**АНОТАЦІЯ**

**Структура та обсяг роботи.** Пояснювальна записка індивідуальної частини № 1 комплексного дипломного проекту складається з чотирьох розділів, містить 112 сторінок, 33 рисунків, 13 таблиці, 3 додатки, 21 джерел(о).

Індивідуальна частина № 2 присвячена складанню плану перевезень однорідної продукції із урахування вантажомісткості транспортних засобів, їх обмеженої кількості та пріоритетів замовлень.

У розділі загальних положень наведена постановка задачі.

У розділі з математичного забезпечення наведена математична постановка задачі, обґрунтовано обрані підході для розв’язання. Були розроблені алгоритм для розв’язання задачі складання плану перевезень однорідної продукції із урахуванням вантажомісткості транспортних засобів. Проведено порівняльний аналіз алгоритмів на основі отриманих експериментальних даних.

У технологічному розділі наведена інструкція користувача та наведена методика випробувань програмного продукту.

У розділі з охорони праці наведені гігієнічні норми для приміщень, у яких експлуатуються ЕОМ.

МАРШРУТ, ЦИКЛ, ЗАДАЧА МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ, ВАНТАЖОМІСТКІСТЬ, ПРІОРИТЕТ, ЗАДАЧА КЛАСТЕРИЗАЦІЇ, МЕТАЕВРИСТИКА, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, БДЖОЛИНИЙ АЛГОРИТМ, ЗАДАЧА КОМІВОЯЖЕРА.

**ABSTRACT**

**The structure and scope of paper.** Explanatory note of the individual part № 1 of the complex thesis project consists of 4 sections, contains 112 pages, 10 figures, 13 tables, 3 appendices, 21 references.

The individual part № 2 devoted to transportation in consideration of cargo capacity and limited number of vehicles and in consideration of order priority.

In the part of general provisions problem formulation is given.

The section of mathematical support contains mathematical formulation of the problem and approaches to problem solving reasoning. Algorithms for solving the problem of the homogenous products transporting in consideration of cargo capacity of vehicles were developed. A comparative analysis based on experimental data was performed.

The software part contains specification of functions described in detail.

In the section of labor protection hygiene requirements to the room, where computers are used, are given.

ROUTE, LOOP, VEHICLE ROUTING PROBLEM, CAPACITY, PRIORITY, DATA CLUSTERING, METAHEURISTIC, MULTI‑OBJECTIVE OPTIMIZATION, BEES ALGOTIRHM, TRAVELING SALESMAN PROBLEM.

**ЗМІСТ**

[ВСТУП 8](#_Toc325271401)

[1 ЗАГАЛЬНІ ПОЛОЖЕННЯ 10](#_Toc325271402)

[Висновок до розділу 10](#_Toc325271403)

[2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕНЯ 11](#_Toc325271404)

[2.1 Змістовна постановка задачі 11](#_Toc325271405)

[2.2 Математична постановка задачі 11](#_Toc325271406)

[2.3 Обґрунтування методу розв’язання 14](#_Toc325271407)

[2.4 Опис 2-етапного алгоритму для розв’язання CVRPP 15](#_Toc325271408)

[2.4.1 Перший етап – кластеризація вершин графу 15](#_Toc325271409)

[2.4.1.1 Багатокритеріальна оптимізація 16](#_Toc325271410)

[2.4.1.2 Скаляризація векторної цільової функції 18](#_Toc325271411)

[2.4.1.3 Алгоритм ланцюга найближчих сусідів 20](#_Toc325271412)

[2.4.1.4 Бджолиний алгоритм кластеризації 20](#_Toc325271413)

[2.4.2 Другий етап – побудова маршрутів відповідно до проведеної кластеризації 21](#_Toc325271414)

[2.5 Порівняльний аналіз отриманих результатів 21](#_Toc325271415)

[Висновок до розділу 23](#_Toc325271416)

[3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ 25](#_Toc325271417)

[3.1 Архітектура програмного забезпечення 25](#_Toc325271418)

[3.1.1 Специфікація функцій 25](#_Toc325271419)

[Висновок до розділу 36](#_Toc325271420)

[4 РОЗДІЛ З ОХОРОНИ ПРАЦІ 37](#_Toc325271421)

[4.1 Загальні вимоги до виробничих приміщень 37](#_Toc325271422)

[4.2 Характеристика робочого місця 39](#_Toc325271423)

[4.3 Мікроклімат 40](#_Toc325271424)

[4.4 Характеристика випромінювання 41](#_Toc325271425)

[4.5 Освітлення 41](#_Toc325271426)

[4.6 Виробничий шум 42](#_Toc325271427)

[4.7 Електробезпека 42](#_Toc325271428)

[4.8 Пожежна безпека 43](#_Toc325271429)

[4.9 Інструкція із техніки безпеки при роботі із ПК 44](#_Toc325271430)

[Висновок до розділу 44](#_Toc325271431)

[ЗАГАЛЬНІ ВИСНОВКИ 45](#_Toc325271432)

[ПЕРЕЛІК ПОСИЛАНЬ 47](#_Toc325271433)

[ДОДАТОК А Графічний матеріал 48](#_Toc325271434)

# ВСТУП

Логістика – частина економічної науки, предмет якої полягає в організації раціонального процесу просування товарів і послуг від постачальників сировини до споживачів, функціонування сфери обігу продукції, товарів, послуг, управління товарними запасами, створення інфраструктури руху товару. Логістика спрямована на оптимізацію витрат і раціоналізацію процесу виробництва, збуту і супутнього сервісу як в рамках одного підприємства, так і для групи підприємств.

У індивідуальній частині № 2 розглядається задача складання плану перевезень продукції із урахуванням вантажомісткості та обмеженої кількості транспортних засобів, а також пріоритетів замовлень.

Призначенням комплексу задач є забезпечення логіста засобами автоматизованого створення плану перевезень продукції з мінімальними сумарними витратами.

Метою індивідуальної частини № 2 є складання плану перевезень однорідної продукції із урахуванням вантажомісткості транспортних засобів, що дозволить зменшити сумарні витрати на перевезення та штрафи за неврахування пріоритетів замовлень. Для досягнення цієї мети необхідно вирішити такі задачі, як розподілення клієнтів між транспортними засобами та складання маршрутів для кожного транспортного засобу відповідно до виконаного розподілення.

**Практичне значення одержаних результатів.** Розроблено евристичні, метаевристичні алгоритми, а також їх гібридні модифікації для розв’язання задач, що у сукупності складають собою задачу маршрутизації транспортних засобів із урахуванням вантажомісткості транспортних засобів та їх обмеженої кількості. Проведено порівняльний аналіз ефективності роботи розроблених методів на основі отриманих експериментальних даних.

# ЗАГАЛЬНІ ПОЛОЖЕННЯ

## Висновок до розділу

У даному розділі описані загальні положення індивідуальної частини № 2 дипломного проекту. Надано детальний опис предметного середовища поставленої задачі.

Також визначене призначення та мета комплексу, а також перелічені задачі, які необхідно розв’язати для досягнення встановленої мети.

# МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕНЯ

## Змістовна постановка задачі

Логістична компанія здійснює перевезення однорідної продукції, наприклад, зерна або вугілля, із складів до клієнтів. Клієнти та склади розташовані у різних містах. Кожний склад має однорідний парк транспортних засобів, вантажомісткість яких є обмеженою.

Компанія має виконати певну кількість замовлень, але наявна на момент виконання цих замовлень кількість транспортних засобів є недостатньою для обслуговування усіх клієнтів. За негайне виконання замовлення компанія отримує оплату за терміновість. Це може бути як фіксована ставка, так і відсоток від замовлення.

Необхідно скласти такий план перевезень, який з одного боку зменшуватиме витрати на перевезення продукції, а з іншого збільшить сумарні виплати за терміновість.

Умовно можна вважати, що компанія виплачує штраф клієнтам, яких не обслуговує негайно. Тоді задача полягає у приписуванні маршрутів окремим транспортними засобам, за якого сумарні затрати на перевезення продукції та виплату штрафів будуть мінімальними.

Кожний транспортний засіб має починати свій маршрут із міста, у якому розташований склад, та закінчувати його у тому ж місті.

## Математична постановка задачі

Постановка класичної задачі маршрутизації транспортних засобів (VRP) наведена у загальній частині дипломного проекту у п. 3.2.1.. Наведемо основні відмінності задачі маршрутизації транспортних засобів з урахуванням вантажомісткості та обмеженої кількості транспортних засобів, та пріоритетів замовлень (Capacitated VRP with Priority– CVRPP) від класичної VRP.

Автопарк складається із обмеженої кількості однакових наявних на даних момент транспортних засобів, вантажомісткість яких . Із кожним споживачем пов’язане замовлення .

