–179.
56. H. Bast, S. Funke, D. Matijevic, P. Sanders, D. Schultes. In Transit to Constant Time Shortest-Path Queries in Road Networks. *ALLENEX*. 2007. Pp. 89–98.
57. C. Gavoille, D. Peleg, S. P'erenes, R. Raz. Distance labeling in graphs. *Algorithms*. 2004. Pp. 85–112.

## ДОДАТОК А

### Огляд інтерфейсу інформаційної системи ILS

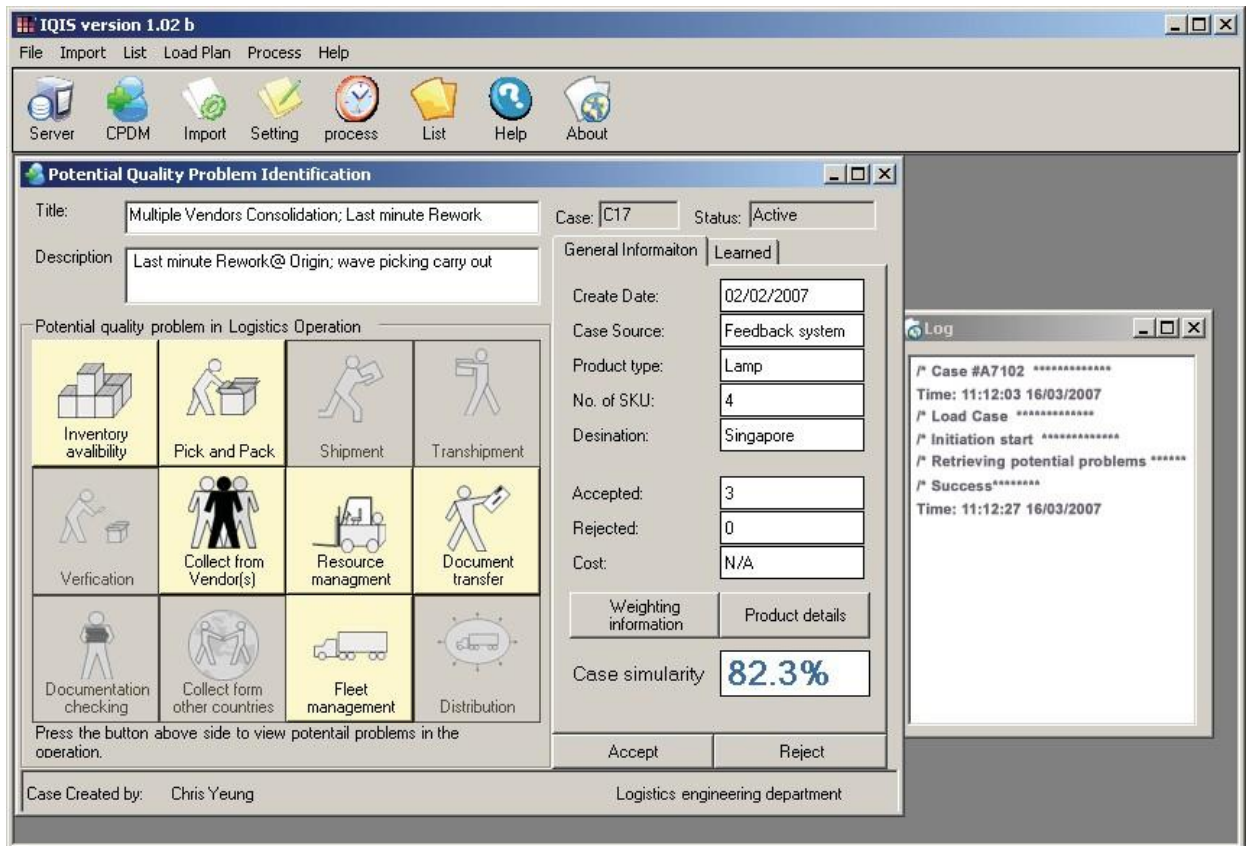


Рисунок А.1 – Інтерфейс системи ILS



## ДОДАТОК Б

### Алгоритм роботи інформаційної системи ILS

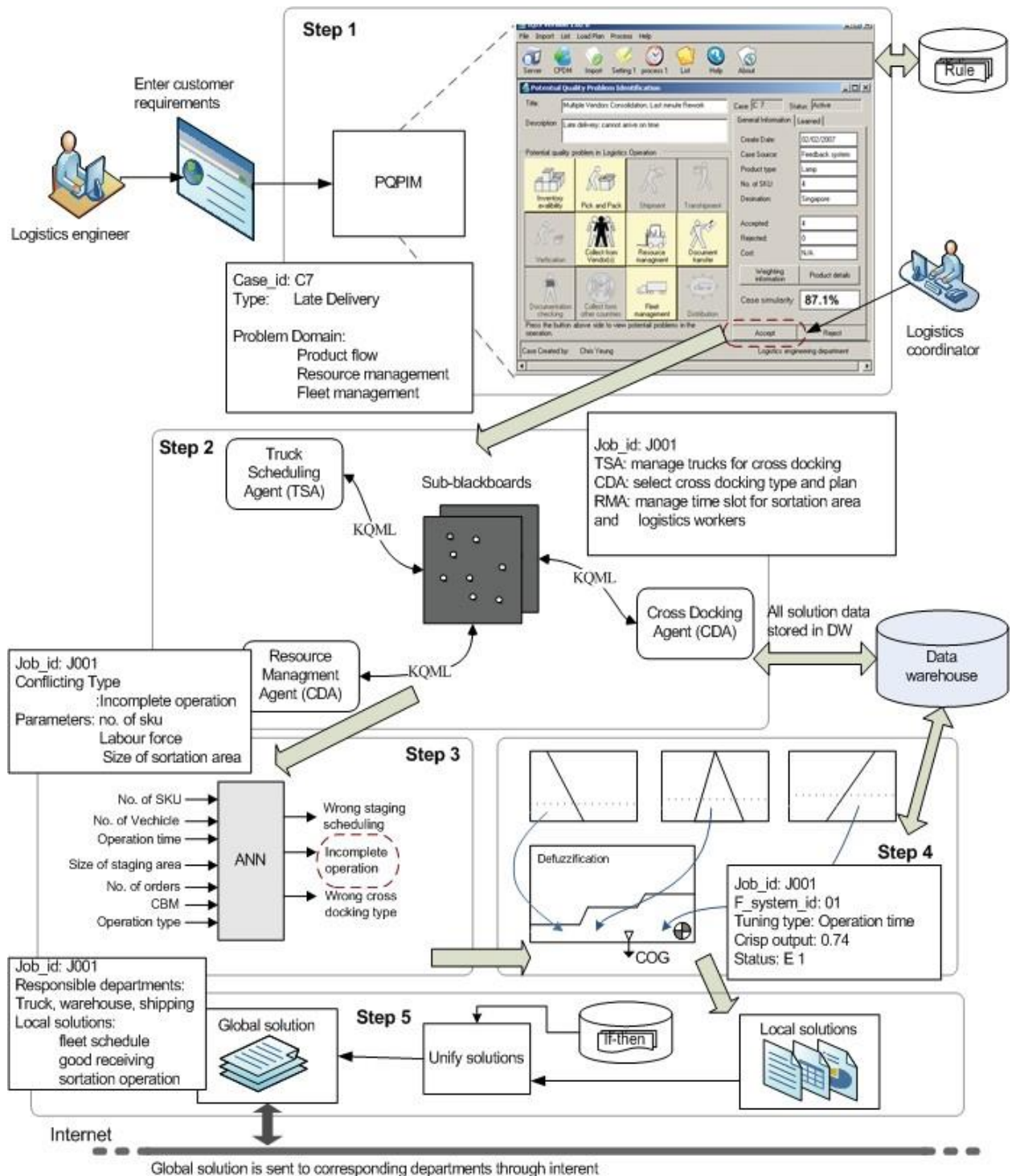


Рисунок Б.1 – Алгоритм роботи системи ILS

**ДОДАТОК В****Алгоритм роботи інформаційної системи**

Рисунок В.1 – Схема роботи інформаційної системи

## ДОДАТОК Г

## Вхідні данні для аналізу ефективності інформаційної системи

Supplier code	Product code	Number of items	Length	Width	Height	Weight	Nesting height	Inventory cost	Max stackability
0062069400	762432908R	3	780	570	478	81,72	47	2	4
0062069400	663606500R	3	780	570	478	96,256	47	3	4
0029918200	550206080R	6	1200	1000	975	577,8	45	7	100
0028091900	403007072R	10	1400	1225	1160	278,5	0	15	100
0062069400	745326954R	5	780	570	478	100,96	47	1	4
0029938000	272704463R	16	1400	1200	1450	179,4	45	10	100
0062069400	852480062R	1	1600	1200	975	1471,6	45	26	100
0062069400	764131108R	1	1900	1200	750	713,62	45	13	100
0029938000	272707927R	12	1400	1200	1450	179,4	45	10	100
0062060800	172022957R	8	1600	1200	1350	303,92	45	5	100
0028091900	403008854R	11	1400	1225	1160	317	0	16	100
0062069400	763571615R	1	1200	1000	750	227,528	45	5	100
0062069400	901323145R	1	780	570	478	101,52	47	3	4
0062069400	625189279R	1	780	570	478	270,08	47	6	4
0025439300	802102463R	3	1600	1200	930	454,873	45	8	100
0062069400	8201677033	3	1600	1200	930	547,3	45	1	100
0062060800	172039318R	6	1600	1200	1350	303,92	45	5	100
0062069400	8201677033	4	1600	1200	930	547,3	45	1	100
