

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління

“ ” 20 21 p.

«Маршрутизація транспортних засобів з часовими вікнами»

Київ – 2021 року

РЕФЕРАТ

Магістерська дисертація: 108 с., 27 рис., 14 табл., 67 джерел.

Актуальність. Згідно з 30м щорічним звітом з логістики від The Council of Supply Chain Management Professionals (CSCMP) [1] лише в США за рік було витрачено понад 1.64 трильйонів доларів на логістичні та транспортні операції, а кінцева вартість продукту, що потрапляє до споживача, може складатись до 70% з транспортних витрат.

Тож перед галуззю транспортної логістики однією з важливіших задач постає економія ресурсів та мінімізація екологічного сліду при транспортуванні вантажів. Для вирішення цієї проблеми стоїть питання розробки алгоритмів та програмних продуктів, що будуть скорочувати маршрути транспортних засобів. В останні роки значно збільшилась частка персональних доставок, які прив'язанні до зайнятості клієнтів, актуальним є питання врахування часових вподобань одержувачів вантажу.

Математичне формулювання цієї задачі відоме як задача маршрутизації транспортних засобів (далі VRP) з урахуванням часових вікон (далі VRPTW), яка накладає певні часові обмеження на обслуговування клієнтів транспортної мережі. Робота присвячена дослідженню та удосконаленню розв'язання задачі VRPTW.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів» (№ ДР 0117U000919).

Мета роботи і завдання дослідження. Метою є підвищення ефективності методів розв'язання задачі маршрутизації транспортних засобів з часовими вікнами. Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати відомі результати розв'язання задачі

маршрутизації транспортних засобів;

- удосконалити існуючі алгоритми розв’язання задачі маршрутизації транспортних засобів з урахуванням часових вікон за рахунок модифікації та поєднання метаевристик;

- розробити програмну реалізацію розроблених алгоритмів;

- провести дослідження ефективності розроблених алгоритмів.

Об’єкт дослідження – процес організації транспортних перевезень.

Предмет дослідження – задача маршрутизації транспортних засобів з часовими вінками.

Методи дослідження, застосовані в роботі, базуються на методах дослідження операцій, зокрема на метаевристичних алгоритмах.

Наукова новизна отриманих результатів. Розроблені модифікований та гібридний алгоритми розв’язання задачі VRPTW.

Публікації. Основні теоретичні та практичні положення викладено в матеріалах VI всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021).

МАРШРУТИЗАЦІЯ ТРАНСПОРТНИХ ЗАСОБІВ, ЗАДАЧА VRP, ЗАДАЧА VRPTW, ЧАСОВІ ВІКНА, МЕТАЕВРИСТИЧНІ АЛГОРИТМИ, ГІБРИДНИЙ АЛГОРИТМ, АСО, ТАБУ ПОШУК.

ABSTRACT

Master dissertation: 108 p., 27 fig., 14 tab, 67и sources.

The relevance. According to the 30th Annual State of Logistics Report by the Council of Supply Chain Management Professionals (CSCMP) there was spent over 1.64 trillion on logistics and transportation operations in USA [1]. The final costs of the product that reaches the consumer can consist up to 70% of transportation costs.

Therefore, one of the most important tasks for the transportation logistics industry is to save resources and minimize the environmental footprint during transportation of goods. To solve this problem, it is necessary to develop algorithms and software products that will decrease the routes of vehicles. In recent years significantly increased the proportion of personal delivery that employment associated with a client topical issue taking into account time preferences consignees.

The mathematical formulation of this problem is known as the vehicle routing problem (VRP) with time windows (VRPTW), which imposes certain time constraints on the service of customers. The work is devoted to research and improvement of the VRPTW problem. The work is devoted to the study and improvement of solving VRPTW problem.

Relationship of work with scientific programs, plans, themes. The work was done at the department of computer-aided management and data processing systems of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» within the theme «Effective methods for solving problems of scheduling theory» (№ DR 0117U000919).

Purpose and objectives of the study. The goal of the research is to is to minimize the total cost of transportation of products to customers in a certain time period. To achieve this goal it is necessary to solve the following **tasks**:

- to analyze known results of solving the Vehicle Routing Problem;
- to improve the existing algorithms for solving the problem of vehicle routing considering time windows by modifying and combining metaheuristics;

- to develop a software implementation of the developed algorithms;
- to conduct research on the effectiveness of the developed algorithms.

The object of study – the process organisation of transportation.

Purpose of the study – Vehicle Routing Problem with Time Windows.

Methods used in the paper are based on the methods of operations research, such as metaheuristics algorithms.

Scientific novelty. New modified and hybrid algorithms developed for solving VRPTW.

Publications. The results of the research were published in the materials of VI Ukrainian scientific and practical conference of young scientists and students "Information Systems and Management Technologies" (ISTU-2021).

VEHICLE ROUTING, VRP PROBLEM, VRPTW PROBLEM, THE TIME WINDOW, METAHEURISTICS ALGORITHM, HYBRID ALGORITHM, ACO TABU SEARCH.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	10
1 АНАЛІЗ ЗАДАЧ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ ТА ІСНУЮЧИХ МЕТОДІВ РОЗВ’ЯЗАННЯ	12
1.1 Класифікація задач оптимізації маршрутів транспортних засобів....	12
1.2 Опис класичної задачі vehicle routing problem (vrp)	18
1.2.1 Постановка задачі	19
1.2.2 Математична постановка задачі	20
1.3 Огляд різновидів задач маршрутизації транспортних засобів	22
1.4 Аналіз класичних алгоритмів розв'язання задач маршрутизації транспорту	28
1.4.1 Точні методи.....	29
1.4.2 Евристичні методи.....	31
1.4.3 Метаевристичні методи.....	32
1.5 Огляд існуючих програмних продуктів	40
1.6 Постановка задачі дослідження.....	40
Висновки до розділу	41
2 РОЗРОБКА МОДИФІКАЦІЙ АЛГОРИТМУ МУРАШИНИХ КОЛОНІЙ ТА ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ЗАДАЧІ VRPTW	42
2.1 Математична постановка задачі VRPTW	43
2.2 Алгоритми ant colony optimization для розв’язання VRPTW	46
2.2.1 Опис класичного алгоритму ant colony optimization	46
2.3 Переваги та недоліки мурашиного алгоритму.....	52
2.4 Модифікації алгоритму мурашиних колоній	54
2.5 Розробка гібридного алгоритму	55
Висновки до розділу	57
3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	58

3.1 Засоби розробки	58
3.2 Архітектура програмного забезпечення	60
3.2.1 Структура класів програмного забезпечення	60
3.2.1 Специфікація функцій програмного забезпечення	61
3.3 Вхідні данні	64
3.4 Вихідні данні	65
3.5 Інструкція користувача.....	65
Висновки до розділу	69
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ	70
4.1 Мета досліджень	70
4.2 Тестові данні.....	70
4.3 Результати досліджень ефективності запропонованих модифікацій алгоритму мурашиних колоній.....	71
4.4 Аналіз результатів досліджень	76
Висновки до розділу	77
ВИСНОВКИ.....	95
РЕКОМЕНДАЦІЇ.....	97
ПЕРЕЛІК ПОСИЛАНЬ	98
ДОДАТОК А.....	104
Класифікація характеристик vgr.....	105
Ієрархічна схема узагальнень і розширень vgr	106
Класифікація найпоширеніших алгоритмів розв’язання vgr.....	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ТЗ – транспортний засіб.

ПЗ – програмне забезпечення.

CO- Combinatorial optimization, комбінаторна оптимізація.

VRP - Vechicle Routing Problem, задача маршрутизації транспортних засобів.

AVRP - Asymmetric cost matrix VRP, Асиметрична задача маршрутизації.

CVRP - Capacitated Vehicle Routing Problem, задача маршрутизації транспортних засобів з урахуванням завантаженості транспортних засобів.

DCVRP - Distance-Constrained VRP, задача маршрутизації транспорту з обмеженням по відстані.

HVRP - Heterogeneous fleet VRP, задача маршрутизації транспорту з різнорідним парком.

FSMVRP - Fleet Size and Mix VRP, задача маршрутизації транспорту з необмеженим парком.

HVRPM - HVRP with Multiple use of vehicles, задача маршрутизації транспорту з множинним виїздами .

MDVRP Multiple Depots VRP, задача маршрутизації транспорту з декількома депо.

MDVRPI MDVRP with Inter-Depot Routes, задача маршрутизації транспорту з дозавантаженням.

OVRP - Open VRP, відкрита задача маршрутизації транспорту.

PVRP - Periodic delivery VRP - періодична задача маршрутизації транспорту.

PVRP-SC - PVRP with Service Choice, розширена версія PVRP з вибіркоvim обслуговуванням.

PDVRP - Pickup-and-delivery VRP, задача маршрутизації транспорту з вивезенням і доставкою.

SDVRP - Split-delivery VRP, задача маршрутизації з роздільною доставкою.

Stochastic VRP - VRP з довільними даними.

FVRP - Fuzzy VRP, нечітка задача маршрутизації.

DVRP - Dynamic VRP, динамічна задача маршрутизації транспорту.

VRPSF - Vehicle Routing Problem with Satellite Facilities, задача маршрутизації транспорту з супутніми складами.

GVRP - Green VRP).

VRPTW - VRP with Time Windows, VRP з часовими вікнами.

MDCVRPTW - Multi-Depot Capacitated Vehicle Routing Problem with Time Window, задача маршрутизації транспорту з множиною депо, обмеженням по вантажопідйомності і часовими вікнами.

ВСТУП

Задача маршрутизації транспортних засобів (VRP) полягає у визначенні оптимального набору маршрутів, які повинен виконувати парк транспортних засобів (ТЗ) для обслуговування певного набору клієнтів, і це одна з найважливіших і вивчених комбінаторних задач оптимізації.

З того часу, як Данциг і Рамзер ввели задачу в 1959 році, минуло понад 40 років [3]. Вони описали реальне застосування щодо доставки бензину до СТО та запропонували перше формулювання математичного програмування та алгоритмічний підхід. Кілька років потому, в 1964 році, Кларк і Райт запропонували ефективну жадібну евристику, яка вдосконалила підхід Данцига-Рамзера [4]. Після цих двох основних робіт було запропоновано сотні моделей та алгоритмів для оптимального та наближеного розв'язання різних версій VRP [11]. Значний вклад у дослідження задач VRP внесли також Фред Гловер, який вперше описав Табу пошук та ввів термін «метаетристика» [5]; Маріус Соломон, який досліджував підходи чутливих часових аспектів ланцюга постачань [6], часові обмеження при плануванні [7], маршрутизацію та управління складськими системами [8]; Марко Доріго, що запропонував метаетристичний “Мурашиний алгоритм” [9], [10].

Сучасні моделі VRP значно відрізняються від тих, що були запропоновані Данцигом і Рамзером та Кларком і Райтом, оскільки з кожним роком підвищується необхідність враховувати складнощі реального життя (час поїздки, що залежить від дорожнього трафіку; часові вікна для завантаження та розвантаження; попит, який динамічно змінюється з плином часу; та інші). Ці особливості істотно ускладнюють задачу: найбільші екземпляри VRP, які можуть бути послідовно вирішені за допомогою найбільш ефективних точних алгоритмів, містять близько 20 клієнтів, тоді як більші екземпляри можуть бути вирішені оптимально лише в окремих випадках. Реальні задачі значно більші за масштабами: тисячі клієнтів та

десятки складів, чисельні транспортні засоби, різноманітні обмеження, тож евристики та метаевристики значно більше застосовувані для даних об'ємів.

Швидкість обробки та ємність пам'яті сучасних комп'ютерів зростає в геометричній прогресії, що дозволяє вирішувати більші екземпляри VRP, кожного року пропонується нові різновиди задач VRP та розробляються нові алгоритми їх розв'язання, все це стимулює прогрес у дослідницькій галузі та розробку комерційного програмного забезпечення для VRP. На сьогодні програмне забезпечення VRP використовується тисячами компаній, зокрема Coca-Cola Enterprises та Anheuser-Busch Inbev, у багатьох галузях промисловості.

Інтерес до VRP обумовлений як практичною значущістю, так і значними труднощами:

- алгоритм розв'язання задачі змінюється при додаванні хоча б одного обмеження;
- під час практичної реалізації алгоритмів розв'язання VRP при впровадженні програмного забезпечення – нові умови потребують реінжинірингу.

Тож, важливість пошуку ефективних методів розв'язання VRP підкреслює необхідність впровадження нових підходів та модернізації вже відомих алгоритмів за певними критеріями.

1 АНАЛІЗ ЗАДАЧ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ ТА ІСНУЮЧИХ МЕТОДІВ РОЗВ'ЯЗАННЯ

1.1 Класифікація задач оптимізації маршрутів транспортних засобів

Дослідження Eksioglu, Vural, & Reisman [11] виявило 1638 журнальних статей з VRP як основною темою, опублікованою між 1959 та 2014 роком. Ряд книг та значна кількість наукових праць також внесли свій вклад в VRP літературу, яка існує сьогодні. За словами Eksioglu et al., література, у якій досліджується VRP, зростає експоненціально у розмірі 6% щороку [11]. Кількість публікацій на тему VRP зображено на рисунках 1.1 та 1.2.

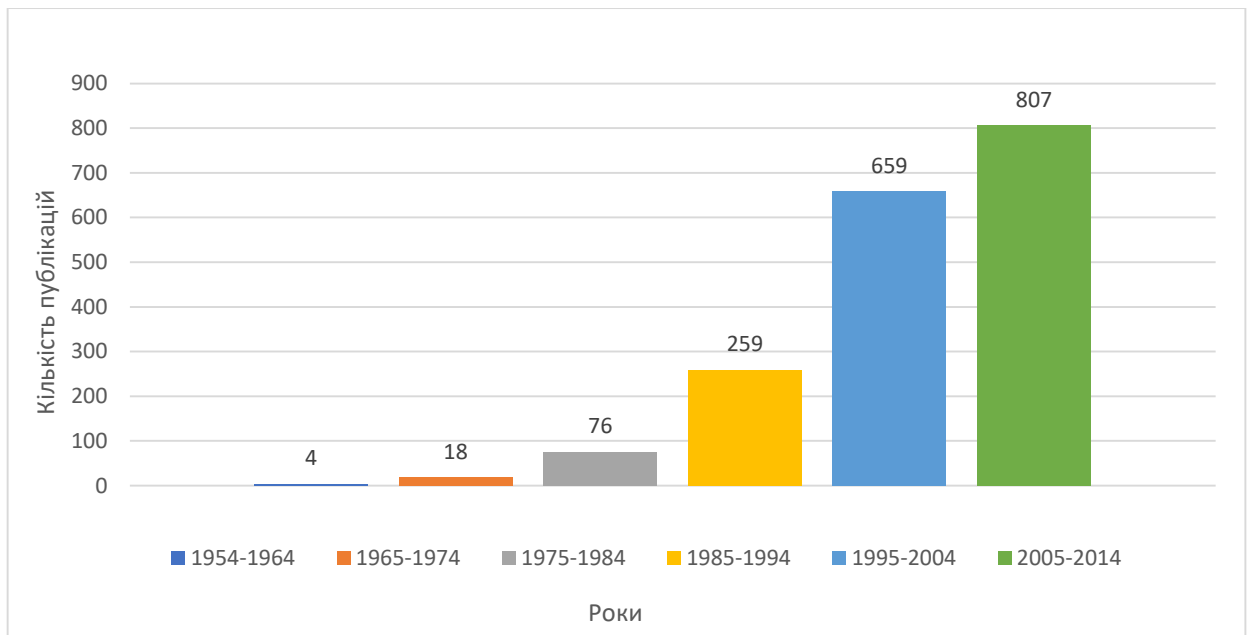


Рисунок 1.1 - Кількість публікацій на тему VRP, опублікованих у проміжок з 1954 по 2014 років

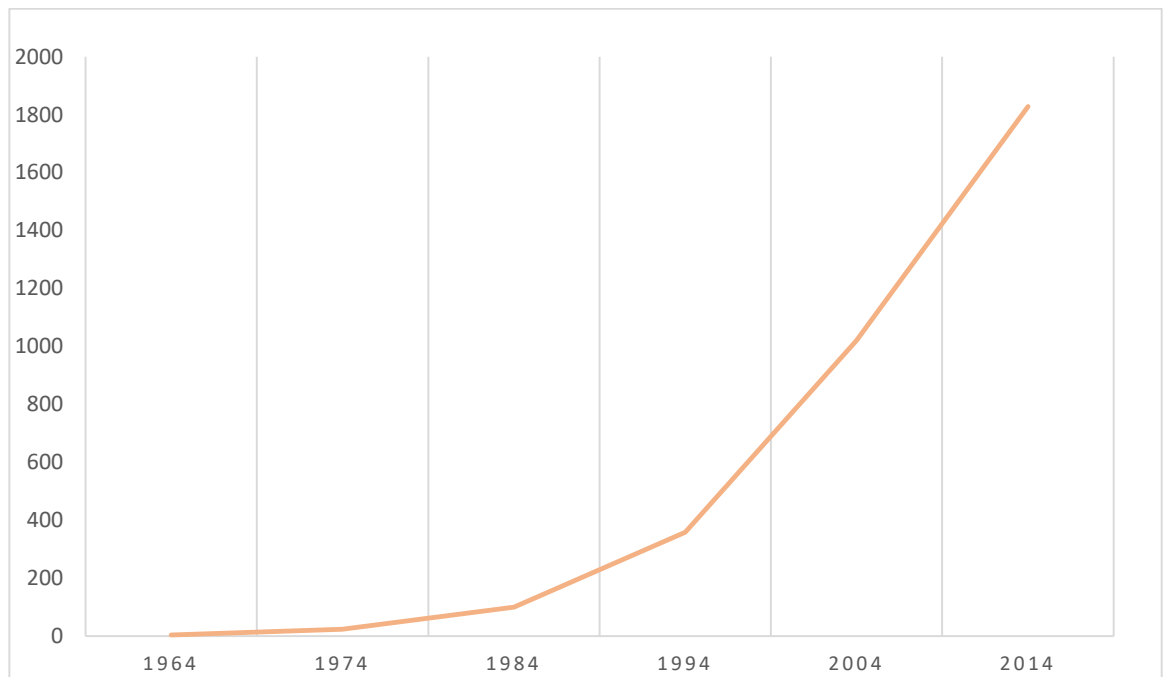


Рисунок 1.2 – Кумулятивний розподіл кількості публікацій на тему VRP за проміжок з 1954 по 2014 років

Найбільша кількість публікацій у області VRP була зроблена наступними дослідниками: Gilbert Laporte (66 разів), Michel Gendreau (42 рази), Jean-Yves Potvin (41 раз), Bruce L. Golden (34 рази), Marius M. Solomon (22 рази), а Christos D. Tarantilis (19 разів) [12].

Велика кількість публікацій на тему VRP задач зумовлена їх різноманітністю. Головна спільна риса у різних VRP – зосередженість у розгляді одного або декількох обмежень (характеристик) математичної моделі.

Було виділено наступні основні характеристики VRP [13]:

а) Кількість пунктів постачання:

- 1) один пункт постачання (наприклад, постачання або обслуговування різних клієнтів зі складу);
- 2) декілька пунктів постачання (наприклад, постачання вантажів від декількох виробників).

б) Кількість пунктів споживання:

- 1) один пункт споживання (наприклад, продукція з різних місць доставляється на склад або в пункт сертифікації);

2) декілька пунктів споживання.

в) Кількість критеріїв:

- 1) однокритеріальні (при оцінці оптимальності маршрутів враховується тільки один критерій - або відстань, або час, або інша характеристика);
- 2) багатокритеріальні (використовується кілька, як правило, конфліктуючих критеріїв).

г) Параметри доріг:

- 1) дорога характеризується одним параметром;
- 2) дорога характеризується кількома параметрами (відстань, пропускна здатність, габарити, мито чи інша фіксована плата, обмеження по швидкості і ін.).

д) Кількість вантажу:

- 1) вантаж відсутній (наприклад, виконання сервісних робіт);
- 2) кількість вантажу характеризується дійсним числом (безперервна задача):
 - одним;
 - декількома;
- 3) кількість вантажу характеризується цілим числом (дискретна задача):
 - одним;
 - декількома.

е) Види вантажу:

- 1) однотипний вантаж;
- 2) багатономенклатурний вантаж:
 - всі вантажі дозволені;
 - вантажі заборонені до перевезення у тих чи інших ТЗ;
 - для кожного ТЗ є обмеження по різним видам вантажів, що перевозяться;

- існують заборони на спільні перевезення вантажів деяких видів одним ТЗ.

ж) Кількість транспортних підприємств:

- 1) одне (автопарк зосереджений в одному місці);
- 2) кілька (автопарк територіально розподілений);
- 3) збігаються з пунктами постачання.

з) Кількість ТЗ:

- 1) одне ТЗ;
- 2) фіксована кількість доступних ТЗ;
- 3) необмежена кількість доступних ТЗ.

и) Однорідність парку ТЗ:

- 1) всі ТЗ однакові;
- 2) різні типи ТЗ мають відмінні характеристики:
 - відрізняються по одній характеристиці (наприклад, місткістю);
 - відрізняються за багатьма характеристиками (вантажопідйомність, вартість використання, швидкість та ін.).

к) Обмеження на місткість ТЗ:

- 1) без обмежень (кількістю вантажу можна знехтувати);
- 2) передбачені обмеження тільки по одному параметру;
- 3) передбачені обмеження за кількома параметрами.

л) Вимоги до перевезення:

- 1) при перевезенні вантажу з кожного пункту постачання в кожен пункт споживання може використовуватися один ТЗ незалежно від вантажу, що перевозиться;
- 2) при перевезенні вантажу з кожного пункту постачання один ТЗ може розвозити вантажі в різні пункти споживання;
- 3) в кожен пункт споживання вантажі з різних пунктів постачання доставляє один ТЗ;

- 4) ТЗ можуть збирати вантаж в різних пунктах виробництва і перевозити в різні пункти споживання.

м) Додаткові умови:

- 1) кожен вид вантажу повинен бути доставлений з деякого пункту виробництва в певний пункт споживання (наприклад, перевезення пасажирів);
- 2) не допускається порожній пробіг (крім початкового і кінцевого відрізків маршруту);
- 3) кожне ТЗ може зробити тільки одну поїздку;
- 4) деякі пункти споживання є також пунктами постачання:
 - вивіз з пунктів споживання здійснюється паралельно доставці;
 - вивіз з пунктів споживання здійснюється після доставки

н) Обмеження за часом:

- 1) максимальна тривалість маршрутів перевезень обмежена;
- 2) час безперервного перебування ТЗ в дорозі обмежена;
- 3) часові вікна:
 - доставка в пункти споживання здійснюється в зазначений інтервал часу;
 - вивіз з пунктів виробництва здійснюється в зазначений інтервал часу;
- 4) крос-докінг.

о) Обмеження по відстані:

- 1) загальна довжина всіх маршрутів обмежена;
- 2) максимальна пройдена ТЗ відстань обмежена.

п) Обмеження по пунктам виробництва і споживання:

- 1) деякі ТЗ не можуть відвідувати певні пункти;
- 2) пункти розподілені між ТЗ.

р) Обмеження збалансування:

- 1) збалансування тривалості маршрутів;
- 2) збалансування довжин маршрутів;
- 3) збалансування кількості обслуговуваних пунктів;
- 4) збалансування кількості вантажу, що доставляється.

