A Study on Robot and Drone Assisted Delivery Problems and Their Solutions

Abdullahi Mohammed Jingi

A thesis presented for degree of

Doctor of Philosophy

School of Mathematics, Statistics and Actuarial Science

University of Essex

April 2024

Dedication

I dedicate this thesis to my loving parents, affectionate wife (Ruqayya Gidado Goni) and adorable children (Fatima and Muhammad).

Declaration

The work in this thesis is based on research carried out with my supervisor Prof. Xinan Yang at the School of Mathematics, Statistics and Actuarial Science, No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is all my own work, unless referenced, to the contrary, in the text.

Abdullahi Mohammed Jingi
April 2024

Acknowledgements

All praises and thanks to Almighty Allah for everything!

I would like to express my deep appreciation to the following individuals, institutions, and organizations whose unwavering support and contributions have been indispensable in the successful completion of this thesis:

1. My supervisor, Prof. Xinan Yang: Your guidance, expertise, and unwavering support have been invaluable throughout this research journey. Xinan, I am profoundly grateful for your time, patience, motivation, and immense knowledge. Your supervisory guidance has been a cornerstone, significantly contributing to both my PhD research and thesis writing. Thank you once again for placing your trust in me.

2. Prof. Abdel Salhi, the Chair of my supervisory board: Thank you for your guidance, mentorship, and valuable insights that have greatly enriched my research experience.

3. My beloved wife, Ruqayya Gidado Goni: Your unwavering patience, encouragement, love, care, and understanding throughout the years dedicated to this endeavor have been a source of strength. Thank you for being my emotional support and for taking care of our children during these challenging years. Your presence has made this journey more meaningful.

4. My children, Fatima and Muhammad: Your smiles and understanding have made the sacrifices made in pursuit of this thesis all the more worthwhile.

5. My family and family-in-law: Your unwavering love, support, and encouragement have been the bedrock of my journey. Special thanks to my parents, Alhaji Muhammad Jingi and Hajiya Fadimatu Muhammad Jingi, as well as

my numerous brothers, sisters, and friends. Your collective support has been invaluable throughout this journey.

6. My dear friends and colleagues, Zurkannain and Hosang: Thank you for your unwavering support and encouragement during this challenging but rewarding academic endeavor.

7. The dedicated staff of the School of Mathematics, Statistics and Actuarial Sciences at the University of Essex: Thank you for fostering an environment conducive to learning and research.

8. Tertiary Education Trust Fund (TETFund) and Adamawa State University, Mubi: Your generous sponsorship and financial support made it possible for me to pursue and complete my studies and research.

Your collective support and guidance have been instrumental in bringing this thesis to fruition, and for that, I am profoundly grateful.

Abstract

The surge in online retail, driven by population growth, technology, and the Covid-19 pandemic, emphasizes the need for efficient last-mile delivery. This thesis explores Robot and Drone-Assisted Delivery Problems, addressing the need for innovative solutions in the integration of trucks and delivery robots or drones.

Recent technological advancements allow drones to be launched or collected from moving vehicles without human intervention, challenging the common notion of restricting these actions to when the truck is stationary. Chapter 3 of this thesis introduces the Covering Salesman Problem with Nodes and Segments Using Drones (CSPNS-D), a problem whose approach determines the maximum coverage area of nodes or links serviced by a drone as the truck traverses the corresponding node or link. Three Mixed Integer Linear Programming (MILP) models, representing various drone coverage areas, were proposed to minimize the truck's working span. Results demonstrate optimal solutions for up to 35 customers, with substantial savings compared to the Traveling Salesman Problem (TSP). Additionally, a computationally efficient link removal heuristic is presented for larger instances.

Chapter 4 introduces the Robot-Assisted Delivery Problem (RADP), integrating trucks, robots, and local depots. Two consistent MILP models, RADP-1 and RADP-2, optimize delivery schedules, with RADP-1 proving more efficient due to a large number of feasible operations in RADP-2. RADP-1 considers each arc and node separately in the modelling process, whereas RADP-2 treats combinations of arcs and nodes as a single operation. Unlike the CSPNS-D models, RADP only allows launching and collection of robots when the truck is stationary. RADP models, like CSPNS-D, pose NP-hard challenges, necessitating heuristic approaches for larger problems. The proposed P-Heur and K-Heur heuristics prioritize operations based

on the node-time ratio and employ a K-Means algorithm for cluster decomposition and MILP solution, respectively, effectively addressing larger-scale challenges.

Publications

Most of the work of this thesis has been published or is going to appear for publication.

- Parts of Chapter 4 in this thesis have been published in :
  Abdullahi Mohammed Jingi, Xinan Yang. "Robot-Assisted Delivery Problems and Their Exact Solution". *Optimization and Learning, 6th International Conference, OLA 2023, Malaga, Spain, May 3-5, 2023 Proceedings* ***1824****,PP 341-353*

- Parts of Chapter 4 have also been submitted to the IMA Journal of Management Mathematics for publication in a special issue entitled *"Mathematical Models for Sustainable Supply Chains and Operations (2023)"*. The paper, titled "Enhancing Delivery Efficiency Using Robot-Assisted Systems," authored by Abdullahi Mohammed Jingi and Xinan Yang, has been reviewed and is currently undergoing revision.

