

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр

за освітньо-професійною програмою

«Комп'ютерні науки»

зі спеціальності

122 – Комп'ютерні науки

тема роботи:

***«Удосконалення логістичних процесів за допомогою
генетичного алгоритму»***

Виконав студент гр. КН-21 _____ Мотрій А. М.

Керівник _____ Жосан А. А.

Нормоконтроль _____ Маринич І. А.

Завідувач кафедри _____ Рубан С. А.

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Зав. кафедрою: к.т.н. Рубан С.А.

« 29 » січня 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студентові групи КН-21 Мотрій Андрію Миколайовичу

1. Тема кваліфікаційної роботи: «Удосконалення логістичних процесів за допомогою генетичного алгоритму»

затверджено наказом по університету № 254с від 13.05.2025 р.

2. Термін здачі кваліфікаційної роботи: 06.06.2025 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 72с., додатки, презентація у Microsoft PowerPoint (13 слайдів) в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-2 доц. Жосан А. А.

Нормоконтроль доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>01.04.25</i>
2	<i>Розділ 1</i>	<i>05.04.25</i>
3	<i>Розділ 2</i>	<i>01.05.25</i>
4	<i>Висновки</i>	<i>25.05.25</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>28.05.25</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>20.05.25</i>
7	<i>Підготовка доповіді до захисту</i>	<i>05.06.25</i>

6. Дата видачі завдання: 28.01.2025р.

Керівник _____ / Жосан А. А./

7. **Запевнення:** Я, Мотрій Андрій Миколайович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент _____ /Мотрій А. М./

АНОТАЦІЯ

Мотрій А.М. Удосконалення логістичних процесів за допомогою генетичного алгоритму : кваліфікаційна робота бакалавра : 122 – Комп'ютерні науки. Кривий Ріг. Криворізький національний університет, 2025. 72 с.

Актуальність теми зумовлена постійним зростанням обсягів вантажоперевезень у глобальному світі та необхідністю оптимізації логістичних процесів для зменшення витрат та підвищення ефективності транспортування. Використання інтелектуальних технологій, зокрема генетичних алгоритмів, стає необхідним для вирішення складних логістичних задач, таких як планування маршрутів, оптимізація графіків і управління ресурсами.

Метою роботи є дослідження та розробка методів оптимізації плану перевезень за допомогою інтелектуальних технологій, зокрема генетичних алгоритмів.

У першому розділі було здійснено огляд існуючих програмних засобів для оптимізації логістичних задач. Також було розглянуто особливості оптимізації в галузі логістики та сам процес оптимізації.

У другому розділі проведено порівняння ефективності різних підходів до оптимізації логістичних процесів, таких як використання Microsoft Excel, Matlab і програмної реалізації генетичних алгоритмів. В результаті проведеного аналізу було обрано саме генетичний алгоритм як найбільш підходящий для вирішення логістичних задач, оскільки він дозволяє ефективно працювати з великими і складними просторами пошуку, де інші методи можуть бути малоефективними.

Ключові слова:

ВАНТАЖОПЕРЕВЗЕННЯ, ГЕНЕТИЧНИЙ АЛГОРИТМ, ЕВРИСТИЧНІ МЕТОДИ, ІНТЕЛЕКТУАЛЬНІ ТЕХНОЛОГІЇ, ЛОГІСТИКА, ОПТИМІЗАЦІЯ, ПЛАН ПЕРЕВЕЗЕНЬ

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОПИС ОСНОВНИХ ПОНЯТЬ.....	8
1.1 Огляд існуючих програмних засобів для оптимізації логістичних задач.....	8
1.1.1 Інструмент <i>Lp_solve</i>	9
1.1.2 Інструмент <i>IBM ILOG CPLEX Optimization Studio</i>	10
1.1.3 Інструмент <i>Gurobi Optimizer</i>	11
1.1.4 Інструмент <i>Xpress-MP</i>	12
1.1.5 Інструмент <i>FICO Xpress</i>	13
1.1.6 Інші рішення.....	16
1.2 Особливості оптимізації у логістиці.....	20
1.3 Процес оптимізації.....	21
1.4 Аналіз алгоритмів оптимізації.....	23
1.4.1 Мурашиний алгоритм.....	25
1.4.2 Генетичний алгоритм.....	25
1.4.3 Алгоритм імітації відпалу.....	26
Висновки до розділу.....	26
РОЗДІЛ 2. АНАЛІЗ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ ТА РЕАЛІЗАЦІЯ ОПТИМІЗАЦІЙНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ....	28
2.1 Застосування інтелектуальних технологій для логістичних завдань....	28
2.1.1 Технології електронного обміну даними.....	30
2.1.2 Маркування в логістиці.....	31
2.1.3 Методи безконтактної ідентифікації.....	31
2.1.4 Програмний комплекс WMS.....	32
2.2 Застосування генетичного алгоритму для оптимізації перевезень.....	32
2.3 Застосування Microsoft Excel для оптимізації плану перевезень.....	36

2.5 Застосування програмної реалізації ГА для оптимізації плану перевезень.....	47
2.6 Підсумковий аналіз отриманих результатів оптимізації.....	52
Висновки до розділу.....	54
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	57
ДОДАТОК А. Розрахункові дані	60
ДОДАТОК Б. Лістинг програмної реалізації генетичного алгоритму	66
ДОДАТОК В. Отримані результати оптимізації логістичних перевезень	70

ВСТУП

Логістика є однією з найважливіших сфер економіки в сучасному світі. Вона відповідає за організацію та управління потоками матеріальних, інформаційних і фінансових ресурсів між постачальниками та споживачами. Логістика є ключовим чинником для забезпечення ефективності та конкурентоспроможності компаній, а також для розвитку глобальної економіки.

Ось кілька основних аспектів важливості логістики в сучасному світі:

- Логістика забезпечує ефективне переміщення товарів і послуг, що дозволяє компаніям знижувати витрати на транспортування, зберігання та обробку товарів, одночасно підвищуючи рівень обслуговування клієнтів.
- Логістика сприяє розвитку міжнародної торгівлі, забезпечуючи ефективну співпрацю між підприємствами з різних країн для обміну товарами та послугами.
- Логістика підтримує принципи сталого розвитку, допомагаючи підприємствам знижувати свій екологічний слід, наприклад, через оптимізацію транспортних маршрутів і зменшення кількості відходів.

У сучасному світі, де глобалізація та цифровізація набувають все більшої ваги, логістика стає складнішою та різноманітнішою. Вона вимагає від спеціалістів високого рівня кваліфікації та глибоких знань у галузі новітніх технологій. Логістика є однією з ключових сфер, що сприяє успіху підприємств та розвитку світової економіки.

Штучний інтелект має великий потенціал для оптимізації процесів у логістиці. Ось кілька найбільш популярних методів його застосування:

- *Управління запасами:* ШІ може бути використаний для аналізу даних про продажі, прогнозування попиту та оптимізації рівня запасів, що допомагає знижувати витрати на склади та уникати дефіциту.
- *Оптимізація маршрутів:* Використання ШІ для аналізу даних про дорожній рух, час доставки та інших чинників дозволяє створювати найбільш

ефективні маршрути для транспортних засобів і кур'єрів, що знижує час доставки та витрати на транспортування.

- *Відстеження доставки*: ШІ дозволяє відслідковувати вантажі в режимі реального часу, що дає змогу компаніям своєчасно здійснювати доставку та надавати клієнтам актуальну інформацію про статус їхніх замовлень.

- *Прогнозування збоїв*: За допомогою аналізу історичних даних про збої, ШІ може передбачати потенційні проблеми в майбутньому, допомагаючи компаніям запобігти чи мінімізувати наслідки збоїв.

Штучний інтелект також може бути застосований для автоматизації багатьох логістичних процесів, таких як обробка замовлень, управління складами та транспортування. Це дозволяє компаніям підвищити продуктивність і знизити витрати на людські ресурси. Один з його основних напрямків - генетичні алгоритми - мають великий потенціал для значного поліпшення ефективності та результативності логістичних операцій. Ось кілька конкретних прикладів використання ШІ в логістиці:

- *UPS* застосовує ШІ для прогнозування попиту на доставку. Вони аналізують дані про продажі, погодні умови та інші фактори, щоб визначити кількість вантажівок, які будуть потрібні на кожен конкретний день.

- *Amazon* використовує ШІ для оптимізації роботи своїх розподільчих центрів. Компанія застосовує датчики та алгоритми ШІ для відстеження руху товарів по складу та оптимізації маршрутів працівників.

- *FedEx* використовує ШІ для вдосконалення своїх маршрутних мереж. Алгоритми ШІ аналізують інформацію про дорожній рух, погодні умови та інші чинники для створення найефективніших маршрутів для кур'єрів.

Для досягнення поставленої мети необхідно вирішити наступні задачі::

- Проаналізувати поточний стан на ринку програмного забезпечення для оптимізації логістичних задач.

- Дослідити визначення логістики як процесу.

- Провести аналіз найбільш поширених алгоритмів оптимізації.

- Оцінити застосування інтелектуальних інформаційних технологій у сфері логістики.
- Реалізувати практичну оптимізацію плану перевезень із використанням трьох різних методів оптимізації.
- Провести порівняльний аналіз результатів оптимізації, на основі яких зробити висновок про доцільність використання генетичного алгоритму для вирішення задачі оптимізації логістичних процесів.

Очікується, що застосування генетичних алгоритмів у логістиці буде зростати у міру розвитку технологій і їх доступності. Штучний інтелект має потенціал суттєво змінити логістичну галузь, роблячи її більш ефективною, гнучкою та стійкою.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОПИС ОСНОВНИХ ПОНЯТЬ

1.1 Огляд існуючих програмних засобів для оптимізації логістичних задач

Існує безліч програмних рішень для оптимізації логістичних задач, які можна класифікувати за різними параметрами, такими як:

– *Тип логістичної задачі* – програмне забезпечення може бути розроблене для вирішення різних логістичних проблем, наприклад:

- планування маршрутів;
- розробка графіків;
- оптимізація складування;
- вдосконалення транспортування;
- управління запасами.

– *Методологія оптимізації* – програмні рішення можуть застосовувати різні методи оптимізації, зокрема:

- симплекс метод;
- динамічне програмування;
- метод розподілу;
- генетичні алгоритми.

– *Масштаб задачі* – програмне забезпечення може бути орієнтоване на вирішення задач різних розмірів: від невеликих задач, що підлягають ручному вирішенню, до великих, що вимагають високопродуктивних обчислювальних ресурсів.

При виборі програмного забезпечення для оптимізації логістичних задач необхідно враховувати кілька важливих факторів:

– тип задачі: програма повинна бути спеціалізована для конкретного виду логістичної задачі;

- масштаб задачі: програмне забезпечення повинно відповідати вимогам щодо розміру та складності задачі;
- бюджет: вартість програмного забезпечення може бути суттєвою, тому важливо оцінити його економічну доцільність для організації.

Організації можуть знайти відповідне рішення, яке максимально задовольняє їхні вимоги та бюджети.

1.1.1 Інструмент *Lp_solve*

Це безкоштовний інструмент для розв'язання задач лінійного програмування (*LP*) та змішаного цілочислового програмування (*MIP*). Він використовується для моделювання та оптимізації логістичних задач, таких як розподіл ресурсів, планування маршрутів, управління запасами та інші. Програмне забезпечення підтримує два формати файлів: *MPS* та власний формат *LP_solve*, а також має можливість роботи з користувацькими форматами через інтерфейс *XLI*.

Переваги

- *Безкоштовність і відкритий код*: Ліцензія *LGPLv2.1* дозволяє безкоштовно використовувати та модифікувати програму.
- *Кросплатформність*: Підтримка різних операційних систем, що забезпечує гнучкість у використанні.
- *Потужні алгоритми*: Використання симплекс-методу для задач *LP* та методу гілок і меж для задач *MIP* забезпечує ефективне розв'язання широкого спектра задач.
- *Інтерфейси для різних мов програмування*: Наявність інтерфейсів для мов *R*, *Node.js*, *Rust* та ін. дозволяє інтегрувати *Lp_solve* у різні програмні середовища.
- *Інтегроване середовище розробки (IDE)*: Для користувачів *Windows* доступне спеціалізоване середовище для розробки та тестування моделей.

Недоліки

- *Обмежений графічний інтерфейс:* Відсутність розвиненого графічного інтерфейсу може ускладнити використання для користувачів без технічного досвіду.

- *Потребує знань у програмуванні:* Для ефективного використання необхідні базові знання програмування та роботи з командним рядком.

- *Обмежена підтримка складних моделей:* Може бути менш ефективним при розв'язанні дуже великих або складних задач порівняно з комерційними рішеннями.

Lp_solve є потужним безкоштовним інструментом для розв'язання задач лінійного та змішаного цілочислового програмування, що може бути корисним у вирішенні логістичних задач. Однак для складніших або великих моделей можуть бути доцільні комерційні рішення з розвиненішими інтерфейсами та додатковими можливостями.

1.1.2 Інструмент *IBM ILOG CPLEX Optimization Studio*

Це потужне комерційне програмне забезпечення для розв'язання задач лінійного, цілочисельного та квадратичного програмування. Воно широко використовується для моделювання та оптимізації складних логістичних задач, таких як планування маршрутів, управління ланцюгами постачання та розподіл ресурсів.

Переваги

- *Різноманітність методів оптимізації:* Підтримує симплекс-метод, методи внутрішньої точки та гілок і меж, що дозволяє ефективно розв'язувати широкий спектр задач.

- *Моделювання та програмування:* Надає інтегроване середовище для розробки моделей з використанням *Optimization Programming Language (OPL)* та підтримує обмеження програмування

- *Інтерфейси для різних мов програмування:* Підтримує C++, C#, Java та Python, що забезпечує гнучкість інтеграції в різні програмні середовища

- *Висока продуктивність:* Оптимізовано для розв'язання великих та складних задач, забезпечуючи швидке знаходження рішень.
- *Інтеграція з іншими інструментами:* Може працювати з такими системами, як *AIMMS*, *AMPL*, *GAMS*, що розширює можливості моделювання та аналізу.

