

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи та алгоритми оптимізації логістичних процесів в складних
системах»

на здобуття освітнього ступеня бакалавра

зі спеціальності 124 Системний аналіз

(код, найменування спеціальності)

освітньо-професійної програми Системний аналіз

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

_____, Станіслав БОГДАН
(підпис) (ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти група САД-41

Станіслав БОГДАН
(ім'я, ПРІЗВИЩЕ)

Керівник:

канд.тех.наук,
доцент

Ігор ПАТРАКЕСВ
(ім'я, ПРІЗВИЩЕ)

Рецензент:

к.т.н., доцент

(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти бакалавр
Спеціальність 124 Системний аналіз
Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри ІПЗАС
Каміла СТОРЧАК
“ ____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Богдану Станіславу Валентиновичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Методи та алгоритми оптимізації логістичних процесів в складних системах

Керівник кваліфікаційної роботи: Ігор ПАТРАКЕСВ канд. тех. наук, доцент
(ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від 20 лютого 2026р. № 51

2. Строк подання кваліфікаційної роботи «01» червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Теоретичні положення транспортної логістики, задач TSP та VRP.
2. Матриця відстаней транспортної мережі міста Києва та перелік точок доставки.
3. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз теоретичних основ оптимізації логістичних процесів.
2. Дослідження алгоритмів оптимізації маршрутів та їх теоретичне обґрунтування.
3. Практична реалізація та порівняльний аналіз методів оптимізації логістичних маршрутів

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/ п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02-03.03.26	
2.	Аналіз теоретичних основ оптимізації логістичних процесів	04.03-17.03.26	
3.	Дослідження алгоритмів оптимізації маршрутів.	18.03-31.03.26	
4.	Практична реалізація та порівняльний аналіз методів оптимізації логістичних маршрутів.	01.04-24.04.26	
5.	Висновки по роботі	25.04-16.05.26	
6.	Розробка демонстраційних матеріалів, доповідь.	19.05-20.05.26	
7.	Оформлення бакалаврської роботи	21.05-25.05.26	

Здобувач вищої освіти

(підпис)

Станіслав БОГДАН

(ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Ігор ПАТРАКЕСВ

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

У дипломній роботі досліджено методи та алгоритми оптимізації логістичних процесів у складних транспортно-розподільчих системах та виконано їх порівняльну апробацію на реальній задачі маршрутизації. Основну увагу приділено аналізу ефективності різних класів оптимізаційних алгоритмів в умовах несиметричних матриць відстаней, характерних для реальних міських транспортних мереж.

В роботі систематизовано теоретичні основи задачі комівояжера (TSP) та задачі маршрутизації транспортних засобів (VRP), розглянуто їх математичні постановки та доведено належність до класу NP-важких задач комбінаторної оптимізації. Проаналізовано сучасні підходи до розв'язання задач маршрутизації: точні методи (повний перебір, метод гілок і меж, динамічне програмування), конструктивні евристики та метоевристичні підходи. Розглянуто особливості несиметричної задачі комівояжера (ATSP) та вплив асиметричності матриці відстаней на вибір алгоритму оптимізації.

В процесі виконання дипломної роботи реалізовано три методи оптимізації маршрутів: метод найближчого сусіда (жадібна конструктивна евристика), метод заощаджень Кларка–Райта (ощадна евристика) та генетичний алгоритм (еволюційна метоевристика). Практична апробація виконана на реальній транспортній мережі міста Києва, що складається з центрального складу та 10 точок доставки з несиметричною матрицею відстаней розміром 11×11 .

Проведено моделювання роботи кожного алгоритму та виконано їх порівняльний аналіз за показниками якості рішення, обчислювальної складності та часу виконання. Встановлено, що всі три методи забезпечують суттєве скорочення загального пробігу — на 55–57% порівняно з базовим нераціональним плануванням (207 км проти 89,7–92,6 км). Найкращий результат показав генетичний алгоритм (89,7 км), що перевищив детерміновані евристики на 2,6–3,2% при допустимому зростанні часу обчислення.

За результатами дослідження сформульовано практичні рекомендації щодо вибору методу оптимізації залежно від розміру задачі, вимог до якості рішення та допустимого часу розрахунку. Запропоновано використання комбінованого підходу для динамічних задач маршрутизації: швидке початкове рішення за допомогою евристики з подальшим покращенням засобами генетичного алгоритму. Отримані результати можуть використовуватись у логістичних компаніях, службах кур'єрської доставки та поштових операторах для автоматизованого планування транспортних маршрутів у міському середовищі.

Кваліфікаційна робота містить 3 розділи, висновки, список використаних джерел та додатки.

Ключові слова: ЛОГІСТИКА, ОПТИМІЗАЦІЯ МАРШРУТІВ, ЗАДАЧА КОМІВОЯЖЕРА, TSP, VRP, НЕСИМЕТРИЧНА МАТРИЦЯ ВІДСТАНЕЙ, МЕТОД НАЙБЛИЖЧОГО СУСІДА, МЕТОД КЛАРКА–РАЙТА, ГЕНЕТИЧНИЙ АЛГОРИТМ, КОМБІНАТОРНА ОПТИМІЗАЦІЯ, ТРАНСПОРТНА МЕРЕЖА, ЕВРИСТИКА.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1	11
ТЕОРЕТИЧНІ ОСНОВИ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ПРОЦЕСІВ...	11
1.1. Логістика як наукова дисципліна та об'єкт оптимізації	11
1.2. Математичні моделі логістичних задач оптимізації	13
1.3. Класифікація методів оптимізації в логістиці	15
1.4. Огляд сучасного стану досліджень	17
1.5. Постановка задачі дослідження	19
1.6. Висновки до розділу 1	20
РОЗДІЛ 2	22
АЛГОРИТМИ ОПТИМІЗАЦІЇ МАРШРУТІВ: ТЕОРЕТИЧНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ.....	22
2.1. Жадібні конструктивні евристики.....	22
2.2. Метод заощаджень Кларка–Райта	24
2.3. Генетичні алгоритми	26
2.4. Теоретичне порівняння алгоритмів	30
2.5. Несиметрична задача комівояжера (ATSP) та її особливості	32
2.6. Метрики оцінювання ефективності алгоритмів	33
2.7. Висновки до розділу 2	33
РОЗДІЛ 3	35
ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ МАРШРУТІВ.....	35
3.1. Постановка задачі та опис вхідних даних	35
3.2. Метод найближчого сусіда	38
3.3. Метод заощаджень Кларка–Райта	41
3.4. Генетичний алгоритм	44
3.5. Порівняльний аналіз методів	47
3.6. Практичні рекомендації щодо впровадження алгоритмів оптимізації....	49
3.7. Висновки до розділу 3	54
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59

ВСТУП

Актуальність теми. В умовах стрімкого розвитку глобальних торговельних мереж, зростання обсягів електронної комерції та посилення конкуренції на ринку логістичних послуг оптимізація транспортних і розподільчих процесів стає дедалі важливішою. Логістичні витрати становлять від 10 до 30% від загальної вартості продукції, а нераціональна маршрутизація транспортних засобів залишається чи не найбільшою статтею зайвих витрат у ланцюгах постачання. Міжнародні дослідження стабільно показують, що грамотна оптимізація маршрутів здатна знизити транспортні витрати на 20–40% — і для цього зовсім не потрібно купувати нові автомобілі чи розширювати парк. [1; 2]

Задача маршрутизації транспортних засобів (Vehicle Routing Problem, VRP) та її окремий випадок; задача комівояжера (Travelling Salesman Problem, TSP); належать до класу NP-важких комбінаторних задач оптимізації. Навіть для відносно невеликих транспортних мереж повний перебір усіх можливих маршрутів перетворюється на практично нездійсненне завдання: для 10 точок доставки простір розв'язків містить $10! = 3\,628\,800$ варіантів. Саме тому пошук і порівняння ефективних алгоритмів — як простих детермінованих евристик, так і складніших еволюційних підходів — є одним із центральних завдань сучасної прикладної математики. [3; 4]

Особливої актуальності ця тема набуває в умовах українських міських транспортних мереж, де матриця відстаней майже ніколи не є симетричною — одностороннє сполучення, київський рельєф і особливості дорожньої мережі роблять шлях «туди» і «назад» різними, що одразу ставить питання про адаптацію класичних алгоритмів до реального міського середовища.

Зв'язок роботи з науковими програмами, планами, темами. Дипломна робота виконана відповідно до навчального плану підготовки бакалаврів та пов'язана з науковою діяльністю кафедри у галузі дослідження операцій, теорії

алгоритмів та комп'ютерних методів оптимізації складних систем.

Мета і завдання дослідження. Метою дипломної роботи є дослідження, систематизація та порівняльний аналіз методів і алгоритмів оптимізації логістичних процесів у складних транспортно-розподільчих системах, а також практична апробація цих методів на реальній задачі маршрутизації для транспортної мережі міста Києва.

Для досягнення поставленої мети вирішуються такі завдання:

- 1) провести аналіз сучасного стану проблеми оптимізації логістичних процесів та огляд існуючих методів розв'язання задачі маршрутизації транспортних засобів;
- 2) дослідити теоретичні основи задачі комівояжера (TSP) та задачі маршрутизації транспортних засобів (VRP) як базових моделей логістичної оптимізації;
- 3) реалізувати та порівняти три ключових методи оптимізації маршрутів: метод найближчого сусіда, метод заощаджень Кларка–Райта та генетичний алгоритм;
- 4) провести практичну апробацію методів на транспортній мережі міста Києва, що складається з центрального складу та 10 точок доставки з несиметричною матрицею відстаней;
- 5) виконати порівняльний аналіз ефективності методів за показниками якості рішення, обчислювальної складності та часу виконання;
- 6) сформулювати рекомендації щодо вибору методу оптимізації залежно від параметрів конкретної логістичної задачі.

Об'єктом дослідження є процеси планування та оптимізації маршрутів у складних логістичних системах із несиметричними матрицями відстаней.

Предметом дослідження є методи та алгоритми комбінаторної оптимізації, застосовувані для розв'язання задачі комівояжера та суміжних задач маршрутизації, зокрема: конструктивні жадібні евристики (метод найближчого сусіда), ощадні алгоритми (метод Кларка–Райта) та еволюційні метаевристики

(генетичний алгоритм).

Методи дослідження. У роботі використовуються методи теорії графів та комбінаторної оптимізації для формалізації задачі маршрутизації; методи обчислювальної математики та алгоритмічного аналізу для оцінювання складності алгоритмів; теорія еволюційних обчислень для обґрунтування генетичного алгоритму; методи системного аналізу та порівняльного оцінювання для аналізу результатів. Практична апробація виконується на реальних географічних даних транспортної мережі міста Києва із застосуванням несиметричних матриць відстаней.

Наукова новизна одержаних результатів. У роботі набуло подальшого розвитку порівняльне дослідження детермінованих евристичних і стохастичних еволюційних методів оптимізації маршрутів в умовах несиметричних матриць відстаней, характерних для реальних міських транспортних мереж. Вперше проведено кількісну оцінку ефективності трьох класів алгоритмів (жадібна евристика, ощадний метод, генетичний алгоритм) на єдиній тестовій задачі для транспортної мережі Києва із 11 вузлами та встановлено, що еволюційний підхід забезпечує перевагу у 2,6–3,2% порівняно з детермінованими евристиками при прийнятному зростанні обчислювальних витрат.

Практичне значення одержаних результатів. Розроблені та апробовані методи оптимізації маршрутів можуть бути безпосередньо застосовані в логістичних компаніях, службах кур'єрської доставки та поштових операторах для автоматизованого планування транспортних маршрутів. Практична реалізація на прикладі мережі доставки міста Києва підтвердила можливість скорочення загального пробігу на 55–57% порівняно з нераціональним базовим плануванням (207 км проти 89,7–92,6 км). Сформульовані рекомендації щодо вибору методу залежно від розміру задачі та вимог до часу розрахунку мають безпосередню практичну цінність для фахівців із логістики.

Загальний обсяг роботи складає 59 сторінок основного тексту, 20 таблиць та 4 рисунки. Список використаних джерел містить 12 найменувань.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ПРОЦЕСІВ

1.1. Логістика як наукова дисципліна та об'єкт оптимізації

Логістика — це міждисциплінарна наукова дисципліна та практична сфера діяльності, що вивчає процеси планування, організації, управління та контролю матеріальних, інформаційних і фінансових потоків від місця їх виникнення до кінцевого споживача. Саме слово походить від грецького «logistikos» — мистецтво розрахунку. У широкий вжиток воно увійшло у другій половині XX століття: спочатку в армії, де без чіткої організації постачання неможливо виграти жодної операції, а потім поступово перейшло і в бізнес.

Сучасна логістика охоплює широке коло завдань: управління запасами та складуванням, транспортування та маршрутизацію, митне оформлення, інформаційне забезпечення ланцюгів постачання, зворотню логістику та ін. Серед усіх цих напрямків особливе місце займає транспортна логістика — власне, безпосередній рух товарів між вузлами мережі.

Із системної точки зору логістична мережа являє собою складну систему — множину взаємопов'язаних елементів (вузлів і дуг), поведінка якої визначається нелінійними взаємодіями між компонентами та не може бути зведена до простої суми властивостей окремих елементів. Саме через цю складність і масштаб без математичного апарату оптимізації тут не обійтися.

1.1.1. Структура та складові логістичної системи

Логістична система в загальному вигляді складається з таких ключових компонентів:

- постачальники сировини та матеріалів — початкові вузли ланцюга постачання;
- виробничі підприємства — вузли перетворення вхідних ресурсів на готову продукцію;
- розподільчі центри та склади — вузли зберігання та

консолідації/декоміплектації вантажів;

- транспортні маршрути — дуги мережі, що з'єднують вузли та характеризуються відстанню, часом і вартістю переміщення;
- кінцеві споживачі — термінальні вузли, для яких здійснюється весь логістичний процес;
- інформаційна підсистема — забезпечує моніторинг, координацію та управління потоками в реальному часі.

Ефективність функціонування логістичної системи оцінюється за сукупністю показників: загальні логістичні витрати, рівень обслуговування клієнтів (своєчасність та повнота виконання замовлень), оборотність запасів, коефіцієнт використання транспортних засобів, та ін. Оптимізація передбачає досягнення найкращих значень цих показників при існуючих ресурсних обмеженнях.

1.1.2. Логістичні витрати та їх структура

Логістичні витрати є ключовим показником ефективності управління ланцюгом постачання. За даними Європейської логістичної асоціації (ELA), вони становлять у середньому 10–15% від виторгу компанії у промислово розвинених країнах і до 25–30% — у країнах з менш розвиненою інфраструктурою. Структура логістичних витрат включає:

- транспортні витрати (40–60% від загальних логістичних витрат) — паливе, амортизація, оплата праці водіїв, дорожні збори;
- витрати на утримання запасів (20–40%) — складські витрати, страхування, вартість заморожених коштів;
- адміністративні та інформаційні витрати (10–20%) — планування, диспетчеризація, ІТ-системи.

Враховуючи, що транспортні витрати є найбільшою статтею, скорочення пробігу транспортних засобів шляхом оптимізації маршрутів має безпосередній і значний економічний ефект. За оцінками фахівців, кожні 10% скорочення транспортних витрат дозволяють знизити загальну собівартість продукції на 1,5–

3%.