- Parts of Chapter 3 are formatted as a journal article and are currently undergoing proofreading for submission for publication.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Online shopping is becoming increasingly prevalent, particularly in developed countries, gradually replacing the traditional in-store shopping experience where customers physically visit retail stores. The emergence of online shopping has had a substantial impact on retail companies, boosting their sales and reshaping business dynamics. It enables retailers to enhance customer service by ensuring timely delivery to specified locations.

In-store shopping has witnessed a significant decline, primarily due to the global COVID-19 pandemic. This shift in consumer behavior may persist even after the pandemic subsides. Online retail has experienced remarkable growth over the past few decades, driven by technological advancements, population growth, and urbanization. Business-to-Consumer sales have been consistently increasing at a high rate annually. The COVID-19 pandemic has further accelerated online retail sales as people turned to online shopping while confined to their homes. In

the UK, the proportion of online spending reached an all-time high during the COVID-19 pandemic in April 2020, surging to 30.7%, a substantial increase from the 19.1% reported in April 2019 before the pandemic [2]. Government-imposed stay-at-home orders during the pandemic have altered shopping habits, which are likely to endure beyond the crisis, favoring e-Commerce and digital shopping experiences. As the number of online shoppers continues to rise, retail companies are confronted with challenges such as maximizing profit, delivering a satisfactory customer experience, and reducing carbon emissions. Therefore, they seek more efficient methods for distributing goods across their logistics networks.

Despite a reduction in road transport in the UK in 2020 due to nationwide lockdowns, the transportation sector remains the largest emitter, responsible for nearly a quarter of emissions (24%), surpassing even the energy supply sector at 21% [3]. To mitigate the negative impact of conventional trucks for last-mile deliveries, which include high logistics costs due to rising fuel and energy expenses, longer delivery times, and extended truck operation hours, several innovative concepts have been proposed. These include automated delivery robots and aeriel drones, designed to minimize truck travel time and emissions during deliveries [4], [5]. Companies like Amazon, DHL, and Google have introduced automated modes of freight distribution. For instance, Amazon's Prime Air and DHL's Parcelcopter are used to deliver packages in areas with low accessibility or long delivery times. Google's Project Wing is developing drones capable of delivering larger items than Prime Air and Parcelcopter [6], [7]. Mercedes Benz has partnered with Starship Technologies to create autonomous robots for last-mile deliveries, monitored by

customers via smartphones, allowing cargo compartment access upon the robot's arrival [8].

While both drones and robots offer similar advantages, such as faster deliveries, increased sales, and reduced traffic and emissions [1], they possess distinct strengths and weaknesses. Table 1.1 below provides an overview of the delivery challenges associated with autonomous robots, drones, and traditional trucks across key areas. Drones are known for their faster travel between two points in Euclidean space [9]. On the other hand, robots exhibit superiority in various aspects, with the potential for multiple compartments and a higher covering range for multiple deliveries in a single trip, along with a greater payload capacity to compensate for their slower speed [10, 11]. Although both contribute to emissions reduction, drones tend to consume more energy than robots, especially in windy conditions, whereas robots are less sensitive to weather conditions [12]. Drones also face higher legal concerns due to legislation, as they are perceived to pose greater threats to public infrastructure [10]. Finally, robots move silently on pedestrian walks without privacy interference [11], whereas drones can generate noise pollution that might raise public health concerns [13].

Table 1.1: Delivery Challenges of emerging and traditional delivery method [1]

Delivery Challenges	Emerging Deliveries		Traditional Deliveries
	(ADRs)	Drones	Trucks/Vans
Payload Capacity	Medium	Low	High
Speed	Low	High	Traffic Dependent
Range	Low-Medium	Low	High
Control	Autonomous	Autonomous	Human
Traffic impact	Depends	Low	High
Energy efficiency	High	Medium	Low
Legal concerns	Medium	High	Low
Noise	Low	High	High

Autonomous Delivery robots (ADRs) and autonomous aerial drones are used in conjunction with traditional trucks to optimize delivery schedules. Due to their limited capacity, they serve as assisting modes in conventional freight distribution.

1.2 Thesis Organization

Chapter 2 provides an overview of drone-assisted and robot-assisted delivery problems, detailing various variants of the Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) involving drone/robot-assisted delivery mechanisms.

Chapter 3 introduced the Covering Salesman Problem with Nodes and Segments using Drones (CSPNS-D), a problem integrating a single truck with multiple drones allowing for robot launch and collection while the truck is en route. Three Mixed Integer Linear Programming (MILP) models, considering different drone coverage areas, were formulated to minimize delivery time in the drone-assisted truck delivery system. The models were tested on benchmark data sets, and a link/node removal heuristic is proposed for larger instances. The chapter also explores insights into drone flying capabilities and geographic features of the service area.

