MASTERARBEIT | MASTER'S THESIS

Titel | Title

SOLVING THE MULTI-DEPOT ATTENDED HOME DELIVERY PROBLEM WITH DYNAMIC CUSTOMER ACCEPTANCE

verfasst von | submitted by

Minh Trang Hoang

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 977
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Business Analytics
Betreut von | Supervisor:	Univ.-Prof. Dr. Jan Fabian Ehmke

Abstract (DE)

Diese Dissertation befasst sich mit dem Multi-Depot-Attended-Home-Delivery-Problem (AHD) mit dynamischer Kundenakzeptanz und konzentriert sich auf die Optimierung der Akzeptanzraten sowie der Erfüllungskosten durch die Analyse variierender Depotanzahlen und Zeitfensterdauern. Die Studie integriert fortschrittliche Optimierungstechniken, darunter Beam Search, lokale Suche (2-opt) und Simulated Annealing, zur effizienten Routen- und Zeitplanerstellung in dynamischen Umgebungen.

Die Ergebnisse zeigen, dass mehrere Depots die Systemleistung gegenüber Einzeldepot-Modellen erheblich verbessern. Ein Multi-Depot-System erhöht die Kundenakzeptanz um 5–15% und senkt die Erfüllungskosten um über 5%. Zudem ermöglicht es kürzere Lieferzeitfenster von 2 Stunden bis 30 Minuten bei stabiler Akzeptanzrate, was die Skalierbarkeit und Effizienz solcher Konfigurationen unterstreicht.

Diese Arbeit hebt die praktischen Vorteile verteilter Operationen hervor, wie verbesserte Arbeitslastverteilung, optimierte Routenplanung und Echtzeitentscheidungen durch dynamische Akzeptanzmechanismen. Diese Beiträge bieten einen robusten Rahmen zur Bewältigung der Komplexität dynamischer Kundenakzeptanz in AHD-Systemen und liefern wertvolle Erkenntnisse für skalierbare, kundenorientierte Logistiklösungen.

Keywords: multi-depot, VRPTW, MDVRPTW, dynamic customer acceptance

Abstract (EN)

This thesis addresses the multi-depot attended home delivery (AHD) problem with dynamic customer acceptance, focusing on optimizing customer acceptance rates and fulfillment costs by analyzing the impact of varying the number of depots and time window durations. To achieve this, the study integrates advanced optimization techniques, including Beam Search, local search 2-opt, and the meta-heuristic Simulated Annealing, for efficient routing and scheduling in dynamic environments.

Key findings reveal that incorporating multiple depots significantly enhances system performance compared to traditional single-depot models. A multi-depot system increases customer acceptance rates by over 5% and up to 15% while reducing fulfillment costs by more than 5%. Furthermore, it enables the reduction of delivery time windows from 2 hours to as short as 30 minutes, maintaining a competitive acceptance rate and demonstrating the scalability and efficiency of multi-depot configurations.

This work highlights the practical benefits of distributing operations across multiple depots, such as improved workload balance, enhanced route optimization, and real-time decision-making through dynamic acceptance mechanisms. These contributions provide a robust framework for addressing the complexities of dynamic customer acceptance in AHD systems, offering insights into scalable and customer-centric logistics solutions.

Keywords: multi-depot, VRPTW, MDVRPTW, dynamic customer acceptance

Acknowledgements

I extend my heartfelt gratitude to Prof. Ehmke and my incredible supervisor, Steffen Elting, for their invaluable guidance, support, and encouragement throughout this project. Your expertise and insights have been instrumental in the success of this work, and I am deeply appreciative of the time and effort you have dedicated to helping me achieve my goals.

I would like to express my sincere gratitude to all the lecturers whose excellent classes have greatly strengthened my knowledge and skills, enabling me to undertake and complete this thesis. Your dedication to teaching and the insights you have shared have been invaluable to my academic journey.

Author's declaration

I hereby declare that this thesis, titled "Solving the Multi-depot Attended Home Delivery Problem with Dynamic Customer Acceptance", is my original work and has not been submitted for any degree or diploma at any other university or institution. All sources of information and data used in this work have been appropriately acknowledged.

I certify that the content of this thesis is the result of my own research and analysis, except where explicitly stated otherwise.

SIGNED: .... DATE: ....

List of Tables

List of Figures

Figure **Page**

Table of Contents

Chapter 1

Introduction

The COVID-19 pandemic has boosted the demand for online shopping and home delivery across the globe, and it is likely that some shifts in demand will also have long-lasting effects (OECD, 2020). The Europe E-Commerce Report 2023 (Lone, 2023) indicated the significance of B2C e-commerce, revealing that 78% of EU internet users purchase products and services online, compared to 67% in 2018. The COVID-19 pandemic greatly increased the growth of online shopping, creating a strong foundation for the future of e-commerce. One of the most important changes was the rise of grocery home delivery, which became an essential service for many people. This shift changed how consumers shop and forced retailers to adapt, making grocery delivery a key part of the growing e-commerce market.

Benefitting from the shift in consumer habits, grocery home delivery presents vast opportunities and holds more potential than ever before. Several startups have entered the e-grocery industry in recent years but have failed to prove themselves. Grocery e-commerce requires high development costs to provide a wide variety of products, intricate reservation systems, and delivery systems, making it difficult for small startups to manage effectively. For instance, JOKR, a Berlin-based unicorn startup that aimed to deliver groceries in under 15 minutes across several countries, including Germany and the Netherlands, decided to exit the European market due to high operational costs and logistical challenges in 2022 (Lockwood and Burroughs, 2022). JOKR sets up many mini storefronts on the side streets in some specific areas, once the order has been placed, the deliverer picks it up and delivers it to the customer instantly. The high maintenance costs and difficulty with inventory management put JOKR under a lot of pressure to achieve profitability. Gorillas, once known as Germany's fastest-growing grocery delivery unicorn, failed to achieve profitability after a period of substantial cash burn in the market. Gorillas' model involved owning and operating their own dark stores, where they stored and sold groceries directly to customers. Gorillas' failure can be attributed to operational and workforce-related challenges (Apptunix, 2023). Many couriers expressed dissatisfaction with their working conditions with high delivery pressures. They were often required to search through hundreds of shelves in the warehouse to locate items for each order in a short period of time to ensure fast-delivery. In 2022, Gorillas was sold to its competitor Getir, a Turkish rapid grocery delivery company, for $1.2 billion. Getir, however, could not escape the harsh realities of the competitive rapid delivery market. Getir decided to quit the UK, Germany, the Netherlands and the US to focus on its home market of Turkey in 2024 (Butler, 2024).

The failure of rapid grocery delivery startups can be primarily attributed to the complexities of managing the supply chain, delivery logistics, inventory management, and high marketing costs. Companies like JOKR, Gorillas, and Getir faced significant challenges in maintaining efficient supply chains capable of supporting instant deliveries. Managing deliveries within tight timeframes placed pressure on couriers, further increasing operational inefficiencies. In addition, these companies invested heavily in marketing to acquire and retain customers in a competitive market, further driving up costs. The combination of these challenges made it difficult for these startups to scale profitably, ultimately contributing to their struggles.

With the advantages of physical stores spread throughout the city and the decline of numerous competitors in e-grocery, two largest supermarket chains in Austria, BILLA and SPAR, have prioritized enhancing their online shopping platforms. For instance, BILLA, a leading supermarket chain in Austria, has focused on boosting its efforts for its BILLA e-shop and online shopping platform. According to REWE (2023), BILLA expanded its online product range by 50% over three years, increasing from approximately 8,000 to 12,000 items. This expansion includes over 2,300 plant-based products, more than 1800 organic items, and around 1,300 gluten-free options. BILLA e-shop now guarantees delivery within 3 hours, replacing the previous next-day delivery service. Similarly, since 2021, SPAR Austria has significantly enhanced its online presence and delivery services to meet the growing demand for e-grocery shopping with more than 20,000 products. In 2023, SPAR Austria upgraded its INTERSPAR online shop by implementing composable commerce technology, aiming to improve customer experience and boost sales (Contentstack, 2023).

Learning from the failures of short-term commitment delivery services, many organizations understand that ensuring efficient and cost-effective last-mile delivery operations is essential, especially during the final stage of attended home delivery (AHD), where the customer must be present to receive the merchandise. The last mile is currently regarded as one of the most expensive, least efficient, and most polluting sections of the entire supply chain (Gevaers et al., 2011). The struggles of companies like JOKR, Gorillas, and Getir highlight the complexities of maintaining rapid and reliable delivery services, especially within the constraints of tight delivery windows. In these cases, the lack of a scalable, sustainable delivery model led to high operational costs and increased inefficiencies, ultimately preventing these companies from achieving profitability. As a result, most grocery retailers prefer only scheduled delivery rather than instant delivery, as scheduled delivery is more environmentally friendly and has better logistical efficiency and cost-effectiveness. Scheduled delivery allows companies to plan routes and delivery times in advance and optimizing the use of available resources. Typically, delivery vehicles collect groceries from one or more stores and sequentially deliver them to customers living in nearby areas. This approach minimizes travel distances, reduces fuel consumption, and lowers greenhouse gas emissions, along with sustainability goals. With the increasing popularity of online grocery shopping, managing deliveries efficiently has become more challenging, particularly in urban areas where traffic congestion and high population density add complexity.

To address the difficulty in delivery, businesses are focusing on improving multi-depot delivery systems, which enhance flexibility and scalability. By distributing multiple depots across a city, retailers can handle higher order volumes while ensuring that no single depot becomes a bottleneck. This allows for a more even workload distribution, enabling quicker response times to customer orders and improved reliability in delivery schedules. Additionally, a well-implemented multi-depot strategy reduces overall operational costs by cutting down on unnecessary vehicle lease cost and fuel expenses. It also allows for better inventory management, as products can be distributed across multiple locations based on demand patterns. For instance, high-demand items can be stocked closer to areas with frequent orders, reducing the need for cross-city deliveries and improving order fulfillment speed.

Overall, the shift towards scheduled delivery and multi-depot strategies represents a significant evolution in grocery delivery logistics. By optimizing delivery routes, depot usage, and resource allocation, grocery retailers can achieve operational excellence, reduce costs, and provide superior service to their customers—all while minimizing their environmental impact. This approach underscores the importance of strategic planning and technological integration in meeting the growing demand for efficient and sustainable e-grocery solutions.

The multi-depot attended home delivery problem is a complicated problem that considers both demand management and delivery optimization. While significant research has been conducted on single-depot AHD problems, the extension to multi-depot systems—especially considering dynamic customer acceptance—remains an area with substantial research potential. Multi-depot AHD systems introduce additional layers of complexity, such as coordinating between multiple depots, managing overlapping delivery areas, and balancing workloads across depots, which are not present in single-depot models. This research concentrates solely on optimizing delivery systems, excluding considerations related to demand management. The primary focus of this work is to analyze the impact of varying the number of depots and the duration of time windows on both customer acceptance rates and fulfillment costs within the framework of the Attended Home Delivery (AHD) problem. The results can also be used as a benchmark for horizontal collaboration solutions in the AHD problem (Elting et al., 2024*a*).

Chapter 2

Literature Review

This literature review provides the necessary knowledge on the Vehicle Routing Problem (VRP) and the concept of attended home delivery (AHD). It also explores algorithms used to solve VRP, highlighting both classical and modern approaches. The review aims to present a basic understanding of the current state of research in these areas, offering insights into the challenges and advancements in optimizing vehicle routing and delivery operations.

2.1 Overview of Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP)

2.1.1 Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) has been extensively researched in combinatorial optimization with many valuable real-life applications. TSP involves determining the most efficient route connecting several cities and returning to the initial city to minimize the total cost (time, distance). Despite its simple definition, TSP is classified as an NP-hard problem, meaning that solving it exactly becomes increasingly difficult as the number of cities grows. This has made it a key topic in research on computational complexity and optimization. Dantzig et al. (1954) made a significant first contribution with their linear programming techniques to successfully tackle a TSP problem involving 49 cities. Throughout the years, TSP has become a widely studied problem in optimization techniques and has significantly influenced the advancement of various algorithms in operations research and computer science.

The Traveling Salesman Problem (TSP) has been the subject of extensive research for many years, resulting in the development of numerous theoretical solutions. The most straightforward approach is to test all possible options, yet this strategy is also the most time-consuming and costly, especially when dealing with a large number of instances. Researchers have explored exact methods for solving TSP, which guarantee finding the best solution. Techniques such as branch-and-bound (A. H. Land, 1960), branch-and-cut (M. Padberg, 1987), and dynamic programming (Michael Held, 1961) have been developed to tackle small and medium dataset. While these methods are powerful, their computational cost grows quickly with the number of cities, making them impractical for large problems.

To address this limitation, researchers have developed heuristic and metaheuristic approaches that provide approximate solutions. Early heuristic methods, which provide probabilistic outputs, such as Sthe Savings Heuristic (Clarke and Wright, 1964), Greedy Heuristic, and Insertion Heuristic (Rosenkrantz, 1977) focused on creating efficient but approximate solutions. These methods inspired more advanced metaheuristics, including Simulated Annealing (Kirkpatrick, 1983) and Tabu Search (Glover, 1986). These approaches are widely used because they can handle larger problems and explore many possible solutions effectively, even though they do not guarantee the best solution.

2.1.2 Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) consists of finding the optimal set of vehicle routes for some constraints, initially proposed by Dantzig and Ramser (1959). The VRP expands upon the concepts of TSP by considering vehicle constraints. The TSP can be treated as a relaxed classic VRP with no restrictions on capacity. Extending from TSP, the objective of VRP can be single-objective or multi-objective. The primary objective of the VRP is to minimize costs, which could include total travel distance, total travel time, or the number of vehicles used, depending on the problem context. The problem may also involve multiple objectives, such as reducing fuel consumption, minimizing emissions, or balancing workloads among vehicles, leading to the formulation of multi-objective VRP variants. The flexibility of the VRP framework makes it highly adaptable to a variety of industries, including transportation, e-commerce, waste collection, and public services.

Similar to TSP, solving the VRP for large instances can be challenging due to its NP-hard characteristics. As a result, researchers have proposed numerous variants of the VRP to address specific logistical scenarios. These include the Capacitated VRP (CVRP), which incorporates vehicle capacity limits; the VRP with Time Windows (VRPTW), where deliveries must be made within specified time intervals; and the Multi-Depot VRP (MDVRP), which involves multiple starting points for vehicles. The VRP has been extensively studied using different methods. Other notable variants include the Open VRP (where vehicles do not return to the depot), the Stochastic VRP (which accounts for uncertainties in demand or travel times), and the VRP with Pickup and Delivery (VRPPD), where items must be transported between specific locations.

One notable review, "The Vehicle Routing Problem: State-of-the-Art Classification and Review" by Braekers (2016), analyzes literature published between 2009 and 2015. In 2021, Shi-Yi Tan analyzes recent literature published between 2019 and August of 2021. Both studies categorized VRP models based on constraint types and classified solution algorithms into three main categories: exact, heuristic, and metaheuristic methods. According to this summarization, metaheuristic methods dominate the VRP research landscape, reflecting their ability to handle large-scale and complex problems effectively. For instance, in Braekers (2016) research, metaheuristics accounted for a significant proportion of the reviewed studies with **71.25%** of the number of models.