Зрозуміло, що мінімальна кількість транспортних засобів, необхідних для виконання усіх замовлень, дорівнює:

|  |  |
| --- | --- |
|  | (2.1) |

Але часто буває так, що компанія володіє меншою кількістю транспортних засобів . Тоді необхідно відмовитись від частини замовлень, таким чином, щоб з одного боку зменшити витрати на перевезення, а з іншого збільшити виплати за терміновість. Застосовуючи інше формулювання задачі необхідно скласти такий план перевезень, щоб сумарні затрати на перевезення та затрати, пов’язані із виплатою штрафів були мінімально можливими.

Із кожним споживачем пов’язаний штраф , який зобов’язана виконати логістична компанія у разі невиконання замовлення цього споживача.

Наведемо математичне формулювання CVRPP (для випадку наявності одного складу) у вигляді задачі цілочисельного лінійного програмування. Використаємо наступну нотацію:

* – бінарна змінна, що приймає значення 1, якщо маршрут ‑ого транспортного засобу включає безпосередній переїзд із міста у місто , і значення 0 у іншому випадку;
* – бінарна змінна, що приймає значення 1, якщо замовлення ‑го споживача виконане -им транспортного засобу, і значення 0 у іншому випадку.
* – бінарна змінна, що приймає значення 1, якщо -ий споживач не обслуговується негайно (тобто з цим споживачем пов’язані збитки/штраф), і значення 0 у іншому випадку.

Цільова функція – мінімізація сумарних затрат на перевезення продукції вартості та затрат, пов’язаних із виплатою штрафів:

|  |  |
| --- | --- |
|  | (2.2) |

Обмеження:

|  |  |
| --- | --- |
|  | (2.3) |
|  | (2.4) |
|  | (2.5) |
|  | (2.6) |
|  | (2.7) |
|  | (2.8) |
|  | (2.9) |
|  | (2.10) |
|  | (2.11) |
|  | (2.12) |

Обмеження (2.3) гарантують, що кожний споживач буде відвіданий не більше одного разу (або відвіданий один раз одним транспортним засобом, або не відвіданий зовсім). Обмеження (2.4) забезпечують виїзд транспортного засобу від кожного відвіданого ним споживача. Обмеження (2.5) гарантують виїзд кожного транспортного засобу із міста-складу, забезпечуючи таким чином використання усіх наявних транспортних засобів. Обмеження (2.6) унеможливлюють утворення у розв’язку маршрутів, які не містять склад. Обмеження (2.7) гарантують, що клієнт обслуговується транспортним засобом , тільки якщо він проїжджає через місто . Обмеження (2.8) унеможливлюють можливість перевезення транспортним засобом продукції обсягом більше ніж їх вантажомісткість. Обмеження (2.9) встановлюють збитки (штрафи) для вузлів, які не були обслужені.

## Обґрунтування методу розв’язання

При розроблені алгоритму розв’язання задачі CVRPP були використані результати розв’язання задачі CVRP, яка є задачею маршрутизації транспортних засобів із урахуванням із вантажомісткості та визначення оптимальної кількості необхідних транспортних засобів (див. розділ 2 індивідуальної частини № 1 дипломного проекту).

Враховуючи ефективність розроблених для CVRP алгоритмів було вирішено використовувати, ті самі підходи. І розробити подібні алгоритми, які будуть враховувати відмінності CVRPP від CVRP.

Обраний метод складається з 2 етапів: кластеризації вершин та складання маршрутів відповідно до проведеної кластеризації.

Для задачі кластеризації були модифіковані методи ланцюга найближчих сусідів та бджолиного алгоритму кластеризації для CVRP (див п. 2.4.1.1 індивідуальної частини № 1).

Для другого етапу складання маршрутів застосовується бджолиний алгоритм розв’язання задачі комівояжера, оскільки для розв’язання даної задачі він на практиці показав себе як дуже ефективний метод.

## Опис 2-етапного алгоритму для розв’язання CVRPP

### Перший етап – кластеризація вершин графу

Постановка задачі кластеризації наведена у п. 3.4.2.1.1 загальної частини дипломного проекту. Основною проблемою задачі маршрутизації транспортних засобів із врахуванням вантажомісткості та пріоритетів замовлень є те, що необхідно враховувати не тільки вартість перевезень, а й збитки (штрафи) від відмови негайного обслуговування.

Вершини графу необхідно розбити на кластерів, із яких кластери відповідають клієнтами, які терміново обслуговуються наявними транспортними засобами. Кластер містить усіх клієнтів, обслуговування яких буде здійснюватися пізніше.

Зменшення витрати на перевезення, досягається зменшенням сумарної довжини утворених маршрутів. Для розбиття вершин за територіальною ознакою критерієм однорідності може слугувати функція мінімізації сумарного квадратичного відхилення точок кластерів (а також вершини-складу) від їхніх центрів:

|  |  |
| --- | --- |
|  | (2.13) |

де – центр мас кластеру .

Більш детальна інформація про кластеризацію точок за територіальною ознакою наведена у загальній частині дипломного проекту у п. 3.4.2.1.1.

З іншого боку необхідно зменшити затрати пов’язані із накладанням штрафу у разі нетермінового здійснення замовлення:

|  |  |
| --- | --- |
|  | (2.14) |

Наведені цільові функції (2.12) та (2.13) є конфліктуючими і до того ж пов’язані із різними одиницями вимірювання. Тобто маємо задачу багатокритеріальної оптимізації. Опишемо загальну постановку задачі багатокритеріальної оптимізації та підходи до її розв’язання.

#### Багатокритеріальна оптимізація

Багатокритеріальна оптимізація – процес одночасної оптимізації двох або більші конфліктуючих функцій у заданій області визначення. Задача багатокритеріальної оптимізації формулюється наступним чином:

|  |  |
| --- | --- |
|  | (2.15) |
|  | (2.16) |

Задача багатокритеріальної оптимізації полягає у пошуку вектора цільових змінних, що задовольняють накладеним обмеженням і оптимізують векторну функцію, елементі якої відповідають цільовим функціям. Ці функції є математичним описом критерію задовільності і, як правило, взаємно конфліктують. Тому у даному випадку «оптимізувати» означає знайти такий розв’язок, за якого значення цільових функцій задовольнятимуть постановника задачі.

Для можливості оцінювання якості знайдених розв’язків зазвичай розглядають наступні точки у області значення цільової функції.

* ідеальна точка ;
* утопічна точка ;
* надир ;

Ідеальна точка визначається як вектор , кожна координата якої має оптимальне значення відповідної складової цільової функції:

|  |  |
| --- | --- |
|  | (2.17) |

Точка надир є «найнижчою» точкою і визначається наступним чином

|  |  |
| --- | --- |
|  | (2.18) |

Утопічну точку обчислюють на основі ідеальної:

|  |  |
| --- | --- |
|  | (2.19) |

де , – одиничний вектор.

Критерій Парето. Вектор розв’язку називається оптимальним за Парето , якщо не існує такого, що для усіх та , хоча б для одного . Цільовий вектори є оптимальним за Парето, якщо відповідний йому вектор із області визначення також оптимальний за Парето. Діапазон значень оптимальних за Парето розв’язків у області допустимих значень надає корисну інформацію про досліджувану задачу, якщо цільові функції обмежені областю визначення. Нижні границі оптимальної за Парето множини представлені у «ідеальному» цільовому векторі . Його компоненти отримані шляхом мінімізації кожної цільової функції у межах області визначення.

Для отримання оптимальних за Парето розв’язків часто використовують методи скаляризації. Оскільки цільова функція задачі багатокритеріальної оптимізації має векторні значення, її перетворюють у функцію із скалярним значення. Таким чином, задача багатокритеріальної оптимізації зводиться до задачі з однією скалярною цільовою функцією. Розглянемо деякі методи скаляризації:

* метод зважених сум – у якості скалярної цільової функції використовується сума окремих зважених цільових функцій;
* методи зміни обмежень – одна із цільових функції залишається у якості єдиного критерію ефективності, а інші перетворюються на обмеження.

#### Скаляризація векторної цільової функції

Для скаляризації векторної цільової функції, координати якої є цільовими функціями (2.13) та (2.14) було вирішено застосувати метод зважених сум. Тобто необхідно сформулювати єдину цільову функцію у вигляді:

|  |  |
| --- | --- |
|  | (2.20) |

де та – цільові функції (2.13) та (2.14) відповідно;

, – вага цільових функцій та відповідно.

Було висунуто припущення, що між довжиною маршруту у кластері та сумарним квадратичним відхиленням точок кластеру від його центру існує функціональна залежність. Тоді, по критерію (див. формулу (2.13)) можна оцінити довжину маршруту, а ,отже, і вартість здійснення перевезення за цим маршрутом.