Недоліки

- *Вартість ліцензії:* Як комерційне програмне забезпечення, *CPLEX* вимагає придбання ліцензії, що може бути дорогим для малих підприємств або окремих користувачів.
- *Складність освоєння:* Через багатофункціональність та широкий спектр можливостей може знадобитися значний час для навчання та освоєння.
- *Обмеження безкоштовної версії:* Існують обмеження щодо розміру задач та функціональності в безкоштовних версіях, що може бути недостатньо для вирішення великих або складних задач.

IBM ILOG CPLEX Optimization Studio є потужним інструментом для вирішення складних задач оптимізації в логістиці, пропонуючи широкий спектр методів та можливостей. Однак його висока вартість та складність освоєння можуть бути перешкодою для деяких користувачів.

1.1.3 Інструмент *Gurobi Optimizer*

Це потужний комерційний оптимізатор, який застосовується для розв'язання задач лінійного програмування (*LP*), змішаного цілочислового програмування (*MIP*), квадратичного програмування (*QP*), а також їх комбінацій (*MIQP*, *MIQCP*). Він широко використовується в логістиці для задач планування маршрутів, управління запасами, розподілу ресурсів та інших складних оптимізаційних задач.

Переваги

- *Висока продуктивність:* *Gurobi* відомий своєю швидкістю та ефективністю в розв'язанні великих та складних задач оптимізації.

- *Широкий спектр підтримуваних задач:* Підтримує різноманітні типи задач, включаючи лінійні, квадратичні та змішані цілочислові задачі.
- *Інтерфейси для різних мов програмування:* Підтримує *Python*, *C++*, *Java*, *.NET*, *MATLAB*, *R*, що забезпечує гнучкість у використанні.
- *Інтеграція з машинним навчанням:* Може бути використаний разом з методами машинного навчання для покращення процесу прийняття рішень.
- *Підтримка академічних ліцензій:* Надає безкоштовні ліцензії для академічних установ, що сприяє розвитку освіти та досліджень.

Недоліки

- *Вартість ліцензії:* Як комерційне програмне забезпечення, *Gurobi* вимагає придбання ліцензії, що може бути дорогим для малих підприємств або індивідуальних користувачів.
- *Складність освоєння:* Для ефективного використання необхідні знання в галузі оптимізації та програмування.

Gurobi Optimizer є потужним та швидким інструментом для розв'язання складних оптимізаційних задач, що робить його ідеальним вибором для великих підприємств та дослідницьких установ. Однак його висока вартість та складність освоєння можуть бути перешкодою для малих підприємств або індивідуальних користувачів.

1.1.4 Інструмент *Xpress-MP*

Це комерційне програмне забезпечення для математичного моделювання та оптимізації, яке дозволяє формувати, вирішувати та аналізувати задачі лінійного, квадратичного та змішаного цілочислового програмування.

Переваги

- *Різнороманітність методів оптимізації:* Підтримує симплекс-метод, методи внутрішньої точки та гілки та межі, що дозволяє ефективно вирішувати широкий спектр задач.

- *Моделювання та програмування:* Надає інтегроване середовище для розробки моделей за допомогою мови моделювання *Mosel* та підтримує обмеження програмування.
- *Інтерфейси для різних мов програмування:* Підтримує C, C++, Java, .NET, Python та MATLAB, що забезпечує гнучкість інтеграції в різні програмні середовища.
- *Паралельна обробка:* Містить можливості для розподіленого обчислення, що дозволяє вирішувати великі задачі за допомогою паралельної обробки.
- *Інтеграція з іншими інструментами:* Може працювати з такими системами, як AIMMS, AMPL та GAMS, що розширює можливості моделювання та аналізу.

Недоліки

- *Вартість ліцензії:* Як комерційне програмне забезпечення, *FICO Xpress-MP* вимагає придбання ліцензії, що може бути дорогим для малих підприємств або окремих користувачів.
- *Складність освоєння:* Через багатофункціональність та широкий спектр можливостей може знадобитися значний час для навчання та освоєння.
- *Ресурсомісткість:* Вирішення великих та складних задач може вимагати значних обчислювальних ресурсів.

Xpress-MP є потужним інструментом для розв'язання складних задач оптимізації в логістиці, пропонуючи широкий спектр методів та можливостей. Однак його висока вартість та складність освоєння можуть бути перешкодою для деяких користувачів.

1.1.5 Інструмент *FICO Xpress*

Це потужне комерційне програмне забезпечення для математичного моделювання та оптимізації, яке дозволяє вирішувати задачі лінійного, змішаного цілочислового та нелінійного програмування. Воно широко

використовується для розв'язання складних логістичних задач, таких як планування маршрутів, управління запасами, розподіл ресурсів та інші.

Переваги

- *Широкий спектр підтримуваних задач:* Підтримує лінійне, змішане цілочислове, квадратичне, конічне та нелінійне програмування.
- *Потужні алгоритми оптимізації:* Включає методи симплекса, внутрішньої точки, гілок і меж, а також глобальні та локальні методи для нелінійних задач.
- *Моделювальна мова Mosel:* Надає зручну мову для формулювання задач та інтеграцію з іншими мовами програмування через інтерфейси.
- *Інтеграція з MATLAB та Python:* Підтримує обмін даними в пам'яті з MATLAB та Python, що спрощує інтеграцію в існуючі робочі процеси.
- *Паралельна обробка та масштабованість:* Підтримує паралельне вирішення задач та ефективно працює з великими обсягами даних.

Недоліки

- *Вартість ліцензії:* Як комерційне програмне забезпечення, *FICO Xpress* вимагає придбання ліцензії, що може бути дорогим для малих підприємств або індивідуальних користувачів.
- *Складність освоєння:* Для ефективного використання необхідні знання в галузі оптимізації та програмування.
- *Ресурсомісткість:* Вирішення великих та складних задач може вимагати значних обчислювальних ресурсів.

FICO Xpress є потужним та гнучким інструментом для вирішення складних задач оптимізації в логістиці, пропонуючи широкий спектр методів та можливостей. Однак його висока вартість та складність освоєння можуть бути перешкодою для деяких користувачів.

Xpress-MP і *FICO Xpress* є двома різними продуктами компанії *FICO*, і хоча вони мають схожу назву, вони орієнтовані на різні цілі.

Xpress-MP є частиною сімейства рішень *FICO Xpress Optimization*. Це основне програмне забезпечення для математичної оптимізації, яке дозволяє

розв'язувати задачі лінійного, змішаного цілочислового, квадратичного програмування та задачі з обмеженнями. *Xpress-MP* орієнтований на розв'язання великих та складних задач, де потрібна висока швидкість і ефективність.

- Підтримує різноманітні методи оптимізації.
- Має інтерфейси для різних мов програмування.
- Підходить для великих та складних задач.
- Висока вартість ліцензії.
- Має складний інтерфейс і вимагає глибоких знань в галузі

математичного програмування.

FICO Xpress — це більш загальна платформа для математичної оптимізації, яка включає в себе інструменти для моделювання, вирішення задач, та аналізу результатів. Вона підтримує більшість типів задач оптимізації, включаючи лінійне програмування, змішане цілочислове програмування, квадратичне програмування та інші. Є платформою, яка поєднує рішення для моделей оптимізації, що застосовуються не тільки в логістиці, але й у багатьох інших галузях, таких як фінанси, енергетика тощо.

- Включає підтримку моделей на основі *Mosel*, яка спрощує розробку та аналіз моделей.
- Підтримує паралельні обчислення, що покращує продуктивність для великих задач.
- Має інтерфейси для Python, MATLAB, C++, Java та інших мов.
- Потрібна комерційна ліцензія, що може бути дорогою.
- Вимагає значного часу для навчання та освоєння.

Xpress-MP — це спеціалізоване програмне забезпечення для вирішення задач оптимізації, яке більше зосереджене на математичному моделюванні та розв'язуванні задач.

FICO Xpress є більш універсальним інструментом для моделювання та оптимізації, що надає додаткові можливості для інтеграції з іншими інструментами та мовами програмування.

1.1.6 Інші рішення

Окрім наведених вище рішень, також варто звернути увагу на такі інструменти, як *AMPL*, *MOSEK*, *SCIP*, *COIN-OR* та *OptaPlanner*.

Вибір залежить від специфіки задачі, розміру проекту та бюджету.

AMPL (A Mathematical Programming Language) — це потужна мова моделювання для задач оптимізації. Вона підтримує багато різних типів задач, включаючи лінійне програмування (LP), цілочислове програмування (MIP), квадратичне програмування (QP) та інші.

- Потужний інтерфейс для створення складних моделей.
- Підтримка інтеграції з різними солверами, включаючи CPLEX, Gurobi, Xpress, та іншими.
- Легко використовуваний синтаксис для математичних моделей.
- Потрібна ліцензія на використання.
- Потрібно певне навчання для ефективного використання.

MOSEK — це ще один потужний комерційний оптимізатор, який спеціалізується на лінійному, змішаному цілочисловому та квадратичному програмуванні. Він також підтримує конічні задачі та нелінійні задачі.

- Швидкість обчислень, оптимізовано для великих задач.
- Підтримка розв'язання великих задач паралельно.
- Має дуже сильну підтримку квадратичного та конічного програмування.

- Комерційна ліцензія може бути дорогою.
- Для нових користувачів може бути складним для освоєння.

SCIP (Solving Constraint Integer Programs) — це безкоштовний оптимізатор для змішаного цілочислового програмування. Він є одним з найкращих безкоштовних інструментів для розв'язання таких задач, хоча він також підтримує лінійні та нелінійні задачі.

- Безкоштовний (відкритий код).
- Може розв'язувати широке коло задач, включаючи MIP та нелінійне програмування.

- Висока ефективність при розв'язанні складних задач.
- Вимагає більш глибоких знань математичного програмування.
- Обмежена документація порівняно з комерційними рішеннями.

COIN-OR (Computational Infrastructure for Operations Research) — це набір інструментів для операційного дослідження, який включає багато бібліотек для розв'язання різних задач оптимізації, таких як лінійне, цілочислове та змішане програмування.

- Відкритий код і безкоштовне використання.
- Підтримує різні алгоритми для вирішення задач.
- Має велику спільноту користувачів і хорошу документацію.
- Не завжди має таку ж продуктивність, як комерційні рішення.
- Вимагає досвіду в програмуванні для використання всіх можливостей.

OptaPlanner — це бібліотека для Java, яка орієнтована на розв'язання задач оптимізації, включаючи планування, маршрутизацію та інші логістичні задачі. Вона використовує алгоритми пошуку в просторі рішень, такі як алгоритми відновлення та метаевристичні методи (наприклад, алгоритм імітації відпалу).

- Безкоштовний та з відкритим кодом.
- Спеціалізований на задачах планування та маршрутизації.
- Легко інтегрується з Java-додатками.
- Підходить лише для задач, пов'язаних з плануванням та маршрутизацією.
- Може бути менш ефективним для великих задач порівняно з комерційними рішеннями.

Підведемо підсумок аналізу

З наведеного аналізу існуючих рішень можна зробити наступні висновки.

Lp_solve — це найпростіший і найдоступніший інструмент для розв'язання задач лінійного програмування та змішаного цілочислового

програмування. Він ідеально підходить для невеликих проектів, де не потрібні складні функціональні можливості та висока продуктивність.

CPLEX і *Gurobi* — потужні комерційні рішення для вирішення складних задач оптимізації, які використовуються у великих організаціях та дослідженнях. Вони обидва мають високу продуктивність і підтримку різноманітних алгоритмів. Однак ці рішення можуть бути дорогими і складними для освоєння, що робить їх менш підходящими для малих підприємств.

FICO Xpress та *Xpress-MP* — також є комерційними рішеннями, які підтримують широкий спектр задач і методів оптимізації. Вони дозволяють створювати складні моделі та включають можливості для паралельної обробки даних. *FICO Xpress* надає додаткові інструменти для моделювання та інтеграції з іншими мовами програмування, такими як Python та MATLAB, тоді як *Xpress-MP* є більш вузькоспеціалізованим інструментом для розв'язання оптимізаційних задач.

Рекомендації:

- Якщо вам потрібно безкоштовне і просте рішення для базових задач, *Lp_solve* може бути найкращим вибором.
- Для великих і складних проектів, де потрібно вирішувати численні задачі оптимізації, варто звернути увагу на *CPLEX*, *Gurobi* або *MOSEK*, враховуючи їх високу продуктивність і потужні можливості.
- Якщо ваша задача вимагає комплексного підходу та підтримки широкого спектра задач, тоді *FICO Xpress* чи *Xpress-MP* можуть бути відмінним варіантом, при цьому зверніть увагу на вартість та складність освоєння.
- Для малих та середніх підприємств, які працюють з простішими задачами, може бути достатньо рішень з відкритим кодом, таких як *SCIP* або *COIN-OR*.

Таблиця 1.1 - Порівняння програмних рішень

ПЗ	Тип задач	Алгоритми	Інтерфейс	Ліцензія	Платформи	Переваги	Недоліки
Lp_solve	LP, MIP	Симплекс, гілки та межі	Командний рядок	LGPLv2.1	Кросплатформне	Безкоштовність, простота у використанні, відкритий код	Обмежена підтримка складних задач, потребує знань програмування
CPLEX	LP, MIP, QP	Симплекс, внутрішньої точки, гілки та межі	Графічний та текстовий	Комерційна	Windows, Linux, AIX	Висока продуктивність, різноманітні методи, інтеграція з іншими інструментами	Вартість, складність освоєння
Gurobi	LP, MIP, QP, MIQP, MIQCP	Симплекс, внутрішньої точки, гілки та межі	Графічний та текстовий	Комерційна	Windows, Linux, macOS	Швидкість, високі можливості для складних задач, підтримка машинного навчання	Вартість, складність освоєння
FICO Xpress	LP, MIP, QP, QCQP, SOCP, NLP	Симплекс, внутрішньої точки, гілки та межі, глобальні та локальні методи	Графічний та текстовий	Комерційна	Windows, Linux, macOS	Потужні алгоритми, інтеграція з Python та MATLAB, паралельне вирішення задач	Вартість, складність освоєння
Xpress-MP	LP, MIP, QP, QCQP, NLP	Симплекс, внутрішньої точки, гілки та межі	Графічний та текстовий	Комерційна	Windows, Linux, macOS	Широкий спектр підтримуваних задач, потужні алгоритми, зручна мова моделювання	Вартість, складність освоєння

1.2 Особливості оптимізації у логістиці

Оптимізація логістичних процесів - це процес знаходження найбільш ефективного способу вирішення логістичних задач, орієнтуючись на визначені показники результативності. До таких показників можуть належати скорочення витрат, підвищення прибутковості, покращення сервісу для клієнтів або зменшення негативного впливу на довкілля.