Metaheuristic algorithms for various VRP variants were comprehensively reviewed by Elshaer and Awad (2020). This review covers a wide range of metaheuristics, including Genetic Algorithms, Simulated Annealing, Tabu Search, and Ant Colony Optimization, among others.

2.1.3 Vehicle Routing Problem with Time Window (VRPTW)

As mentioned above, the VRPTW is an extensive optimization problem that extends to the Vehicle Routing Problem (VRP) by including further constraints that specify deliveries or pickups must occur within certain periods of the day, known as time windows. This problem is especially relevant in industries like e-commerce and grocery delivery, where committing to customer-specified delivery time windows is critical for maintaining high-quality standards and operational efficiency. The VRPTW involves a fleet of vehicles, each with limited capacity, delivering goods from a central depot to multiple customers. Additionally, vehicles must return to the depot after completing their routes. The problem's complexity arises from the need to optimize routes while commiting to time constraints, often making VRPTW instances computationally challenging.

The Vehicle Routing Problem with Time Windows (VRPTW) finds extensive application across various industries where timely and efficient delivery is essential. In e-commerce and last-mile delivery, companies like Amazon and Shopee leverage VRPTW to optimize routes, ensuring customer-specified delivery windows are met while minimizing costs and environmental impact. Similarly, grocery delivery services use VRPTW to manage the distribution of perishable goods, maintaining freshness and efficiency. In healthcare logistics, VRPTW ensures the timely delivery of medicines, vaccines, and diagnostic samples, where meeting strict time windows is critical to maintaining treatment schedules and supporting life-saving interventions. Municipal services, such as waste collection and recycling, rely on VRPTW to schedule pickups during specific time periods to minimize disruption and fuel consumption.

Effective solutions to VRPTW must balance route efficiency with strict adherence to time windows. This requires leveraging a combination of computational techniques that include both exact methods and heuristic approaches. Advances in hybrid techniques, which combine exact and heuristic methods, have shown promise in achieving both accuracy and scalability for VRPTW. A comprehensive review of these algorithms and their applications to VRPTW is provided in the following section, highlighting their strengths, limitations, and suitability for various real-world scenarios.

2.2 Attended Home Delivery Problem with Dynamic Customer Acceptance

2.2.1 Attended Home Delivery Problem

AHD has emerged as a critical component of e-commerce logistics. The AHD problem is a combination of technological advancement and strategic business methods to guarantee effective delivery operations. Attended home delivery services require the customer to be present during delivery. AHD is common for home services and products that require special handling, such as groceries or large appliances, or delivery with recipient verification. The provider and the customer must agree on a delivery time frame during the booking process. Usually, the service providers offer customers some possible time windows, and the customers choose one. The AHD's challenge is balancing delivery costs and customer service.

AHD is a complicated organization that involves relationships and trade-offs among many decisions and management. It is then more convenient to classify planning levels:

- Strategic (long term): decisions determine general development policies and broadly shape the company's approach to customer engagement and resource allocation. In AHD, some well-known strategies are service differentiation, market segmentation, infrastructure, and capability development. The objective of service differentiation and market segmentation is to provide consumers with more services that are compatible with their needs, thereby increasing the number of new and loyal customers. In addition, infrastructure and capability development (facility location, network design) must be enhanced to meet shifts in demand and capacity. Long-term investments in facility locations, warehouse networks, and delivery capabilities are critical to adapting to demand fluctuations and scaling operations. Examples include establishing regional hubs and expanding fleet capacity to serve growing customer bases effectively. Implementing these strategies requires a deep understanding of market trends, financial forecasting, and phased execution to balance growth with operational feasibility.

- Tactical (medium term): involves developing specific strategies to efficiently allocate resources and support the achievement of long-term business objectives. In the context of AHD, this involves key decisions such as determining the appropriate fleet size to balance demand and operational costs, defining delivery time windows that align with customer preferences while maintaining logistical efficiency, and setting order cut-off times to ensure sufficient preparation for delivery planning. For instance, ensuring the fleet size is neither excessive nor insufficient optimizes resource utilization, while well-defined time windows enhance customer satisfaction without compromising route optimization. These tactical decisions are critical for translating strategic goals into effective operational practices.

- Operational (short term): focuses on the real-time management of delivery services to ensure smooth and efficient execution. In AHD, it involves decisions such as assigning vehicles to customer orders based on proximity, capacity, and route efficiency, dynamically offering time slots that align with demand and operational constraints, and addressing disruptions like traffic delays or last-minute changes by rerouting vehicles or reallocating resources. These decisions are supported by advanced tools such as real-time tracking, predictive analytics, and dynamic routing algorithms, which help maintain service quality and minimize disruptions.

One of the most significant challenges in AHD lies in optimizing the allocation of delivery slots to ensure both efficiency and flexibility. Service providers must offer time windows that are narrow enough to meet customer expectations yet broad enough to optimize route efficiency and delivery cost.

The work of this thesis is considered an operational solution since it aims to optimize the time windows offered to customers and the total number of accepted orders with the set-up of multi-depot AHD with dynamic customer acceptance.

2.2.2 Attended Home Delivery Problem with Dynamic Customer Acceptance

The process of Attended Home Deliveries comprises five primary steps: (1) receiving a request from a customer, (2) assessing the feasibility of the request, (3) offering customer possible time windows, (4) customer chooses a specific available time window (5) the provider and customer both agree on a delivery time-frame.

A major challenge in this process lies in the feasibility check. This step involves reviewing all possible time windows to find those that fit within the current delivery schedule, vehicle availability, and capacity limits. It requires careful evaluation to ensure the time windows offered can be accommodated without disrupting the current delivery plan. A key challenge lies in ensuring the algorithm used to find available time windows is fast enough to evaluate all possible options in real time. The system must quickly assess operational constraints, such as vehicle capacity and existing schedules, to present the customer with feasible time windows without delays.

Once the customer selects a time window, it may sometimes result in less efficient delivery routes. For example, the customer's location during their chosen time slot might require the delivery vehicle to take a longer or less direct route, increasing travel time or costs. However, the provider is still committed to delivering within the agreed time frame, even if it affects route efficiency. This demonstrates the challenge of balancing customer choice with operational effectiveness. To handle these situations, advanced tools like real-time routing systems and predictive analytics are used to reduce inefficiencies and maintain high service quality.

2.2.2.1 Dynamic Customer Acceptance

In AHD, once the customer's order is received (1), the providers must consider whether they are available to deliver it (2); this can be known as the acceptance process. The acceptance mechanism has two main approaches: static and dynamic. As described by Campbell and Savelsbergh (2005), static customer acceptance is based on accepting a fixed number of requests per available time window, while dynamic acceptance offers possible time windows if the order's delivery location can feasibly be inserted into the existing route plan.

One simple method of dynamic customer acceptance is **myopic approach**, which is commonly used by many online supermarkets. The myopic approach makes decisions based on the current state of the system, such as available vehicle capacity and existing routes, without accounting for future orders. This approach is straightforward and computationally inexpensive, however, it lacks mechanisms to anticipate how accepting a current order might limit the ability to accept more profitable or operationally efficient orders later. Therefore, to address these limitations, advanced dynamic customer acceptance methods, such as sampling-based approaches or predictive analytics, have been developed. These methods incorporate historical and real-time data to evaluate the potential impact of current decisions on future operations.

The work of Ehmke and Campbell (2014) showed that the time-dependent dynamic acceptance mechanism provides high profitability and service quality in urban areas. In their study, three advanced variants of dynamic acceptance mechanisms by considering time-dependent, including DYN, DYN-SBF, and DYN-BUF, were implemented. **DYN** determines whether to accept an order by verifying that the number of required vehicles does not exceed the limit provided when adding that order. **DYN-SBF** is an extension of DYN that adds the same buffer time for each order to reduce the likelihood of lateness. **DYN-BUF** utilizes time-dependent and stochastic travel time data to decline requests that could result in tour plan lateness. For urban areas with 10-minute service time, the number of accepted requests with static acceptance is **103.8** and **105.7** for SLOT and TD-SLOT methods. DYN results with **116.4** requests, **92%** of tour plans running late, however, the number of lateness is relatively small (**3.1** lateness for 95th lateness percentile). DYN-SBF, with a 5-minute buffer, only **75%** of the number of the DYN is accepted, leading to less lateness (**1** lateness for 95th lateness percentile).

In the work of Köhler et al. (2020), authors outline various methods for customer acceptance in attended home delivery in real time based on request's impact on the current route plan and potential future requests. The work implemented three sampling-based techniques, which use historical order data to simulate potential future bookings and evaluate whether adding the current request is reasonable. **Instance-based sampling** uses entire historical booking days as samples, effectively preserving temporal patterns but limiting adaptability to new demand trends and relying heavily on recent data for accuracy. **Order-based sampling** combines individual orders from multiple booking days, offering flexibility and adaptability but potentially losing key temporal patterns essential for accurate future predictions. **Distribution-based sampling** combines historical order features based on their distributions, effectively capturing customer behavior patterns but being computationally intensive. In general, instance-based sampling results in a revenue increase of at least **5%** and up to **28%** compared to the myopic strategy. This research concludes that dynamic methods face trade-offs between speed and accuracy, as real-time evaluation of multiple samples increases computational demands. Additionally, data recency plays a crucial role as using shorter time frames is more effective, while older data may not accurately represent current customer behavior.

Overall, dynamic customer acceptance enhances efficiency in real-time order processing but comes with challenges in maintaining quick decision-making without compromising accuracy. Achieving this balance requires fast, reliable algorithms supported by high-quality data and accurate risk prediction to evaluate costs and feasibility within tight time constraints.

In the context of this thesis and the provided dataset, the focus is limited to implementing the **myopic approach** for customer acceptance. This approach evaluates orders based on their immediate feasibility and impact without incorporating forward-looking strategies or dynamic sampling methods.

2.2.2.2 Time window setting

As highlighted by Burian et al. (2023), the design of attended home delivery services is heavily influenced by the set of time windows offered to customers. Many studies have shown a trade-off between customer service and routing efficiency when offering shorter time windows. Campbell and Savelsbergh (2005) demonstrate an **18%** profit loss when time windows are reduced from 3 hours to 30 minutes in densely populated areas. Ehmke and Campbell (2014) find that reducing time windows from 2 hours to 30 minutes results in a **17%** decrease in the number of accepted requests. This indicates that while shorter time windows improve customer service, they significantly reduce routing efficiency and overall profitability. This fundamental trade-off between customer and business preferences necessitates an improved approach to time-window management and vehicle routing. Despite its operational inefficiency, static time window approach is simple and computationally inexpensive, making them widely used in traditional delivery systems.

Some recent works have found a flexible solution for the time window length to enhance the trade-offs. Namely, Köhler et al. (2020) offered customers different lengths of time window based on the arrival time. The Long-Short (LS) method was implemented with good results in that study. The LS offers long (standard) time windows for early-arriving customers and

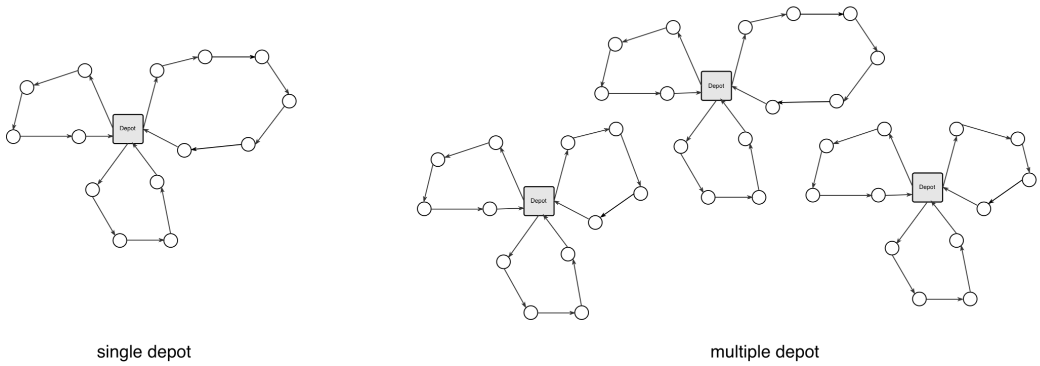

Figure 2.1: Comparison between VRP and MDVRP

only switches to short ones when a large portion of the route is defined. When comparing LS to the simple technique, LS accepts **81.5** customers, while the simple method, which offers short time windows to all customers, only accepts **60.8** customers. This represents an increase of **34%** in revenues.

Strauss et al. (2021), Yildiz and Savelsbergh (2020), and Lang (2021) focused on dynamic demand management strategies for time window optimization. By leveraging dynamic pricing approaches, they proposed offering delivery flexibility incentives and bundling time windows based on customers' receiving behaviors. These strategies aimed to improve customer satisfaction and provide greater choice while maintaining operational efficiency by guiding customer selections toward cost-effective options.

In this study, we aim to enhance the routing solution by addressing the multi-depot vehicle routing problem with fixed time windows (MDVRPTW). The flexibility of time-window settings is beyond the scope of this research.

2.3 Multi-depot Vehicle Routing Problem with Time Window

As described above, the process of Attended Home Deliveries comprises five primary steps, and this thesis assesses the feasibility of request (2) by solving the Multi-depot Vehicle Routing Problem with Time Windows (MDVRPTW).

The MDVRP is generally well-known as a complex VRP variant known as an NP-hard problem, where several different depots are considered instead of one. Implementing a multi-depot is more feasible when a single depot is overwhelmed with orders and an additional order is received. The variant of MDVRP, MDVRPTW, involves customer demands that must be satisfied within a specific time window.

Highlighted below are some recent advancements in the study of MDVRP. Li (2018) explored the optimization of MDVRP by leveraging shared depot resources. They employed a hybrid genetic algorithm with adaptive local search to maximize potential benefits and demonstrated the advantages of a shared-depot setting. Salhi (2014) proposed a variable neighborhood search approach to solve the multi-depot vehicle routing problem considering heterogeneous vehicle fleet. The experiments demonstrated that the Variable Neighborhood Search (VNS) method is effective for solving the MDVRP. Mir Ehsan Hesam Sadati (2021) formulated the MDVRP as a Mixed Integer Linear Programming (MILP) model and design a hybrid algorithm that integrates General Variable Neighborhood Search (GVNS) with Tabu Search. This research provided high quality solutions in short computation times. Building on these advancements, Sadati et al. (2021) proposed another version of VNS with tabu shaking to address complex multi-depot scenarios efficiently. The study of Sadati et al. successfully achieved six new best-known solutions in the MDVRP.

The Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is one of the most significant extensions of the classic MDVRP. This variant introduces time window constraints, which are essential for ensuring on-time deliveries in real-world applications such as logistics and e-commerce. There have been some projects on the MDVRPTW, namely the work of Dondo and Cerdá and Xu et al.. Dondo and Cerdá (2007) presents a three-phase algorithm using cluster-based method to solve multi-depot and heterogonous VRP. Polacek et al. (2004) proposed Variable Neighborhood Search methods (CROSS-exchange and iCROSS-exchange) for MDVRPTW in the shaking step and obtained the final optimal solution by a 3-opt local search. Xu et al. (2012) used a hybrid method of insertion and exchange for a shaking process, and then applied a local search algorithm, VNS, to find the best solution. Tie Jun Wang (2012) aplied particle swarm optimization to tackle the MDVRPTW. Wang (2023) presented an advanced two-stage hybrid algorithm of 3D k-harmonic means clustering algorithm and genetic algorithm to solve MDVRPTW. The hybrid strategy was proved to be an effective and successful approach for MDVRPTW.

Most methods for solving the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) use up to three phases to systematically address its complexity and constraints. These phases often include clustering or depot assignment, route construction, and route optimization. By dividing the problem into manageable steps, these methods ensure a more structured and efficient approach to finding feasible and high-quality solutions.

A particularly effective strategy for tackling the MDVRPTW involves the combination of different types of heuristics, leveraging their complementary strengths. Hybrid approaches, which integrate multiple heuristics or metaheuristics, have proven to be especially powerful for solving large-scale and complex VRP. For instance, combining local search with population-based algorithms or enhancing traditional heuristics with adaptive mechanisms often results in better performance compared to standalone methods.

This section reviews three key types of heuristics commonly applied to Vehicle Routing Problems: **construction heuristics, local search heuristics, and metaheuristic algorithms**. Each type offers unique methodologies and practical applications:

1. Construction Heuristics: These methods focus on building initial solutions in a straightforward manner, such as the Clarke-Wright Savings Algorithm or nearest-neighbor approaches. While these solutions may not be optimal, they provide a strong starting point for further optimization.

2. Local Search Heuristics: These techniques improve existing solutions by exploring their neighborhood, making incremental changes to reduce costs or improve feasibility. Common methods include 2-opt, 3-opt, and CROSS exchanges, which are particularly effective for refining routes in MDVRPTW.

3. Metaheuristic Algorithms: These advanced techniques, such as Genetic Algorithms (GA), Tabu Search (TS), and Simulated Annealing (SA), are designed to explore the solution space more extensively. They often incorporate mechanisms to escape local optima and handle complex constraints like time windows and depot allocations.

By understanding and applying these heuristic approaches, researchers and practitioners can tackle the intricate challenges of MDVRPTW with greater efficiency and effectiveness. This section reviews three types of heuristics commonly used to address Vehicle Routing Problems (VRP), providing insights into their methodologies and applications in tackling complex routing challenges.

2.3.1 Constructive Heuristics

Constructive heuristics offer a quick and easy-to-implement method. These heuristics build solutions step-by-step, ensuring computational efficiency and feasibility. The constructive heuristics start with an empty solution and continuously add components until a complete solution is obtained.

To construct an initial solution for VRP, the savings heuristic of Clarke and Wright (1964) is one of the best-known algorithms for its efficiency and simplicity. The fundamental concept of this algorithm is to allocate a distinct path for each client node initially, then minimize the overall cost by merging the routes.

Besides Clarke-Wright savings method, many researchers implement Solomon's sequential heuristics (Solomon, 1987) for VRPTW. A heuristic by Solomon, time-oriented nearest-neighbor, starts every route by the customer closest to the depot. At each iteration, select the next customer to visit based on the shortest travel time, adjusted for time window constraints and the vehicle's current position. This selection considers both the distance and the time required to service the customer within their specified time window.

The most successful insertion heuristic of Solomon (1987) is the I1 insertion heuristic for VRPTW. It constructs routes incrementally by adding customers in each iteration into the best possible position that minimizes the increase in travel distance and adheres to time window constraints. The heuristic evaluates all feasible insertion points for each customer and selects the one with the least cost increase, ensuring efficient and feasible route construction. This method is widely recognized for its effectiveness in handling time-sensitive delivery and pickup scenarios. Other well-known constructive heuristics are sweep algorithm by Gillett and Miller (1974) and insertion heuristic by Christofides and Eilon (1972).

2.3.2 Local Search

Local search, also known as improvement heuristic, is a heuristic method for approaching computationally difficult optimization problems. It iteratively explores the solution space by moving from one solution to a neighboring solution in search of an improved solution. The local search algorithm starts with an initial solution and iteratively moves the current feasible solution to a neighboring solution until the local optimal solution is found. In the context of the Vehicle Routing Problem (VRP), common neighborhood structures are **(λ, λ)-exchange** and **k-opt.**

(λ, λ)-exchange

This method selects two segments of up to λ nodes and switches their locations. Two most important types in this method are the $(1, 0)$-exchange, known as *"relocation"*, and the $(1, 1)$-exchange, known as *"swap"* with the complexity of $O(n)$.

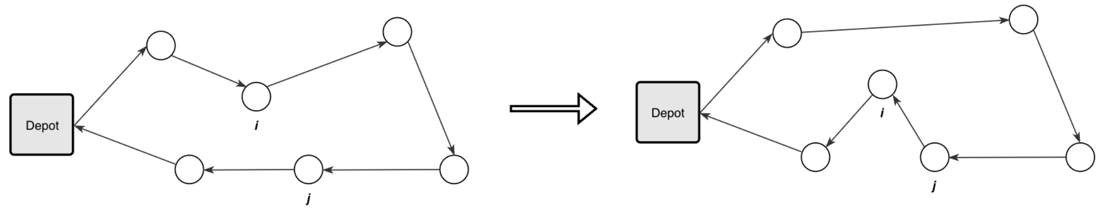

Figure 2.2: (1,0)-exchange or Relocation for single route

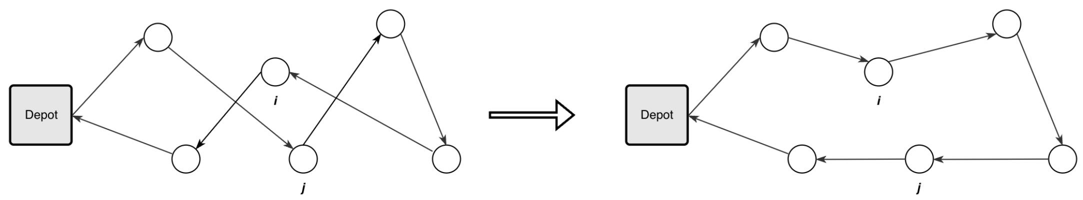

Figure 2.3: (1,1)-exchange or Swap for single route

In TSP with multiple routes, this method can switch nodes on different routes. However, in VRP, all nodes should remain in the same route for vehicle restrictions (time-window, capacity).

k-opt

This method removes k arcs from the solution and adds another k arcs to reconnect the route. Two most popular types of this method are 2-opt and 3-opt. Croes (1958) first

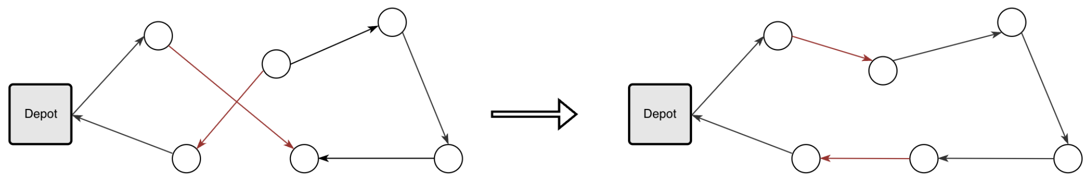

Figure 2.4: 2-opt algorithm for single route

introduced 2-opt algorithm for solving TSP. In the same year, F. Bock presented his 3-opt at 14th ORSA National meeting and the manuscript is unpublished. Lin (1965) defined k-opt algorithm and presented an efficient way to implement 3-opt for TSP.

In the work of Christofides and Eilon (1972), they presented a performance comparison among different values of k. It is stated that 3-opt has a major improvement in comparison with 2-opt while 4-opt only has a small improvement in comparison with 3-opt.

Some studies use 3-opt as the final step which leverages its powerful local search capabilities to refine and improve solutions obtained from metaheuristics, leading to better overall results in solving the VRP. This approach combines the global search strength of metaheuristics with the local optimization power of 3-opt, creating a robust and efficient solution process. Namely, the work of Polacek et al. (2004) in MDVRPTW obtained the final solution by 3-opt after using the VNS method to find a good solution. In the same year, Prins (2004) presented a hybrid version of Genetic Algorithm for the VRP with the mutation step using 3-opt.

2.3.3 Meta-heuristics

Using only constructive or local search heuristics to find optimal solution is sometimes inefficient since they can be trapped in local optima and miss better solutions. Metaheuristics, on the other hand, use some strategies to extend solution space, which enables them to escape from local optima and thus hopefully reach a better solution.

Simulated Annealing (SA)

Simulated Annealing (SA) is an optimization method inspired by the process of cooling metals. It searches for the best solution by exploring different possibilities and sometimes accepting worse solutions to avoid getting stuck in local optimal results. SA is simple, flexible, and widely used for solving complex problems in many areas. Wang (2013) proposed a Simulated Annealing (SA) approach combined with an integer programming model to address the Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows (VRPSPDTW). The performance of SA was compared with results obtained using a Genetic Algorithm (GA), demonstrating that SA is an effective metaheuristic for solving VRPSPDTW. Baños (2013) developed a solution for the Vehicle Routing Problem with Time Windows (VRPTW) by employing Parallel Simulated Annealing on a single workstation. The method demonstrated the effectiveness of parallel processing in solving complex VRPTW instances efficiently. Yassen (2017) proposed a Simulated Annealing setup embedded within a harmony search algorithm to solve the VRPTW. The hybrid approach leveraged the strengths of Simulated Annealing and achieved excellent results when compared to Solomon's benchmark for the vehicle routing problem with time windows. Simulated Annealing (SA) is proved to be highly flexible, applicable to combinatorial, continuous, and mixed-integer problems, and works well on irregular or non-differentiable landscapes. However, this flexibility often requires many iterations, especially at high temperatures, leading to significant computational effort, particularly for large-scale problems.

Tabu Search (TS)

Cordeau et al. (1997) first introduced a heuristic method - Tabu Search (TS) to solve VRP. The idea of TS is to fully scan a neighborhood of the current solution and make the best possible move, even if it deteriorates the objective function. A tabu list tracks recent moves to prevent cycling, while aspiration criteria allow overriding restrictions if a move significantly improves the solution. TS approaches have been widely used, and many successful versions of TS have been developed for different variants of VRP, including CVRP (Barbarosoglu and Ozgur (1999)), HFVRP (Leung et al. (2011)), and MDVRPTW (Cordeau et al. (2001)). The work of Cordeau et al. (2001) on Tabu Search to solve MDVRPTW concludes that the solution TS does not always provide the best solution, however, the speed of execution and memory usage are its vast advantages. Cordeau (2004) continued applying Tabu Search to various VRP variants, including MDVRP, PVRPTW, and MDVRPTW. He used forward time slack to reduce time window violations when accepting infeasible solutions. While forward time slack added computational time, the study showed a dramatic improvement in solution quality, reducing the average cost. Pureza (2012), Ma (2012), and Zhang (2017) use Tabu Search for VRPTW while considering another constraint. The study by Pureza addressed the inclusion of additional delivery personnel when necessary and solved the VRPTW using Tabu Search and Ant Colony Optimization heuristics to obtain minimum-cost routes. Ma developed a Tabu Search heuristic with an adaptive penalty mechanism to address new restrictions on road links related to vehicle passing tonnage. Zhang proposed a hybrid approach combining Tabu Search and the Artificial Bee Colony algorithm to address pallet loading constraints. The study implemented a reassignment strategy, including node swapping and relocation, as well as route-level operations such as route swapping and partial route reversal. This approach was proven to effectively improve the pure Tabu method. Gmira (2021) proposed a method for solving the time-dependent vehicle routing problem with time windows on a road network. The initial solution is generated using a greedy insertion heuristic, while the neighborhood of the current solution is explored using CROSS exchanges combined with Tabu Search.

Genetic Algorithm (GA)

Genetic Algorithm (GA), a well-known algorithm inspired by the biological evolution process, was first proposed by Holland (1992). The GA was first implemented in solving VRPTW by Thangiah (1995), called GIDEON. Thangiah (1998) extended this method by developing a successful hybrid GA for the VRPTW by using Simulated Annealing and Tabu Search methods to improve the solution, demonstrating the potential of hybrid techniques in addressing complex VRPTW challenges.. Berger et al. (2003) proposed a competitive and cost-effective GA for VRPTW, which considers two distinct populations of solutions: one for optimizing total travel distance and another for minimizing the time window violation. Maroof et al. (2013) proposed a cutting-edge Hybrid Genetic Algorithm combined with the Solomon Insertion Heuristic (HGA-SIH) to identify the most efficient routes for vehicle routing problems. Guo et al. (2017) presented a solution to the Vehicle Routing Problem for Battery Electric Vehicles with time window constraints, considering the limited travel distance imposed by battery capacity. The method employs a Genetic Algorithm (GA) enhanced with a penalty function to address lateness, ensuring efficient route planning while respecting time and battery limitations. The popularity of hybrid approaches in GA stems from their flexible structure, which allows for problem-specific adjustments at various stages of the algorithm, such as selection, crossover, and mutation. This adaptability makes GA highly suitable for combining with other optimization methods, like local search, heuristics, and metaheuristics, to enhance performance and better address complex constraints in VRP. For example, combining GA with specialized operators or local search strategies can improve solution diversity and convergence rates, making them more effective for solving large-scale problems. Further recent genetic operators modified for usage with the VRP were reviewed in the work of Ochelska-Mierzejewska et al. (2021), showcasing their enhanced effectiveness in handling the intricacies of VRP variants.

Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO), a metaheuristic algorithm inspired by the behavior of ants to find optimal paths between food sources and their nest, was first proposed by Dorigo et al. (1999). ACO was applied to solve CVRP by Skok et al. (2000), Reimann et al. (2004) , and Yu et al. (2009) and was proven to be efficient and effective in solving VRP. Yu et al. (2009) integrated ACO with a mutation operation from Genetic Algorithms to expand the solution space. The study also mentioned two approaches for handling capacity violations: imposing a high penalty cost on infeasible solutions and resolving capacity constraints by adjusting delivery amounts. While the running time of this ACO version is not ideal, the solution quality achieved is significantly high, making it a strong candidate for solving complex routing problems where accuracy is prioritized over speed.

ACO has also been successfully applied to solve MDVRPs, such as the work of Yu et al. (2011) with a parallel improved ACO (PIACO) for MDVRP. The results reveal that the PIACO used in this research can find the best-known solutions in 15 problems among the 23 problems in comparison to several algorithms. Yongbo Li (2019) introduced an improved Ant Colony Optimization (IACO) algorithm for the multi-depot green vehicle routing problem. The results show that IACO delivers satisfying performance, achieving higher solution quality compared to conventional ACO, making it a more effective approach for optimizing environmentally friendly vehicle routing when considering multi depots.