с) Послідовність перевезень:

- 1) задана відповідність пунктів виробництва до пунктів споживання;
- 2) поділ фаз доставки і вивезення;
- 3) обмеження на послідовність завантаження-вивантаження вантажу – стековий принцип (LIFO);
- 4) задані інші вимоги до послідовності перевезень.

т) Періодичність:

- 1) планування маршрутів для одноразового виконання;
- 2) планування здійснюється на тривалий період;
- 3) планування регулярних маршрутів.

у) Фактори, що визначають вартість перевезення:

- 1) залежність вартості перевезення тільки від самого шляху;
- 2) залежність вартості перевезення від шляху і завантаження ТЗ:
 - пропорційність залежність;
 - складна функціональна залежність.

ф) Умови повернення ТЗ:

- 1) ТЗ повинні повертатися до вихідного депо;
- 2) ТЗ повинні завершувати маршрут у якому-небудь депо;
- 3) ТЗ можуть завершувати маршрут в будь-якому місці.

х) Топологія:

- 1) симетрична задача (характеристики шляху не залежать від вибору напрямку);
- 2) метрична (відстань описується евклідовою метрикою);
- 3) неметрична (відстань не описується евклідовою метрикою);

- 4) асиметрична задача (характеристики шляху залежать від вибору напрямку);
- 5) складна топологія.

ц) Стохастичність:

- 1) детерміновані дані;
- 2) випадкові дані (компоненти задачі мають випадкове поводження):
 - випадкові параметри доріг;
 - випадкові запити;
 - випадкові умови (обмеження);
 - випадкові характеристики ТЗ.

ч) Визначеність:

- 1) статична задача, тобто замовлення в процесі доставки не змінюються;
- 2) динамічна задача, тобто замовлення можуть додаватися або скасовуватися.

Класифікація пов'язана, головним чином, з автомобільним транспортом, але основна маса представлених характеристик справедлива також для інших видів. Наведене сімейство властивостей дозволяє сформувати множину задач транспортної логістики, в тому числі задач маршрутизації. Системне представлення класифікації характеристик VRP наведено на рисунку А.1.

1.2 Опис класичної задачі Vehicle Routing Problem (VRP)

Задача маршрутизації транспортних засобів (VRP) - загальна назва цілого класу задач, в яких для певної кількості географічно розподілених клієнтів повинен бути визначений набір маршрутів для парку транспортних засобів, що базуються в одному або декількох депо. Таким чином, задача полягає у виявленні оптимальної стратегії доставки, яка може мінімізувати

загальну вартість перевезень для транспортування продукту з джерел до клієнтів.

Метою VRP є забезпечення доставки набору клієнтів з відомими вимогами за транспортними маршрутами з мінімальною вартістю, що починається та закінчується у депо.

На рисунку 1.3 зображено задача VRP та один з можливих результатів.

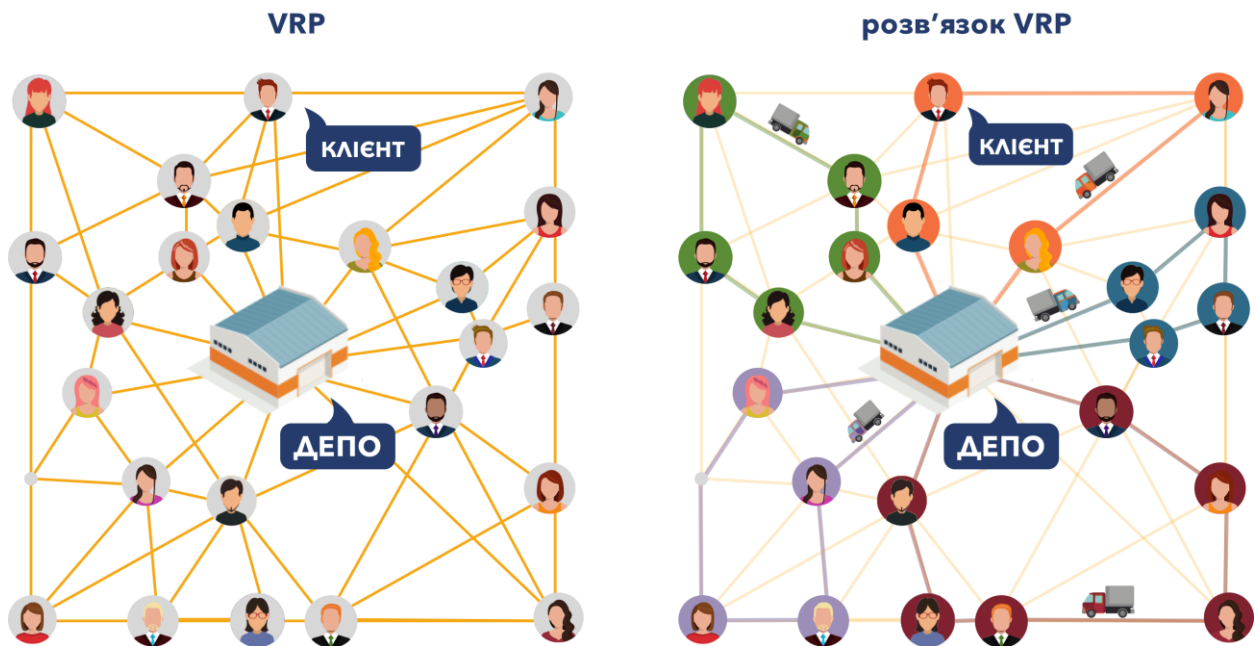


Рисунок 1.3 – Приклад VRP та один з варіантів її розв'язання

1.2.1 Постановка задачі

У класичній VRP визначено набір клієнтів (їх попит, місця розташування), розташування депо, кількість транспортних засобів, що доступні для транспортування продукції та їх місткість.

Зі складу товар повинен бути доставлений клієнтам в обсязі, який задовольняє їх потреби. Кожен ТЗ має почати і завершити маршрут у депо та здійснити доставку мінімум одному клієнту.

Задача полягає у визначенні розподілу клієнтів і транспортних засобів між маршрутами та послідовності обслуговування клієнтів на маршруті. Мета полягає у знаходженні розв'язку, мінімізуючого загальну вартість транспортування. Крім того, розв'язок має враховувати обмеження, що кожен клієнт обслуговується тільки один раз і в повному обсязі, а загальний попит на

кожен маршрут має бути в межах потужності транспортного засобу. Транспортні витрати визначаються як вартість переїзду від будь-якої точки в будь-яку іншу точку.

1.2.2 Математична постановка задачі

Класична задача оптимізації транспортних засобів відноситься до задач комбінаторної оптимізації [14]. Тож, формально класичну задачу VRP можна представити у вигляді графу $G = (N, A)$, де N – множина вершин, що відповідають набору клієнтів і позначаються $1, 2, \dots, n$, а вершина 0 відповідає центральному складу (депо) з якого починають і в якому закінчують свій маршрут всі ТЗ; A – множина дуг, які з'єднують відповідні вершини графа (клієнтів), якщо i – один клієнт, а j – інший, то дуга, що їх з'єднує позначається $(i, j) \in A$.

Кожен клієнт характеризується певним попитом d_i , $i \in N$. Кожній дузі ставиться у відповідність час t_{ij} – час переїзду ТЗ від клієнта i до клієнта j , цей час включає в себе час обслуговування клієнта i : s_i ; вартість переїзду ТЗ з i в j складає c_{ij} .

Індексом $k \in V$ далі буде позначатися відповідний ТЗ (де V – кількість ідентичних ТЗ вантажопідйомністю q). Змінні x_{ijk} приймають значення $\{0, 1\}$, де 1 означає, що ТЗ рухається від вершини i до вершині j , 0 – зворотне.

Виходячи з наведених вище позначень, математична модель задачі VRP буде наступною (обмеження 1.2 – 1.7), де потрібно мінімізувати цільову функцію (1.1):

$$\sum_{k \in V} \sum_{(i,j) \in A} c_{ij} x_{ijk} \rightarrow \min, \quad (1.1)$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \text{для } \forall i \in N, \quad (1.2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq q, \text{ для } \forall k \in V, \quad (1.3)$$

$$\sum_{j \in N} x_{0jk} = 1, \text{ для } \forall k \in V, \quad (1.4)$$

$$\sum_{i \in N}^K x_{ihk} - \sum_{j \in N}^N x_{hjk} = 0, \text{ для } \forall h \in N, \forall k \in V, \quad (1.5)$$

$$\sum_{j \in N} x_{i,n+1,k} = 1, \text{ для } \forall k \in V, \quad (1.6)$$

$$x_{ijk} \in \{0,1\}, \forall (i,j) \in A, \forall k \in V. \quad (1.7)$$

Загальна вартість всіх маршрутів всіх ТЗ визначається цільовою функцією (1.1). Обмеження (1.2) має на увазі, що обслуговування кожного клієнту відбувається одним ТЗ і лише один раз. Обмеження (1.3) визначає, що транспортний засіб не може обслужити більше клієнтів, ніж дозволяє його вантажопідйомність. Обмеженням (1.4) визначається, що кожен ТЗ виїжджає з депо лише один раз. Обмеження (1.5) означає, що ТЗ може виїхати з вершини h , тільки якщо він в'їхав у цю вершину. Обмеження (1.6) показує, що всі ТЗ повертаються в депо, та лише один раз. Це обмеження впливає з обмежень (1.4) і (1.5).

Розв'язком задачі є:

- розбиття множини вершин на підмножини (маршрути);
- задання порядку обходу на кожній підмножині.

Розв'язок є прийнятним (допустимим), якщо всі маршрути задовольняють обмеженням задачі. VRP є узагальненням задачі комівояжера (Travelling salesman problem, TSP), отже, є NP-важкою. TSP – це VRP з одним транспортним засобом без обмежень, немає депо, клієнти без попиту.

1.3 Огляд різновидів задач маршрутизації транспортних засобів

Різноманітність характеристик створило величезний набір відокремлених гілок дослідницьких ліній VRP. Кожна дослідницька гілка була визначена аббревіатурою розглянутих обмежень або атрибутів задачі оптимізації. Багато з цих окремих галузей були об'єднанні для створення нових «основних» гілок. Розглянемо найпоширеніші у науковій літературі VRP:

Asymmetric cost matrix VRP (AVRP) – Асиметрична задача маршрутизації. У загальному випадку, асиметрична задача відрізняється від симетричною тим, що вона моделюється орієнтованим графом. Вартість переходу від i -го клієнта до j -го клієнта відрізняється від переходу від j -го клієнту до i -го клієнта. [14]

Capacitated Vehicle Routing Problem (CVRP) – задача маршрутизації транспортних засобів з урахуванням завантаженості транспортних засобів. У випадку CVRP до класичної задачі VRP додаються обмеження на місця споживання, а потужність ТЗ стає обмеженою [16].

Обмеження $\sum_{i \in N} d_i \sum_{j \in N} x_{ijk} \leq q$, для $\forall k \in V$ – обсяг вантажів на кожному маршруті не повинен перевищувати заданої величини q (однакової для всіх ТЗ) [17], [18].

Метою є мінімізувати парк ТЗ, задіяних у доставці, та сукупний час.

Distance-Constrained VRP (DCVRP) - задача маршрутизації транспорту з обмеженням по відстані. Загальна довжина дуг у маршруті не може перевищувати максимальну довжину маршруту. Це обмеження може замінити обмеження ємності або доповнити його [22].

Метою є врахування обмеженої ємності паливного баку.

Heterogeneous fleet VRP (HVRP) - задача маршрутизації транспорту з різномірним парком. Використовуються різні види транспортних засобів одночасно, маршрути розробляються відповідно до потужності кожного ТЗ.

Коли кількість транспортних засобів необмежена, задача називається ***Fleet Size and Mix VRP (FSMVRP)***. Якщо певний тип транспортного засобу не може досягти деяких клієнтів з будь-якої причини, то задача стає ***Site-Dependent VRP (SVRP)***. Крім того, якщо транспортному засобу дозволяється виконувати більше однієї поїздки, то ми вирішуємо ***HVRP with Multiple use of vehicles (HVRPM)*** [15].

Multiple Depots VRP (MDVRP) - задача маршрутизації транспорту з декількома депо. Замість одного центрального складу маємо кілька складів, з яких можливе обслуговування клієнтів. Таким чином, кожним ТЗ закріплюється депо, в якому він починає та закінчує свій маршрут, різні ТЗ можуть бути прикріплені до різних депо. Метою є мінімізувати парк транспорту та загальний час перевезень. [21 – 23] Розширена задача ***MDVRP with Inter-Depot Routes (MDVRPI)*** передбачає дозавантаження ТЗ в будь-якому депо по шляху проходження [24-26].

Open VRP (OVRP) – відкрита задача маршрутизації транспорту. Заплановані маршрути можуть бути закінчені не у депо. Таким чином, віддаленість останнього клієнта від депо не впливає на загальну вартість розв'язку. Застосовується коли підприємство не має власного парку ТЗ, а користується кур'єрськими послугами [27].

Periodic delivery VRP (PVRP) - періодична задача маршрутизації транспорту. Оптимізація здійснюється на декілька днів (в той час, як зазвичай робиться щоденно). За обраний проміжок часу клієнти можуть обслуговуватись декілька разів з різною періодичністю [28]. Існує також розширена версія ***PVRP з вибіркоким обслуговуванням - PVRP with Service Choice (PVRP-SC)***, де частота відвідування клієнтів встановлюється по ходу розв'язання задачі. Це дозволяє отримати більш оптимальні маршрути і дає переваги при обслуговуванні клієнтів [29].

Pickup-and-delivery VRP (PDVRP) - задача маршрутизації транспорту з вивезенням і доставкою. Кожен клієнт характеризується двома величинами - 1 – потрібно доставити, 2 – потрібно вивезти від клієнта та доставити у депо.

Клієнти можуть як отримувати, так і відправляти товари. Тобто деяка кількість товарів повертається до депо від споживачів.. У доповнення до обмеження, що загальне навантаження і загальні доставки по маршруту не можуть перевищувати ємність транспортного засобу, також ця ємність не має бути перевищена в будь-якій точці маршруту [15].

Одним з варіантів задачі *PDVRP* є, коли вивезений товар не повертається до депо, але має бути доставлений іншому клієнту, наприклад, транспортування людей. У деяких випадках транспортні засоби повинні вивозити та доставляти товари до одних клієнтів за один візит (***Simultaneous Pickup-and-delivery VRP***) – наприклад, нові та повернуті пляшки. Також важливий варіант *VRP - 1 – M – 1* (“one-to-many-to-one”), всі товари, які мають бути доставлені – забираються лише з депо, всі товари, що мають бути вивезені - доставляються лише в депо [30]. Також варіантами *PDVRP* є ***Delivery-First, Pickup-Second VRP або VRP with Backhauls (VRPB)*** - всі товари повинні бути доставлені клієнтам, перш ніж відбудеться будь-яке вивезення від клієнтів-постачальників [31] та ***Mixed Pickup and Delivery VRP (VRMPD)***- вивезення і доставка здійснюються в будь-якій послідовності [32].

Варто окремо відзначити варіант *VRP* зі стековим принципом «останнім прийшов - першим вийшов» - ***VRP with LIFO*** . Аналогічна *VRPPD* за винятком додаткового обмеження: розвантаження [33] ТЗ може здійснюватися тільки в зворотному порядку завантаження, тобто першим завжди вивантажується останній завантажений товар. Зазвичай використовується при мультиноменклатурному вантажі для скорочення часу завантаження і розвантаження ТЗ, тому що відпадає необхідність переставляти товар. Однак в літературі добре вивчений тільки випадок з одним ТЗ - ***TSP with Pickup and Delivery and LIFO (TSPPDL)***.

Split-delivery VRP (SDVRP) - задача маршрутизації з роздільною доставкою. Клієнт може обслуговуватися різними транспортними засобами одночасно, якщо це дозволить знизити загальну вартість. Це послаблення

базової задачі є важливою у випадках, коли замовлення клієнта перевищує ємність транспортного засобу.

Stochastic VRP - *VRP з довільними даними*. Деякі компоненти задачі (один або декілька) можуть мати довільну поведінку – наприклад: час поїздки (**VRP with Stochastic Travel Time (VRPST)**), множина клієнтів (**VRP with Stochastic Clients (VRPSC)**) [34], час обслуговування клієнта (**VRP with Stochastic Service Time (VRPSSST)**), попит клієнта (**VRP with Stochastic Demands (VRPSD)**) [34]. Розв’язання SVRP зазвичай відбувається в два підходи. Перший етап дає розв’язок не враховуючи випадкові змінні. Наступний етап полягає в корекції наявного розв’язку після розрахунку випадкових значень.

Fuzzy VRP (FVRP) - *нечітка задача маршрутизації*. В практичній площині дуже важко отримати чіткі значення параметрів. Використання методів теорії нечітких множин дозволяє успішно моделювати задачі, що містять елементи невизначеності і суб'єктивності.

Dynamic VRP (DVRP) - *динамічна задача маршрутизації транспорту*. Передбачається зміна деяких параметрів в процесі розв’язання задачі та динамічний перерахунок маршруту з урахуванням нової інформації. До таких зазвичай відносяться поломки ТЗ, дорожні затори, нові замовлення, раптові виклики і т.д. Найчастіше розглядається випадок, коли нові клієнти можуть з'являтися протягом дня, тобто після того, як ТЗ покине депо [27]. Таким чином, по мірі обслуговування клієнтів умови задачі змінюються і необхідно динамічно перераховувати маршрути з урахуванням нової інформації, що надходить. Як правило, заздалегідь відома статистика появи нових замовлень, на основі якої створюються імітаційні моделі для розробки і випробування алгоритмів.

Vehicle Routing Problem with Satellite Facilities (VRPSF) - *задача маршрутизації транспорту з супутніми складами*. Особливістю умов задачі є, що деякі ТЗ можуть дозавантажуватись на маршруті не повертаючись в депо. Типовим є випадок задачі *Vehicle Routing Problem with Satellite Facilities*,

коли безліч споживачів очікують регулярних поставок від одного центрального постачальника [30].

Green VRP (GVRP). Цей варіант VRP спрямований на включення різних екологічних проблем у процес оптимізації, наприклад викиди парникових газів, забруднення, відходи та, наслідки використання „екологічніших” конфігурацій автопарка ін.

VRP with Time Windows (VRPTW) – *VRP з часовими вікнами*. Кожен клієнт пов'язаний з часовим інтервалом і може обслуговуватися лише протягом цього інтервалу. Задіяні параметри: рамки часового вікна (початок та кінець), тривалість обслуговування, тривалість переїзду між пунктами, час очікування. Якщо часове вікно не почалось до приїзду ТЗ, ТЗ очікує відкриття пункту обслуговування. Якщо часове вікно закінчилось до приїзду ТЗ, пункт вже не може бути обслуговуваний [19]. Також можна розглянути набір часових вікон для кожного клієнта (**VRP with Multiple Time-Windows**). Також ці часові вікна можуть бути гнучкими залежно від деяких додаткових витрат (**VRP with Soft Time-Windows**) [20].

В останнє десятиліття з'явився новий варіант VRP – **Consistent VRP (ConVRP)** – *Послідовний VRP*, в якому пріоритетом стає довіра між розподільчими компаніями та замовниками. Тому компанії будують послідовні маршрути протягом певного періоду, що зменшує кількість водіїв, з якими доводиться стикатися кожному клієнту, а отже, і тертя між ними.

Перелік існуючих типів задач маршрутизації можливо представити у вигляді ієрархічної системи (Рисунок А.2). З неї видно, що в основі більшості різновидів VRP передбачається базове обмеження вантажопідйомності (CVRP), яке в явному вигляді при описі може не вказуватися. Очевидно, за рахунок сполучень розглянутих вище різновидів VRP можна отримати безліч варіантів її постановок. Найцікавішими для досліджень вважаються конкретні комбінації умов і обмежень для визначення нових областей застосування VRP, що дозволяють наблизитися до реальних умов.

Однак, не зважаючи на те, що інтерес до задач класу VRP зростає з кожним роком, все ж таки залишається багато проблем, що пов'язано з:

- кількістю обмежень, які можуть бути застосовані до однієї задачі, бо при додаванні хоча б одного нового обмеження змінюється математична модель задачі, а як наслідок, і алгоритм її розв'язання;
- практичною реалізацією алгоритмів розв'язання VRP задач, бо на практиці при впровадженні ПЗ постають нові умови, які вимагають реінжиніринг.

Таким чином, задача маршрутизації транспортних засобів є важливою на сьогоднішній день та потребує нового підходу до її розв'язання, яке дозволить використовувати вже відомі алгоритми розв'язання, але буде ефективнішим за певним критерієм.

1.4 Аналіз класичних алгоритмів розв'язання задач маршрутизації транспорту

VRP – це задача комбінаторної оптимізації, в якій число допустимих маршрутів експоненційно зростає при збільшенні числа клієнтів, і належить до класу складності NP. Для розв'язання таких задач розмірністю більше 14 не можуть ефективно використовуватись точні методи, оскільки перебір можливих розв'язків для знаходження оптимуму займає колосального часу обчислень навіть при відносно невеликій розмірності задачі. До точних методів відносяться такі методи, як метод гілок та меж, метод гілок та відсікань, динамічне програмування. Вони є розвитком методу повного перебору з відсівом підмножин допустимих розв'язків, що завідомо не містять оптимальних розв'язків. Однак час їх обчислень все одно зростає занадто швидко, а в гіршому випадку - аналогічно повному перебору.

Для реальних прикладів задач застосовуються наближені алгоритми [39]. Існує велика кількість методів розв'язання VRP, значна частина яких –

евристичні та метаевристичні методи, що пропонують кращий баланс між якістю розв'язку та обчислювальним часом. Метаевристика слугує трьома основним цілям: прискорити розв'язання задачі, опрацьовувати більше вузлів та отримувати надійніші алгоритми. Метаевристика забезпечує прийнятні розв'язки в адекватні терміни для розв'язання важких та комплексних задач.

Побудова початкового розв'язку полягає у створенні одного або декількох допустимих розв'язків, котрі будуть стартовою точкою при наступному етапі покращення. У звичайному випадку використовується евристичний процес побудови, який починається з порожнього розв'язку та поступово додає компоненти розв'язку доки не буде отримано допустимий розв'язок. При цьому найчастіше застосовується жадібний алгоритм з використанням різноманітних евристичних функцій. Процес покращення розв'язку намагається модифікувати початковий допустимий розв'язок, базуючись на визначеній метриці, доки не буде отримано оптимального розв'язку або не закінчиться час, відведений на роботу алгоритму. Використовується поняття “множина розв'язків” - множина допустимих розв'язків/станів, котрі досяжні з поточного стану. Для знаходження кращого розв'язку метаевристичні алгоритми змінюють поточний стан одним зі станів з множини розв'язків. Цей процес відповідає локальному пошуку в просторі станів. Локальний пошук використовує так звані локальні операції, які змінюють один або декілька маршрутів намагаючись отримати кращий (дешевший) розв'язок.

На рисунку А.3 зображена класифікація найпоширеніших алгоритмів розв'язання VRP.