Враховуючи що функція із врахуванням її ваги приводиться до грошових одиниць, вагу функції можна вважати рівною .

Тобто загальну цільову функцію можна представити у вигляді:

|  |  |
| --- | --- |
|  | (2.21) |

де – функція, що оцінює довжину маршруту у кластерів за сумарним квадратичним відхиленням точок кластеру від його центру;

– вартість однієї одиниці довжини маршруту.

Величина є фіксованою. Наприклад, це може бути вартість бензину, затраченого на 100 км тощо. Основним завданням для скаляризації векторної цільової функції у даному випадку є виявлення функціональної залежності . Для цього була проведена серія експериментів, у результаті якої було отримано результат, зображений на рисунку 2.1.

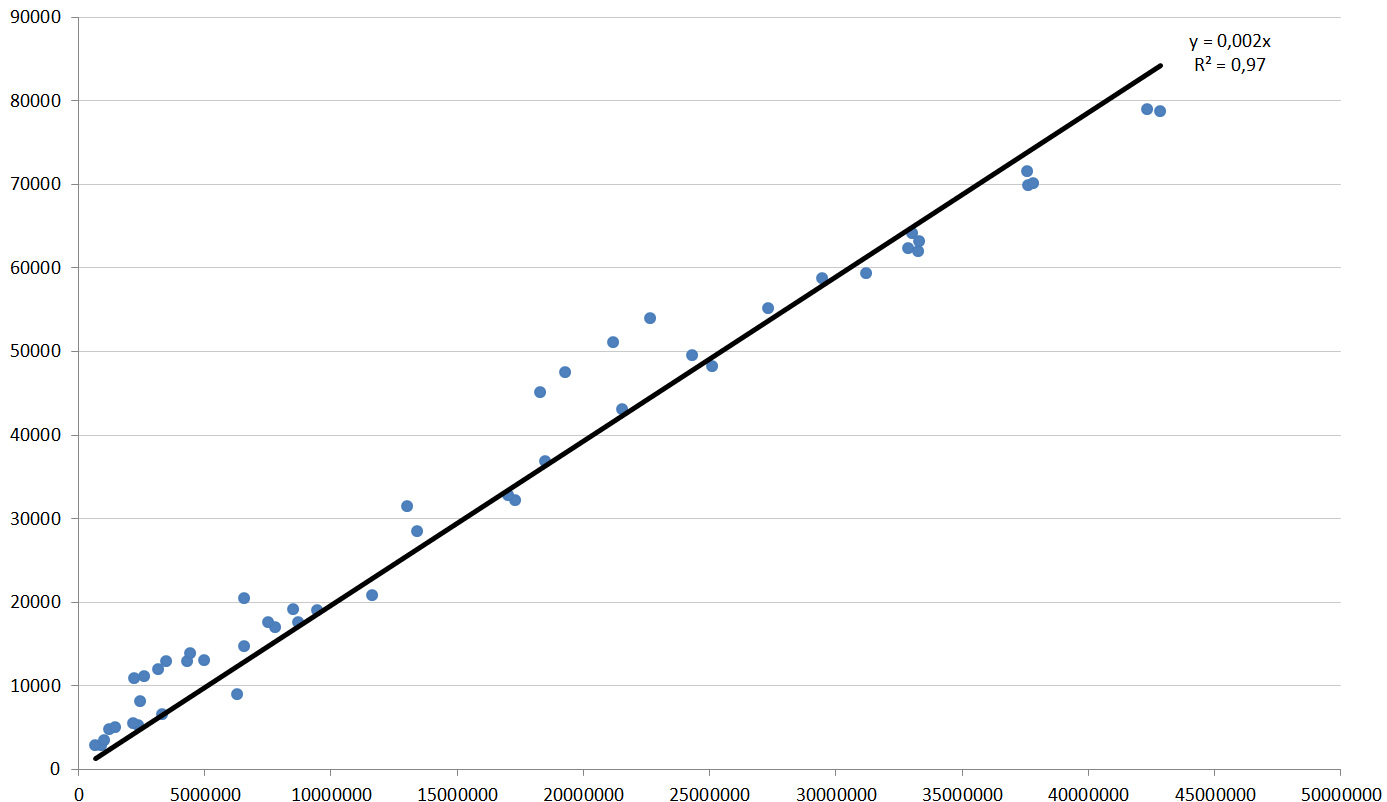


Рисунок 2.1 – Функціональна залежність між довжиною маршруту у кластері (вертикальна вісь) та сумарним квадратичним відхиленням точок кластеру від його центру (горизонтальна вісь)

Як видно з рисунку залежність є лінійною, тобто:

|  |  |
| --- | --- |
|  | (2.22) |

Програмними засобами було визначено що ця залежність із точністю 97 % має вигляд . Тобто в середньому довжина маршруту у кластері у 500 разів менша за сумарне квадратичне відхилення точок кластеру від його центру.

Отже, загальна цільова функція має вигляд:

|  |  |
| --- | --- |
|  | (2.23) |

Нижче на рисунку 2.2 наведений графік, який підтверджує висунуту гіпотезу. На графіку зображено реальна довжина маршруту у кластері та оцінки цієї довжини отримана на основі сумарного квадратичного відхилення точок кластеру від його центру за формулою (2.23).

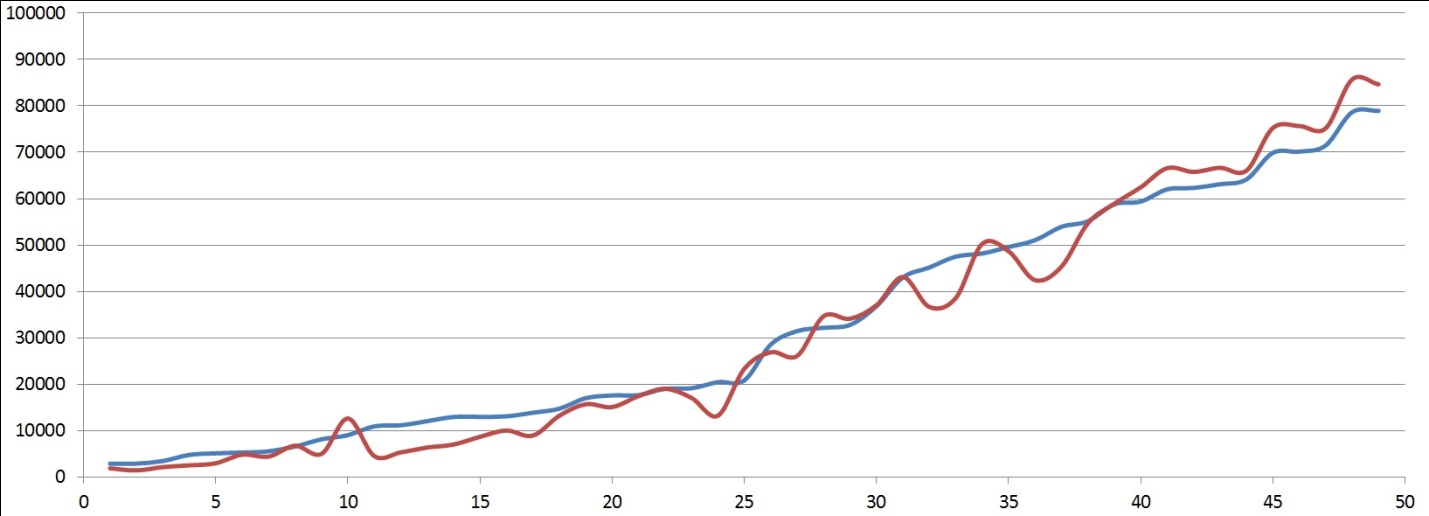


Рисунок 2.2 – Порівняння реальною довжини маршруту у кластері (блакитна крива) та оцінки цієї довжини (червона крива)

#### Гібридна модифікація алгоритму ланцюга найближчого сусіда та бджолиного алгоритму

Оскільки кількість транспортних засобів у випадку задачі CVRPP є обмеженою застосування алгоритму ланцюга найближчого сусіда, розробленого для CVRP (детальнішу інформацію див. у п. 2.4.1.1.1 індивідуальної частини № 1 дипломного проекту), є неможливим.

Тому вирішено було модифікувати його. Для цього після запуску алгоритму ланцюга найближчого сусіда, застосовується бджолиний алгоритм вибору кластерів. Розпишемо цей крок більш детально.

Після застосування алгоритму ланцюга найближчого сусіда, множина вершини розбивається на кластерів.