1.2. Математичні моделі логістичних задач оптимізації

Математичне моделювання — це той інструмент, що дозволяє перекласти реальну логістичну проблему на мову чітких формул і алгоритмів. Як тільки задача набуває математичної форми, до неї стають застосовними потужні методи теорії оптимізації та дослідження операцій.

1.2.1. Загальна постановка задачі комбінаторної оптимізації

Більшість логістичних задач належать до класу задач комбінаторної оптимізації. У загальному вигляді задача комбінаторної оптимізації формулюється так:

знайти $x^* \in X$ таке, що $f(x^*) = \min \{ f(x) : x \in X \}$,

де X — скінченна (але, як правило, надзвичайно велика) множина допустимих рішень; $f : X \rightarrow \mathbb{R}$ — цільова функція, що підлягає мінімізації; x^* — оптимальне рішення.

Складність задачі визначається потужністю множини $|X|$, яка для більшості логістичних задач зростає експоненційно зі збільшенням розмірності. Саме ця властивість унеможливорює точне вирішення великих задач методом повного перебору та обумовлює необхідність розробки ефективних алгоритмів.

1.2.2. Задача комівояжера (TSP)

Задача комівояжера (Travelling Salesman Problem, TSP) — одна з найвідоміших і водночас «вічнозелених» задач комбінаторної оптимізації. Вперше сформульована ще в 1930-х роках, вона й сьогодні не дає спокою дослідникам у галузі теоретичної інформатики та прикладної математики.

Формальна постановка. Дано повний орієнтований граф $G = (V, E)$, де $V = \{v_0, v_1, \dots, v_n\}$ — множина вузлів (міст), E — множина дуг, кожна з яких має вагу (відстань) $d(i, j) \geq 0$. Необхідно знайти гамільтонів цикл мінімальної загальної довжини, тобто такий маршрут, що:

– починається і закінчується в одному вузлі (депо або складі);

- відвідує кожен вузол рівно один раз;
- мінімізує суму ваг усіх пройдених дуг.

Цільова функція:

$F(\pi) = d(v_0, \pi_1) + \sum_{i=1}^{n-1} d(\pi_i, \pi_{i+1}) + d(\pi_n, v_0) \rightarrow \min$, де $\pi = (\pi_1, \dots, \pi_n)$ — перестановка вузлів $\{v_1, \dots, v_n\}$.

Розрізняють симетричну TSP (де $d(i,j) = d(j,i)$ для всіх пар) та несиметричну ATSP (Asymmetric TSP), де відстані між вузлами залежать від напрямку переміщення. ATSP краще відповідає реальним умовам міського трафіку, де існують вулиці з одностороннім рухом, різна щільність потоку в різних напрямках та інші асиметричні чинники.

TSP належить до класу NP-важких задач: жоден з відомих алгоритмів не здатен гарантовано знаходити оптимальний маршрут за поліноміальний час. Кількість різних маршрутів (гамільтонових циклів) для n вузлів дорівнює $(n-1)!/2$ для симетричної та $(n-1)!$ для несиметричної задачі. Так, для $n = 10$ маємо $10! = 3\,628\,800$ варіантів, а для $n = 20$ — вже $20! \approx 2,4 \cdot 10^{18}$, отже навіть найпотужніші сучасні комп'ютери не здатні перебрати всі варіанти за прийнятний час.

1.2.3. Задача маршрутизації транспортних засобів (VRP) [3; 4]

Задача маршрутизації транспортних засобів (Vehicle Routing Problem, VRP) є узагальненням TSP на випадок кількох транспортних засобів і є найбільш широко застосовуваною моделлю в транспортній логістиці. Вперше сформульована Дантцигом і Рамзером у 1959 році як задача доставки бензину на заправні станції. [5]

Класична постановка VRP: маємо склад (депо) та n клієнтів, кожен з яких має попит q_i . Є m ідентичних транспортних засобів вантажопідйомністю Q . Необхідно розробити маршрути (по одному для кожного транспортного засобу), що:

- обслуговують кожного клієнта рівно один раз;
- починаються і закінчуються в депо;
- не перевищують вантажопідйомність транспортних засобів ($\sum q_i \leq Q$ для

кожного маршруту);

– мінімізують загальну пройдену відстань.

Існує численна кількість варіацій VRP, що враховують додаткові реальні обмеження: часові вікна обслуговування клієнтів (VRPTW), різнорідний парк транспортних засобів (HVRP), повернення порожніх транспортних засобів (VRP with backhauls), розподілену доставку (Split Delivery VRP) та ін. Як TSP, так і базова VRP є NP-важкими задачами: навіть для 10 точок простір рішень складає $10! = 3\,628\,800$ варіантів, і повний перебір тут фізично нездійснений. Обидві задачі потребують ефективних алгоритмічних підходів.

1.3. Класифікація методів оптимізації в логістиці

Для розв'язання задач логістичної оптимізації розроблено чимало різних підходів — вони суттєво відрізняються і за принципом роботи, і за тим, які гарантії якості дають, і за областю застосування. Зручно розглядати їх у трьох великих групах.

1.3.1. Точні методи

Точні методи — це ті, що гарантують знаходження глобально оптимального рішення. До цього класу належать:

Метод повного перебору (грубої сили). Систематично переглядає весь простір допустимих рішень. Гарантує оптимальність, але має факторіальну складність $O(n!)$, що робить його непридатним для задач із $n > 12$ –15.

Метод гілок і меж (Branch and Bound). Організовує пошук у вигляді дерева рішень із відсіканням гілок, що свідомо не ведуть до кращого рішення. Є найефективнішим точним методом для TSP і VRP малого та середнього розміру (до 50–60 вузлів для практичних задач).

Метод динамічного програмування (алгоритм Гелда–Карпа). Забезпечує оптимальне рішення TSP за складністю $O(2^n \cdot n^2)$, що є суттєво кращим за факторіальний перебір, але все ж експоненційним. Придатний для $n \leq 20$ –25.

Методи цілочисельного лінійного програмування (ILP). Формалізують

задачу як систему лінійних обмежень із булевими змінними та розв'язують її за допомогою симплекс-методу та методу відтинань. Ефективні для задач зі складними обмеженнями, але обчислювально дорогі.

1.3.2. Наближені методи (евристики та метаевристики)

Наближені методи не дають гарантії глобального оптимуму — зате вміють знаходити якісні рішення за розумний час. Їх поділяють на два підкласи.

Конструктивні евристики будують рішення з нуля, покроково додаючи елементи відповідно до певного простого правила:

- метод найближчого сусіда — на кожному кроці обирає найближчий невідвіданий вузол;
- метод найдешевшого включення — додає вузол, що мінімізує приріст загальної вартості;
- метод Кларка–Райта — об'єднує окремі поїздки, максимізуючи заощадження;
- рандомізована конструкція (GRASP) — вводить елемент випадковості для уникнення локальних оптимумів.

Локальні методи покращення стартують від наявного рішення та ітеративно вдосконалюють його шляхом локальних змін:

- 2-opt та 3-opt — перебудовують маршрут, видаляючи 2 або 3 ребра та повторно з'єднуючи вузли;
- метод Or-opt — переміщує один або кілька вузлів у кращу позицію маршруту;
- метод Ліна–Кернігана (LK) — узагальнений k-opt із змінним числом заміन на кожній ітерації; дає результати, близькі до оптимальних.

Метаевристики є вищим рівнем абстракції над локальними методами. Вони управляють процесом пошуку глобально, дозволяючи виходити з локальних оптимумів:

- імітація відпалювання (Simulated Annealing, SA) — приймає гірші рішення з певною ймовірністю, що зменшується з часом;
- пошук заборон (Tabu Search, TS) — веде список "заборонених" кроків для запобігання циклюванню;

- мурашиний алгоритм (Ant Colony Optimization, ACO) — імітує поведінку мурашок та феромонний слід;
- генетичні алгоритми (GA) — імітують природний добір та еволюцію популяції розв'язків;
- алгоритм рою часток (PSO), великий розподілений пошук (LNS) та ін.

Таблиця 1.1 – Порівняльна характеристика класів методів оптимізації

Клас методів	Гарантія оптим.	Складність	Область застосування
Повний перебір	Так	$O(n!)$	$n \leq 12$
Гілок і меж	Так	Ехр.	$n \leq 50-60$
Динамічне прогр.	Так	$O(2^n \cdot n^2)$	$n \leq 25$
Конструктивні евристики	Ні	$O(n^2)$	$n = 10-1000$
Локальне покращення	Ні	$O(n^2-n^3)$	$n = 10-500$
Метаеввристики	Ні	$O(\text{iter} \cdot n^2)$	Будь-який n

1.4. Огляд сучасного стану досліджень

1.4.1. Розвиток методів оптимізації маршрутів

Формальне вивчення задачі комівояжера має довгу історію — перші кроки зроблено ще в 1930-х роках. Першу формальну математичну постановку TSP зробив Карл Менгер у 1932 році. У 1954 році Данціг, Фулкерсон і Джонсон розв'язали TSP з 49 містами США за допомогою методу лінійного програмування, що стало першим значним практичним досягненням у цій галузі.

Справжній переломний момент настав у 1970-х роках, коли Річард Карп (1972) довів NP-важкість гамільтонового циклу, а разом із ним і TSP. Це відкриття, по суті, змінило напрямок усієї галузі й підштовхнуло дослідників до активного пошуку евристичних рішень. У 1973 році Лін і Кернігана

запропонували алгоритм LK, який досі залишається одним з найефективніших методів локального покращення для TSP.

Паралельно розвивалися методи для VRP: після піонерської роботи Данцига і Рамзера (1959) та алгоритму Кларка–Райта (1964) значним кроком стала постановка задачі VRP з часовими вікнами (VRPTW) у 1980-х роках. Сьогодні VRP та її варіації є одними з найбільш активно досліджуваних задач у дослідженні операцій.

1.4.2. Еволюційні та природонавіяні алгоритми

З 1990-х років у задачах маршрутизації дедалі активніше почали застосовувати підходи, натхненні природою. Генетичні алгоритми, вперше застосовані для TSP Голландом у 1975 році та вдосконалені у роботах Гольдберга, Мюлен берга та ін., стали стандартним інструментом у логістичній оптимізації.

Мурашиний алгоритм (ACO), запропонований Марко Доріго у 1991 році як дипломна робота і розширений у 1996–2004 роках, продемонстрував виключну ефективність для TSP і VRP. Ключова ідея — імітація поведінки мурашиної колонії, де феромонний слід на коротших маршрутах посилюється з часом, що веде популяцію до оптимального рішення. [9]

На початку 2000-х років великий розподілений пошук (Large Neighborhood Search, LNS), запропонований Шоу (1998), перетворився на один з найефективніших методів для задач з обмеженнями, зокрема VRPTW. Сучасні варіанти Adaptive LNS (ALNS, Ропке та Піцінгер, 2006) адаптивно обирають оператори руйнування та відновлення рішення.

1.4.3. Сучасні тенденції та застосування штучного інтелекту

Останнє десятиліття ознаменувалося активним впровадженням методів машинного навчання та глибокого навчання у задачі комбінаторної оптимізації. Зокрема, pointer networks (Вінялс, Фортунато, Джайтлі, 2015) та навчання з підкріпленням (reinforcement learning) демонструють перспективні результати для TSP і VRP, навчаючись будувати маршрути безпосередньо з даних без явного

програмування евристик. [10]

Паралельно розвиваються гібридні методи, що поєднують евристики, метаевристики та точні підходи. Так, методи column generation у поєднанні зі LNS дозволяють розв'язувати практичні задачі VRPTW з сотнями клієнтів до квазі-оптимального рівня. Хмарні обчислення та паралельна реалізація алгоритмів суттєво розширюють масштаб розв'язуваних задач.

У практичному застосуванні сучасні логістичні платформи (Amazon Logistics, DHL, Ukrposhta та ін.) використовують складні системи підтримки прийняття рішень на основі оптимізаційних алгоритмів. На практиці такі системи дозволяють скоротити транспортні витрати на 15–30% і суттєво підвищити відсоток своєчасних доставок.

1.5. Постановка задачі дослідження

Аналіз науково-практичної літератури та стану задачі дозволяє сформулювати конкретну задачу даного дослідження.

1.5.1. Контекст та обґрунтування вибору методів

Для практичного дослідження обрано задачу TSP як базову модель маршрутизації. Незважаючи на удавану простоту постановки, TSP охоплює всі ключові складнощі задач маршрутизації і є стандартним бенчмарком для оцінки алгоритмів оптимізації. Несиметрична версія задачі (ATSP) обрана навмисно, оскільки вона точніше відображає реальні умови міського трафіку.

Для порівняльного аналізу обрано три репрезентативних методи, що належать до різних класів:

- 1) Метод найближчого сусіда — як представник простих жадібних конструктивних евристик з лінійною складністю відносно числа операцій вибору та загальною складністю $O(n^2)$. Демонструє базовий рівень якості, досяжний без будь-якої складної оптимізаційної логіки.
- 2) Метод заощаджень Кларка–Райта — як представник структурних конструктивних евристик, що враховують взаємний вплив усіх пар точок.

Практично значимий алгоритм, що широко застосовується у комерційних логістичних системах.

3) Генетичний алгоритм — як представник стохастичних метаввристик, що здатні глобально досліджувати простір рішень та знаходити рішення, недоступні детермінованим методам.

1.5.2. Критерії порівняльного аналізу

Порівняння методів проводитиметься за наступними критеріями:

- якість рішення — довжина знайденого маршруту та відхилення від найкращого знайденого рішення;
- відносна ефективність — відсоток скорочення пробігу порівняно з нераціональним базовим плануванням;
- обчислювальна складність — теоретична оцінка $O(\cdot)$ залежно від розмірності задачі;
- час виконання — практичний час розрахунку на задачі з 11 вузлами;
- детермінованість — здатність відтворювати однаковий результат при повторних запусках;
- масштабованість — рекомендована область застосування за розміром задачі.

Практична апробація виконується на реальній транспортній мережі міста Києва, що складається з 11 вузлів (центральный склад та 10 точок доставки) з несиметричною матрицею відстаней розміром 11×11 , де значення $d(i,j)$ отримані з реальних дорожніх відстаней з урахуванням правил дорожнього руху.

1.6. Висновки до розділу 1

У першому розділі розглянуто теоретичні основи, необхідні для дослідження методів оптимізації логістичних процесів у складних системах. Встановлено, що логістика є міждисциплінарною науковою дисципліною, що вивчає матеріальні та інформаційні потоки у складних розподільчих мережах, а транспортні витрати становлять найбільшу частку (40–60%) у структурі загальних логістичних витрат.

Ключовою математичною моделлю для задач маршрутизації є задача комівояжера (TSP) та її узагальнення — задача маршрутизації транспортних засобів (VRP). Обидві задачі є NP-важкими, що принципово обмежує застосовність точних методів і обумовлює необхідність використання евристичних та метаевристичних підходів для задач практичного розміру.

Проведено систематизацію методів оптимізації за трьома класами: точні методи (повний перебір, гілок і меж, динамічне програмування), конструктивні та поліпшувальні евристики (метод найближчого сусіда, метод заощаджень Кларка–Райта, 2-opt, 3-opt), та метаевристики (генетичні алгоритми, мурашиний алгоритм, імітація відпалювання). Кожен клас має власний профіль придатності: точні методи — для малих задач ($n \leq 50$), евристики — для задач середнього розміру в умовах обмеженого часу, метаевристики — для складних задач великої розмірності з вимогами до якості рішення.