1.4.1 Точні методи

Точні методи гарантують отримання оптимальних розв'язків. Однак практичне застосування таких алгоритмів можливе лише для задач невеликої розмірності [34]. Точні підходи засновані на точних математичних алгоритмах. Такі методи розв'язання обраного класу задач засновані на повному переборі всіх можливих розв'язків, і є неефективними при

розв'язанні задач великої розмірності через їх великих часових витрат (час розв'язання експоненціально залежить від розмірності задачі).

Метод гілок та меж (Branch-and-Bound)

Метод гілок та меж (МГМ) - це метод, заснований на розумному переборі можливих розв'язків комбінаторної задачі оптимізації. Принцип полягає у розподілі простору розв'язків на непересічні підмножини, які представлені вузлами дерева розгалуження. Потім алгоритм досліджує гілки дерева відповідно до стратегії маршруту [40]. Щоб уникнути вивчення всього дерева, перед створенням нового вузла в дереві алгоритм оцінює вузол, порівнюючи значення найкращого можливого розв'язку, яке можна знайти у відповідному піддереві, з найкращим поточним розв'язком [13]. Якщо кращий розв'язок не може належати піддереву, укоріненому на розглянутому вузлі, піддерево відкидається. В іншому випадку процес розвідки триває. Багато комбінаторних задач оптимізації можна вирішити, більш-менш ефективно, використовуючи стратегію МГМ.

Метод гілок та відсікань (Branch-and-Cut)

Головна ідея методу гілок та відсікань (МГВ) полягає у зменшенні пошукового простору кандидатів шляхом додавання нових обмежень. Загалом, алгоритм МГВ працює аналогічно до МГМ, але з додатковим кроком розбиття обмежень [41].

Динамічне програмування (Dynamic Programming)

Динамічне програмування (ДП) відноситься до процесу спрощення складної задачі шляхом її рекурсивного розподілу на менші підзадачі. Задачі, які вирішуються за допомогою динамічного програмування, як правило, можуть бути розділені на етапи з прийняттям рішення на кожному етапі. Кожен етап також має ряд станів, пов'язаних з ним [35]. Рішення на одному етапі перетворює поточний стан на стан наступного етапу. Рішення про перехід до наступного стану залежить лише від поточного стану, а не від попередніх станів чи рішень. Динамічне програмування top-down засноване на

збереженні результатів певних обчислень для подальшого використання [36]. Динамічне програмування bottom-up рекурсивно перетворює складне обчислення в серію більш простих обчислень.

1.4.2 Евристичні методи

Евристичні методи – сукупність прийомів в пошуку розв’язання задачі, які дозволяють обмежити перебір, тобто проводиться відносно обмежений пошук у просторі розв’язків, і, як правило, знаходять розв’язок в межах прийняттого часу. Недолік даних методів в тому, що вони є приблизними, розв’язок, отриманий даними методами, може бути далеким від оптимального. Перевага в тому, що вони дозволяють знайти прийнятний розв’язок NP-повних задач великої розмірності за прийнятний час [37]. Евристичні методи часто протиставляються формальним методам розв’язання, які спираються на точні математичні алгоритми.

Евристичні алгоритми поділяються на три основні категорії.

Конструктивні алгоритми. Виконують поступову побудову розв’язку, відстежуючи зростання його вартості, але не мають фази подальшого поліпшення: Nearest Neighbour, Greedy, Savings, Insertion.

Двофазні (кластерні) алгоритми. Задача розбивається на дві операції - угруповання вершин для кожного майбутнього маршруту (кластеризація) і розв’язання TSP для кожної отриманої групи: (алгоритм Фішера і Джайкумара, алгоритм замітання (Sweep Algorithm), алгоритм пелюсток (Petal Algorithm), алгоритм Османа). У двофазних алгоритмах можлива наявність зворотного зв’язку між етапами розв’язку. (Petal, Sweep, Cluster first-route second, Rout first – cluster second).

Покращуючі алгоритми. Спочатку ведеться пошук деякого розв’язку, а потім робляться спроби обміну вершин (ребер) всередині кожного маршруту або між маршрутами: (евристики поліпшення багатьох маршрутів). (Inner route, Intra route, k-opt, 2-opt, 3-opt).

1.4.3 Метаевристичні методи

Термін «метаевристика», що походить від грецького прийменника «meta» (в значенні «вищий рівень») і «heuristic» (від грец. Εὐρίσκειν, «шукати»), був вперше введений Фредом Гловером [6] в 1986 р для позначення алгоритмічних схем вищого рівня, спрямованих на ефективне вивчення простору пошуку складних оптимізаційних задач [40]. Метаевристики описують великий (а також основний) підрозділ в стохастичній оптимізації - великому класі алгоритмів і методів, які в тій чи іншій мірі використовується випадковість для пошуку оптимального (близького до оптимального або субоптимального) розв'язку складних задач.

Метаевристики є сучасним потужним і надзвичайно популярним класом оптимізаційних методів, що дозволяють знаходити розв'язки для широкого кола задач. Сила метаевристик полягає в їх здатності вирішувати складні задачі без знання простору пошуку, саме тому ці методи дають можливість вирішувати складні задачі оптимізації [13]. Метаевристичні методи спираються на досконалому вивченні найперспективніших частин простору розв'язків. Такі розв'язки показують себе краще ніж отримані за допомогою класичних евристик. Особливість метаевристичних алгоритмів в тому, що вони не дають точного опису порядку дій для розв'язання задач, і кожен з них додатково конкретизується шляхом підбору значень керуючих параметрів.

Іншими словами, метаевристика - це метод розв'язання обчислювальних задач шляхом комбінування існуючих процедур з відкритим інтерфейсом і закритою реалізацією, яке призводить до максимально ефективного розв'язку. Метаевристичний підхід зазвичай застосовується для розв'язання задач, що не мають задовільного специфічного для задачі алгоритму, або в тому випадку, коли немає практичної необхідності реалізовувати такий алгоритм. Найбільш часто метаевристики використовуються у розв'язку задач комбінаторної оптимізації, але також вони можуть застосовуватися для будь-якої іншої задачі, яку можна звести до розв'язання логічних рівнянь. На відміну від традиційних алгоритмів оптимізації і ітераційних методів, метаевристики не

гарантують, що в глобальному масштабі оптимальний розв'язок може бути знайдено для деякого класу задач в силу стохастичності. Однак під час пошуку по великим наборам допустимих розв'язків метаевристики часто дозволяють знайти вдалий розв'язок з меншими обчислювальними і часовими витратами.

Виділяють наступні властивості метаевристик [43]:

- метаевристика є стратегією, яка управляє процесом пошуку;
- метою метаевристики є ефективне дослідження простору пошуку для знаходження (суб) оптимального розв'язку;
- методи, які реалізуються метаевристичним алгоритмом, варіюються від простого локального пошуку до складного процесу навчання;
- метаевристичний алгоритм є наближеним і зазвичай недетермінованим;
- у метаевристики закладається механізм, що запобігає застряганням в локальному оптимумі;
- основні положення метаевристик допускають абстрактний опис;
- метаевристика може використовувати проблемно-орієнтоване знання в формі евристик, керованих високорівневою стратегією;
- передові метаевристики використовують досвід, накопичений в процесі пошуку і представлений у вигляді пам'яті, для управління пошуком.

Основна частина літератури по темі метаевристик має експериментальний характер, описуючи емпіричні результати, засновані на імітаційному моделюванні і програмному експерименті з алгоритмами. Однак існують також деякі формальні теоретичні результати, що обґрунтовують можливість знаходження глобального оптимуму [43].

Практично всі метаевристики запропоновані на основі ідей, що з'явилися під час спостереження процесів живої і неживої природи.

Алгоритм імітаційного відпалу (Simulated annealing)

Імітаційний відпал (SA) - це добре відомий мета-евристичний метод пошуку, який успішно застосовується для розв'язання багатьох комбінаторних задач оптимізації. Це hill-climbing алгоритм (сходження на пагорб) з доданою можливістю вирватися з місцевих оптимумів у просторі пошуку. Однак, хоча він дає чудові розв'язки, він дуже повільний у порівнянні з простою процедурою сходження на пагорб.

Термін імітаційний відпал базується на фізичному процесі відпалу твердих речовин, коли відбувається кристалізація речовини, де ми намагаємося мінімізувати енергію системи, використовуючи повільне охолодження, поки атоми не досягнуть стабільного стану. Техніка повільного охолодження дозволяє атомам металу вибудовуватися і утворювати правильну кристалічну структуру, що має високу щільність і низьку енергію, перехід атома з одного осередку в інший відбувається з певною ймовірністю, причому ймовірність зменшується з пониженням температури. Початкова температура і швидкість зниження температури називається графіком відпалу.

Вирішуючи задачу комбінаторної оптимізації (CO) з використанням SA, ми починаємо з певного можливого розв'язку задачі. Потім ми намагаємося оптимізувати цей розв'язок, використовуючи метод, аналогічний відпалу твердих речовин. Сусід цього розв'язку створюється за допомогою відповідного методу, і розраховується вартість (або допустимість) нового розв'язку. Якщо новий розв'язок є кращим за поточний з точки зору зменшення витрат (або підвищення придатності), новий розв'язок приймається. Якщо новий розв'язок не є кращим від поточного розв'язку, новий розв'язок приймається з певною ймовірністю. Ймовірність прийняття зазвичай встановлюється на $\exp^{(-\Delta / T)}$, де Δ - це зміна вартості між старим та новим розв'язком, а T - поточна температура. Таким чином, ймовірність експоненційно зменшується в міру погіршення руху.

Процедура SA має менше шансів застрягти в місцевому оптимумі порівняно з простим hill-climbing, оскільки погані шляхи все ще мають шанс

бути здійсненими. Спочатку температуру відпалу вибирають високою, так що ймовірність прийняття також буде високою, і приймаються майже всі нові розв'язки. Потім температуру поступово знижують, так що ймовірність прийняття низькоякісних розв'язків буде дуже малою, і алгоритм працює більш-менш, як hill-climbing, тобто високі температури дозволяють краще досліджувати пошуковий простір, тоді як нижчі температури дозволяють регулювання кращого розв'язку. Процес повторюють до тих пір, поки температура не наблизиться до нуля або подальшого поліпшення не вдасться досягти. Це аналогічно атомам твердої речовини, що досягають кристалізованого стану. На рисунку 1.5 показано, як випадкові рухи вгору можуть дозволити алгоритму SA уникати локальних оптимумів у задачі мінімізації.

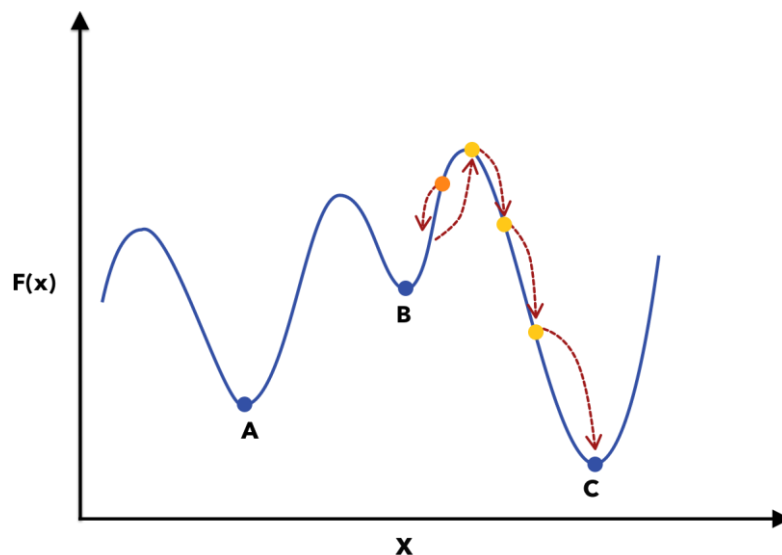


Рисунок 1.5 - Імітаційний відпал: випадкові підйоми вгору та вихід з місцевих оптимумів

Застосування SA до задач СО також вимагає вибору як загальних параметрів SA, так і конкретних рішень. Вибір параметрів SA має вирішальне значення для роботи алгоритму. Такими параметрами є: значення початкової температури T , температурна функція, яка визначає, як температура буде змінюватися з часом, кількість ітерацій $N(t)$, що виконуються при кожній температурі, та критерій зупинки для припинення алгоритму.

Генетичний алгоритм (Genetic Algorithm)

Генетичні алгоритми (ГА) є широко відомим типом метаевристичних алгоритмів, яким на сьогодні приділяється підвищена увага у всьому світі. Генетичні алгоритми - це еволюційні пошукові алгоритми, що використовують механіку природного відбору та природню генетику для пошуку розв'язків задач.

ГА еволюціонує популяцію особин, кодованих як хромосоми, шляхом створення нових поколінь нащадків за допомогою ітеративного процесу до тих пір, поки деякі критерії збіжності не будуть виконані. Такі критерії можуть, наприклад, стосуватися максимальної кількості поколінь, зближення до однорідної популяції, складеної з подібних особин, або отримання оптимального розв'язку. Потім декодується найкраща утворена хромосома, забезпечуючи відповідний розв'язок.

Генетичні алгоритми працюють із популяцією розв'язків-особин, а не лише з одним розв'язком, тому вони здійснюють багатосторонній пошук одночасно. Кожна особина представляє потенційний розв'язок задачі.

Створення особини нового покоління включає три основні етапи.

Етап відбору складається з випадкового вибору двох батьків з популяції для спаровування. Ймовірність обрати представника популяції, як правило, пропорційна його придатності для підкреслення генетичної якості при збереженні генетичного різноманіття. Тут придатність відноситься до міри прибутку, корисності чи блага, яку слід максимізувати під час вивчення простору розв'язків.

Етап рекомбінації або розмноження використовує гени вибраних батьків для отримання потомства, яке сформує наступне покоління.

Етап мутації полягає у випадковій модифікації деяких генів окремої особини одночасно для подальшого вивчення простору розв'язків та забезпечення або збереження генетичного різноманіття. Виникнення мутації, як правило, пов'язане з низькою ймовірністю. Отже, процедура мутації застосовується для забезпечення різноманітності популяції.

Нове покоління створюється шляхом повторення процесів відбору, розмноження та мутації, поки всі хромосоми в новій популяції не замінять хромосоми зі старої. Тому для забезпечення ефективного пошуку необхідний належний баланс між генетичною якістю та різноманітністю серед популяції.

Меметичні алгоритми (МА) - це «вдосконалена» версія GA, оскільки для кожного нащадка застосовується метод місцевого вдосконалення. Обидва алгоритми належать до вищої категорії еволюційних алгоритмів.

Табу пошук (Tabu Search)

Основна концепція Табу пошуку (TS) є одним із найчастіше застосовуваних алгоритмів і залежить від принципу продовження пошуку, навіть якщо був досягнутий локальний оптимум. TS досліджує простір розв'язків, рухаючись на кожній ітерації від розв'язку до найкращого розв'язку в підмножині його сусідства. На відміну від класичних методів спаду, поточний розв'язок може погіршуватися від однієї ітерації до іншої. Таким чином, щоб уникнути циклічності, розв'язки, що мають деякі атрибути нещодавно досліджених розв'язків, тимчасово оголошуються табу або забороняються. Тривалість, коли атрибут залишається табу, називається його табу-мандатом, і вона може змінюватися протягом різних інтервалів часу. Статус табу можна замінити, якщо дотримані певні умови; це називається критерієм прагнення, і це трапляється, наприклад, коли розв'язок табу кращий, ніж будь-який раніше знайдений розв'язок.

В результаті тенденція відхилитися від наміченого шляху, може як негативно впливати на пошук, так і позитивно. Метод Табу діє таким чином, за винятком, що нові шляхи не вибираються випадково. Натомість пошук Табу здійснюється згідно з припущенням, що немає сенсу в прийнятті нового рішення, якщо воно не уникає вже дослідженого шляху. Це гарантує, що нові регіони простору розв'язків задач будуть досліджені з метою уникнення локальних мінімумів та остаточного пошуку бажаного розв'язку.

Початковий розв'язок, як правило, будується з якоюсь найдешевшою евристикою. Після створення початкового розв'язку робиться спроба

вдосконалити його за допомогою локального пошуку з однією або кількома структурами сусідства та найкращою стратегією.

Пошук у змінних околах (Variable Neighbourhood Search)

Пошук у змінних околах (VNS) - порівняно «молода» мета-евристика, запропонована Гансеном та Младеновичем [47]. Основна ідея заснована на дослідженні простору пошуку шляхом поступового збільшення розміру окола, в межах якого генерується новий розв'язок, до досягнення певної умови зупинки. Крім того, щоразу, коли в поточному околі створюється новий розв'язок, до цього розв'язку застосовується локальний пошук, щоб оптимізувати його до прийняття рішення про заміну.

Етапи евристики пошуку у змінних околах включають фазу вдосконалення, яка використовується для можливого вдосконалення даного розв'язку, і так звану фазу струшування, яка використовується з метою розв'язання локальних мінімумів. Фаза вдосконалення та процедура струшування разом із етапом зміни околиць виконуються по черзі до виконання заздалегідь визначеного критерію зупинки. Іншими словами, наступні три кроки повторюються, поки не буде виконано якийсь критерій зупинки:

- а) процедура струшування;
- б) процедура вдосконалення;
- в) крок зміни сусідства.

Мета-евристика VNS пропонує безліч привабливих функцій, які спонукали дослідників використовувати її для розв'язання задач СО. По-перше, алгоритм простий, легкий для розуміння та реалізації та, значною мірою, незалежний від параметрів. По-друге, добре відомі переміщення по околицям та методи локального пошуку можуть бути легко інтегровані кількома способами та застосовані в рамках VNS. По-третє, це надійна техніка, яка може бути використана в різних типах задач, майже не змінюючи базової структури алгоритму. Нарешті, його можна легко гібридизувати з іншими

евристичними та мета-евристичними підходами для розв'язання важких задач оптимізації.

Алгоритм мурашиних колоній (Ant Colony Optimization)

Оптимізація мурашиними колоніями (ACO) - це метоевристичний метод, натхненний поведінкою справжніх мурах.

При розв'язанні комбінаторної задачі оптимізації за допомогою ACO проблема може бути візуалізована як пошук найкоротшого шляху у зваженому графіку. Штучні мурахи (програмні агенти) співпрацюють, щоб знайти найкращий шлях, будуючи поетапно розв'язок, що складаються з вузлів графів. Кожен етап додає вибраний вузол до часткового розчину стохастичним способом, але з ухилом від кількості феромону, доступного на вузлах або ребрах, що їх з'єднують. Шлях феромонів, оновлюється мурахами на кожній ітерації, що дозволяє мурахам співпрацювати у пошуку хороших розв'язків задач. Значення феромону з часом також зменшуються, подібно до випаровування справжнього феромону. Таким чином, розв'язки низької якості вилучаються з подальшого розгляду в міру того, як пошук прогресує.

Застосування ACO до складних комбінаторних задач оптимізації є привабливим завдяки гнучкості та надійності алгоритму. Насправді, методику можна застосовувати для розв'язання широкого кола задач лише з мінімальними змінами основного алгоритму.

Останнім часом було запропоновано багато алгоритмів для розв'язання завдань різних задач комбінаторної оптимізації. При цьому, було показано, що ACO алгоритми дуже ефективні в поєднанні зі спеціалізованими місцевими процедурами пошуку розв'язку для симетричних і асиметричних задач комівояжера (TSP/ATSP, [49], [50], [51]), задач послідовного впорядкування (SOP, [52]), квадратичної задачі про призначення (QAP, [53]), бікватратичних задача про призначення [54]. Одним з найбільш ефективних ACO була колонія мурах System (ACS, [54], [55]), яка містить особливу процедуру поновлення сліду феромона, що активізує пошук в околиці кращого розв'язку.

Дослідники, намагаючись оптимально вирішити VRP, поєднують алгоритми та евристики, щоб використати переваги кожної процедури. Особливо за останні кілька років для досягнення кращих результатів розробляються гібридні алгоритми, що поєднують метоевристики.

1.5 Огляд існуючих програмних продуктів

Серед програмних продуктів, що розв'язують близькі до задачі маршрутизації транспортних засобів, можна виділити наступні:

- Sterling Transportation Management System (IBM) – займається оптимізацією ланцюжка поставок транспортних мереж шляхом планування і аналізу вхідних та вихідних процесів транспортування [58];
- ArcLogistics – інструмент для планування і оптимізації роботи парку транспортних засобів: імпорту замовлень, розрахунку оптимальних маршрутів, створення маршрутних листів, звітів та аналізу ефективності роботи [59];
- ANTOR LogisticsMaster – програмний продукт, який дозволяє здійснювати автоматизацію планування доставки продукції за різними критеріями маршрутів з урахуванням моделі транспортної мережі, пропуску транспорту в певні міські зони, реальних "вікон доставки", об'єму і ваги продукції і т.п. [60];

Таким чином, задача маршрутизації транспортних засобів є актуальною на сьогоднішній день, враховуючи, що існує певна кількість розроблених алгоритмів та програмних продуктів. Але все ж таки задача VRP потребує нового підходу, який був би комбінацією вже відомих алгоритмів розв'язання та їх модифікацій і використовував їх переваги, збільшуючи їх ефективність.

1.6 Постановка задачі дослідження

Мета магістерської роботи – підвищення ефективності методів розв'язання задачі маршрутизації транспортних засобів з часовими вікнами.

Для досягнення поставленої мети, необхідно вирішити такі **задачі**:

- проаналізувати відомі результати розв’язання задачі маршрутизації транспортних засобів;
- удосконалити існуючі алгоритми розв’язання задачі маршрутизації транспортних засобів з урахуванням часових вікон за рахунок модифікації та поєднання метаевристик;
- розробити програмну реалізацію розроблених алгоритмів;
- провести дослідження ефективності розроблених алгоритмів.

Об’єкт дослідження – процес організації транспортних перевезень.

Предмет дослідження – задача маршрутизації транспортних засобів з часовими вікнами.

Висновки до розділу

Було наведено огляд існуючих типів задач маршрутизації транспортних засобів та методів їх розв’язання. У зв’язку з тим, що підвищення ефективності планування перевезень за рахунок мінімізації сумарної вартості транспортних перевезень є одним з важливіших критеріїв оптимізації роботи організації в галузі транспортної логістики, можна зробити висновок про велику практичну значущість задач класу VRP, практичний інтерес до задач даного класу. З метою збільшення ефективності роботи відомих алгоритмів розв’язання даного типу задач, була сформульована відповідна постановка задачі та мета дослідження.

2 РОЗРОБКА МОДИФІКАЦІЙ АЛГОРИТМУ МУРАШИНИХ КОЛОНІЙ ТА ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ЗАДАЧІ VRPTW

Задача маршрутизації транспорту з часовими вікнами (VRPTW) - це розширення задачі маршрутизації транспортних засобів з обмеженою вантажопід'ємністю (CVRP), де обслуговування кожного клієнта повинно відбуватись у відповідному інтервалі часу, який називається часовим вікном. Часові вікна можуть бути жорсткими або м'якими. У разі виникнення жорстких вікон, транспортний засіб, який прибуває занадто рано до клієнта, повинен почекати, поки клієнт не буде готовий розпочати обслуговування. Очікування перед початком часового вікна не несе витрат, окрім часових. У випадку м'яких часових вікон, кожне часове вікно може бути порушене з доданням штрафних витрат. Часові вікна можуть бути односторонніми, наприклад, вказані як кінцевий час доставки.