Розв’язки при застосуванні бджолиного алгоритму (загальну інформацію про бджолиний алгоритм див. у п. 3.4.1.1 загальної частини дипломного проекту) формуються наступним чином: випадковим чином із кластерів обирається . Обрані кластерів відповідають клієнтам, що будуть обслужені відповідними транспортними засобами, решта кластерів об’єднуються у кластер клієнтів, які будуть обслужені наступного разу. Для сформованого розв’язку критерій ефективності обчислюється за формулою (2.23). Метою бджолиного алгоритму є знаходження такого розв’язку, який мінімізує цільову функцію (2.23). Сусідніми розв’язки формуються заміною одного із обраних кластерів, випадковим кластером, що не був обраний для обслуговування.

#### Бджолиний алгоритм кластеризації

Бджолиний алгоритм кластеризації для CVRPP розбиває задану множину вершин на кластерів, із яких кластери відповідають клієнтами, які терміново обслуговуються наявними транспортними засобами. Кластер містить усіх клієнтів, обслуговування яких буде здійснюватися пізніше.

Початкові розв’язки формуються наступним чином: випадкові вершини із множини вершин додаються до кластеру до тих пір, поки дозволяє місткість кластеру. Як тільки кластер заповнюється, створюється новий, який так само заповнюється випадковим чином. Як тільки заповнився -ий кластер, формування розв’язку завершується, усі точки, що не ввійшли до жодного із створених кластерів формують кластер .

Стратегія пошуку сусідніх розв’язків така сама як у випадку бджолиного алгоритму кластеризації для задачі CVRP (див. п. 2.4.1.1.2 індивідуальної частини № 1 дипломного проекту).

Розв’язки оцінюється за сумарними витратами, затраченими на перевезенням та на виплату штрафів.

### Другий етап – побудова маршрутів відповідно до проведеної кластеризації

Другий етап даного алгоритму розв’язання CVRP полягає у побудові маршрутів для кластерів . Він є таким самим як і другий етап даного алгоритму для розв’язання VRP. Для знаходження окремих маршрутів необхідно розв’язати задачу комівояжера для кластерів (див. п. 3.4.2.2 загальної частини дипломного проекту).

## Порівняльний аналіз отриманих результатів

Метою індивідуальної частини № 2 дипломного проекту була не просто розробка методу розв’язання задачі складання плану перевезень однорідної продукції із врахуванням вантажомісткості та обмеженої кількості засобів та пріоритетів замовлення із ціллю зменшення сумарних витрат на перевезення та виплату штрафів, а розробка ефективного методу розв’язку, тобто такого, що буде знаходити прийнятні розв’язки за прийнятний час. Тому було розроблено декілька алгоритмів, і у цьому випадку важливим пунктом є порівняльний аналіз створених алгоритмів.

Аналіз алгоритмів був проведений на основі експериментальних даних, отриманих для задач із різними початковими вхідними даними. Аналіз алгоритмів проводився за двома напрямами: точність розв’язку, який він забезпечує, та час за який цей розв’язок було отримано.

Експерименти проводились на випадкових моделях. Кожен із алгоритмів запускався на одній і тій самі моделі декілька разів, після чого оцінювалось середнє значення отриманого результату та середній час, затрачений на знаходження розв’язку.

У таблиці 2.1 наведені результати проведення експериментів.

Таблиця 2.1 – Результати застосування розроблених алгоритмів на різних моделях задачі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Кількість замовників** | **2-х етапний метод із застосуванням бджолиного алгоритму для кластеризації** | | **2-х етапний метод із застосуванням модифікованого алгоритму ланцюга найближчих сусідів** | |
| **Розв’язок** | **Час** | **Розв’язок** | **Час** |
| Кількість транспортних засобів – 3 шт., вантажомісткість – 50 | | | | |
| 10 | 114,54 | 0,09 | 126,49 | 0,11 |
| 15 | 153,39 | 0,24 | 147,5 | 0,16 |
| 20 | 189,27 | 0,41 | 183,8 | 0,26 |
| 25 | 224,51 | 0,46 | 222,34 | 0,46 |
| 30 | 296,3 | 0,53 | 297,75 | 0,53 |
| 35 | 323,66 | 0,67 | 319,66 | 0,56 |
| 40 | 358,3 | 0,8 | 359,55 | 0,62 |
| Кількість транспортних засобів – 3 шт., вантажомісткість – 150 | | | | |
| 20 | 119,67 | 0,15 | 125,04 | 0,15 |
| 25 | 133,53 | 0,38 | 146,58 | 0,39 |
| 30 | 176 | 0,77 | 177,14 | 0,48 |
| 35 | 161,96 | 0,78 | 162,15 | 0,59 |
| 40 | 190,05 | 1,27 | 198,01 | 1 |
| 45 | 216,19 | 1,69 | 214,27 | 1,22 |
| 50 | 274,5 | 2,2 | 277,66 | 1,36 |
| 55 | 307,35 | 2,91 | 298,78 | 1,34 |
| 60 | 375,8 | 3 | 370,86 | 1,96 |
| 65 | 368,17 | 3,39 | 358,6 | 2,34 |
| 70 | 444,18 | 4,03 | 439 | 1,46 |
| 75 | 523,87 | 5,22 | 515,78 | 3,7 |
| 80 | 561,76 | 8,2 | 561,32 | 3,19 |

Продовження табл. 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Кількість замовників** | **2-х етапний метод із застосуванням бджолиного алгоритму для кластеризації** | | **2-х етапний метод із застосуванням модифікованого алгоритму ланцюга найближчих сусідів** | |
| **Розв’язок** | **Час** | **Розв’язок** | **Час** |
| Кількість транспортних засобів – 4 шт., вантажомісткість – 150 | | | | |
| 70 | 357,96 | 7,57 | 366,8 | 4,29 |
| 75 | 472,29 | 9,24 | 454,04 | 5,37 |
| 80 | 476,9 | 12,33 | 470,75 | 3,23 |
| 85 | 520,31 | 18,21 | 518,68 | 5,08 |
| 90 | 569,35 | 18,12 | 567,21 | 6,53 |
| 95 | 631,7 | 24,16 | 623,15 | 6,83 |
| 100 | 634,89 | 22,73 | 627,7 | 4,83 |
| 105 | 745,96 | 26,43 | 738,95 | 3,48 |
| 110 | 759,53 | 18,91 | 758,65 | 7,7 |
| 115 | 800,83 | 35,05 | 804,26 | 12,19 |
| 120 | 871,69 | 55,12 | 866,58 | 13,2 |
| Кількість транспортних засобів – 4 шт., вантажомісткість – 250 | | | | |
| 100 | 402,61 | 33,34 | 395,38 | 9,95 |
| 105 | 408,07 | 25,23 | 405,77 | 7,93 |
| 110 | 459,63 | 37,64 | 474,15 | 17,61 |
| 115 | 479,89 | 61,76 | 485,78 | 26,9 |
| 120 | 506,57 | 56,85 | 501,43 | 16,01 |
| 125 | 648,03 | 68,58 | 653,56 | 16,54 |
| 130 | 634,58 | 82,53 | 633,01 | 27,66 |
| 135 | 688,07 | 83,45 | 695,15 | 21,73 |
| 140 | 737,59 | 145,46 | 730,35 | 58,58 |
| 145 | 806,79 | 93,56 | 806,47 | 14,61 |
| 150 | 820,01 | 152,84 | 816,53 | 58,83 |

Продовження табл. 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Кількість замовників** | **2-х етапний метод із застосуванням бджолиного алгоритму для кластеризації** | | **2-х етапний метод із застосуванням модифікованого алгоритму ланцюга найближчих сусідів** | |
| **Розв’язок** | **Час** | **Розв’язок** | **Час** |
| Кількість транспортних засобів – 4 шт., вантажомісткість – 350 | | | | |
| 130 | 423,64 | 74,52 | 419,11 | 14,96 |
| 135 | 524,19 | 108,52 | 536,46 | 23,37 |
| 140 | 574,76 | 112,67 | 587,56 | 17,99 |
| 145 | 674,2 | 138,56 | 678,4 | 25,39 |
| 150 | 687,38 | 138,72 | 691,2 | 60,53 |
| 155 | 715,75 | 178,19 | 712,65 | 51,32 |
| 160 | 766,91 | 157,6 | 755,9 | 84,93 |
| 165 | 861,1 | 167,2 | 849,29 | 73,32 |
| 170 | 867,94 | 270,21 | 875,92 | 119,49 |
| 175 | 919,81 | 276,93 | 920,24 | 105,04 |
| 180 | 901,17 | 187,43 | 901,48 | 111,63 |
| 185 | 1036,15 | 328,99 | 1041,64 | 74,75 |
| 190 | 1097,87 | 251,53 | 1100,32 | 66,56 |
| 195 | 1111,74 | 399 | 1136,01 | 36,24 |
| 200 | 1051,42 | 250,49 | 1063,84 | 72,09 |

У додатку А, лист ??? (???) наведені порівняльні графіки застосування розроблених алгоритмів для розв’язання задачі маршрутизації транспортних засобів із різноманітними вхідними даними.