0062060800	172039318R	2	1600	1200	1350	303,92	45	5	100
0002893800	769235146R	1	1200	1000	975	246	45	4	100
0062069400	8201677103	2	1200	1000	545	729,5	45	2	100
0062060800	172022957R	1	1600	1200	1350	303,92	45	5	100
0062069400	776507388R	3	1200	1000	750	229,88	45	4	100
0062069400	8201590963	1	780	570	478	160,56	47	1	4
0029938000	272704463R	12	1400	1200	1450	179,4	45	10	100
0062069400	625191617R	1	780	570	478	269,32	47	6	4
0062060800	172022957R	7	1600	1200	1350	303,92	45	5	100
0029938000	272707927R	31	1400	1200	1450	179,4	45	10	100
0062060800	172038007R	17	1600	1200	1350	207,72	45	5	100
0062069400	843408889R	1	1200	1000	750	267,8	45	6	100
0029938000	272704463R	32	1400	1200	1450	179,4	45	10	100
0062069400	8201745175	11	1600	1200	930	438,4	45	1	100
0029938000	272704463R	35	1400	1200	1450	179,4	45	10	100
0062060800	172022957R	8	1600	1200	1350	303,92	45	5	100

Рисунок Г.1 – Фрагмент вхідних даних для предметів

Id truck	Length	Width	Height	Max weigl	Max densi	Max weigl	Cost	EMmm	EMmr	CM	Cjfm	Cjfc	Cjfh	EM	EJhr	Ejcr	Eleh
P230711301	14500	2400	2800	30000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P188731303	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P188731301	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P199731001	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P200731301	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P200731303	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P237708201	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P196709601	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P216731001	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P235711301	14500	2400	2800	30000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P202730501	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P202730503	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670
