



**OPTIMIZING DELIVERY EFFICIENCY THROUGH
VEHICLE MODIFICATIONS: A COMPARATIVE
ANALYSIS OF STANDARD AND SIDE-DOOR TRUCKS
IN CAPACITATED VEHICLE ROUTING WITH
TIME WINDOWS**

BY

BUREEVARAT VONGBUREE

**AN INDEPENDENT STUDY SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF ENGINEERING (LOGISTICS AND SUPPLY
CHAIN SYSTEMS ENGINEERING)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2023**

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

INDEPENDENT STUDY

BY

BUREEVARAT VONGBUREE


ENTITLED

OPTIMIZING DELIVERY EFFICIENCY THROUGH VEHICLE
MODIFICATIONS: A COMPARATIVE ANALYSIS OF STANDARD AND
SIDE-DOOR TRUCKS IN CAPACITATED VEHICLE ROUTING WITH
TIME WINDOWS

was approved as partial fulfillment of the requirements for
the degree of Master of Engineering (Logistics and Supply Chain Systems Engineering)

on May 29, 2024

Member and Advisor



(Associate Professor Sun Olapiriyakul, Ph.D.)

Member



(Associate Professor Suchada Rianmora, D.Eng.)

Director



(Professor Pruettha Nanakorn, D.Eng.)

| | |
|-------------------------|---|
| Independent Study Title | OPTIMIZING DELIVERY EFFICIENCY THROUGH VEHICLE MODIFICATIONS: A COMPARATIVE ANALYSIS OF STANDARD AND SIDE-DOOR TRUCKS IN CAPACITATED VEHICLE ROUTING WITH TIME WINDOWS |
| Author | Bureevarat Vongburee |
| Degree | Master of Engineering (Logistics and Supply Chain Systems Engineering) |
| Faculty/University | Sirindhorn International Institute of Technology/ Thammasat University |
| Advisor | Associate Professor Sun Olapiriyakul, Ph.D. |
| Academic Years | 2023 |

ABSTRACT

This research explores how vehicle modifications, specifically adding side doors to trucks, impact delivery efficiency in the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). By using a simulation-based approach, the study compares standard trucks with modified trucks across a case involving 15 customers, one depot, and a fleet of 5 trucks. The findings show that side doors significantly reduce service times per pallet, leading to shorter route times and improved operational efficiency. A cost-benefit analysis confirms the economic feasibility of these modifications by calculating a payback period. This research offers valuable insights into enhancing delivery operations through vehicle modifications.

Keywords: Capacitated Vehicle Routing Problem, Time Windows, Vehicle Modifications, Logistics Efficiency, Simulation, Cost-Benefit Analysis

ACKNOWLEDGEMENTS

I would like to express my gratitude to everyone who played a role in the successful completion of this study. The assistance, guidance, and inspiration provided by all have been invaluable throughout this research.

First and foremost, I extend my deepest appreciation to Assistant Professor Sun Olapiriyakul for his valuable suggestions and constructive feedback, which have greatly enhanced the efficiency of my study and research methodology. His expertise and dedication were crucial in directing and elevating the quality of this study.

I also owe a profound debt of gratitude to Assistant Professor Sun Olapiriyakul for his continuous guidance, valuable insights, and unwavering support. His proficiency and commitment were essential in shaping the direction and enhancing the quality of this study.

Furthermore, I would like to thank the many scholars, researchers, and authors whose contributions laid the groundwork and framework for my research. Their knowledge and advice played a significant role in shaping my understanding of the subject matter.

Lastly, I would like to express my deepest appreciation to my family for their constant support, patience, and understanding throughout my studies. Your love and support have been the motivation that kept me going through difficult times.

Bureevarat Vongburee

TABLE OF CONTENTS

| | Page |
|---|------|
| ABSTRACT | (1) |
| ACKNOWLEDGEMENTS | (2) |
| LIST OF TABLES | (5) |
| LIST OF FIGURES | (6) |
| LIST OF SYMBOLS/ABBREVIATIONS | (7) |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 The Vehicle Routing Problem | 1 |
| 1.2 The Capacitated Vehicle Routing Problem with Time Windows | 2 |
| 1.2.1 The Capacitated Vehicle Routing Problem | 2 |
| 1.2.2 The Vehicle Routing Problem with Time Windows | 3 |
| 1.3 Problem Statement | 4 |
| 1.4 Objective | 4 |
| CHAPTER 2 REVIEW OF LITERATURE | 5 |
| 2.1 Relevant Theory and General Knowledge | 5 |
| 2.1.1 The Vehicle Routing Problem | 5 |
| 2.1.2 The Capacitated Vehicle Routing Problem with Time Windows | 6 |
| 2.2 Applications and Case Studies | 7 |
| CHAPTER 3 METHODOLOGY | 8 |
| 3.1 Data Collection | 8 |
| 3.2 Data Analysis | 11 |
| 3.3 Optimization Model | 11 |

| | |
|--|-----|
| | (4) |
| 3.3.1 The Capacitated Vehicle Routing Problem with Time Windows Without Truck Modifications | 11 |
| 3.3.2 The Capacitated Vehicle Routing Problem with Time Windows With Modified Trucks | 14 |
| CHAPTER 4 RESULTS AND DISCUSSION | 16 |
| 4.1 The Result for Lotus Company | 16 |
| 4.1.1 The Capacitated Vehicle Routing Problem with Time Windows Without Truck Modifications | 16 |
| 4.1.2 The Capacitated Vehicle Routing Problem with Time Windows With Modified Trucks | 21 |
| 4.2 Calculating the Payback Period for Truck Modifications | 26 |
| 4.3 Practical Contributions of Vehicle Modifications in CVRPTW | 27 |
| CHAPTER 5 CONCLUSION | 28 |
| 5.1 Evaluating the Impact of Truck Modifications | 28 |
| 5.2 Assessing the Payback Period for Truck Modifications | 28 |
| REFERENCES | 30 |
| APPENDICES | 32 |
| APPENDIX A | 33 |
| APPENDIX B | 40 |
| BIOGRAPHY | 47 |

LIST OF TABLES

| Tables | Page |
|--|------|
| 3.1 Lists the 15 branches of Lotus Company | 9 |
| 3.2 Demand data for each customer | 10 |
| 3.3 Time windows data for each customer | 10 |
| 3.4 Travel time data | 10 |
| 3.5 Travel distance data | 11 |
| 4.1 Route for truck 1 of CVRPTW without modified trucks | 16 |
| 4.2 Route for truck 2 of CVRPTW without modified trucks | 17 |
| 4.3 Route for truck 3 of CVRPTW without modified trucks | 17 |
| 4.4 Route for truck 4 of CVRPTW without modified trucks | 18 |
| 4.5 Route for truck 5 of CVRPTW without modified trucks | 18 |
| 4.6 Total load and time of all routes for CVRPTW without modified trucks | 18 |
| 4.7 Route for truck 1 of CVRPTW with modified trucks | 21 |
| 4.8 Route for truck 2 of CVRPTW with modified trucks | 22 |
| 4.9 Route for truck 3 of CVRPTW with modified trucks | 22 |
| 4.10 Route for truck 4 of CVRPTW with modified trucks | 23 |
| 4.11 Route for truck 5 of CVRPTW with modified trucks | 23 |
| 4.12 Total load and time of all routes for CVRPTW with modified trucks | 23 |

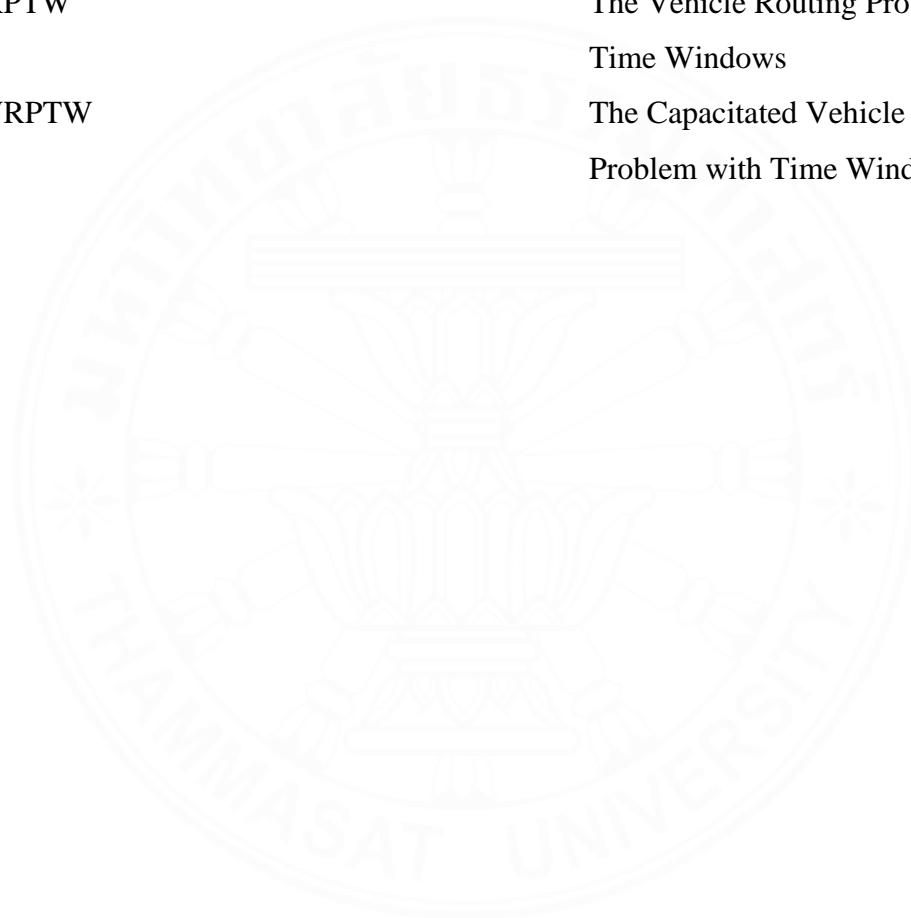
LIST OF FIGURES

| Figures | Page |
|---|------|
| 1.1 Illustration of the Vehicle Routing Problem | 2 |
| 1.2 Illustration of the Capacitated Vehicle Routing Problem with Time Windows | 3 |
| 3.1 Map of Lotus Company's Delivery Points | 9 |
| 4.1 Map of all routes for CVRPTW without modified trucks | 20 |
| 4.2 Map of all routes for CVRPTW with modified trucks | 25 |



LIST OF SYMBOLS/ABBREVIATIONS

| Symbols/Abbreviations | Terms |
|------------------------------|---|
| VRP | The Vehicle Routing Problem |
| CVRP | The Capacitated Vehicle Routing Problem |
| VRPTW | The Vehicle Routing Problem with Time Windows |
| CVRPTW | The Capacitated Vehicle Routing Problem with Time Windows |



CHAPTER 1

INTRODUCTION

Improving logistics efficiency is essential for business success in today's rapidly changing market, where customer expectations are constantly evolving. Customer satisfaction, influenced by timely deliveries, quality service, and the ability to meet specific needs, is crucial. Addressing these elements not only enhances customer loyalty but also strengthens the business's competitive position. Success in this area requires expertise in managing the Vehicle Routing Problem (VRP) and its variations, such as the Capacitated VRP (CVRP) and VRP with Time Windows (VRPTW), critical for optimizing operations and fulfilling customer demands. Effective implementation of these strategies ensures prompt deliveries and customized services, leading to lasting customer satisfaction and loyalty.