To the best of my knowledge, although many methods have been developed for the VRPTW, there is still a lot of potential for research in the field of MDVRPTW. The integration of techniques for solving MDVRPTW with varying numbers of depots and time-window lengths to optimize the decision-making process in the AHD problem has yet to be thoroughly investigated. This sparks the idea of interpreting how the number of depots affects customer acceptance rate and fulfillment cost in the AHD problem.

Chapter 3

Problem Formulation

The solution to the Multi-Depot Attended Home Delivery (AHD) problem with Dynamic Customer Acceptance involves two primary steps:

1. **Feasibility Check:** This step determines the feasibility of fulfilling a customer's order by solving the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). If no feasible time slot is available, the order is declined. Conversely, if at least one time window is feasible, the process advances to the next step.
2. **Selection Procedure:** In this step, all available time slots are presented to the customer. Based on the customer's preference, either one time slot is selected or the offer is declined.

The work of this thesis is considered an operational solution, and it aims to:

- maximize the acceptance rate
- analyze the effect of the number of depots and time window length on the acceptance rate and fulfillment cost

3.1 Notation

Sets

I : set of delivery locations (customers)

D : set of depots
V_d : set of delivery vehicles at depot d
N : set of all nodes, $N = I \cup D$

Parameters

CV_k : fixed cost for vehicle k
CT_k : transportation cost for vehicle k per time unit
CR_k : labor cost for delivery person for vehicle k per time unit
t_{ij} : transportation time between node i and j
st_i : service time at customer i
e_i : earliest time at customer i (lower-bound of the time window)
l_i : latest time at customer i (upper-bound of the time window)

Variables

$$X_{ijk} = \begin{cases} 1 & \text{vehicle } k \text{ travels from customer } i \text{ to customer } j \\ 0 & \text{otherwise} \end{cases}$$

T_i : vehicle arrival time at customer i

T_k : total time used by vehicle k

3.2 Mathematical Model for MDVRPTW

The process begins when a customer places an order. The provider then offers possible time windows based on its feasibility by solving the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW).

The MDVRPTW can be formulated as follows: there are a number of depots with a limited number of vehicles. The vehicle's capacity is not considered in this problem. The elements in the set of customers and depots are considered nodes in a directed graph. Non-negative values of travel distance and travel duration to reach node j from node i are defined in each $arc(i, j)$.

The thesis considers the dynamic approach: based on the combined set of current routes and the new request, r_{new}, an actual MDVRPTW solution is computed and checked for feasibility. If there is a feasible solution, r_{new} is accepted; otherwise, it is rejected (Ehmke and Campbell, 2014).

In the version of the problem we study, vehicles do not necessarily leave their depots at time 0: due to time windows, it may be convenient to delay the departure time to reduce waiting time at customer locations. Besides, deliverers are allowed to wait at a location before its time window starts.

The objective is to maximize the acceptance rate and satisfy the following restrictions:

1. Each customer is visited once by only one vehicle
2. Customers must be served within the ordered time window
3. Each tour must begin and end at the same depot
4. The number of available vehicles must not be exceeded

My framework assumes that:

1. Each depot's position and number of vehicles are pre-defined and unchanged
2. The vehicle's capacity limit is not considered
3. The customer's order cannot be changed or canceled

$$\sum_{d \in D} \sum_{k \in V_d} \sum_{j \neq i \in N} X_{ijk} = 1 \quad \forall i \in N \tag{1}$$

$$e_i \leq T_i \leq l_i \quad \forall i \in I \tag{2}$$

$$\sum_{j \in I} X_{djk} = 1 \quad \forall d \in D, \forall k \in V_d \tag{3}$$

$$\sum_{i \in I} X_{idk} = 1 \quad \forall d \in D, \forall k \in V_d \tag{4}$$

$$\sum_{j \in N} X_{ijk} - \sum_{j \in N} X_{jik} = 0 \quad \forall i \in I, \forall k \in V_d \tag{5}$$

$$\sum_{k \in V_d} \sum_{j \in I} X_{djk} \leq |V_d| \quad \forall d \in D \tag{6}$$

Constraint (1) ensures that each customer is visited once by only one vehicle. Constraint (2) ensures that the customer is serviced during the time window. Constraints (3) and (4) ensure each vehicle starts and ends at the same depot. Constraint (5) is flow conservation to ensure that the vehicle arrives at a customer node, it must also leave that customer node. Constraint (6) ensures the number of vehicles departing from depot d does not exceed the number of available vehicles at that depot.

For each incoming request, the algorithm systematically evaluates all available time windows to determine the best placement for the request in the tour plan. The number of times the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is solved corresponds to the total number of time windows under consideration for that request. For each time window, the algorithm identifies the most suitable position within the existing tour plan that minimizes additional cost or disruption while satisfying all constraints, such as time windows and vehicle capacity.

Once the nearest-optimal position is identified for each time window, the algorithm compiles a list of all feasible time windows and presents them to the customer. This allows the customer to choose their preferred time window based on convenience or specific requirements. After the customer makes their selection, the request is finalized and integrated into the tour plan at the designated position, ensuring it aligns with all constraints and does not negatively impact the overall efficiency of the routes. This iterative process ensures that each request is handled optimally while maintaining flexibility and customer satisfaction. By dynamically updating the tour plan with each new request, the algorithm achieves a balance between operational efficiency and customer preference, resulting in a robust and adaptable solution for solving the MDVRPTW.

3.3 Selection Procedure

Once we offer all available time windows, the customer will decide the final delivery time. As the dataset does not include the customer's time preference, the customer's decision will be selected based on an approximate method. This study considers the *unequal demand* approach mentioned in the work of Köhler et al. (2020). Köhler et al. (2020) divides the customers into 2 groups: customers who only accept the time windows in the first half of the day and customers who only accept the time windows in the second half of that day. Regardless of the time window's length, customers will randomly choose a provided time window based on their group.

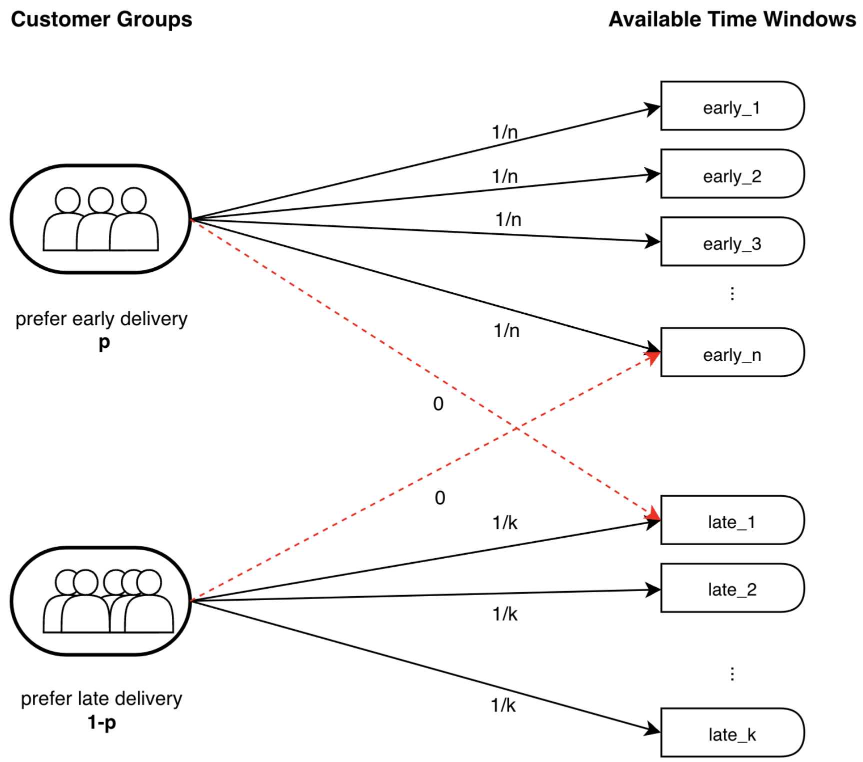

Figure 3.1: Customer's selection procedure considering unequal demand

For instance, for those who prefer early delivery, there are n available early time windows, $early_1, early_2, ..., early_n$, the probability of choosing one of the early spots is $\frac{1}{n}$. There might be some available late time windows, however, the probability of choosing them in this case is 0. There is a possibility that a customer who prefers early delivery may deny all offers and cancel the request if all available time windows are in the late timeframe and reverse.

3.4 Evaluation Procedure

As mentioned above, the primary focus of this work is to analyze the impact of varying the number of depots and the duration of time windows on both customer acceptance rates and fulfillment cost.

- Acceptance rate (AR): the proportion of delivery requests or orders that are accepted and successfully integrated into the routing plan.

$$AR = \frac{\text{number of successful requests}}{\text{total number of requests}}$$

- Fulfillment cost (FC): refers to the total expenses to deliver a product or service to a customer. In the context of grocery home delivery, the fulfillment cost includes costs in storage, order processing, delivery, and customer service. However, in the context of this thesis, fulfillment cost are simplified by only considering delivery cost. The cost is defined by the following function:

$$FC = \sum_{k \in V_d} (CV_k + CT_k \cdot T_k + CR_k \cdot T_k)$$

Chapter 4

Algorithm Implementation

This chapter presents a detailed method of the integrated algorithmic approach to solve the Multi-Depot Attended Home Delivery (AHD) problem, particularly in the context of dynamic customer acceptance. The methodology combines three popular algorithms: 2-opt, beam search heuristics, and simulated annealing. This integrated approach aims to optimize delivery routes while maximizing the number of accepted requests.

4.1 Solving Multi-depot Vehicle Routing Problem with Time Window

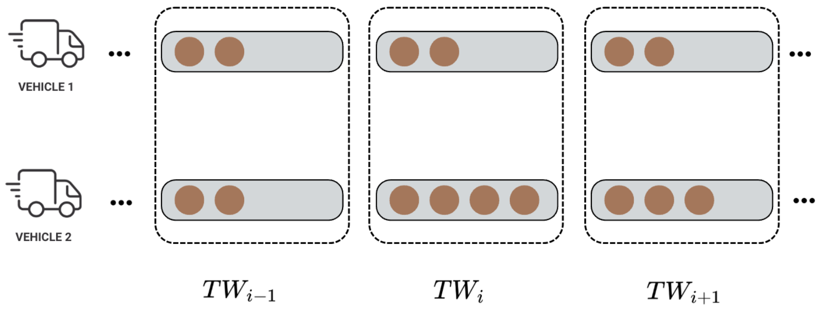

Figure 4.1: Multi-depot Vehicle Routing Problem with Time Window

4.1.1 Beam Search

Beam Search heuristics

Beam search is an advanced algorithm used in generate sequences. Beam search maintains multiple hypotheses to find the best overall sequence.

Akeb et al. (2013) introduced an approximate algorithm based on beam search to address the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). This algorithm operates in three phases: clustering, beam search, and 2-opt local search. The study highlights that beam search explores multiple paths concurrently, thereby enhancing the likelihood of identifying optimal solutions. Shati (2024) utilized Neural Sequence Generation to address variants of the VRP by employing beam search. Their experiments demonstrated that beam search effectively satisfies stringent requirements with minimal impact on solution quality.

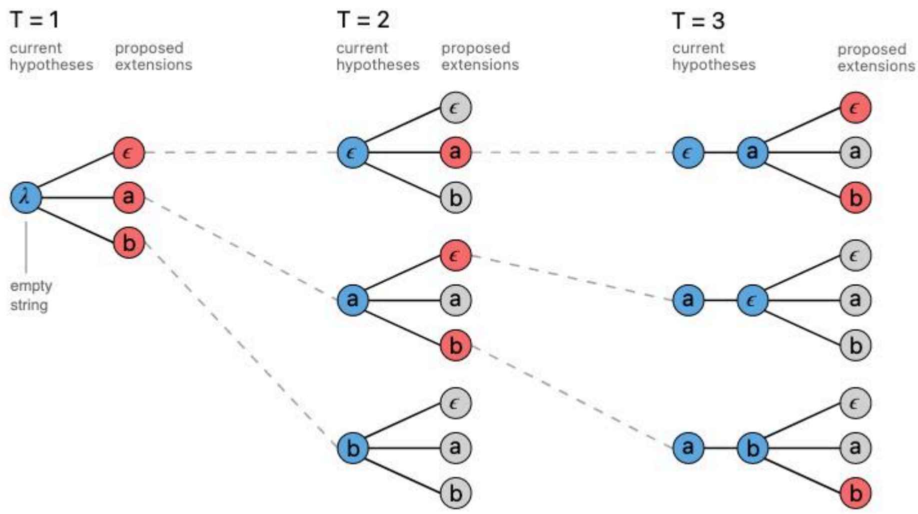

Figure 4.2: Beam Search algorithm (Source: https://fancyerii.github.io/books/ctc/)

Since the MDVRPTW needs to be solved multiple times to identify all available time windows, it is crucial that the algorithm is both fast and efficient. Additionally, in the context of this thesis, the supplier must promptly provide customers with available delivery time frames, further emphasizing the need for a quick and reliable solution. Beam search is well-suited to address these requirements. Its ability to efficiently explore the solution space while reducing computational time.

Since the time window constraints are strict, the algorithm must avoid excessive shuffling of routes or schedules, as this can lead to violations of other requests' time windows. Furthermore, given the limited execution time and the impossibility of modifying or canceling any accepted requests, beam search's deterministic pruning and ranking mechanism provide a structured and controlled exploration process.

Algorithm: Once a new request with location information pops up, it is added to TW_i of each vehicle's tour.

1. The new request r_{new} is added to TW_i
2. Run 2-opt for the **last node** in TW_0 (depot) and P random permutations of nodes in TW_1
3. k **unique solutions** with the **smallest duration** are remained for the next iteration despite the infeasibility
4. For each remaining solution from TW_1, the algorithm continues running 2-opt for its last node and P random permutations of nodes in TW_2
5. The process repeats for TW_3, TW_4, ..., TW_n

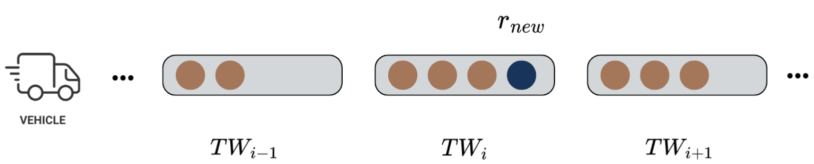

Figure 4.3: The new request r_{new} is added to TW_i

To evaluate the feasibility of a new request r_{new} within time window i, the request is added to TW_i, and our algorithm is applied to determine whether at least one feasible solution exists for the tour plan after the inclusion of the new request.