Раціоналізація логістики здатна забезпечити бізнесу суттєві вигоди. Серед основних переваг можна виділити:

- скорочення витрат, пов'язаних із транспортуванням, зберіганням та іншими логістичними операціями;
- покращення якості обслуговування клієнтів завдяки своєчасному наданню товарів та послуг;
- зменшення екологічного сліду шляхом оптимізації перевезень і зниження використання ресурсів.

Процес оптимізації може реалізовуватись через різноманітні підходи, зокрема:

- використання математичних моделей, що дозволяють точно змодельовати логістичні процеси і знайти найкращі рішення;
- аналіз логістичних даних, який дає змогу виявити важливі закономірності та тренди, що допомагають покращити ефективність;
- технології машинного навчання, які автоматизують процес пошуку оптимальних рішень, ґрунтуючись на великому обсязі даних.

Ось кілька прикладів застосування оптимізації в логістичних процесах:

- Оптимізація маршрутів транспортування — це процес пошуку найбільш ефективних шляхів доставки товарів або вантажів. Це дозволяє знизити витрати на транспортування та гарантувати своєчасну доставку.
- Оптимізація управління запасами — полягає у визначенні оптимального рівня товарних запасів на складах. Цей підхід допомагає

мінімізувати витрати на зберігання та забезпечити наявність товарів у потрібний момент.

- Оптимізація розташування складів — включає визначення найбільш вигідних місць для розміщення складів, що дозволяє зменшити транспортні витрати та покращити рівень обслуговування клієнтів.

Застосування таких методів оптимізації є ключовим для підвищення ефективності національної логістичної системи. Вони допомагають компаніям не тільки заощаджувати фінанси, але й покращувати сервіс для клієнтів, а також знижувати негативний вплив на навколишнє середовище.

1.3 Процес оптимізації

Оптимізація як процес є серією кроків, спрямованих на досягнення найкращого можливого результату в конкретних умовах. Для цього використовуються різні методи, такі як математичне моделювання, аналіз даних та машинне навчання.

Процес оптимізації можна поділити на такі етапи:

- Ідентифікація проблеми. На цьому етапі важливо чітко визначити, з якою проблемою потрібно впоратися. Це може бути зменшення витрат, збільшення прибутку, поліпшення рівня обслуговування клієнтів або інші завдання.

- Формулювання математичної моделі. Після цього розробляється модель, що адекватно описує ситуацію. Вона повинна враховувати всі ключові змінні та обмеження, що мають значення для вирішення проблеми.

- Вибір оптимізаційного методу. На наступному етапі потрібно вибрати конкретну методологію для пошуку рішення. Це може бути один з численних підходів, які застосовуються для різних видів задач.

- Розв'язання математичної моделі. Після формулювання моделі застосовується обраний метод для пошуку рішення. Це може включати такі

техніки, як симплекс-метод, динамічне програмування, метод найменших квадратів чи метод Монте-Карло.

– Аналіз отриманого результату. На заключному етапі оцінюється знайдене рішення, щоб переконатися, що воно справді оптимальне і відповідає поставленим вимогам.

Приклади застосування оптимізації в різних галузях:

- Виробництво: оптимізація виробничих процесів, ефективне планування ресурсів та раціональне розміщення обладнання.
- Логістика: оптимізація транспортних маршрутів, управління запасами та ефективне розташування складів.
- Фінанси: оптимізація структури інвестиційних портфелів, стратегічне управління ризиками та оптимізація готівкових потоків.

Оптимізацію як процес можна умовно уявити як низку етапів, що ілюструється на рисунку 1.1.



Рисунок 1.1 – Схема оптимізаційного процесу

Оптимізація здійснюється на основі алгоритмів математичного програмування і поділяється на три основні типи: структурну, параметричну та структурно-параметричну. Під час структурної оптимізації здійснюється змінення самої структури об'єкта, в той час як при параметричній оптимізації коригуються параметри (номінали) елементів, що складають цю структуру. Завдання таких типів вирішуються за допомогою алгоритмів дискретного, безперервного та дискретно-безперервного математичного програмування, відповідно.

В залежності від кількості критеріїв, що лежать в основі оптимізації, останню поділяють на однокритеріальну й багатокритеріальну. Варто зазначити, що зазвичай неможливо одночасно досягти максимальних значень за кількома критеріями.

Оптимізація поділяється на два основні типи в залежності від наявності обмежень: оптимізація без обмежень та оптимізація з обмеженнями.

Як для структурного, так і для параметричного синтезу об'єктів використовуються різні оптимізаційні алгоритми:

- Структурний синтез здійснюється за допомогою методів дискретного математичного програмування;
- Параметричний синтез застосовує методи безперервного математичного програмування;
- Структурно-параметричний синтез вимагає використання дискретно-безперервного математичного програмування.

У випадку параметричного синтезу, коли структура об'єкта вже відома (задається), підбираються параметри (номінали) елементів так, щоб мінімізувати або максимізувати цільову функцію. Важливо зазначити, що існуючі оптимізаційні алгоритми не завжди гарантують досягнення глобального оптимуму, однак це не є суттєвим недоліком.

1.4 Аналіз алгоритмів оптимізації

Алгоритми оптимізації – це методи, що застосовуються для знаходження оптимальних рішень для математичних задач. Такі рішення часто визначаються на основі певних обмежень і цільової функції.

Існує безліч видів алгоритмів оптимізації, які можна класифікувати за різними ознаками. Одним із найбільш розповсюджених критеріїв є спосіб, яким ці алгоритми генерують рішення.

Алгоритми нульового порядку стартують з випадкового вибору рішення та застосовують локальні методи для покращення цього вибору. До таких

алгоритмів можна віднести метод проб і помилок, координатний спуск і метод стохастичного градієнта.

Алгоритми першого порядку використовують інформацію про градієнт цільової функції для пошуку кращого результату. Прикладом таких алгоритмів є метод найменших квадратів, метод Ньютона та метод Нелдера-Міда.

Алгоритми другого порядку використовують дані про гессеан цільової функції для знаходження оптимального рішення. Серед них можна виділити метод Гаусса-Ньютона та різні квазі-ньютонові методи.

Алгоритми динамічного програмування використовують рекурентні співвідношення для обчислення оптимальних рішень. До прикладів належать методи Дейкстри та Беллмана.

Алгоритми симплекс-методу застосовують матричну алгебру для розв'язання задач лінійної оптимізації.

Вибір найбільш підходящого алгоритму оптимізації залежить від специфічних характеристик поставленої задачі. До важливих факторів, які потрібно враховувати, належать тип цільової функції, кількість обмежень, складність завдання та доступні ресурси.

Евристичні алгоритми – це методи пошуку, що використовуються в задачах оптимізації, коли точний пошуковий алгоритм відсутній або його обчислення є надто витратним. Часто їх називають наближеними алгоритмами, оскільки вони не гарантують отримання оптимального рішення, але зазвичай дозволяють знайти рішення, близьке до найкращого. Евристичні алгоритми використовують правила або принципи, що, як очікується, приведуть до ефективного рішення. Серед найпопулярніших евристичних методів можна виділити генетичні алгоритми, алгоритми оптимізації на основі мурашиної колонії та методи рою частинок.

1.4.1 Мурашиний алгоритм

Ant Colony Optimization (оптимізація мурашиної колонії) - це потужний поліноміальний метод для наближеного розв'язання задачі комівояжера та подібних завдань пошуку маршрутів на графах. Цей підхід був розроблений бельгійським ученим Марко Доріго. Основна ідея методу полягає в моделюванні поведінки мурашок, які шукають найкоротший шлях від своєї колонії до джерела їжі.

Алгоритм спирається на природне явище - маркування шляхів феромоном, яке мурахи використовують для позначення найкращих маршрутів. Процес починається з розміщення мурашок у вершинах графа, після чого вони починають рухатись по ньому. Напрямок руху кожної мурашки визначається імовірнісним методом, що ґрунтується на формулі, яка враховує як кількість феромону на шляху, так і відстань між точками.

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^N l_k^q \cdot f_k^p} \quad (1.1),$$

де, P_i - імовірність переходу по дорозі i ; f_i - кількість феромонів на i -ому переході; q - величина, що визначає «жадібність» алгоритму; l_i - довжина i -ого переходу; p - величина, що визначає «стадність» алгоритму.

Результат, отриманий за допомогою алгоритму, може бути не зовсім точним і навіть одним з гірших варіантів. Однак завдяки імовірнісному характеру алгоритму, його повторення може призвести до отримання більш точного та оптимального рішення.

1.4.2 Генетичний алгоритм

Genetic algorithm (генетичний алгоритм) - це еволюційний метод пошуку, який застосовується для розв'язання задач оптимізації та моделювання шляхом поетапного підбору, комбінування і варіацій параметрів, що є шуканими, за допомогою механізмів, подібних до біологічної еволюції.

Особливістю генетичного алгоритму є акцент на використання операції «схрещення», яка виконує рекомбінацію можливих рішень-кандидатів. Цей

процес аналогічний біологічному схрещенню і має важливе значення для розвитку нових, більш ефективних рішень.

Генетичні алгоритми є корисними для пошуку оптимальних рішень у великих та складних просторах пошуку, де традиційні методи можуть бути неефективними.[9]

1.4.3 Алгоритм імітації відпалу

Simulated annealing (імітація відпалу) - це загальний метод розв'язання задач глобальної оптимізації, зокрема дискретних та комбінаторних задач, який імітує фізичний процес відпалу металів. Під час цього процесу відбувається поступове охолодження матеріалу, що дозволяє досягти стабільного стану з мінімальною енергією. Аналогічно, алгоритм імітації відпалу поступово "охолоджує" систему рішень, зменшуючи ймовірність прийняття поганих рішень.

Багато задач у сфері проектування, планування та виробництва можна змодельовати за допомогою таких алгоритмів. У першому випадку для пошуку оптимальних розв'язків використовуються алгоритми, які займають значний час і потребують великих обчислювальних ресурсів. У другому випадку застосовуються евристичні методи, що дозволяють знаходити наближені рішення з меншою кількістю обчислень. Алгоритм імітації відпалу є одним з найбільш відомих локальних пошукових методів, ефективно застосовуваних у різних галузях. [10].

Висновки до розділу:

У першому розділі було здійснено огляд існуючих програмних засобів для оптимізації логістичних задач, серед яких розглянуті інструменти, такі як Lp_solve, IBM ILOG CPLEX Optimization Studio, Gurobi Optimizer, Xpress-MP, FICO Xpress та інші рішення. Також було розглянуто особливості оптимізації в галузі логістики та сам процес оптимізації.

Особливу увагу було приділено аналізу трьох основних алгоритмів оптимізації: мурашиному алгоритму, генетичному алгоритму та алгоритму імітації відпалу. Кожен з цих алгоритмів має свої переваги та застосування в залежності від типу задачі.

У результаті проведеного аналізу було обрано саме генетичний алгоритм як найбільш підходящий для вирішення логістичних задач, оскільки він дозволяє ефективно працювати з великими і складними просторами пошуку, де інші методи можуть бути малоефективними. Генетичний алгоритм, завдяки своїм еволюційним механізмам, здатен знаходити оптимальні або наближені до оптимальних рішення в умовах обмежених ресурсів та складності задач, що робить його ідеальним вибором для оптимізації логістичних процесів.

РОЗДІЛ 2

АНАЛІЗ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ ТА РЕАЛІЗАЦІЯ ОПТИМІЗАЦІЙНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ

2.1 Застосування інтелектуальних технологій для логістичних завдань

Інтелектуальні інформаційні технології (ІТ) являють собою набір технологій, які використовують принципи штучного інтелекту для обробки та аналізу даних. Вони активно застосовуються в різних сферах економіки, зокрема в логістиці [14].

ІТ можуть бути ефективно використані в логістиці для вирішення низки завдань, серед яких:

- *Маршрутизація:* ІТ дозволяють планувати маршрути для доставки товарів або поїздок. Наприклад, алгоритми машинного навчання можуть допомогти визначити оптимальні маршрути, враховуючи такі фактори, як дорожні умови, час доставки та вартість.
- *Управління запасами:* ІТ допомагають у визначенні оптимальних рівнів запасів товарів. Так, штучні нейронні мережі можуть бути використані для прогнозування попиту та визначення найкращого рівня запасів для його задоволення.
- *Розподіл ресурсів:* ІТ можуть сприяти розподілу товарів між складами або робіт між працівниками. Наприклад, генетичне програмування дозволяє оптимізувати розподіл товарів, враховуючи витрати на зберігання на різних складах.
- *Управління логістичними центрами:* ІТ можуть використовуватись для моніторингу роботи логістичних центрів. Наприклад, комп'ютерний зір може бути застосований для відслідковування переміщення товарів у центрі.

– Управління ланцюгами поставок: ІТ допомагають управляти ланцюгами поставок. Системи аналізу великих даних можуть виявляти проблеми в ланцюгах поставок та пропонувати заходи для їх усунення.

ІТ мають кілька переваг у порівнянні з традиційними методами логістики. Вони сприяють:

– Покращенню ефективності логістичних процесів: ІТ дають можливість логістичним компаніям приймати більш обґрунтовані рішення, що сприяє зниженню витрат та збільшенню продуктивності.

– Збільшенню прозорості логістичних процесів: ІТ забезпечують доступ до точнішої інформації про стан логістичних операцій, що дає змогу краще контролювати та управляти ними.

– Швидшій адаптації до змін: ІТ дозволяють компаніям оперативно реагувати на зміни зовнішніх умов, таких як коливання попиту на товари або зміни в дорожніх умовах.

ІТ є важливим інструментом для підвищення ефективності логістичних процесів. Вони допомагають компаніям здобувати конкурентні переваги та покращувати якість обслуговування клієнтів.

З розвитком ІТ їх впровадження в логістиці буде лише зростати, що дозволить компаніям ставати ще більш ефективними та конкурентоспроможними.

Тепер розглянемо основні напрямки розвитку інформаційних технологій.

2.1.1 Технології електронного обміну даними

Electronic Data Interchange (EDI, технологія електронного обміну даними) дозволяє автоматизувати процеси створення, відправлення, отримання та обробки електронних документів, а також інтегрувати їх з існуючими бізнес-додатками.