Chapter 4 introduces the Robot-Assisted Delivery Problem (RADP), integrating a conventional truck, robots, and local depots into a comprehensive model where a customer can be serviced by either the truck, a robot, or from a local depot. Two distinct but consistent Mixed Integer Linear Programming (MILP) models, RADP-1 and RADP-2, were developed to optimize delivery schedules, minimizing

the total truck working time. The chapter also presents the performance of the models on small-scale scenarios.

Chapter 5 proposes two heuristic approaches, P-Heur and K-Heur, to enhance RADP solutions for larger instances. P-Heur is a greedy heuristic that prioritizes operations based on the node-time ratio within the RADP-2 model, while K-Heur decomposes the problem into sub-problems and addresses them using CPLEX. The heuristics are discussed in detail with an illustration. The chapter also presents the results of the heuristics on larger instances.

Chapter 6 provides an overall conclusion based on the results from the previous chapters. It discusses the pros and cons of the work and outlines potential directions for future research.

Appendix A proves Proposition 1, which asserts that any point P on the boundary of the ellipse defined by the foci (E, B) lies inside the ellipse defined by (D, B).

Appendix B proves Proposition 2, which states that the Link-Node-Link Operation extends the coverage area of the Link-Node Operation for all $\alpha > 1$.

Appendix C presents statistical comparisons of CSPNS-D's results on benchmark instances.

Chapter 2

Literature Review

2.1 Introduction

This chapter discusses various literature related to the Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), and their respective variants that incorporate delivery robots and drones within the delivery system. In recent years, a spectrum of innovative concepts for last-mile deliveries has emerged. These concepts includes alternative fuel vehicles, autonomous/self-driving vehicles, and controlled access to car trunks. Interested readers looking for a more in-depth understanding of these concepts can refer to [4] and [14]. Interestingly, among these innovations, the use of autonomous vehicles such as drones and robots stands out as one of the most popular options.

Our research aligns with existing work involving unmanned aerial drones and autonomous robot-based delivery models. Consequently, we explore these aspects within the literature discussion below.

2.2 The Drone Delivery

A rich literature exists on drone-assisted delivery compared to robot-assisted delivery. Despite their shared capability for autonomous operation, notable differences between the two vehicles were discussed earlier in the previous chapter.

Drone-assisted delivery can be viewed as an extension of the typical Traveling Salesman Problem (TSP). To establish the foundation for our discussion, we first provide a review of the TSP in Section 2.2.1. On the other hand, considering our specific problem setting involving drones, it is worth noting that the drone can take off at any point along the truck route to serve its customers. The area covered by the drone's flying range from the nodes and links of a given truck route exemplifies the so-called Covering Salesman Problem (CSP). Therefore, in Section 2.2.2, we delve into the CSP and explore its applications in existing works, paving the way for the introduction of our Covering Salesman Problem with Nodes and Segments using Drones (CSPNS-D) model.

2.2.1 Objects Assisted Traveling Salesman Problem

The Traveling Salesman Problem (TSP) stands out as one of the most well-known challenges in combinatorial optimization. Numerous models and methodologies have been developed to address both the traditional TSP and its extensions across various domains. For a more comprehensive overview, please consult [15, 16]. However, our focus in this section is directed towards extended TSP models and methodologies tailored for truck-drone delivery applications.

In [17], objects to synchronize in VRP are categorized into various types, leading to different solution approaches. Specifically, an object attached to a truck that cannot operate independently falls under the category of non-autonomous vehicles. An example of a non-autonomous VRP is the Truck and Trailer Routing Problem (TTRP), where a customer can be served either by a truck with a combined trailer or by the truck alone without a trailer [17]. Due to the NP-hard nature of such problems, TTRP solutions typically rely on (meta-)heuristics such as Tabu Search [18, 19] and Simulated Annealing [18–20]. In the context of CSPNS-D, a customer can be served either by a truck carrying multiple drones or by a drone alone. A crucial distinction is that a drone can serve a customer independently, whereas a trailer cannot. Therefore, drones are considered autonomous vehicles [17].

[5] introduced the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling TSP (PDSTSP). The FSTSP is applicable in situations where the distribution center is relatively distant from customer locations, and a single drone is synchronized to work in connection with the delivery truck. The truck carries the drone onboard from the distribution center to customer locations, where it can be launched to serve customer orders. However, if the distance from the distribution center to customer locations falls within the drone's flight range, the PDSTSP is considered. In this case, the truck and the drone operate independently from the distribution center to customer locations, with no synchronization required between the two vehicles. A mixed-integer linear formulation of the FSTSP is provided, which is then solved by a two-stage greedy heuristic. In the first stage, a Traveling Salesman Problem (TSP) solution (Truck only) is found, and in the second stage, this solution is improved by examining if

some orders can be served by the drone. One remarkable feature of the FSTSP is that the drone can take-off and land only at customer locations while the truck is stopped. The PDSTSP is introduced in cases where a significant number of customers are within the flight range of the drone from the distribution center. In this scenario, these customers are served by the drone launched from the distribution center. Due to the NP-hard nature of these problems, commercial solvers might take several hours to solve the models, even with instances involving only ten customers, to optimality.