1.1 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a critical concern in operational research and logistics management, aimed at finding the most optimal way to deliver goods or services to a diverse group of customers with specific needs. This challenge involves analyzing and selecting the most efficient routes to meet delivery tasks while considering various constraints such as limited vehicle capacity, specified delivery deadlines, and individual customer preferences. Developing solutions for the VRP is key to enabling businesses to deliver products efficiently, enhance customer satisfaction, and reduce operational costs.

The VRP has evolved to encompass a wider range of real-world scenarios, such as the Capacitated Vehicle Routing Problem (CVRP), which deals with vehicle load limitations, and the Vehicle Routing Problem with Time Windows (VRPTW), adding the complexity of scheduled delivery times. Through the development and application of effective VRP solutions, businesses can expect to improve the quality of their delivery services, reduce operational times, and increase customer satisfaction levels. Additionally, this can significantly reduce logistics costs. Therefore, the VRP and its variations are not only of theoretical interest but also of great practical importance in

applying real-world business strategies to enhance competitive capabilities and efficiency of logistics systems across various industries.

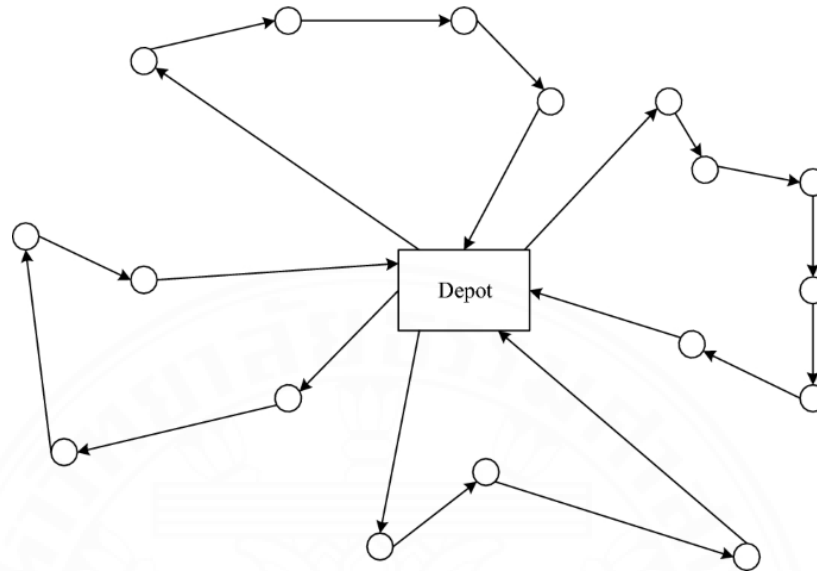


Figure 1.1 Illustration of the Vehicle Routing Problem.

Figure 1 illustrates a graphical diagram of transportation routes connecting a central warehouse (Depot) to various destinations (Customers), which is a representation of the Vehicle Routing Problem (VRP). This diagram is utilized to simulate and optimize the distribution of goods from the central warehouse to customers along various routes.

1.2 The Capacitated Vehicle Routing Problem with Time Windows

1.2.1 The Capacitated Vehicle Routing Problem

The Capacitated Vehicle Routing Problem (CVRP) poses a significant challenge in logistics and supply chain management, aiming to find the most efficient delivery routes for vehicles under specific capacity constraints. Originating from the Vehicle Routing Problem (VRP), the CVRP introduces an additional challenge by considering the vehicle capacity to ensure that the quantity of goods transported does not exceed the vehicle's maximum load capacity.

The principle of the CVRP is to minimize the total distance traveled or the overall cost while still servicing every customer without violating the capacity constraints of each vehicle. This problem requires not only determining the most efficient set of routes for a fleet of vehicles starting and ending at a central depot but

also allocating groups of customers to each vehicle in the most suitable manner under capacity constraints.

1.2.2 The Vehicle Routing Problem with Time Windows

In the landscape of logistics and transportation, a critical dimension emerges with the introduction of 'Time Windows' in Vehicle Routing Problems (VRP). This concept brings the element of time into the routing equation, defining specific time intervals during which each customer or location must be serviced. This temporal dimension introduces a layer of complexity to the optimization process, reflecting the dynamic nature inherent in many logistical operations.

The integration of Time Windows into VRP recognizes the practical reality that not all deliveries or service calls can occur at any given moment. Businesses, customers, and service providers impose constraints on when operations can occur, influenced by factors such as business hours, traffic patterns, and delivery windows. Effectively managing these time constraints becomes pivotal, not only for achieving cost efficiency but also for meeting customer expectations and operational requirements. As the demand for more responsive and time-conscious logistics solutions grows, the role of Time Windows in VRP becomes increasingly significant for businesses navigating the intricate interplay of spatial and temporal considerations.

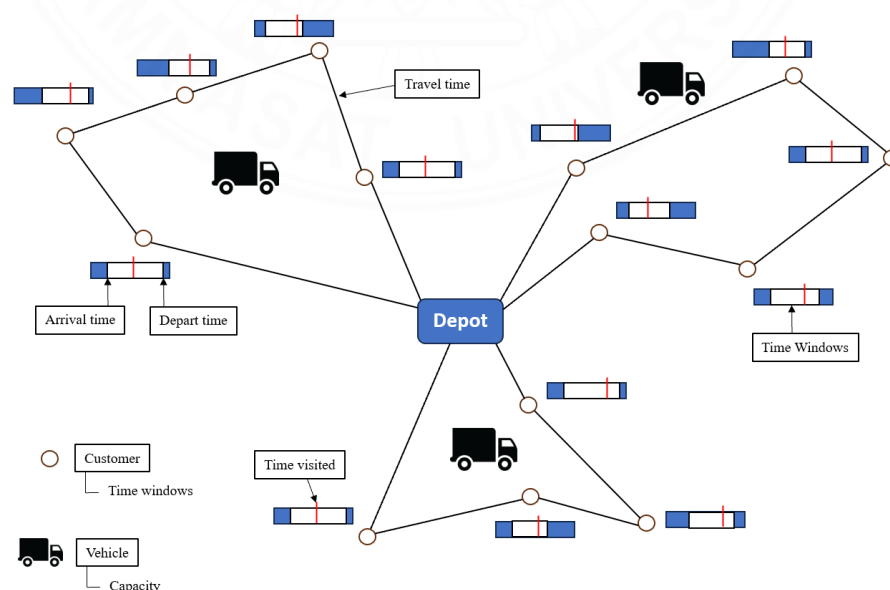


Figure 1.2 Illustration of the Capacitated Vehicle Routing Problem with Time Windows.

Figure 2 illustrates a diagram for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), showing the distribution routes from a central warehouse (Depot) to various customers. The diagram includes specified travel times between the depot and each customer, the arrival time, and departure time at each location. Multiple vehicles are displayed, each with a defined capacity, and the routes each vehicle will take according to the CVRPTW model.

1.3 Problem Statement

This study focuses on optimizing logistics operations by addressing the challenge of the Capacitated Vehicle Routing Problem (CVRP) integrated with customer preferences and time windows. The core issue is to develop a routing solution that minimizes travel costs and adheres to vehicle capacity constraints while also accommodating specific customer delivery preferences and time constraints. The aim is to enhance both operational efficiency and customer satisfaction in logistics management.

1.4 Objective

The objective is to create an optimization model that minimizes operational costs, adheres to vehicle capacity limits, respects delivery time windows, and minimizes the impact of side-door modifications on delivery trucks. This aims to assess how these modifications enhance delivery efficiency.

CHAPTER 2

REVIEW OF LITERATURE

In this literature review, we survey and analyze research related to the Vehicle Routing Problem (VRP), specifically the Capacitated Vehicle Routing Problem with Time Window (CVRPTW) and its extension to vehicle modification. We aim to understand the methods and technologies used to address these challenges, and analyze obstacles and opportunities for further research. This review focuses on identifying gaps in current research, and propose new directions that may lead to improvements in delivery services. Respond precisely to customer needs.

2.1 Relevant Theory and General Knowledge

2.1.1 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a primary research topic within the field of operations research and logistics management, focusing on optimizing vehicle routes from a central hub to various destinations. The research journey of VRP began with the foundational work of Dantzig and Ramser (1959), presenting "The truck dispatching problem." This study not only laid the groundwork for VRP research but also opened pathways for the development of new methods to solve this complex issue. Subsequent research by Clarke and Wright (1964) advanced the field by developing a heuristic method emphasizing the reduction of total delivery distances. This approach has become a cornerstone in VRP research, continuing to be used and refined in current studies. The book "The Vehicle Routing Problem" compiled by Toth and Vigo (2002) provided a comprehensive summary of VRP research and developments, introducing new techniques and methodologies for a better understanding and solution of VRP. This work is one of the most important references in the VRP literature. The application of genetic algorithms to solve the VRP by Baker and Ayechev (2003) exemplifies the use of complex computational techniques to find efficient VRP solutions, showcasing the potential of computational technology in tackling complex problems. Research by Zirour (2008) expanded the scope of VRP by exploring new models and solutions, including specific constraints such as timing and vehicle capacity. Further contribution from Golden et al., (2008) "The Vehicle Routing Problem: Latest Advances and New

Challenges," represents the most up-to-date summarization of VRP research, presenting the latest advancements and identifying new challenges faced by researchers in the VRP field. This work emphasizes the dynamic nature of VRP research, illustrating how changing business needs and technological developments continue to shape research directions and methodologies in VRP. Lastly, the survey by Kumar and Panneerselvam (2012) offered a comprehensive overview of VRP and its emerging variants, summarizing the developments and research in VRP and pointing towards future directions and challenges.

In conclusion, the literature on VRP demonstrates ongoing development in methods and applications, from foundational theoretical studies to advanced technological research. These studies not only enhance theoretical understanding and practice but also enable efficient real-world problem-solving in logistics and delivery systems.

2.1.2 The Capacitated Vehicle Routing Problem with Time Windows

The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) extends the classic Vehicle Routing Problem (VRP) by integrating vehicle capacity constraints and specific delivery time windows. This problem plays a vital role in enhancing logistics and distribution systems across various industries. Beginning with the work of El Hassani et al., (2008), a hybrid ant colony system approach was proposed for both CVRP and CVRPTW, showcasing its efficacy in improving delivery routes. This methodology balances the exploration of new paths with the exploitation of known efficient routes, adeptly managing the complexities of CVRPTW. Lin et al., (2010) introduced a new hybrid method combining Genetic Algorithm, Tabu Search, and Chaotic Search to tackle CVRPTW. This approach utilizes the strengths of each method to escape local optima and find efficient solutions, underscoring the importance of integrating various techniques to address CVRPTW's complexity. Arenas et al., (2017) applied the CVRPTW model to the collection of food donations, demonstrating the model's potential in managing logistics for charitable purposes. This study highlights the versatility of CVRPTW solutions in operational contexts ranging from commercial deliveries to humanitarian logistics. Romero-Gelvez et al., (2019) explored a CVRPTW model for cargo collection using a fleet with varied capacities, adding to the literature

by focusing on resource heterogeneity, a common scenario in logistics and delivery systems. Wahyuningsih et al., (2019) discussed various Vehicle Routing Problem with Time Windows models and their application in optimizing distribution. Their work examines the adaptation of CVRPTW models to meet specific operational requirements, indicating the ongoing evolution and modification of routing models to enhance distribution strategies. Natalia et al., (2021) tackled CVRP and CVRPTW using a bee algorithm approach to optimize the waste collection transportation issue. This study emphasizes the use of nature-inspired algorithms for environmental sustainability, showcasing the potential of CVRPTW models in supporting efficient and environmentally responsible logistics solutions.