EDI була розроблена для того, щоб усунути недоліки традиційного обміну документами в логістиці, який, на жаль, досі є поширеним у багатьох

секторах економіки України. Серед основних недоліків традиційного обміну можна виокремити: довгий цикл отримання та підтвердження замовлення, недостатньо формалізований процес погодження змін у замовленнях, високі витрати, пов'язані з постійними телефонними переговорами і уточненнями, а також затримки у виставленні рахунків.

2.1.2 Маркування в логістиці

Маркування - це процес нанесення знаків, написів та малюнків на товар або тару з метою їх ідентифікації, а також для вказівки на способи перевезення, обробки та зберігання продукції. Це сприяє підвищенню ефективності товарообігу, забезпеченню безпеки та якості товару, відповідності стандартам і міжнародним нормам, а також дозволяє здійснювати більш ефективний контроль на всіх етапах логістичного ланцюга.

Носії маркування можуть бути різними, включаючи етикетки, ярлики, вкладиші, бирки, контрольні стрічки, клейма, штампи та інші елементи. Метою умовних позначень є ідентифікація товару певної компанії та виділення його серед інших аналогічних товарів.

2.1.3 Методи безконтактної ідентифікації

Технології безконтактної ідентифікації включають технічні засоби, організаційні заходи та послідовність дій, що забезпечують розпізнавання і реєстрацію об'єктів і прав, а також введення цієї інформації в комп'ютер без використання клавіатури, в реальному часі.

Сьогодні існує кілька відомих технологій безконтактної ідентифікації, серед яких: карткові технології, біометричні технології, технології штрихового кодування, а також технології радіочастотної ідентифікації (RFID-технології, Radio Frequency Identification)

2.1.4 Програмний комплекс WMS

Warehouse Management System (WMS) - це апаратно-програмний комплекс, призначений для ефективного управління розміщенням і переміщенням товарів на складі. Його основною відмінністю від традиційних складських облікових систем є те, що, крім суто облікових функцій, він виконує активну управлінську роль. Вона аналізує завдання, що стоять в черзі на виконання, оптимізує маршрути руху навантажувально-транспортного устаткування та надає сигнал складському персоналу щодо пріоритетних завдань.

WMS автоматизує наступні процеси:

- *Прийом товарів:* WMS дозволяє реєструвати прибуття товарів на склад, перевіряти їхній стан і розміщувати їх на відповідних стелажах.
- *Відвантаження товарів:* Система допомагає готувати замовлення до відвантаження, комплектувати їх і здійснювати відправку товарів клієнтам.
- *Управління запасами:* WMS забезпечує контроль рівня запасів товарів, прогнозування попиту та здійснення замовлень у постачальників.
- *Облік складських операцій:* Система веде облік складських операцій, таких як прийом і відвантаження товарів, а також переміщення товарів між стелажми та іншими зонами складу.

2.2 Застосування генетичного алгоритму для оптимізації перевезень

Генетичний алгоритм (genetic algorithm) - це еволюційний алгоритм пошуку, який застосовується для вирішення задач оптимізації та моделювання шляхом поступового підбору, комбінування і варіацій шуканих параметрів. Це відбувається з використанням механізмів, що нагадують процеси біологічної еволюції.

Отримання даних (Data Mining) - це процес виявлення прихованих закономірностей або взаємозв'язків між змінними у великих обсягах необроблених даних. Зазвичай цей процес поділяється на задачі класифікації,

модельовання та прогнозування. Сучасні підприємства, дослідницькі проекти та інтернет генерують величезні обсяги даних. Процес добування даних дозволяє автоматично аналізувати ці масиви даних, застосовуючи методи математичної статистики, штучні нейронні мережі, теорію нечітких множин або генетичні алгоритми. Основною метою такого аналізу є виявлення правил та закономірностей.

Задача в генетичному алгоритмі кодується так, щоб її рішення можна було подати у вигляді масиву, подібного до структури хромосоми. Спочатку генерується випадкова популяція, де кожен індивід (або особина) є початковим елементом масиву. Потім ці особи оцінюються за допомогою функції пристосованості, яка визначає їхню «пристосованість» або здатність до виживання в контексті вирішення задачі. Кожна особина отримує значення пристосованості, що дає змогу визначити її потенціал до подальшого розвитку.

На основі цих оцінок здійснюється відбір осіб, які пройдуть до етапу схрещення (селекція). Після відбору застосовуються генетичні оператори - найчастіше це оператори схрещення (crossover) та мутації (mutation), які генерують нове покоління осіб. Операція схрещення поєднує характеристики двох батьківських особин, а мутація випадковим чином змінює частину генетичної інформації.

Згенеровані особи нового покоління також оцінюються за допомогою функції пристосованості, після чого знову проводиться селекція та мутація. Процес еволюції триває кілька поколінь, поки не буде досягнуто певного критерію зупинки алгоритму, такого як максимальна кількість поколінь або досягнення оптимального рішення.

Критерії зупинки генетичного алгоритму можуть бути різними, зокрема:

- *Знаходження глобального або надоптимального вирішення:* алгоритм припиняється, коли досягнуто найкраще можливе рішення, яке більше не піддається покращенню.

- *Вичерпання числа поколінь:* якщо алгоритм не знайшов оптимальне рішення до визначеного числа поколінь, він припиняється.

– *Вичерпання часу, відпущеного на еволюцію*: якщо досягнуто обмеження за часом, алгоритм завершується, навіть якщо оптимальне рішення не знайдено.

Генетичні алгоритми особливо ефективні для пошуку рішень у дуже великих і складних просторах пошуку, де традиційні методи можуть бути малоефективними або неможливими для застосування. Завдяки здатності до паралельної еволюції, генетичні алгоритми можуть досліджувати широкий спектр можливих варіантів, підвищуючи ймовірність знайти рішення навіть у складних ситуаціях.

Етапи роботи генетичного алгоритму можна описати наступним чином:

– *Створення початкової популяції*: Генерується випадкова популяція осіб (рішень), яка буде використовуватися для подальшого пошуку оптимального рішення.

– *Обчислення функції пристосованості для осіб популяції (оцінювання)*: Для кожної особини обчислюється значення функції пристосованості, яке відображає її здатність вирішувати поставлену задачу. Це значення визначає, наскільки особина «пристосована» до умов середовища.

– *Повторювання до виконання критерію зупинки алгоритму*:

– *Вибір індивідів із поточної популяції (селекція)*: На основі значення пристосованості вибираються особини, які будуть брати участь у наступних етапах, таких як схрещення та мутація.

– *Схрещення або/та мутація*: Застосовуються генетичні оператори - схрещення (поєднання генетичної інформації двох батьків) та мутація (випадкові зміни в генетичній інформації) для створення нових осіб, що можуть мати кращі характеристики.

– *Обчислення функції пристосованості для всіх осіб*: Оновлені особини оцінюються за допомогою функції пристосованості для визначення їх здатності до вирішення задачі.

– *Формування нового покоління*: Після оцінки нові особини формують наступне покоління, яке заміняє попереднє. Це покоління продовжує еволюцію в пошуках оптимального рішення.

Цей процес повторюється до виконання одного з критеріїв зупинки (наприклад, досягнення оптимального рішення або вичерпання часу чи покоління).

Узагальнену схему роботи генетичного алгоритму можна представити у вигляді графічної діаграми (рисунок 2.1), яка демонструє взаємозв'язок і послідовність етапів.

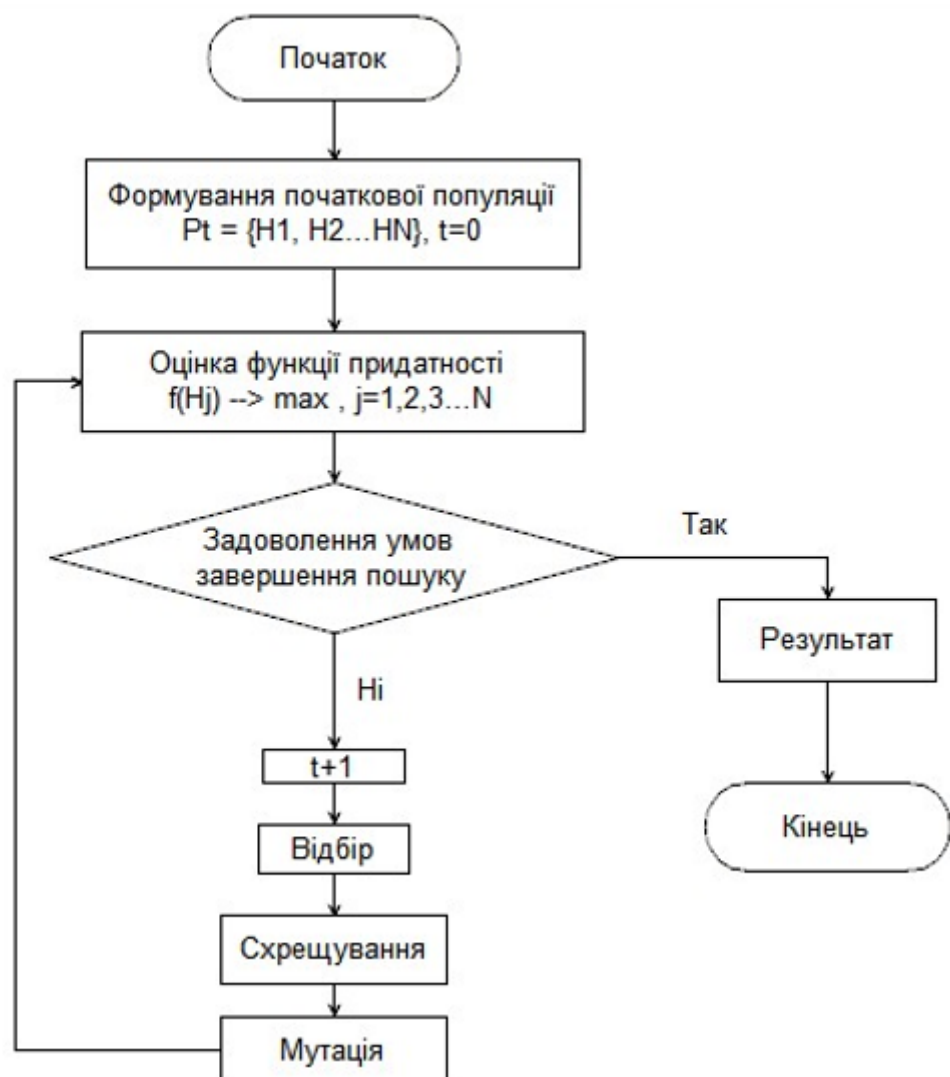


Рисунок 2.1– Блок-схема роботи ГА

2.3 Застосування Microsoft Excel для оптимізації плану перевезень

За допомогою додатку Пошук рішень пакету Microsoft Excel проведемо оптимізацію плану перевезень продукції підприємства, що складається з двох складів та умовних 50 зон (об'єктів виробництва) з сформованим автопарком 100 автомобілів різного типу. Необхідні дані для розрахунку, а саме: характеристики транспортних засобів, таблиця відстаней між зонами, загальні розрахунки витрат на транспортування за маршрутом Зона – Склад 1 наведено у Додатку А.

Розрахунок виконаємо за допомогою матриць розміром 10×10 (10 зон ×10 автомобілів), тому для всіх розрахунків введемо такі обмеження:

1. Кількість машин в 1 зоні (не повинна перевищувати 4 од.):

$$\begin{aligned}
 &X_1+X_2+X_3+X_4+X_5+X_6+X_7+X_8+X_9+X_{10} \leq 4 \\
 &X_{11}+X_{12}+X_{13}+X_{14}+X_{15}+X_{16}+X_{17}+X_{18}+X_{19}+X_{20} \leq 4 \\
 &X_{21}+X_{22}+X_{23}+X_{24}+X_{25}+X_{26}+X_{27}+X_{28}+X_{29}+X_{30} \leq 4 \\
 &X_{31}+X_{32}+X_{33}+X_{34}+X_{35}+X_{36}+X_{37}+X_{38}+X_{39}+X_{40} \leq 4 \\
 &X_{41}+X_{42}+X_{43}+X_{44}+X_{45}+X_{46}+X_{47}+X_{48}+X_{49}+X_{50} \leq 4 \\
 &X_{51}+X_{52}+X_{53}+X_{54}+X_{55}+X_{56}+X_{57}+X_{58}+X_{59}+X_{60} \leq 4 \\
 &X_{61}+X_{62}+X_{63}+X_{64}+X_{65}+X_{66}+X_{67}+X_{68}+X_{69}+X_{70} \leq 4 \\
 &X_{71}+X_{72}+X_{73}+X_{74}+X_{75}+X_{76}+X_{77}+X_{78}+X_{79}+X_{80} \leq 4 \\
 &X_{81}+X_{82}+X_{83}+X_{84}+X_{85}+X_{86}+X_{87}+X_{88}+X_{89}+X_{90} \leq 4 \\
 &X_{91}+X_{92}+X_{93}+X_{94}+X_{95}+X_{96}+X_{97}+X_{98}+X_{99}+X_{100} \leq 4
 \end{aligned}$$

2. Загальна кількість машин в автопарку:

$$X_1+X_{11}+X_{21}+X_{31}+X_{41}+X_{51}+X_{61}+X_{71}+X_{81}+X_{91} \leq 8$$

$$X_2+X_{12}+X_{22}+X_{32}+X_{42}+X_{52}+X_{62}+X_{72}+X_{82}+X_{92} \leq 10$$

$$X_3+X_{13}+X_{23}+X_{33}+X_{43}+X_{53}+X_{63}+X_{73}+X_{83}+X_{93} \leq 8$$

$$X_4+X_{14}+X_{24}+X_{34}+X_{44}+X_{54}+X_{64}+X_{74}+X_{84}+X_{94} \leq 12$$

$$X_5+X_{15}+X_{25}+X_{35}+X_{45}+X_{55}+X_{65}+X_{75}+X_{85}+X_{95} \leq 12$$

$$X_6+X_{16}+X_{26}+X_{36}+X_{46}+X_{56}+X_{66}+X_{76}+X_{86}+X_{96} \leq 10$$

$$X_7+X_{17}+X_{27}+X_{37}+X_{47}+X_{57}+X_{67}+X_{77}+X_{87}+X_{97} \leq 8$$

$$X_8+X_{18}+X_{28}+X_{38}+X_{48}+X_{58}+X_{68}+X_{78}+X_{88}+X_{98} \leq 12$$

$$X_9+X_{19}+X_{29}+X_{39}+X_{49}+X_{59}+X_{69}+X_{79}+X_{89}+X_{99} \leq 12$$

$$X_{10}+X_{20}+X_{30}+X_{40}+X_{50}+X_{60}+X_{70}+X_{80}+X_{90}+X_{100} \leq 8$$