[21] modified the FSTSP by allowing a single truck with multiple drones into the multiple FSTSP (mFSTSP). Optimum solutions for only 8 nodes is obtained by their MILP, they proposed a heuristic approach that consist solving a sequence of three sub-problems.[22] introduced the capability for drones (when empty) to pick up items in addition to their delivery function within the Parallel Drone Scheduling TSP (PDSTSP). In this scenario, if a drone completes a drop, it can either fly back to the depot for the next delivery or proceed to the next customer location for a pick-up. The study extended the PDSTSP into the PDSTSP drop-pickup (PDSTSP+DP), involving multiple trucks, multiple drones, and multiple depots. The problem is modeled using a constraint programming formulation for the PDSTSP+DP, and exact solutions were acquired for small instances.

The study conducted by [6] investigated a Drone Scheduling Problem (DSP) wherein a single truck is equipped with multiple drones and customer shipments. Due to the limited capacity of each drone, it can only carry a single customer shipment. Once loaded, the drone takes off from the top of the truck, delivers the shipment to the respective customer, and then returns to the truck. The truck

follows a predetermined route with fixed stop points where the take-off and landing of drones occur. In this setup, all customers are serviced by drones, and the truck is solely used as a mobile depot for launching and collecting drones. The authors presented two mixed-integer programming models for the problem, aiming to minimize the total time of the delivery tour with all drones returned. One of the models has proven successful in solving instances with up to one hundred customers in just a few minutes of computational time. On a similar note, [23] introduced the Travelling Salesman Problem with Drone (TSP-D), exploring the concept of transporting drones and shipments along a given truck route while considering the parallelization of different delivery tasks to reduce the total time required to serve all customers. The TSP-D approach takes into account the maximum flight distance of drones and the speed ratio between drones and trucks, inherited by our Covering Salesman Problem with Nodes and Segments using Drones (CSPNS-D) presented in Chapter 3. One distinction between TSP-D and FSTSP is that the latter allows trucks and drones to travel either in tandem or independently from the depot, while the former only permits trucks to travel in tandem with the drones from the depot. The problem is formulated as an integer programming model capable of solving instances with up to twelve customers to optimality. Additionally, heuristic approaches involving greedy and exact partitioning algorithms, along with local searches, are proposed. The numerical results provide insights into the maximum savings that can be achieved by TSP-D compared to traditional truck-only delivery.

To tackle problems with larger instances, [24] employed a dynamic programming approach for the TSP-D. They divided the problem into three sub-problems

and prevented sub-problems from growing by restricting the number of valid operations, thereby optimizing overall running time. In a different approach, [25] applied a branch-and-bound approach to the TSP-D and introduced boost heuristics. Furthermore, [26] proposed various Mixed-Integer Linear Programming (MILP) formulations for special cases of the TSP-D. Additionally, they employed an exact branch-and-price algorithm based on a set partitioning formulation, achieving the optimal solution for problems with up to 39 customers.

In the study conducted by [27], a multi-truck, multi-drone vehicle routing problem with drones (VRPD) is explored. In this scenario, a drone is dispatched from the top of a truck to transport a package to a single customer and then returns to the top of the truck. The truck is also permitted to make deliveries to customers. The primary objective is to minimize the total time required to complete the tour, ensuring that all vehicles are returned to the central depot. The study derived several worst-case results, which depend on the number of drones per truck and the speed of the drones relative to the speed of the truck. An extension of this work is presented by the same authors in [28]. In this subsequent work, the authors build upon the earlier paper's worst-case results and establish connections with the vehicle routing problem (VRP) and Amdahl's Law. They conclude that the VRPD model offers more significant practical advantages compared to traditional VRP approaches.

In the study by [29] and [30], a two-echelon vehicle routing problem with drones (2EVRPD) is investigated, where the truck operates at the first level and the drone at the second level. In the former, the problem involves multiple trucks, each having limited space to carry a certain number of drones. The drones can

make deliveries to multiple customers in one flight before returning to the truck. The authors assumed that multiple drones are not allowed to be launched or retrieved at the same customer node at any given time. The problem is modeled as a mixed-integer programming formulation that minimizes the total truck arrival time of all trucks at the depot. Exact optimal solutions for the model were obtained for fewer than 10 nodes with the CPLEX solver, while solving larger problems could require too much time to obtain the exact solution. In contrast, the latter considers a problem with a single truck carrying only one drone, but the drone can also make multiple deliveries in one flight before returning to the truck. The truck serves a dual role as both a delivery resource and a mobile depot for the drone. The authors proposed a two-stage route-based modeling approach that optimizes both the main route of the truck and the drone flying routes to complete the delivery of all parcels, considering the effect of varying payload on energy consumption. They presented an algorithm based on Simulated Annealing (SA), which proves to be efficient for different scales. Experimental results of sensitivity analysis tests indicate that the employment of drones can save more costs when there are more light parcels for delivery.