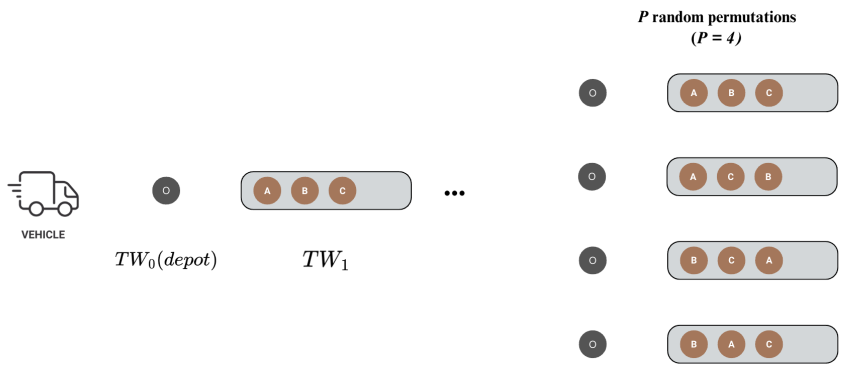

Figure 4.4: Depot and P random permutations of TW_1 nodes

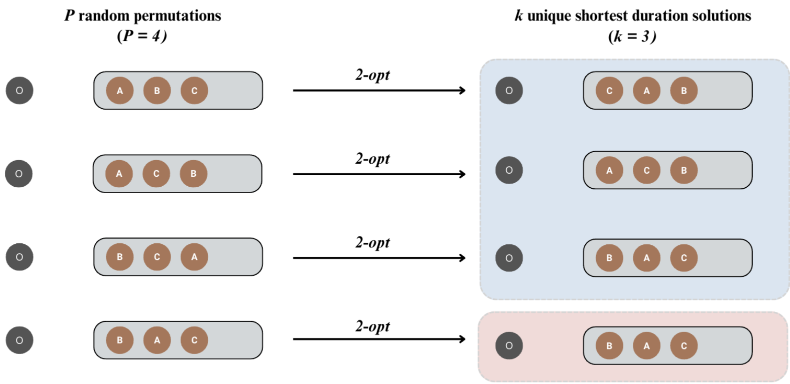

Figure 4.5: Remain k solutions with smallest duration

The local search algorithm 2-opt is implemented to optimize the last node of the previous time window and P permutations of the current time window. For the initial iteration, the current time window is TW_1, and the last node of the previous time window (TW_0) is assigned by default as the corresponding depot. Instead of considering all possible permutations of the current time window, a limited number of P permutations are selected to reduce computational time and make the approach more practical and cost-effective in real-world applications. This selective approach balances optimization quality with operational feasibility.

The goal of treating each time window separately and applying beam search to the end node of each time window, as well as to all nodes of the subsequent time window, is to prevent the reassignment of accepted nodes to a different time window, which is not permitted. Additionally, the challenge lies in determining the connecting nodes between two time windows. To optimize the overall distance, it is crucial to identify a suitable connecting point that ensures seamless transitions while maintaining efficiency.

In some cases, the results from 2-opt algorithm of some permutations are similar. We remain k best unique solutions. If there are less than k unique solutions, all the solutions are remained for the next iteration. After applying beam search from TW_0 to TW_n for each tour plan of each vehicle, we obtain the best sampled tour plans for each vehicle. However, during this process, the new request r_{new} is added to each vehicle's tour without strictly enforcing feasibility. This means that r_{new} or some other nodes in the plan may be infeasible.

In each iteration, we obtain at most kP tour plan candidates. For each time window, after solving the MDVRP for $|V_d|$ times, where $|V_d|$ represents the total number of vehicles, the new request r_{new} is added to the tour plan of the vehicle that results in **the smallest change in total duration** compared to its corresponding previous tour plan. This approach ensures minimal disruption to the existing plans while accommodating the new request efficiently.

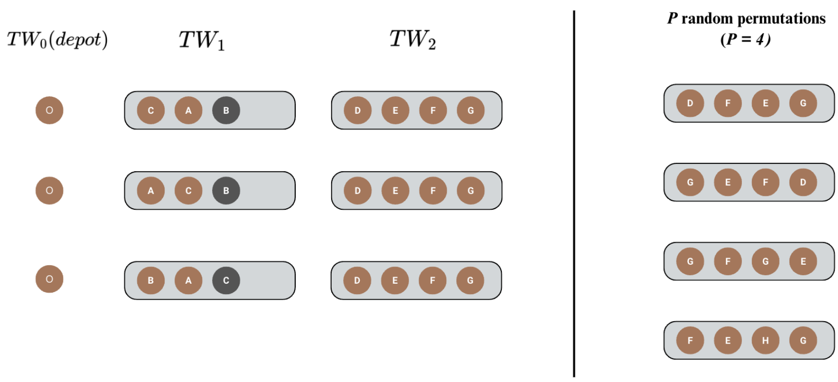

Figure 4.6: The process repeats for the next iteration of TW_2 (1)

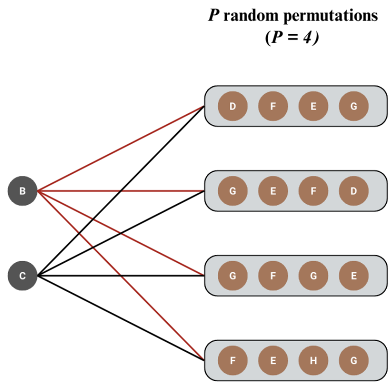

Figure 4.7: The process repeats for the next iteration of TW_2 (2)

The more nodes a vehicle tour plan contains, the longer the calculation time and the higher the probability of infeasibility. Therefore, the move and swap method described in the next section can potentially reduce infeasibility while achieving an optimal solution. This method allows for more efficient use of vehicle tours, ensuring they are utilized effectively instead of being underused due to inefficient tour plans

4.1.2 Shaking Step (Move & Swap)

The shaking step, which involves techniques such as swap, move, and cross, is commonly used in VRPTW and Multi-Depot VRP (MDVRP) due to its effectiveness in improving solution quality and overcoming optimization challenges. These methods introduce controlled perturbations to the solution by altering routes, such as swapping nodes between routes, moving customers to different vehicles, or crossing route segments. This helps escape local optima and explore new areas of the solution space, which is essential for avoiding premature convergence on suboptimal solutions. Additionally, shaking steps improve the overall efficiency of the solution by redistributing requests more effectively across vehicles, balancing loads, and meeting time window constraints. They also address inefficiencies like underutilized routes or excessive travel distances, enabling the optimization process to produce better-balanced and more feasible solutions. This approach was proved to be considerable effective in the work of Polacek et al. (2004), Xu et al. (2012) for MDVRPTW, Sadati et al. (2021) for MDVRP, and many others.

In the context of this thesis, as all nodes are not known at the time of acceptance, the previously generated solution may no longer be optimal when a new request arises, especially for the concept of multi-depot. To address this, the move-and-swap method is applied. In this approach, a **single node** is randomly selected from a vehicle's tour and either moved or swapped into the **corresponding** time window of another vehicle's tour. This method can be applied iteratively, with each iteration having an equal probability (50%) of a move or a swap.

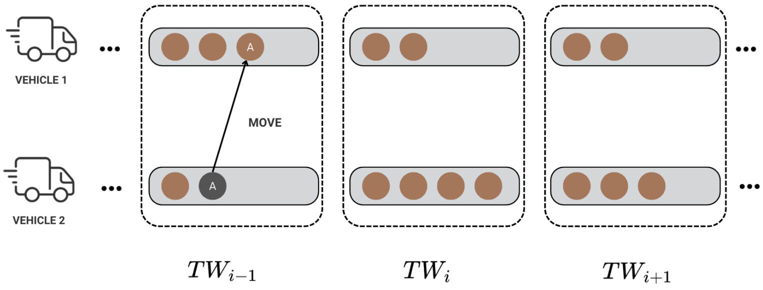

Figure 4.8: Move Method

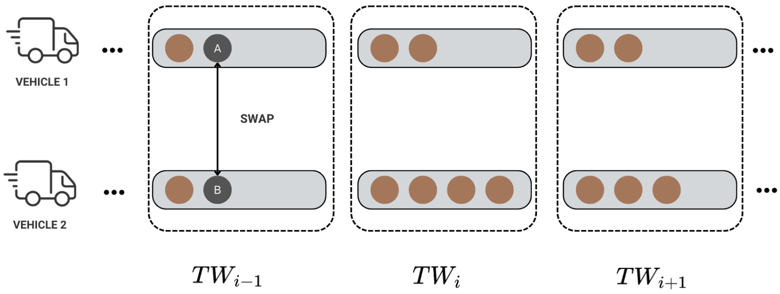

Figure 4.9: Swap Method

By **Greedy approach**, the solution with the smallest **cost function** is selected (the cost function is defined in 4.1.4).

4.1.3 Simulated Annealing

Simulated Annealing is a meta-heuristic algorithm designed to overcome the limitations of greedy approaches, which often become trapped in local optima due to their tendency only to accept improvements.

Simulated Annealing begins with an initial solution and seeks to iteratively improve it by introducing random perturbations. These perturbations are accepted based on a probabilistic criterion, which allows the algorithm to explore a broader solution space. The probability of accepting a worse solution is deliberately set high in the early stages, enabling exploration beyond local optima, and gradually decreases as the number of iterations increases, focusing on refinement. This balance between exploration and exploitation makes Simulated Annealing an effective strategy for finding near-optimal solutions in complex optimization problems.

SA is particularly suitable for large-scale problems with many nodes, vehicles, and depots. Additionally, SA can be customized to include domain-specific constraints like time windows and depot assignments through penalty terms or feasibility checks.

The classic acceptance criterion of SA is based on the Boltzmann probability distribution.

$$p = \begin{cases} 1 & \text{if } C_{i+1} \leq C_i \\ -\exp\left(\frac{C_{i+1}-C_i}{kT}\right) & \text{if } C_{i+1} > C_i \end{cases}$$

where C is the cost value, k is the Boltzmann constant, and T is the initial *temp*.

4.1.4 Cost Function

In the framework of the thesis, the two main factors we need to consider are acceptance rate (AR) and fulfillment cost (FC) described in 3.4.

In the context of AHD with dynamic customer acceptance, we ensure that no confirmed request is delayed or cancelled, giving AR priority over FC.

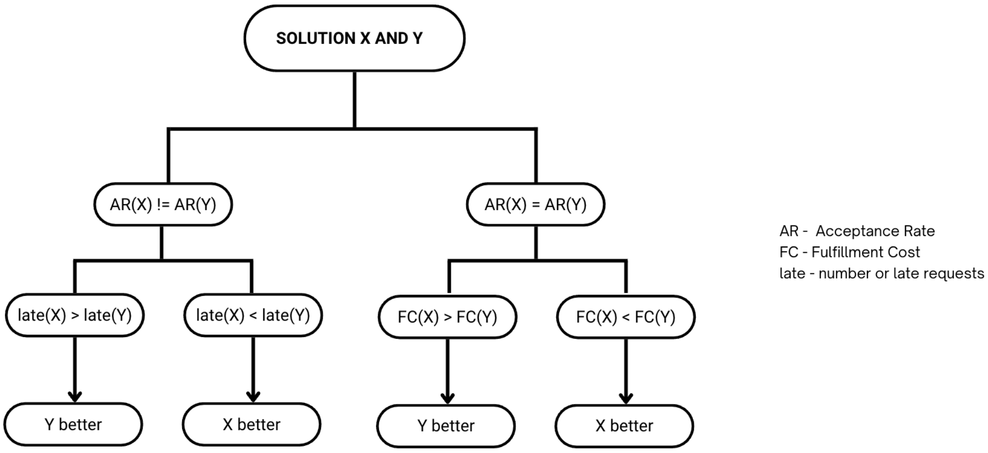

Figure 4.10: Comparison of Two Routing Solutions

This diagram outlines a decision-making process for comparing two solutions, X and Y, based on three key metrics: **Acceptance Rate (AR)**, **Fulfillment Cost (FC)**, and the number of late requests (**late**). The process starts by evaluating the Acceptance Rate (AR). If $AR(X) \neq AR(Y)$, the number of late requests is then considered. If $late(X) > late(Y)$, Solution Y is deemed better, whereas if $late(X) < late(Y)$, Solution X is preferred. If the Acceptance Rates are equal $AR(X) = AR(Y)$, the comparison moves to Fulfillment Cost (FC). In this case, if $FC(X) > FC(Y)$, solution Y is better, and if $FC(X) < FC(Y)$, solution X is better. The metrics are prioritized in this order to ensure that Acceptance Rate, as the primary criterion, is evaluated first, followed by the number of late requests to measure time-related feasibility, and finally Fulfillment Cost to assess overall efficiency. This approach provides a structured and systematic way of comparing solutions to determine the optimal one.

In certain situations, even if accepting a new request requires starting a new tour, this approach is preferred over adding the request to an existing tour if it risks delaying other orders. To enforce this, a significant penalty w_{late} for late requests is set into the cost function. The cost function is defined as:

$$cost(X) = FC + late(X) * w_{late}$$

$$w_{late} \geq max(duration) * (CT_k + CR_k) + max(vehicles) * CV_k$$

where

CV_k	: fixed cost for vehicle k
CT_k	: transportation cost for vehicle k per time unit
CR_k	: labor cost for delivery person for vehicle k per time unit
w_{late}	: penalty for each late request
$max(duration)$	: maximum duration of a tour
$max(vehicles)$	: total number of vehicles

4.2 Selection Procedure

After solving the MDVRP for $|V_d|$ times for each time window, where $|V_d|$ represents the total number of vehicles, the new request r_{new} is added to the tour plan of a vehicle. However, it is important to note that not every time window can yield a feasible solution for accommodating r_{new}.

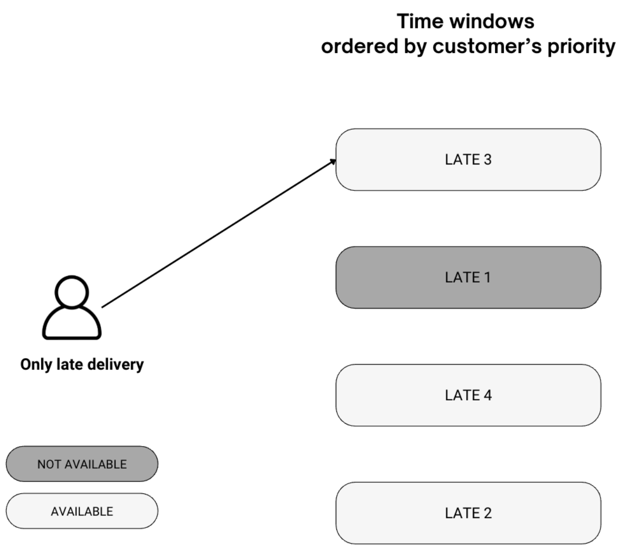

Figure 4.11: Customer Selection Using Unequal Demand Approach

Once the availability of each time window TW_i is checked, all the available time windows are offered to the customer. The selection strategy is followed by the unequal demand approach of Köhler et al. (2020) mentioned in 3.3.

In the framework of this thesis, an additional priority mechanism is incorporated to further refine the selection process. Each customer is assigned a priority list and categorized into one of two delivery preference groups: "early delivery" or "late delivery." These groups are determined based on predefined probabilities p and $1-p$, respectively.

To enhance the flexibility and personalization of the time window assignment, a priority list of time windows is generated for each customer through random permutation. This randomized approach ensures variability in customer preferences while considering the feasibility of the available time windows. Once the priority list is established, the final time window selection is made by identifying the highest-ranked available time window in the customer's list.