2.2 Applications and Case Studies

This literature review explores several studies employing diverse techniques and programs to address complex issues in car routing. It presents various case studies:

Sooktip and Wattanapongsakorn (2015) focus on identifying desired solutions for the Capacitated Vehicle Routing Problem (CVRP), analyzing multiple objectives to find a suitable balance between cost and service quality, considering stakeholders' preferences. This aids in making logistics and transportation decisions more efficient.

Yogaswara, (2015) study examines the transportation routes and costs from a single depot in the JABABEKA Industrial Area, applying a specific analysis to the urban and industrial environment. It introduces new insights into adapting traditional logistic models to meet the specific constraints and demands of an urban area.

Bruniecki et al. (2016) present a real-world case study dealing with managing car routing problems for diverse fleets with additional time windows and constraints. They utilize programming models and analysis to tackle challenges in managing diverse fleets, offering useful methods for solving complex routing problems.

Tebaldi et al. (2020) adapt water wave algorithms to the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), applied in a real-world case study with propane data. This research demonstrates new approaches and potential improvements when leveraging technology and data analysis for route optimization.

CHAPTER 3

METHODOLOGY

3.1 Data Collection

In an era where the demand for personalized delivery services is increasing. Research focused on integrating customer needs into the high-capacity vehicle time windows (CVRPTW) has become an important part of logistics and supply chain management developments. This chapter aims to investigate and analyze the implementation and results of the proposed solution for CVRPTW based on a setting using real world data and modifications to vehicles to enhance performance. Subsequently, it aims to determine the payback period.

In this study, we utilize real-world data concerning a fleet of 5 trucks, with a single depot located in Klong Toey, and tasked with delivering goods to all 15 branches of the Lotus company. To analyze and plan the most efficient delivery routes, the collected data includes travel times from one location to another, each truck's capacity limited to 25 pallets, service times at each truck to 10 minutes, time windows at each branch and demand in each customer. Notably, the location of each site and the travel time data used for the analysis are derived from Google Maps, providing access to accurate and current information on distances and travel durations between points. All this information is meticulously analyzed to identify the most suitable delivery routes that not only meet customer needs but are also logistically feasible. The analysis focuses on developing delivery strategies that can enhance customer satisfaction while simultaneously optimizing operational efficiency.

From the above, lists the 15 branches of Lotus Company is shown in Table 3.1, locations in all 15 Lotus branches as shown in Figure 3.1, demand data for each customer is shown in Table 3.2, Time windows data for each customer is shown in Table 3.3, travel time data is shown in Table 3.4 and travel distance data is shown in Table 3.5.

Table 3.1 Lists the 15 branches of Lotus Company.

| Number | Branches |
|--------|------------------------|
| 0 | Depot Klong Toey |
| 1 | Banfah - Lam Luk Ka |
| 2 | Chaengwattana |
| 3 | Charan Sanitwong |
| 4 | Fortune Town |
| 5 | Klong 2 |
| 6 | Ladprao |
| 7 | Laksi |
| 8 | Pathum Thani |
| 9 | Pinklao |
| 10 | Prachacheun |
| 11 | Rama 1 |
| 12 | Rangsit |
| 13 | Rangsit – Nakhon Nayok |
| 14 | Rattathibet |
| 15 | Wanghin |

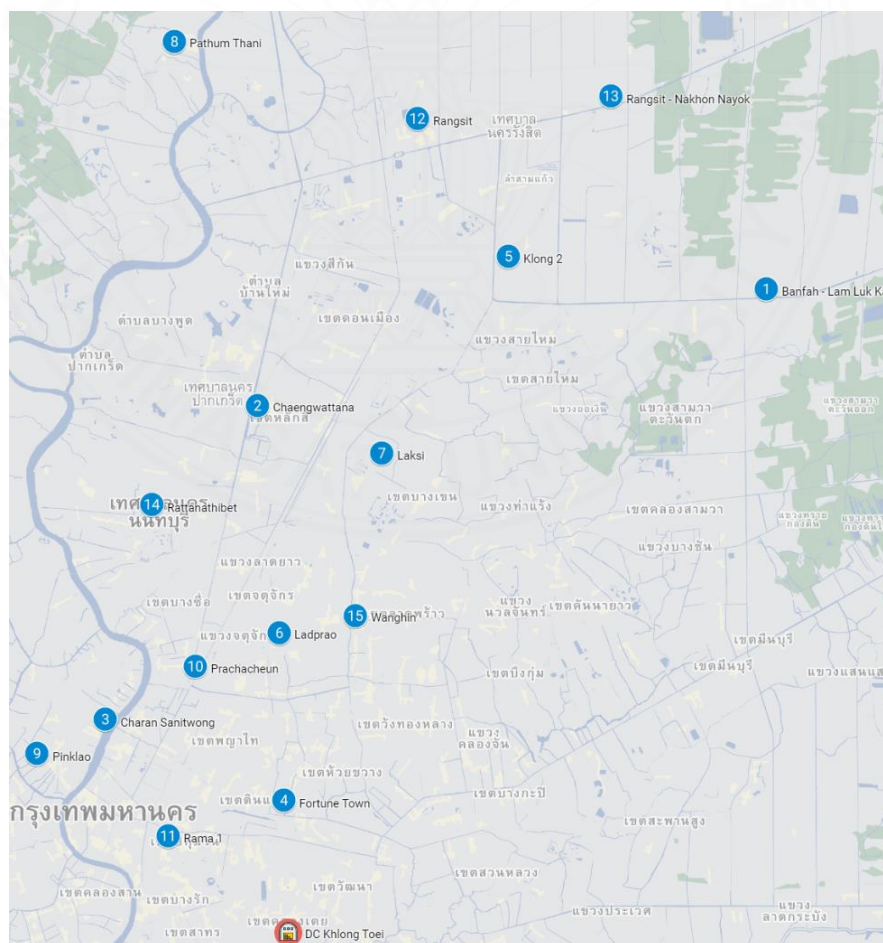
**Figure 3.1** Map of Lotus Company's Delivery Points.

Table 3.2 Demand data for each customer.

| | | Branches | | | | | | | | | | | | | | |
|---------------|--------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | Depot | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Demand | 0 | 12 | 10 | 8 | 5 | 6 | 7 | 10 | 7 | 9 | 10 | 5 | 4 | 12 | 3 | 8 |

Table 3.3 Time windows data for each customer.

| Location | Arrive (Minutes) | Depart (Minutes) |
|------------------------|-------------------------|-------------------------|
| Depot Klong Toey | 8:00 | 18:00 |
| Banfah - Lam Luk Ka | 9:00 | 13:00 |
| Chaengwattana | 9:00 | 14:00 |
| Charan Sanitwong | 12:00 | 16:00 |
| Fortune Town | 10:00 | 15:00 |
| Klong 2 | 11:00 | 16:00 |
| Ladprao | 12:00 | 17:00 |
| Laksi | 12:00 | 15:00 |
| Pathum Thani | 10:00 | 14:00 |
| Pinklao | 9:00 | 14:00 |
| Prachacheun | 10:00 | 14:00 |
| Rama 1 | 12:00 | 16:00 |
| Rangsit | 10:00 | 14:00 |
| Rangsit – Nakhon Nayok | 11:00 | 16:00 |
| Rattanathibet | 10:00 | 15:00 |
| Wanghin | 13:00 | 15:00 |

Table 3.4 Travel time data (minute).

| From\To | Depot | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------------|--------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Depot | 0 | 37 | 29 | 27 | 17 | 36 | 20 | 30 | 40 | 26 | 22 | 18 | 34 | 88 | 28 | 30 |
| 1 | 35 | 0 | 30 | 40 | 34 | 17 | 33 | 24 | 38 | 44 | 38 | 38 | 24 | 16 | 39 | 31 |
| 2 | 29 | 28 | 0 | 25 | 26 | 20 | 18 | 10 | 28 | 29 | 23 | 30 | 20 | 24 | 23 | 22 |
| 3 | 25 | 44 | 23 | 0 | 20 | 33 | 20 | 26 | 37 | 6 | 15 | 18 | 32 | 91 | 21 | 25 |
| 4 | 17 | 33 | 26 | 23 | 0 | 36 | 20 | 30 | 38 | 24 | 20 | 18 | 35 | 87 | 23 | 23 |
| 5 | 33 | 17 | 29 | 31 | 30 | 0 | 28 | 19 | 28 | 35 | 28 | 34 | 14 | 75 | 31 | 32 |
| 6 | 16 | 33 | 20 | 15 | 12 | 25 | 0 | 16 | 32 | 18 | 12 | 15 | 24 | 82 | 15 | 15 |
| 7 | 28 | 27 | 15 | 25 | 24 | 25 | 15 | 0 | 38 | 30 | 22 | 29 | 24 | 81 | 20 | 18 |
| 8 | 39 | 37 | 22 | 35 | 37 | 25 | 34 | 31 | 0 | 32 | 30 | 36 | 18 | 24 | 27 | 36 |
| 9 | 25 | 45 | 23 | 10 | 24 | 36 | 22 | 29 | 32 | 0 | 18 | 17 | 35 | 100 | 22 | 26 |
| 10 | 19 | 37 | 17 | 13 | 15 | 29 | 14 | 21 | 31 | 17 | 0 | 17 | 28 | 89 | 16 | 18 |
| 11 | 15 | 34 | 21 | 17 | 14 | 35 | 17 | 26 | 35 | 13 | 16 | 0 | 31 | 91 | 20 | 23 |
| 12 | 33 | 26 | 21 | 26 | 30 | 13 | 24 | 20 | 19 | 34 | 27 | 33 | 0 | 68 | 30 | 27 |
| 13 | 88 | 63 | 82 | 87 | 85 | 70 | 82 | 77 | 79 | 91 | 84 | 90 | 70 | 0 | 87 | 85 |
| 14 | 23 | 34 | 15 | 19 | 20 | 27 | 18 | 18 | 26 | 21 | 18 | 22 | 25 | 86 | 0 | 17 |
| 15 | 26 | 31 | 21 | 20 | 16 | 30 | 12 | 35 | 34 | 23 | 22 | 24 | 30 | 87 | 18 | 0 |

Table 3.5 Travel distance data (kilometer).