While majority of studies on drone-assisted deliveries focused on drones being launched from a vehicle, servicing a customer and returning to the vehicle only when the truck stops at customer locations or depots, there are few studies that allow drones to perform en-route operations. Drone operations might start and end along the links travelled by a truck. in a study by [31], they proposed an extension to the VRP known as the Vehicle Routing Problem with Drones and En-route Operations (VRPDERO). This model involves multiple trucks and multiple

drones. Unlike traditional models where drones take off and land only at customer locations, in VRPDERO, drones can also take off and land along the route traveled by the truck. In cases where the drone is significantly faster than the truck, it can perform multiple operations and return to the truck as it travels along its route from one location to another. While each drone operation is limited to serving only one customer, multiple operations can be executed considering the faster speeds of drones compared to trucks. The approach for en-route operations begins with the optimal Traveling Salesman Problem (TSP), building the routes by inserting drone-service orders using a greedy approach. However, it's worth noting that adjusting the drone launching and landing nodes on the link is deemed expensive in their proposed method.

Another work closely related to our work is the work presented in [32] which introduces the enroute truck-drone operations that allow drones taking-off on the truck links as well as at customer locations. This work deploys the TSP-D model proposed by [23], and considers the application scenario with a single truck and a single drone. The approach starts by finding the TSP solution using Lin-Kernighan heuristic proposed by [33], then constructs the TSP-D solution considering only nodes as possible taking-off or landing points. To further improve this solution, taking-off and landing points are re-located to the TSP links by drawing circles from the drone covered nodes, considering the maximum flying capability of the drone. In summary, the enroute solution of [32] is constructed by performing a sequence of local improvements on the optimal (or sub-optimal) TSP, while our approach does not rely on solving the TSP, but rather constructs the optimal truck-drone solution directly from scratch. In addition, as the taking-off and

landing points on the links are found by drawing circles, the drone flying route proposed by [32] has to form isosceles triangles with the truck link. Our models, on the other hand, considers less restrictive drone operations where the lengths of the outbound and inbound drone routes can be nonidentical. Our synchronization requirement is also higher as [32] allows truck waiting for the drone but we do not in all models except one.

Our approach differs from the ones mentioned above as they primarily depend on optimizing the Traveling Salesman Problem (TSP), seeking optimal routes. In contrast, we focus on determining the maximum coverage area of nodes and links that a drone can service when the corresponding node or link is traversed by the truck. This approach eliminates the need to explicitly consider where drones should take off or land, streamlining the analysis process.

2.2.2 Covering Salesman Problem

The Covering Salesman Problem (CSP) was initially introduced by [34] as a generalized traveling salesman problem. The objective of the CSP is to identify the minimum-cost tour that visits a selected subset of given nodes, with the additional constraint that every node not included in the tour must fall within a specified covering radius of the selected tour nodes. The authors presented a mathematical formulation for the CSP and introduced a heuristic approach inspired by both the set covering problem and the Traveling Salesman Problem (TSP). A specific case of the CSP is the Covering Tour Problem (CTP), as outlined in [35]. In the CTP, the node set is divided into two distinct subsets: one set represents the nodes that must be visited by the tour, and the other denotes the nodes that are to be covered.

Expanding on the concept, the Generalized Covering Salesman Problem (GCSP), introduced by [36], assumes that each node must be covered at least k times. The GCSP defines three variations by imposing constraints on the number of times the tour should visit each node [36]

A particularly relevant variant of the CSP for our problem is the Covering Salesman Problem with Nodes and Segments (CSPNS), as proposed by [37]. In this variation, the authors extended the covering concept to include areas within the radius of both traveled nodes and edges. They formulated a mathematical model for CSPNS and were able to find exact solutions for problems involving up to 100 nodes. Additionally, they proposed a local search heuristic that begins with the solution obtained from the Traveling Salesman Problem (TSP).

Another variant of the Covering Salesman Problem (CSP) that is conceptually connected to the Covering Salesman Problem with Nodes and Segments using Drones (CSPNS-D) is the Time Constrained Maximal Covering Salesman Problem (TCMCSP), established by [38]. In TCMCSP, the node set is categorized into depots, customers, and facilities. Facilities have demands from customers within their cover distance. The primary objective of TCMCSP is to maximize the number of covered customers within a specified time limit. The authors present mathematical formulations and a heuristic algorithm to address TCMCSP. Furthermore, [39] specifically focus on a special case of TCMCSP, where each facility has differently weighted elements in the Time Constrained Maximal Covering Salesman Problem with Weighted Demands and Partial Coverage. This particular problem aims to find a tour that maximizes the total demands of a subset of customers within a restricted time frame

Our study extends the concept of CSPNS to introduce the CSPNS-D problem, as presented in Chapter 3. It specifically examines the maximum covering areas of various drone operations. To address this, three Mixed-Integer Linear Programming (MILP) models are developed, each tailored to cover different choices of drone operations.