For example, in the figure above, the customer is classified as having a late delivery preference. As a result, only late time windows are included in their choice list. Among these, the time window $late_3$ is the top-ranked and available option. Therefore, the customer selects $late_3$.

Chapter 5

Experimental Results

5.1 Dataset

The dataset used in this thesis is Replication Data for Collaborative Transportation for Attended Home Delivery (OA edition) by Elting et al. (2024*b*). The dataset includes realistic instances of the problem by querying driving durations using the Open Source Routing Machine (OSRM) and OpenStreetMap data between random addresses in the city of Vienna.

The dataset includes 200 samples with different settings for a total number of customer requests (150 and 300) and the degree of service area overlap among depots (0, 0.25, 0.50, 0.75, and 1). The number of depots in each sample is 3.

This thesis focuses on the context of grocery delivery in dense urban areas, where the high density of customers and numerous depots introduces significant logistical challenges. To enhance the practicality of the analysis, only the dataset instances with 300 requests and a service area overlap of 0.50 or higher are utilized. The overlap rate in this context reflects the extent to which customer requests are shared between neighboring depots. At an overlap rate of 0, all requests for each depot are confined strictly to its immediate neighborhood. As the overlap rate increases, more requests lie within the neighboring areas of other depots, reflecting greater interconnectivity and complexity in the service area. This increase in overlap rate aligns more closely with real-world urban scenarios, where depots often share service responsibilities due to dense populations and overlapping delivery zones.

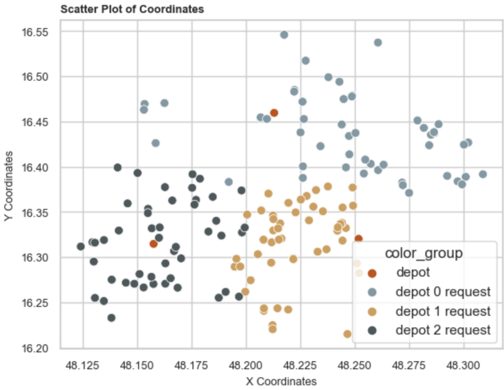

Figure 5.1: Overlap rate $o = 000$

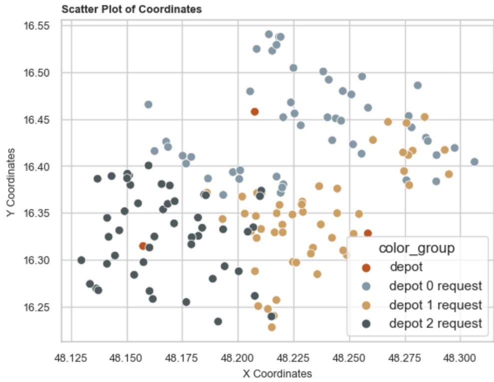

Figure 5.2: Overlap rate $o = 020$

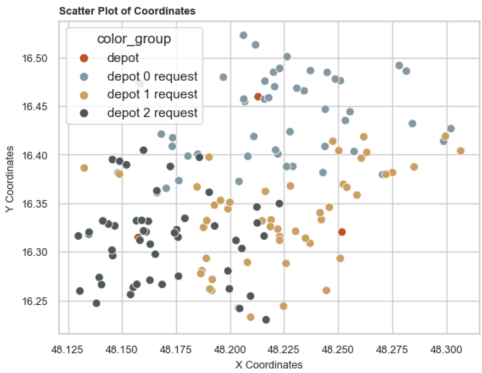

Figure 5.3: Overlap rate $o = 050$

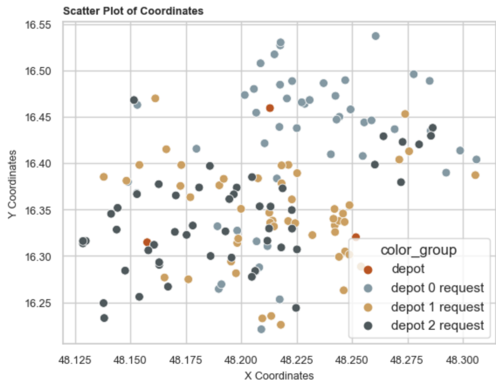

Figure 5.4: Overlap rate $o = 075$

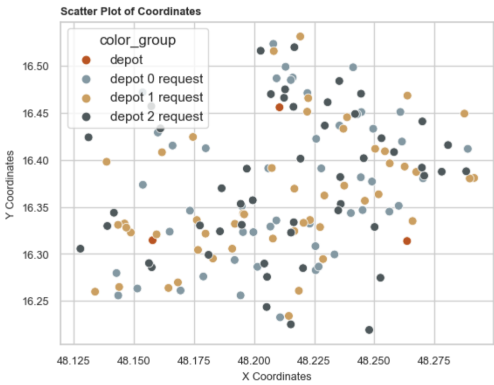

Figure 5.5: Overlap rate $o = 100$

5.2 Experimental Settings

The problem considers that (1) number of depots is 3, (2) each depot has maximum 3 vehicles, (3) total number of requests is 300, (4) working hour is separated into early delivery (10:00 to 14:00) and late delivery (14:00 to 18:00), (5) service time at each location is 240 seconds (6) tour plan must be scheduled before departing from the depot and cannot be changed during delivery process.

- **Algorithm setting:** To compare the effects of each method, we analyze the results across three different cases using the default parameter settings as in the table below
  - Only Local Search with Greedy approach.
  - Beam Search and the Move/Swap method.
  - Beam Search with the Move/Swap method and Simulated Annealing.

Parameter	**Value**
Number of beam search solutions (k)	5
Number of random permutations (P)	5
Iterations for Move/Swap and Simulated Annealing	200
Initial Temperature for SA	20
Probability of customer group for early session (p)	0.3
Probability of customer group for late session ($1 - p$)	0.7

Table 5.1: Parameter settings for the algorithm.

- **Depot setting:** To analyze the effect of using multiple depots versus a single depot, we compare the results of the following cases.
  - $[0], [1], [2]$: Three independent single depots.
  - $[0], [1, 2]$: Depot 0 works independently, depots 1 and 2 operate together.
  - $[0, 1], [2]$: Depot 2 works independently, depots 0 and 1 operate together.
  - $[0, 2], [1]$: Depot 1 works independently, depots 0 and 2 operate together.
  - $[0, 1, 2]$: All three depots operate together.

- **Time-window setting:** We analyze how the algorithm handles different time window lengths, with the expectation that shorter time windows result in better service.
  - 2-hour length: $tw = 2$
  - 1-hour length: $tw = 1$
  - 30-minute length: $tw = 0.5$
- **Parameters:** To compare fulfillment costs, we set the labor cost and transportation cost at \$30 and \$10 per hour, respectively, and the vehicle lease cost at \$250 per vehicle per day.
  - $CT_k = 10$
  - $CR_k = 30$
  - $CV_k = 250$

All results represent the averages from three independent runs, with each run considering 20 files, where the overlap rates are 0.50 and 0.75.

5.3 Computational Results

In this section, the proposed algorithm implementation will be analyzed. Comparisons among different algorithm, depot, and time-window length settings are presented below. In the table, results are presented as Local Search with Greedy approach/Beam Search and the Move-Swap method/Beam Search with the Move-Swap method and Simulated Annealing. The variables c and tw represent the maximum number of available vehicles and the time window length, respectively.

Acceptance Rate: According to the results, with the Move/Swap method, the improvement is more considerable as constraints tighten. For example, with $c = 9|tw = 0.5$ and depot setting [0, 1, 2], the acceptance rate improves from **0.884** to **0.943** (increase **5.9%**) only by adding the Move/Swap method. Correspondingly, with $c = 6|tw = 0.5$ and depot setting [0, 1, 2], the improvement rate is **7.1%**. Simulated Annealing consistently gets the highest acceptance rates across all depot configurations and time window lengths. The improvement is particularly significant for $tw = 0.5$ and $c = 6$, where constraints are stricter, and the algorithm's ability to explore a broader solution space while avoiding local optima becomes crucial.

	$c = 9\|tw = 2$	$c = 9\|tw = 1$	$c = 9\|tw = 0.5$
[0], [1], [2]	0.946 / 0.997 / **1.000**	0.899 / 0.981 / **0.988**	0.879 / 0.918 / **0.926**
[0], [1,2]	0.953 / 0.998 / **0.999**	0.906 / 0.986 / **0.993**	0.871 / 0.934 / **0.949**
[0, 1], [2]	0.948 / 0.999 / **1.000**	0.902 / 0.991 / **0.997**	0.888 / 0.937 / **0.939**
[0, 2], [1]	0.945 / 0.999 / **1.000**	0.890 / 0.983 / **0.997**	0.870 / 0.925 / **0.941**
[0, 1, 2]	0.953 / **1.000** / **1.000**	0.908 / 0.996 / **1.000**	0.884 / 0.943 / **0.970**

Table 5.2: Acceptance Rate for $c = 9$

	$c = 6\|tw = 2$	$c = 6\|tw = 1$	$c = 6\|tw = 0.5$
[0], [1], [2]	0.748 / 0.768 / **0.778**	0.637 / 0.680 / **0.683**	0.562 / 0.599 / **0.632**
[0], [1, 2]	0.788 / 0.810 / **0.847**	0.675 / 0.714 / **0.753**	0.601 / 0.633 / **0.681**
[0, 1], [2]	0.769 / 0.814 / **0.835**	0.680 / 0.733 / **0.767**	0.580 / 0.640 / **0.688**
[0, 2], [1]	0.778 / 0.819 / **0.849**	0.663 / 0.733 / **0.763**	0.597 / 0.659 / **0.679**
[0, 1, 2]	0.803 / 0.865 / **0.898**	0.730 / 0.812 / **0.832**	0.651 / 0.722 / **0.772**

Table 5.3: Acceptance Rate for $c = 6$

Fulfillment Cost: Move/Swap significantly reduces fulfillment costs, demonstrating its effectiveness in resource allocation and routing optimization. For example, with $c = 9|tw = 0.5$ and depot setting [0, 1, 2], the cost without Move/Swap is **4231/3960**, whereas with Move/Swap it drops to **3877**. Simulated Annealing achieves the lowest fulfillment costs in almost all scenarios. The method effectively balances exploration and exploitation, ensuring near-optimal solutions are found, particularly in collaborative depot configurations or when time windows are shorter.

	$c = 9 \| tw = 2$	$c = 9 \| tw = 1$	$c = 9 \| tw = 0.5$
$[0], [1], [2]$	4032 / 3702/ **3625**	4186 / 3963 / **3900**	4286 / 4111 / **4092**
$[0], [1, 2]$	3983 / 3654 / **3556**	4121 / 3876 / **3795**	4221 / 4010 / **3970**
$[0, 1], [2]$	3979 / 3650 / **3604**	4146 / 3894 / **3818**	4268 / 4057 / **3989**
$[0, 2], [1]$	3998 / 3668 / **3624**	4146 / 3923 / **3845**	4290 / 4075 / **4043**
$[0, 1, 2]$	3913 / 3543 / **3423**	4045 / 3818 / **3732**	4231 / 3960 / **3877**

Table 5.4: Fulfillment Cost for $c = 9$

	$c = 6 \| tw = 2$	$c = 6 \| tw = 1$	$c = 6 \| tw = 0.5$
$[0], [1], [2]$	2506 / 2489/ **2467**	2583 / 2563 / **2560**	2606 / 2586 / **2578**
$[0], [1, 2]$	2489 / 2448 / **2428**	2559 / 2522 / **2511**	2582 / 2555 / **2548**
$[0, 1], [2]$	2457 / 2429 / **2399**	2556 / 2516 / **2498**	2580 / 2554 / **2549**
$[0, 2], [1]$	2475 / 2424 / **2403**	2570 / 2527 / **2499**	2589 / 2543 / **2525**
$[0, 1, 2]$	2426 / 2335 / **2276**	2525 / 2468 / **2425**	2539 / 2502 / **2480**

Table 5.5: Fulfillment Cost for $c = 6$

Using a multi-depot setting offers a significant improvement in delivery service while maintaining a competitive acceptance rate and cost efficiency. For a single depot, the acceptance rate is **0.778** with $c = 6|tw = 2$, while the fulfillment cost is **2467**. In comparison, with three depots, the acceptance rate is slightly lower at **0.772** for $c = 6|tw = 0.5$, and the fulfillment cost increases only marginally to **2480**. This demonstrates that the multi-depot approach not only sustains a comparable acceptance rate and fulfillment cost but also significantly reduces the delivery time window from 2 hours to just 30 minutes, thereby optimizing both the delivery route and service quality.

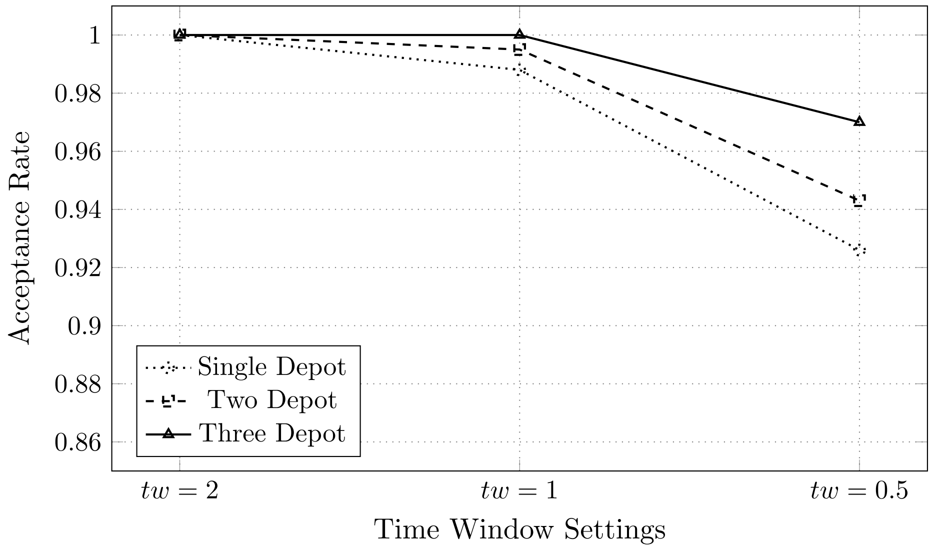

Figure 5.6: Acceptance Rate vs Time Window Settings $c = 9$

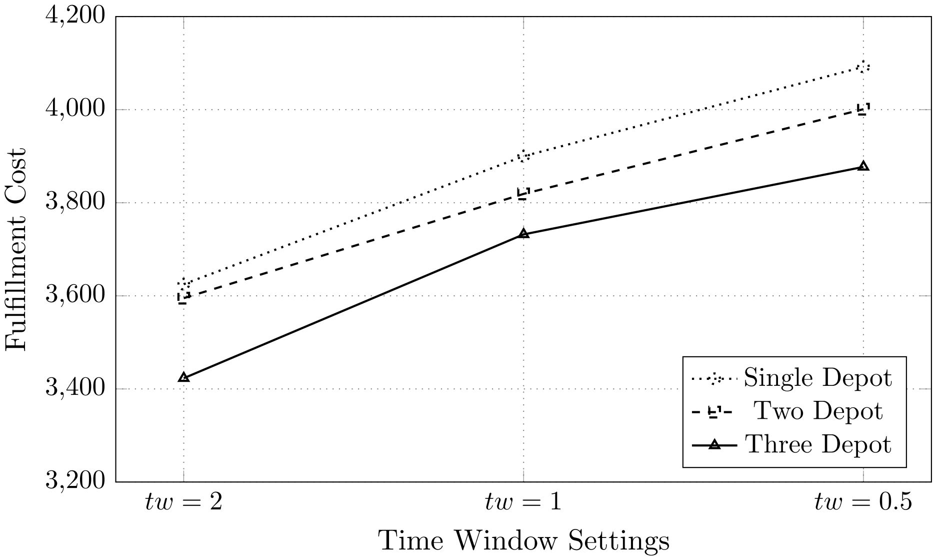

Figure 5.7: Fulfillment Cost vs Time Window Settings $c = 9$

As shown in the line chart for $c = 9$, all depot configurations achieve an acceptance rate of 1.0 when $tw = 2$, indicating no variation under large time windows. However, configurations with a greater number of depots demonstrate improved resilience to tighter time windows.