Часові інтервали в реальному житті виникають у задачах, з якими стикаються бізнес-організації, які працюють за гнучкими графіками. Конкретні задачі з жорсткими вікнами включають патрульну службу безпеки, банківські розсилки, поштові доставки, збір промислових відходів, доставку продуктів, маршрути шкільних автобусів та розповсюдження міських газет. У цьому розділі використовується варіант жорстких часових вікон, який є найбільш вивченим.

Задача маршрутизації транспортних засобів з врахуванням часових вікон (Vehicle Routing Problem with Time Windows, VRPTW) полягає в знаходженні таких маршрутів транспортних засобів, які будуть задовольняти потреби клієнтів в товарі та враховувати часові рамки, в яких клієнт може бути обслуговуваний, та мінімізації загальних витрат на перевезення. Якщо транспортний засіб прибув до пункту обслуговування раніше початку встановленого клієнтом часового проміжку – ТЗ має зачекати доки клієнт буде вільний для обслуговування, якщо ТЗ прибув до пункту обслуговування

після закінчення часового проміжку – клієнт не може бути обслуговуваний [56].

Передбачається, що перевезення товару відбувається K транспортними засобами, всі транспортні засоби починають та завершують свої маршрути в депо.

У існуючій літературі про VRPTW кількість транспортних засобів, доступних для обслуговування клієнтів, зазвичай вважається необмеженою, і цільова функція залежить від обраного методу розв'язання [56]. Для точних методів метою є мінімізація загальної пройденої відстані. Для евристики основною метою є також мінімізація кількості використовуваних транспортних засобів, а вторинною - мінімізація загальної пройденої відстані. З цього загального твердження можуть бути винятки.

2.1 Математична постановка задачі VRPTW

Математична модель задачі маршрутизації транспортних засобів з урахуванням часових вікон представлена наступними формулами:

$$\sum_{i=0}^n \sum_{j=0, j \neq i}^n \sum_{k=1}^K c_{ij} x_{ijk} \rightarrow \min, \quad (2.1)$$

$$\sum_{k=1}^K \sum_{j=1}^n x_{0jk} \leq 1, \quad (2.2)$$

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} \leq 1, \text{ для } k \in \overline{1, K}, \quad (2.3)$$

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^n x_{ijk} = 1, \text{ для } i \in \overline{1, n}, \quad (2.4)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^n x_{ijk} = 1, \text{ для } j \in \overline{1, n}, \quad (2.5)$$

$$\sum_{i=1}^N q_i \sum_{j=0, j \neq i}^N x_{ijk} \leq Q, \quad \text{для } k \in \overline{1, K}, \quad (2.6)$$

$$\sum_{i=1}^n \sum_{j=0, j \neq i}^n x_{ijk} (w_i + s_i + t_{ij}) \leq T_k, \quad \text{для } k \in \overline{1, K}, \quad (2.7)$$

$$t_0 = w_0 = s_0 = 0, \quad (2.8)$$

$$x_{ijk} (t_i + w_i + s_i + t_{ij}) \leq t_j, \quad \text{для } i \in \overline{1, n}, j \in \overline{1, n}, (i \neq j), k \in \overline{1, K}, \quad (2.9)$$

$$e_i \leq (t_i + w_i) \leq l_i, \quad \text{для } i \in \overline{1, n}, \quad (2.10)$$

Де відомими є:

c_{ij} – вартість переїзду з i в j , $(i, j) \in N$;

t_{ij} – тривалість переїзду з i в j , $(i, j) \in N$;

q_i – потреби в товарі пункту i , $i \in N$;

s_i – тривалість обслуговування в пункті i , $i \in N$;

e_i – початок часового вікна в пункті i , $i \in N$;

l_i – кінець часового вікна в пункті i , $i \in N$;

K – загальна кількість транспортних засобів;

T_k – максимальна тривалість (час) маршруту для k -го транспортного засобу, $k \in K$;

Q – місткість ТЗ;

$N, n+1$ – кількість пунктів, включаючи депо.

Змінними є:

t_i – час прибуття в пункт i , $i \in N$;

w_i – час очікування в пункті i , $i \in N$.

$$x_{ijk} = \begin{cases} 1, & \text{якщо маршрут } (i, j) \text{ був обслугован } k - \text{им ТЗ} \\ 0, & \text{в іншому випадку} \end{cases}.$$

Таким чином, зміст складових моделі такий:

(2.1) – цільова функція, що мінімізує витрати на перевезення;

(2.2) – k -ий ТЗ може виїхати із депо не більше одного разу;

(2.3) – якщо k -ий ТЗ виїхав з депо, то він має туди повернутися;

(2.4) – із i -го пункту k -им ТЗ можна виїхати лише один раз;

(2.5) – в j -ий пункт k -им ТЗ можна в'їхати лише один раз;

(2.6) – місткості ТЗ;

(2.7) – загальний час перебування в маршруті k -ого ТЗ не може перевищувати максимальної тривалості маршруту для k -го ТЗ;

(2.8) – початкові умови;

(2.9) – добуток часу приїзду в i -ий пункт, часу очікування в i -ому пункті, часу обслуговування в i -ому пункті та часу переїзду з i -го пункту в j -ий не може бути більшою за час приїзду в j -ий пункт;

(2.10) – час приїзду в i -ий пункт і час очікування в i -ому пункті мають попадати в часове вікно i -го пункту.

Для розв'язання задач VRPTW у літературі було запропоновано різні методи: як точні, так і наближені.

За своєю складністю задачі VRPTW відносяться до NP-повних задач. Для розв'язання таких задач розмірністю більше 20 вузлів точні методи можуть використовуватись лише коли можна знехтувати часом пошуку розв'язку, оскільки перебір можливих розв'язків для знаходження оптимуму займає колосального часу обчислень навіть при відносно невеликій розмірності задачі.

Для розв'язання реальних практичних задач застосовуються евристичні і метаевристичні методи [48]. Евристичні методи можуть застосовуватись для знаходження наближених розв'язків за прийнятний час, вони не гарантують знаходження розв'язків високої якості, хоча вони можуть бути виявлені емпірично. Метаевристичні методи базуються на ретельному вивченні найперспективніших частин простору розв'язків, що сприяє виграшу у якості відносно класичних евристик, тож за допомогою метаевристик можливе знаходження прийнятних розв'язків за обґрунтований час.

Гібридні алгоритми, де поєднано кілька ідей евристик або метаевристик, широко використовуються для розв'язання різних комбінаторних задач оптимізації. У цій роботі було запропоновано новий гібридний алгоритм оптимізації мурашиних колоній з покращеннями у

поєднанні з алгоритмом Табу пошуку. Алгоритм показує хорошу продуктивність та результати для відомого набору даних Соломона [9].

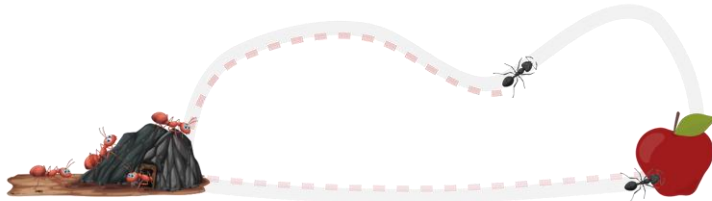
2.2 Алгоритми Ant Colony Optimization для розв’язання VRPTW

2.2.1 Опис класичного алгоритму Ant Colony Optimization

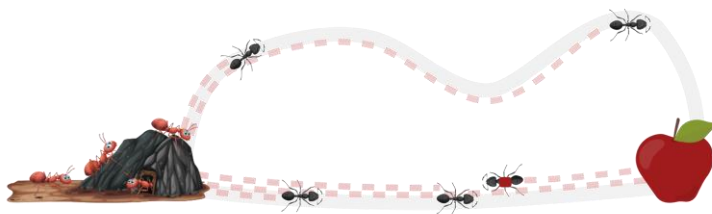
В даний час в рамках напрямку «Природних обчислень» (Natural Computing) активно створюються і вдосконалюються методи глобальної оптимізації, інспіровані природними системами. У цих методах штучного інтелекту відображені механізми прийняття рішень та адаптації, сформовані природним шляхом в наслідок мільйонів років еволюції [54]. Подібним чином мурахи за більш ніж 100 мільйонів років існування пристосувалися до виживання в різних умовах практично по всьому світу, утворюють популяцію 10^{16} особин, загальна вага яких становить від 10 до 25% земної біомаси наземних тварин [62]. Мурашині колонії при відносно простій поведінці кожної окремої особини представляють складну соціальну структуру і здатні вирішувати складні задачі, наприклад, знаходити найкоротший шлях за допомогою хімічної регуляції. Рисунок 2.1 наочно зображує поведінку справжніх мурашок у пошуках джерел їжі.



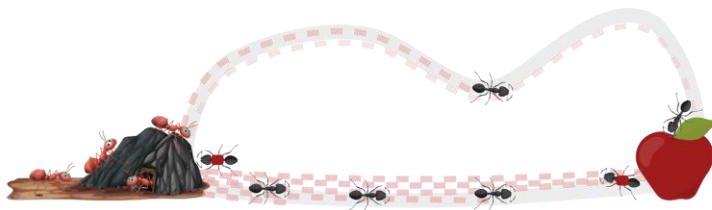
Дві мурахи починають рух з однаковою ймовірністю.



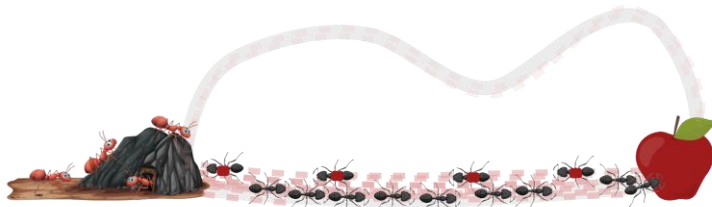
Мурха, що обрала коротший шлях, швидше знаходить джерело їжі.



Концентрація феромону на коротшому шляху зростає швидше ніж на довшому.



Велика кількість мурах починають обирати шлях з шільнішим феромоном.

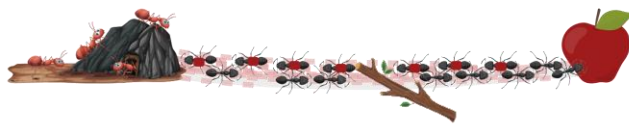


З часом мурахи майже виключно обирають короткий шлях, а на довшому шляху феромон випаровується.

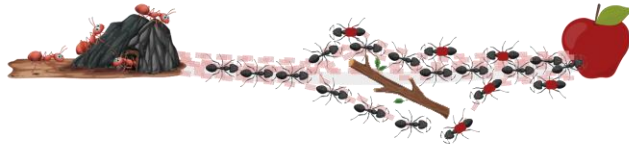
Рисунок 2.1 – Поведінка мурах під час пошуку джерела їжі

Феромоновий слід дозволяє мурахам знаходити нові маршрути у разі виникнення перешкод на існуючому маршруті. Якщо найкоротший шлях недоступний через виникнену перешкоду – мурахам необхідно знайти новий найкоротший шлях. Спочатку вони рівноймовірно будуть оминати перешкоду з обох сторін як на шляху до їжі, так і на зворотньому шляху. Ті мурахи, які оберуть короткий шлях – будуть швидше його проходити в обидві сторони, тож відповідно він більше буде збагачений феромоном. Оскільки чим вище концентрація феромону на шляху – тим ймовірніше пересування мурах по цьому шляху – наступні мурахи оберуть ций шлях з більшою ймовірністю,

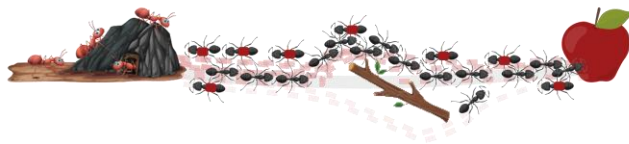
збагачуючи шлях феромоном допоки він буде залишатись доступним.
(рисунок 2.2).



При виникненні перешкоди мурахи успішно адаптуються і знаходять новий оптимальний шлях.



Мурахи рівноймовірно обходять перешкоду.



Мурахи, що обрали короткий шлях, будуть швидше оминати перешкоду та досягати джерела їжі й повертатись назад, збагачуючи феромоном короткий шлях.

Рисунок 2.2 – Поведінка мурах під час виникнення перешкоди

Така поведінка мурах привернула увагу багатьох дослідників, які вивчали механізми взаємодії особин колонії. Ставлячи експерименти з вибору між двома шляхами нерівної довжини, що ведуть від гнізда до джерела їжі, біологи помітили, що, як правило, мурашки в кінцевому рахунку використовують короткий маршрут [61].

Модель такої поведінки полягає в наступному:

- спочатку мурахи рухаються випадковим чином від гнізда до джерела їжі;
- коли один з мурах знаходить їжу, він повертається назад в гніздо, залишаючи за собою слід особливого ферменту – феромону;
- інші мурахи сприймають «запах» відкладеного феромону і намагаються слідувати зазначеному шляху;
- феромонні стежки породжують асинхронну і непряму схему комунікації, завдяки якій мурахи побічно обмінюються один з одним інформацією по мірі виявлення альтернативних шляхів до їжі: чим

вище концентрація феромону на шляху, тим імовірніше, що інші мурахи посліднують йому;

- коротші шляхи вимагають менше часу і тому частіше за інших позначаються феромонами, а висока концентрація феромону, в свою чергу, робить короткий шлях привабливішим, залучаючи все більше число мурах;
- рідко використовувані довгі шляхи, в кінцевому підсумку, зникають через зниження кількості феромону в процесі випаровування.

Така непряма комунікація в мурашиній колонії, звана стигмергією, за рахунок описаних механізмів позитивної та негативної зворотного зв'язку забезпечує можливість виявлення і закріплення найкоротшого шляху. При цьому можна виділити наступні основні принципи поведінки мурашиної колонії [61]:

- простота дій кожного мурашки;
- відсутність централізованого контролю;
- непрямий обмін інформацією за допомогою відкладання феромона;
- випаровування феромону з плином часу, що забезпечує адаптивність поведінки.

Мурашиний алгоритм, також відомий як алгоритм оптимізації мурашиними колоніями (Ant Colony Optimization Algorithm), є відомою метаевристикою, що відноситься до методів ройового інтелекту, суть якої полягає в аналізі та використанні описаної моделі поведінки колонії мурах стосовно розв'язання різних задач пошуку маршрутів на графах. Перший мурашиний алгоритм запропонував у своїй докторській дисертації Марко Доріго в 1992 р для задачі пошуку оптимального шляху на графі [9]. Основні етапи роботи мурашиного алгоритму в загальному вигляді представлені на структурній схемі на рисунку 2.3. Віртуальні мурахи послідовно будують розв'язки задач, пересуваючись по ребрах графа. При цьому з найбільшою ймовірністю для руху вибираються ребра з малою довжиною і великою кількістю феромону, заданого деяким умовним значенням для кожного ребра.

Феромон оновлюється на кожній ітерації: збільшується на побудованих маршрутах пропорційно їх довжині і зменшується на всьому графі з урахуванням коефіцієнта випаровування.

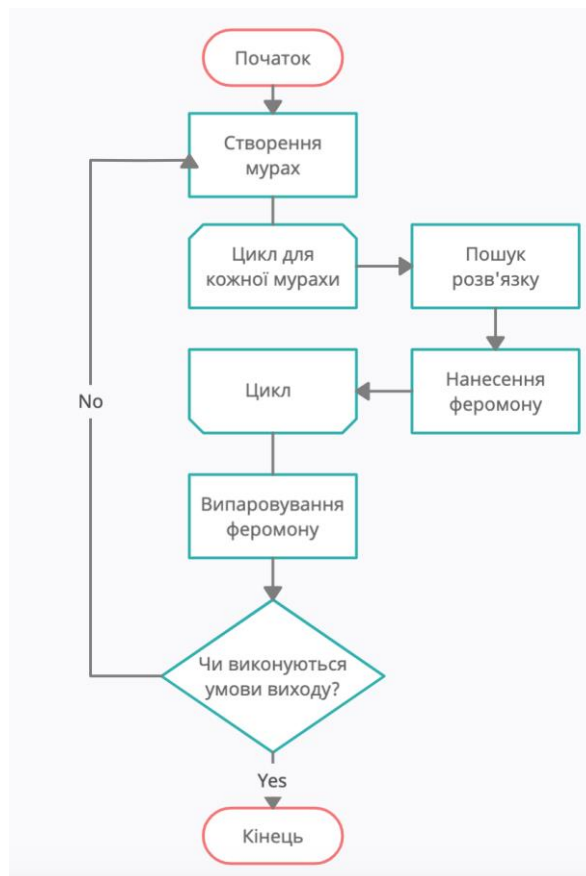


Рисунок 2.3 – Структурна схема алгоритму мурашиних колоній в загальному вигляді

З часом більшість мурах будуть обирати саме найкоротший шлях завдяки накопиченню великої частки феромону на ньому, що є позитивним зворотнім зв'язком. Для уникнення застрягання в локальному оптимумі моделюється також і негативний зворотній зв'язок у вигляді випаровування феромону. При моделюванні поведінки мурах на графі, де ребра є можливими шляхами переміщення мурах, найзбагаченіший феромоном шлях по ребрам графу – розв'язок задачі, результат роботи АСО.

Питання про збіжність мурашиних алгоритмів представляє теоретичний інтерес для наукових досліджень. На поточний момент існують докази гарантованої збіжності мурашиного алгоритму, однак через його стохастичний час збіжності є невизначеним [61].

Узагальнена схема алгоритму

Мурашиний алгоритм, незалежно від модифікацій, можна представити в наступному вигляді [62]:

Крок 1 Поки умови виходу не виконані повторювати кроки:

Крок 1.1 Створити мурах

Крок 1.2 Знайти розв'язок

Крок 1.3 Оновити феромон

Крок 1.4 Додаткові дії (опціонально)

Створити мурах

Умови задачі визначають початкову точку вміщення мурах, що є визначальним для кожної задачі. Існують такі варіанти розміщення особин: починають з одної точки, починають з різних точок (можуть повторюватись, або не повторюватись).

При створенні мурах невеликим позитивним числом визначається початковий рівень феромону, для запобігання нульових ймовірностей переходу на початковому кроці[62].

Знайти розв'язок

В кожному моменті часу визначається ймовірність переходу між вершинами за наступною формулою:

$$p_{ij}(t) = \frac{\tau_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{j \in J_i} \tau_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}, \quad (2.11)$$

де:

J_i – множина вершин, в які дозволено перехід з вершини i ;

$\tau_{ij}(t)$ – рівень феромону на дузі $i - j$ на кроці t ;

d_{ij} – відстань між вершинами i та j ;

α, β – константні параметри.

При $\alpha = 0$ вибір найближчого міста найбільш ймовірний, тобто алгоритм стає жадібним. При $\beta = 0$ феромон є єдиним аргументом вибору, що призводить до локального оптимуму (субоптимальних розв'язків). Тож з'являється необхідність експериментального пошуку компромісу між цими величинами [62].

Оновити феромон

Рівень феромона оновлюється відповідно до формули:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k \in M_{ij}} \frac{L_{\min}}{L_k}, \quad (2.12)$$

де:

ρ – інтенсивність випаровування;

M_{ij} – множина мурашок, що пройшли по дузі (i, j) ;

L_k – вартість поточного розв'язку (довжина маршруту) для k -го мурахи;

$L_{\min}$ – параметр, що має значення порядку вартості оптимального розв'язку (передбачувана ідеальна ціна розв'язку в задачі на мінімум);

$\frac{L_{\min}}{L_k}$ – феромон, який відкладався k -им мурахою на ребрі (i, j) . Таким чином,

чим гірше розв'язок (чим більше відповідне L_k), в яке увійшло ребро (i, j) , тим менше феромона буде відкладено мурахою на цьому ребрі [62].

Додаткові дії

Як правило, на цьому етапі використовується алгоритм локального пошуку, але він також можливий на фінальному етапі після пошуку всіх розв'язків.

2.3 Переваги та недоліки мурашиного алгоритму

Оскільки VRP є NP-важкою, за обмеженням, VRPTW також NP-важка. Насправді навіть пошук можливого розв'язку VRPTW для фіксованої кількості транспортних засобів сам по собі є NP-важкою задачею.

Мурашиний алгоритм не гарантує отримання оптимального розв'язку, але має поліноміальну складність, що дозволяє в короткі терміни отримувати наближений розв'язок важких задач, таких як VRP. Точні методи мають факторіальну складність, тому при кількості пунктів понад 10 зазвичай не ефективні

Переваги АСО

Для VRP мурашині алгоритми порівняльно ефективні, вони працюють краще, ніж інші глобальні оптимізації для VRP (нейронні мережі, генетичні алгоритми) [63].

Порівняння мурашиних алгоритмів з генетичними (Genetic Algorithms):

- алгоритм орієнтується на пам'ять всієї колонії (на відміну від пам'яті попереднього покоління);
- випадковий вибір шляху і пам'ять колонії мінімізують схильність до неоптимального початкового розв'язку;
- адаптовані до динамічних змін;
- застосовуються до багатьох різних задач.

Недоліки

До недоліків можна віднести:

- а) теоретичний аналіз ускладнений тим, що:
 - розподіл ймовірностей змінюється ітеративно;
 - дослідження в більшій мірі є експериментальним, ніж теоретичним;
- б) збіжність гарантується, але час збіжності не визначений;
- в) зазвичай необхідне застосування додаткових методів таких, як локальний пошук;
- г) алгоритм залежить від параметрів, які підбираються тільки виходячи з експериментів.

2.4 Модифікації алгоритму мурашиних колоній

В роботі була запропонована модифікація АСО, у яку додаються правила побудови допустимих розв'язків [1].

Основною складовою АСО алгоритму є знаходження ймовірності переходу з i -ого пункту в j -ий в момент часу t [62]. Межі часових вікон безпосередньо впливають на ймовірність переходу в ту чи іншу вершину, бо чим менше значення кінця часового вікна в певній вершині, тим вірогідніше потрібно обрати цю вершину. Для врахування наявності часових вікон ймовірності переходу визначатимемо наступною формулою:

$$p_{ij}(t) = \frac{\tau_{ij}(t)^\alpha \cdot \left(\frac{1}{d_{ij}}\right)^\beta \cdot \left(\frac{1}{r_j}\right)^\gamma}{\sum_{j \in J_i} \tau_{ij}(t)^\alpha \cdot \left(\frac{1}{d_{ij}}\right)^\beta \cdot \left(\frac{1}{r_j}\right)^\gamma}, \quad (2.13)$$

де:

J_i – множина вершин, в яких дозволено перехід з вершини i ;

$\tau_{ij}(t)$ – рівень феромону на ребрі $i - j$ на кроці t ;

d_{ij} – відстань між вершинами i та j ;

$r_j = l_j - t$ – залишок часу до закінчення часового вікна $[e_j, l_j]$ в j -ому пункті;

α, β, γ – константні параметри ($\alpha, \beta, \gamma \in [0,1]$).