| From\To | Depot | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Depot | 0.0 | 35.7 | 25.9 | 16.7 | 6.7 | 35.7 | 15.2 | 23.8 | 45.7 | 15.8 | 16.0 | 8.0 | 36.1 | 43.8 | 23.6 | 20.3 |
| 1 | 39.8 | 0.0 | 26.4 | 37.4 | 34.8 | 14.6 | 28.4 | 20.5 | 35.6 | 41.0 | 33.3 | 38.1 | 23.1 | 18.0 | 30.0 | 30.0 |
| 2 | 26.8 | 27.3 | 0.0 | 18.5 | 20.3 | 16.4 | 12.9 | 6.8 | 23.4 | 22.1 | 8.0 | 9.3 | 16.8 | 24.4 | 14.5 | 14.5 |
| 3 | 15.8 | 40.5 | 18.8 | 0.0 | 11.0 | 30.4 | 13.3 | 19.9 | 36.3 | 4.2 | 8.0 | 9.3 | 30.7 | 38.4 | 13.2 | 16.4 |
| 4 | 7.2 | 33.7 | 19.3 | 12.2 | 0.0 | 25.2 | 8.5 | 15.8 | 41.1 | 13.0 | 10.3 | 7.5 | 29.5 | 16.9 | 16.9 | 10.0 |
| 5 | 33.9 | 12.3 | 18.0 | 29.0 | 27.4 | 0.0 | 20.0 | 12.2 | 21.9 | 32.7 | 24.9 | 29.8 | 9.5 | 11.9 | 21.6 | 21.6 |
| 6 | 14.0 | 31.5 | 12.9 | 11.8 | 8.4 | 22.7 | 0.0 | 10.8 | 33.8 | 14.8 | 6.3 | 9.9 | 23.1 | 30.8 | 10.6 | 7.3 |
| 7 | 21.8 | 22.5 | 5.9 | 16.9 | 15.3 | 11.4 | 8.0 | 0.0 | 26.8 | 20.6 | 12.8 | 17.7 | 14.7 | 25.9 | 9.5 | 9.5 |
| 8 | 41.5 | 32.7 | 18.0 | 31.3 | 36.7 | 21.0 | 28.6 | 24.3 | 0.0 | 35.0 | 27.2 | 35.1 | 14.0 | 21.3 | 22.8 | 32.0 |
| 9 | 14.1 | 42.3 | 20.6 | 5.0 | 12.6 | 32.2 | 15.1 | 21.7 | 38.1 | 0.0 | 9.7 | 7.3 | 32.5 | 40.2 | 15.0 | 18.2 |
| 10 | 15.0 | 34.5 | 12.8 | 7.1 | 10.2 | 24.3 | 6.1 | 13.9 | 30.3 | 9.6 | 0.0 | 8.3 | 24.7 | 32.4 | 8.7 | 10.3 |
| 11 | 7.3 | 37.8 | 20.9 | 9.0 | 6.2 | 27.7 | 9.7 | 18.3 | 38.4 | 7.5 | 8.8 | 0.0 | 30.6 | 38.3 | 16.2 | 14.8 |
| 12 | 36.2 | 20.8 | 20.5 | 31.5 | 31.2 | 9.1 | 22.6 | 14.7 | 14.3 | 35.2 | 27.4 | 32.3 | 0.0 | 9.6 | 27.1 | 24.1 |
| 13 | 42.9 | 15.2 | 27.3 | 36.9 | 38.0 | 11.1 | 29.3 | 21.4 | 22.6 | 40.5 | 32.8 | 39.0 | 10.0 | 0.0 | 33.8 | 30.9 |
| 14 | 22.6 | 33.8 | 8.6 | 11.0 | 17.8 | 22.9 | 9.8 | 13.2 | 22.5 | 14.6 | 8.4 | 16.2 | 23.2 | 30.9 | 0.0 | 11.5 |
| 15 | 15.2 | 24.9 | 14.1 | 14.4 | 9.8 | 20.0 | 5.5 | 10.6 | 35.0 | 18.1 | 10.3 | 15.2 | 23.3 | 34.1 | 11.7 | 0.0 |

3.2 Data Analysis

The analysis and extraction of results in this study are conducted using Python on the Google Colab platform, specifically utilizing the OR Tools library for data computation and analysis. The use of OR Tools not only facilitates efficient handling of complex calculations but also ensures a smooth and effective data analysis process. Employing Google Colab alongside the OR Tools library combines powerful tools for data analysis, enabling the resolution of problems and the identification of optimal delivery routes with high efficiency. This integration of tools significantly streamlines the analytical process, allowing for an effective approach to solving the study's challenges and achieving the best possible outcomes.

3.3 Optimization Model

3.3.1 The Capacitated Vehicle Routing Problem with Time Windows Without Truck Modifications

The analysis and design of solutions for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) necessitates the use of a detailed and comprehensive mathematical model. This model is utilized for simulating complex delivery scenarios to understand and address the intricate delivery challenges. The

forthcoming mathematical model is tailored to refine the most efficient delivery routes, considering the conditions of travel. Additionally, it considers the imperative to maintain the capability to serve customers within predetermined time frames without exceeding the available vehicle capacity limits.

Finally, the mathematical formulation for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) is the following:

$$\text{Min} \quad \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ijk} \quad (3.1)$$

$$\text{Subject to} \quad \sum_{k \in V} \sum_{j \in N} X_{ijk} = 1, \quad \forall i \in C, \quad (3.2)$$

$$\sum_{i \in C} \sum_{j \in N} q_i X_{ijk} \leq c, \quad \forall k \in V, \quad (3.3)$$

$$\sum_{i \in N} X_{0jk} = 1, \quad \forall k \in v, \quad (3.4)$$

$$\sum_{i \in N} X_{ijk} - \sum_{i \in N} X_{jik} = 0, \quad \forall j \in C, \forall k \in v, \quad (3.5)$$

$$\sum_{i \in N} X_{ijk} X_{i,n+1,k} = 1, \quad \forall k \in v, \quad (3.6)$$

$$X_{ijk} (S_{ik} + T_{ij} - S_{jk}) \leq 0, \quad \forall i, j \in N, \forall k \in v, \quad (3.7)$$

$$e_i \leq S_{ik} \leq l_i, \quad \forall i \in N, \forall k \in v, \quad (3.8)$$

$$S_{ik} + (t_k q_i) \leq l_i, \quad \forall i \in N, \forall k \in v, \quad (3.9)$$

$$S_{ik} \in R^+, \quad \forall i \in N, \forall k \in v, \quad (3.10)$$

$$X_{ijk} \in \{0,1\}, \quad \forall i \in N, \forall k \in v, \quad (3.11)$$

Indices:

i and j : Indexes used to represent customers or points that trucks must visit. Generally, i represents the starting point and j represents the destination in a trip between customers.

k : Index used to represent a truck within the fleet of available trucks.

V : The set containing all trucks in the problem, with k being a member of the set V .

N : The set containing all points in the model, including the central depot and all customer points.

C : The set containing all customer points that trucks must visit, excluding the central depot.

Variables:

X_{ijk} : A binary variable that indicates whether truck k travels from customer i to customer j (1 if there is a trip, 0 otherwise).

S_{ik} : The arrival time at customer i by truck k .

Parameters:

d_{ij} : The travel time between customer i and j .

q_i : The demand of customer i

c : The maximum capacity of the truck.

e_i and l_i : The allowed time window for starting and ending the service at customer i .

T_{ij} : The travel time from customer i and j .

t_k : The service of truck k .

The objective function (1) seeks to minimize the total travel time across all routes and is essential for optimizing the operational costs of truck routing. Constraint (2) ensures that each customer is visited exactly once. This is crucial in truck routing to guarantee that all customer demands are met without redundancy. Constraint (3) mandates that the total load carried by a truck does not exceed its maximum capacity. These respects the physical limitations of the truck and ensures that deliveries remain feasible. Constraint (4) ensures that every truck departs from the depot, signifying the start of each route. Constraint (5), known as the flow conservation or balance constraint, ensures that for every customer visited by a truck, there is an exit to another customer, hence maintaining a consistent flow of deliveries without leaving any truck stranded. Constraint (6) dictates that all trucks must return to the depot after completing their deliveries, concluding each route. Constraint (7) controls the timing of departures from customers, ensuring that the subsequent visit does not occur before the current service and travel times are completed, hence maintaining a logical sequence of service. Constraints (8) enforce the arrival start times at each customer to be within their predefined time windows, ensuring that all deliveries are made within acceptable and agreed-upon times. Constraint (9) ensures that the arrival and start of service at each customer location, when combined with the service duration, does not exceed the limits of the predefined time windows. Constraint (10) defines the arrival time as a non-

negative real number, reflecting the non-negative nature of time. Constraint (11) characterizes the decision variables X_{ijk} as binary, indicating whether a truck k travels from customer i to customer j , encapsulating the routing decisions within the model.

3.3.2 The Capacitated Vehicle Routing Problem with Time Windows with Modified Trucks

This case study aims to enhance and improve the efficiency of delivery operations by proposing modifications to the trucks used in transportation. Originally, the trucks only had rear doors, but to speed up the loading process and reduce service times, side doors were added to the trucks. This modification has reduced service time at each location from 10 minutes to just 5 minutes.

Under this modification, the fleet consists of 2 modified trucks with the additional side doors and 3 standard trucks, making up a total of 5 trucks. The operation still utilizes the same underlying data, including travel times between locations, customer demands, and the specified delivery time windows. The physical modification of the trucks has led to changes in the model's equations to reflect the reduced service times.

As a result, the equations need to be adjusted to reflect the shorter service times and improved management of the delivery trucks. These equations represent the new requirements and help in analyzing and planning more efficient delivery routes in a rational manner. The study aims to develop delivery strategies that not only meet customer needs but also enhance operational efficiency.

Below is the mathematical formulation for the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) which has been adapted for trucks that have been modified. The formulation will include an additional equation, which is Constraint 12, and there is a modification where Constraint 7 has been changed to Constraint 13.

$$t_{ij}^{type} = t_{service}^{type}, \quad \forall i \in N, \forall k \in v, \quad (3.12)$$

$$X_{ijk}(S_{ik} + t_{ij}^{type} - S_{jk}) \leq 0, \quad \forall i, j \in N, \forall k \in v, \quad (3.13)$$

Parameters:

t_{ij}^{type} : the service time at customer i , which may vary depending on the type of truck in use. In this case, the service time could be dependent on the movement between points i and j and the type of truck k making the trip. For example, if a truck type with side doors requires less time for loading or providing services at a customer location, this time would be used to define the route.

$t_{service}^{type}$: the standard service time for a truck of type k . This is a fixed time established for each type of truck, and it does not change regardless of the truck 's travel from or to any point in the network.

Constraint (12) defines the service time at customer i (t_{ij}^{type}) as being equal to a predefined standard service time ($t_{service}^{type}$) for truck type k . This equation ensures that the service time for a particular truck type is consistent across all customer points. For instance, if modified trucks with side doors are known to have a faster service time, this would be reflected as a shorter ($t_{service}^{type}$).

Constraint (13) provides a time-window constraint that incorporates the variable service times. It ensures that the service start time at customer i plus the service time for the truck type minus the service start time at customer j is non-positive, indicating that a truck cannot begin service at the next customer before completing service at the current one. This reflects the fact that service at each customer must be completed within the designated time window, adjusted for the specific service times of the truck types.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 The result for lotus company

This research focuses on experimenting to measure the impact of innovation in the transportation industry, specifically minor modifications to trucks used in the delivery of goods. The innovation in question involves modifying trucks to allow for faster and more efficient service delivery to destinations. By dividing the experiment into 2 case study is without truck modifications and another with modifications this study can cover the impacts on service time and total travel time in the delivery process.