Із графіків стає зрозуміло, що чим кращий розв’язок видає алгоритм, тим більше часу він потребує на знаходження цього розв’язку, при чому при лінійному зменшенні знайденого розв’язку, час пошуку цього розв’язку росте логарифмічно.

З точки зору як точності отриманого результату, так і затраченого часу кращим виявися 2-етапний метод із застосуванням гібридної модифікації алгоритму ланцюга найближчого сусіда із бджолиним алгоритмом.

## Висновок до розділу

Розділ з математичного забезпечення присвячений формулюванню змістовної та математичної постановки задачі маршрутизації транспортних засобів із врахуванням вантажомісткості транспортних засобів, їх обмеженої кількості та пріоритетів замовлень, розробленню математичного апарату для розв’язання цієї задачі.

У розділі детально описано розроблені методи та алгоритми розв’язання задачі та обґрунтовано їх вибір.

Також важливим пунктом розділу з математичного забезпечення є проведення порівняльного аналізу розроблених алгоритмів із метою визначення найбільш ефективного з них.

# ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

## Засоби розробки

При створенні програмного продукту середовищем розробки було обрано Visual Studio 2010 для програмування мовою C#, використовуючи плагін ReSharper.

Дане середовище дозволяє розробляти як консольні застосунки, так і застосунки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторингу коду. Вбудовані інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу програми, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні доповнення (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду, додавання нових наборів інструментів.

Плагін ReSharper дуже зручно використовувати при розробці мовою C#. Він проводить статичний аналіз коду (пошук помилок в коді до компіляції) в масштабі всього рішення, передбачає додаткові засоби автозаповнення, навігації, пошуку, підсвічування синтаксису, форматування, оптимізації та генерації коду, надає 40 автоматизованих видів рефакторинга.

C# – сучасна мова програмування, яка має дуже багато переваг перед такими мовами, як C/C++ або Java. Ось неповний їх список:

* C# створювався паралельно з платформою Microsoft .NET Framework і повною мірою враховує всі його можливості;
* C# є повністю об’єктно-орієнтованою мовою, де навіть типи, вбудовані в мову, представлені класами;
* C# є потужною об’єктною мовою з можливостями наслідування і універсалізації;
* C# є спадкоємцем мов C/C++, зберігаючи кращі риси цих популярних мов програмування. Спільний з цими мовами синтаксис, знайомі оператори мови полегшують перехід програмістів від С++ до C#;
* мова C# зберігла основні риси C++, але в той же час стала простіше і надійніше. Простота і надійність, головним чином, пов’язані з тим, що на C# хоча і допускаються, але не заохочуються такі небезпечні властивості С++ як покажчики, адресація, розіменування, адресна арифметика;
* завдяки платформі Microsoft .NET Framework, що стала надбудовою над операційною системою, програмісти C # отримують ті ж переваги роботи з віртуальною машиною, що і програмісти Java. Ефективність коду навіть підвищується, оскільки виконавча середа CLR є компілятором проміжної мови, в той час як віртуальна Java-машина є інтерпретатором байт-коду;
* потужна бібліотека платформи підтримує зручність побудови різних типів застосунків на C#, дозволяючи легко будувати Web-служби, інші види компонентів, достатньо просто зберігати і отримувати інформацію з бази даних.

Для збереження інформації було обрано систему управління реляційними базами даних Microsoft SQL Server 2008 R2.

Це сучасна відмовостійка СУБД, яку дуже зручно використовувати при розробці застосунків на основі платформи Microsoft .NET Framework. Платформа надає спеціальні засоби для роботи з Microsoft SQL Server.

Ця СУБД використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства.

В якості системи контролю версій було використано розподілену систему керування версіями файлів git.

Використання розподіленої системи контролю версій дає наступні переваги (над централізованими):

* періодична синхронізація декількох комп'ютерів під управлінням одного розробника (робочого комп'ютера, домашнього комп'ютера, ноутбука і так далі). Використання розподіленої системи позбавляє від необхідності виділяти один з комп'ютерів в якості сервера, а синхронізація виконується за потребою;
* спільна робота над проектом невеликою територіально розподіленою групи розробників без виділення загальних ресурсів. Як і в попередньому випадку, реалізується схема роботи без головного сервера, а актуальність репозиторіїв підтримується періодичними синхронізації за схемою «кожен з кожним».
* великий розподілений проект, учасники якого можуть довгий час працювати кожен над своєю частиною, при цьому не мають постійного підключення до мережі. Такий проект може використовувати централізований сервер, з яким синхронізуються копії всіх його учасників. Можливі й більш складні варіанти - наприклад, зі створенням груп для роботи за окремими напрямками всередині більшого проекту. При цьому можуть бути виділені окремі «групові» сервери для синхронізації роботи груп, тоді процес остаточного злиття змін стає деревовидним: спочатку окремі розробники синхронізують зміни на групових серверах, потім оновлені репозиторії груп синхронізуються з головним сервером. Можлива робота і без «групових» серверів, тоді розробники однієї групи синхронізують зміни між собою, після чого будь-який з них (наприклад, керівник групи) передає зміни на центральний сервер.

Використання git дає наступні переваги (над аналогічними розподіленими системами контролю версій):

* висока продуктивність;
* продумана система команд;
* репозиторії git можуть поширюватися і оновлюватися загальносистемними файловими утилітами резервного копіювання та оновлення, завдяки тому, що фіксації змін і синхронізації не змінюють існуючі файли з даними, а лише додають нові (за винятком деяких службових файлів, які можуть бути автоматично оновлені за допомогою наявних у складі системи утиліт);
* для поширення репозиторію по мережі досить будь-якого веб-сервера.

## Вимоги до технічного та програмного забезпечення

Головною метою розробки даного програмного продукту є обробка даних про складські приміщення, об’єми товарів і замовлень з метою отримання плану перевезень між складами і споживачами. Він призначений для використання співробітником підприємства, що працює у відповідному напрямі.

Для можливості роботи з даним програмним продуктом до складу технічних засобів повинні входити:

1. комп’ютер з наступною конфігурацією:

1) процесор з тактовою частотою не нижче 1.4 ГГц;

1. об’єм оперативної пам’яті не менше 512 МБ;
2. наявність 20 Мб вільного простору на жорсткому диску;
3. графічна карта з об’ємом відео-пам’яті не менше 32 Мб;
4. додатково має бути встановлене наступне програмне забезпечення:
5. операційна система Windows XP/Vista/7;
6. СУБД Microsoft SQL Server 2008 R2;
7. платформа Microsoft .NET Framework 4.0;
8. комп’ютерна периферія, до складу якої входить:
9. монітор;
10. мишка;
11. клавіатура.

## Архітектура програмного забезпечення

Діаграма класів, послідовності та компонентів програмного забезпечення наведені у п. 4.1 загальної частини дипломного проекту. Діаграма класів наведена у додатку В, лист?? (Схема структурна класів) загальної частини дипломного проекту.

Даний розділ присвячений опису функцій розроблених класів програмного забезпечення.

### Специфікація функцій

Опис функцій основних класів програмного забезпечення наведений у таблиці 3.1.