Задля прискорення знаходження розв'язку та покращення отриманих розв'язків АСО запропоновано такі правила:

- відсікання неперспективних маршрутів, якщо поточне значення цільової функції більше за вже знайдене краще (дозволить зменшити кількість неефективних ітерацій);
- достроковий перехід на наступну ітерацію, якщо на поточній ітерації було знайдено m послідовних однакових розв'язків (не спостерігалось m послідовних покращень знайденого розв'язку);

- оновлювати феромон після кожного мурахи, який побудував повний маршрут зі значенням цільової функції меншою за вже краще знайдене;
- обнуляти феромон на початку кожної ітерації для унеможливлення зациклення алгоритму на знайденому кращому розв'язку з попередньої ітерації.

2.5 Розробка гібридного алгоритму

Останнім часом було запропоновано багато алгоритмів АСО для розв'язання різних задач комбінаторної оптимізації. Проте дослідження показують, що ефективність результатів підвищується, якщо об'єднати кілька метаевристик. Розробка гібридних алгоритмів є перспективним напрямком, оскільки гібридний алгоритм поєднує в собі переваги використовуваних алгоритмів.

Для підвищення ефективності модифікованого алгоритму мурашиних колоній запропоновано розробити гібридний алгоритм, який поєднує в собі переваги двох метаевристик, що дозволить досягти поліпшення результатів роботи алгоритму в цілому [1], а саме алгоритм оптимізації мурашиних колоній та табу-пошук.

Табу-пошук — метаевристика, яка здатна значно підвищити продуктивність локального пошуку за рахунок використання структур пам'яті. Ідеї алгоритмів АСО і Табу пошуку дуже різні. Табу пошук не ґрунтується на інформації з попередньої ітерації, як це робить АСО, а випадково вибирає напрямок, який може збільшити простір пошуку. На кожній ітерації вивчаються околиці поточного розв'язку методами локального пошуку, і кращий розв'язок обирається в якості нового поточного розв'язку. Однак, на відміну від класичної методики локального пошуку, процедура не зупиниться на першому локальному оптимумі. Якщо потенційний розв'язок було знайдено раніше або якщо були порушені задані правила, то він позначається

як «табу» (заборонене) з тією метою, щоб алгоритм не розглядав даний розв'язок повторно, тому що він є неперспективним. Кращий розв'язок в околиці буде обрано в якості поточного розв'язку, навіть якщо він гірше, ніж поточний розв'язок. Такий підхід дозволяє методу відійти від локальних оптимумів і вивчити велику частину простору пошуку [62].

Схема гібридного алгоритму.

Запропонована схема гібридного алгоритму базується на схемі АСО з додаванням ітерацій Табу пошуку [1]:

Крок 1. Повторювати наступні кроки доки на протязі θ ітерацій рекордний розв'язок не буде покращуватись:

Крок 1.1 Створити мурах.

Крок 1.2 Знайти розв'язки.

Крок 1.3 Для кожного з b найкращих розв'язків, отриманих на поточній ітерації, застосувати Табу-пошук:

Крок 1.3.1 Створити Табу список.

Крок 1.3.2 Доки рекордний розв'язок не буде покращуватись на протязі ϑ ітерацій:

Крок 1.3.3 Визначити околиці поточного рекордного розв'язку, які виключають розв'язки додані до Табу списку.

Крок 1.3.4 Знайти найкращий розв'язок в околиці.

Крок 1.3.5 Оновити табу список.

Крок 1.4 Оновити феромон.

Запропонована схема гібридного алгоритму базується на схемі АСО з додаванням ітерацій Табу пошуку, що дозволяє якісно поліпшити показники обох метаевристик, бо розроблений гібридний алгоритм поєднує в собі переваги обох метаевристик, а як наслідок, вдається досягти істотного поліпшення результатів знаходження розв'язку задачі [1].

Висновки до розділу

Було представлено математичну модель задачі VRPTW та метод її розв’язання. Було обґрунтовано вибір гібридного алгоритму та його метаевристик. Для АСО було наведено узагальнену схему алгоритму, перераховані основні його недоліки та переваги. Було запропоновано модифікації АСО та гібридний алгоритм її розв’язання, розроблено гібридний алгоритм з використанням поліпшеного АСО та Табу пошуку.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Для реалізації гібридного алгоритму з використанням модифікованого алгоритму оптимізації мурашиних колоній та табу пошуку для задачі маршрутизації транспортних засобів з часовими вікнами було розроблено програмне забезпечення з використанням мови програмування Python.

Python – це високорівнева об'єктно-орієнтовна крос-платформна інтерпретовна мова програмування, що характеризується інтерактивністю, модульністю, динамічністю, портативністю.

Вибір цієї мови обґрунтований наступними перевагами.

Обширні бібліотеки – в python інтегровані великі стандартні бібліотеки, які включають переважну більшість поширених задач програмування, що дозволяє значно зменшити довжину коду та спростити використання мови.

Гнучкість – python легко розширюється до інших популярних мов програмування, наприклад C, C++, Java, а також легко вбудовується в інші мови програмування, що дає змогу включати можливості функцій python, використовуючи інші мови програмування.

Портативність – доки в код не включено унікальних системних функцій – код python з легкістю можна перенести на іншу платформу.

Продуктивність – простота, бібліотеки функції інтеграції процесів, модульні системи тестування та розширені можливості управління python дають переваги ефективності розробників, збільшенню швидкості та продуктивності програм ніж при використанні інших мов, розробники можуть зосередитись на унікальному коді під конкретний проект, використовуючи наявні бібліотеки.

Простота – оскільки більшість функцій інкапсульовано у бібліотеки, то програмувати, читати та розуміти код на python дуже просто, відсутність фігурних дужок спрощує читання коду та полегшує отладку.

Інтерпретовність – у python оператори виконують по одному, що спрощує отладку відносно скомпільованих мов.

IoT – python застосовується для інтернету речей, який стає дедалі актуальнішим.

Мультифункціональність та інтегровність – python застосовний у багатьох платформах – для web, mobile, front-eng, back-end розробки. Також на python можлива розробка скриптів для конкретних сценаріїв, з уникненням розробки глобального програмного забезпечення. Python також популярна мова для data sciencета machine learning.

Велике ком'юніті – інклюзивна велика спільнота програмістів, що використовують python – підтримка програмістів на етапах розробки, наявність експертних тематичних джерел та розробка додаткових бібліотек.

Довіра у великих проектах – останнім часом Python обирають такі корпорації як Google, YouTube, Dropbox, Yahoo, використовуються у галузях освіти, науки, охорони здоров'я, фінансів.

Бібліотеки, що використовуються у проєкті, зображені на рисунку 3.1

```
from PyQt5 import QtWidgets, QtCore, QtGui, Qt
import random
import math
import copy
import os
import io
import time
import datetime
```

Рисунок 3.1 - Бібліотеки, що використовуються у проєкті

Призначення використаних бібліотек:

- PyQt5 – модуль для роботи з GUI додатків;
- random – модуль, що дозволяє використовувати випадкові значення;
- math – модуль з вбудованими математичними операціями;
- copy – модуль, що дозволяє скопіювати об'єкти (стандартно в python при копіюванню створюється посилання на цільовий об'єкт);

- `os` – модуль для автоматичної роботи з задачами операційної системи (створити/видалити папку, отримати її зміст, відкрити файл по шляху, змінити директорію);
- `time` – модуль, що дозволяє використовувати різні функції, пов'язанні з часом;
- `datetime` – модуль забезпечує класи для маніпуляцій з датами та часом.

В якості IDE для розробки програмного продукту було використано Atom (рисунок 3.2) – крос-платформне IDE з відкритим кодом, розроблена GitHub`ом.



Рисунок 3.2 - IDE Atom.

3.2 Архітектура програмного забезпечення

3.2.1 Структура класів програмного забезпечення

Клас *POINT* відповідає за введення даних задачі вручну: отримує координати точки (клієнта), що були додані на граф, данні про клієнта (попит, часове вікно, час обслуговування) та парк ТЗ: кількість ТЗ та їх місткість.

Клас *WINDOW* відповідає за роботу програми та створення і роботу з інтерфейсом, візуалізацію роботи алгоритму: відповідає за отримання та збереження даних задачі із файлу або із класу *POINT*, розрахунок матриць відстаней, ініціює роботу гібридного алгоритму, відстежує події роботи курсора, зберігає отримані дані задачі, візуалізує етапи роботи та результат алгоритму.

Клас *WORKER* відповідає за реалізацію алгоритмів.

3.2.1 Специфікація функцій програмного забезпечення

Таблиця 3.1 містить опис функцій програмної реалізації гібридного алгоритму.

Таблиця 3.1 – Специфікація функцій програмної реалізації гібридного алгоритму

Назва функції	Вхідні параметри	Опис функції
POINT.__init__	x, y – координати x, y; type_point – тип точки (клієнт/депо).	Зберігає інформацію про клієнта чи депо.
POINT.send_data	-	Відправляє в потік отриманні дані задачі
CALC_WORKER.__init__	count_ants – кількість мурах; max_vehicle_number – макс. кількість ТЗ, distance_matrix – матриця відстаней, data – дані задачі, vehicle_capacity – місткість ТЗ, alpha, beta, gamma – коеф. ймовірностей, count_final_iterations – к-сть ітерацій для виходу з алгоритму, start_pheromone – початковий феромон, rate_minimum_distance, evaporation_pheromone_rate – коефіцієнт випаровування феромону, count_best_ants – кількість кращих мурах для ТАБУ, count_tabu_iterations – кількість ітерацій табу пошуку.	Створює потік для роботи алгоритму.
CALC_WORKER.run	-	Реалізовує гібридний алгоритм

Продовження таблиці 3.1

Назва функції	Вхідні параметри	Опис функції
CALC_WORKER. calc_ant	Ant – масив даних про мураку	Рахує загальну відстань та шлях, що пройшли мурахи
CALC_WORKER. tabu_search	result – масив мурах популяції	Реалізовує табу пошук
CALC_WORKER. return_transition	current_point – поточний клієнт, car - T3, available_points – доступні клієнти, pheromone_matrix – матриця феромонів	Шукає наступного клієнта для T3
CALC_WORKER. return_transition_asa	current_point – поточний клієнт, car - T3, available_points – доступні клієнти, pheromone_matrix – матриця феромонів	Шукає наступного клієнта для T3 з урахуванням часових вікон
CALC_WORKER. is_avaliable	start_point – початковий клієнт, finish_point – наступний клієнт, car - T3	Розрахунок можливості відвідування клієнта
CALC_WORKER. return_time	start_point – початковий клієнт, finish_point – наступний клієнт, car - T3	Розрахунок часу пересування
CALC_WORKER. is_finish	result – масив мурах популяції	Перевірка умов виходу з алгоритму
CALC_WORKER. update_pheromone_matrix	pheromone_matrix - матриця феромонів, ants – масив мурах, best_distance – кращий маршрут	Оновлення феромонів у матриці
CALC_WORKER. return_values	result – масив мурах популяції	Повертає ребра, по яким пройшовся мураха та відстань
CALC_WORKER. return_best	result – масив мурах популяції	Шукає кращу мураку

Продовження таблиці 3.1

Назва функції	Вхідні параметри	Опис функції
CALC_WORKER. terminate_worker	-	Зупинка розрахунків
WINDOW. __init__	-	Створює інтерфейс програми
WINDOW.resizeEvent	event - подія	Масштабує вікно
WINDOW. read_file	path – шлях до файлу	Зчитує данні задачі із файлу
WINDOW. add_point	-	Запускає очікування кліку для додавання клієнта
WINDOW.add_depart	-	Запускає очікування кліку для додавання депо
WINDOW. start_calc	-	Створює екземпляри класу CALC_WORKER, запускає алгоритми
WINDOW. calc_distance_matrix	-	Рахує матрицю відстаней
WINDOW. return_distance	i, j - клієнти	Рахує відстань між двома клієнтами
WINDOW. mousePressEventFrame	event - подія	Реагує на клік по полю відображення клієнтів
WINDOW. mouseScrollEventFrame	event - подія	Реагує на прокрутку
WINDOW. accept_data	values – масив даних про отриману точку	Додає до розрахунків отриманого клієнта
WINDOW. draw_point	number – номер точки	Малює точку(клієнта)
WINDOW. draw_depart	-	Малює депо

Продовження таблиці 3.1

Назва функції	Вхідні параметри	Опис функції
WINDOW. draw_lines	signal - сигнал	Малює та зберігає маршрути
WINDOW. clear_scene	-	Очищує сцену
WINDOW. stop_work	-	Зупиняє роботу алгоритму
WINDOW. clear_data	-	Очищує всю отриману та розраховану інформацію
WINDOW. init_car_colors	-	Генерує кольори маршрутів для кожної машини

3.3 Вхідні данні

Вхідні дані – це дані, які надходять до системи і які необхідні для її подальшої роботи. Вхідні данні реєструються в програмі за допомогою форм, або за допомогою імпортування з файлу.

Дані про депо та парк ТЗ:

- кількість ТЗ, які можна задіяти у доставці;
- місткість ТЗ.

Дані про клієнта :

- ідентифікатор (номер);
- координати;
- потреба у товарі;
- початок та кінець часового вікна;
- час обслуговування.

Дані для роботи алгоритму:

- коефіцієнти α, β, γ .

3.4 Вихідні дані

Вихідними даними є сформований план перевезень продукції, у якому враховані потреби всіх клієнтів, часові вікна їх обслуговування та який є мінімальним по витратам на перевезення.

3.5 Інструкція користувача

В даному розділі розміщено інструкцію запуску та користування розробленою програмною реалізацією гібридного алгоритму покращеного алгоритму оптимізації мурашиними колоніями та табу пошуком.

Для початку роботи з програмою, необхідно розмістити файл програми у папку поточного user на жорсткому диску, запустити термінал та ввести команду “python3 app.py” (рисунок 3.3).

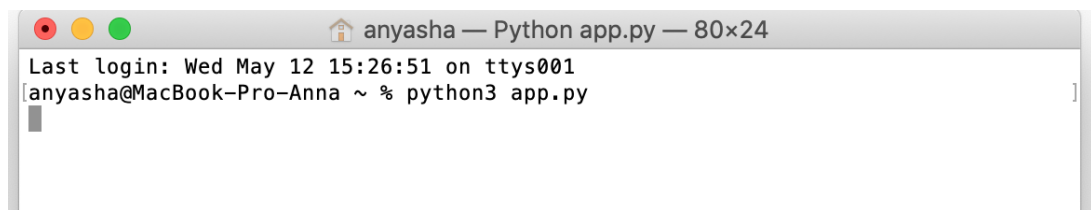


Рисунок 3.3 – Команда “python3 app.py” у терміналі

Після чого відбудеться запуск програми (рисунок 3.4).

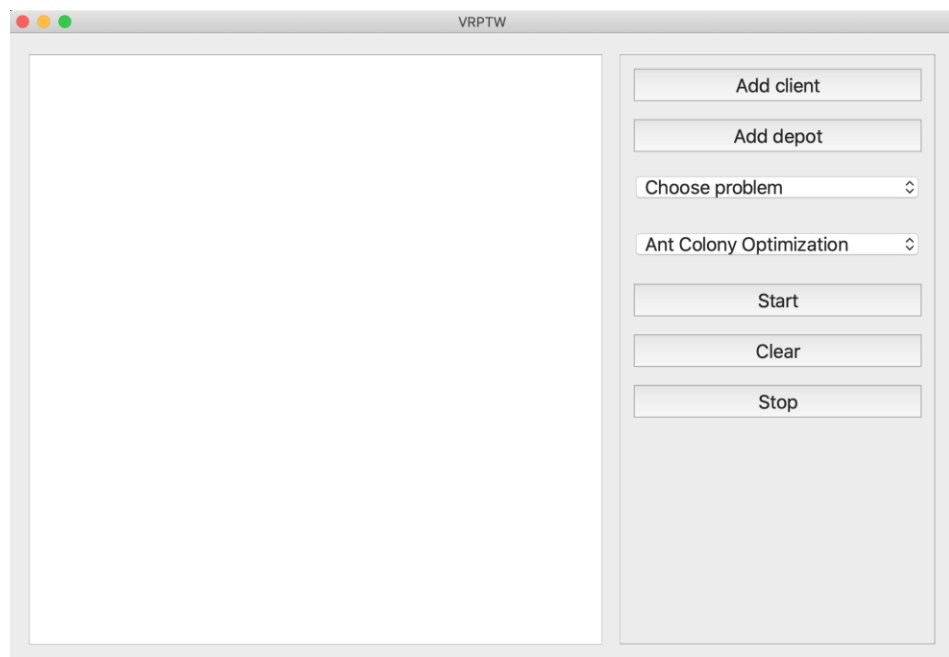


Рисунок 3.4 – Вікно програми після запуску

На сторінці відображаються кнопки для додавання клієнтів і депо вручну, або зчитування задачі з файлу. Для введення даних задачі вручну, необхідно додати точки, що відповідають розташуванню клієнтів та депо, та ввести всю необхідну інформацію.

Процес додавання клієнтів:

- а) натиснути кнопку “Add client”;
- б) поставити крапку у полі форми (рисунок 3.5);

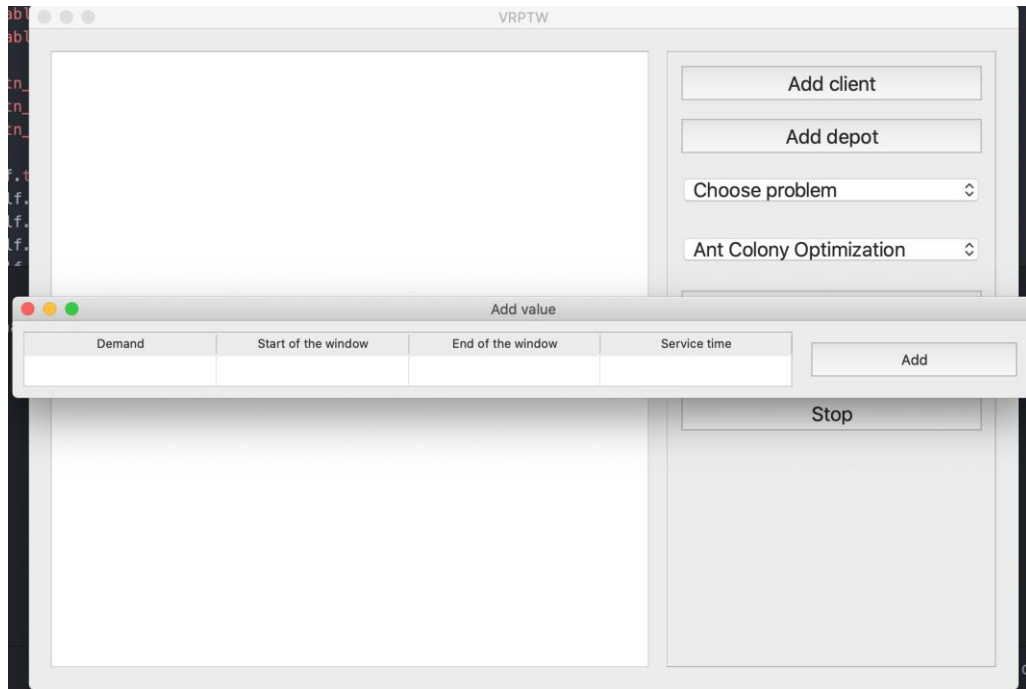


Рисунок 3.5 – Додання клієнта

- в) ввести параметри у відповідні поля (рисунок 3.6);

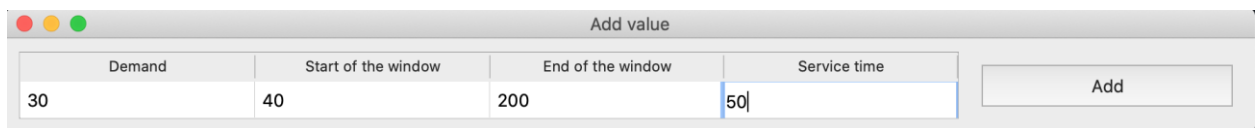


Рисунок 3.6 – додання даних про клієнта

- г) натиснути кнопку «Add».

Процес додавання депо:

- а) натиснути кнопку “Add depot”;
- б) поставити крапку на полі форми (рисунок 3.7);

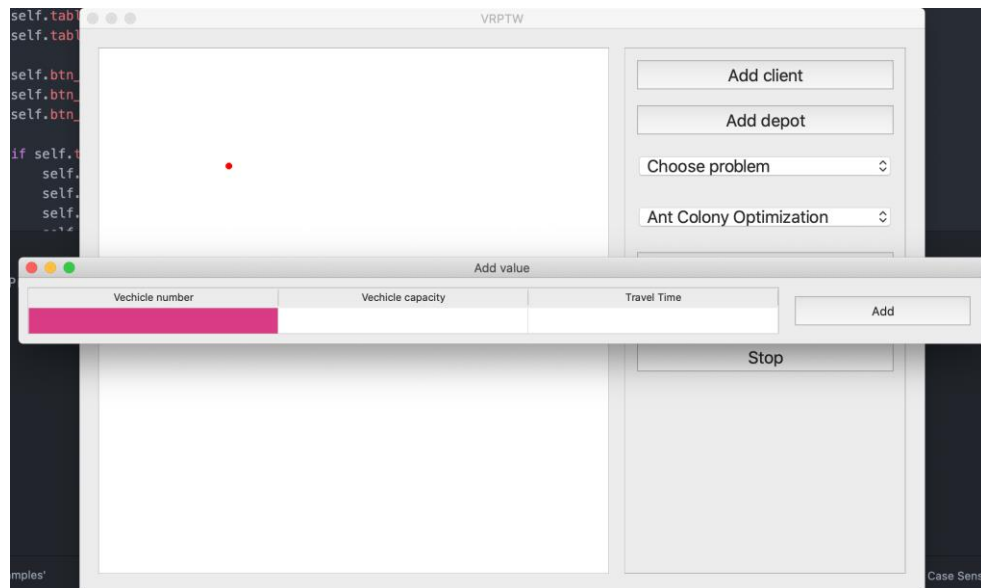


Рисунок 3.7 – додання депо

в) ввести параметри у відповідні поля (рисунок 3.8);

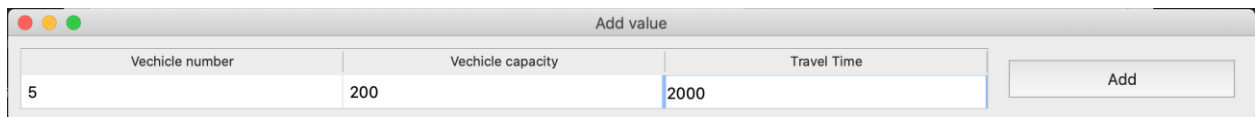


Рисунок 3.8 – додання даних про депо

г) натиснути кнопку “Add”.

Процес запуску задачі з файлу:

Обрати задачу зі списку (рисунок 3.9).