Specifically, the 3-depot configuration achieves up to a **5%** improvement in acceptance rate while saving **5%** fulfillment cost for $tw = 0.5$ compared to the single-depot configuration. For $c = 6$, a 3-depot configuration reduces fulfillment costs compared to a single depot, with savings increasing as the time window widens, such as an **8%** reduction for $tw = 2$.

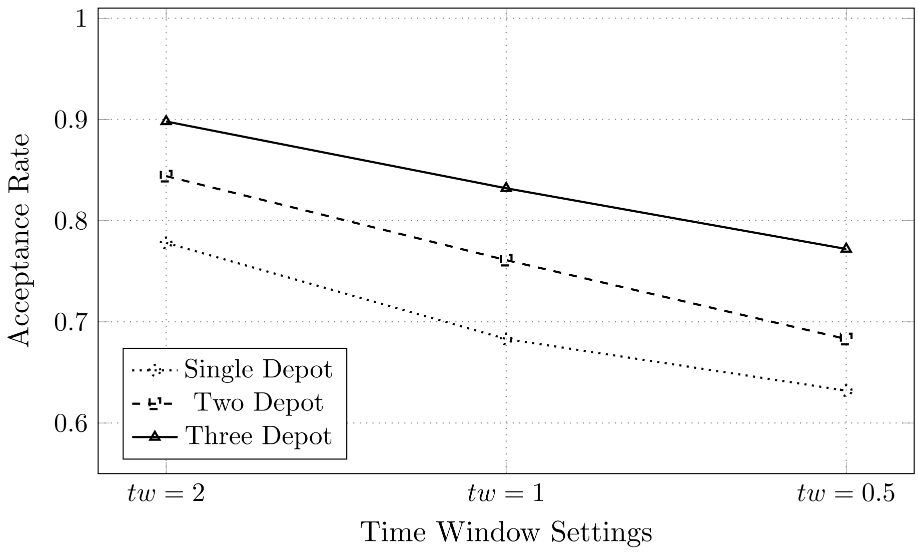

Figure 5.8: Acceptance Rate vs Time Window Settings $c = 6$

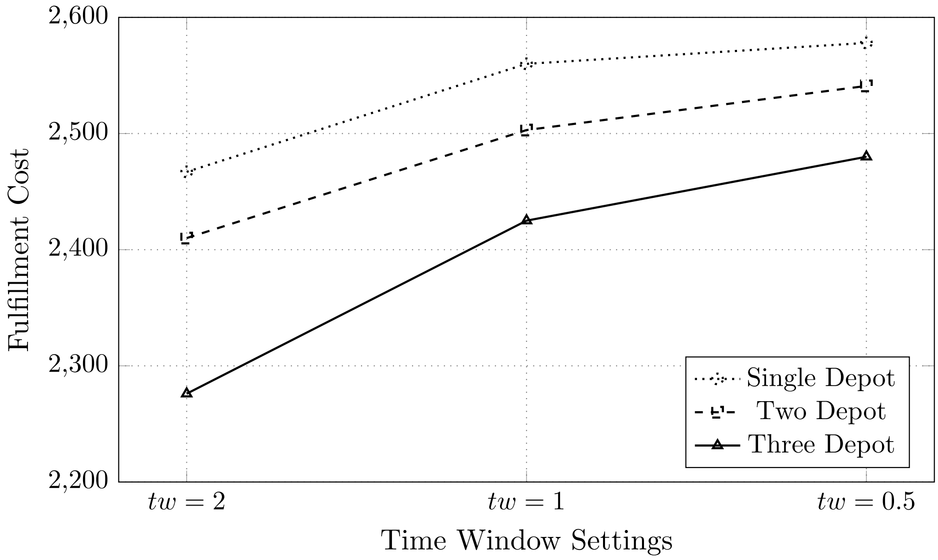

Figure 5.9: Fulfillment Cost vs Time Window Settings $c = 6$

Execution Time: All experiments were executed on a MacBook Air (Apple M2, 16 GB RAM) running macOS Sonoma 14.1.1.

	Single-depot	2-depot	3-depot
$tw = 0.5$	0.367 / 23.14 / 22.91	0.609 / 22.47 / 22.62	1.12 / 22.89 / 23.19
$tw = 1$	0.338 / 16.94 / 16.32	0.528 / 17.38 / 16.87	0.897 / 18.28 / 17.72
$tw = 2$	0.849 / 54.06 / 53.48	1.739 / 61.78 / 63.58	3.536 / 76.63 / 74.61

Table 5.6: Average Execution Time Across Different Depot Configurations (in seconds)

The table above presents the average **execution time** across different depot configurations for various time windows. The results indicate that while the total processing time for $tw = 1$ is more efficient compared to other configurations. However, by observing from the scatter plot, most individual requests are processed slower in $tw = 1$ than those in the $tw = 0.5$. This can be attributed to the stricter conditions imposed by the smaller time window, which often leads to situations where certain requests cannot find feasible solutions within the limited scheduling constraints.

For $tw = 0.5$, execution times mostly range between 0.05 to 0.35, with an average around 0.10 to 0.15 seconds each request. For $tw = 1$, execution times increase to around 0.1 to 0.80, with an average near 0.20 to 0.30 seconds. For $tw = 2$, execution times show an exponential trend with an average around 0.30 to 0.40 seconds.

For $tw = 0.5$ and $tw = 1$, execution times remain relatively low and stable in the early indices but linearly increase. Available delivery slots become increasingly limited as more requests are processed, leading to higher complexity in accommodating new orders, with a significant increase after index 200. As the tour becomes fuller, the optimization process requires more computational effort to fit additional requests while maintaining efficiency and service constraints. For $tw = 2$, the execution time increases exponentially. The larger time window initially allows for more flexibility in routing decisions; however, as the tour progressively fills up, finding feasible placements for new requests within the remaining slots becomes exponentially more difficult. Compared to smaller time windows, where execution time grows more gradually, the exponential increase in $tw = 2$ suggests that while larger planning horizons provide opportunities for better resource allocation early on, they also introduce challenges in later stages.

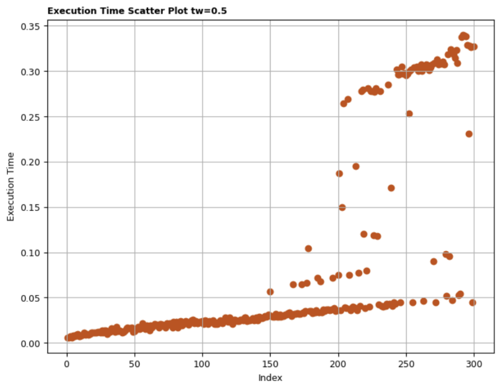

Figure 5.10: Execution Time for $tw = 0.5$

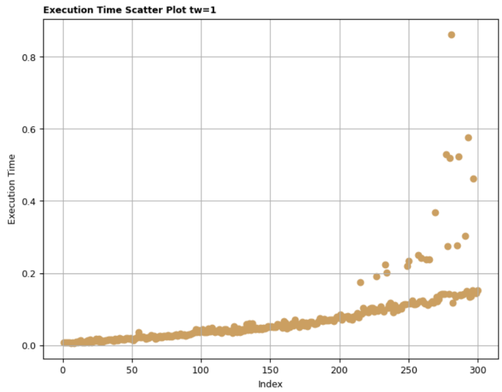

Figure 5.11: Execution Time for $tw = 1$

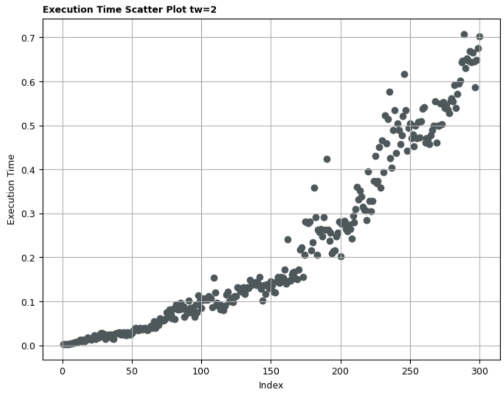

Figure 5.12: Execution Time for $tw = 2$

In practical applications, an execution time under 1 second is generally acceptable for providing customers with available time slots, ensuring a smooth user experience. The results indicate that for $tw = 0.5$ and $tw = 1$, execution times is highly practical for real-time applications. For $tw = 2$, it still remains within the acceptable range, making it suitable for scenarios where more comprehensive planning is required. Overall, for all tested time windows, the algorithm's execution time is practical.

Chapter 6

Conclusion

This thesis explored the multi-depot attended home delivery (AHD) problem with dynamic customer acceptance, aiming to optimize customer acceptance rates and fulfillment costs by analyzing the impact of varying the number of depots and time window durations. The study implements a combination of advanced optimization algorithms, including Beam Search, local search 2-opt, and Simulated Annealing, to develop efficient solutions for routing and scheduling in a dynamic environment.

The results demonstrate that incorporating multiple depots significantly enhances system performance compared to single-depot. Generally, a multi-depot system increased customer acceptance rates by over **5%** and up to **15%** while reducing fulfillment costs by more than **5%**. In particular, a multi-depot setting significantly enhances delivery service by reducing the time window from 2 hours to just **30 minutes** while maintaining a competitive acceptance rate. These improvements highlight the benefits of distributing operations across multiple depots, including enhanced scalability, better workload balance, and improved route optimization. Furthermore, the integration of dynamic acceptance mechanisms was shown to enable real-time decision-making while maintaining service quality and cost efficiency.

The contributions of this research can be summarized in two main aspects. First, it provides a practical framework for addressing the challenges of dynamic customer acceptance in multi-depot AHD systems. By prioritizing acceptance rate (AR) over fulfillment costs (FC), the proposed model ensures service reliability while minimizing delays and cancellations. Second, it offers insights into depot configurations, time window settings, and overall system performance, making it applicable to real-world scenarios where e-commerce logistics are increasingly dynamic and customer-centric.

While this study focused on operational optimization within a fixed set of constraints, there are several avenues for future research. Incorporating stochastic elements, such as variable travel times, uncertain demand patterns, or real-time traffic data, could further enhance the robustness of the proposed solutions. Additionally, integrating vehicle capacity constraints and exploring heterogeneous fleets could improve applicability to industries with diverse logistical requirements. Investigating flexible time-window management strategies or dynamic pricing models may also provide deeper insights into improving customer satisfaction and profitability. Lastly, extending the model to consider collaborative logistics, where multiple providers share resources and depots, could offer innovative solutions to challenges in the AHD domain.

In conclusion, this thesis makes a contribution to the field of attended home delivery by addressing the complexities of multi-depot systems with dynamic customer acceptance. The findings highlight the need for advanced optimization techniques to address changing market demands and support future improvements in last-mile delivery systems. This work connects academic research with practical applications, providing a strong basis for further progress in logistics and transportation management.

Chapter 7

Limitation and Future Work

Limitations

While this thesis provides significant insights into optimizing multi-depot attended home delivery (AHD) systems with dynamic customer acceptance, certain limitations and opportunities for future work remain.

One limitation is the lack of integration of historical data on customer demand. Understanding historical demand patterns can help improve the allocation of resources, such as modifying the number of vehicles assigned to each depot based on expected demand fluctuations. Incorporating such data could enhance the scalability and efficiency of the system, especially during peak periods or in regions with varying demand densities.

Another limitation is the absence of real data on customer preferences. The current model relies on assumptions or generalized data to simulate customer behavior. However, using real-world data on customer preferences, such as delivery time flexibility, service expectations, and regional trends, would make the model more realistic and aligned with actual customer needs. This integration could improve the accuracy of dynamic acceptance mechanisms and lead to better customer satisfaction.

Future Work

Furthermore, the problem addressed in this thesis focuses on grocery delivery, where vehicle capacity plays a critical role due to the size and weight of orders. However, the current model does not explicitly account for capacity constraints, which are essential for realistic and effective route planning in grocery logistics. Incorporating capacity constraints into the optimization framework would enhance the practicality and applicability of the model to real-world scenarios.

Additionally, the problem considered in this study is limited to next-day delivery. While this is a common practice in grocery delivery, there is potential to extend the model to include same-day delivery or more dynamic delivery scenarios, such as pickup-and-delivery problems. In such cases, vehicles would be allowed to pick up orders from any depot and continue delivering them, enabling greater flexibility and efficiency. This approach could better address customer demands for faster and more flexible delivery options.

Future work should explore integrating vehicle capacity constraints into the optimization model to improve its application to grocery logistics. Moreover, expanding the scope to include pickup-and-delivery problems with flexible depot usage would provide a more comprehensive solution, addressing the evolving demands of the logistics industry. These enhancements would not only improve operational efficiency but also support the development of more adaptive and customer-focused delivery systems.

References

A. H. Land, A. G. D. (1960), 'An automatic method of solving discrete programming problems', *Econometrica* **28**(3), 497.
URL: *https://doi.org/10.2307/1910129*

Akeb, H., Bouchakhchoukha, A. and Hifi, M. (2013), A beam search based algorithm for the capacitated vehicle routing problem with time windows, *in* '2013 Federated Conference on Computer Science and Information Systems', pp. 329–336.