4.1.1 The Capacitated Vehicle Routing Problem with Time Windows Without Modified Trucks

In this case, an analysis of the delivery outcomes was conducted using trucks without any modifications, based on a case study of the Lotus company and conducted on the Google Colab platform. The results obtained are as follows:

Table 4.1 Route for truck 1 of CVRPTW without modified trucks.

| Route for truck 1 | | | | |
|----------------------------------|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 08:23 | 0 | 0 |
| 1 | 09:00 | 11:00 | 12 | 120 |
| 13 | 11:16 | 13:16 | 24 | 120 |
| Depot | 14:44 | - | | |
| Load of the route: 24 pallets | | Time of the route: 381 minutes | | Distance of the route: 96.6 kilometers |

Table 4.2 Route for truck 2 of CVRPTW without modified trucks.

| Route for truck 2 | | | | |
|----------------------------------|--------------|-----------------------------------|-----------------------|--|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 09:20 | 0 | 0 |
| 8 | 10:00 | 11:10 | 7 | 70 |
| 12 | 11:28 | 12:08 | 11 | 40 |
| 5 | 12:21 | 13:21 | 17 | 60 |
| 15 | 13:53 | 15:13 | 25 | 80 |
| Depot | 15:39 | - | | |
| Load of the route: 25 pallets | | Time of the route: 379 minutes | | Distance of the route: 105.6 kilometers |

Table 4.3 Route for truck 3 of CVRPTW without modified trucks.

| Route for truck 3 | | | | |
|----------------------------------|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 09:32 | 0 | 0 |
| 14 | 10:00 | 10:30 | 3 | 30 |
| 2 | 10:45 | 12:25 | 13 | 100 |
| 7 | 12:35 | 14:15 | 23 | 100 |
| Depot | 14:43 | - | | |
| Load of the route: 23 pallets | | Time of the route: 311 minutes | | Distance of the route: 60.9 kilometers |

Table 4.4 Route for truck 4 of CVRPTW without modified trucks.

| Route for truck 4 | | | | |
|----------------------------------|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 11:40 | 0 | 0 |
| 6 | 12:00 | 13:10 | 7 | 70 |
| 10 | 13:22 | 15:02 | 17 | 100 |
| 3 | 15:15 | 16:35 | 25 | 80 |
| Depot | 17:00 | - | | |
| Load of the route: 25 pallets | | Time of the route: 320 minutes | | Distance of the route: 44.3 kilometers |

Table 4.5 Route for truck 5 of CVRPTW without modified trucks.

| Route for truck 5 | | | | |
|----------------------------------|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 11:42 | 0 | 0 |
| 11 | 12:00 | 12:50 | 5 | 50 |
| 9 | 13:03 | 14:33 | 14 | 90 |
| 4 | 14:57 | 15:47 | 19 | 50 |
| Depot | 16:04 | | | |
| Load of the route: 19 pallets | | Time of the route: 320 minutes | | Distance of the route: 35.3 kilometers |

Table 4.6 Total load and time of all routes for CVRPTW without modified trucks.

| | |
|------------------------------|------------------|
| Total load of all routes | 116 pallets |
| Total time of all routes | 1653 minutes |
| Total distance of all routes | 342.6 kilometers |

In this case, the result of the delivery presents route planning for truck 1 through 5 to deliver goods to various locations. The travel time and service time are calculated differently depending on the location and the quantity of goods to be delivered, including the time spent returning to the base after completing the deliveries.

For truck 1 shown in table 4.1, delivery starts at the depot at 08:23. Then, it proceeds to Banfah - Lam Luk Ka at 09:00, servicing 12 pallets for 2 hours, and then to Rangsit – Nakhon Nayok at 11:16, servicing 12 pallets for 2 hours. The vehicle returns to the depot at 14:44, with a total route time of 381 minutes, delivering a total of 24 pallets.

For truck 2 shown in table 4.2, delivery begins at the depot at 09:20. Then, it moves to Pathum Thani at 10:00, servicing 7 pallets for 70 minutes, and then to Rangsit at 11:28, servicing 4 pallets for 40 minutes. The route continues to Klong 2 at 12:21, servicing 6 pallets for 60 minutes, and then to Wanghin at 13:53, servicing 8 pallets for 80 minutes. The vehicle returns to the depot at 15:39, with a total route time of 379 minutes, delivering a total of 25 pallets.

For truck 3 shown in table 4.3, delivery starts at the depot at 09:32. Then, it moves to Rattana Thibet at 10:00, servicing 3 pallets for 30 minutes, and then to Chaengwattana at 10:45, servicing 10 pallets for 100 minutes. The route continues to Laksi at 12:35, servicing 10 pallets for 100 minutes. The vehicle returns to the depot at 14:43, with a total route time of 311 minutes, delivering a total of 23 pallets.

For truck 4 shown in table 4.4, delivery begins at the depot at 11:40. Then, it moves to Ladprao at 12:00, servicing 7 pallets for 70 minutes, and then to Prachacheun at 13:22, servicing 10 pallets for 100 minutes. The route continues to Charan Sanitwong at 15:15, servicing 8 pallets for 80 minutes. The vehicle returns to the depot at 17:00, with a total route time of 320 minutes, delivering a total of 25 pallets.

For truck 5 shown in table 4.5, delivery starts at the depot at 11:42. Then, it moves to Rama 1 at 12:00, servicing 5 pallets for 50 minutes, and then to Pinklao at 13:03, servicing 9 pallets for 90 minutes. The route continues to Fortune Town at 14:57, servicing 5 pallets for 50 minutes. The vehicle returns to the depot at 16:04, with a total route time of 262 minutes, delivering a total of 19 pallets.

In total, all trucks deliver a combined total of 116 units and spend a total of 1653 minutes on deliveries shown in table 4.6, indicating the challenges of efficient delivery while meeting capacity and time window requirements. And it can be represented in the form of a map as follows:

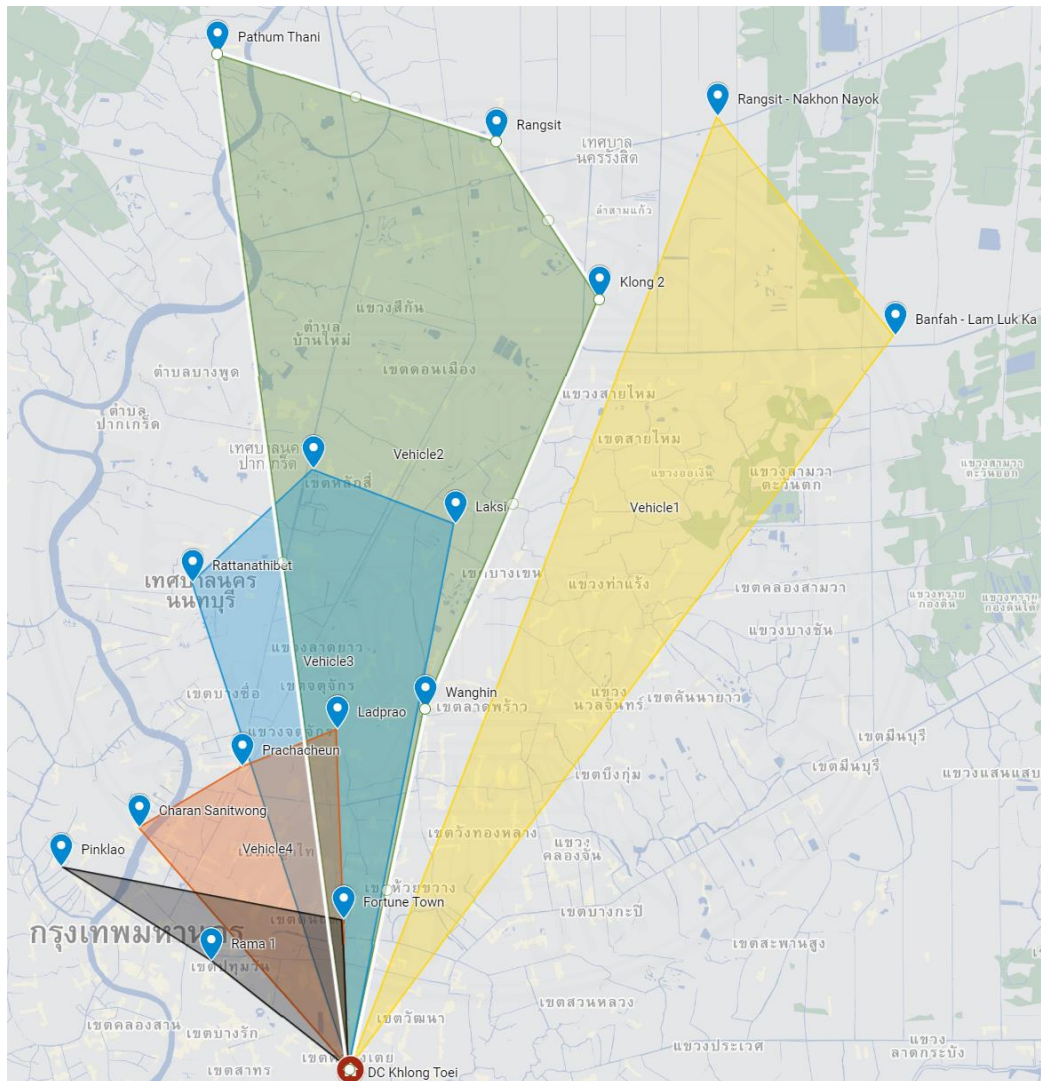


Figure 4.1 Map of all routes for CVRPTW without modified trucks.

Figure 4.1 illustrates the results of the delivery routes for all five trucks, as detailed in the comprehensive results tables. Each truck's route is color-coded for clear distinction: Vehicle 1 is represented by yellow, Vehicle 2 by green, Vehicle 3 by blue, Vehicle 4 by orange, and Vehicle 5 by black.

4.1.2 The Capacitated Vehicle Routing Problem with Time Windows With Modified Trucks

To enhance the efficiency of the delivery process and reduce the service time at each delivery location, the implementer modified two trucks by adding side doors. This modification reduced the service time per pallet from the original 10 minutes to just 5 minutes. The cost of modifying each truck was 33,000 baht, making the total cost for the two modified trucks 66,000 baht.