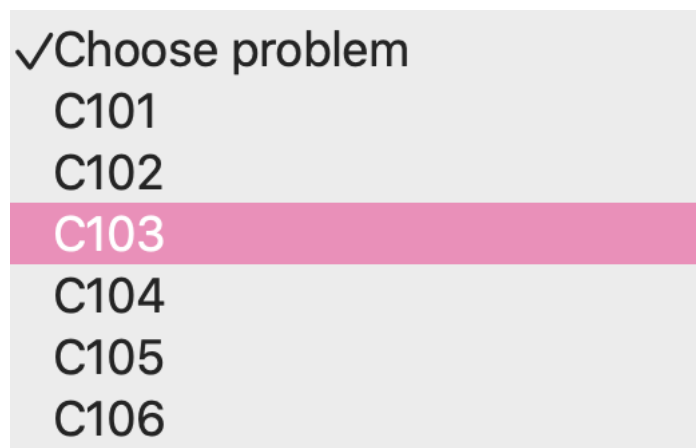


Рисунок 3.9 – список задач Соломона

Запуск розрахунків:

а) обрати алгоритм (рисунок 3.10);

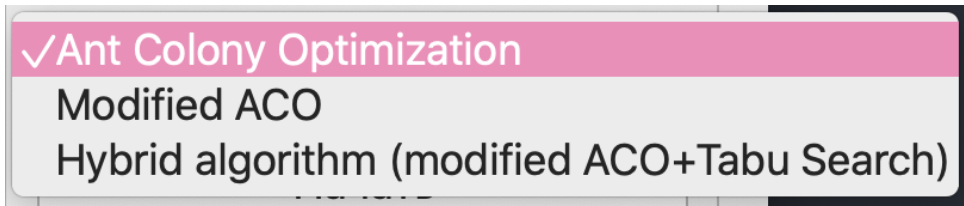


Рисунок 3.10 – Перелік алгоритмів для розрахунків

б) натиснути «Начать».

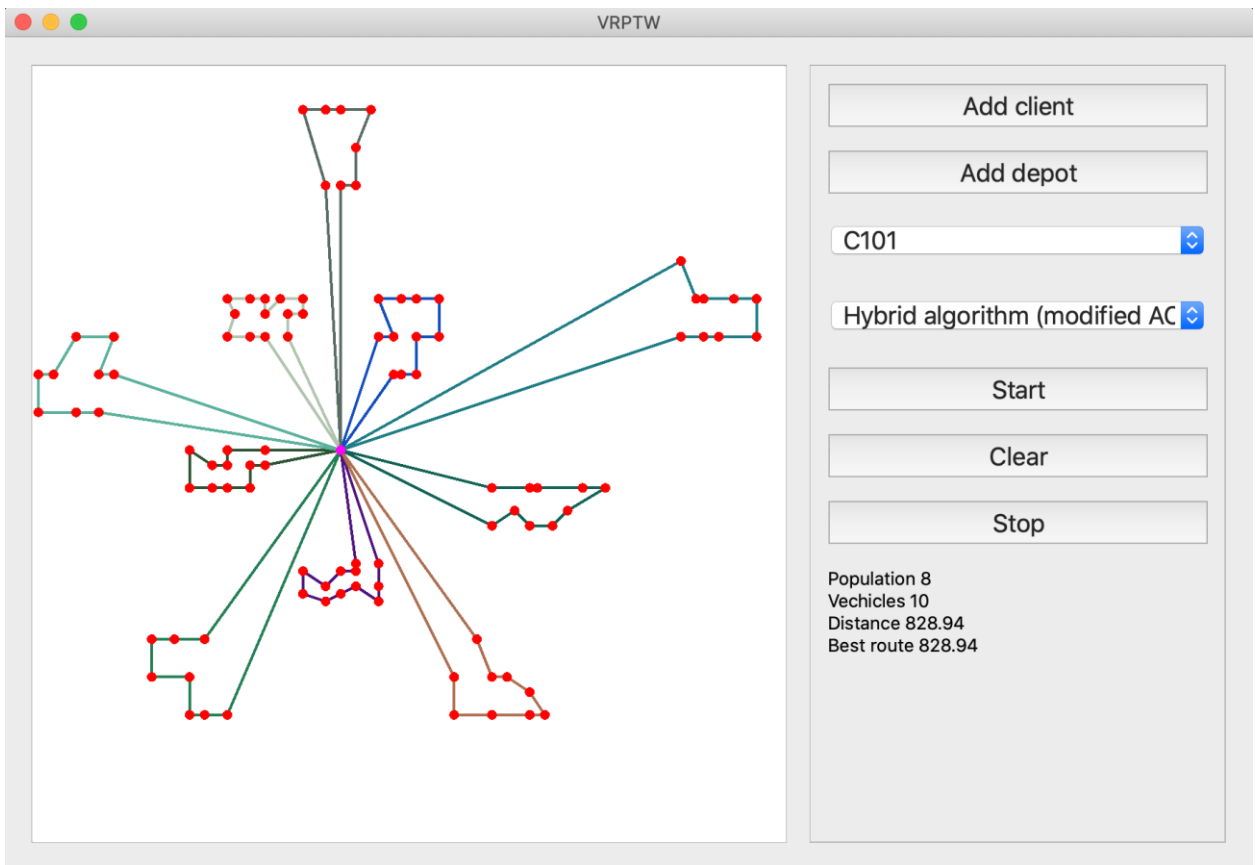


Рисунок 3.11 – Отримані результати.

У терміналі відображаються маршрути у текстовому вигляді (рисунок 3.12):

```
Problem C101 ALPHA 0.9 BETA 1.9 GAMMA 1.1 16 May 2021 16:57:52
algorithm work time 56.82956385612488
Distance 828.9368669428341 Vehicles 10
[0, 5] [5, 3] [3, 7] [7, 8] [8, 10] [10, 11] [11, 9] [9, 6] [6, 4] [4, 2] [2, 1] [1, 75] [75, 0] [0, 43] [43, 42] [42, 41] [41, 40] [40, 44] [44, 46] [46, 45] [45, 48] [48, 51] [51, 50] [50, 52] [52, 49] [49, 47] [47, 0] [0, 20] [20, 24] [24, 25] [25, 27] [27, 29] [29, 30] [30, 28] [28, 26] [26, 23] [23, 22] [22, 21] [21, 0] [0, 90] [90, 87] [87, 86] [86, 83] [83, 82] [82, 84] [84, 85] [85, 88] [88, 89] [89, 91] [91, 0] [0, 67] [67, 65] [65, 63] [63, 62] [62, 74] [74, 72] [72, 61] [61, 64] [64, 68] [68, 66] [66, 69] [69, 0] [0, 98] [98, 96] [96, 95] [95, 94] [94, 92] [92, 93] [93, 97] [97, 100] [100, 99] [99, 0] [0, 57] [57, 55] [55, 54] [54, 53] [53, 56] [56, 58] [58, 60] [60, 59] [59, 0] [0, 32] [32, 33] [33, 31] [31, 35] [35, 37] [37, 38] [38, 39] [39, 36] [36, 34] [34, 0] [0, 13] [13, 17] [17, 18] [18, 19] [19, 15] [15, 16] [16, 14] [14, 12] [12, 0] [0, 81] [81, 78] [78, 76] [76, 71] [71, 70] [70, 73] [73, 77] [77, 79] [79, 80] [80, 0]
```

Рисунок 3.12 – Маршрути

Висновки до розділу

У даному розділі було описано програмну реалізацію гібридного алгоритму модифікованого алгоритму оптимізації мурашиних колоній та табу пошуку, обґрунтовано обрану мову програмування, наведено опис бібліотек, класів та специфікацій функцій, використаних при розробці, надано опис покрокової інструкцію взаємодії з програмним продуктом.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1 Мета досліджень

Метою досліджень є оцінка ефективності розроблених модифікацій мурашиного алгоритму та розробленого гібридного алгоритму для розв'язання задачі VRPTW.

4.2 Тестові данні

Для дослідження результатів розробленого гібридного алгоритму було використано класичний набір з 56 задач, розроблених М. Solomon у 1987 році. Бенчмарк складається з шести різних типів задач (C1, C2, R1, R2, RC1, RC2). Кожен набір даних містить від восьми до дванадцяти задач розмірністю 25,50,100 вузлів. Задачі типу C характеризується кластеризованими клієнтами, часові вікна яких були створені на основі відомого розв'язку. Задачі типу R характеризуються сгенерованими рівномірно випадковим чином за площею клієнтами. Задачі типу RC поєднують випадково розміщених і кластеризованих клієнтів. Типи задач 1 мають вузькі часові вікна та невелику потужність автомобіля. Типи задач 2 мають широкі часові вікна та велику потужність автомобіля. Тому розв'язки задач типу 2 вміщують меншу кількість маршрутів із більшою кількістю клієнтів на маршрут, ніж у задачах типу 1 [6].

Експерименти здійснюються шляхом виконання трьох підходів для кожної задачі, які зупиняються після встановленого часу обчислення. Розв'язки усереднюються для кожного типу задач, результат додається в таблицю. На рисунку 4.1 зображено приклад файлу з задачею C101, де використовуються номери, координати, потреби у товарі, час обслуговування та часові вікна клієнтів.

C101							
VEHICLE NUMBER 25		CAPACITY 200					
CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE	TIME
0	40	50	0	0	1236	0	
1	45	68	10	912	967	90	
2	45	70	30	825	870	90	
3	42	66	10	65	146	90	
4	42	68	10	727	782	90	
5	42	65	10	15	67	90	
6	40	69	20	621	702	90	
7	40	66	20	170	225	90	
8	38	68	20	255	324	90	

Рисунок 4.1 – Задача 101 з бенчмарку Соломона

Даний метод тестування актуальний для розробників зі всього світу. Він дозволяє порівнювати результати з вже відомими кращими розв'язками, оцінювати швидкість алгоритму та рівень відхилення від оптимального розв'язку.

4.3 Результати досліджень ефективності запропонованих модифікацій алгоритму мурашиних колоній

Для тестування роботи алгоритму спочатку необхідно визначити ефективні значення коефіцієнтів, що відповідають за рівень випаровування феромону, залежність маршруту від феромону, відстані до наступного клієнта, часового вікна. Для кожної задачі було підібрано експериментальним шляхом найбільш вдалі коефіцієнти, на основі цього дослідження обрано усереднені значення для кожного класу задач Соломона.

В ході досліджень було сгенеровано 3312 варіантів коефіцієнтів α, β, γ для кожної задачі розмірності 25 від 0.7 до 3 з кроками 0.1, 0.2, 0.3 відповідно. Кожен з варіантів було протестовано та експериментально знайдено кращий розв'язок та середнє значення коефіцієнтів для усіх класів. (0.9, 1.9, 1.1). Асо є стохастичним алгоритмом, середнє квадратичне відхилення результатів 12.6.

В результаті проведених експериментальних досліджень, було встановлено, що для задач Соломона, покращений АСО знаходить розв'язок

краще в середньому на 15% ніж ACO, а гібридний алгоритм в середньому на 23% .

Дослідження швидкості роботи алгоритмів показало, що швидкість роботи алгоритмів залежить від класу задач (рисунок 4.2), розмірності задач (рисунок 4.3) та константних параметрів алгоритмів, наприклад – кількості особин популяції мурах (рисунок 4.4) та кількості ітерацій.

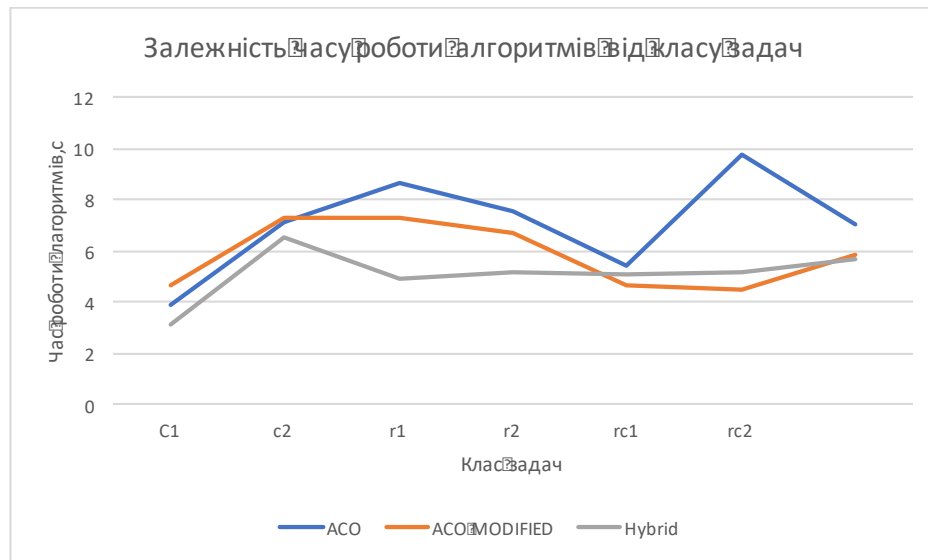


Рисунок 4.2 – Графік залежності часу роботи алгоритмів від класу задач на прикладі розмірності 25.

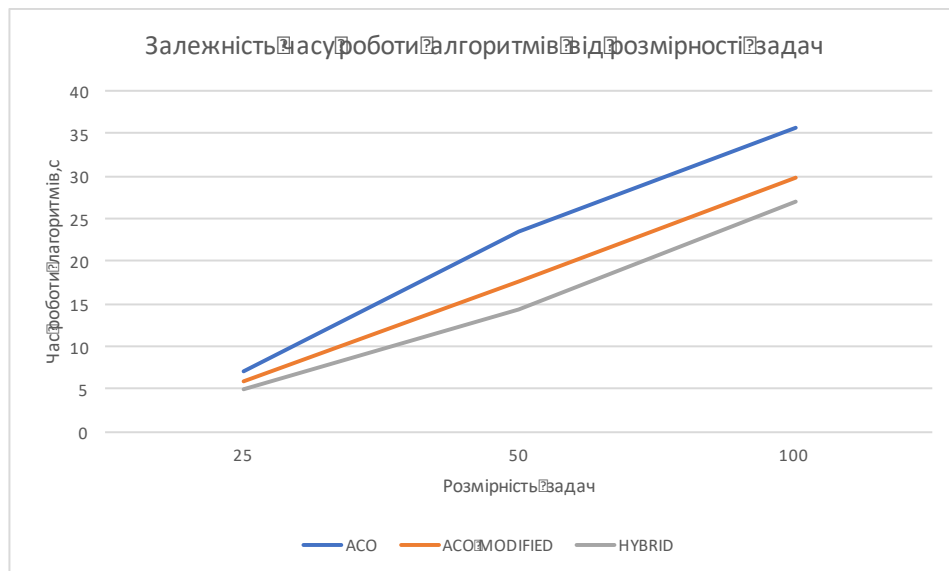


Рисунок 4.3 – Графік середнього часу знаходження розв'язку алгоритмами для задач розмірності 25, 50, 100

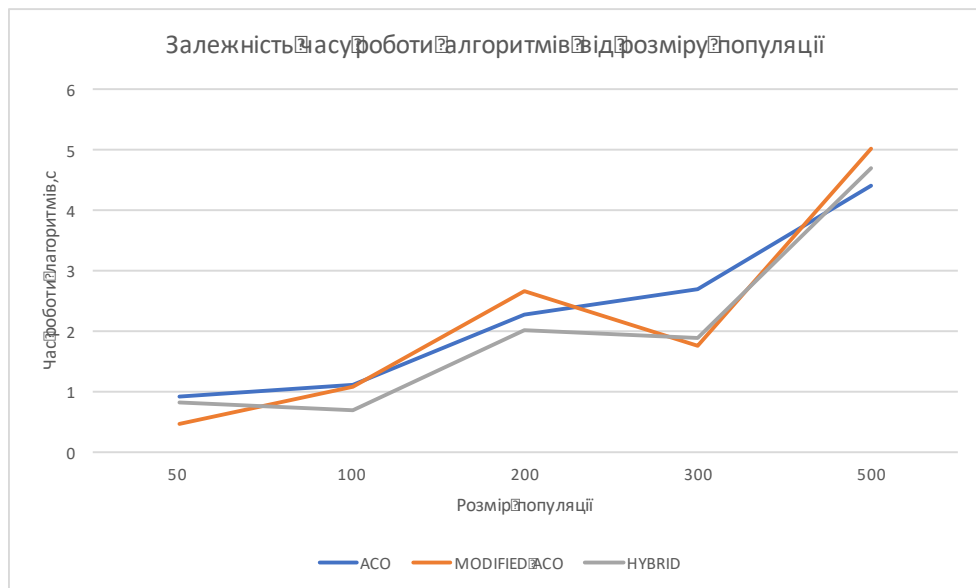


Рисунок 4.4 – Графік залежності часу роботи алгоритмів від кількості особин популяції (50, 100, 200, 300, 500) на прикладі задачі C102.

Наступним етапом експериментальних досліджень стало порівняння результатів роботи гібридного алгоритму з всесвітньо відомими кращими результатами по загальній довжині побудованого маршруту та кількості задіяних ТЗ. Результати досліджень наведені в таблиці 4.1.

Таблиця 4.1 – Результати експериментів

Клас тестових задач	Оптим. Розв'язок		АСО + евристики		Відхилення
	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ	
R101.25	617,1	8	653,2	8	5,85
R102.25	547,1	7	578,3	8	5,70
R103.25	454,6	5	486,7	5	7,06
R104.25	416,9	4	443,2	4	6,31
R105.25	530,5	6	565,2	6	6,54
R106.25	465,4	3	498,9	5	7,20
R107.25	424,3	4	458,2	4	7,99
R108.25	397,3	4	428,8	4	7,93
R109.25	441,3	5	481,7	5	9,15
R110.25	444,1	4	480,9	5	8,29
R111.25	428,8	5	466,9	4	8,89
R112.25	393	4	423,9	4	7,86
R201.25	463,3	4	531,4	2	14,70
R202.25	410,5	4	456,2	4	11,13
R203.25	391,4	3	449,3	2	14,79

Продовження таблиці 4.1

Клас тестових задач	Оптим. розв'язок		АСО + евристики		Відхилення
	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ	
R204.25	355	2	399,7	2	12,59
R205.25	393	3	453,2	2	15,32
R206.25	374,4	3	421,8	2	12,66
R207.25	361,6	3	416,7	2	15,24
R208.25	328,2	1	353,7	1	7,77
R209.25	370,7	2	417,5	2	12,62
R210.25	404,6	3	472,6	2	16,81
R211.25	350,9	2	413,1	1	17,73
C101.25	191,3	3	191,8	3	0,26
C102.25	190,3	3	196,5	3	3,26
C103.25	190,3	3	199,5	3	4,83
C104.25	186,9	3	193,9	3	3,75
C105.25	191,3	3	191,8	3	0,26
C106.25	191,3	3	191,8	3	0,26
C107.25	191,3	3	192,6	3	0,68
C108.25	191,3	3	191,8	3	0,26
C109.25	191,3	3	193,2	3	0,99
C201.25	214,7	2	215,5	2	0,37
C202.25	214,7	2	223,1	1	3,91
C203.25	214,7	2	221,2	2	3,03
C204.25	213,1	2	223,5	2	4,88
C205.25	214,7	2	215,5	2	0,37
C206.25	214,7	2	231,3	2	7,73
C207.25	214,5	2	232,2	2	8,25
C208.25	214,5	2	216,4	2	0,89
RC101.25	461,1	4	462,6	4	0,33
RC102.25	351,8	3	371,3	4	5,54
RC103.25	332,8	3	353,8	4	6,31
RC104.25	306,6	3	324,9	3	5,97
RC105.25	411,3	4	439,1	5	6,76
RC106.25	345,5	3	361,3	3	4,57
RC107.25	298,3	3	308,2	3	3,32
RC108.25	294,5	3	310,6	3	5,47
RC201.25	360,2	3	391,6	3	8,72
RC202.25	338	3	367,3	2	8,67
RC203.25	326,9	3	363,4	2	11,17
RC204.25	299,7	3	331,9	2	10,74
RC205.25	338	3	361,9	3	7,07

Продовження таблиці 4.1

Клас тестових задач	Оптим. розв'язок		АСО + евристики		Відхилення
	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ	
RC206.25	324	3	370,3	2	14,29
RC207.25	298,3	3	334,3	2	12,07
RC208.25	269,1	2	313,6	1	16,54

Було проведено дослідження ефективності у порівнянні з іншими алгоритмами, зокрема з PCP (Precedence Constraint Posting) методом, викладеним в роботі [66] та гібридним генетичний алгоритмом, описаним в статі [67]. Результати порівнянь наведено в таблиці 4.2

Таблиця 4.2 – Порівняння з іншими алгоритмами

Клас тестових задач	Кращий відомий розв'язок		Розроблений гібридний алгоритм		рсп		GA-PCP algorithm	
	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ	Загальна довжина маршруту	Кількість задіяних ТЗ
C104.25	186,9	3	193,9	3	189,8	2	186,9	3
R104.25	416,9	4	443,2	4	467,8	3	449,3	5
RC108.25	294,5	3	310,6	3	300,4	5	294,7	5
C204.25	213,1	2	223,5	2	290,9	2	282,05	2
R208.25	328,2	1	353,7	1	332	2	328,5	2
RC208.25	269,1	1	313,6	1	285	2	270,9	2

Запропонований гібридний алгоритм (див. п. 2.5.3) експериментально порівнювався з вже відомими результатами [56] роботи алгоритмів АСО та Табу пошуку на тестових задачах Соломона [6], що зображено в таблиці 4.3.

Таблиця 4.3 – Порівняння розв'язків алгоритмів

	Кращий відомий розв'язок	Алгоритм мурашиних колоній	Табу пошук	Розроблений гібридний алгоритм
	Загальна довжина маршруту			
C1.100	826,7	851,47	844,22	840,55
C2.100	589,24	628,47	609,11	604,12
R1.100	1156,36	1396,55	1233,65	1201,24
R2.100	932,35	1007,58	960,25	958,11
RC1.100	1368,23	1167,35	1422,18	1408,14
RC2.100	1075,67	1165,87	1121,39	1112,24

4.4 Аналіз результатів досліджень

Не зважаючи на те, що час знаходження та прийнятність розв'язку безпосередньо залежить від кількості мурах та ітерацій та коефіцієнтів, що відповідають за рівень випаровування феромону, залежність маршруту від феромону, відстані до наступного клієнта, часового вікна, очевидним є висновок, що цей час вдалось зменшити за допомогою модифікації АСО алгоритму та розробленого на базі нього гібридного алгоритму.

Тож, в результаті розробки гібридного алгоритму модифікованого АСО та Табу пошуку, вдалося досягнути знаходження розв'язку задачі за менший час, у порівнянні з класичним АСО алгоритмом та іншими запропонованими евристичними для VRPTW.

Основний висновок, який був отриманий в результаті досліджень гібридного алгоритму з оптимальними та всесвітньо відомими кращими наближеними результатами по загальній довжині побудованого маршруту та кількості задіяних ТЗ, це те, що алгоритм має різну ефективність для різних класів тестових задач. Так, було встановлено середній рівень відхилення розв'язку гібридного алгоритму від кращого відомого розв'язку по загальній довжині маршруту:

- для класу задач R1 на 7,4%;
- для класу задач R2 на 13,76%;
- для класу задач C1 на 1,62%;
- для класу задач C2 на 3,68%;
- для класу задач RC1 на 4,78%;
- для класу задач RC2 на 11,16%.

По кількості задіяних ТЗ на маршруті розроблений гібридний алгоритм давав наступні результати в порівнянні з кращими відомими розв'язками:

- для класу задач R1 такий самий результат в 67% та кращий результат у 8%;

- для класу задач R2 на такий самий результат в 36,4% та кращий результат у 63,6%;
- для класу задач C1 такий самий на 100%;
- для класу задач C2 такий самий результат в 88% та кращий результат у 12%;
- для класу задач RC1 такий самий на 63%;
- для класу задач RC2 такий самий результат в 25% та кращий результат у 75%.