Apptunix (2023), 'The rise and fall of gorillas: What happened to germany's fastest-growing grocery delivery unicorn?'.
Accessed: February 27, 2023.
URL: *https://www.apptunix.com/blog/what-went-wrong-gorillas-business-model/*

Barbarosoglu, G. and Ozgur, D. (1999), 'A tabu search algorithm for the vehicle routing problem', *Computers amp; Operations Research* **26**, 255–270.
URL: *https://dx.doi.org/10.1016/s0305-0548(98)00047-1*

Baños, R. (2013), 'A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows', *Expert Systems with Applications* **40**(5), 1696–1707.
URL: *https://doi.org/10.1016/j.eswa.2012.09.012*

Berger, J., Barkaoui, M. and Bräysy, O. (2003), 'A route-directed hybrid genetic approach for the vehicle routing problem with time windows', *INFOR: Information Systems and Operational Research* **41**, 179–194.
URL: *https://dx.doi.org/10.1080/03155986.2003.11732675*

Braekers, K. (2016), 'The vehicle routing problem: State of the art classification and review', *Computers amp; Industrial Engineering* **99**, 300–313.
URL: *https://doi.org/10.1016/j.cie.2015.12.007*

Burian, M., Köhler, C., Campbell, A. M. and Ehmke, J. F. (2023), 'Service time window selection for attended home deliveries: a case study for urban and rural areas', *Central European Journal of Operations Research* **32**, 267–294.
URL: *https://dx.doi.org/10.1007/s10100-023-00879-9*

Butler, S. (2024), 'Delivery firm getir to quit uk, europe and us and focus on turkey'.
Accessed: April 29, 2024.
URL: *https://www.theguardian.com/business/2024/apr/29/delivery-firm-getir-quit-uk-europe-us-turkey*

Campbell, A. M. and Savelsbergh, M. W. P. (2005), 'Decision support for consumer direct grocery initiatives', *Transportation Science* **39**, 313–327.
URL: *https://dx.doi.org/10.1287/trsc.1040.0105*

Christofides, N. and Eilon, S. (1972), 'Algorithms for large-scale travelling salesman problems', *Operational Research Quarterly (1970-1977)* **23**, 511.
URL: *https://dx.doi.org/10.2307/3007964*

Clarke, G. and Wright, J. W. (1964), 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research* **12**, 568–581.
URL: *https://dx.doi.org/10.1287/opre.12.4.568*

Contentstack (2023), 'Spar austria takes first step with composable commerce technology to future-proof its online presence and help drive sales'.
Accessed: July 12, 2023.
URL: *https://www.contentstack.com/company/press/spar-austria-takes-first-step-with-composable-commerce-technology-to-future-proof-its-online-presence-and-help-drive-sales*

Cordeau, J.-F. (2004), 'Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows', *Journal of the Operational Research Society* **55**(5), 542–546.
URL: *https://doi.org/10.1057/palgrave.jors.2601707*

Cordeau, J.-F., Laporte, G. and Mercier, A. (2001), 'A unified tabu search heuristic for vehicle routing problems with time windows', *Journal of the Operational Research Society* **52**, 928–936.
URL: *https://dx.doi.org/10.1057/palgrave.jors.2601163*

Cordeau, J., Gendreau, M. and Laporte, G. (1997), 'A tabu search heuristic for periodic and multi-depot vehicle routing problems', *Networks* **30**, 105–119.
URL: *https://dx.doi.org/10.1002/(sici)1097-0037(199709)30:2¡105::aid-net5¿3.3.co;2-n*

Croes, G. A. (1958), 'A method for solving traveling-salesman problems', *Operations Research* **6**, 791–812.
URL: *https://dx.doi.org/10.1287/opre.6.6.791*

Dantzig, G. B. and Ramser, J. H. (1959), 'The truck dispatching problem', *Management Science* **6**, 80–91.
URL: *https://dx.doi.org/10.1287/mnsc.6.1.80*

Dantzig, G., Fulkerson, R. and Johnson, S. (1954), 'Solution of a large-scale traveling-salesman problem', *Journal of the Operations Research Society of America* **2**, 393–410.
URL: *https://dx.doi.org/10.1287/opre.2.4.393*

Dondo, R. and Cerdá, J. (2007), 'A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows', *European Journal of Operational Research* **176**, 1478–1507.
URL: *https://dx.doi.org/10.1016/j.ejor.2004.07.077*

Dorigo, M., Caro, G. D. and Gambardella, L. M. (1999), 'Ant algorithms for discrete optimization', *Artificial Life* **5**, 137–172.
URL: *https://dx.doi.org/10.1162/106454699568728*

Ehmke, J. F. and Campbell, A. M. (2014), 'Customer acceptance mechanisms for home deliveries in metropolitan areas', *European Journal of Operational Research* **233**, 193–207.
URL: *https://dx.doi.org/10.1016/j.ejor.2013.08.028*

Elshaer, R. and Awad, H. (2020), 'A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants', *Computers amp; Industrial Engineering* **140**, 106242.
URL: *https://dx.doi.org/10.1016/j.cie.2019.106242*

Elting, S., Ehmke, J. F. and Gansterer, M. (2024*a*), 'Collaborative transportation for attended home deliveries', *Networks* **84**, 84–106.
URL: *https://dx.doi.org/10.1002/net.22216*

Elting, S., Ehmke, J. F. and Gansterer, M. (2024*b*), 'Replication data for: Collaborative transportation for attended home delivery (oa edition) version v1)'.
URL: *https://doi.org/doi:10.11587/VJ01D5*

Gevaers, R., de Voorde, E. V. and Vanelslander, T. (2011), 'Characteristics and typology of last-mile logistics from an innovation perspective in an urban context', *City Distribution and Urban Freight Transport* .
URL: *https://dx.doi.org/10.4337/9780857932754.00009*

Gillett, B. E. and Miller, L. R. (1974), 'A heuristic algorithm for the vehicle-dispatch problem', *Operations Research* **22**, 340–349.
URL: *https://dx.doi.org/10.1287/opre.22.2.340*

Glover, F. (1986), 'Future paths for integer programming and links to artificial intelligence', *Computers amp; Operations Research* **13**(5), 533–549.
URL: *https://doi.org/10.1016/0305-0548(86)90048-1*

Gmira, M. (2021), 'Tabu search for the time-dependent vehicle routing problem with time windows on a road network', *European Journal of Operational Research* **288**(1), 129–140.
URL: *https://doi.org/10.1016/j.ejor.2020.05.041*

Guo, Z., Yang, L., Jiang, X. and Sheng, G. (2017), 'The electric vehicle routing problem with time windows using genetic algorithm', *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* pp. 635–639.
URL: *https://api.semanticscholar.org/CorpusID:26380815*

Holland, J. H. (1992), 'Adaptation in natural and artificial systems'.
URL: *https://dx.doi.org/10.7551/mitpress/1090.001.0001*

Kirkpatrick, S. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.
URL: *https://doi.org/10.1126/science.220.4598.671*

Köhler, C., Ehmke, J. F. and Campbell, A. M. (2020), 'Flexible time window management for attended home deliveries', *Omega* **91**, 102023.
URL: *https://dx.doi.org/10.1016/j.omega.2019.01.001*

Lang, M. A. K. (2021), 'Anticipative dynamic slotting for attended home deliveries', *Operations Research Forum* **2**(4).
URL: *https://doi.org/10.1007/s43069-021-00086-9*

Leung, S. C., Zhou, X., Zhang, D. and Zheng, J. (2011), 'Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem', *Computers amp; Operations Research* **38**, 205–215.
URL: *https://dx.doi.org/10.1016/j.cor.2010.04.013*

Li, J. (2018), 'Benefit analysis of shared depot resources for multi-depot vehicle routing problem with fuel consumption', *Transportation Research Part D: Transport and Environment* **59**, 417–432.
URL: *https://doi.org/10.1016/j.trd.2018.01.026*

Lin, S. (1965), 'Computer solutions of the traveling salesman problem', *Bell System Technical Journal* **44**, 2245–2269.
URL: *https://dx.doi.org/10.1002/j.1538-7305.1965.tb04146.x*

Lockwood, T. and Burroughs, C. (2022), 'Rapid grocery delivery startup jokr has quietly exited europe without finding a buyer as rivals shutter or sell'.
URL: *https://www.businessinsider.com/grocery-delivery-jokr-quits-europe-2022-4*

Lone (2023), '2023 european e-commerce report', *Amsterdam/Brussels: Amsterdam University of Applied Sciences Ecommerce Europe* .

M. Padberg, G. R. (1987), 'Optimization of a 532-city symmetric traveling salesman problem by branch and cut', *Operations Research Letters* **6**(1), 1–7.
URL: *https://doi.org/10.1016/0167-6377(87)90002-2*

Ma, H. (2012), 'An investigation into the vehicle routing problem with time windows and link capacity constraints', *Omega* **40**(3), 336–347.
URL: *https://doi.org/10.1016/j.omega.2011.08.003*

Maroof, A., Ayvaz, B. and Naeem, K. (2013), 'Logistics optimization using hybrid genetic algorithm (hga): A solution to the vehicle routing problem with time windows (vrptw)', *IEEE Access* **12**, 36974–36989.

Michael Held, R. M. K. (1961), 'A dynamic programming approach to sequencing problems', *Proceedings of the 1961 16th ACM national meeting on* - pp. 71.201–71.204.
URL: *https://doi.org/10.1145/800029.808532*

Mir Ehsan Hesam Sadati, B. (2021), 'A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem', *Transportation Research Part E: Logistics*

and Transportation Review **149**, 102293.
URL: *https://doi.org/10.1016/j.tre.2021.102293*

Ochelska-Mierzejewska, J., Poniszewska-Marańda, A. and Marańda, W. (2021), 'Selected genetic algorithms for vehicle routing problem solving', *Electronics* **10**, 3147.
URL: *https://dx.doi.org/10.3390/electronics10243147*

OECD (2020), 'E-commerce in the time of covid-19. in oecd policy responses to coronavirus (covid-19)'.

Polacek, M., Hartl, R. F., Doerner, K. and Reimann, M. (2004), 'A variable neighborhood search for the multi depot vehicle routing problem with time windows', *Journal of Heuristics* **10**, 613–627.
URL: *https://dx.doi.org/10.1007/s10732-005-5432-5*

Prins, C. (2004), 'A simple and effective evolutionary algorithm for the vehicle routing problem', *Computers amp; Operations Research* **31**, 1985–2002.
URL: *https://dx.doi.org/10.1016/s0305-0548(03)00158-8*

Pureza, V. (2012), 'Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the vrptw', *European Journal of Operational Research* **218**(3), 636–647.
URL: *https://doi.org/10.1016/j.ejor.2011.12.005*

Reimann, M., Doerner, K. and Hartl, R. F. (2004), 'D-ants: Savings based ants divide and conquer the vehicle routing problem', *Computers amp; Operations Research* **31**, 563–591.
URL: *https://dx.doi.org/10.1016/s0305-0548(03)00014-5*

REWE (2023), 'Billa online store has expanded its range by 50 percent in three years'. Accessed: November 21, 2023.
URL: *https://rewe-group.at/en/newsroom/2023/11/billa-online-store-has-expanded-its-range-by-50-percent-in-three-years*

Rosenkrantz, D. J. (1977), 'An analysis of several heuristics for the traveling salesman problem', *SIAM Journal on Computing* **6**(3), 563–581.
URL: *https://doi.org/10.1137/0206041*

Sadati, , Çatay, B. and Aksen, D. (2021), 'An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems', *Computers Operations Research* **133**, 105269.

Salhi, S. (2014), 'The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation', *Computers amp; Operations Research* **52**, 315–325.
URL: *https://doi.org/10.1016/j.cor.2013.05.011*

Shati, P. (2024), 'Neural sequence generation with constraints via beam search with cuts: A case study on vrp', *Proceedings of the International Symposium on Combinatorial Search* **17**, 118–126.
URL: *https://doi.org/10.1609/socs.v17i1.31549*

Shi-Yi Tan, W.-C. Y. (2021), 'The vehicle routing problem: State-of-the-art classification and review', *Applied Sciences* **11**(21), 10295.
URL: *https://doi.org/10.3390/app112110295*

Skok, M., Skrlec, D. and Krajcar, S. (2000), The genetic algorithm method for multiple depot capacitated vehicle routing problem solving, *in* 'KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516)', Vol. 2, pp. 520–526 vol.2.

Solomon, M. M. (1987), 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research* **35**, 254–265.
URL: *https://dx.doi.org/10.1287/opre.35.2.254*

Strauss, A., Gülpınar, N. and Zheng, Y. (2021), 'Dynamic pricing of flexible time slots for attended home delivery', *European Journal of Operational Research* **294**, 1022–1041.
URL: *https://dx.doi.org/10.1016/j.ejor.2020.03.007*

Thangiah, S. (1995), 'Vehicle routing with time windows using genetic algorithms', *The Practical Handbook of Genetic Algorithms* .
URL: *https://dx.doi.org/10.1201/9781420050073.ch11*

Thangiah, S. R. (1998), 'A hybrid genetic algorithm, simulated annealing and tabu search heuristic for vehicle routing problems with time windows', *Practical Handbook of Genetic Algorithms* pp. 347–384.
URL: *https://dx.doi.org/10.1201/9780429128356-9*

Tie Jun Wang, K. J. W. (2012), 'Study on multi-depot vehicle routing problem based on cloud adaptive particle swarm optimization', *Applied Mechanics and Materials* **253-255**, 1369–1373.
URL: *https://doi.org/10.4028/www.scientific.net/amm.253-255.1369*

Wang, C. (2013), 'Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows', *IFIP Advances in Information and Communication Technology* pp. 170–177.
URL: *https://doi.org/10.1007/978-3-642-41263-9$_2$1*

Wang, Y. (2023), 'A clustering-based extended genetic algorithm for the multidepot vehicle routing problem with time windows and three-dimensional loading constraints', *Applied Soft Computing* **133**, 109922.
URL: *https://doi.org/10.1016/j.asoc.2022.109922*

Xu, Y., Wang, L. and Yang, Y. (2012), 'A new variable neighborhood search algorithm for the multi depot heterogeneous vehicle routing problem with time windows', *Electronic Notes in Discrete Mathematics* **39**, 289–296.
URL: *https://dx.doi.org/10.1016/j.endm.2012.10.038*

Yassen, E. T. (2017), 'An adaptive hybrid algorithm for vehicle routing problems with time windows', *Computers amp; Industrial Engineering* **113**, 382–391.
URL: *https://doi.org/10.1016/j.cie.2017.09.034*

Yildiz, B. and Savelsbergh, M. (2020), 'Pricing for delivery time flexibility', *Transportation Research Part B: Methodological* **133**, 230–256.
URL: *https://dx.doi.org/10.1016/j.trb.2020.01.004*

Yongbo Li, Hamed Soleimani, M. Z. (2019), 'An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives', *Journal of Cleaner Production* **227**, 1161–1172.
URL: *https://doi.org/10.1016/j.jclepro.2019.03.185*

Yu, B., Yang, Z.-Z. and Xie, J.-X. (2011), 'A parallel improved ant colony optimization for multi-depot vehicle routing problem', *Journal of the Operational Research Society* **62**, 183–188.
URL: *https://dx.doi.org/10.1057/jors.2009.161*

Yu, L., Li, M., Yang, Y. and Yao, B. (2009), 'An improved ant colony optimization for vehicle routing problem', *Logistics* .
URL: *https://dx.doi.org/10.1061/40996(330)493*

Zhang, D. (2017), 'A hybrid algorithm for a vehicle routing problem with realistic constraints', *Information Sciences* **394-395**, 167–182.
URL: *https://doi.org/10.1016/j.ins.2017.02.028*