The analysis and mathematical model modifications applied in writing Python code on the Google Colab platform led to the following results:

Table 4.7 Route for truck 1 of CVRPTW with modified trucks.

| Route for truck 1 with modified truck | | | | |
|---------------------------------------|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 10:07 | 0 | 0 |
| 1 | 10:44 | 11:44 | 12 | 60 |
| 13 | 12:00 | 13:00 | 24 | 60 |
| Depot | 14:28 | - | | |
| Load of the route: 24 pallets | | Time of the route: 261 minutes | | Distance of the route: 96.6 kilometers |

Table 4.8 Route for truck 2 of CVRPTW with modified trucks.

| Route for truck 2 with modified truck | | | | |
|---------------------------------------|--------------|-----------------------------------|-----------------------|--|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 09:43 | 0 | 0 |
| 4 | 10:00 | 10:25 | 5 | 25 |
| 14 | 10:48 | 11:03 | 8 | 15 |
| 8 | 11:29 | 12:04 | 15 | 35 |
| 12 | 12:22 | 12:42 | 19 | 20 |
| 5 | 12:55 | 13:25 | 25 | 30 |
| Depot | 13:58 | - | | |
| Load of the route: 25 pallets | | Time of the route: 255 minutes | | Distance of the route: 103.1 kilometers |

Table 4.9 Route for truck 3 of CVRPTW with modified trucks.

| Route for truck 3 without modified truck | | | | |
|--|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 11:21 | 0 | 0 |
| 2 | 11:50 | 13:30 | 10 | 100 |
| 7 | 13:40 | 15:20 | 20 | 100 |
| Depot | 15:48 | - | | |
| Load of the route: 20 pallets | | Time of the route: 267 minutes | | Distance of the route: 54.6 kilometers |

Table 4.10 Route for truck 4 of CVRPTW with modified trucks.

| Route for truck 4 without modified truck | | | | |
|--|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 12:20 | 0 | 0 |
| 10 | 12:42 | 14:22 | 10 | 100 |
| 15 | 14:40 | 16:00 | 18 | 80 |
| 6 | 16:12 | 17:22 | 25 | 70 |
| Depot | 17:38 | - | | |
| Load of the route: 25 pallets | | Time of the route: 318 minutes | | Distance of the route: 45.9 kilometers |

Table 4.11 Route for truck 5 of CVRPTW with modified trucks.

| Route for truck 5 without modified truck | | | | |
|--|--------------|-----------------------------------|-----------------------|---|
| Location | Arrival time | Depart time | Capacity (Pallets) | Service time (Minutes) |
| Depot | - | 11:33 | 0 | 0 |
| 3 | 12:00 | 12:50 | 8 | 80 |
| 9 | 13:26 | 14:56 | 17 | 90 |
| 11 | 15:13 | 16:03 | 22 | 50 |
| Depot | 16:18 | | | |
| Load of the route: 22 pallets | | Time of the route: 285 minutes | | Distance of the route: 35.6 kilometers |

Table 4.12 Total load and time of all routes for CVRPTW with modified trucks.

| | |
|------------------------------|------------------|
| Total load of all routes | 116 pallets |
| Total time of all routes | 1386 minutes |
| Total distance of all routes | 335.7 kilometers |

In the second case, which involved two modified trucks, the delivery outcomes demonstrated significant improvements in service time and overall travel time. The modifications clearly reduced the service time at each delivery point, due to the addition of side doors which facilitated quicker loading and unloading.

For truck number 1 shown in table 4.7, which has been modified, the delivery starts at the depot at 10:07. Then it goes to Banfah - Lam Luk Ka at 10:44, servicing 12 pallets within 1 hour. After that, it proceeds to Rangsit – Nakhon Nayok at 12:00, again servicing 12 pallets for an hour. The truck returns to the depot at 14:28, completing the route in a total of 261 minutes and delivering a total of 24 pallets. This demonstrates a reduction in transportation time compared to the previous case due to the modification of the truck, which allows for faster servicing at each location.

For truck number 2 shown in table 4.8, which has also been modified, the delivery begins at the depot at 09:43. Then it moves to Fortune Town at 10:00, servicing 5 pallets in 25 minutes. Afterward, it heads to Rattanathibet at 10:48, servicing 3 pallets in 15 minutes. It then continues to Pathum Thani at 11:29, servicing 7 pallets in 35 minutes, and so on. After that, it proceeds to Rangsit at 12:22, servicing 4 pallets for 20 minutes. After that, it proceeds to Klong 2 at 12:55, servicing 6 pallets for 30 minutes. The truck returns to the depot at 13:58, completing the route in a total of 255 minutes and delivering a total of 25 pallets. It's evident that the route of the truck has changed due to the modification, allowing for more deliveries within the time windows and thus increasing the overall delivery capacity.

Truck number 3 shown in table 4.9, starting its delivery from the depot at 11:21, travels to Chaengwattana at 11:50, servicing 10 pallets in 100 minutes. It then proceeds to Laksi at 13:40, servicing another 10 pallets in 100 minutes. Finally, it returns to the depot at 15:48, completing the route in a total of 267 minutes and delivering a total of 20 pallets.

Truck number 4 shown in table 4.10, starting its delivery from the depot at 12:20, travels to Prachacheun at 12:42, servicing 10 pallets in 100 minutes. It then proceeds to Wanghin at 14:40, servicing 8 pallets in 80 minutes. Continuing to Ladprao at 16:12, it services 7 pallets in 70 minutes before returning to the depot at 17:38, completing the route in a total of 318 minutes and delivering a total of 25 pallets.

Truck number 5 shown in table 4.11, starting its delivery from the depot at 11:33, travels to Charan Sanitwong at 12:00, servicing 8 pallets in 80 minutes. It then proceeds to Pinklao at 13:26, servicing 9 pallets in 90 minutes. Continuing to Rama 1 at 15:13, it services 5 pallets in 50 minutes before returning to the depot at 16:18, completing the route in a total of 285 minutes and delivering a total of 22 pallets.

It is evident that trucks 3, 4, and 5, which have not been modified, also experienced changes in their delivery routes due to the modified trucks. This is because the modifications allow for faster servicing at each location, freeing up time for additional deliveries, which affected the routes of trucks 3, 4, and 5 as well.

These improvements resulted in a total operational time for all deliveries in the second case study of 1,386 minutes shown in table 4.12, a reduction from 1,653 minutes in the first scenario, demonstrating the effectiveness of the truck modifications in reducing service times and enhancing delivery efficiency. These modifications not only improved delivery capabilities and the management of service times but also facilitated better route planning and adjustments to deliveries in response to customer needs.

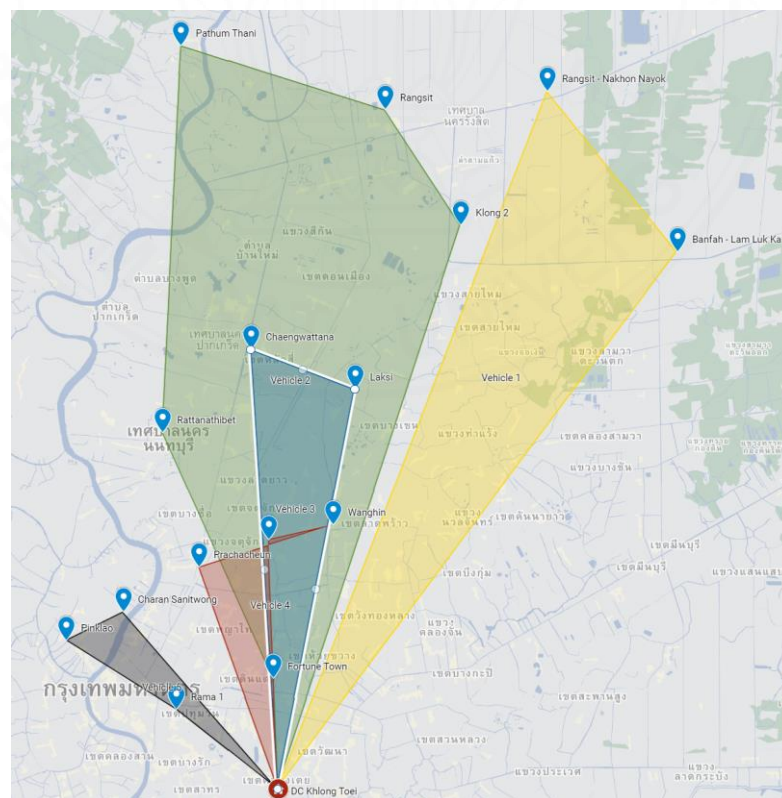


Figure 4.2 Map of all routes for CVRPTW with modified trucks.

Figure 4.2 illustrates the results of the transportation routes for all five trucks, two of which have been modified, as detailed in the comprehensive results tables. Each truck's route is color-coded for clear identification: Vehicle 1 is represented by yellow, Vehicle 2 by green, Vehicle 3 by blue, Vehicle 4 by orange, and Vehicle 5 by black. It is evident that the transportation routes of the trucks have changed, illustrating the impact of the modifications on their routing efficiency.

4.2 Calculating the Payback Period for truck Modifications

In this research, we analyzed the resource savings resulting from modifying trucks by adding side doors, which can reduce service time at each stop. We compared the modification between modified and standard trucks, focusing on transportation time for each route. Initial analysis revealed a total time saving of 267 minutes and a total distance saving of 6.9 kilometers per route due to the improved modifications.

To calculate related costs and payback period, we used assumptions including fuel costs at 4 baht per kilometer, driver wages at 100 baht per hour. From these parameters, we calculated the cost savings per route, including driver wages savings of 445 baht, fuel savings of 27.6 baht, totaling savings of 472.6 baht per route.

With a total modification cost of 66,000 baht for the trucks, approximately 139.7 routes are required to break even. This implies that at least 140 routes must be operated to recover the investment made in modifying the trucks. This analysis illustrates the potential for cost reduction and operational efficiency, which are critical considerations in managerial and financial decision-making for transportation businesses.

4.3 Practical Contributions of Vehicle Modifications in CVRPTW

This study elucidates the diverse benefits that can lead to improvements in agility and efficiency in logistics and delivery operations. One of the key practical outcomes is the reduction of service time per pallet by half, which significantly enhances the speed and frequency of deliveries. Moreover, the study conducts a cost-benefit analysis to demonstrate the value of investing in vehicle modifications. The research suggests that despite the initial costs of modifications, the long-term benefits, such as reduced labor costs, decreased fuel usage, and improved delivery efficiency, make the investment worthwhile.

Additionally, the modifications tested can be adapted and scaled for various sizes and types of operations. This flexibility adds practical value and serves as a model for similar efficiency-enhancing projects. It also supports sustainable development in logistics. The improvements in delivery and reduction in service times help vehicles use less fuel and lead to lower carbon emissions per delivery. These modifications align with global goals to reduce the carbon footprint of the logistics industry, offering practical methods to enhance business practices in an environmentally friendly way.

CHAPTER 5

CONCLUSION

5.1 Evaluating the Impact of Truck Modifications

This study illustrates the improved outcomes of using modified trucks compared to unmodified ones, specifically in terms of reduced transportation times and distances. Initially, the five unmodified trucks required a total of 1,653 minutes to complete their deliveries, reflecting a high demand for service time per unit. Conversely, with the modifications, the total delivery time was reduced by 267 minutes, and the transportation distance was decreased by 6.9 kilometers. Additionally, there were changes in the delivery routes compared to those used by the unmodified trucks. These reductions and route changes not only demonstrate increased transportation efficiency but also emphasize the benefits of reducing service time. The modifications enabled the trucks to optimize their routes more effectively and improve their ability to meet high customer demands.

5.2 Assessing the Payback Period for Truck Modifications

The total investment for modifying all trucks amounted to 66,000 baht, a substantial sum. Assessing the payback period necessitated considering the cost per delivery, inclusive of labor costs per hour and fuel costs per kilometer. Upon calculation, the cost per delivery was found to be 472.6 baht. With the total time savings of 267 minutes and a distance of 6.9 kilometers per delivery, tangible savings were evident. The modification costs could be recovered within approximately 140 deliveries.

While the initial costs of truck modifications were high, the study results indicate that this investment was valuable when considering the reduction in delivery times and the enhanced responsiveness to market demands. This investment not only improved delivery capabilities and service time management but also increased flexibility in adjusting travel routes and deliveries according to customer needs. Moreover, it aids in long-term cost reduction, as reduced transportation time may lower labor costs per hour. In the case of companies paying wages daily, it might eliminate

overtime wages due to cases where unmodified trucks might necessitate overtime to meet delivery deadlines. Similarly, in the second scenario where trucks were modified, the absence of overtime hours for drivers could lead to savings in labor costs.