Ефективність запропонованого гібридного алгоритму підтвердилась дослідженнями, які показали, що ефективність результатів роботи алгоритму підвищується, якщо об'єднати та модифікувати кілька метаевристик. Розробка гібридних алгоритмів є перспективним напрямком, оскільки вони поєднують в собі переваги використаних евристик. Тож, за результатами експериментальних досліджень, було встановлено, що гібридний алгоритм може досягти розв'язку, який є близьким до широко відомих кращих розв'язків для задач усіх класів.

Висновки до розділу

Було проведено аналіз розроблених модифікованого алгоритму оптимізації мурашиних колоній та гібридного алгоритму, що базується на модифікованому АСО та Табу пошуку. За результатами експериментальних досліджень було встановлено, що модифікація АСО позитивно впливає на роботу АСО, а запропонований на базі нього гібридний алгоритм є досить ефективним і дає співставні розв'язки у порівнянні з відомими кращими розв'язками. В результаті, було оцінено ефективність розробленого гібридного алгоритму та модифікації АСО для розв'язання задачі VRPTW, а як наслідок, була досягнута мета досліджень.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї стартап-проекту

Аналіз змісту ідеї та загальних напрямків використання стартап-проекту “Інформаційна система планування обслуговування доріг міста з врахуванням часових вікон”, наведено у таблиці 5.1.

Таблиця 5.1 – Зміст ідеї стартап-проекту

Ідея	Можливі застосування	Вигоди для користувачів
Автоматизація процесу планування обслуговування доріг міста з врахуванням часових вікон, який складається з побудови маршрутів для таких видів робіт:	1. Очищення дорожнього покриття від сміття, бруду, сторонніх предметів.	1. Забезпечення нормального руху транспортних засобів.
	2. Мийка дорожнього покриття та очищення від пилу.	2. Підтримання стану дорожнього покриття.
	3. Нанесення та відновлення зношеної розмітки.	3. Уникнення аварійних ситуацій.
	4. Розчищення автомобільних доріг від снігу.	4. Відновлення транспортного руху в найкоротші терміни.
	5. Розподілення протиожедних матеріалів.	5. Надання можливості безпечного руху транспорту.
	6. Ремонтні роботи на дорожньому полотні.	6. Ремонткування дорожнього покриття.

Слід зазначити, що в загальному застосування інформаційної системи планування обслуговування доріг міста в усіх напрямках з врахуванням часових вікон забезпечує для користувачів більш ефективно та економічно вигідне планування процесів, що пов’язані з різними видами робіт на дорогах, та надає можливість задавати певні часові проміжки, що, дозволить, наприклад, мінімізувати перешкоду міському руху робочим транспортом під час заторів.

Після огляду існуючих аналогів для подальшого порівняння було обрано Ecofleet, Maanteeamet та Транснавігація.

Ecofleet – це міжнародна компанія, що спеціалізується на розробці програмних продуктів для управління автомобільним парком і робочою командою на базі GPS. Ecofleet надає можливість оптимізації, обліку та документації роботи, а також мінімізації управлінського навантаження.

Enterprise resource planning - це програмне забезпечення для управління бізнес-процесами, яке управляє та інтегрує фінансову діяльність компанії, ланцюжок поставок, операції, звітність, виробництво та діяльність з персоналом.

Транснавігація – автоматизована система навігаційного диспетчерського контролю за роботою машин і механізмів для утримання автомобільних доріг. Система забезпечує контроль за використанням дорожньої техніки, облік і аналіз виконання планів по обслуговуванню доріг і магістралей, регулювання відхилень, що виникають при виконанні заданих робіт.

Визначимо відмінності інформаційної системи планування обслуговування доріг міста з урахуванням часових вікон від конкурентів та її техніко-економічні переваги і недоліки. В таблиці 5.2 проаналізовано чим відрізняється стартап-проект від існуючих аналогів.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Потенційні товари/концепції конкурентів				W ⁵	N ⁶	S ⁷
		1	2	3	4			
1.	Доступність в Україні	+	-	-	-			S
2.	Обслуговування міських вулиць	+	+	+	-		N	
3.	Обслуговування магістралей країни	-	+	+	+	W		

Продовження таблиці 5.2

№	Техніко-економічні характеристики ідеї	Потенційні товари/концепції конкурентів				W ⁵	N ⁶	S ⁷
		1	2	3	4			
4.	Динамічне задання депо і доріг	+	-	-	-			S
5.	Зимове утримання доріг	+	+	-	+		N	
6.	Роботи по розмітці доріг	+	-	-	+		N	
7.	Мийка та підмітання дорожнього покриття	+	+	-	+		N	
8.	Розрахунок найкращого маршруту	+	-	-	-			S
9.	Врахування директивних термінів та часових вікон	+	-	-	-			S
10.	Врахування пріоритетності доріг	+	-	-	-			S
11.	Відображення фактичного місцезнаходження транспортних засобів	-	+	+	+	W		
12.	Врахування дорожніх заторів в реальному часі	+	-	-	-			S
13.	Документування плану виконання робіт	+	+	+	+		N	
14.	Підрахунок вартості виконаних робіт	+	+	-	+		N	
15.	Формування звітності по завершенню виконання робіт	-	+	+	+	W		

Позначення до таблиці 5.2:

1 – Інформаційна система планування обслуговування доріг міста;
 2 – Ecofleet; 3 – Транснавігація; 4 – Maanteeamet; 5 – Слабка сторона; 6 –
 Нейтральна сторона; 7 – Сильна сторона.

5.2 Технологічний аудит ідеї стартап-проекту

Визначимо перелік технологій, за допомогою яких можна реалізувати ідею стартап-проекту (таблиця 5.3).

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	№ п/п	Технології її реалізації	Наявність технологій	Доступність технологій
Автоматизація процесу планування обслуговування доріг міста.	1.	JavaScript – мова програмування.	Наявна	Доступна
	2.	Node.js – серверна платформа.	Наявна	Доступна
	3.	React.js – бібліотека для розробки інтерфейсів користувача.	Наявна	Доступна
	4.	Leaflet – бібліотека для роботи з інтерактивною картою.	Наявна	Доступна
	5.	MySQL – реляційна система управління базами даних.	Наявна	Доступна
	6.	Алгоритм побудови маршрутів.	Необхідно розробити	Доступна
	7.	OpenStreetMap – відкрита географічна карта світу.	Наявна	Доступна
	8.	Mapbox Directions API – API для створення маршрутизації по карті від Mapbox.	Наявна	Доступна
	9.	GoogleMaps API – картографічний сервіс.	Наявна	Недоступна

Продовження таблиці 5.3

Ідея проекту	№ п/п	Технології її реалізації	Наявність технологій	Доступність технологій
	10.	Google Directions API – API для створення маршрутизації по карті від Google.	Наявна	Недоступна
	11.	Google Sheets API – онлайн-сервіс для роботи з таблицями Excel.	Наявна	Доступна
<i>Обрані технології для реалізації ідеї проекту:</i> JavaScript, Node.js, React.js, Leaflet, MySQL, Алгоритм побудови маршрутів, OpenStreetMap, Mapbox Directions API, Google Sheets API.				

Такі технології, як GoogleMaps API та Google Directions API, недоступні для даного стартап-проекту через їхнє платне використання.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначимо потенційні групи майбутніх користувачів інформаційної системи планування обслуговування доріг міста. У таблиці 5.4. описано потреби кожної з можливих груп.

Таблиця 5.4 – Характеристика потенційних користувачів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Підвищення ефективності обслуговування доріг	1. Державні підприємства («Укравтодор»)	На сьогоднішній день в Україні серед підприємств, що займаються обслуговуванням доріг, переважають державні та муніципальні підприємства. Така ситуація склалася через грошові проблеми оплати праці підрядників: приватний бізнес, на відміну від держпідприємств, не готовий до затримки оплати.	– невисока вартість товару; – зручність та простота у використанні; – ефективність впровадження; – швидкість роботи; – доступність у будь-який час.
2.	Зменшення витрат	2. Муніципальні підприємства («Київавтодор»)		
3.	Зменшення пройденої відстані	3. Приватні підприємства		
4.	Зниження часу виконання дорожніх робіт			
5.	Врахування часових обмежень			

Далі необхідно розглянути фактори, що будуть сприяти впровадженню стартап-проекту на потенційний ринок, та фактори, що можуть цьому перешкодити.

Фактори можливостей:

- розширення мережі доріг;
- збільшення кількості транспортних засобів на дорогах;

- зменшення дорожніх перешкод;
- спрощення роботи обслуговуючих підприємств;
- необхідність підвищення якості та швидкості виконання робіт;
- економічна доцільність;
- вилучення людського фактору та упередженості через автоматизацію процесів;
- потреба у зменшенні управлінського навантаження.

Фактори загроз:

- сталі процеси та небажання вводити інновації;
- відсутність фінансування у підприємства;
- невелика ділянка чи обсяг робіт, що виконує підприємство.

Як можна побачити, наразі виведення на ринок інформаційної системи планування обслуговування доріг міста є достатньо актуальним. Наступним кроком є дослідження умов конкуренції на ринку (таблиця 5.5).

Таблиця 5.5 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції: монополія	З проаналізованих раніше існуючих аналогів, можна зробити висновок, що в Україні немає іншої подібної системи, яку могли б використовувати підприємства як альтернативу до стартап-проекту, що розробляється.	Стартап-проект не має близьких конкурентів, тому це спрощує його вихід на ринок.
2. Рівень конкурентної боротьби: національна	Може використовуватися підприємствами з будь-яких міст України.	На даному етапі демонструється робота системи на певному районі міста Києва. В подальшому потрібно

Продовження таблиці 5.5

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
		розширювати зону впровадження системи.
3. Галузева ознака: міжгалузева	Система є програмним продуктом, призначеним для використання у галузі обслуговування доріг.	Потрібно ретельно дослідити галузі, що пов'язані зі стартап-проектом.
4. Конкуренція за видами товарів: товарно-видова	Замість системи, що розробляється, можливо використовувати тільки систему того ж самого призначення, що буде виконувати схожі функції.	При розробці системи потрібно врахувати якомога більше особливостей сфери діяльності та додати функції, яких немає у систем схожого призначення.
5. Характер конкурентних переваг: нецінова	Конкуренція відбувається в неціновому форматі за рахунок функціональних можливостей та ефективності використання.	Потрібно підвищувати якість та надійність роботи системи, а також перелік наданих можливостей.
6.Інтенсивність не марочна	Оскільки це стартап, то проект є новим та не має відомої торговельної марки.	Потрібно займатися розкруткою стартап-проекту, щоб зацікавити цільову аудиторію.

З огляду на поточну конкурентну ситуацію можна зробити висновок, що інформаційна система планування обслуговування доріг міста з урахуванням часових вікон може бути конкурентоспроможним та затребуваним товаром поміж своєї цільової аудиторії.

На основі попередньо опрацьованої інформації проведемо аналіз факторів конкурентоспроможності інформаційної системи планування обслуговування доріг міста з урахуванням часових вікон (таблиця 5.6).

Таблиця 5.6 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Доступність системи на території України	Відсутність подібних аналогів на території України.
2.	Зручність, простота і наочність у використанні	Не потребує додаткового навчання та залучення нових робітників, мінімізація людського фактору.
3.	Якісне планування маршрутів	Мінімізація часу проходження транспортними засобами всіх доріг.
4.	Врахування часових обмежень	Директивні терміни є природніми обмеженнями в реальних умовах, проте більшість програмних застосувань не мають можливості враховувати такі обмеження.
5.	Поділ доріг згідно їх пріоритетів	Зазвичай райони міста поділяються на декілька зон відповідно до їх значущості для надання якнайкращого сервісу.
6.	Врахування інтенсивності дорожнього руху в реальному часі	Ефективніше планування маршрутів з врахуванням поточної ситуації на дорогах або швидке корегування маршруту в разі непередбачуваних ситуацій (аварія, затор тощо).

Завершальним етапом аналізу ринкових можливостей запуску стартап-проекту є SWOT-аналіз, що наведено у таблиці 5.7.

Таблиця 5.7 – SWOT-аналіз стартап-проекту

<i>Сильні сторони:</i> – доступність системи; – зручність, простота і наочність у використанні; – якісне планування маршрутів; – врахування часових обмежень; – ранжування доріг згідно їх пріоритетів; – врахування інтенсивності дорожнього руху. <i>Можливості:</i> – збільшення обсягів робіт, що виконуються підприємством; – зниження складності планування маршрутів; – уникнення збитків на підприємстві через некоректне планування маршрутів; – запобігання допущення помилок працівниками, що займаються складанням планів маршрутів; – відсутність потреби у додаткових фахівцях, що займаються плануванням маршрутів.	<i>Слабкі сторони:</i> – немає можливості відслідковувати поточне місцезнаходження транспортних засобів; – обмеженість у використанні лише на міських дорогах; – відсутність звітності та аналізу; – новий проект, що не має відомої марки. <i>Загрози:</i> – сформовані процеси планування на підприємстві не підлягають внесенню змін; – нестача коштів на підприємстві для виконання робіт; – використання додаткового інструменту планування не має сенсу через невеликі обсяги робіт та/або ділянки обслуговування.
---	---

5.4 Розроблення ринкової стратегії проекту

Для розроблення ефективної ринкової стратегії стартап-проекту перш за все визначимо характеристики цільових груп можливих споживачів (таблиця 5.8). Потенційними клієнтами для інформаційної системи планування

обслуговування доріг міста є підприємства, що займаються виконанням різних видів робіт на міських вулицях.

Таблиця 5.8 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Державні підприємства	Готовність до сприйняття нижче середнього.	Відповідальні за обслуговування доріг лише державного значення (за межами населених пунктів). На теперішній момент система призначена для використання на міських дорогах. Проте у веденні держпідприємств всі рівні дорожніх організацій країни, тому можливий низький рівень попиту.	Висока.	Складний вхід у сегмент через високий рівень значущості підприємств.
2.	Муніципальні підприємства	Готовність до сприйняття нижче середнього, оскільки рідко	Виконують 80% робіт по обслуговуванню доріг місцевого значення (дороги вулиць міст та ін. населених пунктів), тому рівень попиту	Висока.	Вхід у сегмент середньої важкості через те, що на цих

Продовження таблиці 5.8

п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
		впроваджують нові підходи.	може бути досить високим.		підприємствах довготривало встановлені процеси роботи, до яких не вносяться зміни.
3.	Приватні підприємства	Висока готовність до впровадження нових рішень, бо прагнуть працювати краще та ефективніше.	Виконують 20% робіт по обслуговуванню доріг місцевого значення, тому рівень попиту є середнім.	Висока.	Досить простий вхід у сегмент, бо такі підприємства зазвичай прагнуть отримати якнайбільший прибуток.

Обрано наступні цільові групи: муніципальні та приватні підприємства через досить високий рівень попиту, простоту входу у сегмент та готовність до сприйняття продукту.

Обрана стратегія охоплення ринку – стратегія диференційованого маркетингу, а також визначена базова стратегія розвитку як стратегія спеціалізації, що не передбачає охоплення всього ринку, а навпаки зосередження на конкретних цільових сегментах та задоволенні їхніх потреб.

Далі визначимо базову стратегію конкурентної поведінки. Стартап-проект буде залучати до своєї інформаційної системи нових користувачів, а не переманювати існуючих у конкурентів. Хоча основні характеристики стартап-проекту будуть відповідати реалізованим можливостям у конкурентів, все ж таки у системі буде розроблено ряд інноваційних рішень, які зроблять її відмінною від аналогів. З огляду на це, а також на те, що проект буде новатором на ринку, можна зробити висновок, що кращою стратегією конкурентної поведінки буде стратегія заняття конкурентної ніші. В цьому разі обирається декілька ринкових сегментів та формується прихильність споживачів, розвиваються відмінні властивості системи, що відповідають потребам та попиту на ринку.

На основі вже визначених рішень щодо ринкової поведінки сформуємо комплекс ключових асоціацій, що дозволять споживачам ідентифікувати позицію стартап-проекту:

- якісне планування маршрутів для обслуговування міських доріг завдяки мінімізації часу проходження транспортними засобами шляхів сполучень;
- врахування суттєвих обмежень щодо директивних термінів виконання робіт, часових вікон та місткості машин;
- підвищення ефективності планування через корегування маршрутів відповідно до інтенсивності дорожнього руху в реальному часі.

5.5 Розроблення маркетингової програми стартап-проекту

У таблиці 5.9 описано маркетингову концепцію стартап-проекту.

Таблиця 5.9 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1.	Виконання якісного обслуговування доріг.	Підвищення ефективності обслуговування доріг.	Корегування маршрутів в реальному часі у разі заторів на дорогах, аварій тощо.
2.	Підрахунок вартості виконаних робіт.	Зменшення витрат та уникнення збитків.	Мінімізація вартості обслуговування в результаті роботи алгоритму.
3.	Підрахунок пройденої відстані.	Зменшення пройденої відстані.	Мінімізація пройденої відстані обслуговування в результаті роботи алгоритму.
4.	Підрахунок часу виконання робіт.	Зниження часу виконання дорожніх робіт.	Мінімізація часу обслуговування в результаті роботи алгоритму.
5.	Врахування обмежень, що виникають під час обслуговування доріг.	Врахування часових обмежень та місткості машин.	Можливість розділяти дороги за класами пріоритетів та встановлення їм у відповідність директивних термінів, врахування потреби поновлювати ресурси.
6.	Пришвидшення процесу планування маршрутів.	Зниження складності планування маршрутів.	Автоматизованість процесу планування, а також зручність, простота і наочність у використанні системи.

Стартап-проект буде захищено від копіювання через захист прав на інтелектуальну власність. А далі розробимо концепцію маркетингових комунікацій (таблиця 5.10).

Таблиця 5.10 – Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Щоденне планування маршрутів.	– електро пошта; – телефонний зв'язок; – особисті зустрічі; – презентації на	Якісне та швидке складання планів маршрутів.	Надання інформації щодо існування інструменту для планування маршрутів.	Зручність, швидкість і простота системи у використанні.
2.	Комунікація диспетчера з водіями транспортних засобів для видачі планів.	тематичних заходах, конференціях тощо.	Полегшення керування персоналом.	Переконання у полегшенні праці робітників та уникнення людського фактору.	Передача та оновлення планів у онлайн режимі, відсутність потреби у додаткових фахівцях.
3.	Повільна модернізація процесів роботи.		Автоматизація процесу планування маршрутів.	Схиляння до рішення про покупку.	Переконання у необхідності впровадження нових підходів до роботи.

Продовження таблиці 5.10

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
4.	Отримання більшого прибутку.		Мінімізація вартості обслуговування.	Інформування про ефективність використання розробленого алгоритму.	Демонстрація переваг алгоритму.
5.	Покращення обслуговування доріг міста.		Врахування обмежень.	Нагадування про існування у реальній практиці природніх обмежень.	Можливість розподілу доріг за класами пріоритетів та встановлення відповідних їм директивних термінів.
6.	Пришвидшення обслуговування доріг міста.		Мінімізація часу обслуговування та пройденої відстані.	Інформування про ефективність використання розробленого алгоритму.	Демонстрація роботи алгоритму.

Продовження таблиці 5.10

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
7.	Прийняття рішень щодо зміни маршрутів у разі непередбачуваних ситуацій (затори, аварії тощо).		Корегування маршрутів в реальному часі у випадку непередбачуваних ситуацій.	Надання інформації про врахування у системі інтенсивності дорожнього руху.	В разі зміни маршруту водії можуть бачити оновлений план якнайшвидше онлайн.

Висновки

Після проведеного аналізу було визначено, ринкова реалізація стартап-проекту можлива та доцільна. Як показав аналіз цільових груп потенційних споживачів найбільший рівень попиту присутній у двох сегментах ринку: муніципальні підприємства та приватні підприємства, що займаються обслуговуванням та ремонтом доріг. Саме тому потрібно використовувати стратегію зосередження на конкретних цільових групах та задоволенні їхніх потреб. Інформаційна система планування обслуговування доріг міста є конкурентоспроможним проектом та має перспективи для впровадження, хоча в деяких групах потенційних клієнтів наявний досить високий бар'єр входу у сегмент. З огляду на те, що проект буде новатором на ринку, а також, що склалися сприятливі можливості для виведення на ринок даної системи, можна зробити висновок про доцільність подальшої імплементації проекту.

ВИСНОВКИ

Магістерська дисертація присвячена задачі маршрутизації транспортних засобів з урахуванням часових вікон. Метою є мінімізація сумарної вартості транспортних перевезень продукції споживачам з урахуванням часових вікон.

У першому розділі здійснено аналіз задачі маршрутизації транспортних засобів та надано огляд існуючих різновидів задач і методів їх розв'язання. Було наведено перелік програмних продуктів, що вирішують близькі до задачі маршрутизації транспортних засобів. З метою збільшення ефективності роботи відомих алгоритмів розв'язання даного типу задач, була сформульована відповідна постановка задачі та мета дослідження.

Другий розділ присвячено розробці модифікації алгоритму оптимізації мурашиних колоній та розробці гібридного алгоритму для задачі VRPTW. Було представлено математичну модель задачі VRPTW та метод її розв'язання. Для алгоритму мурашиних колоній було наведено узагальнену схему алгоритму, перераховані основні його недоліки та переваги. Було запропоновано модифікацію алгоритму АСО. Крім того, було розроблено гібридний алгоритм на базі модифікованого АСО та Табу пошуку для задачі маршрутизації транспортних потоків з урахуванням часових вікон.

У третьому розділі міститься опис програмного забезпечення. Крім того, розписано покрокова інструкція роботи з програмою, що розв'язує задачу маршрутизації транспортних засобів з урахуванням часових вікон.

У четвертому розділі наведено мету, результати та аналіз зроблених досліджень щодо ефективності запропонованих алгоритмів. За результатами експериментальних досліджень було встановлено, що запропоновані модифікований та гібридний алгоритми є досить ефективним і дає співставні розв'язки у порівнянні з кращими відомими розв'язками. В результаті, було оцінено ефективність розробленого гібридного алгоритму для розв'язання задачі VRPTW, а як наслідок, була досягнута мета досліджень. Дослідження підтвердили ефективність запропонованих модифікованого АСО та

гібридного алгоритмів. Модифікація АСО алгоритму підвищує ефективність класичного алгоритму, гібридний алгоритм може досягти розв'язку, який є близьким до широко відомих кращих розв'язків для задач різних типів.

У результаті роботи над магістерською дисертацією було:

- проаналізовано відомі результати розв'язання задачі маршрутизації транспортних засобів;
- розроблено модифікацію алгоритму оптимізації мурашиних колоній та гібридний алгоритм, який поєднує у собі модифікований АСО та табу пошук;
- розроблено програмну реалізацію алгоритмів;
- проведено дослідження ефективності розроблених алгоритмів.

РЕКОМЕНДАЦІЇ

Актуальною сферою для дослідження та покращення гібридного алгоритму є динамічні зміни константних параметрів (α, β) в процесі роботи алгоритму, а також експериментальне дослідження діапазону зміни феромону ρ , в межах якого буде досягатися найкращий розв'язок. Ефективним є мінімізація неефективних ітерацій та відсікання завідомо неоптимальних маршрутів.

Крім цього, якісним поліпшенням метаевристики є розробка гібридних алгоритмів, які поєднують в собі переваги більше двох метаевристик і дозволяє досягти суттєвого поліпшення результатів.