Огляд сучасного стану досліджень підтвердив актуальність обраної теми: від піонерських робіт 1950–70-х років до сучасного застосування методів глибокого навчання, дослідження задач маршрутизації залишається одним з центральних напрямків обчислювальної математики та дослідження операцій. У наступних розділах на основі закладеної теоретичної бази виконується практична реалізація та порівняльний аналіз трьох обраних методів оптимізації.

РОЗДІЛ 2

АЛГОРИТМИ ОПТИМІЗАЦІЇ МАРШРУТІВ: ТЕОРЕТИЧНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ

2.1. Жадібні конструктивні евристики

Жадібні алгоритми — мабуть, найбільш інтуїтивний підхід до побудови маршрутів. Ідея проста: на кожному кроці робимо найкращий хід з тих, що доступні прямо зараз, не думаючи про те, що буде далі. Такі алгоритми не переглядають прийнятих рішень і будують рішення один раз — за один прохід від початку до кінця.

Жадібні евристики привабливі з практичної точки зору завдяки низькій обчислювальній складності, простоті реалізації та прозорості логіки роботи. Проте їхня Але є й суттєвий недолік: жадібні алгоритми схильні потрапляти в локальні пастки — рішення, що виглядають вдало на початку, можуть суттєво погіршити підсумковий результат.

2.1.1. Метод найближчого сусіда: детальний аналіз

Метод найближчого сусіда (Nearest Neighbour Heuristic, NN) — мабуть, найбільш відомий жадібний алгоритм для задачі комівояжера. Його логіку легко зрозуміти інтуїтивно: уявіть мандрівника, який щоразу їде до найближчого пункту, що ще не відвіданий.

2.1.2. Формальний опис алгоритму NN

Вхідні дані: матриця відстаней $D = \{d(i,j)\}$, $i,j \in \{0,1,\dots,n\}$, де вузол 0 — початковий (депо).

Вихідні дані: маршрут $\pi = (0, \pi_1, \pi_2, \dots, \pi_n, 0)$ та його загальна довжина $F(\pi)$.

Алгоритм:

1. Ініціалізація: $visited = \{0\}$; $route = [0]$; $current = 0$.
2. Поки $|visited| < n+1$, виконувати:
 3. $next = \operatorname{argmin}\{d(current, j) : j \notin visited\}$
 4. $route.append(next)$; $visited.add(next)$; $current = next$.
5. $route.append(0)$. Обчислити $F(\pi) = \sum d(route[k], route[k+1])$.

Вибір наступного вузла на кроці t формалізується як:

$$\text{next}(t) = \operatorname{argmin} \{ d(v_t, j) : j \notin V_t \},$$

де V_t — множина вже відвіданих вузлів на кроці t , v_t — поточний вузол.

2.1.3. Аналіз складності та властивостей алгоритму NN

Обчислювальна складність алгоритму NN становить $O(n^2)$: на кожному з n кроків виконується пошук мінімуму серед не більше ніж n невідвіданих вузлів. Для задачі з $n = 10$ точок доставки це означає не більше $10+9+8+\dots+1 = 55$ операцій порівняння — надзвичайно швидко навіть для найпростішого обладнання.

Просторова складність: $O(n)$ для зберігання масиву відвіданих вузлів та поточного маршруту.

Теоретична оцінка якості рішення: для евклідової метрики алгоритм NN гарантовано дає рішення не гірше ніж у $(\lceil \log_2 n \rceil + 0,5)$ разів від оптимального. На практиці для задач середнього розміру алгоритм дає рішення на 20–25% гірші за оптимум при симетричній матриці та до 30–35% — при несиметричній.

Таблиця 2.1 – Характеристики алгоритму методу найближчого сусіда

Характеристика	Значення
Часова складність	$O(n^2)$
Просторова складність	$O(n)$
Тип	Детермінований, конструктивний
Гарантія оптимальності	Відсутня
Середнє відхилення від оптимуму	20–30%
Рекомендований розмір задачі	$n = 10\text{--}10\,000$
Чутливість до стартового вузла	Висока (результат залежить від вибору початкової точки)

2.1.4. Покращення алгоритму: множинні старти

Одним з простих, але ефективних способів підвищення якості методу найближчого сусіда є стратегія множинних стартів (Multi-Start NN): алгоритм запускається по одному разу з кожного з n вузлів як стартового, а серед отриманих n маршрутів обирається найкоротший.

Складність Multi-Start NN становить $O(n^3)$, що залишається поліноміальним і прийнятним для задач до $n \approx 500\text{--}1000$. Покращення якості порівняно з одиночним запуском, як правило, складає 5–15%.

2.2. Метод заощаджень Кларка–Райта

Метод заощаджень (Clarke-Wright Savings Algorithm, CWS) — один із найбільш «практичних» алгоритмів маршрутизації, розроблений у 1964 році з конкретними прикладними задачами в голові. Запропонований Джефрі Кларком та Джоном Райтом у 1964 році, він і сьогодні є стандартним інструментом у багатьох комерційних TMS-системах. [6]

2.2.1. Концепція заощаджень

Ідея методу ґрунтується на понятті заощадження (savings) — вигоди від об'єднання двох окремих рейсів в один комбінований маршрут. Розглянемо базовий сценарій: є депо O та дві точки доставки i та j . Якщо здійснювати окремі поїздки до кожної точки, загальна відстань становить:

$$V_{ij} = d(O,i) + d(i,O) + d(O,j) + d(j,O) = 2 \cdot d(O,i) + 2 \cdot d(O,j)$$

Якщо ж об'єднати обидві точки в один маршрут $O \rightarrow i \rightarrow j \rightarrow O$, відстань становить:

$$C_{ij} = d(O,i) + d(i,j) + d(j,O)$$

Заощадження від об'єднання визначається як різниця між базовою та об'єднаною відстанями:

$$s(i,j) = V_{ij} - C_{ij} = d(O,i) + d(O,j) - d(i,j)$$

Фізичний зміст формули зрозумілий: $s(i,j)$ показує, скільки кілометрів ми виграємо, замінивши два окремі рейси одним спільним маршрутом. Чим більше заощадження — тим вигідніше об'єднання. Якщо $s(i,j)$ менше нуля, краще залишити точки у окремих маршрутах.

2.2.2. Формальний опис алгоритму CWS

Вхідні дані: матриця відстаней $D = \{d(i,j)\}$; депо 0; множина клієнтів $\{1, 2, \dots, n\}$.

Алгоритм:

6. Ініціалізація: сформувати n окремих маршрутів $\{0 \rightarrow 1 \rightarrow 0, 0 \rightarrow 2 \rightarrow 0, \dots, 0 \rightarrow n \rightarrow 0\}$. Загальна базова відстань $B_0 = \sum_i (d(0,i) + d(i,0))$.
7. Обчислення заощаджень: для кожної пари (i,j) , де $i < j$ та $i,j \in \{1, \dots, n\}$, обчислити $s(i,j) = d(0,i) + d(0,j) - d(i,j)$. Отримати список S із $n(n-1)/2$ значень.
8. Сорткування: впорядкувати список S за спаданням значень $s(i,j)$.
9. Злиття маршрутів: послідовно переглядати відсортований список. Для кожної пари (i,j) перевірити умову злиття: вузол i є останнім у своєму маршруті (перед поверненням до 0) та j є першим у своєму маршруті (після виходу з 0), i при цьому i та j належать до різних маршрутів. Якщо умова виконана — об'єднати маршрути, видаливши повернення до 0 з першого та виїзд із 0 до другого.
10. Повторювати крок 4 до вичерпання списку або формування єдиного маршруту.

Складність алгоритму CWS: обчислення заощаджень — $O(n^2)$; сорткування — $O(n^2 \log n)$; злиття — $O(n^2)$. Загальна складність: $O(n^2 \log n)$.

2.2.3. Властивості та обмеження методу CWS

Метод заощаджень має ряд важливих властивостей:

- детермінованість — при однакових вхідних даних алгоритм завжди дає однаковий результат, що спрощує верифікацію та відтворення розрахунків;
- природне розширення на VRP — якщо транспортні засоби мають обмежену вантажопідйомність Q , умова злиття доповнюється перевіркою: сумарний попит об'єднаного маршруту не перевищує Q ;
- незалежність від початкового маршруту — на відміну від NN, результат не залежить від порядку обходу вузлів на початку;
- субоптимальність при несиметричній матриці — формула заощаджень

розроблялась для симетричного випадку; при ATSP значення $s(i,j) \neq s(j,i)$, що ускладнює коректну оцінку вигоди від злиття.

Таблиця 2.2 – Порівняння методу NN та методу заощаджень CWS

Критерій	Метод NN	Метод CWS
Принцип побудови	Покроковий вибір найближчого вузла	Злиття маршрутів за максимальним заощадженням
Складність	$O(n^2)$	$O(n^2 \log n)$
Детермінованість	Так (за фіксованого старту)	Так
Залежність від стартового вузла	Висока	Відсутня
Розширення на VRP	Складне	Природне
Якість рішення (TSP)	~20–25% від оптимуму	~15–25% від оптимуму
Ефективність при кластерах	Добра	Дуже добра

2.3. Генетичні алгоритми

Генетичні алгоритми — це напрямок штучного інтелекту, що буквально черпає натхнення з природи: з принципів природного відбору, схрещування і мутації. Теоретичну основу ГА заклав Джон Голланд у своїй фундаментальній монографії «Adaptation in Natural and Artificial Systems» (1975), а практичне застосування для оптимізаційних задач широко розвинув Девід Голдберг у 1989 році. [7; 8]

2.3.1. Біологічна аналогія та базові концепції

Центральна ідея проста: весь простір допустимих рішень трактується як «середовище», а кожне конкретне рішення — як «організм» з певним рівнем пристосованості. Алгоритм підтримує цілу «популяцію» таких рішень і покоління за поколінням вдосконалює її, доки не отримає задовільний результат.

Відповідність між біологічними та алгоритмічними концепціями:

Таблиця 2.3 – Відповідність біологічних понять та елементів генетичного алгоритму

Біологічне поняття	Аналог у ГА	Для задачі TSP
Особина	Хромосома	Один можливий маршрут доставки
Ген	Елемент хромосоми	Окрема точка доставки
Популяція	Набір рішень	Сукупність можливих маршрутів
Пристосованість	Значення цільової функції	Загальна довжина маршруту
Схрещування	Комбінування батьківських рішень	Формування нового маршруту з частин двох маршрутів
Мутація	Випадкова зміна рішення	Перестановка або інверсія точок у маршруті

2.3.2. Кодування рішень для задачі TSP

Вибір схеми кодування є критично важливим для ефективності ГА, оскільки від нього залежать властивості операторів схрещування та мутації. Для задачі TSP розроблено кілька підходів до кодування.

Перестановкове кодування (найпоширеніше). Хромосома являє собою впорядковану перестановку вузлів:

$\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$, де $\pi_i \in \{1, \dots, n\}$, $\pi_i \neq \pi_j$ при $i \neq j$.

Маршрут читається як: $0 \rightarrow \pi_1 \rightarrow \pi_2 \rightarrow \dots \rightarrow \pi_n \rightarrow 0$. Для $n = 10$: хромосома (3, 7, 1, 9, 5, 2, 8, 4, 6, 10) кодує маршрут $O \rightarrow C \rightarrow G \rightarrow A \rightarrow I \rightarrow E \rightarrow B \rightarrow H \rightarrow D \rightarrow F \rightarrow J \rightarrow O$.

Перестановкове кодування є найбільш природним для TSP, оскільки гарантує коректність хромосоми (кожен вузол відвідується рівно один раз) та не вимагає додаткових механізмів виправлення помилок кодування.

2.3.3. Оператори генетичного алгоритму

Оператор відбору (Selection).

Відбір батьківських особин для схрещування є ключовим механізмом "природного добору" в ГА. Метод повинен надавати перевагу більш

пристосованим особинам, але зберігати різноманіття популяції для запобігання передчасній збіжності.

Турнірний відбір (Tournament Selection): з поточної популяції випадково обирається k особин ("турнірна група"); серед них визначається особина з найкращим значенням цільової функції ("переможець турніру"); ця особина стає одним із батьків. Розмір турніру k регулює "селективний тиск": при $k = 2$ — слабкий відбір, при $k = n$ — детермінований вибір найкращого. Для задач TSP зазвичай $k = 3-7$.

Оператор схрещування (Crossover).

Схрещування комбінує генетичний матеріал двох батьків для отримання дочірніх особин, що успадковують корисні ознаки обох батьків. Для перестановкового кодування стандартні оператори (одноточковий, двохточковий) не придатні, оскільки можуть породжувати некоректні перестановки (з повтореннями вузлів). Для TSP розроблені спеціальні оператори:

ОХ (Order Crossover) — оператор порядкового схрещування:

11. Обрати випадковий сегмент $[l, r]$ у батьківській хромосомі P_1 .
12. Скопіювати сегмент $P_1[l..r]$ у відповідні позиції дочірньої хромосоми D .
13. Заповнити решту позицій D генами з P_2 у порядку їх появи, починаючи з позиції $r+1$ (циклічно), пропускаючи вже присутні у D гени.

Приклад ОХ для $n = 8$ (сегмент позиції 3–5):

$P_1 = (1 \ 4 \mid 2 \ 7 \ 5 \mid 3 \ 8 \ 6)$

$P_2 = (3 \ 8 \mid 6 \ 2 \ 1 \mid 7 \ 4 \ 5)$

$D = (8 \ 6 \mid 2 \ 7 \ 5 \mid 1 \ 4 \ 3)$

ОХ зберігає відносний порядок генів від P_2 поза скопійованим сегментом, що є важливою властивістю для TSP (відносний порядок вузлів часто несе корисну інформацію про структуру маршруту).

Оператор мутації (Mutation).

Мутація вносить малі випадкові зміни у хромосому для забезпечення

різноманіття популяції та запобігання передчасній збіжності до локального оптимуму. Для TSP:

Swap mutation (перестановка двох генів): з імовірністю p_m обрати дві випадкові позиції i та j ; поміняти місцями гени $\pi[i]$ та $\pi[j]$. Складність: $O(1)$. Зміна маршруту: два вузли міняються місцями.

Inversion mutation (інверсія підрядка): обрати сегмент $[l, r]$ та перевернути порядок генів у ньому. Еквівалентна операції 2-opt на маршруті.

2.3.4. Загальна схема генетичного алгоритму для TSP

Повна процедура ГА для задачі TSP:

14. Ініціалізація: згенерувати початкову популяцію P із P особин — випадкових перестановок вузлів $\{1, \dots, n\}$.
15. Оцінка: обчислити значення цільової функції $f(\pi_i) = F(\pi_i)$ (довжину маршруту) для кожної особини.
16. Перевірка критерію зупинки: якщо досягнуто максимальну кількість поколінь G або виконано інший критерій — зупинитись, повернути найкращу знайдену особину.
17. Відбір: для кожної з P пар батьків застосувати турнірний відбір.
18. Схрещування: застосувати ОХ до кожної пари батьків; отримати P нащадків.
19. Мутація: до кожного нащадка з імовірністю p_m застосувати swap mutation.
20. Формування нового покоління: об'єднати батьківську популяцію та нащадків; обрати P найкращих (елітна стратегія) або повністю замінити батьківське покоління нащадками.
21. Перейти до кроку 2.