REFERENCES

- Arenas, I. G. P., Sánchez, A. G., Armando, C., Solano, L., & Medina, L. B. R. (2017). CVRPTW model applied to the collection of food donations. In *Proceedings of the International Conference on Industrial Engineering and Operations Management* (pp. 1307-1314).
- Baker, B. M., & Ayechev, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5), 787-800.
- Bruniecki, K., Chybicki, A., Moszyński, M., & Bonecki, M. (2016, September). Towards solving heterogeneous fleet vehicle routing problem with time windows and additional constraints: real use case study. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 725-730). IEEE.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4), 568-581.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- El Hassani, A. H., Bouhafs, L., & Koukam, A. (2008). A hybrid ant colony system approach for the capacitated vehicle routing problem and the capacitated vehicle routing problem with time windows (p. 57). Rijeka: IntechOpen.
- Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The vehicle routing problem: latest advances and new challenges* (Vol. 43). Springer Science & Business Media.
- Kumar, S. N., & Panneerselvam, R. (2012). A survey on *the vehicle routing problem and its variants*.
- Lin, L., Hu, J., & Shen, B. (2010). A new hybrid method of Genetic Algorithm, Tabu and Chaotic search for CVRPTW. *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 336-340.
<https://doi.org/10.1109/ICICISYS.2010.5658353>
- Natalia, C., Triyanti, V., Setiawan, G., & Haryanto, M. (2021). Completion of capacitated vehicle routing problem (cvrp) and capacitated vehicle routing

- problem with time windows (cvrptw) using bee algorithm approach to optimize waste picking transportation problem. *Journal of Modern Manufacturing Systems and Technology*, 5(2), 69-77.
- Romero-Gelvez, J. I., Gonzales-Cogua, W. C., & Herrera-Cuartas, J. A. (2019). Cvrptw model for cargo collection with heterogeneous capacity-fleet. In *Applied Informatics: Second International Conference, ICAI 2019, Madrid, Spain, November 7–9, 2019, Proceedings 2* (pp. 173-184). Springer International Publishing.
- Sooktip, T., & Wattanapongsakorn, N. (2015). Identifying preferred solutions for multi-objective optimization: application to capacitated vehicle routing problem. *Cluster Computing*, 18, 1435-1448.
- Tebaldi, L., Murino, T., & Bottani, E. (2020). An adapted version of the water wave optimization algorithm for the capacitated vehicle routing problem with time windows with application to a real case using probe data. *Sustainability*, 12(9), 3666.
- Toth, P., & Vigo, D. (Eds.). (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics.
- Wahyuningsih, S., Satyananda, D., & Octoviana, L. T. (2019). Vehicle Routing Problem with Time Windows Variants and its Application in Distribution Optimization.
- Yogaswara, Y. (2015). Route and Transportation Costs Analysis Considering the City Logistic System for Single Depot Problem (Case Study: JABABEKA Industrial Area Companies). *Proceeding ISIEM*.
- Zhang, H., Ge, H., Yang, J. et al. (2022). Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms. *Arch Computat Methods Eng* 29, 195–221. <https://doi.org/10.1007/s11831-021-09574-x>
- Zirour, M. (2008). Vehicle routing problem: models and solutions. *Journal of Quality Measurement and Analysis JQMA*, 4(1), 205-218.



APPENDICES

APPENDIX A

THE CODE FOR CASE STUDY 1

1. Import and Setup

The script starts by importing necessary modules from the OR-Tools library. It defines the problem data, such as travel times between locations, time windows for deliveries, vehicle capacities, and the number of vehicles.

```
!pip install ortools
# Installs the OR-Tools library, which includes optimization tools for solving complex
routing problems.
from __future__ import print_function
# Ensures that the print function behaves like Python 3.x even if used in Python 2.x.
from ortools.constraint_solver import routing_enums_pb2
# Imports specific enums needed for setting up the routing parameters.
from ortools.constraint_solver import pywrapcp
# Imports the main constraint solver interface from OR-Tools.
# Defines a matrix of travel times between locations (including the depot).
time_matrix = [
    [0, 37, 29, 27, 17, 36, 20, 30, 40, 26, 22, 18, 34, 88, 28, 30],
    [35, 0, 30, 40, 34, 17, 33, 24, 38, 44, 38, 38, 24, 16, 39, 31],
    [29, 28, 0, 25, 26, 20, 18, 10, 28, 29, 23, 30, 20, 24, 23, 22],
    [25, 44, 23, 0, 20, 33, 20, 26, 37, 6, 15, 18, 32, 91, 21, 25],
    [17, 33, 26, 23, 0, 36, 20, 30, 38, 24, 20, 18, 35, 87, 23, 23],
    [33, 17, 29, 31, 30, 0, 28, 19, 28, 35, 28, 34, 14, 75, 31, 32],
    [16, 33, 20, 15, 12, 25, 0, 16, 32, 18, 12, 15, 24, 82, 15, 15],
    [28, 27, 15, 25, 24, 25, 15, 0, 38, 30, 22, 29, 24, 81, 20, 18],
    [39, 37, 22, 35, 37, 25, 34, 31, 0, 32, 30, 36, 18, 24, 27, 36],
    [25, 45, 23, 10, 24, 36, 22, 29, 32, 0, 18, 17, 35, 100, 22, 26],
    [19, 37, 17, 13, 15, 23, 14, 21, 31, 17, 0, 17, 28, 89, 16, 18],
```

```
[15, 34, 21, 17, 14, 35, 17, 26, 35, 13, 16, 0, 31, 91, 20, 23],
[33, 26, 21, 26, 30, 13, 24, 20, 19, 34, 27, 33, 0, 68, 30, 27],
[88, 63, 82, 87, 85, 70, 82, 77, 79, 91, 84, 90, 70, 0, 87, 85],
[23, 34, 15, 19, 20, 37, 18, 18, 26, 21, 18, 22, 25, 75, 0, 17],
[26, 31, 21, 20, 16, 30, 12, 35, 34, 23, 22, 24, 30, 87, 18, 0]]
```

Travel times from the depot to each customer and back.

```
time_depot = [0,55,57,80,75,47,64,70,69,66,53,79,61,85,71,66]
```

Specifies the allowable time windows for each location, including the depot.

```
time_windows = [
  (480, 1440), # depot
  (540, 780), # 1
  (540, 840), # 2
  (720, 960), # 3
  (600, 900), # 4
  (660, 960), # 5
  (720, 1020), # 6
  (720, 900), # 7
  (600, 840), # 8
  (540, 840), # 9
  (600, 840), # 10
  (720, 960), # 11
  (600, 840), # 12
  (660, 960), # 13
  (600, 900), # 14
  (780, 900) # 15
]
```

]

Specifies the demand (e.g., quantity to pick up or deliver) at each location, including zero for the depot.

```
demands = [0,12,10,8,5,6,7,10,7,9,10,5,4,12,3,8]
```

num_vehicles = 5 # Specifies the number of vehicles available.

vehicle_capacities = [25, 25, 25, 25, 25] # Specifies the capacity of each vehicle.

```
depot_index = 0
```

```
# Indicates that the depot's index is 0 in the time matrix and demand array.
time_limit_seconds = 1200 # Sets a computation time limit for solving the problem.
```

2. Data Model Function

The `create_data_model` function is defined to encapsulate the problem data in a dictionary, setting up parameters like the time matrix, time windows, vehicle capacities, demands, and depot information.

```
def create_data_model():
    """Stores the data for the problem in a dictionary."""
    data = {}
    data['time_matrix'] = time_matrix
    data['time_windows'] = time_windows
    data['num_vehicles'] = num_vehicles
    data['demands'] = demands
    data['vehicle_capacities'] = vehicle_capacities
    data['depot'] = depot_index
    data['service_time'] = 10 # Sets the service time at each location.
    return data
```

3. Solution Printing

The `print_solution` function extracts and prints the detailed routing solution, including the route for each vehicle, the sequence of visits, arrival and departure times, and the load carried.

```
def print_solution(data, manager, routing, solution):
    """Extracts and prints the solution of the routing problem."""
    total_load = 0
    total_route_time = 0 # Initializes the total route time to zero.
    print('Objective: { } minutes'.format(solution.ObjectiveValue()))
    # Iterates over each vehicle to extract and print its route.
    for vehicle_id in range(data['num_vehicles']):
```

```

index = routing.Start(vehicle_id)
plan_output = 'Route for vehicle { }:\n'.format(vehicle_id)
route_load = 0
# Loop continues until the end of the route is reached.
while not routing.IsEnd(index):
    node_index = manager.IndexToNode(index)
    route_load += data['demands'][node_index]
    time_dimension = routing.GetDimensionOrDie('Time')
    arrival_time = solution.Min(time_dimension.CumulVar(index))
    leave_time = arrival_time + data['service_time'] *
data['demands'][node_index]
    plan_output += 'Place {0:2d} Arrive at {1:02d}:{2:02d} Depart at
{3:02d}:{4:02d} (Load {5:2d}) Service Time: {6}min\n'.format(
    node_index, arrival_time // 60, arrival_time % 60, leave_time // 60,
leave_time % 60, route_load, data['service_time'] * data['demands'][node_index])
    index = solution.Value(routing.NextVar(index))
# Once the route is complete, the final destination time and load are calculated.
plan_output += 'Return to depot Arrive at { :02d}:{ :02d}\n'.format(arrival_time //
60, arrival_time % 60)
plan_output += 'Load of the route: { }\n'.format(route_load)
plan_output += 'Time of the route: { }min\n\n'.format(time_of_the_route)
print(plan_output)
total_load += route_load
total_route_time += time_of_the_route
print('Total Load of all routes: { }'.format(total_load))
print('Total Time of all routes: { }min'.format(total_route_time))

```

4. Main Function

The main function sets up and solves the routing problem. It starts by initializing the routing index manager and the routing model using the problem data. It also sets service times for each node based on demands.

```

def main():
    """Defines the main execution logic for setting up and solving the routing
    problem."""
    data = create_data_model() # Initializes the problem data.
    manager = pywrapcp.RoutingIndexManager(len(data['time_matrix']),
    data['num_vehicles'], data['depot']) # Creates a manager for the routing indices.
    routing = pywrapcp.RoutingModel(manager) # Creates the routing model.

```

5. Callbacks and Constraints

Time Callback

Defines a callback that computes the travel time between nodes, adding service times based on the demand at each node.

```

# Creates and registers a callback that calculates the travel time between locations.
def time_callback(from_index, to_index):
    from_node = manager.IndexToNode(from_index)
    to_node = manager.IndexToNode(to_index)
    service_time = 10 * data['demands'][from_node]
# Calculates additional service time based on demands.
    return data['time_matrix'][from_node][to_node] + service_time
    transit_callback_index = routing.RegisterTransitCallback(time_callback)
    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)

```

Demand Callback

Sets up a callback for handling demands at each node and adds a capacity constraint to ensure vehicles do not exceed their load limits.

```

# Adds capacity constraints.
def demand_callback(from_index):
    from_node = manager.IndexToNode(from_index)
    return data['demands'][from_node]

```

```

demand_callback_index =
routing.RegisterUnaryTransitCallback(demand_callback)
routing.AddDimensionWithVehicleCapacity(
    demand_callback_index,
    0, # No slack for the capacity.
    data['vehicle_capacities'], # Maximum capacity for each vehicle.
    True, # Start cumulative variable at zero.
    'Capacity')

```

Time Dimension

Adds a time dimension for managing route timing, setting constraints for each location based on predefined time windows.