3. Додатні змінні:

$$X_1, \dots, X_{100} \geq 0$$

Оптимізація логістики за маршрутом Зона-Склад 1:

Таблиця 2.1 - Розрахунок для зон 1-10 (Вартість: 12363,4 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зпл 554М	Зпл-130	
1	0	4	0	0	0	0	0	0	0	0	4
2	0	0	0	0	0	0	0	4	0	0	4
3	0	0	0	0	4	0	0	0	0	0	4
4	0	2	0	0	2	0	0	0	0	0	4
5	0	0	0	0	0	4	0	0	0	0	4
6	0	0	0	0	0	0	0	4	0	0	4
7	0	0	0	0	0	4	0	0	0	0	4
8	0	0	0	0	0	0	0	4	0	0	4
9	0	0	0	0	4	0	0	0	0	0	4
10	0	4	0	0	2	2	0	0	0	0	4
	0	10	0	0	12	10	0	8	0	0	

Таблиця 2.2 - Розрахунок для зон 11-20 (Вартість: 9106.9 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130	
11	0	2	0	0	2	0	0	0	0	0	4
12	0	0	0	0	0	0	0	4	0	0	4
13	0	0	0	0	0	4	0	0	0	0	4
14	0	0	0	0	0	0	0	4	0	0	4
15	0	0	0	0	2	2	0	0	0	0	4
16	0	0	0	0	4	0	0	0	0	0	4
17	0	0	0	0	0	4	0	0	0	0	4
18	0	0	0	0	0	0	0	4	0	0	4
19	0	0	0	0	4	0	0	0	0	0	4
20	0	4	0	0	0	0	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.3 - Розрахунок для зон 21-30 (Вартість: 12489.6 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130	
21	0	0	0	0	0	0	0	4	0	0	4
22	0	0	0	0	0	4	0	0	0	0	4
23	0	0	0	0	0	0	0	4	0	0	4
24	0	0	0	0	4	0	0	0	0	0	4
25	0	0	0	0	2	2	0	0	0	0	4
26	0	2	0	0	2	0	0	0	0	0	4
27	0	0	0	0	4	0	0	0	0	0	4
28	0	0	0	0	0	4	0	0	0	0	4
29	0	0	0	0	0	0	0	4	0	0	4
30	0	4	0	0	0	0	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.4 - Розрахунок для зон 31-40 (Вартість: 12041.5 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зпл 554M	Зпл-130	
31	0	0	0	0	4	0	0	0	0	0	4
32	0	0	0	0	0	0	0	4	0	0	4
33	0	0	0	0	0	4	0	0	0	0	4
34	0	2	0	0	0	2	0	0	0	0	4
35	0	0	0	0	0	4	0	0	0	0	4
36	0	0	0	0	0	0	0	4	0	0	4
37	0	4	0	0	0	0	0	0	0	0	4
38	0	0	0	0	0	0	0	4	0	0	4
39	0	0	0	0	4	0	0	0	0	0	4
40	0	0	0	0	4	0	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.5 - Розрахунок для зон 41-50 (Вартість:9354.8 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зпл 554M	Зпл-130	
41	0	0	0	0	0	4	0	0	0	0	4
42	0	0	0	0	0	0	0	4	0	0	4
43	0	0	0	0	0	4	0	0	0	0	4
44	0	0	0	0	0	0	0	4	0	0	4
45	0	0	0	0	4	0	0	0	0	0	4
46	0	4	0	0	0	0	0	0	0	0	4
47	0	0	0	0	4	0	0	0	0	0	4
48	0	0	0	0	4	0	0	0	0	0	4
49	0	2	0	0	0	2	0	0	0	0	4
50	0	0	0	0	0	0	0	4	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Оптимізація логістики за маршрутом Зона-Склад 2:

Таблиця 2.6 - Розрахунок для зон 1-10 (Вартість: 10176.85 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130	
1	0	4	0	0	0	0	0	0	0	0	4
2	0	0	0	0	0	4	0	0	0	0	4
3	0	0	0	0	4	0	0	0	0	0	4
4	0	2	0	0	2	0	0	0	0	0	4
5	0	0	0	0	0	4	0	0	0	0	4
6	0	0	0	0	0	0	0	4	0	0	4
7	0	0	0	0	0	0	0	4	0	0	4
8	0	0	0	0	0	0	0	4	0	0	4
9	0	0	0	0	4	0	0	0	0	0	4
10	0	0	0	0	2	2	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.7 - Розрахунок для зон 11-20 (Вартість: 12835.35 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130	
11	0	0	0	0	4	0	0	0	0	0	4
12	0	0	0	0	0	0	0	4	0	0	4
13	0	0	0	0	0	4	0	0	0	0	4
14	0	0	0	0	4	0	0	0	0	0	4
15	0	0	0	0	0	2	0	2	0	0	4
16	0	2	0	0	2	0	0	0	0	0	4
17	0	0	0	0	0	4	0	0	0	0	4
18	0	0	0	0	2	0	0	2	0	0	4
19	0	0	0	0	0	0	0	4	0	0	4
20	0	4	0	0	0	0	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.8 - Розрахунок для зон 21-30 (Вартість: 13820.8 грн.)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130	
21	0	0	0	0	4	0	0	0	0	0	4
22	0	0	0	0	0	4	0	0	0	0	4
23	0	0	0	0	0	0	0	4	0	0	4
24	0	0	0	0	4	0	0	0	0	0	4
25	0	0	0	0	0	0	0	4	0	0	4
26	0	0	0	0	0	4	0	0	0	0	4
27	0	0	0	0	0	0	0	4	0	0	4
28	0	2	0	0	0	2	0	0	0	0	4
29	0	0	0	0	4	0	0	0	0	0	4
30	0	4	0	0	0	0	0	0	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.9 - Розрахунок для зон 31-40 (Вартість: 10485.8 грн)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил ММЗ 554М	Зил-130	
31	0	0	0	0	2	2	0	0	0	0	4
32	0	2	0	0	2	0	0	0	0	0	4
33	0	0	0	0	4	0	0	0	0	0	4
34	0	4	0	0	0	0	0	0	0	0	4
35	0	0	0	0	0	4	0	0	0	0	4
36	0	0	0	0	0	0	0	4	0	0	4
37	0	0	0	0	4	0	0	0	0	0	4
38	0	0	0	0	0	0	0	4	0	0	4
39	0	0	0	0	0	4	0	0	0	0	4
40	0	0	0	0	0	0	0	4	0	0	4
	0	6	0	0	12	10	0	12	0	0	

Таблиця 2.10 - Розрахунок для зон 41-50 (Вартість: 10485.8 грн)

Зона	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил ММЗ 554М	Зил-130	
41	0	0	0	0	0	4	0	0	0	0	4
42	0	0	0	0	0	0	0	4	0	0	4
43	0	0	0	0	0	4	0	0	0	0	4
44	0	0	0	0	0	0	0	4	0	0	4
45	0	0	0	0	4	0	0	0	0	0	4
46	0	4	0	0	0	0	0	0	0	0	4
47	0	2	0	0	2	0	0	0	0	0	4
48	0	0	0	0	4	0	0	0	0	0	4
49	0	0	0	0	2	2	0	0	0	0	4
50	0	0	0	0	0	0	0	4	0	0	4
	0	6	0	0	12	10	0	12	0	0	

2.4 Застосування пакету Matlab для оптимізації плану перевезень

MatLab (від англ. Matrix Laboratory) - це сучасний програмний продукт для виконання матричних обчислень. Пакет був розроблений з урахуванням потреб різних користувачів і на сьогодні є високоякісним інструментом, який поєднує в собі можливості для обчислень, візуалізації та програмування в зручному форматі, де завдання та їх рішення подаються у вигляді, що максимально наближений до математичного. MatLab є стандартним інструментом для роботи в численних галузях математики та науки.

Основним елементом обробки даних у MatLab є масив, що дозволяє ефективно вирішувати завдання, пов'язані з використанням матриць та векторів. Важливу роль у MatLab відіграють спеціалізовані групи програм, які називаються Toolboxes.

Toolboxes - це набір функцій MatLab (М-файлів), що надають можливість вирішувати специфічні типи задач, застосовуючи спеціалізовані методи. Вони

використовуються для обробки сигналів, систем управління, нейронних мереж, нечіткої логіки, моделювання та інших напрямків.

Система MatLab складається з п'яти основних компонентів:

1. *Мова MatLab* - мова високого рівня для роботи з матрицями, яка включає можливості управління потоками, функціями, структурами даних, введенням-виведенням, а також підтримує об'єктно-орієнтоване програмування. Це дає змогу створювати як прості програми, так і великі та складні додатки.

2. *Середовище MatLab* - набір інструментів і засобів, з якими взаємодіє користувач або програміст під час роботи з MatLab. Воно включає інструменти для керування робочим простором, введенням і виведенням даних, а також для створення, контролю та налагодження М-файлів і додатків.

3. *Керована графіка* - графічна система MatLab, яка пропонує команди для візуалізації дво- та тривимірних даних, обробки зображень, анімації та створення ілюстраційних графіків.

4. *Бібліотека математичних функцій* - велика колекція обчислювальних алгоритмів, що включає як елементарні функції (сума, синус, косинус, комплексна арифметика), так і більш складні (обернена матриця, знаходження власних значень, функції Бесселя, швидке перетворення Фур'є).

5. *Програмний інтерфейс* - бібліотека, що дозволяє писати програми на Сі та Фортрані, які взаємодіють з MatLab. Вона також містить засоби для виклику програм з MatLab через динамічний зв'язок.

Окрім основних компонентів, у MatLab є додаток Simulink, який є інтерактивною системою для моделювання нелінійних динамічних систем. Ця програма пропонує середовище, що керується мишею, дозволяючи моделювати процеси за допомогою перетягування блоків на екрані та їх маніпуляції. Simulink працює з лінійними, нелінійними, безперервними, дискретними та багатовимірними системами.

Blocksets - це додатки до Simulink, які надають бібліотеки блоків для спеціалізованих застосувань, таких як зв'язок, обробка сигналів, енергетичні системи.

Real-Time Workshop - програма, яка дозволяє генерувати код з блокових діаграм і виконувати його на різних системах реального часу, як показано на рисунку 2.2.

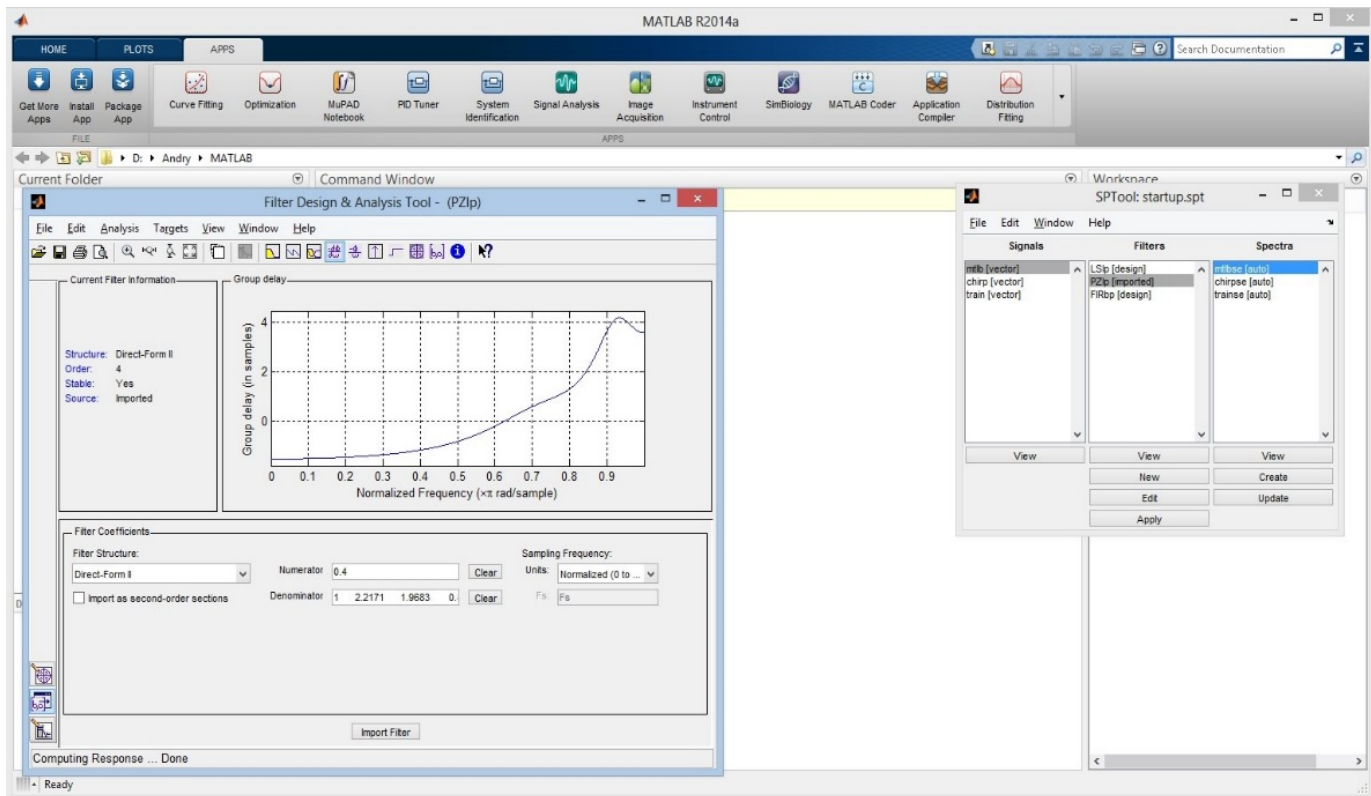


Рисунок 2.2 – Робоче вікно пакету оптимізації

MatLab можна ефективно використовувати для оптимізації плану транспортних перевезень завдяки його потужним математичним і алгоритмічним можливостям. Для цього можна застосовувати різні методи та інструменти, що включають оптимізацію, моделювання і візуалізацію даних. Ось кілька основних підходів, як це можна зробити:

1. Задача оптимізації маршруту (Транспортна задача)

– *Математичні моделі:* Транспортні перевезення часто можна описати математичними моделями, які включають обмеження на вартість, час

і вантажопідйомність. MatLab дозволяє застосовувати методи лінійного програмування для розв'язання таких задач.