2.3.5. Аналіз складності та налаштування параметрів

Обчислювальна складність одного покоління ГА: $O(P \cdot n)$ — оцінка пристосованості кожної особини; $O(P \cdot k)$ — турнірний відбір; $O(P \cdot n)$ — схрещування та мутація. Загальна складність G поколінь: $O(G \cdot P \cdot n)$.

Вибір параметрів ГА є важливим практичним питанням, оскільки вони суттєво впливають на якість і швидкість збіжності алгоритму:

Таблиця 2.4 – Рекомендовані параметри генетичного алгоритму для задачі TSP

Параметр	Рекомендоване значення	Наслідки відхилення
Розмір популяції P	50–200 особин	Малий $P \rightarrow$ передчасна збіжність; великий $P \rightarrow$ уповільнення
Кількість поколінь G	100–500	Мало поколінь $\rightarrow$ недостатній пошук; багато $\rightarrow$ зайві витрати
Ймовірність мутації p_m	0.01–0.20	$p_m < 0.01 \rightarrow$ стагнація; $p_m > 0.3 \rightarrow$ деградація до випадкового пошуку
Розмір турніру k	3–7	$k = 2 \rightarrow$ слабкий відбір; $k \geq n/2 \rightarrow$ втрата різноманіття
Оператор схрещування	OX або PMX	Некоректні оператори породжують повтори вузлів
Частка еліти	5–10%	Без еліти $\rightarrow$ втрата найкращого рішення між поколіннями

2.4. Теоретичне порівняння алгоритмів

2.4.1. Аналіз простору пошуку

Три розглянутих методи принципово відрізняються одне від одного тим, яку частину простору можливих маршрутів вони реально переглядають.

Метод NN будує рівно одне рішення за один прохід (або n рішень при множинних стартах). Він є детермінованим і повністю визначеним стартовим вузлом. Простір пошуку, фактично досліджений NN, є нескінченно малою часткою від загального $|\Omega| = (n-1)!$.

Метод CWS також будує одне рішення, але через механізм обчислення та ранжування заощаджень він неявно враховує глобальну структуру задачі (відносні відстані між усіма парами точок), що забезпечує, як правило, кращу якість рішення, ніж NN.

Генетичний алгоритм веде популяцію з P особин протягом G поколінь. Завдяки схрещуванню хромосом він здатен досліджувати зовсім різні куточки простору маршрутів — у тому числі ті, до яких жодна з детермінованих евристик ніколи б не дістала. Загальна кількість оцінених рішень: $P \cdot G$.

2.4.2. Збіжність та якість рішень

Детерміновані конструктивні евристики (NN, CWS) мають детерміновану збіжність: алгоритм завжди завершується за фіксовану кількість кроків, результат відтворюваний. Проте якість рішення жорстко обмежена: алгоритм не може виправити помилки, зроблені на ранніх кроках.

Генетичний алгоритм має стохастичну збіжність: результат може варіюватись між запусками. Однак ГА здатний виходити з локальних оптимумів завдяки рекомбінації та мутації. Теорема схем (Schemata Theorem) Голланда пояснює механізм збіжності: короткі, часто зустрічаючі схеми (шаблони хромосом) з вищою пристосованістю отримують експоненційно більше ресурсів у наступних поколіннях.

2.4.3. Зведена порівняльна таблиця

Таблиця 2.5 – Теоретичне порівняння алгоритмів оптимізації маршрутів

Критерій	Метод NN	Метод CWS	Генетичний алг.
Клас алгоритму	Жадібна евристика	Ощадна евристика	Еволюційна метаевристика
Часова складність	$O(n^2)$	$O(n^2 \log n)$	$O(G \cdot P \cdot n)$
Просторова складність	$O(n)$	$O(n^2)$	$O(P \cdot n)$
Детермінованість	Так	Так	Ні (стохастичний)
Гарантія оптимальності	Відсутня	Відсутня	Відсутня
Якість рішення	Задовільна (–20–30%)	Добра (–15–25%)	Висока (–5–15%)
Виходи з локальних мін.	Відсутня	Відсутня	Є (мутація/рекомбін.)

Придатність для VRP	Обмежена	Природна	Гнучка (з обмеженнями)
Складність реалізації	Низька	Середня	Висока
Рекомендований розмір n	10–10000	10–200	10–500

2.5. Несиметрична задача комівояжера (ATSP) та її особливості

Більшість теоретичних результатів для TSP отримані для симетричного випадку, де $d(i,j) = d(j,i)$. Однак реальні транспортні мережі, як правило, несиметричні. Несиметрична задача комівояжера (Asymmetric TSP, ATSP) формально відрізняється тим, що матриця відстаней D є несиметричною: $d(i,j) \neq d(j,i)$ для принаймні деяких пар (i,j) .

Причини несиметричності у реальних транспортних мережах:

- вулиці з одностороннім рухом, що змушують прокладати маршрути в обхід;
- різна щільність трафіку в протилежних напрямках у різні часи доби;
- особливості рельєфу та топографії — підйоми і спуски впливають на час та витрати пального;
- технологічні особливості окремих розв'язків, де дозволений в'їзд лише з певних напрямків.

Для транспортної мережі Києва несиметричність матриці є значною: наприклад, відстань від складу О (Хрещатик) до точки А (Голосіївський просп.) становить 8,8 км, тоді як зворотній маршрут $A \rightarrow O = 10,7$ км, що дає різницю у 21,6%. Такі відхилення суттєво впливають на оптимальну структуру маршруту.

Адаптація розглянутих алгоритмів до ATSP:

- метод NN — природно працює з несиметричними матрицями, оскільки на кожному кроці використовує конкретне значення $d(\text{current}, j)$, а не $d(j, \text{current})$;
- метод CWS — формула заощаджень $s(i,j) = d(O,i) + d(O,j) - d(i,j)$ дає різні результати для пари (i,j) та (j,i) . При злитті необхідно явно враховувати напрямок з'єднання: вузол i — кінець першого маршруту, j — початок другого;

— генетичний алгоритм — не вимагає модифікацій, оскільки оцінка пристосованості безпосередньо обчислює загальну довжину орієнтованого маршруту.

2.6. Метрики оцінювання ефективності алгоритмів

Для коректного порівняльного аналізу методів необхідно визначити стандартні метрики оцінювання. У даному дослідженні використовуються такі показники.

Довжина маршруту $F(\pi)$, км: основна метрика якості — загальний пробіг транспортного засобу за весь маршрут. Обчислюється за формулою цільової функції TSP. Менше значення — кращий результат.

Відхилення від базової відстані Δ_0 , %:

$$\Delta_0 = (B_0 - F(\pi)) / B_0 \times 100\%,$$

де B_0 — базова відстань (сума окремих поїздок до кожної точки і назад). Показує ефект оптимізації відносно найгіршого можливого планування.

Відхилення від найкращого рішення Δ^* , %:

$$\Delta^* = (F(\pi) - F^*) / F^* \times 100\%,$$

де F^* — найкраще знайдене рішення серед усіх методів. Показує, наскільки метод поступається найкращому конкуренту.

Обчислювальна складність $O(\cdot)$: теоретична оцінка кількості операцій як функція розмірності задачі n , що характеризує масштабованість алгоритму.

Практичний час виконання t , мс: реальний час розрахунку маршруту для конкретної задачі. Важливий для застосувань із жорсткими часовими обмеженнями (оперативна диспетчеризація).

2.7. Висновки до розділу 2

У другому розділі виконано детальний теоретичний аналіз трьох алгоритмів оптимізації маршрутів, обраних для порівняльного дослідження. Встановлено принципові відмінності між розглянутими підходами.

Метод найближчого сусіда відзначається мінімальною складністю $O(n^2)$ та

граничною простотою реалізації. Жадібна стратегія вибору локально найближчого вузла забезпечує прийнятні результати для більшості конфігурацій задачі, але систематично погіршує якість маршруту на завершальних кроках — так звана «проблема хвоста» жадібного алгоритму.

Метод заощаджень Кларка–Райта являє собою більш структурований підхід: через концепцію заощадження $s(i,j) = d(O,i) + d(O,j) - d(i,j)$ алгоритм неявно враховує глобальну структуру задачі. Це дозволяє ефективно виявляти та об'єднувати географічно близькі кластери точок. Природне розширення на задачу VRP робить CWS особливо практично значимим.

Генетичний алгоритм принципово відрізняється від двох попередніх методів стохастичним характером та здатністю досліджувати широкий простір рішень через механізми схрещування (оператор ОХ) та мутації. Ціною вищих обчислювальних витрат $O(G \cdot P \cdot n)$ ГА здатний знаходити рішення, принципово недоступні для детермінованих конструктивних евристик.

Проведений теоретичний аналіз підтвердив обґрунтованість вибору саме цих трьох методів для порівняльного дослідження: вони репрезентують три різних рівні складності та підходи до пошуку рішення — від простого жадібного конструювання до глобального стохастичного пошуку. У наступному розділі ці методи апробуються на реальній транспортній задачі та надається кількісна оцінка їхньої практичної ефективності.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ МАРШРУТІВ

3.1. Постановка задачі та опис вхідних даних

Для того щоб порівняти алгоритми не на абстрактних даних, а в умовах, наближених до реальних, була сформульована конкретна задача маршрутизації — розвезення вантажів по десяти точках міста Києва. За типом це — несиметрична задача комівояжера (ATSP), де відстань «туди» і «назад» між більшістю точок відрізняється.

Причина несиметричності — суто практична: одностороннє сполучення, нерівна пропускна здатність вулиць у різних напрямках, київські пагорби та особливості дорожньої мережі. Зокрема, відстань від точки О (Хрещатик) до точки А (Голосіївський проспект) становить 8,8 км, тоді як зворотній маршрут $A \rightarrow O = 10,7$ км — різниця 21,6%, що унеможливорює застосування алгоритмів, розрахованих на симетричні матриці.

3.1.1. Опис мережі точок доставки

Транспортна мережа складається з 11 вузлів: центрального складу (мітка О) та 10 точок доставки (мітки А–J), розташованих у різних адміністративних районах міста Києва. Склад розташований на вул. Хрещатик, 22 у Шевченківському районі та є початковою і кінцевою точкою маршруту. Автомобіль має відвідати кожну точку рівно один раз і повернутися на базу.

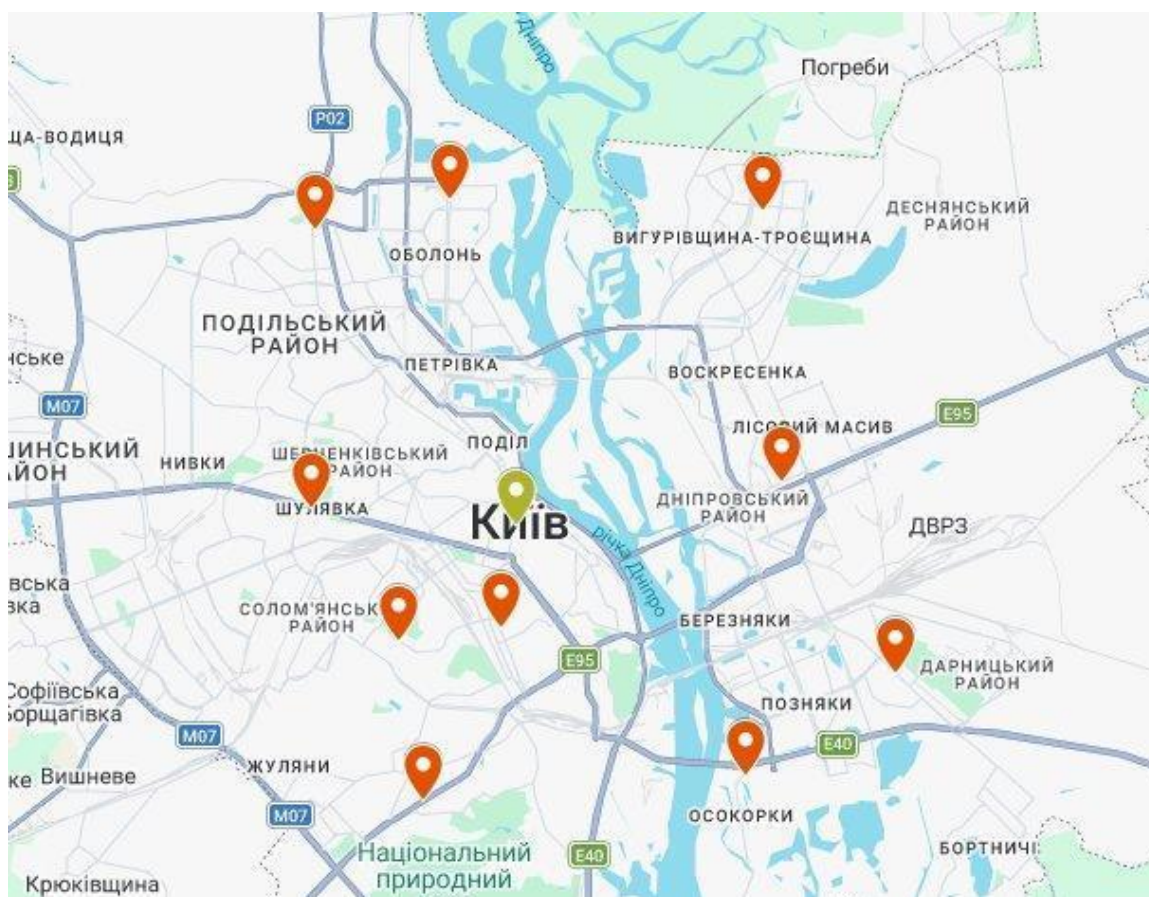


Рис. 3.1 – Карта транспортної мережі міста Києва: центральний склад (О, жовта мітка) та 10 точок доставки (А–І, оранжеві мітки)

Таблиця 3.1 – Перелік вузлів транспортної мережі

Мітка	Адреса	Район	Тип вузла
О	вул. Хрещатик, 22	Шевченківський	Центральний склад
А	просп. Голосіївський, 114	Голосіївський	Точка доставки
В	просп. Берестейський, 45	Солом'янський	Точка доставки
С	Харківське шосе, 164	Дарницький	Точка доставки
Д	вул. Данькевича Костянтина, 14	Деснянський	Точка доставки
Е	просп. Оболонський, 51В	Оболонський	Точка доставки
Ф	вул. Велика Васильківська, 102	Голосіївський	Точка доставки
Г	вул. Солом'янська, 22	Солом'янський	Точка доставки
Н	вул. Андрія Малишка, 15/1	Дніпровський	Точка доставки
І	вул. Центральна, 12	Дарницький	Точка доставки

J	вул. Вишгородська, 52	Оболонський	Точка доставки
---	-----------------------	-------------	----------------

Матриця відстаней $D = \{d(i,j)\}$ розміром 11×11 є несиметричною: $d(i,j) \neq d(j,i)$ для більшості пар вузлів. Значення $d(i,j)$ відповідає відстані у кілометрах від вузла i до вузла j , виміряній по реальних дорогах з урахуванням діючих правил дорожнього руху.