Adds time window constraints.

```

time = 'Time'
routing.AddDimension(
    transit_callback_index,
    0, # No slack for time.
    999999, # High upper bound for the cumulative time variable.
    False, # Do not force the start cumulative variable to zero.
    time)

```

```
time_dimension = routing.GetDimensionOrDie(time)
```

Adds time window constraints for each location.

```

for location_idx, time_window in enumerate(data['time_windows']):
    index = manager.NodeToIndex(location_idx)
    time_dimension.CumulVar(index).SetRange(
        time_window[0], time_window[1])

```

Adds constraints for vehicle start and end times.

```

for vehicle_id in range(data['num_vehicles']):
    index = routing.Start(vehicle_id)
    time_dimension.CumulVar(index).SetRange(
        data['time_windows'][0][0],

```

```

    data['time_windows'][0][1])
# Optimizes the starting and ending times of routes.
    for i in range(data['num_vehicles']):
routing.AddVariableMinimizedByFinalizer(time_dimension.CumulVar(routing.Start(
i)))
routing.AddVariableMinimizedByFinalizer(time_dimension.CumulVar(routing.End(i
)))

```

6. Optimization and Solution

Defines search parameters, including the strategy and time limit for the solver. It then attempts to solve the routing problem and prints the solution if found.

```

# Sets the parameters for the routing algorithm.
search_parameters = pywrapcp.DefaultRoutingSearchParameters()
search_parameters.first_solution_strategy = (
    routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC)
search_parameters.time_limit.seconds = time_limit_seconds
# Sets the time limit for solving.
solution = routing.SolveWithParameters(search_parameters) # Solves the problem.
# Prints the solution.
if solution:
    print_solution(data, manager, routing, solution)
else:
    print("No solution found.")
return solution
solution = main() # Executes the main function and returns the solution.

```

APPENDIX B

THE CODE FOR CASE STUDY 2

1. Import and Setup

The script starts by importing necessary modules from the OR-Tools library. It defines the problem data, such as travel times between locations, time windows for deliveries, vehicle capacities, and the number of vehicles.

```
!pip install ortools
# Install the OR-Tools library, which provides tools for solving complex optimization
# problems.
from __future__ import print_function
from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp
# Define a matrix that represents the travel times between each pair of locations.
time_matrix = [
    [0, 37, 29, 27, 17, 36, 20, 30, 40, 26, 22, 18, 34, 88, 28, 30],
    [35, 0, 30, 40, 34, 17, 33, 24, 38, 44, 38, 38, 24, 16, 39, 31],
    [29, 28, 0, 25, 26, 20, 18, 10, 28, 29, 23, 30, 20, 24, 23, 22],
    [25, 44, 23, 0, 20, 33, 20, 26, 37, 6, 15, 18, 32, 91, 21, 25],
    [17, 33, 26, 23, 0, 36, 20, 30, 38, 24, 20, 18, 35, 87, 23, 23],
    [33, 17, 29, 31, 30, 0, 28, 19, 28, 35, 28, 34, 14, 75, 31, 32],
    [16, 33, 20, 15, 12, 25, 0, 16, 32, 18, 12, 15, 24, 82, 15, 15],
    [28, 27, 15, 25, 24, 25, 15, 0, 38, 30, 22, 29, 24, 81, 20, 18],
    [39, 37, 22, 35, 37, 25, 34, 31, 0, 32, 30, 36, 18, 24, 27, 36],
    [25, 45, 23, 10, 24, 36, 22, 29, 32, 0, 18, 17, 35, 100, 22, 26],
    [19, 37, 17, 13, 15, 23, 14, 21, 31, 17, 0, 17, 28, 89, 16, 18],
    [15, 34, 21, 17, 14, 35, 17, 26, 35, 13, 16, 0, 31, 91, 20, 23],
    [33, 26, 21, 26, 30, 13, 24, 20, 19, 34, 27, 33, 0, 68, 30, 27],
    [88, 63, 82, 87, 85, 70, 82, 77, 79, 91, 84, 90, 70, 0, 87, 85],
    [23, 34, 15, 19, 20, 37, 18, 18, 26, 21, 18, 22, 25, 75, 0, 17],
```

```

[26, 31, 21, 20, 16, 30, 12, 35, 34, 23, 22, 24, 30, 87, 18, 0]]
# Defines the time to reach each location from the depot.
time_depot = [0,55,57,80,75,47,64,70,69,66,53,79,61,85,71,66]
# Specifies the time windows for each location, including the depot.
time_windows = [
    (480, 1440), # depot
    (540, 780), # 1
    (540, 840), # 2
    (720, 960), # 3
    (600, 900), # 4
    (660, 960), # 5
    (720, 1020), # 6
    (720, 900), # 7
    (600, 840), # 8
    (540, 840), # 9
    (600, 840), # 10
    (720, 960), # 11
    (600, 840), # 12
    (660, 960), # 13
    (600, 900), # 14
    (780, 900) # 15
]
# Specifies the demands at each location, including zero at the depot.
demands = [0,12,10,8,5,6,7,10,7,9,10,5,4,12,3,8]
num_vehicles = 5 # Number of vehicles available for routing.
vehicle_capacities = [25, 25, 25, 25, 25] # Capacity of each vehicle, defining how
much each can carry.
depot_index = 0 # Index of the depot in the data arrays.
time_limit_seconds = 1200 # Sets a time limit for the solver to find a solution.

```

2. Data Model Function

The `create_data_model` function is defined to encapsulate the problem data in a dictionary, setting up parameters like the time matrix, time windows, vehicle capacities, demands, and depot information.

```
def create_data_model():
    """Store the problem data in a structured dictionary to pass around."""
    data = { }
    data['time_matrix'] = time_matrix
    data['time_windows'] = time_windows
    data['num_vehicles'] = num_vehicles
    data['demands'] = demands
    data['vehicle_capacities'] = vehicle_capacities
    data['depot'] = depot_index
    data['service_times'] = [5, 10]
    # Different service times per vehicle type (e.g., 5 minutes for modified trucks).
    data['vehicle_types'] = [0, 0, 1, 1, 1]
    # Vehicle types, where 0 might indicate standard and 1 modified.
    return data
```

3. Solution Printing

The `print_solution` function extracts and prints the detailed routing solution, including the route for each vehicle, the sequence of visits, arrival and departure times, and the load carried.

```
def print_solution(data, manager, routing, solution):
    """Print the routing solution in a readable format."""
    total_load = 0
    total_time = 0
    time_dimension = routing.GetDimensionOrDie('Time')
    # Retrieve the time dimension defined earlier.
```

```

for vehicle_id in range(data['num_vehicles']):
    index = routing.Start(vehicle_id) # Get the start index for this vehicle.
    plan_output = f'Route for vehicle {vehicle_id}: Type
{data["vehicle_types"][vehicle_id]}\n'
    route_load = 0
    while not routing.IsEnd(index):
        node_index = manager.IndexToNode(index)
        next_index = solution.Value(routing.NextVar(index))
        route_load += data['demands'][node_index]
        arrival_time = solution.Min(time_dimension.CumulVar(index))
        depart_time = arrival_time +
data['service_times'][data['vehicle_types'][vehicle_id]] * data['demands'][node_index]
        plan_output += f'Place {node_index} Arrive at {arrival_time //
60:02d}:{arrival_time % 60:02d} Depart at {depart_time // 60:02d}:{depart_time %
60:02d} (Load {route_load})\n'
        index = next_index
    end_index = routing.End(vehicle_id)
    end_arrival_time = depart_time + data['time_matrix'][node_index][data['depot']]
    total_route_time = end_arrival_time -
solution.Min(time_dimension.CumulVar(routing.Start(vehicle_id)))
    plan_output += f'Return to depot Arrive at {end_arrival_time //
60:02d}:{end_arrival_time % 60:02d}\n'
    plan_output += f'Load of the route: {route_load}\n'
    plan_output += f'Time of the route: {total_route_time}min\n\n'
    print(plan_output)
    total_load += route_load
    total_time += total_route_time
print(f'Total load of all routes: {total_load}')
print(f'Total time of all routes: {total_time}min')

```

4. Main Function

The main function sets up and solves the routing problem. It starts by initializing the routing index manager and the routing model using the problem data. It also sets service times for each node based on demands.

```
def main():
    """Setup and solve the vehicle routing problem."""
    data = create_data_model() # Initialize the problem data.
    manager = pywrapcp.RoutingIndexManager(len(data['time_matrix']),
data['num_vehicles'], data['depot']) # Create a manager for routing indices.
    routing = pywrapcp.RoutingModel(manager) # Initialize the routing model.
```

5. Callbacks and Constraints

Time Callback

This snippet sets up a callback function to return the travel time between any two locations, which is essential for the solver to understand the costs associated with moving from one point to another.

Create and register a callback function that returns the travel time between nodes.

```
def time_callback(from_index, to_index):
    from_node = manager.IndexToNode(from_index)
    to_node = manager.IndexToNode(to_index)
    return data['time_matrix'][from_node][to_node]
    # Return travel time between nodes.
transit_callback_index = routing.RegisterTransitCallback(time_callback)
routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)
    # Set the cost evaluator for all vehicles using the callback.
```

Demand Callback

These lines add constraints for vehicle capacity and set up time windows for each location using the OR-Tools routing model's dimension functions.

Create and register a callback for demands at each location.

```
def demand_callback(from_index):
    from_node = manager.IndexToNode(from_index)
    return data['demands'][from_node] # Return the demand at the node.
demand_callback_index =
routing.RegisterUnaryTransitCallback(demand_callback)
routing.AddDimensionWithVehicleCapacity(demand_callback_index, 0,
data['vehicle_capacities'], True, 'Capacity') # Add capacity constraints.
```

Time Dimension

Adds a time dimension for managing route timing, setting constraints for each location based on predefined time windows.

Add time dimension to respect time windows for each location.

```
time = 'Time'
routing.AddDimension(transit_callback_index, 0, 99999, False, time)
time_dimension = routing.GetDimensionOrDie(time)
for location_idx, time_window in enumerate(data['time_windows']):
    if location_idx == data['depot']:
        continue # Skip depot for setting time windows.
    index = manager.NodeToIndex(location_idx)
    time_dimension.CumulVar(index).SetRange(time_window[0], time_window[1])
# Set time windows for each location.
```

6. Optimization and Solution

Defines search parameters, including the strategy and time limit for the solver. It then attempts to solve the routing problem and prints the solution if found.

Set up and configure search parameters for the solver.

```
search_parameters = pywrapcp.DefaultRoutingSearchParameters()
search_parameters.first_solution_strategy =
routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC
```

Set initial search strategy.

```
solution = routing.SolveWithParameters(search_parameters)
```

Solve the problem with the configured parameters.

```
if solution:
```

```
    print_solution(data, manager, routing, solution) # Print the solution if found.
```

```
else:
```

```
    print('Solution not found. Please check the problem constraints.')
```

Print error if no solution.

```
if __name__ == '__main__':
```

```
    main() # Execute the main function.
```

BIOGRAPHY

| | |
|-----------|--|
| Name | Burevarat Vongburee |
| Education | 2021: Bachelor of Science in Integrated Program in Management Mathematics (IMMA) Faculty of Science and Technology Thammasat University |