2.3 The Robot Delivery

Similar to drones, autonomous robots have the potential to revolutionize last-mile delivery, improving efficiency and reducing operational costs, especially in the retail sector [40]. Despite the rapid growth in studies on last-mile delivery involving autonomous robots, there is a limited number of articles that model and optimize the operations of robot-based last-mile deliveries compared to drones.

The introduction of a scheduling procedure involving truck-based robot-assisted delivery can be attributed to [8], who proposed a framework for launching autonomous robots from a single truck to serve deliveries with time windows. The process involves loading customer shipments and robots onto a truck from a central store or depot and transporting them to a drop-off point in the city center. At the drop-off point, one or more robots are loaded with shipments and released to perform last-mile deliveries. After completing the deliveries, each robot autonomously returns to the drop-off point. Simultaneously, the truck proceeds to the next drop-off point to release more robots. If all the robots have been deployed, but the truck still carries more shipments, it heads to the nearest decentralized robot depot to pick up additional robots and launch them for the

next set of deliveries. It is worth noting that the single truck serves solely as a mobile distribution center for the robots and does not handle customer orders. The authors made an assumption of a network with an unlimited number of robots at every robot depot, which may not align with practicality since, in reality, it's economically unfeasible for companies to invest in a vast fleet of robots that remain largely unused most of the time.

In their study, [41] conducted simulations to evaluate the performance of parcel delivery models for two different scenarios. The first scenario does not involve the use of delivery robots; instead, parcels are delivered by trucks originating from a central hub. The delivery region is divided into patches, with each patch containing one or more micro-depots. Customers are assigned to these micro-depots, enabling the rerouting of parcels to the nearest micro-depot in case of delivery failures. The second scenario explores a combination of conventional trucks and delivery robots. This model assumes that only a small percentage of parcels (0-3%) are designated for delivery by robots. When such packages are identified, a separate truck is responsible for transporting them from hubs to the designated micro-depots, where the robots take over the final delivery to customers. Each micro-depot is serviced by a single robot. Simulation results indicate that the use of delivery robots for time window-based deliveries does not significantly impact the total tour length or total fuel consumption, reflecting operational costs. However, this approach raises concerns about its economic feasibility, as dedicating an entire truck to transport a small percentage of parcels may not be cost-effective.

Another study that adopts a similar approach of transporting robots on a conventional truck and deploying them from the truck is presented in [42]. In

contrast to [8], where each robot is assigned a single delivery capacity, this study assumes that a robot can serve up to six customers. To evaluate the efficiency of these robots in terms of total delivery time, the authors employed continuous approximation methods. A comparative analysis was conducted, pitting this approach against the use of a standard delivery truck. The comparison was made with the assumption that a truck has the capacity to carry a maximum of eight robots, and each robot can travel up to four miles to complete deliveries. The study's findings suggest that autonomous robots may outperform traditional delivery vans, especially when the average delivery time per customer is high or when customer densities increase.

In their research, [43] conducted a study that explores the location-routing problem employing multi-compartment robots with a consideration for customer time windows. In this scenario, the robots start and terminate their tours at a designated robot hub and have the capability to make deliveries to multiple customers before returning to the hub. To enhance efficiency in terms of electricity consumption, the study assumes that the robots have the ability to swap their batteries to mitigate delays associated with charging times. The authors formulate the problem as a mixed-integer programming problem, which simultaneously addresses various aspects. It determines the optimal locations of robot hubs, defines the set of tours, and calculates the required number of robots to minimize the total costs for one working day. These costs encompass the rental cost of the utilized robots, the personnel cost associated with configuring the robots, and the delivery cost for all tours.

A novel approach, the Two-Echelon Van-Based Robot Hybrid Pickup and Deliveries (2E-VRHPD), was introduced by [44]. In this system, vans equipped with small robots travel along tier-1 routes, making stops at designated parking nodes to drop off and/or pick up robots. Vans also use these stops to replenish their robot supply and swap robot batteries. Some customers reside in areas where vans face restrictions or are practically unable to access due to geographical constraints. These customers are referred to as 'robot customers' and are exclusively served by robots. In contrast, another group of customers, known as 'van customers,' can be served either by vans or robots. Robots navigate tier-2 routes, which are inaccessible to vans, to cater to the needs of robot customers. This means that robots have the ability to serve all customers, whereas vans can only serve a subset of them due to the geographical locations of some customers. The authors explore hybrid pickup and delivery operations, where both vans and robots load customer goods from a depot and deliver them to the customer. Alternatively, they can pick up goods from a customer or supplier and deliver them to another customer or depot. The proposed problem is modeled as a mixed-integer program that takes into account factors such as time, freight, and energy. To address larger instances, an adaptive search algorithm is proposed as a solution. It is important to note that 2E-VRHPD differs from our proposed Robot-Assisted Delivery Problems (RADP) in a crucial aspect. In 2E-VRHPD, some customers (the robot customers) are exclusively served by robots, while in RADP, there are no customer restrictions that require exclusive robot service. Instead, local depots with positive demand are visited by trucks, which supply goods for onward delivery to customers by robots stored in the depot. One of the most recent