- *Функції для лінійного програмування:* В MatLab є функції для розв'язання лінійних задач, зокрема функція `linprog`, яка дозволяє мінімізувати або максимізувати функцію за допомогою лінійних обмежень.

2. Моделювання і оптимізація мережі перевезень

- *Мережеві моделі:* Використання алгоритмів для оптимізації маршрутів у транспортних мережах, таких як алгоритм Дейкстри для пошуку найкоротших шляхів або алгоритм пошуку максимальної потужності для потоків.

- *Simulink:* Використовуючи Simulink, можна створити модель транспортної мережі, де кожен блок буде відповідати певній ділянці мережі або транспортному засобу. Це дозволяє провести моделювання різних сценаріїв і оптимізувати загальні витрати.

3. Моделювання часу та вартості перевезень

- *Моделювання обмежень і вартості:* MatLab дозволяє моделювати не лише фізичні маршрути, а й враховувати обмеження, такі як час, витрати пального, штрафи за запізнення, максимальна вага вантажу тощо.

- *Часові витрати:* Для оптимізації перевезень можна використовувати моделювання і аналіз даних, таких як час у дорозі, з урахуванням різних факторів (трафік, погода, час доби). В MatLab можна використовувати статистичні та машинні навчання для прогнозування і оптимізації.

4. Оптимізація через генетичні алгоритми

- *Генетичні алгоритми:* Якщо задача оптимізації складна і не має прямого розв'язку, генетичні алгоритми можуть бути корисними для пошуку найкращих маршрутів. В MatLab є функція для реалізації генетичних алгоритмів, які допомагають у пошуку оптимальних варіантів для складних задач, таких як оптимізація маршрутів, вантажів та транспортних засобів.

5. Обробка та візуалізація даних

- *Аналіз даних:* MatLab дозволяє аналізувати великі обсяги даних, таких як трафік на маршрутах, швидкість транспорту, погодні умови, попит на перевезення та інші важливі параметри.

- *Візуалізація:* Для кращого розуміння і прийняття рішень можна використовувати графічні можливості MatLab для візуалізації маршруту перевезень, витрат та часових інтервалів на різних етапах логістичної мережі.

6. Моделювання складу та логістики

- *Моделювання складу:* Якщо оптимізація плану перевезень пов'язана з організацією роботи складів, MatLab дозволяє створити моделі складів і інтегрувати їх з моделями транспортних перевезень для пошуку оптимальних рішень.

- *Системи управління запасами:* Використання Toolboxes MatLab для управління запасами і перевезеннями допоможе в оптимізації потоків товарів між різними пунктами.

7. Аналіз та прогнозування попиту

- *Прогнозування попиту:* Використовуючи статистичні та машинні методи, MatLab може передбачити попит на перевезення в різні періоди часу, що допоможе в оптимізації планів перевезень і вантажопотоків.

У нашому випадку потрібно оптимізувати маршрут доставки товарів між кількома складами та пунктами розподілу. Можна використовувати лінійне програмування для мінімізації вартості перевезення, враховуючи відстані між складами, вартості транспорту та обмеження на кількість товарів.

Використовуючи MatLab, можна створити інтерактивне середовище для перевірки різних варіантів плану перевезень, що дає змогу швидко адаптуватися до змін і приймати обґрунтовані рішення для зменшення витрат та підвищення ефективності транспортних операцій.

Для оптимізації в Matlab були використані наступні налаштування:

Розмір популяції: 100

Кількість ітерацій: 500

Тип кросоверу: 2 точки з ймовірністю 80%

– *Мінімум додаткової інформації*: Генетичні алгоритми для вирішення задач використовують лише цільову функцію, не залучаючи похідні чи іншу додаткову інформацію.

– *Рандомізація операцій*: Генетичні алгоритми застосовують ймовірнісні, а не детерміновані, правила для вибору та виконання операцій.

Ці особливості дозволяють генетичним алгоритмам бути потужним інструментом для розв’язання складних задач оптимізації, де традиційні методи можуть бути менш ефективними.

Перераховані властивості забезпечують високу стійкість генетичних алгоритмів [16-17].

Генетичний алгоритм було реалізовано за допомогою мови C# з використання розрахункових даних, що наведені у Додатку Б.

На рисунках 2.4 – 2.7 наведені фрагменту лістингу коду, що відповідають завданню початкових даних, схрещуванню та ініціалізації.

```
using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
    const int PopulationSize = 100;
    const int Generations = 200;
    const double MutationRate = 0.05;
    const double CrossoverRate = 0.7;
    const int NumVehicles = 100;

    static Random rand = new Random();
```

Рисунок 2.4 – Лістинг створення розрахунку

```
// Вхідні дані
static double[] fuelConsumption = new double[NumVehicles]; // л/100км
static double[] capacities = new double[NumVehicles];      // вантажопідйомність
static double distance1 = 50; // км до Складу 1
static double distance2 = 80; // км до Складу 2
static double totalCargo = 500; // загальний вантаж, що треба перевезти

static void Main()
{
    // Генеруємо випадкові характеристики авто
    for (int i = 0; i < NumVehicles; i++)
    {
        fuelConsumption[i] = rand.Next(15, 30); // від 15 до 30 л/100 км
        capacities[i] = rand.Next(3, 10);      // від 3 до 10 тонн
    }
}
```

Рисунок 2.5 – Лістинг вхідні дані

```
// Схрещування
while (newPopulation.Count < PopulationSize)
{
    var parent1 = TournamentSelection(scored);
    var parent2 = TournamentSelection(scored);

    int[] child = rand.NextDouble() < CrossoverRate
        ? Crossover(parent1, parent2)
        : (int[])parent1.Clone();

    Mutation(child);
    newPopulation.Add(child);
}

population = newPopulation;
}
```

Рисунок 2.6 – Лістинг схрещування

```
// Ініціалізація популяції
List<int[]> population = new List<int[]>();
for (int i = 0; i < PopulationSize; i++)
    population.Add(GenerateRandomChromosome());

for (int gen = 0; gen < Generations; gen++)
{
    // Оцінка
    var scored = population.Select(ch => new { Chromosome = ch, Fitness = Fitness(ch) })
        .OrderBy(x => x.Fitness)
        .ToList();

    // Вивід прогресу
    if (gen % 10 == 0)
        Console.WriteLine($"Покоління {gen}, мін. витрати палива: {scored[0].Fitness:F2}");
}
```

Рисунок 2.7 – Лістинг ініціалізації популяції

Для прикладу наведемо результати розрахунків оптимізації плану транспортних перевезень для Зона – Склад 1.

```
Results: 17395.42 [12.815s]
[ [ 0, 4, 0, 0, 0, 0, 0, 0, 0, 0 ],
  [ 0, 3, 0, 0, 0, 0, 1, 0, 0, 0 ],
  [ 0, 0, 0, 1, 3, 0, 0, 0, 0, 0 ],
  [ 1, 0, 0, 0, 2, 0, 0, 0, 1, 0 ],
  [ 1, 1, 1, 1, 0, 0, 0, 0, 0, 0 ],
  [ 1, 0, 0, 0, 1, 0, 1, 1, 0, 0 ],
  [ 1, 0, 0, 3, 0, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 3, 0, 1, 0, 0 ],
  [ 0, 0, 1, 0, 0, 2, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 1, 0, 3, 0 ] ]
```

Рисунок 2.8 - Розрахунок для зон 1-10

```
Results: 16275.63 [16.110s]
[ [ 0, 0, 0, 0, 2, 2, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 4, 0, 0 ],
  [ 2, 0, 0, 0, 0, 1, 0, 0, 1, 0 ],
  [ 0, 0, 0, 1, 2, 0, 0, 0, 1, 0 ],
  [ 0, 0, 0, 1, 1, 0, 2, 0, 0, 0 ],
  [ 0, 0, 0, 1, 2, 0, 0, 0, 1, 0 ],
  [ 1, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 2, 1, 1 ],
  [ 2, 0, 0, 0, 0, 2, 0, 0, 0, 0 ],
  [ 2, 0, 0, 0, 0, 0, 1, 0, 1, 0 ] ]
```

Рисунок 2.9 - Розрахунок для зон 11-20

```
Results: 16391.22 [27.073s]
[ [ 2, 0, 0, 0, 0, 0, 1, 1, 0, 0 ],
  [ 0, 1, 0, 0, 0, 0, 1, 2, 0, 0 ],
  [ 0, 0, 0, 0, 4, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 1, 0, 1, 0, 0, 2, 0 ],
  [ 0, 3, 0, 0, 1, 0, 0, 0, 0, 0 ],
  [ 0, 0, 1, 1, 1, 1, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 4, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 1, 1, 1, 1, 0 ],
  [ 0, 0, 0, 2, 0, 0, 0, 0, 2, 0 ],
  [ 1, 1, 0, 0, 2, 0, 0, 0, 0, 0 ] ]
```

Рисунок 2.10 - Розрахунок для зон 21-30

```
Results: 16457.75 [33.047s]
[ [ 1, 2, 0, 0, 1, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 3, 0, 0, 0, 1 ],
  [ 0, 0, 0, 1, 0, 0, 2, 1, 0, 0 ],
  [ 0, 0, 0, 0, 1, 2, 0, 1, 0, 0 ],
  [ 1, 0, 0, 2, 0, 0, 1, 0, 0, 0 ],
  [ 0, 0, 0, 1, 1, 1, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 1, 0, 1, 0, 0 ],
  [ 0, 2, 0, 0, 0, 1, 1, 0, 0, 0 ],
  [ 1, 0, 0, 0, 0, 2, 0, 0, 1, 0 ],
  [ 0, 1, 1, 0, 1, 0, 1, 0, 0, 0 ] ]
```

Рисунок 2.11 - Розрахунок для зон 31-40

```
Results: 15212.38 [113.162s]
[ [ 0, 0, 0, 0, 0, 1, 0, 2, 1, 0 ],
  [ 0, 0, 0, 2, 1, 0, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 0, 2, 0, 0, 0 ],
  [ 0, 1, 0, 0, 0, 0, 2, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 0, 2, 0, 0, 0 ],
  [ 1, 0, 0, 0, 1, 1, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 0, 1, 1, 1, 1, 0 ],
  [ 1, 0, 0, 1, 1, 0, 0, 0, 1, 0 ],
  [ 0, 1, 0, 0, 0, 1, 0, 0, 2, 0 ],
  [ 0, 1, 0, 0, 1, 1, 1, 0, 0, 0 ] ]
```

Рисунок 2.12 - Розрахунок для зон 41-50

Результати розрахунків для Зона – Склад 2 наведено у Додатку В:

2.6 Підсумковий аналіз отриманих результатів оптимізації

Для зручності порівняння отриманих результатів представимо їх у табличному вигляді (дивись таблицю 2.11)

Таблиця 2.11 – Підсумкові результати

№ кластеру	MS Excel	MATLAB	Програмний засіб
Оптимізація перевезень по маршруту Зона-Склад1			
1	12363.4 грн	35598.7 грн	17943.95 грн
2	9106.9 грн	23305.6 грн	12151.3 грн
3	12489.6 грн	32149.0 грн	18622.7 грн
4	12041.5 грн	31279.8 грн	16852.88 грн
5	9354.8 грн	25384.67 грн	15091.83 грн
Оптимізація перевезень по маршруту Зона-Склад2			
1	13394.5 грн	38380.45 грн	19604.12 грн
2	10176.85 грн	25292.5 грн	15645.65 грн
3	12835.35 грн	30862.5 грн	18788.95 грн
4	13820.8 грн	40251.3 грн	18954.8 грн
5	10485.8 грн	27359.5 грн	15302.55 грн

Для аналізу впливу вхідних параметрів на час виконання алгоритму було проведено 10 тестів з використанням першого набору даних. Результати тестування представлені в таблиці 2.12.

Таблиця 2.12 – Продуктивність роботи програмного засобу

№ Тесту	Початкові умови		Результат роботи алгоритму, грн	Час роботи алгоритму, сек
	Розмір популяції	Кількість генерацій		
1	100	500	17395.42	12.815
2	100	1000	16275.63	16.110
3	200	500	16391.22	27.073
4	200	1000	16457.75	33.047
5	1000	200	15212.38	113.162
6	2000	200	14047.15	234.420
7	500	3000	17129.58	149.094
8	1000	5000	14676.45	446.450
9	5000	1000	15484.08	993.404
10	2500	2500	14252.13	726.708

Для більш наглядного представлення проведемо порівняння у графічному вигляді:

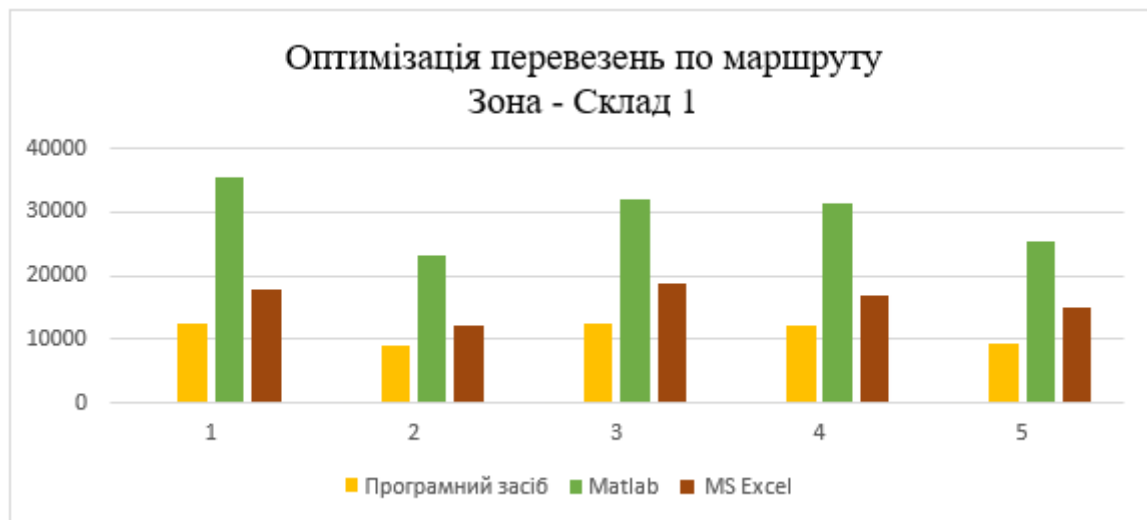


Рисунок 2.13 – Зона – Склад 1



Рисунок 2.14 – Зона – Склад 2

Висновки до розділу:

У результаті проведеного аналізу застосування інтелектуальних технологій для вирішення логістичних завдань було показано, що сучасні технології, такі як електронний обмін даними, безконтактна ідентифікація, маркування та програмні комплекси для управління складами (WMS), значно оптимізують процеси в логістиці, знижуючи витрати та підвищуючи ефективність.