P202730505	14940	2500	2950	24000	1500	100000	1500	12000	31500	7808	3800	1040	3330	7300	7630	2350	1670

Рисунок Г.2 – Фрагмент вхідних даних для транспорту

## ДОДАТОК Д

### Результат прохождения Unicheck



Имя пользователя: Кафедра економічної кібернетики Максишко Натал...	ID проверки: 1015998613
Дата проверки: 12.12.2023 18:17:26 EET	Тип проверки: Doc vs Internet
Дата отчета: 12.12.2023 18:18:13 EET	ID пользователя: 100005897

Название файла: Зарубін О.С.\_8\_0512-ек

Количество страниц: 71 Количество слов: 14157 Количество символов: 108275 Размер файла: 381.58 KB ID файла: 1015681675

## 14% Совпадения

Наибольшее совпадение: 6.63% с Интернет-источником (<http://cit-journal.com.ua/index.php/cit/issue/download/36/47>)

14% Источники из Интернета 288 ..... Страница 73

Поиск совпадений с Библиотекой не производился

## 0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

## 0% Исключений

Нет исключенных источников

## Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 15

Рисунок Д.1 - Результат прохождения Unicheck

**Декларація академічної доброчесності  
здобувача вищої освіти ЗНУ**

Я Зарубін Олексій Станіславович, студент другого курсу, форми навчання денна, факультету економічного, спеціальності 051 – Економіка, адреса електронної пошти [omgkingkarl@gmail.com](mailto:omgkingkarl@gmail.com),

- підтверджую, що написана мною кваліфікаційна робота на тему *«Інформаційна система підтримки прийняття рішень комбінованої задачі транспортної логістики»* відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою Інтернет-системи, а також на архівування роботи в базі даних цієї системи.

Дата \_\_\_\_\_

Підпис \_\_\_\_\_

ПБ (студент) Зарубін О. С.

Дата \_\_\_\_\_

Підпис \_\_\_\_\_

ПБ (керівник) Козін І.В.