research studies involving the integration of autonomous robots with conventional trucks was conducted by [45]. In this study, the authors embraced the concept of a two-tier urban delivery network, a concept initially discussed in [46]. In this network, conventional delivery trucks operate at the first tier, while autonomous delivery robots function at the second tier. The operational assumption is that the trucks are responsible for transporting shipments from a central depot to every customer hub, typically during late-night or early-morning hours, with the intention of facilitating deliveries to customers by the robots. The study focused on determining the optimal number of robot hubs and the necessary quantity of robots required to effectively serve all customers. These findings were then compared with the results obtained from truck-based deliveries. The results from the study revealed that robot-based deliveries are not only more economical but can also lead to substantial cost savings, potentially reducing operational costs by up to 70%. In scenarios involving customer time windows, these savings can be even more significant, potentially reaching up to 90%. However, it's essential to acknowledge that due to the trucks exclusively operating at the first tier and not directly delivering to customers, a substantial fleet of robots is necessary in all the robot hubs. While this approach can be cost-effective, it also raises concerns about the initial investment required for acquiring and maintaining the robot fleet.

Existing studies involving robot-assisted delivery in the literature are basically of two setups: The first setup is where the truck loads customer shipments from the distribution center and transports them to local depots for onward delivery to customers by the localized robots. In the second setup, both the robots and the customer shipments are carried onboard the truck from the distribution center

to customer areas. The robots are then loaded with the shipments and launched for last-mile delivery. After delivery, the robots are picked up by the truck and return to the central depot. Both of these setups have their advantages. For example, using local depots could save truck capacity, be easier to manage, and less time-consuming for the truck driver than transporting packages as well as delivery robots. Also, transporting packages and robots using a truck allows simultaneous servicing of customer orders by the two vehicles, which in turn reduces the total delivery time. Considering the advantages of each of the two setups, the best option is to merge them together in the same problem. To the best of our knowledge, no study in the literature has merged the two scenarios together.

Chapter 3

Covering Salesman Problem With Nodes and Segments with Drones (CSPNS-D)

3.1 Introduction

E-retailing is one of the fastest growing sector in recent decades. It has started to play a vital role in more people's daily life especially under the COVID-19 pandemic. More efficient resolution for last mile delivery, i.e. the last stretch of the supply chain from the distribution center to the recipients preferred destination point [47] is therefore urged by sustainable development needs. As one of the most expensive, inefficient, and polluting parts of the supply chain, last mile accounts for 13.75% of total supply chain cost [48]. To cope with the ever growing pressure on last mile delivery, efforts have been put into two main directions, i.e. improving routing efficiency through optimization and developing more sustainable home

delivery techniques, such as using delivery/reception boxes or collection points [49–52], or involving unmanned delivery tools such as drones or robots in the delivery system [5, 7, 8, 23].

Unmanned drone, as a fast-moving and traffic-free delivery tool, attracts increasing attention in the last decade as a means for home delivery after been successfully used in military, communications and nature explorations [53–55]. Amazon prime air is one of the successful business cases. However, drones are restricted by their coverage distance and capacity. Therefore, in past literature they are mainly considered as supplement delivery tools that work jointly with trucks. This is normally referred to as drone-assisted delivery [5, 21–24, 26, 56].

Majority of research on drone-assisted delivery have assumed and restricted that drone should take off and land on customer locations while truck stops [5, 7, 23, 56]. However, technological advancement show that drone can be launched and land on a moving vehicle [57] [58]. These innovative contributions evoke the need for a new operational model. This study proposes a revised drone-assisted truck delivery system where drones can take-off and land on the truck while it is moving, without human interactions. An optimization model is designed to find the best truck and drone routes under this new operations scenario.

When drones are allowed to take-off and land on moving vehicles, amendments on traditional Traveling Salesman Problem (TSP) becomes essential. One potential way is to add dummy nodes on the TSP links to model the taking-off and landing points of drones, which is obviously expensive and troublesome. In this study, instead, we derive the maximum covering area of nodes and links that can be

serviced by a drone if the corresponding node/link is travelled by the truck so as to avoid examining where the drones should take-off or land in details. This would convert the whole problem into a so-called Covering Salesman Problem (CSP). The main contributions of this chapter are:

1. Unlike existing approaches for en-route operations that start from the optimal TSP, building the routes by inserting drone-service orders using a greedy approach and then adjust their taking-off and landing nodes on the link, this study proposes a novel approach right from the beginning that does not rely on the optimal TSP, by exploring the maximum cover range of every link, every link-node and two-links combination.