Зокрема, генетичний алгоритм (ГА) виявився потужним інструментом для оптимізації плану перевезень. Порівняння різних методів оптимізації, зокрема використання Microsoft Excel та Matlab, показало, що генетичний алгоритм демонструє найкращі результати, забезпечуючи ефективне вирішення задач у складних та великих просторах пошуку. Генетичні алгоритми, завдяки своїй здатності адаптуватися до змінних умов та працювати з великими обсягами даних, виявилися найбільш ефективним інструментом для оптимізації перевезень.

Загалом, застосування генетичних алгоритмів для логістичних задач, дозволяє значно покращити управлінські рішення, знизити витрати і підвищити загальну ефективність логістичних процесів.

ВИСНОВКИ

Маршрутизація автотранспортних засобів за допомогою евристичних методів є простим і ефективним підходом, що дозволяє швидко знаходити рішення. Однак цей метод не гарантує отримання оптимального результату на 100%. Наразі розробляються нові підходи, які поєднують гнучкість евристичних методів та точність моделей лінійного програмування, що дозволяє отримувати оптимальні або принаймні кращі результати. Ці методи безумовно є майбутнім у транспортній логістиці.

На сучасному етапі розвитку глобального виробництва логістика стала однією з найбільш розвинених галузей економіки. Це стало можливим завдяки активному та всеосяжному впровадженню інтелектуальних інформаційних технологій, що автоматизують і спрощують робочі процеси.

Зміни торкнулися всього: від алгоритмів обробки даних до потужностей і загальних технологій. Все це спрямовано на здешевлення організації інфраструктури та полегшення діяльності компаній. Основною особливістю логістичних підприємств є необхідність відстеження всіх операцій в режимі реального часу та їх поетапного аналізу, що дозволяє приймати оперативні рішення для покращення якості послуг. Проте існує небагато алгоритмів, які здатні задовольнити ці вимоги, а деякі навіть ускладнюють завдання, що уповільнює впровадження інтелектуальних технологій у логістичну сферу.

Під час виконання роботи були здійснені наступні етапи:

- Проаналізовано поточний стан на ринку програмного забезпечення для оптимізації логістичних задач.
- Досліджено визначення логістики як процесу.
- Проведено аналіз найбільш поширених алгоритмів оптимізації.
- Оцінено застосування інтелектуальних інформаційних технологій у сфері логістики.
- Реалізовано практичну оптимізацію плану перевезень із використанням трьох різних методів оптимізації.

– Проведено порівняльний аналіз результатів оптимізації, на основі яких зроблено висновок про доцільність використання генетичного алгоритму для вирішення задачі оптимізації логістичних процесів.

У результаті проведеного аналізу застосування інтелектуальних технологій для вирішення логістичних завдань було показано, що сучасні технології, такі як електронний обмін даними, безконтактна ідентифікація, маркування та програмні комплекси для управління складами (WMS), значно оптимізують процеси в логістиці, знижуючи витрати та підвищуючи ефективність.

Зокрема, генетичний алгоритм (ГА) виявився потужним інструментом для оптимізації плану перевезень. Порівняння різних методів оптимізації, зокрема використання Microsoft Excel та Matlab, показало, що генетичний алгоритм демонструє найкращі результати, забезпечуючи ефективне вирішення задач у складних та великих просторах пошуку. Генетичні алгоритми, завдяки своїй здатності адаптуватися до змінних умов та працювати з великими обсягами даних, виявилися найбільш ефективним інструментом для оптимізації перевезень.

Загалом, застосування генетичних алгоритмів для логістичних задач, зокрема оптимізації плану перевезень, дозволяє значно покращити управлінські рішення, знизити витрати і підвищити загальну ефективність логістичних процесів.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Lp_solve reference guide menu (2023). url: http://lpsolve.sourceforge.net/5.5/get_sensitivity_obj.htm. (дата звернення: 15.04.2025)
- 2 Web-site of NEOS Server url: <http://www.neos-server.org/neos/> (дата звернення: 15.04.2025)
- 3 Gurobi. url: <https://living-sun.com/uk/programming-languages/680522-programming-language-suggestion-dynamic-and-multidimensional-array-cplex-or-gurobi-programming-languages-dynamic-arrays-cplex-gurobi.html> (дата звернення: 15.04.2025)
- 4 CPLEX Optimizer. High-performance mathematical programming solver for linear programming, mixed-integer programming and quadratic programming. url: <https://www.ibm.com/analytics/cplex-optimizer> (дата звернення: 15.04.2025)
- 5 Gurobi Optimization, Inc., Gurobi Optimizer Reference Manual, 2014. <http://www.gurobi.com/> (дата звернення: 15.04.2025)
- 6 Tawarmalani M., Sahinidis N.V. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*. 2005. 103 (2). P. 225–249.
- 7 Sahinidis N.V. BARON 21.1.13: Global Optimization of Mixed-Integer Nonlinear Programs. User's manual. 2021.
- 8 Kilinc M., Sahinidis N.V. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems in BARON. *Optimization Methods and Software*. 2018. 33. P. 540–562.
- 9 M. Dorigo, M. Birattari & T. Stützle, 2006 Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique.
- 10 Глушков В. М., Івахненко О. М. Генетичні алгоритми та їх застосування. Київ : Наукова думка, 2011. 256 с.
- 11 Blanchard, B. S. *Logistics Engineering and Management*. 6th ed. Upper Saddle River : Pearson Education, 2004. 528 p.

- 12 Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Boston : Addison-Wesley, 1989. 412 p.
- 13 Crainic, T. G. Metaheuristics for Logistics. *International Series in Operations Research & Management Science*. New York : Springer, 2019. 408 p.
- 14 Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem / *Computers & Operations Research*/2004. Vol. 31, No. 12. P. 1985–2002.
- 15 R. Balamurugan, S. S. Sridhar, R. S. Rajesh. Application of Genetic Algorithms in Supply Chain Optimization / *International Journal of Computer Applications*. 2012. Vol. 43, No. 6. P. 35–39.
- 16 A. Sharma, M. Tiwari. Optimization of warehouse layout using genetic algorithms / *International Journal of Production Research* / 2015. Vol. 53, No. 13. P. 4020–4032.
- 17 Rai S. and Ettam R. K. Simulation-based optimization using simulated annealing for optimal equipment selection within print production environments// Winter Simulations Conference (WSC), 2013. PP. 1097-1108.
- 18 Лук'яненко С.О. Числові методи в інформатиці: Навч. посіб. Київ: “Видавництво “Політехніка”, 2007. 140с.
- 19 Крикавський Е.П. Логістика: Навч. посіб. для вузів. Львів: Львівська політехніка, 2009. 263 с.
- 20 Харрісон Алан, Ван Хоук Ремко. Управління логістикою: Розробка стратегій логістичних операцій. Дніпропетровськ: Баланс Бізнес Букс, 2007.
- 21 Горяїнов О.М. Теорія і практика дисципліни «Логістика» (для менеджерів): Навчальний посібник. Харків: НТМТ, 2009. 283 с.
- 22 MOSEK optimization toolbox for MATLAB manual (2023). url: http://docs.mosek.com/7.0/toolbox/Sensitivity_analysis.html. (дата звернення: 10.05.2025)

- 23 Patrascu, M.; Stancu, A.F.; Pop, F. (2014). "HELGA: a heterogeneous encoding lifelike genetic algorithm for population evolution modeling and simulation". *Soft Computing*. 18. P. 2565-2576.
- 24 J. Wang and H. Yuan (2017). System Dynamics Approach for Investigating the Risk Effects on Schedule Delay in Infrastructure Projects/ *Journal of Management in Engineering*
- 25 Шилдт, Г. Програмування на C#, 6-е вид. Київ : Видавництво "Основа", 2018. 784 с.
- 26 Албахарі, Дж. C# 9.0 in a Nutshell: The Definitive Reference, 1-е вид. Нью-Йорк : O'Reilly Media, 2020. 1184 с.
- 27 Тротман, Е. C# для професіоналів. Київ : Видавництво "Основа", 2017. 512 с.
- 28 Моркун Н. В., Завсегдашня І. В. Методичні вказівки до виконання кваліфікаційної роботи бакалавру для студентів спеціальності 122 "Комп'ютерні науки". Кривий Ріг : Видавничий центр КНУ, 2021. 50 с.
- 29 ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).
- 30 ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).
- 31 ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).

ДОДАТОК А

Розрахункові дані

Таблиця А.1 – Характеристики машин

№	Марка авто	Витрати палива, л/100 км	Вантажопідйом ність, т	Кількість машин
1	DAF 95 XF 380	35	20	8
2	DAF 95 XF 430	27	20	10
3	KAMAZ 5320	40	8	8
4	KAMAZ 6520	49	20	12
5	MAN TGS 33.440	35	30	12
6	MAN TGX 18.400	20.4	20	10
7	Mercedes-Benz Actros 1843	28	18	8
8	Renault Premium 420	35	39	12
9	Зил 554М	23	9	12
10	Зил-130	35.2	6	8

Таблиця А.2 - Таблиця відстаней між об'єктами

№ зони	Відстань (км) Зона - Склад 1	Відстань (км) Зона - Склад 2	Вантаж, т
1	9	8	121
2	9	10	287
3	15	15	206
4	7	7	116
5	9	7	259
6	14	15	270
7	13	8	293
8	14	14	253
9	8	7	181
10	14	13	177
11	10	6	112
12	14	10	191
13	7	8	288
14	5	8	265

15	12	6	281
16	5	9	146
17	9	11	238
18	14	15	150
19	14	6	221
20	5	6	124
21	7	13	224
22	8	11	281
23	10	11	283
24	14	13	113
25	14	8	284
26	9	5	278
27	14	5	295
28	9	13	216
29	7	12	285
30	6	7	154
31	10	6	292
32	7	10	231
33	6	12	291
34	9	8	219
35	12	12	251
36	14	15	226
37	14	6	174
38	13	10	267
39	11	7	259
40	11	6	226
41	8	5	278
42	11	13	224
43	9	9	260
44	15	15	183
45	10	11	145
46	12	5	123
47	8	7	147
48	7	7	204
49	10	6	164
50	12	14	183

Таблиця А.3 - Загальні витрати на перевезення за маршрутом Зона-Склад 1

Номер зони	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130
1	980	756	2560	1372	700	571.2	784	560	1288	2956.8
2	2625	2025	7200	3675	1750	1530	2240	1400	3680	8448
3	2887.5	2227.5	7800	4042.5	1837.5	1683	2520	1575	3967.5	9240
4	735	567	2100	1029	490	428.4	686	367.5	1046.5	2464
5	1592.5	1228.5	4620	2229.5	1102.5	928.2	1470	857.5	2334.5	5420.8
6	3675	2835	10200	5145	2362.5	2142	3150	1837.5	5175	11880
7	2100	1620	5920	2940	1400	1224	1904	1120	3036	6899.2
8	3185	2457	8960	4459	2205	1856.4	2940	1715	4669	10595.2
9	1225	945	3220	1715	857.5	714	1078	612.5	1690.5	3819.2
10	2047.5	1579.5	5980	2866.5	1365	1193.4	1820	1137.5	2990	6864
11	630	486	1680	882	420	367.2	588	315	897	2006.4
12	1750	1350	4800	2450	1225	1020	1540	875	2530	5632
13	2100	1620	5760	2940	1400	1224	1792	1120	2944	6758.4
14	1960	1512	5440	2744	1260	1142.4	1680	980	2760	6336
15	1575	1215	4320	2205	1050	918	1344	840	2208	4963.2
16	1260	972	3420	1764	787.5	734.4	1134	630	1759.5	3960
17	2310	1782	6600	3234	1540	1346.4	2156	1347.5	3415.5	7744
18	2100	1620	5700	2940	1312.5	1224	1890	1050	2932.5	6600
19	1260	972	3360	1764	840	734.4	1092	630	1725	3907.2
20	735	567	1920	1029	525	428.4	588	420	966	2217.6
21	2730	2106	7280	3822	1820	1591.2	2366	1365	3737.5	8694.4
22	2887.5	2227.5	7920	4042.5	1925	1683	2464	1540	4048	9099.2
23	2887.5	2227.5	7920	4042.5	1925	1683	2464	1540	4048	9292.8
24	1365	1053	3900	1911	910	795.6	1274	682.5	1943.5	4347.2
25	2100	1620	5760	2940	1400	1224	1792	1120	2944	6758.4
26	1225	945	3500	1715	875	714	1120	700	1782.5	4136