Таблиця 3.2 — Повна матриця відстаней (км), рядок $i \rightarrow$ стовпець j

↓ / →	О	А	В	С	Д	Е	Ф	Г	Н	І	Ј
О	-	8,8	6,9	14,7	15	11,9	3,4	6,1	10,8	14,8	12,5
А	10,7	-	10,6	16	22,4	22,8	6	6,7	14,6	16,1	20,5
В	6,5	10,9	-	19	18,2	13,6	6,6	5,8	15,3	19,1	11,6
С	13,8	15,9	21,6	-	15,2	23,6	13,8	17,8	7,1	7,9	23,3
Д	13,6	22,4	19,3	15,1	-	13,5	17	19,7	8,8	19,8	15,6
Е	11,5	21,7	13,3	23	13,8	-	14	16,2	17,7	23,4	4,3
Ф	4,5	6,9	7,9	13,5	16,3	14	-	4,4	11,3	14,2	13,8
Г	7	5,4	6	17,3	19,1	17,8	3,5	-	15,9	17,4	13
Н	9,7	14,5	18,1	7,6	8,9	16,7	12,1	16,3	-	10,6	17,6
І	13,4	11,9	18	6,9	18,5	23,1	11,3	14,1	11,9	-	23,3
Ј	11,4	19,4	12,5	23,8	16,4	5,4	16,9	15,3	18,4	26,2	-

Формальна постановка задачі. Знайти таку перестановку $\pi = (\pi_1, \pi_2, \dots, \pi_{10})$ вузлів $\{А, В, \dots, Ј\}$, що мінімізує цільову функцію:

$$F(\pi) = d(O, \pi_1) + \sum_{i=1}^9 d(\pi_i, \pi_{i+1}) + d(\pi_{10}, O) \rightarrow \min$$

де $d(i,j)$ — відстань від вузла i до вузла j згідно з матрицею відстаней. Загалом простір розв'язків містить $10! = 3\,628\,800$ можливих маршрутів, що робить перебір усіх варіантів практично неможливим без застосування спеціальних методів.

Базова відстань (без оптимізації). Якщо здійснювати окремий виїзд до кожної точки і повертатись на склад, сумарна відстань становитиме: $V_0 = \sum (d(O,i)+d(i,O)) = 207.0$ км. Завдання оптимізації — суттєво скоротити цей показник шляхом об'єднання поїздок в єдиний раціональний маршрут.

3.2. Метод найближчого сусіда

3.2.1. Теоретичне обґрунтування

Метод найближчого сусіда (англ. Nearest Neighbour Heuristic) — один із найпростіших і найбільш інтуїтивних підходів до задачі комівояжера. Він будує маршрут крок за кроком, щоразу обираючи найближчу точку, яку ще не відвідано.

Обчислювальна складність алгоритму становить $O(n^2)$, де n — кількість вузлів. Для несиметричних матриць, як у нашому випадку, алгоритм зазвичай дає результат приблизно на 20–25% гірший за оптимум — проте він вкрай швидкий і цілком придатний як відправна точка.

3.2.2. Опис алгоритму

Крок 1. Ініціалізація: встановити поточний вузол як склад O , позначити O як відвіданий, маршрут $:= \{O\}$.

Крок 2. Пошук: серед усіх невідведаних вузлів j знайти такий, що мінімізує $d(\text{поточний}, j)$.

Крок 3. Перехід: додати знайдений вузол до маршруту, позначити як відвіданий, встановити його як поточний.

Крок 4. Повторення кроків 2–3 до відвідання всіх вузлів.

Крок 5. Завершення: додати повернення на склад O . Обчислити загальну довжину маршруту.

Вибір вузла на кожному кроці визначається умовою:

$$j^* = \operatorname{argmin} \{ d(v_k, j) : j \notin \text{visited} \}, \text{ де } v_k — \text{поточний вузол}$$

3.2.3. Результати виконання

Таблиця 3.3 – Покрокове виконання методу найближчого сусіда

№ кроку	Поточний вузол	Дія	Обраний вузол	Відстань, км
1	О	Вибір min серед невідвіданих	F	3.4
2	F	Вибір min серед невідвіданих	G	4.4
3	G	Вибір min серед невідвіданих	A	5.4
4	A	Вибір min серед невідвіданих	B	10.6
5	B	Вибір min серед невідвіданих	J	11.6
6	J	Вибір min серед невідвіданих	E	5.4
7	E	Вибір min серед невідвіданих	D	13.8
8	D	Вибір min серед невідвіданих	H	8.8
9	H	Вибір min серед невідвіданих	C	7.6
10	C	Вибір min серед невідвіданих	I	7.9
11	I	Повернення на склад	О	13.4

Отриманий маршрут: О → F → G → A → B → J → E → D → H → C → I → О

Таблиця 3.4 – Розбивка маршруту за відрізками (метод найближчого сусіда)

Відрізок маршруту	Відстань, км
О → F	3.4
F → G	4.4
G → H	5.4

H → I	10.6
I → A	11.6
A → B	5.4
B → J	13.8
J → D	8.8
D → C	7.6
C → E	7.9
E → O	13.4
Разом:	92.3

Аналіз результату. Загальна довжина маршруту за методом найближчого сусіда становить 92.3 км, що на 55.4% менше за базову відстань (207.0 км). Алгоритм успішно використав близькість точок F та G до складу O, формуючи ефективний "ближній кластер" на початку маршруту. Головний недолік — на фінальних кроках алгоритм "збирає залишки": точки, що не потрапили в ранні кластери, часто виявляються далеко одна від одної. Це типова плата за жадібну стратегію.

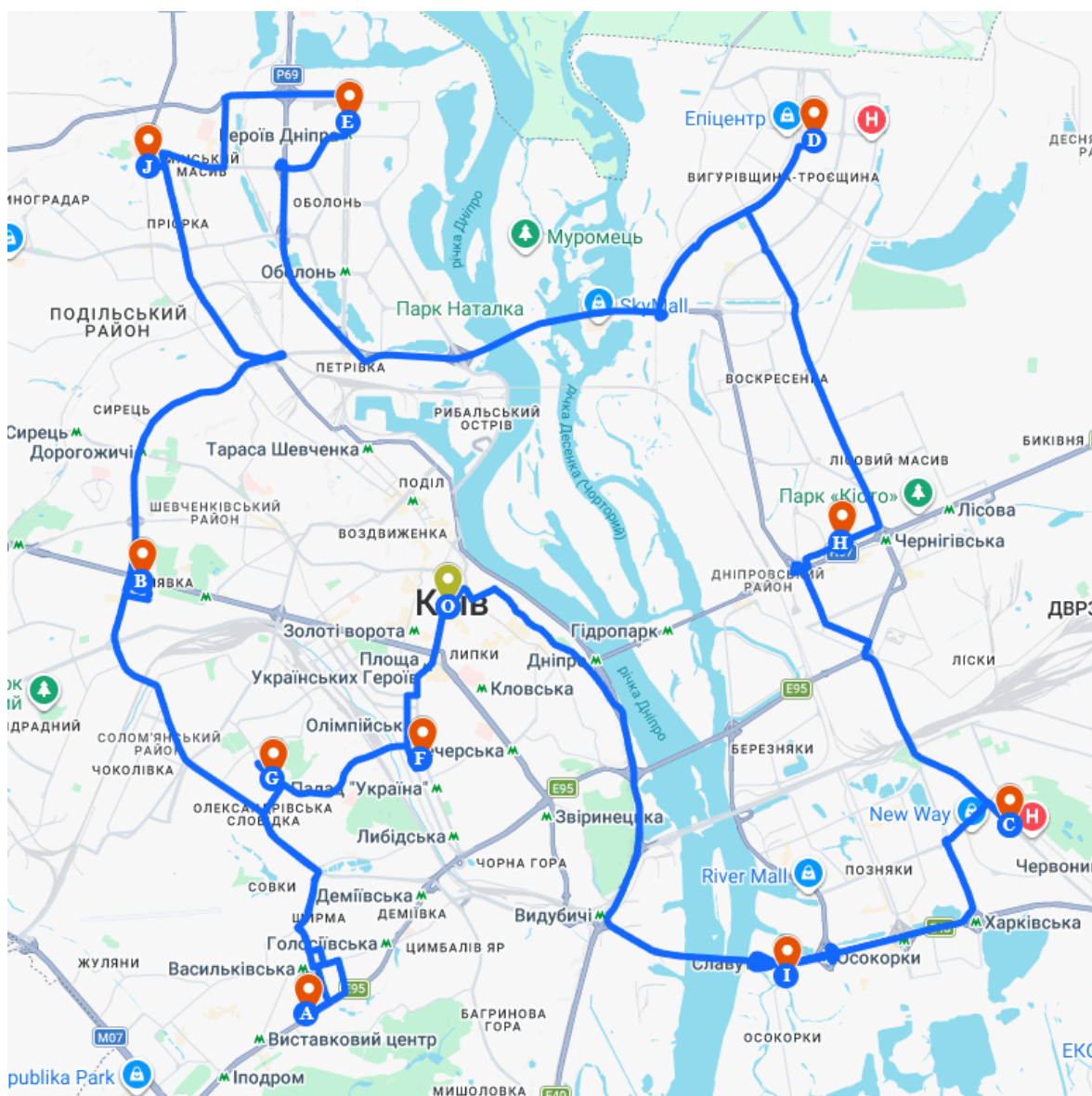


Рисунок 3.2 – Маршрут методу найближчого сусіда (жадібної евристики): $O \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow A \rightarrow B \rightarrow J \rightarrow D \rightarrow C \rightarrow E \rightarrow O$, загальна довжина 92,3 км

3.3. Метод заощаджень Кларка–Райта

3.3.1. Теоретичне обґрунтування

Метод заощаджень (Clarke-Wright Savings Algorithm) розроблений Г. Кларком та Дж. Райтом у 1964 році й є одним з найбільш практично застосовуваних алгоритмів для задач маршрутизації транспорту. Метод ґрунтується на концепції "заощадження": об'єднання двох окремих рейсів до точок i та j в один дозволяє уникнути двох поїздок між складом і кожною з точок, замінивши їх одним

прямим переїздом між точками. [6]

Заощадження від об'єднання точок i та j в один маршрут обчислюється за формулою:

$$s(i, j) = d(O, i) + d(O, j) - d(i, j)$$

де $d(O, i)$ — відстань від складу до точки i , $d(O, j)$ — відстань від складу до точки j , $d(i, j)$ — відстань між точками i та j . Обчислювальна складність: $O(n^2 \log n)$ з урахуванням сортування.

3.3.2. Опис алгоритму

Крок 1. Ініціалізація: кожна точка доставки утворює окремий маршрут $O \rightarrow i \rightarrow O$. Загальна відстань базового рішення: V_0 .

Крок 2. Обчислення заощаджень: для всіх пар (i, j) , $i < j$, обчислити $s(i, j) = d(O, i) + d(O, j) - d(i, j)$. Отримуємо $n(n-1)/2 = 45$ значень заощаджень.

Крок 3. Сортування: впорядкувати всі пари за спаданням значення $s(i, j)$.

Крок 4. Злиття маршрутів: послідовно переглядати відсортований список. Якщо пара (i, j) задовольняє умову злиття (i знаходиться в кінці одного маршруту, j — на початку іншого, і це різні маршрути), виконати злиття.

Крок 5. Повторювати крок 4 до вичерпання списку або формування єдиного маршруту.

3.3.3. Розрахунок заощаджень

Таблиця 3.5 – Топ-10 заощаджень методу Кларка–Райта

№	Пара	$d(i, O)$, км	$d(O, j)$, км	$d(i, j)$, км	$s(i, j)$, км
1	C–I	13.8	14.8	7.9	20.7
2	E–J	11.5	12.5	4.3	19.7
3	C–H	13.8	10.8	7.1	17.5
4	D–H	13.6	10.8	8.8	15.6

5	H–I	9.7	14.8	10.6	13.9
6	C–D	13.8	15.0	15.2	13.6
7	D–E	13.6	11.9	13.5	12.0
8	D–J	13.6	12.5	15.6	10.5
9	A–G	10.7	6.1	6.7	10.1
10	A–C	10.7	14.7	16.0	9.4

Найбільше заощадження дає пара C–I: 20,7 км. Якщо їздити туди окремо, вийде 56,7 км; якщо об'єднати в один маршрут через відрізок C→I (7,9 км) — значно вигідніше.

3.3.4. Результати виконання

Отриманий маршрут: O → E → J → D → H → C → I → A → G → B → F → O

Таблиця 3.6 – Розбивка маршруту за відрізками (метод заощаджень)

Відрізок маршруту	Відстань, км
O → E	11.9
E → J	4.3
J → D	16.4
D → H	8.8
H → C	7.6
C → I	7.9
I → A	11.9
A → G	6.7
G → B	6.0
B → F	6.6
F → O	4.5
Разом:	92.6

Аналіз результату. Загальна довжина маршруту — 92,6 км (заощадження 55,3% від базової відстані). Метод добре виявив географічно близькі кластери точок і суттєво скоротив зайві пробіги між складом та окремими точками. Результат виявився лише на 0,3% гіршим за метод найближчого сусіда. Це пояснюється несиметричністю матриці: коли відстані "туди" і "назад" суттєво різняться, ефект злиття частково нівелюється.

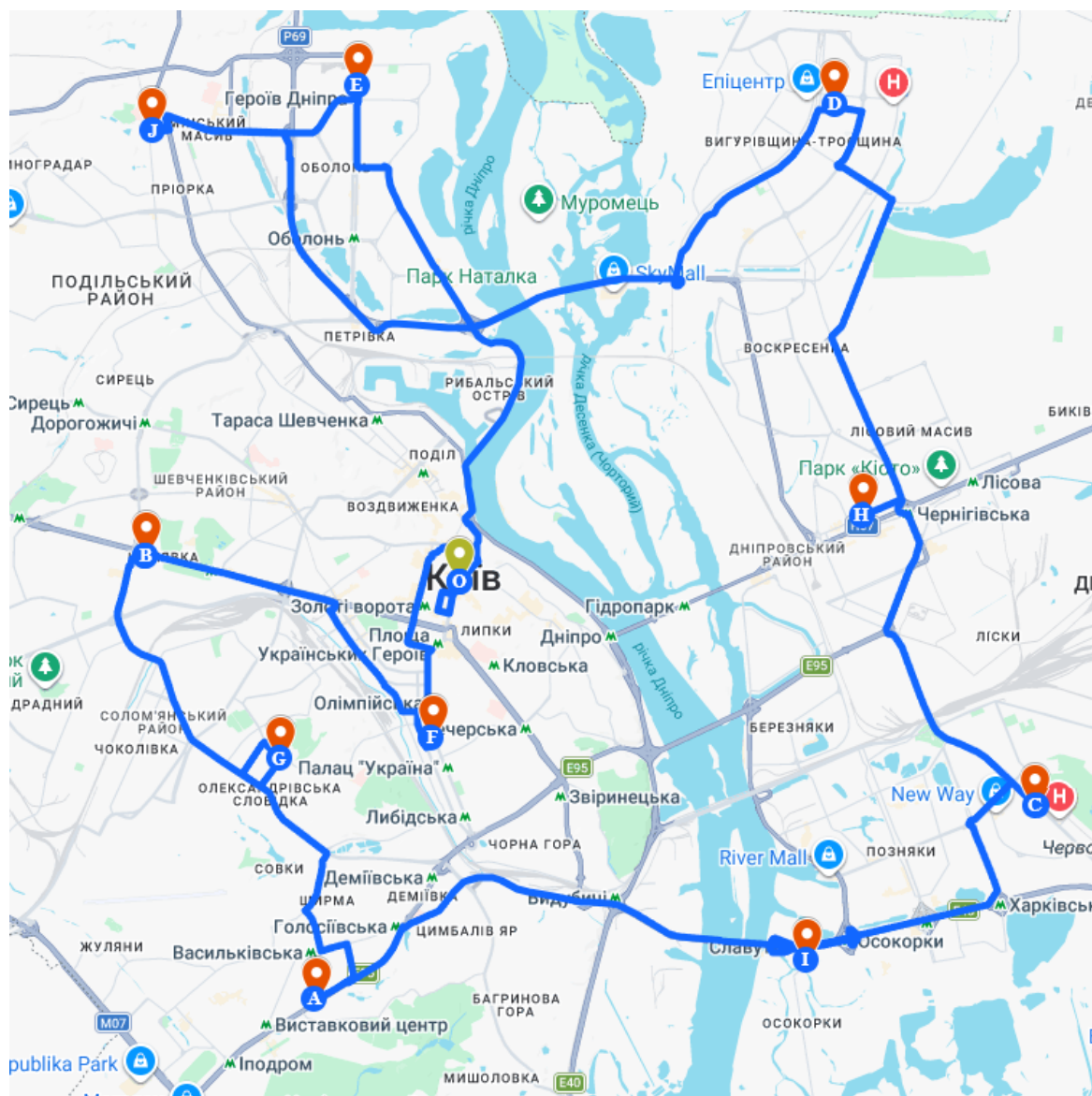


Рисунок 3.3 – Маршрут методу заощаджень Кларка–Райта: $O \rightarrow E \rightarrow J \rightarrow D \rightarrow H \rightarrow C \rightarrow I \rightarrow A \rightarrow G \rightarrow B \rightarrow F \rightarrow O$, загальна довжина 92,6 км

3.4. Генетичний алгоритм

3.4.1. Теоретичне обґрунтування

Генетичний алгоритм — це метаевристика, натхненна еволюційною біологією. Запропонований Голландом у 1975 році, він добре підходить для задач комбінаторної оптимізації, де детерміновані методи вже не справляються. Для задачі TSP ГА переглядає значно більшу частину можливих маршрутів, ніж будь-яка детермінована евристика, і завдяки механізму мутації здатен виходити з локальних оптимумів.