Таблиця 3.1 – специфікація функцій класів програмного забезпечення.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
| Algorithm – абстрактний клас алгоритму, в якому реалізована спільна для ітеративних алгоритмів логіка | public abstract void DrawNodes() | Функція, яка реалізовує графічне відображення вузлів |
| public virtual string Info() | Функція повертає специфічну інформацію про поточний стан алгоритму |
| public virtual void StartingInitialize() | Ініціалізація на початку алгоритму бджолиної колонії для першого розв’язку |
| protected abstract void InnerIteration() | Специфічна ітерація для певного алгоритму |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | private void Iteration() | Проведення специфічної ітерації а також збір загальної статистики по алгоритму, ведення логу |
| public void Iterations (int count = 1) | Проведення певної кількості ітерацій |
| public void IterateToStop() | Виконання ітерації доки не буде досягнуто критерію зупину |
| private void OpenLogFile() | Відкриття лог-файлу для дозапису |
| public void SetNodes (List<Node> nodesForSet) | Задання вузлів для обробки у алгоритмі |
| public void Stop() | Зупинка алгоритму |
| BeesColony – клас-нащадок класу Algorithm. Клас, в якому реалізовано бджолиний алгоритм в загальному вигляді | private Site CreateNewSite() | Створює випадковий розв’язок задачі |
| public void CreateSites() | Генерація початкового набору розв’язків |
| public override void DrawNodes() | Функція, яка реалізовує графічне відображення вузлів |
| public override string Info() | Повертає інформацію про поточний стан алгоритму |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | protected override void InnerIteration() | Ітерація бджолиного алгоритму |
| Site – абстрактний клас, який представляє розв’язок бджолиного алгоритму а також реалізовує спільну логіку пошуку сусідніх розв’язків і переходу до кращого з них | public virtual int CompareTo(object obj) | Функція порівняння даного об’єкту з іншим |
| public void DrawNodes() | Графічне відображення |
| private List<Site> GenerateNeighbours (int count) | Генерація сусідніх розв’язків |
| protected abstract Site GetNeighbour() | Функція генерації сусіднього розв’язку |
| public bool GoToBestNeighbour (int countOfNeightbours) | Перехід до кращого сусіднього розв’язку, якщо такий є |
| protected abstract void GoToNeighbour (Site site) | Перехід до сусіднього розв’язку |
| public abstract List<Node> PrepareToDraw (Color connectionsColor) | Підготувати вузли до графічного відображення |
| SiteVrpTsp – клас-нащадок класу Site. Клас розв’язку задачі VRP, що зводиться до TSP | private void ConnectConsumerToDepot (Node node1, Node node2, Color connectionColor) | Графічно приєднати вузол-споживач до вузла-складу. |
| private static double[,] GeneratePricesByPositions (List<Node> nodes, int depotsCount, int consumersCount) | Згенерувати таблицю відстаней |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | private void GenerateSequence (int depotsCount, int clustersCount, int consumersCount) | Згенерувати послідовність обходження вузлів |
| private static double GetDistance (Node node1, Node node2) | Обчислення відстані від одного вузла до іншого |
| protected override Site GetNeighbour () | Згенерувати сусідній розв’язок |
| protected override void GoToNeighbour (Site site) | Перейти до сусіднього розв’язку |
| private static void Interchange (int[] arr, int i1, int i2) | Поміняти місцями елементи масиву |
| private static void Invert (int[] arr, int i1, int i2) | Інвертувати частину масиву |
| public void InvertRandomPartOfNodesSequence () | Інвертувати випадкову частину масиву |
| public override List<Node> PrepareToDraw (Color connectionsColor) | Підготувати вузли до графічного відображення |
| public override string ToString() | Функція повертає відображення розв’язку у вигляді рядку |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
| SiteClusteringVrp – клас-нащадок класу Site. Клас розв’язку задачі кластеризації для VRP | protected List<Node> GenerateClusters() | Генерація кластерів випадковим чином |
| protected virtual List<Node> GenerateClusters (List<Node> nodesForClusters, Node \_depot) | Генерація кластерів випадковим чином (специфічна для нащадків) |
| protected static void MoveNodeFromOneClusterToAnother (Cluster cluster1, Cluster cluster2) | Перемістити вузол з одного кластеру в інший |
| public static void MoveNodeFromOneClusterToAnother (List<Cluster> clusters) | Перемістити вузол з одного кластеру у колекції до іншого |
| SiteClusteringCvrp – клас-нащадок класу SiteClusteringVrp. Клас розв’язку задачі кластеризаціїї для CVRP | private void AddCluster() | Додати пустий кластер до списку кластерів |
| protected static bool ExchangeNodesInClusters (Cluster c1, Cluster c2) | Перемістити вузол з одного кластеру в другий і з другого в перший |
| public static bool ExchangeNodesInClusters (List<Cluster> clusters) | Перемістити вузли між двома кластерами у списку |
| protected override List<Node> GenerateClusters (List<Node> nodesForClusters, Node \_depot) | Генерація кластерів випадковим чином |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | protected override Site GetNeighbour() | Згенерувати сусідній розв’язок |
| SiteClusteringCvrpp – клас-нащадок класу SiteClusteringCvrp. Клас розв’язку задачі кластеризації для CVRPP | private void FormAllClusters() | Формування списку усіх кластерів |
| protected override Site GetNeighbour() | Згенерувати сусідній розв’язок |
| public static double GetPrice (double length, double kilometerCost, double fines) | Функція повертає ціну перевезення з урахуванням штрафів |
| protected override void GoToNeighbour(Site site) | Перейти до сусіднього розв’язку |
| public override List<Node> PrepareToDraw (Color connectionsColor) | Підготувати вузли до графічного відображення |
| SiteClusteringCvrppNnc – клас-нащадок класу SiteClusteringCvrpp. Клас розв’язку задачі кластеризації алгоритмом Bees CVRPP+NNC | public override void StartingInitialize() | Функція, що виконується на початку бджолиного алгоритму. Обчислення початкового набору кластерів, використовуючи алгоритм near-neighbor chain |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | private void GenerateClustersFromNnc() | Генерація кластерів розв’язку на основі початкового набору кластерів, обчислених алгоритмом near-neighbor chain |
| Node – клас вузла. Зберігає інформацію, що стосується вузла | public void ConnectTo (Node node) | Приєднати поточний вузол до іншого вузла |
| public void ConnectTo (Node node, Color color) | Приєднати поточний вузол до іншого вузла, враховуючи колір з’єднання |
| public void DisconnectFromAll() | Від’єднати поточний вузол від усіх інших вузлів |
| ClusteringAlgorithm – абстрактний клас-нащадок класу Algorithm. Реалізує спільну логіку для алгоритмів кластеризації | protected void AddCluster() | Додати пустий кластер до списку кластерів |
| public override void DrawNodes() | Графічне відображення вузлів, оброблених алгоритмом |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
| KMeans – клас-нащадок класу ClusteringAlgorithm. Клас реалізує логіку алгоритму кластеризації k‑means | private void GenerateClusters() | Генерація початкового набору кластерів |
| protected override void InnerIteration() | Ітерація алгоритму |
| private void SetCenters() | Перемістити центри кластерів відповідно до алгоритму |
| NearestNeighbourChain – клас-нащадок класу ClusteringAlgorithm. Клас реалізує логіку алгоритму кластеризації nearest-neighbor chain | private void GenerateClusters() | Генерація початкового набору кластерів |
| public override string Info() | Функція повертає рядок з необхідною інформацією щодо поточного стану алгоритму |
| protected override void InnerIteration() | Ітерація алгоритму |
| private int NearestCluster (int cluster) | Отримати номер найближчого кластера |
| Cluster – клас кластера. Зберігає інформацію, що стосується кластера | public void AddNode (Node node) | Додати вузол до кластера |
| public void AddNodes (List<Node> nodes) | Додати колекцію вузлів до кластера |
| public List<Node> GetDrawingNodes (Color connectionsColor) | Отримати вузли кластера, з’єднані відповідним кольором |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | public double GetPrice() | Обчислення суми квадратичних відстаней вузлів кластера від його центру |
| public void Merge (Cluster cluster) | Об’єднання кластера із поточним |
| public void RemoveNode (Node node) | Видалити вузол із кластера |
| public void RemoveNodes() | Видалити усі вузли з кластера |
| TaskController – статичний клас, функціями якого є зв’язок користувацького інтерфейсу з класами алгоритмів, графічне відображення вузлів, збереження усіх вузлів, робота з файлами | public static void AddNodeAtScreen (Node.NodeType nodeType, int posX, int posY, int volume = 1, int fine = 0) | Додавання нового вузла і відображення його на екрані |
| private static bool CheckVolume() | Перевірка розміру вузла |
| public static void CreateNewModel() | Створити нову модель (видалити усі старі вузли) |
| private static void DrawNode (Graphics g, Node node) | Графічне відображення вузла на відповідний графічний об’єкт |
| private static void DrawNodes (Graphics g, List<Node> nodes) | Графічне відображення вузлів на певний графічний об’єкт |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | public static void DrawNodes (int index) | Графічно відобразити збережену множину вузлів |
| public static void DrawNodes (List<Node> nodes = null) | Графічне відображення множини вузлів |
| public static Color GetDrawingColor (int i) | Функція повертає колір в залежності від індексу |
| public static bool LoadModel(string fileName) | Завантаження множини вузлів з файлу |
| public static bool SaveModel (string fileName = "") | Збереження множини вузлів до файлу |
| public static void SaveDrawnNodes() | Зберегти множину поточну вузлів для подальшого графічного відображення |
| private static void SetAlgorithm (Algorithm algorithm) | Встановлення поточного алгоритму |
| public static void StartBeesAlgorithm (BeesColony.ProblemType problemType, int clustersCount, int scoutsCount,  int goodSitesCount, int bestSitesCount, int neighboursForGoodSites, int neighboursForBestSites, int clusterCapacityLimit = -1, double kilometerCost = 0) | Запустити алгоритм бджолиної колонії з відповідними коефіцієнтами |

Продовження табл. 3.1.