27	1312.5	1012.5	3700	1837.5	875	765	1190	700	1897.5	4400
28	2502.5	1930.5	7020	3503.5	1820	1458.6	2184	1365	3588	8236.8
29	3150	2430	8640	4410	2100	1836	2688	1680	4416	10137.6
30	980	756	2800	1372	735	571.2	882	490	1449	3203.2
31	1575	1215	4440	2205	1050	918	1428	840	2277	5174.4
32	2100	1620	5800	2940	1400	1224	1820	1050	2990	6864
33	3150	2430	8880	4410	2100	1836	2856	1680	4554	10348.8
34	1540	1188	4480	2156	1120	897.6	1456	840	2300	5209.6
35	2730	2106	7680	3822	1890	1591.2	2352	1470	3864	8870.4
36	3150	2430	8700	4410	2100	1836	2730	1575	4485	10032
37	945	729	2640	1323	630	550.8	840	525	1380	3062.4
38	2450	1890	6800	3430	1575	1428	2100	1225	3450	7920
39	1592.5	1228.5	4620	2229.5	1102.5	928.2	1470	857.5	2334.5	5420.8
40	1260	972	3480	1764	840	734.4	1092	630	1794	4012.8
41	1225	945	3500	1715	875	714	1120	700	1782.5	4136
42	2730	2106	7280	3822	1820	1591.2	2366	1365	3737.5	8694.4
43	2047.5	1579.5	5940	2866.5	1417.5	1193.4	1890	1102.5	3001.5	6969.6
44	2625	2025	6900	3675	1837.5	1530	2310	1312.5	3622.5	8184
45	1540	1188	4180	2156	962.5	897.6	1386	770	2150.5	4840
46	612.5	472.5	1600	857.5	437.5	357	490	350	805	1848
47	980	756	2660	1372	612.5	571.2	882	490	1368.5	3080
48	1347.5	1039.5	3640	1886.5	857.5	785.4	1176	735	1851.5	4188.8
49	945	729	2520	1323	630	550.8	840	525	1311	2956.8
50	2450	1890	6440	3430	1715	1428	2156	1225	3381	7638.4

Таблиця А.4 - Загальні витрати на перевезення за маршрутом Зона –
Склад 2

Номер зони	DAF 95 XF 380	DAF 95 XF 430	KAMAZ 5320	KAMAZ 6520	MAN TGS 33.440	MAN TGX 18.400	Mercedes-Benz Actros 1843	Renault Premium 420	Зил 554М	Зил-130
1	1102.5	850.5	2880	1543.5	787.5	642.6	882	630	1449	3326.4
2	2362.5	1822.5	6480	3307.5	1575	1377	2016	1260	3312	7603.2
3	2887.5	2227.5	7800	4042.5	1837.5	1683	2520	1575	3967.5	9240
4	735	567	2100	1029	490	428.4	686	367.5	1046.5	2464
5	2047.5	1579.5	5940	2866.5	1417.5	1193.4	1890	1102.5	3001.5	6969.6
6	3430	2646	9520	4802	2205	1999.2	2940	1715	4830	11088
7	3412.5	2632.5	9620	4777.5	2275	1989	3094	1820	4933.5	11211.2
8	3185	2457	8960	4459	2205	1856.4	2940	1715	4669	10595.2
9	1400	1080	3680	1960	980	816	1232	700	1932	4364.8
10	2205	1701	6440	3087	1470	1285.2	1960	1225	3220	7392
11	1050	810	2800	1470	700	612	980	525	1495	3344
12	2450	1890	6720	3430	1715	1428	2156	1225	3542	7884.8
13	1837.5	1417.5	5040	2572.5	1225	1071	1568	980	2576	5913.6
14	1225	945	3400	1715	787.5	714	1050	612.5	1725	3960
15	3150	2430	8640	4410	2100	1836	2688	1680	4416	9926.4
16	700	540	1900	980	437.5	408	630	350	977.5	2200
17	1890	1458	5400	2646	1260	1101.6	1764	1102.5	2794.5	6336
18	1960	1512	5320	2744	1225	1142.4	1764	980	2737	6160
19	2940	2268	7840	4116	1960	1713.6	2548	1470	4025	9116.8
20	612.5	472.5	1600	857.5	437.5	357	490	350	805	1848
21	1470	1134	3920	2058	980	856.8	1274	735	2012.5	4681.6
22	2100	1620	5760	2940	1400	1224	1792	1120	2944	6617.6
23	2625	2025	7200	3675	1750	1530	2240	1400	3680	8448
24	1470	1134	4200	2058	980	856.8	1372	735	2093	4681.6
25	3675	2835	10080	5145	2450	2142	3136	1960	5152	11827.2
26	2205	1701	6300	3087	1575	1285.2	2016	1260	3208.5	7444.8

27	3675	2835	10360	5145	2450	2142	3332	1960	5313	12320
28	1732.5	1336.5	4860	2425.5	1260	1009.8	1512	945	2484	5702.4
29	1837.5	1417.5	5040	2572.5	1225	1071	1568	980	2576	5913.6
30	840	648	2400	1176	630	489.6	756	420	1242	2745.6
31	2625	2025	7400	3675	1750	1530	2380	1400	3795	8624
32	1470	1134	4060	2058	980	856.8	1274	735	2093	4804.8
33	1575	1215	4440	2205	1050	918	1428	840	2277	5174.4
34	1732.5	1336.5	5040	2425.5	1260	1009.8	1638	945	2587.5	5860.8
35	2730	2106	7680	3822	1890	1591.2	2352	1470	3864	8870.4
36	2940	2268	8120	4116	1960	1713.6	2548	1470	4186	9363.2
37	2205	1701	6160	3087	1470	1285.2	1960	1225	3220	7145.6
38	3185	2457	8840	4459	2047.5	1856.4	2730	1592.5	4485	10296
39	2502.5	1930.5	7260	3503.5	1732.5	1458.6	2310	1347.5	3668.5	8518.4
40	2310	1782	6380	3234	1540	1346.4	2002	1155	3289	7356.8
41	1960	1512	5600	2744	1400	1142.4	1792	1120	2852	6617.6
42	2310	1782	6160	3234	1540	1346.4	2002	1155	3162.5	7356.8
43	2047.5	1579.5	5940	2866.5	1417.5	1193.4	1890	1102.5	3001.5	6969.6
44	2625	2025	6900	3675	1837.5	1530	2310	1312.5	3622.5	8184
45	1400	1080	3800	1960	875	816	1260	700	1955	4400
46	1470	1134	3840	2058	1050	856.8	1176	840	1932	4435.2
47	1120	864	3040	1568	700	652.8	1008	560	1564	3520
48	1347.5	1039.5	3640	1886.5	857.5	785.4	1176	735	1851.5	4188.8
49	1575	1215	4200	2205	1050	918	1400	875	2185	4928
50	2100	1620	5520	2940	1470	1224	1848	1050	2898	6547.2

ДОДАТОК Б

Лістинг програмної реалізації генетичного алгоритму

```
using System;
using System.Collections.Generic;
using System.Linq;
class Program
{
    const int PopulationSize = 100;
    const int Generations = 200;
    const double MutationRate = 0.05;
    const double CrossoverRate = 0.7;
    const int NumVehicles = 100;
    static Random rand = new Random();
    // Вхідні дані
    static double[] fuelConsumption = new double[NumVehicles]; // л/100км
    static double[] capacities = new double[NumVehicles]; // вантажопідйомність
    static double distance1 = 50; // км до Складу 1
    static double distance2 = 80; // км до Складу 2
    static double totalCargo = 500; // загальний вантаж, що треба перевезти
    static void Main()
    {
        // Генеруємо випадкові характеристики авто
        for (int i = 0; i < NumVehicles; i++)
        {
            fuelConsumption[i] = rand.Next(15, 30); // від 15 до 30 л/100 км
            capacities[i] = rand.Next(3, 10); // від 3 до 10 тонн
        }
        // Ініціалізація популяції
        List<int[]> population = new List<int[]>();
        for (int i = 0; i < PopulationSize; i++)
            population.Add(GenerateRandomChromosome());
    }
}
```

```

for (int gen = 0; gen < Generations; gen++)
{
    // Оцінка
    var scored = population.Select(ch => new { Chromosome = ch, Fitness = Fitness(ch)

})

        .OrderBy(x => x.Fitness)
        .ToList();

    // Вивід прогресу
    if (gen % 10 == 0)
        Console.WriteLine($"Покоління {gen}, мін. витрати палива:
{scored[0].Fitness:F2}");

    // Відбір
    List<int[]> newPopulation = new List<int[]>();

    // Елітизм — беремо 10 найкращих
    for (int i = 0; i < 10; i++)
        newPopulation.Add(scored[i].Chromosome);

    // Схрещування
    while (newPopulation.Count < PopulationSize)
    {
        var parent1 = TournamentSelection(scored);
        var parent2 = TournamentSelection(scored);

        int[] child = rand.NextDouble() < CrossoverRate
            ? Crossover(parent1, parent2)
            : (int[])parent1.Clone();

        Mutation(child);
        newPopulation.Add(child);
    }

    population = newPopulation;
}

// Показ найкращого рішення
var best = population.OrderBy(ch => Fitness(ch)).First();
Console.WriteLine("\nНайкраще розподілення:");

```

```

Console.WriteLine($"Загальні витрати палива: {Fitness(best):F2}");

int count1 = best.Count(g => g == 0);
int count2 = best.Count(g => g == 1);
Console.WriteLine($"На склад 1: {count1} авто, на склад 2: {count2} авто");
}

static int[] GenerateRandomChromosome()
{
    int[] chromosome = new int[NumVehicles];
    for (int i = 0; i < NumVehicles; i++)
        chromosome[i] = rand.Next(2); // 0 або 1
    return chromosome;
}

static double Fitness(int[] chromosome)
{
    double totalFuel = 0;
    double totalCapacity = 0;
    for (int i = 0; i < NumVehicles; i++)
    {
        double dist = chromosome[i] == 0 ? distance1 : distance2;
        totalFuel += (fuelConsumption[i] / 100.0) * dist;
        totalCapacity += capacities[i];
    }
    // Якщо недостатньо вантажу — штраф
    if (totalCapacity < totalCargo)
        totalFuel += 10000;
    return totalFuel;
}

static int[] Crossover(int[] parent1, int[] parent2)
{
    int[] child = new int[NumVehicles];
    int point = rand.Next(NumVehicles);
    for (int i = 0; i < NumVehicles; i++)

```

```

        child[i] = i < point ? parent1[i] : parent2[i];
    }
    return child;
}

static void Mutation(int[] chromosome)
{
    for (int i = 0; i < NumVehicles; i++)
        if (rand.NextDouble() < MutationRate)
            chromosome[i] = 1 - chromosome[i]; // інверсія
    }

static int[] TournamentSelection(List<dynamic> population)
{
    int best = rand.Next(PopulationSize);
    for (int i = 0; i < 2; i++)
    {
        int idx = rand.Next(PopulationSize);
        if (population[idx].Fitness < population[best].Fitness)
            best = idx;
    }
    return population[best].Chromosome;
}
}

```

ДОДАТОК В

Отримані результати оптимізації логістичних перевезень:

Зона Склад 2

```
Results: 17395.42 [12.815s]
[ [ 0, 4, 0, 0, 0, 0, 0, 0, 0, 0 ],
  [ 0, 3, 0, 0, 0, 0, 1, 0, 0, 0 ],
  [ 0, 0, 0, 1, 3, 0, 0, 0, 0, 0 ],
  [ 1, 0, 0, 0, 2, 0, 0, 0, 1, 0 ],
  [ 1, 1, 1, 1, 0, 0, 0, 0, 0, 0 ],
  [ 1, 0, 0, 0, 1, 0, 1, 1, 0, 0 ],
  [ 1, 0, 0, 3, 0, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 3, 0, 1, 0, 0 ],
  [ 0, 0, 1, 0, 0, 2, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 1, 0, 3, 0 ] ]
```

Рисунок В.1 - Розрахунок для зон 1-10

```
Results: 16275.63 [16.110s]
[ [ 0, 0, 0, 0, 2, 2, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 4, 0, 0 ],
  [ 2, 0, 0, 0, 0, 1, 0, 0, 1, 0 ],
  [ 0, 0, 0, 1, 2, 0, 0, 0, 1, 0 ],
  [ 0, 0, 0, 1, 1, 0, 2, 0, 0, 0 ],
  [ 0, 0, 0, 1, 2, 0, 0, 0, 1, 0 ],
  [ 1, 0, 0, 0, 0, 0, 1, 2, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 2, 1, 1 ],
  [ 2, 0, 0, 0, 0, 2, 0, 0, 0, 0 ],
  [ 2, 0, 0, 0, 0, 0, 1, 0, 1, 0 ] ]
```

Рисунок В.2 - Розрахунок для зон 11-20

```
Results: 16391.22 [27.073s]
[ [ 2, 0, 0, 0, 0, 0, 1, 1, 0, 0 ],
  [ 0, 1, 0, 0, 0, 0, 1, 2, 0, 0 ],
  [ 0, 0, 0, 0, 4, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 1, 0, 1, 0, 0, 2, 0 ],
  [ 0, 3, 0, 0, 1, 0, 0, 0, 0, 0 ],
  [ 0, 0, 1, 1, 1, 1, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 4, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 1, 1, 1, 1, 0 ],
  [ 0, 0, 0, 2, 0, 0, 0, 0, 2, 0 ],
  [ 1, 1, 0, 0, 2, 0, 0, 0, 0, 0 ] ]
```

Рисунок В.3- Розрахунок для зон 21-30

```
Results: 16457.75 [33.047s]
[ [ 1, 2, 0, 0, 1, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 3, 0, 0, 0, 1 ],
  [ 0, 0, 0, 1, 0, 0, 2, 1, 0, 0 ],
  [ 0, 0, 0, 0, 1, 2, 0, 1, 0, 0 ],
  [ 1, 0, 0, 2, 0, 0, 1, 0, 0, 0 ],
  [ 0, 0, 0, 1, 1, 1, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 1, 0, 1, 0, 0 ],
  [ 0, 2, 0, 0, 0, 1, 1, 0, 0, 0 ],
  [ 1, 0, 0, 0, 0, 2, 0, 0, 1, 0 ],
  [ 0, 1, 1, 0, 1, 0, 1, 0, 0, 0 ] ]
```

Рисунок В.4 - Розрахунок для зон 31-40

```
Results: 15212.38 [113.162s]
[ [ 0, 0, 0, 0, 0, 1, 0, 2, 1, 0 ],
  [ 0, 0, 0, 2, 1, 0, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 0, 2, 0, 0, 0 ],
  [ 0, 1, 0, 0, 0, 0, 2, 1, 0, 0 ],
  [ 0, 0, 0, 0, 2, 0, 2, 0, 0, 0 ],
  [ 1, 0, 0, 0, 1, 1, 0, 1, 0, 0 ],
  [ 0, 0, 0, 0, 0, 1, 1, 1, 1, 0 ],
  [ 1, 0, 0, 1, 1, 0, 0, 0, 1, 0 ],
  [ 0, 1, 0, 0, 0, 1, 0, 0, 2, 0 ],
  [ 0, 1, 0, 0, 1, 1, 1, 0, 0, 0 ] ]
```

Рисунок В.5 - Розрахунок для зон 41-50