3.4.2. Представлення розв'язку та параметри алгоритму

У задачі TSP хромосома кодується як перестановка вузлів $\{A, B, \dots, J\}$ без повторень. Кожна хромосома являє собою повний маршрут: $O \rightarrow \pi(1) \rightarrow \pi(2) \rightarrow \dots \rightarrow \pi(10) \rightarrow O$. Цільова функція (функція пристосованості): $f(\pi) = F(\pi) =$ загальна довжина маршруту. Мінімізація $f(\pi)$ відповідає максимізації пристосованості особи.

Таблиця 3.7 – Параметри генетичного алгоритму

Параметр	Значення	Одиниця	Обґрунтування
Розмір популяції	$P = 47$	особин	Баланс між різноманітністю та швидкістю збіжності
Кількість поколінь	$G = 300$	ітерацій	Критерій зупинки алгоритму
Ймовірність мутації	$p_m = 0.1$	—	Збереження різноманітності популяції
Розмір турніру	$k = 3$	особин	Селективний тиск відбору
Метод кросинговеру	OX (Order Crossover)	—	Збереження відносного порядку генів
Метод відбору	Турнірний відбір	—	Стохастичний, стійкий до шкалювання

3.4.3. Генетичні оператори

Оператор відбору (турнірний). З поточної популяції випадково обирається $k = 3$ особин. Серед них визначається особина з найменшою довжиною маршруту (найвищою пристосованістю), яка стає батьком. Процедура повторюється двічі для отримання двох батьків.

Оператор кросинговеру (ОХ — Order Crossover). Генетичний матеріал двох батьків P_1 та P_2 комбінується таким чином: (1) обирається випадковий сегмент із P_1 ; (2) цей сегмент копіюється в дочірню хромосому; (3) решта позицій заповнюється генами P_2 у тому порядку, в якому вони зустрічаються, починаючи після кінця скопійованого сегменту. ОХ гарантує отримання коректної перестановки без повторень.

Оператор мутації (перестановка двох генів). З імовірністю $p_m = 0.15$ у хромосомі випадково вибираються дві позиції і відповідні гени міняються місцями. Мутація забезпечує різноманітність популяції та запобігає передчасній збіжності до локального оптимуму.

3.4.4. Збіжність алгоритму

Таблиця 3.8 – Динаміка збіжності генетичного алгоритму

Покоління	Найкраща відстань, км	Приріст якості
0	97,6	—
10	93,6	-4,1%
20	93,6	0,0%
30	89,7	-4,2%
40	89,7	0,0%
50	89,7	0,0%
60	89,7	0,0%
80	89,7	0,0%
100	89,7	0,0%
150	89,7	0,0%

299	89,7	0,0%
-----	------	------

Динаміка збіжності типова для еволюційних методів: перші кілька десятків поколінь дають помітний виграш, потім темп сповільнюється — популяція «насичується» і знаходити нові покращення стає важче. Це сигналізує про те, що популяція поступово вичерпує своє різноманіття і наближається до стаціонарного стану. Стохастична природа алгоритму означає, що результат може незначно відрізнятись між запусками.

3.4.5. Результати виконання

Найкращий знайдений маршрут: $O \rightarrow I \rightarrow C \rightarrow H \rightarrow D \rightarrow E \rightarrow J \rightarrow B \rightarrow G \rightarrow A \rightarrow F \rightarrow O$

Таблиця 3.9 – Розбивка маршруту за відрізками (генетичний алгоритм)

Відрізок маршруту	Відстань, км
$O \rightarrow I$	14.8
$I \rightarrow C$	6.9
$C \rightarrow H$	7.1
$H \rightarrow D$	8.9
$D \rightarrow E$	13.5
$E \rightarrow J$	4.3
$J \rightarrow B$	12.5
$B \rightarrow G$	5.8
$G \rightarrow A$	5.4
$A \rightarrow F$	6.0
$F \rightarrow O$	4.5
Разом:	89.7

Аналіз результату. Генетичний алгоритм знайшов маршрут загальною

довжиною 89.7 км, що становить заощадження 56.7% від базової відстані. На відміну від жадібних методів, ГА не «прив'язаний» до географічно компактних кластерів і може знаходити порядки обходу, що здаються неінтуїтивними, але дають коротший загальний маршрут.

3.5. Порівняльний аналіз методів

3.5.1. Кількісне порівняння результатів

Далі приведені таблиці для порівняння результатів різних методів складання найкоротшого шляху:

Таблиця 3.10 – Порівняльна таблиця методів оптимізації маршруту

Метод	Довжина, км	Відхил. від best	Складність	Час, мс	Тип
Метод найближчого сусіда	92.3	2.9%	$O(n^2)$	< 1	Детерм.
Метод заощаджень К-Р	92.6	3.2%	$O(n^2 \log n)$	< 1	Детерм.
Генетичний алгоритм	89.7	0.0%	$O(G \cdot P \cdot n)$	< 500	Стохаст.

Таблиця 3.11 – Ефективність оптимізації відносно базової відстані

Метод	Базова відст., км	Опт. маршрут, км	Заощадження, км	Заощадження, %
Метод найближчого сусіда	207.0	92.3	114.7	55.4%
Метод заощаджень К-Р	207.0	92.6	114.4	55.3%
Генетичний алгоритм	207.0	89.7	117.3	56.7%

3.5.2. Якісний аналіз

Метод найближчого сусіда демонструє найпростішу реалізацію та

найшвидший час виконання. Алгоритм надійно знаходить прийнятне рішення, формуючи компактні "локальні кластери" на початку маршруту. Проте через жадібну стратегію наприкінці побудови можуть виникати довгі "хвости" — переїзди між географічно віддаленими точками, що залишились невідвіданими. Оптимальний вибір для великих задач ($n > 100$), де швидкість важливіша за точність.

Метод заощаджень Кларка–Райта враховує взаємний вплив усіх пар точок через концепцію заощадження. Алгоритм є детермінованим і відтворюваним, що спрощує верифікацію результатів. Особливо добре він працює, коли точки доставки утворюють виражені географічні кластери, і добре масштабується для задач середнього розміру ($n = 20\text{--}50$). Обмеження: результат залежить від геометрії задачі та може бути субоптимальним при "перехресних" маршрутах.

Генетичний алгоритм забезпечує найбільш глибокий пошук у просторі рішень завдяки стохастичному характеру та механізмам кросинговеру, що дозволяють досліджувати рішення, недоступні детермінованим методам. Алгоритм гнучкий і порівняно легко розширюється на задачі з часовими вікнами або обмеженнями вантажопідйомності. Головний мінус — вищі обчислювальні витрати і стохастичність: два запуски можуть дати різний результат.

3.5.3. Рекомендації щодо застосування

На основі проведеного аналізу сформовано наступні рекомендації щодо вибору методу оптимізації маршруту залежно від умов практичного застосування:

Метод найближчого сусіда доцільно застосовувати, коли: кількість точок доставки перевищує 50; час розрахунку є критичним (оперативне планування); результат використовується як початкове наближення для подальшого покращення.

Метод заощаджень К-Р є оптимальним вибором, коли: задача містить 10–

50 точок доставки; необхідна відтворюваність результату; алгоритм може бути розширений до багатотранспортної задачі (VRP).

Генетичний алгоритм рекомендується, коли: потрібна максимальна якість рішення; задача має додаткові обмеження; допускається час розрахунку від кількох секунд до хвилин; задача може змінюватись динамічно.

3.6. Практичні рекомендації щодо впровадження алгоритмів оптимізації

Числові результати — лише частина відповіді. Не менш важливо зрозуміти, як саме той чи інший алгоритм може виглядати в роботі реальної логістичної компанії. Практична цінність оптимізаційної моделі полягає в переході від «ну, здається, так зручніше» до формалізованого підходу, де рішення спирається на конкретні вимірювані показники.

Для невеликої транспортної мережі, що містить центральний склад і десять точок доставки, усі розглянуті методи можуть бути застосовані на практиці. Однак їхня роль у системі планування маршрутів є різною. Метод найближчого сусіда доцільно використовувати як базовий або контрольний алгоритм, оскільки він швидко будує початковий маршрут і не потребує складного налаштування. Метод Кларка–Райта краще підходить для випадків, коли система має декілька рейсів або кілька транспортних засобів, адже його логіка природно пов'язана з об'єднанням окремих маршрутів. Генетичний алгоритм варто застосовувати тоді, коли головною вимогою є якість рішення, а час розрахунку не є критично обмеженим.

У реальному середовищі логістична задача майже ніколи не зводиться лише до мінімізації відстані. На маршрут впливають затори, стан дорожнього покриття, тимчасові обмеження руху, вартість пального, режим роботи клієнтів, завантаженість автомобіля та пріоритетність замовлень. Тому розроблену модель слід розглядати як базовий етап побудови системи підтримки прийняття рішень. Надалі вона може бути розширена додатковими обмеженнями та критеріями, які зроблять результати ще ближчими до реальних умов міської доставки.

3.6.1. Вибір алгоритму залежно від умов логістичної задачі

Вибір методу оптимізації повинен виконуватися не формально, а з урахуванням конкретних характеристик задачі. Якщо кількість точок доставки невелика, а маршрут потрібно отримати майже миттєво, перевагу доцільно надавати простим евристичним. Вони не гарантують глобального оптимуму, але дозволяють швидко побудувати прийнятне рішення, що особливо важливо для оперативного диспетчерського планування.

Якщо підприємство працює з регулярними маршрутами, які повторюються щодня або щотижня, доцільно застосовувати більш якісні методи пошуку. У такому випадку навіть незначне скорочення пробігу на 2–3% може мати відчутний економічний ефект у довгостроковій перспективі. Наприклад, якщо автомобіль щодня долає 100 км, то навіть скромна економія у 3 км на день дасть 60 км на місяць — і це лише для одного авто. Для парку з кількох машин ефект стає вже цілком відчутним у грошах.

Для задач із високою динамічністю, коли нові замовлення можуть з'являтися протягом дня, доцільним є комбінований підхід. Спочатку система формує швидке рішення за допомогою методу найближчого сусіда або методу заощаджень, а потім, якщо є достатньо часу, виконує додаткове покращення маршруту за допомогою генетичного алгоритму або локального пошуку. Такий підхід дозволяє отримати і швидкість, і якість одночасно.

Таблиця 3.7 – Рекомендації щодо вибору алгоритму залежно від умов задачі

Умова задачі	Доцільний метод	Перевага	Обмеження
Потрібен дуже швидкий розрахунок	Метод найближчого сусіда	Мінімальний час виконання та проста реалізація	Можлива нижча якість маршруту
Є декілька рейсів або транспортних засобів	Метод Кларка–Райта	Добре враховує ефект об'єднання маршрутів	Потребує коректної матриці відстаней
Головна вимога — найкоротший маршрут	Генетичний алгоритм	Краще досліджує простір можливих рішень	Потребує налаштування параметрів
Задача часто змінюється протягом дня	Комбінований підхід	Поеднує швидкість побудови та якість покращення	Складніша програмна реалізація

З наведеної таблиці видно, що універсального алгоритму для всіх умов не існує. Кожен метод має власну область ефективного застосування. Тому для практичного використання доцільно не обмежуватися одним алгоритмом, а реалізувати кілька режимів роботи: швидкий розрахунок, стандартний розрахунок і покращений розрахунок. Користувач або диспетчер може обирати режим залежно від поточної ситуації.

3.6.2. Інтеграція алгоритмів у інформаційну систему підприємства

Для повноцінного використання результатів дослідження алгоритми оптимізації мають бути інтегровані в інформаційну систему підприємства. Така система може складатися з бази даних точок доставки, модуля розрахунку матриці відстаней, модуля оптимізації маршруту та інтерфейсу для відображення результатів. Користувач вводить або імпортує список замовлень, після чого система формує маршрут і показує його у вигляді послідовності точок, загальної довжини та очікуваного часу виконання.

На першому етапі достатньо використовувати статичну матрицю відстаней, як це зроблено у цій роботі. Такий підхід є простим, контрольованим і зручним для навчальної або тестової реалізації. Проте для промислового використання доцільно автоматизувати формування матриці на основі картографічних сервісів або внутрішніх даних підприємства. Це дозволить враховувати актуальні дорожні умови та швидко оновлювати вхідні дані при зміні адрес доставки.

Важливою вимогою до системи є прозорість отриманого результату. Диспетчер повинен бачити не лише готовий маршрут, а й основні показники, на підставі яких система рекомендує саме таку послідовність відвідування точок. До таких показників належать загальна довжина маршруту, відсоток економії порівняно з базовим планом, час розрахунку, кількість ітерацій для генетичного алгоритму та відхилення від найкращого знайденого рішення.

Окрему увагу необхідно приділити можливості ручного коригування маршруту. Навіть якісний алгоритм не завжди може врахувати всі неформалізовані фактори, наприклад домовленість із клієнтом, терміновість

окремого замовлення або тимчасову недоступність певної дороги. Тому оптимальним є режим, у якому система пропонує маршрут, а відповідальний працівник може підтвердити його або внести корективи. Після ручних змін система має перераховувати загальну довжину та показувати, як ці зміни вплинули на ефективність.

3.6.3. Обмеження проведеного дослідження

Проведене дослідження має певні обмеження, які необхідно враховувати під час інтерпретації результатів. По-перше, у роботі використано транспортну мережу з одинадцяти вузлів, що є достатнім для демонстрації принципів роботи алгоритмів, але не відображає повний масштаб задач великих логістичних операторів. У реальних умовах кількість точок доставки може становити десятки, сотні або навіть тисячі адрес, що істотно змінює вимоги до продуктивності алгоритмів.