| **Клас** | **Функція** | **Опис функції** |
| --- | --- | --- |
|  | public static void StartClusteringToTsp() | Запустити вирішення задачі комівояжера для кожного кластеру |
| public static void StartKMeansAlgorithm (int clustersCount) | Запустити алгоритм k‑means |
| public static void StartNearestNeighbourChainAlgorithm (int capacityLimit) | Запустити алгоритм nearest-neighbor chain |
| public static void StopAlgorithm() | Зупинити алгоритм |
| ClusteringToTsp – клас-нащадок класу Algorithm. Виконує вирішення задачі комівояжера для кожного заданого кластеру | public void Calculate (int scoutsCount, int goodSitesCount, int bestSitesCount, int neighboursForGoodSites, int neighboursForBestSites) | Виконати обчислення |
| public override void DrawNodes() | Відобразити графічно вузли задачі |
| SerializationData – клас, що тимчасово зберігає інформацію про вузли, а також записує і зчитує її з файлу | public static bool Serialize(string fileName, SerializationData data) | Серіалізувати інформацію в файл |
| public static bool Deserialize(string fileName, out SerializationData data) | Десеріалізувати інформацію з файлу |

## Висновок до розділу

У розділі наведено детальний опис функцій розроблених класів. Саме ці функції забезпечують розв’язання комплексу задач складання плану перевезень продукції.

Детальний опис архітектури розробленого програмного комплексу, а також вимоги до програмного та технічного забезпечення користувача даного комплексу наведені у загальній частині дипломного проекту.

# РОЗДІЛ З ОХОРОНИ ПРАЦІ

У даному розділі розглядається приміщення, у якому відбувалось дипломне проектування, та дотримання правил охорони під час експлуатації електронно-обчислювальних машин, які викладені у НПАОП 0.00-1.28-10. Ці правила поширюються на всіх суб’єктів господарювання незалежно від форм власності, які у своїй діяльності здійснюють роботу, пов’язану з ЕОМ.

## Загальні вимоги до виробничих приміщень

Згідно НПАОП 0.00-1.28-10 вимоги до приміщень, у яких розміщені робочі місця операторів ЕОМ з ВДТ, викладено у ДСанПіН 3.3.2-007-98.

На рисунку 1.1 наведений план приміщення у якому відбувалось дипломне проектування. Площа приміщення складає 20 м2, висота – 3 м, об’єм приміщення – 60 м3. Відповідно до вимог ДСанПіН 3.3.2-007-98 на одного працюючого має припадати не менше 6 м2 площі та не менше 20 м3 об’єму приміщення. У даному приміщенні ці показники дорівнюють 10 м2 та 30 м3 відповідно. Приміщення розташоване на 5-ому поверсі (розміщення на цокольних поверхах та підвальних приміщеннях заборонено). Світлові прорізи орієнтовані на північ, так що природне світло падає на робоче місце зліва. Віконні прорізи обладнані регульованими жалюзі.

Внутрішнє оздоблення приміщення виконане у світлих тонах. Біла стеля має коефіцієнт відбиття 75%, стіни пофарбовані у світлий блакитний колір (коефіцієнт відбиття – 50%), підлога застелена світлим лінолеумом (коефіцієнт відбиття – 30%). За нормами ДСанПіН 3.3.2-007-98 покриття стелі має мати коефіцієнт відбиття 70-80%, стін – 50-60%, підлоги – 30-50%. Інтер’єр приміщення здійснює неабиякий вплив на психоемоційний стан працюючих та на їх працездатність, тому він розроблявся із урахуванням особливостей роботи операторів ЕОМ. Блакитний колір стін обраний як колір здатний підвищувати працездатність та врівноважувати емоційний стан працюючих.

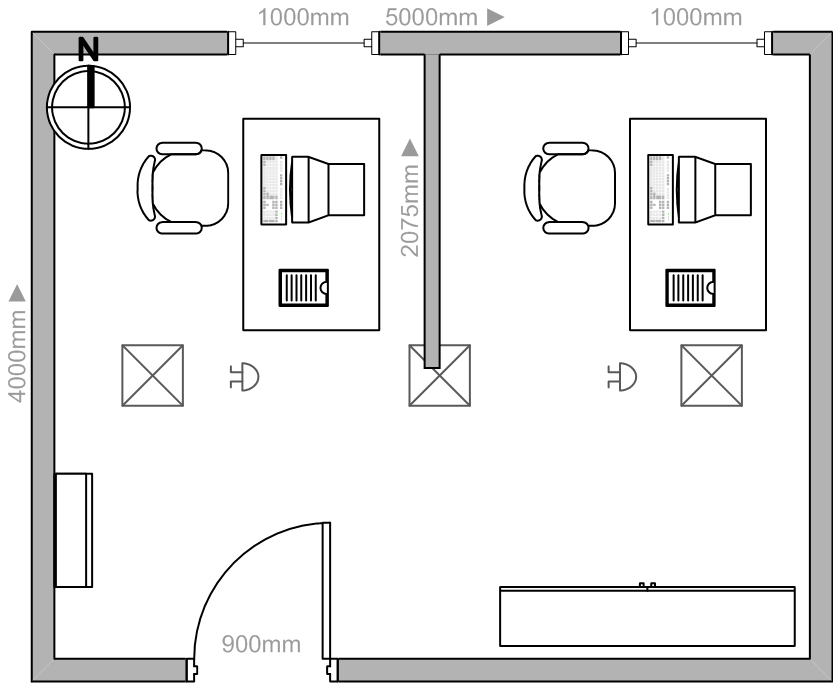


Рисунок 1.1 – План приміщення, у якому відбувалось дипломне проектування

У приміщенні знаходиться 2 робочих місця розділених перегородкою висотою 1,8 м, таким чином для кожного працюючого створений власний простір. У приміщенні також розташована шафа для зберігання документації, професійних книжок та довідників тощо.

Кожного дня у приміщенні проводиться вологе прибирання. Також приміщення оснащене аптечками першої медичної допомоги.

## Характеристика робочого місця

Згідно ДСанПіН 3.3.2-007-98 обладнання і організація робочого місця працюючих з ВДТ ЕОМ і ПЕОМ мають забезпечувати відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності. Робочі місця розташовані так, що природне світло падає зліва.

Поверхня робочого столу має довжину 0,9 м і ширину 1,4 м. Висота столу – 75 см. Габарити поверхні столу забезпечують можливість виконання операцій у зоні досяжності моторного поля. Робочий стіл має простір для ніг завширшки і заввишки 70 см, завглибшки 60 см. Під столом розміщується підставка для ніг.

Конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання і документів. Монітор розташований на відстані 65 см від очей працюючого. Клавіатура та мишка розташовані у висувній панелі столу. Системний блок розміщується у спеціальній ніші під столом. Також стіл має шафки для зберігання документів, канцтоварів тощо.

Робоче місце обладнане підйомно-поворотним стільцем регульованим за висотою, кутом нахилу спинки та відстанню спинки до сидіння. Регулювання за кожним із параметрів здійснюється незалежно і легко. Стілець обладнаний змінними підлокітниками шириною 6 см, завдовжки 30 см. Підлокітники розташовані на висоті 24 см від поверхні сидіння. Висота спинки стільця становить 30 см, його ширина та глибина сидіння – 50 та 40 см відповідно. Поверхня сидіння є плоскою, передній край – заокругленим. Висота поверхні сидіння регулюється у межах 40-50 см. Поверхня сидіння та спинки є напівм’якою із покриттям, що легко чиститься та не електризується.

Усі вищеописані характеристики робочого місця відповідають гігієнічним нормам при роботі з ЕОМ наведеним у ДСанПіН 3.3.2-007-98.

## Мікроклімат

Робота оператора ЕОМ виконується сидячи і не потребує фізичного напруження. Згідно ДСН 3.3.6.042-99 така робота відноситься до категорії «Легка Іа». У ДСанПіН 3.3.2-007-98 встановлено,що у виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату. У таблиці 5.1 наведені оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень, встановлені у ДСН 3.3.6.042-99.

Таблиця 5.1 – Оптимальні величини параметрів мікроклімату в робочій зоні виробничих приміщень для категорії робіт «Легка Іа»

|  |  |  |  |
| --- | --- | --- | --- |
| Період року | Температура повітря, град. С | Відносна вологість, % | Швидкість руху, м/с |
| Холодний | 22-24 | 60-40 | 0,1 |
| Теплий | 23-25 | 60-40 | 0,1 |

У приміщенні усі мікрокліматичні параметри підтримуються у нормі. Регулювання температури повітря здійснюється системою централізованого опалення у холодну пору року, та кондиціонером – у теплу. Вологість повітря також регулюється кондиціонером.