ПЕРЕЛІК ПОСИЛАНЬ

1. А. Д. Баранова, Е. Г. Жданова. Задача маршрутизації транспортних засобів з часовими вікнами // VI Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління»(ІСТУ-2021). – 2021. – Kyiv, Ukraine.
2. CSCMP's 30 th Annual State of Logistics Report and Presentation. // CSCMP's Annual State of Logistics Report ® and Presentation. – 2019. – №30.
3. G. B. Dantzig, J. H. Ramser. The Truck Dispatching Problem // INFORMS. – 1959. – p. 12.
4. G. Clarke, J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points // Operations Research. –1962. – pp. 568–581.
5. F. Glover, M. Laguna. Tabu Search // Springer. –1997. – p. 408.
6. B. Kallehauge, J. Larsen, O. B. G. Madsen, M. M. Solomon. Chapter 3 Vehicle routing problem with time windows //Column Generation Springer. –2005. – pp. 67–98.
7. Burak Eksioglu. The vehicle routing problem: A taxonomic review [Електронний ресурс] / Burak Eksioglu, Arif Volkan Vural, Arnold Reisman – Режим доступу до ресурсу: https://staff.fmi.uvt.ro/~daniela.zaharie/ma2018/projects/biblio/applications/VehicleRouting/VRP_taxonomicReview.pdf.
8. M. M. Solomon, J. Desrosiers. Time Window Constrained Routing and Scheduling Problems. –1988. – pp. 1–13.
9. M. M. Solomon. Algorithms for the VR and SP rith TM Constraints // vol. 35 . –1987. – no. 2. –pp. 254–265.
10. M. Dorigo, T. Stutzle. Ant Colony Optimization: Overview and Recent Advances // no. May. –2009. – pp. 1–34.
11. M. Dorigo, T. Stützle. Ant colony Optimization. – 2004. – p. 321.
12. B. Eksioglu, A. Volkan, A. Reisman. The vehicle routing problem : A taxonomic review // Computers & Industrial Engineering. – 2009. – vol. 57. – no. 4. – pp. 1472–1483.

13. Кубил Виктор Николаевич. ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ РЕШЕНИЯ МОГОКРИТЕРИАЛЬНЫХ ЗАДАЧ МАРШРУТИЗАЦИИ ТРАНСПОРТА НА ОСНОВЕ МУРАВЬИНОГО АЛГОРИТМА : дис. канд. техн. наук : 05.13.01 / Кубил Виктор Николаевич. – Новочеркасск, 2019. – 184 с.
14. A. Volkan, B. Eksioglu, A. Reisman. The vehicle routing problem : A taxonomic review // Computers & Industrial Engineering. – 2009. – pp. 1472–1483
15. C.-Y. Liong. Vehicle routing problem: Models and solutions / C.-Y. Liong, Khairuddin Omar. // Journal of Quality Measurement and Analysis. – 2008.
16. Braekers, K., Ramaekers, K., & Nieuwenh, I. V. (2016). The Vehicle Routing Problem: State of the Art Classification and Review. Computers & Industrial Engineering. ^[1]_{SEP}
17. Eduardo, U., Diego, P. (2017). New benchmark instances for the Capacitated Vehicle Routing Problem. European Journal of Operational Research.
18. Soysal, M., Bloemhof, J. M., & Bektas, T. (2015). The time-dependent two-echelon capaci- tated vehicle routing problem with environmental considerations. International Journal of Production Economics.
19. Ehsan, T., & Vahid, K. (2016). Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. Information Sciences.
20. Maurizio, B., & Ferdinando, P. (2015). A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows. Electronic Notes in Discrete Mathematics.
21. Ilker, K., & Nursel, O. (2015). An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. Computers & Industrial Engineering.
22. Bin, Y., & Zhi-Hua H. (2015). Routing with time-windows for multiple environmental vehicle types. Computers & Industrial Engineering.

23. Li, J., Li, Y., & Pardalos, P. M. (2016). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*.
24. Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*.
25. De Oliveira, F. B., Enayatifar, R., Sadaei, H. J., Guimarães, F. G., & Potvin, J. Y. (2016). A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem. *Expert Systems with Applications*.
26. Jairo, R. M.-T., Julian, L. F., & Santiago, N. I. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*.
27. Li, J., Li, Y., & Panos, M. P. (2016). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*.
28. Tania, R. P. R., Maria, I. G., & Ana B. P. (2019). Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*.
29. Chávez, J. J. S., Escobar, J. W., & Echeverri, M. G. (2016). A Multi-objective Pareto and Colony Algorithm for the Multi-depot Vehicle Routing Problem with Backhauls. *International Journal of Industrial Engineering Computations*.
30. Braekers, K., Ramaekers, K., & Nieuwenh, I. V. (2016). The Vehicle Routing Problem: State the Art Classification and Review. *Computers & Industrial Engineering*.
31. Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In B. L. Golden, S. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem: latest advances and new challenges* (pp. 3–27). Berlin: Springer.
32. Saso, K., & Vili, P. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*.

33. Mustafa, A., & Seyda, T. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering*.
34. Baozhen, Y., & Bin, Y. (2016). An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. *Annals of Operations Research*.
35. Meryem, B., & Abdelmadjid, B. (2016). Quantum Inspired Algorithm for a VRP with Heterogeneous Fleet Mixed Backhauls and Time Windows. *International Journal of Applied Metaheuristic Computing*.
36. Lijun, W., & Zhenzhen, Z. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*.
37. Emmanouil, E., & Christos, D. (2016). The Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Two-Dimensional Loading Constraints. *European Journal of Operational Research*.
38. Cagri, K., & Gilbert, L. (2018). Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*.
39. Sebastian, R., & Andreas, B. (2018). Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints. *European Journal of Operational Research*, 266, 3, 877–894.
40. N. A. El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods // *Journal of King Saud University - Science*. – 2010, Jul. – vol. 22. – no. 3. – pp. 123–131.
41. Manar Ibrahim Hosny. Investigating Heuristic and Meta-Heuristic Algorithms for Solving Pickup and Delivery Problems : дис. докт. філос. наук / Manar Ibrahim Hosny. – Cardiff, 2010. – 242 с.
42. Плітка Олександр. Дослідження і розробка методу визначення об'єкта на зображенні з допомогою алгоритму муравьиной колонії [Електронний ресурс] / Плітка Олександр. – 2018. – Режим доступу до ресурсу: <http://masters.donntu.org/2017/fknt/plitka/diss/indexu.htm>.

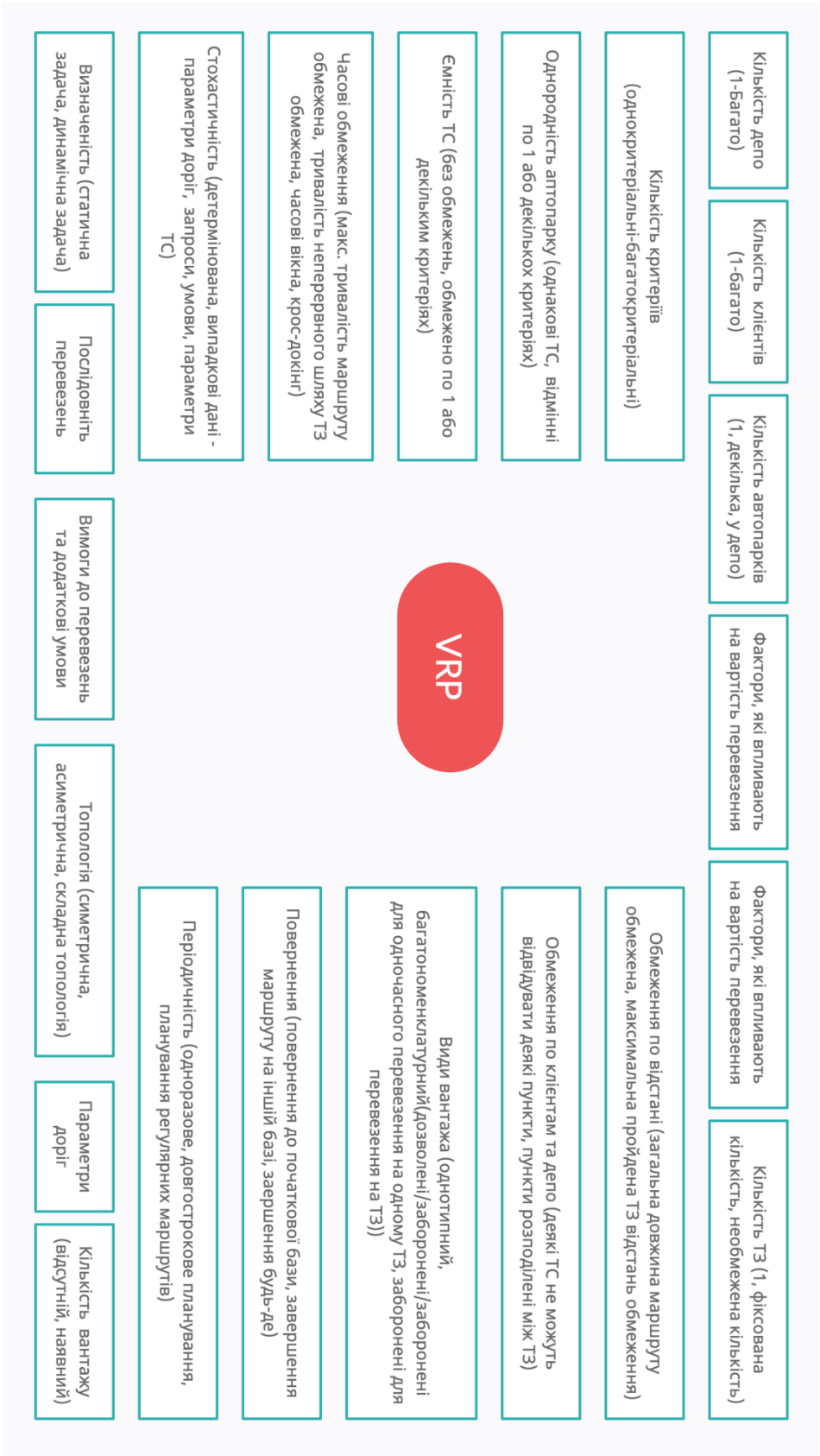
43. D. T. Pham, A. Ghanbarzadeh, E. Коç, S. Otri, S. Rahim, M. Zaidi. The Bees Algorithm // A Novel Tool for Complex Optimisation Problems. – pp. 1–6.
44. С. Н. Эйрих. Обзор методов решения задач маршрутизации транспорта // Труды Международного семинара по интеллектуальному планированию. – 2012. – no. 1. – pp. 22–30.
45. M. L. Fisher. Optimal solution of Vehicle Routing Problems using minimum K-trees // Operations Research. – 1994. – vol. 42. – pp. 626–642.
46. G. A. P. Kinderwater, M. W. Savelsbergh. Vehicle routing: handling edge exchanges // In E.H.L. Aarts and J. K. Lenstra (eds), Local Search in Combinatorial Optimization. –1997.
47. M. L. Fisher, R. Jaikumar. A generalized assignment heuristic for vehicle routing // Networks. – 1981. – vol. 11. – pp. 109–124.
48. Variable neighborhood search: basics and variants / Pierre Hansen, Nenad Mladenovic, Raca Todosijević, Saïd Hanafi. // EURO Journal on Computational Optimization 5(3). – 2016.
49. D. Taillard. Parallel iterative search methods for vehicle routing problems // Networks. – 1993. – vol. 23. – pp. 661–673.
50. J. Kelly, J. P. Xu. A network flow-based tabu search heuristic for the Vehicle Routing Problem // Transportation Science. – 1996. – vol. 30. – pp. 379–393.
51. M. Dorigo, S. Member, L. M. Gambardella. Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. – 1997. – vol. 1. – no. 1. – pp. 53–66.
52. M. Dorigo, L. M. Gambardella. Ant colonies for the travelling salesman problem // Bio Systems. – 1997, Jan. – vol. 43. – no. 2. – pp. 73–81.
53. T. Stützle, M. Dorigo. ACO Algorithms for the Traveling Salesman Problem // Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications. – 1990 . – p.
54. E. D. Taillard, L. M. Gambardella. Adaptive Memories for the Quadratic Assignment Problem // Istituto Dalle Molle Di Studi Sul. – 1997. – pp. 1–18.

55. E. D. Taillard. FAST: Fast ant system // Technical Report. – 1998.
56. M. Dorigo, S. Member, L. M. Gambardella. Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. – 1997. – vol. 1. – no. 1. – pp. 53–66.
57. F. Glover, C. McMillan. The general employee scheduling problem: an integration of MS and AI // Computer and Operations Research. – 1986.
58. B. Yu, Z. Z. Yang, B. Z. Yao. A hybrid algorithm for vehicle routing problem with time windows // Expert Systems with Applications. – 2011, Jan. – vol. 38. –no. 1. – pp. 435–441.
59. IBM [Электроний ресурс]: Sterling Transportation Management System. – 2013. – Режим доступа <http://www.01.ibm.com/software/commerce/products/transportation-management/>.
60. Центр ГИС Аналитик [Электроний ресурс]: ArcLogistics 9.3. – 2013. – Режим доступа <http://www.giscenter.net/news/272-arclogistics-93->.
61. IS intelstride [Электроний ресурс]: ANTOR LogisticsMaster. – Режим доступа <http://intelstride.by/ru/route-planning.html>.
62. Brian Kallehauge. VEHICLE ROUTING PROBLEM WITH TIME WINDOWS / Brian Kallehauge, Jesper Larsen., 2007.
63. М. Чураков, А. Якушев. Муравьиные алгоритмы. – 2006. – pp. 1–15.
64. O. Braysy, M. Gendreau. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics // Transportation Science. – 2005, Feb. – vol. 39. – no. 1. – pp. 119–139.
65. B. Yu, Z. Z. Yang, B. Z. Yao. A hybrid algorithm for vehicle routing problem with time windows // Expert Systems with Applications. – 2011. – vol. 38. –no. 1. –pp. 435–441.
66. Marco Antonio Cruz-Chávez, Ocotlán Díaz-Parra, J. A. Hernández¹, José Crispín Zavala-Díaz, Martín G. Search Algorithm for the Constraint Satisfaction Problem of VRPTW // Fideicomiso SEP-UNAM. – 2006-2007.
67. Marco A. Cruz-Chávez¹, Ocotlán Díaz-Parra, David Juárez-Romero, Martín G. Martínez-Rangel. Memetic Algorithm Based on a Constraint Satisfaction.

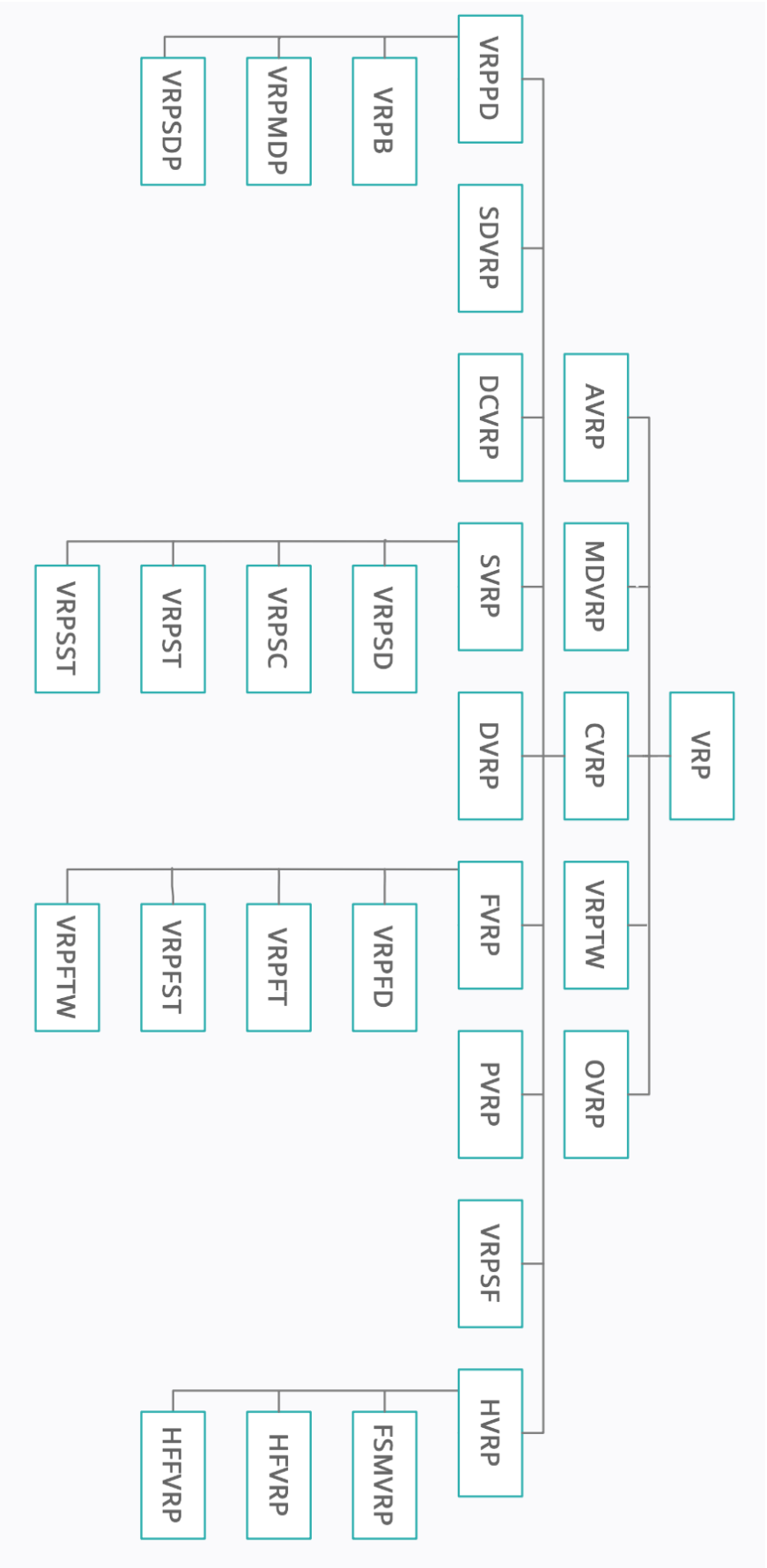
ДОДАТОК А

Графічні матеріали

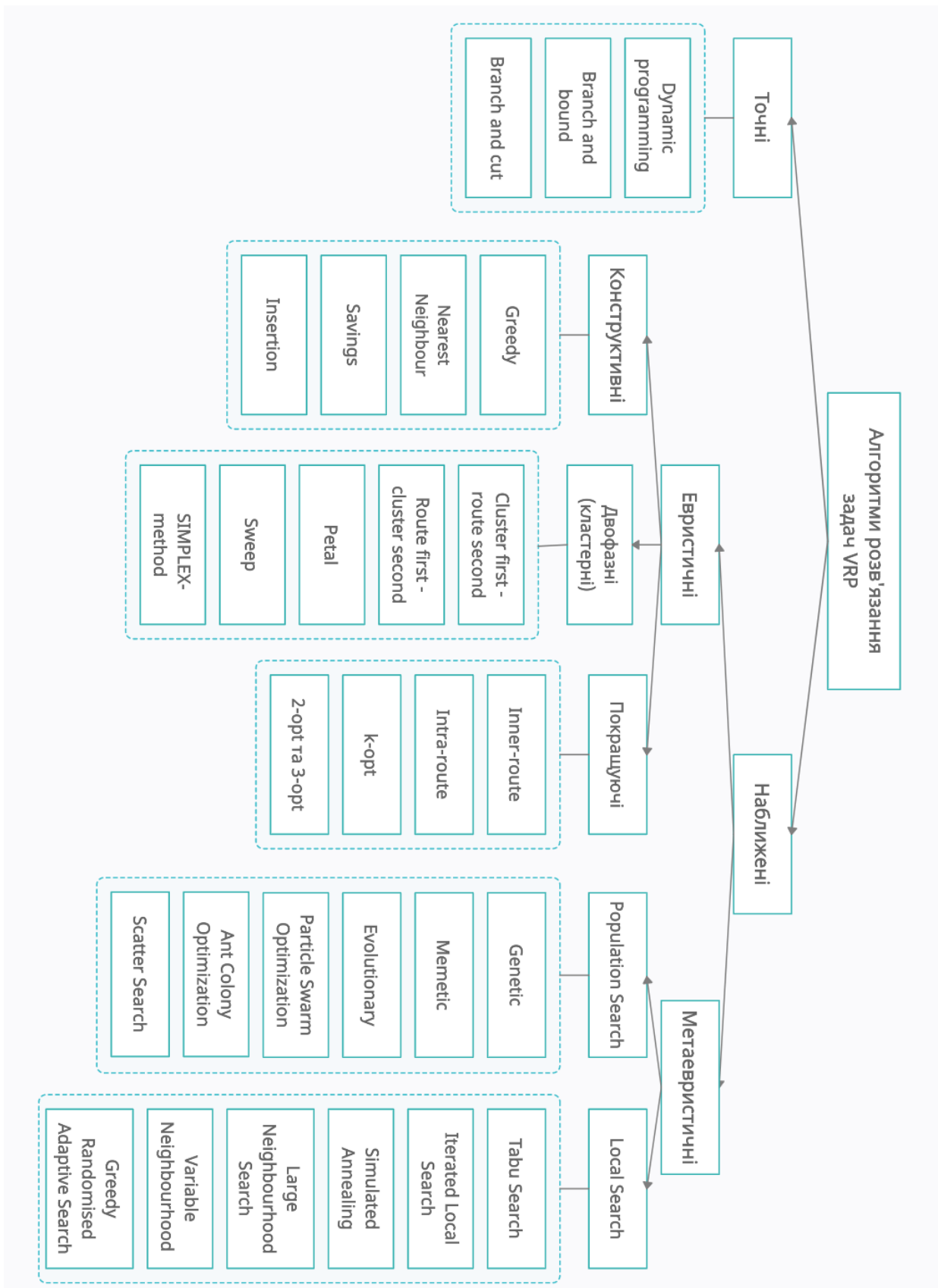
Класифікація характеристик VRP



Ієрархічна схема узагальнень і розширень VRP



Класифікація найпоширеніших алгоритмів розв'язання VRP



Перевірка роботи на плагіат



Ім'я користувача:
Попенко Володимир Дмитрович

Дата перевірки:
18.05.2021 21:48:07 EEST

Дата звіту:
18.05.2021 21:50:57 EEST

ID перевірки:
1007917839

Тип перевірки:
Doc vs Internet + Library

ID користувача:
77149

Назва документа: Baranova_magistr_is91mn_2

Кількість сторінок: 77 Кількість слів: 14227 Кількість символів: 105069 Розмір файлу: 2.67 MB ID файлу: 1008009934

8.68% Схожість

Найбільша схожість: 1.63% з джерелом з Бібліотеки (ID файлу: 1007994648)

6.52% Джерела з Інтернету 194 Сторінка 79

5.52% Джерела з Бібліотеки 211 Сторінка 83

0.1% Цитат

Цитати 2 Сторінка 84

Вилучення списку бібліографічних посилань вимкнене

1.7% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

1.7% Вилучення з Інтернету 1 Сторінка 85

Немає вилучених бібліотечних джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 18