По-друге, основним критерієм оптимізації у роботі виступає довжина маршруту. Такий критерій є логічним і зручним для порівняння, однак у практичній логістиці часто потрібно враховувати декілька критеріїв одночасно. Наприклад, найкоротший маршрут не завжди є найшвидшим через затори або обмеження швидкості. Так само маршрут із мінімальною відстанню може бути менш вигідним за витратами, якщо він проходить через ділянки з частими зупинками, складними розворотами або платними дорогами.

По-третє, у моделі не враховано часові вікна обслуговування клієнтів. У багатьох логістичних задачах клієнт може прийняти доставку лише у певний проміжок часу. Це перетворює задачу на VRPTW і потребує складніших алгоритмічних підходів. Для таких умов просте скорочення відстані вже не є достатнім критерієм, оскільки маршрут має бути не лише коротким, а й таким, що виконує всі часові обмеження.

По-четверте, практична реалізація генетичного алгоритму залежить від вибору параметрів: розміру популяції, кількості поколінь, імовірності мутації та способу відбору. Занадто мала популяція може призвести до передчасної збіжності, а надмірно велика — до зайвих обчислювальних витрат. Тому для

промислового використання необхідно проводити додаткове налаштування параметрів на реальних історичних даних підприємства.

Таблиця 3.8 – Основні обмеження дослідження та способи їх подолання

Обмеження	Вплив на результат	Можливий спосіб подолання
Невелика кількість вузлів	Результати демонструють принцип, але не повністю відображають великі мережі	Провести тестування на 50–100 і більше точках доставки
Оптимізація лише за відстанню	Не враховуються затори, час руху та вартість виконання рейсу	Додати час, витрати та штрафи до цільової функції
Відсутність часових вікон	Маршрут може бути коротким, але незручним для клієнтів	Розширити постановку до VRPTW
Залежність ГА від параметрів	Якість рішення може змінюватися між запусками	Виконати серію експериментів і підібрати параметри

3.6.4. Напрями подальшого розвитку моделі

Подальший розвиток моделі може здійснюватися у кількох напрямках. Перший напрям пов'язаний із переходом від статичної до динамічної маршрутизації. У статичній постановці всі точки доставки відомі до початку розрахунку. У динамічній постановці нові замовлення, скасування або зміни пріоритетів можуть з'являтися вже під час виконання маршруту. Для таких задач потрібні алгоритми швидкого перерахунку, які не будують маршрут з нуля, а адаптують його до нових умов.

Другий напрям полягає у врахуванні часових обмежень. Для цього до кожної точки доставки можна додати часовий інтервал обслуговування, а до кожної дуги графа — очікуваний час руху. У такому випадку цільова функція може враховувати не лише відстань, а й штрафи за запізнення або раннє прибуття. Це дозволить наблизити модель до задач кур'єрської доставки, міської дистрибуції та сервісного обслуговування.

Третій напрям розвитку — урахування вантажопідйомності транспортних засобів та обсягу замовлень. У базовій постановці TSP передбачається один маршрут, у якому всі точки відвідуються один раз. Однак у реальному підприємстві один автомобіль може не мати достатньої місткості для всіх

замовлень. Тому задача повинна розширюватися до VRP із декількома транспортними засобами, обмеженнями за вагою, об'ємом або кількістю місць.

Четвертий напрям пов'язаний із використанням гібридних алгоритмів. Наприклад, генетичний алгоритм може формувати загальну структуру маршруту, а локальний пошук 2-орт або 3-орт — додатково покращувати окремі ділянки. Такий підхід часто дає кращі результати, ніж використання одного алгоритму, оскільки поєднує глобальний пошук із локальним удосконаленням.

П'ятий напрям — розроблення візуального інтерфейсу, у якому результати оптимізації відображаються на карті. Це зробить систему зрозумілішою для кінцевого користувача та полегшить перевірку коректності маршруту. Візуалізація також дозволяє швидко помітити логічні помилки: зайві повернення, надмірні перетини маршруту, не вигідні відхилення або неприродний порядок відвідування точок.

Отже, виконана у роботі реалізація є базовою, але достатньо змістовною основою для подальшого розвитку. Вона демонструє, що навіть порівняно прості евристичні та еволюційні методи можуть суттєво покращити якість логістичного планування. При цьому найбільший практичний ефект може бути досягнутий не за рахунок одного окремого алгоритму, а завдяки комплексному підходу: якісним вхідним даним, правильному вибору методу, можливості ручного контролю та поступовому розширенню моделі під реальні потреби підприємства.

3.7. Висновки до розділу 3

У третьому розділі були реалізовані та порівняні три методи оптимізації маршруту на реальній задачі для транспортної мережі Києва з одинадцятьма вузлів. Вхідні дані представлені несиметричною матрицею відстаней розміром 11×11 , що відповідає реальним умовам міського трафіку.

Базова відстань без оптимізації (сума окремих поїздок до кожної точки) становить 207.0 км. Усі три методи продемонстрували суттєве скорочення загального пробігу: метод найближчого сусіда — 92.3 км (–55.4%), метод заощаджень Кларка–Райта — 92.6 км (–55.3%), генетичний алгоритм — 89.7 км (–56.7%).

Найкращий результат показав генетичний алгоритм — 89,7 км. Практичні дані підтвердили теоретичні очікування: прості жадібні методи виграють у швидкості, але еволюційний підхід дає кращий маршрут, коли є час на розрахунок. Вибір конкретного методу визначається балансом між вимогами до якості рішення, часом розрахунку та складністю реалізації у конкретній логістичній системі.

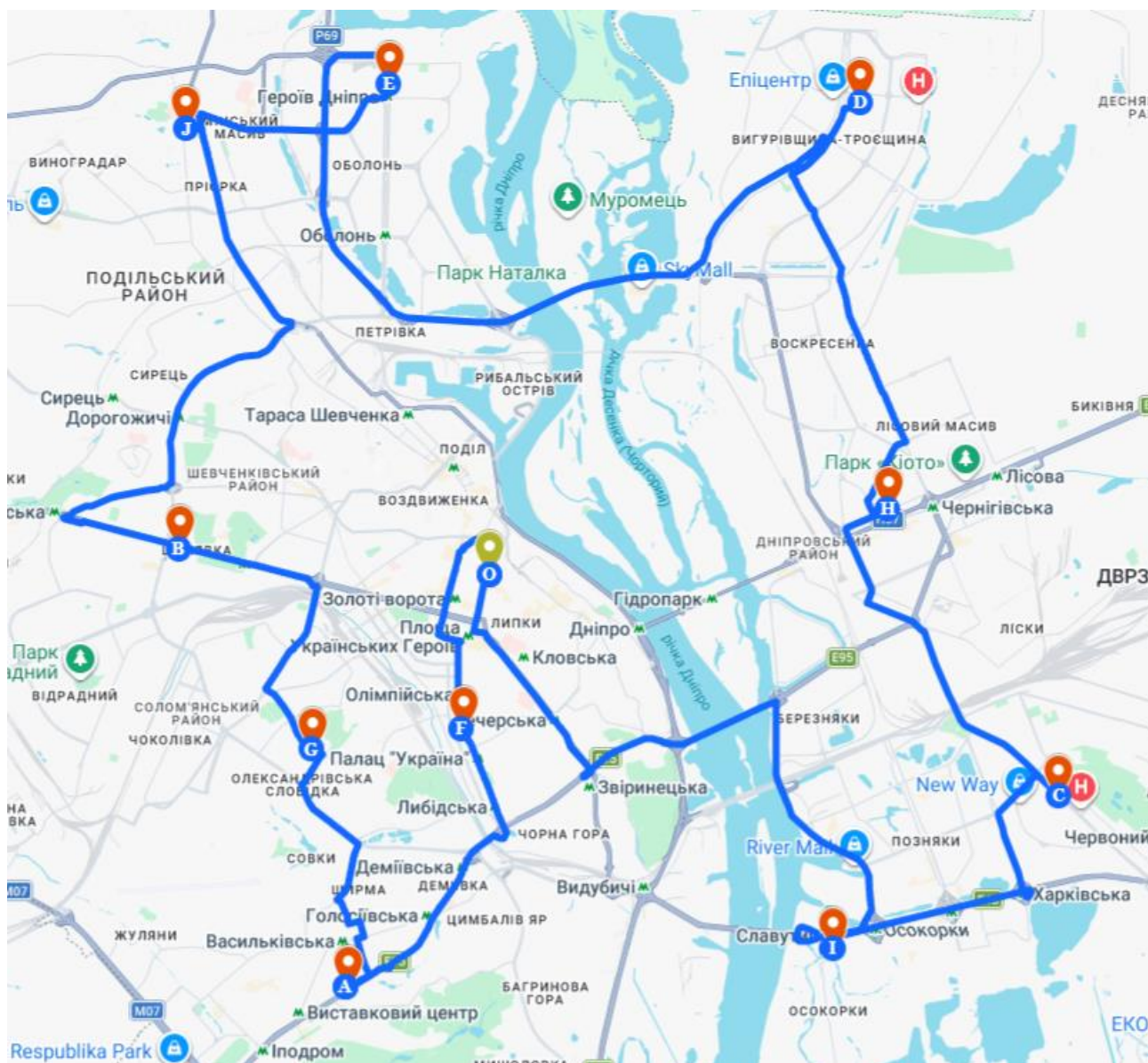


Рисунок 3.4 – Маршрут генетичного алгоритму: $O \rightarrow I \rightarrow C \rightarrow H \rightarrow D \rightarrow E \rightarrow J \rightarrow B \rightarrow G \rightarrow A \rightarrow F \rightarrow O$, загальна довжина 89,7 км (найкращий результат)

ВИСНОВКИ

У бакалаврській роботі досліджено методи оптимізації логістичних маршрутів та виконано їх порівняльну апробацію на реальній транспортній мережі Києва. Нижче наведено основні результати та висновки.

1. У першому розділі систематизовано теоретичні основи оптимізації логістичних процесів. Встановлено, що логістичні витрати загалом становлять від 10 до 30% вартості продукції, а транспортна складова в цій структурі зазвичай найбільша — 40–60%. Сучасна логістична мережа є складною системою з нелінійними взаємодіями між компонентами, що обумовлює необхідність застосування формальних методів комбінаторної оптимізації. Задача комівояжера (TSP) та задача маршрутизації транспортних засобів (VRP) є NP-важкими задачами, для яких кількість допустимих рішень зростає факторіально: для 10 точок простір розв'язків містить $10! = 3\,628\,800$ варіантів, що унеможливорює повний перебір навіть для задач малої розмірності.

2. У другому розділі виконано детальний теоретичний аналіз трьох алгоритмів оптимізації маршрутів. Метод найближчого сусіда вирізняється мінімальною складністю $O(n^2)$ і простотою реалізації, але на завершальних кроках систематично погіршує маршрут через так звану «проблему хвоста». Метод заощаджень Кларка–Райта із складністю $O(n^2 \log n)$ враховує глобальну структуру задачі через концепцію заощадження $s(i,j) = d(O,i) + d(O,j) - d(i,j)$ і природно розширюється на багатотранспортну задачу VRP. Генетичний алгоритм принципово відрізняється стохастичним характером та здатністю досліджувати широкий простір рішень через механізми турнірного відбору, кросинговеру (оператор ОХ) та мутації, що дозволяє виходити за межі локальних оптимумів, недосяжних для детермінованих конструктивних евристик.

3. У третьому розділі проведено практичну апробацію методів на реальній задачі маршрутизації для транспортної мережі міста Києва, що включає центральний склад (вул. Хрещатик) та 10 точок доставки в різних районах міста. Вхідні дані представлені несиметричною матрицею відстаней 11×11 , що

відображає реальні умови міського руху: одностороннє сполучення, різницю пропускнуї спроможності маршрутів у протилежних напрямках (розбіжність до 21,6% для окремих пар вузлів). Базова відстань без оптимізації (сума окремих поїздок до кожної точки і назад) становить 207,0 км.

4. Порівняльний аналіз результатів трьох методів підтвердив їх високу практичну ефективність: метод найближчого сусіда забезпечив маршрут довжиною 92,3 км (скорочення пробігу на 55,4%), метод заощаджень Кларка–Райта — 92,6 км (–55,3%), генетичний алгоритм — 89,7 км (–56,7%). Найкращий результат показав генетичний алгоритм — на 2,6–3,2% кращий за детерміновані евристики при часі обчислення до 500 мс (проти менш ніж 1 мс для евристик). Цей розрив невеликий, але в умовах щоденної роботи він може перетворюватися на відчутну економію. Результат підтверджує: якщо якість маршруту важливіша за миттєву відповідь — еволюційні методи виправдовують свою складність.

5. Наукова новизна роботи полягає у проведенні кількісної оцінки ефективності трьох класів алгоритмів — жадібної евристики, ощадного методу та генетичного алгоритму — на єдиній тестовій задачі з несиметричною матрицею відстаней, що моделює реальні умови транспортної мережі Києва. Вперше для цієї конфігурації задачі встановлено, що еволюційний підхід забезпечує перевагу у 2,6–3,2% порівняно з детермінованими евристичними при обчислювальних витратах, прийнятних для практичного застосування.

6. Практичне значення отриманих результатів полягає в тому, що розроблені та апробовані методи оптимізації маршрутів можуть бути безпосередньо впроваджені в логістичних компаніях, службах кур'єрської доставки та поштових операторах для автоматизованого планування транспортних маршрутів у міському середовищі. Досягнуте скорочення загального пробігу на 55–57% порівняно з нераціональним базовим плануванням (89,7–92,6 км проти 207,0 км) демонструє значний економічний потенціал впровадження алгоритмічних методів оптимізації в реальних логістичних системах.

Перспективи подальшої роботи — розширення моделі до VRPTW з урахуванням часових вікон і вантажопідйомності, а також дослідження

динамічної маршрутизації в умовах реального часу. Цікавим напрямком залишається і розробка гібридних методів — наприклад, де генетичний алгоритм задає загальну структуру маршруту, а локальний пошук 2-opt чи 3-opt шліфує деталі. Окремо варто розглянути застосування методів машинного навчання для адаптивної побудови матриць відстаней та прогнозування оптимальних маршрутів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Christopher M. Logistics and Supply Chain Management. 5th ed. Pearson Education, 2016. 328 p.
2. Chopra S., Meindl P. Supply Chain Management: Strategy, Planning, and Operation. 7th ed. Pearson, 2019. 528 p.
3. Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., Shmoys D. B. The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley, 1985. 476 p.
4. Karp R. M. Reducibility Among Combinatorial Problems. Complexity of Computer Computations. 1972. P. 85–103.
5. Dantzig G. B., Ramser J. H. The Truck Dispatching Problem. Management Science. 1959. Vol. 6, No. 1. P. 80–91.
6. Clarke G., Wright J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research. 1964. Vol. 12, No. 4. P. 568–581.
7. Holland J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975. 183 p.
8. Goldberg D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989. 432 p.
9. Dorigo M., Stützle T. Ant Colony Optimization. MIT Press, 2004. 305 p.
10. Vinyals O., Fortunato M., Jaitly N. Pointer Networks. Advances in Neural Information Processing Systems. 2015. P. 2692–2700.
11. Toth P., Vigo D. Vehicle Routing: Problems, Methods, and Applications. 2nd ed. SIAM, 2014. 463 p.
12. Gutin G., Punnen A. P. The Traveling Salesman Problem and Its Variations. Springer, 2007. 830 p.