У додатку 3 ДСН 3.3.6.042-99, що є обов’язковим для виконання, встановлені такі оптимальні рівні іонізації повітря на 1 см3 повітря:

* 1500-3000 позитивно заряджених іонів;
* 3000-5000 негативно заряджених.

Кондиціонер обладнаний модулем іонізації повітря, що дозволяє регулювати цей параметр і дотримуватись норм.

Рух повітря у приміщенні здебільшого спричинений роботою кондиціонера, який має функцію регулювання швидкості подачі повітря. Тому цей параметр також легко нормується.

## Характеристика випромінювання

Згідно НПАОП 0.00-1.28-10 вимоги, які необхідно дотримуватись при експлуатації ЕОМ, стосовно електромагнітного, ультрафіолетового та інфрачервоного випромінювання та електростатичного поля викладено у ДСанПіН 3.3.2-007-98.

Основними джерелами інфрачервоного та електромагнітного випромінювання та електростатичного поля є кондиціонер, системні блоки та монітори. На сьогодні усі технічні засоби мають інтенсивність випромінювання і напруженість електростатичного поля в десятки разів менші за нормативно встановлені. Наприклад, інтенсивність інфрачервоного випромінювання сучасних моніторів лежить у межах 10-100 мВт/м2.

Люмінесцентні лампи являються єдиним джерелом ультрафіолетового випромінювання у приміщенні, вони містять незначну УФ-складову із ртутного спектра, що цілком задовольняє вимогам.

## Освітлення

У описаному приміщенні освітлення є суміщеним. Природне освітлення здійснюється через світлові прорізи орієнтовані на північ і забезпечує мінімально необхідний для приміщень, у яких експлуатуються ЕОМ, коефіцієнт природної освітленості рівний 1,5% (за нормами ДСанПіН 3.3.2-007-98).

Джерелами штучного світла є 3 світильники, кожний із яких містить 4 люмінесцентних лампи типу ЛБ 30 (світловий потік однієї лампи – 2020 лм), Світильник обладнані дзеркальними ґратами відповідно до вимог ДСанПіН 3.3.2-007-98. Згідно ДСанПіН 3.3.2-007-98 значення освітлення на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Розроблена система штучного освітлення забезпечує освітленість робочого місця у 380 лк.

## Виробничий шум

Згідно обов’язкового для виконання додатку 1 ДСанПіН 3.3.2-007-98 рівень шуму у приміщенні, у якому знаходиться робоче місце програміста, має бути не вищий за 50 дБА.

Основними джерелами шуму у приміщенні є:

* кондиціонер із рівнем шуму 26-36 дБА
* системні блоки, шумність яких 30‑37 дБА.

Сумарний рівень шуму від кондиціонера та комп’ютерів дорівнює 34‑41 дБА, що цілком задовольняє нормам.

## Електробезпека

Згідно НПАОП 0.00-1.28-10 електробезпека будівель та приміщень, де розміщені робочі місця операторів, повинна відповідати вимогам НПАОП 40.1-1.21-98

У приміщенні, у якому проходило дипломне проектування відсутні умови підвищеної небезпеки ураження людини електричним струмом:

* підвищена вологість;
* наявність струмопровідного пилу;
* струмопровідна підлога;
* висока температура повітря (вище +36 С°).

Тобто приміщення відноситься до категорії приміщень без підвищеної небезпеки ураження струмом.

Усе устаткування у даному приміщенні (комп’ютери, кондиціонер, світильник, монітори тощо) має апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Електромережа для живлення ЕОМ з ВДТ виконана як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладений від стійки групового розподільного щита до розеток електроживлення.

Оскільки у приміщенні експлуатується лише 2 ЕОМ з ВДТ встановлення аварійного резервного вимикача, який може вимкнути повністю вимкнути електричне живлення, окрім освітлення, у приміщенні не є обов’язковим.

Електромережа штепсельних розеток виконана за магістральною схемою по 4 розетки у одному колі і прокладена по підлозі попід стінами у пластикових коробах та рукавах з відводами. Усе обладнання підключається до мережі тільки за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення.

## Пожежна безпека

Згідно НАПБ Б.03.002-2007задане приміщення за вибухопожежною та пожежною небезпекою відноситься до категорії В. Увесь простір у приміщенні відповідно до ДНАОП 0.00-1.32-01 становить собою пожежонебезпечну зону класу П-ІІа – простір у приміщенні, у якому знаходяться горючі речовини та матеріали.

У заданому приміщенні можливе виникнення наступних пожеж:

* пожежі класу А – горіння рідких речовин;
* та класу Е – горіння електрообладнання.

Згідно НАПБ Б.03.001-2004 у приміщеннях категорії В площею не більше 50 м2 за можливості виникнення пожеж класу А та Е мінімальна кількість порошкових вогнегасників (якщо вогнегасна речовина придатна до гасіння пожеж А та Е) із зарядом вогнегасної речовини 5 кг дорівнює 2 шт. У приміщенні знаходиться 3 таких вогнегасники.

Приміщення також оснащене системою автоматичної пожежної сигналізації. У даному приміщенні встановлено 2 димових сповісники ДИ-1. Відстань між сповісниками – 2,5 м, максимальна відстань від сповісника до стіни – 2,3 м, що у повній мірі задовольняє нормам ДБН В.2.5-56-2010. Схема розташування димових сповісників наведена на рисунку 1.1.

Важливим елементом пожежної безпеки є план евакуації. Він знаходиться на виході із приміщення біля дверей.

## Інструкція із техніки безпеки при роботі із ПК

***Перед початком роботи*** необхідно переконатися у справності електропроводки, вимикачів, штепсельних розеток, наявності заземлення комп’ютера та цілісності корпусів обладнання. У випадку виявлення будь-яких несправностей починати роботу заборонено.

***Забороняється:***

* самостійно проводити ремонт або змінювати конструкцію ЕОМ та вимикати захисні пристрої;
* класти сторонні предмети на корпус ПК, монітор та периферійні пристрої;
* часто включати та виключати комп’ютер без необхідності;
* продовжувати роботу на ЕОМ у випадку появи нехарактерних сигналів, нестабільного зображення на моніторі тощо;

***Після завершення роботи*** необхідно знеструмити усі засоби обчислювальної техніки та усі периферійні пристрої.

## Висновок до розділу

У даному розділі висунуті гігієнічні вимоги до приміщень, у яких експлуатуються ЕОМ, встановлені державними нормативними правовими актами про охорону праці. Детально описано приміщення, у якому проходило дипломне проектування, та проведений аналіз щодо його відповідності гігієнічним вимогам. Наведені правила та рекомендації для роботи із персональним комп’ютером.

# ЗАГАЛЬНІ ВИСНОВКИ

При виконанні комплексного дипломного проекту були детально розглянуті питання, які виникають у процесі створення плану перевезення продукції із складу до споживачів, та виділені основні ключові етапи та взаємозв’язки між ними притаманні цьому процесу.

Індивідуальна частина № 2 дипломного проекту присвячена складанню плану перевезень однорідної продукції із урахуванням вантажомісткості транспортних засобів, їх обмеженої кількості та пріоритетів замовлень.

Для розв’язання поставленої задачі був проведений ґрунтовний аналіз предметного середовища, а також враховані результати отримані при розв’язанні задачі планування перевезень малогабаритної продукції (див. загальну частину дипломного проекту), а також результати розв’язання задачі складання плану перевезень однорідної продукції із урахуванням вантажомісткості та необмеженим за кількістю парком транспортних засобів (див. індивідуальну частину № 1 дипломного проекту). На основі цих даних була сформульована математична постановка задачі та розроблений математичний апарат для її розв’язку.

Метою даного дипломного проекту була не просто розробка методу розв’язання даної задачі, що дозволить зменшити витрати на перевезення, а створення якомога більш ефективного методу розв’язання задачі, тобто такого методу який буде зберігати баланс між точністю знайденого розв’язку та часом, затраченим на знаходження цього розв’язку. Тому для розв’язання задачі було застосовано декілька підходів та проведений глибокий порівняльний аналіз на основі отриманих експериментальних даних.

Також у даній роботі наведена детальна специфікація функцій класів усього програмного комплексу, який включає у себе складання плану перевезень продукції за різних умов (перевезення малогабаритної продукції, врахування вантажомісткості, пріоритетів замовлень).

Окремим пунктом розглянуто питання охорони праці, а саме сформульовані гігієнічні вимоги до приміщень, у яких експлуатуються ЕОМ, відповідно до державних нормативних правових актів про охорону праці, та наведені правила, яких необхідно дотримуватись при роботі із персональним комп’ютером.

# ПЕРЕЛІК ПОСИЛАНЬ

# ДОДАТОК А Графічний матеріал