Кваліфікаційна робота

На тему: Методи та алгоритми оптимізації логістичних процесів в складних системах

Викнав: Богдан Станіслав Валентинович

Керівник: Патракеєв Ігор Михайлович
канд. тех. наук, доцент

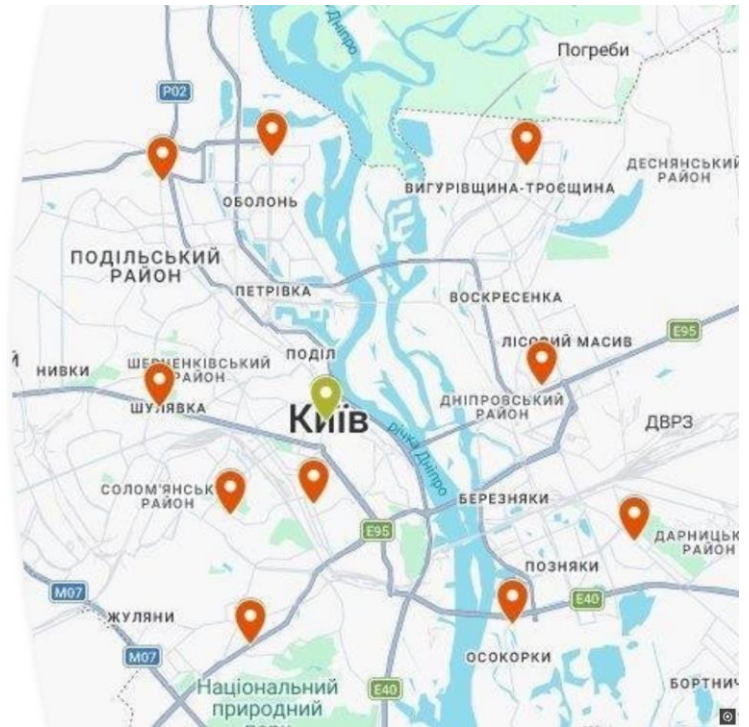
powerpoint.officeapps.live.com – щоб вийти з повноекранного режиму, натисніть клавішу Esc

Мета, об'єкт та предмет дослідження

- **Мета роботи** – підвищення ефективності логістичних процесів у складних транспортно-розподільчих системах шляхом дослідження, порівняння та практичного застосування методів і алгоритмів оптимізації маршрутів доставки.
- **Об'єкт дослідження** – логістичні процеси транспортно-розподільчих систем доставки.
- **Предмет дослідження** – методи, алгоритми та моделі оптимізації маршрутів у логістичних системах, зокрема метод найближчого сусіда, метод заощаджень Кларка–Райта та генетичний алгоритм.

Початкові дані

- 10 точок + точка початку та кінця
- 3 628 800 можливих варіантів перебору
- 3 методи оптимізації маршрутів
- Несиметрична матриця відстаней 11x11



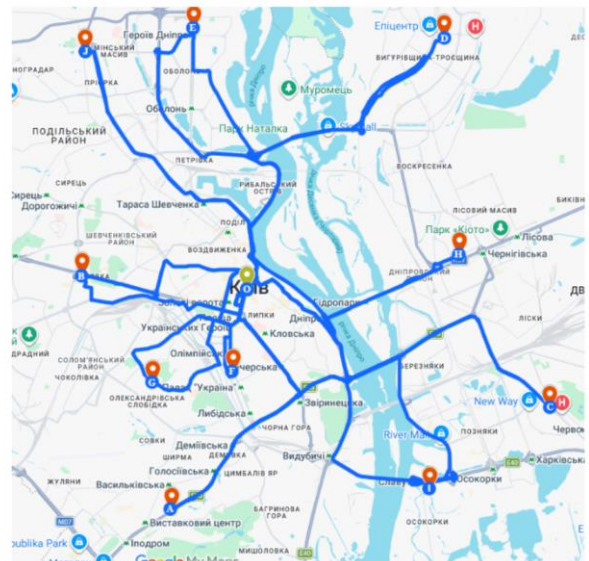
Базовий (неоптимальний) маршрут

Довжина базового маршруту = 207 км

- Формула розрахунку відстані базового маршруту:

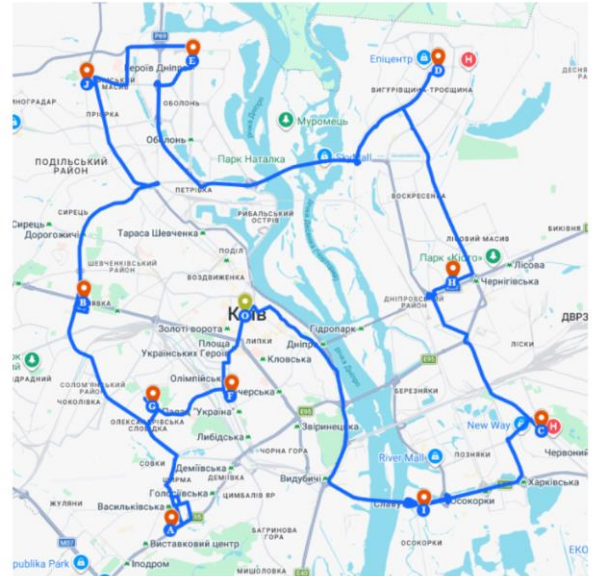
$$O_0 = \sum (d(O, i) + d(i, O)) = 207.0$$
- Формальна постановка задачі:

$$F(\pi) = d(O, \pi_1) + \sum_{i=1}^n d(\pi_i, \pi_{i+1}) + d(\pi_n, O) \rightarrow \min$$
- Методи для розв'язання:
 - Метод найближчого сусіда
 - Метод заощаджень Кларка-Райта
 - Генетичний алгоритм



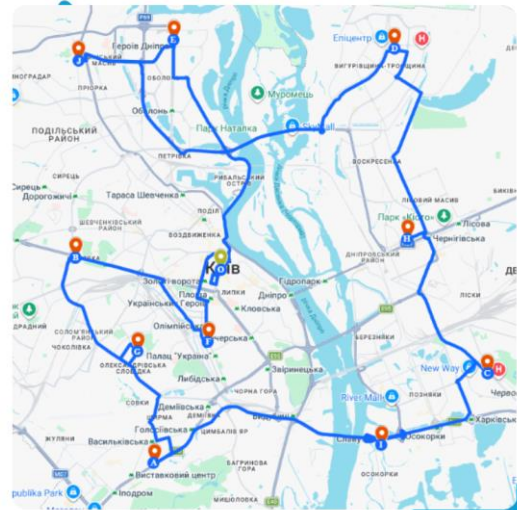
Метод найближчого сусіда

- Формула для розрахунку: $j^* = \operatorname{argmin} \{ d(v_k, j) : j \notin \text{visited} \}$, де v_k — поточний вузол
- Алгоритм роботи:
 - Почати маршрут зі складу О.
 - Знайти найближчу невіддану точку.
 - Перейти до неї та позначити як віддану.
 - Повторювати пошук найближчої точки до відвідування всіх вузлів.
 - Повернутися до складу.
- Отриманий маршрут: $O \rightarrow F \rightarrow G \rightarrow A \rightarrow B \rightarrow J \rightarrow E \rightarrow D \rightarrow H \rightarrow C \rightarrow I \rightarrow O$
- Сумарна відстань 92.3 км
- На 55.4% коротший за базовий маршрут



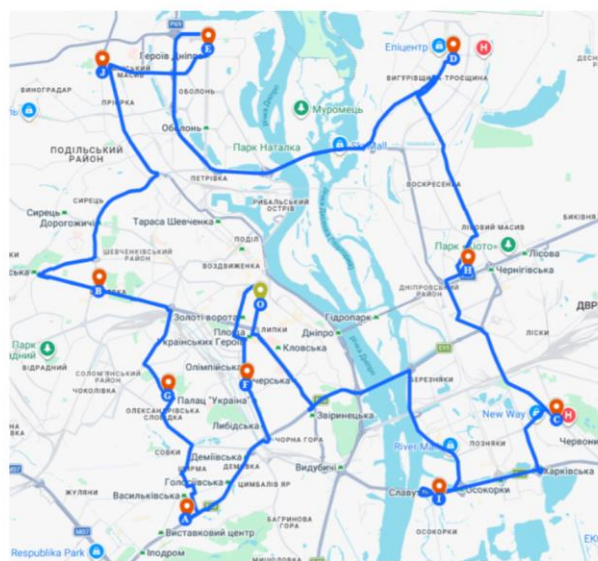
Метод заощаджень Кларка–Райта

- Концепція заощаджень:
 $s(i, j) = d(O, i) + d(O, j) - d(i, j)$
- Алгоритм роботи:
 - Створити окремий маршрут для кожної точки.
 - Обчислити заощадження для всіх пар вузлів.
 - Відсортувати заощадження за спаданням.
 - Послідовно об'єднувати маршрути з найбільшим заощадженням.
 - Отримати фінальний маршрут.
- Отриманий маршрут: $O \rightarrow E \rightarrow J \rightarrow D \rightarrow H \rightarrow C \rightarrow I \rightarrow A \rightarrow G \rightarrow B \rightarrow F \rightarrow O$
- Загальна відстань маршруту: 92,6 км
- На 55.3% коротший за базовий маршрут



Генетичний алгоритм

- Алгоритм генетичного алгоритму:
 - Генерація початкової популяції маршрутів.
 - Обчислення пристосованості.
 - Вибір найкращих особин.
 - Схрещування батьківських рішень.
 - Мутація маршрутів.
 - Формування нового покоління.
 - Перевірка умови завершення.
 - Вибір найкращого маршруту.
- Найкращий знайдений маршрут: O → I → C → H → D → E → J → B → G → A → F → O
- Загальна відстань маршруту: 89.7 км
- На 56.7% коротший за базовий маршрут



Аналіз та порівняння методів

Метод	Довжина шляху	Заощадженно	Відхилення від найк. шляху	Час розрахунку
Метод найближчого сусіда	92.3 км	114.7 км (55.4%)	+2.9%	0.1 мс
Метод Кларка–Райта	92.6 км	114.4 км (55.3%)	+3.2%	0.3 мс
Генетичний алгоритм	89.7 км	117.3 км (56.7%)	0.0%	368 мс
Ваги для критеріїв	0.4	0.3	0.2	0.1

Розрахунок ієрархії

- 1. Метод найближчого сусіда

$$S_{NN} = 0.4Q \frac{89.7}{92.3} + 0.3Q \frac{55.4}{56.7} + 0.2Q \left(1 - \frac{2.9}{3.2}\right) + 0.1Q \frac{2.9}{3.2}$$

$$S_{NN} = 0.4Q \cdot 0.972 + 0.3Q \cdot 0.977 + 0.2Q \cdot 0.094 + 0.1Q \cdot 1$$

$$S_{NN} = 0.389 + 0.293 + 0.019 + 0.100$$

$$S_{NN} \approx 0.801$$
- 2. Метод Кларка–Райта

$$S_{CW} = 0.4Q \frac{89.7}{92.6} + 0.3Q \frac{55.3}{56.7} + 0.2Q \left(1 - \frac{3.2}{3.2}\right) + 0.1Q \frac{2.9}{3.2}$$

$$S_{CW} = 0.4Q \cdot 0.969 + 0.3Q \cdot 0.975 + 0.2Q \cdot 0 + 0.1Q \cdot 0.333$$

$$S_{CW} = 0.388 + 0.293 + 0 + 0.033$$

$$S_{CW} \approx 0.713$$
- 3. Генетичний алгоритм

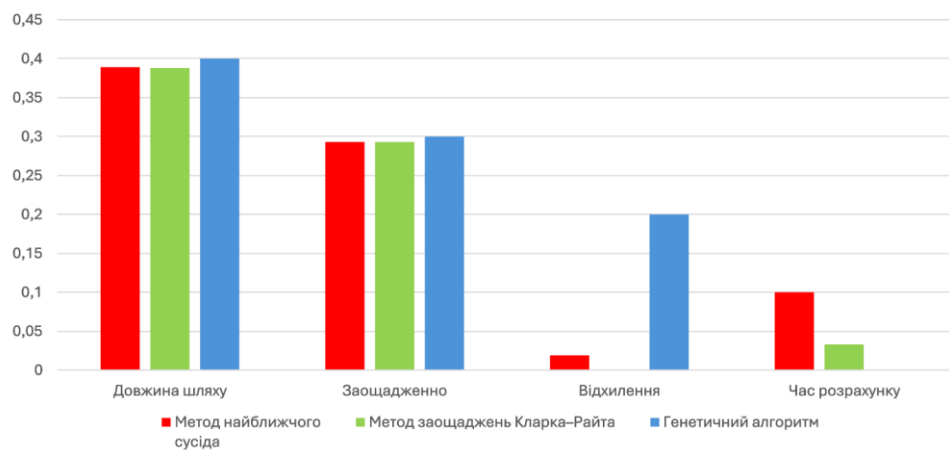
$$S_{GA} = 0.4Q \frac{89.7}{89.7} + 0.3Q \frac{56.7}{56.7} + 0.2Q \left(1 - \frac{0}{3.2}\right) + 0.1Q \frac{0.1}{3.68}$$

$$S_{GA} = 0.4Q \cdot 1 + 0.3Q \cdot 1 + 0.2Q \cdot 1 + 0.1Q \cdot 0.00027$$

$$S_{GA} = 0.400 + 0.300 + 0.200 + 0.000027$$

$$S_{GA} \approx 0.900$$

Оцінка за критеріями

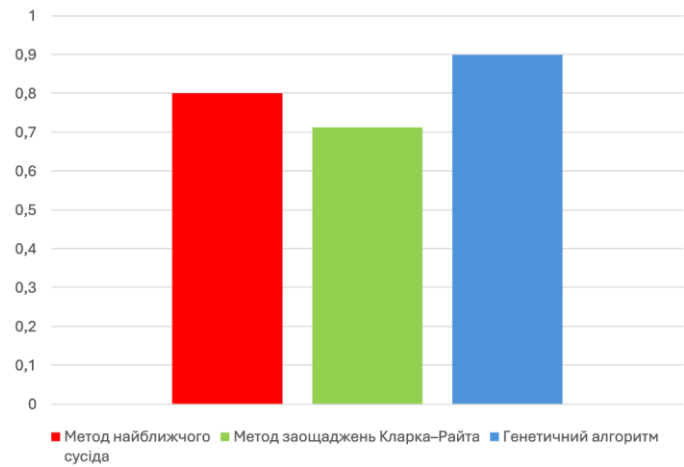


Загальна Оцінка

1. Генетичний алгоритм = 0,9

2. Метод найближчого сусіда = 0,801

3. Метод Кларка–Райта = 0,713



Висновки

- Проведено порівняльний аналіз трьох методів маршрутизації (11 вузлів, Київ)
- Усі три методи скорочують базовий маршрут (207 км) більш ніж на 55%
- Генетичний алгоритм забезпечує найкращий результат — 89.7 км (–56.7% від базового)
- Метод найближчого сусіда — найшвидший (<1 мс), результат лише на 3% гірший за ГА
- Метод Кларка–Райта є ефективним для задач з кількома транспортними засобами (VRP)
- Рекомендовано гібридний підхід: евристика для початкового рішення + ГА для його покращення
- Метод аналізу ієрархій підтвердив перевагу генетичного алгоритму як найефективнішого рішення серед досліджених методів.