2. Three MILP models are developed out of the Covering Salesman Problem (CSP). Considering different sets of drone operations, these three models are compared in different application scenarios where managerial insights are drawn. With these models, optimal solutions for problems with up to 35 orders are achievable in 2 hours.

3. A Link(Node)-removal heuristic is developed to accelerate the solution process for larger instances, which are justified effective through numerical experiments.

3.2 Drone Operations

In this study, we consider a combined last-mile delivery problem with a standard delivery truck carrying a number of drones. Drones can take-off and land while the truck is moving, which means that the drone route can start and finish on edges

as well as nodes of the map. Let $V = \{1, 2, ..., n\}$ be the set of nodes, where $1 \in V$ denotes the depot and the rest the customer locations. Our aim is to minimize the total operational time to serve all customers, some of whom are visited by truck and the others by drones. We assume that drone has a higher speed than truck with a fixed ratio α ($\alpha > 1$).

Considering the fact that the drone can take-off and land on edges, any points on the truck route can be the starting and ending point of a drone operation. This adds extra difficulty to the formulation. In this study, however, we deploy the idea of Covering Salesman Problem (CSP) to introduce the covering range of each node and link of the truck route, which is defined as the area that can be serviced by drone without extra truck waiting time. Suppose all the drones are pre-settled with the information of the customer it is meant to service before loading to the truck, the drone can automatically take-off and land on the truck while the truck is executing its own route without interruption. This enables the truck to move without worrying about the drone throughout the service horizon. Therefore, our aim converts to finding a least-time truck route, where all customer nodes are inside the covering range of the nodes and links of the truck route.

Formulating covering areas is a key step in designing valid drone operations. Drone operations are constructed based on two parameters, i.e., the maximum flight distance of drone d_{max} and the average speed ratio of drone compared to truck α. Since drones are battery powered, the maximum flight distance d_{max} is defined as the farthest range that drone can reach without charging its battery.

Figure 3.1 shows a graphical example of a solution of CSPNS-D. The truck tour (A, B, C, D, A) is represented as red nodes with arrow, while drone-served customers

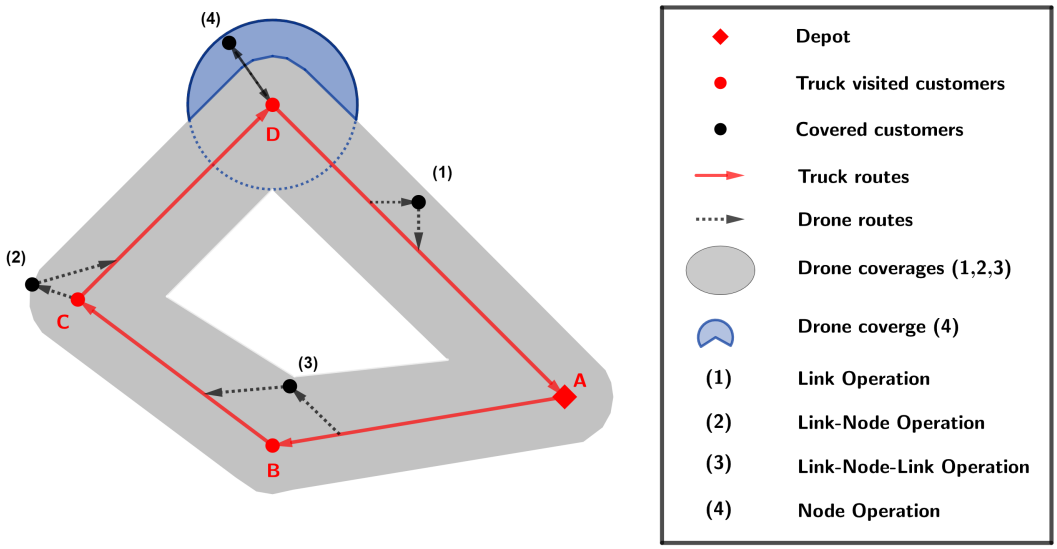


Figure 3.1. Illustration of solution and combined covering area of CSPNS-D

and their operations, (1), (2), (3) and (4), are depicted as black. Shaded areas; grey and blue, indicate two types of drone covering areas. The grey shadow areas represent combined drone coverages for the Link / Link-Node / Link-Node-Link operations which can be performed during truck driving. As the three drone operations do not require extra time for deliveries, the covering areas are generated followed by within certain distances from the truck route. On the other hand, the Node operation designed to cover further located customers requires waiting time for the truck, which can be performed not on the entire truck route, but only on the truck visited nodes. Drone coverage can be extended at the truck visited node by the Node operation. see the blue shaded area in the figure.

In the following sub-sections we present the four operations together with their covering areas in details. Then, in Section 3.3, three MILPs are formulated by combining Link-Node-Link Operation and Node Operation with the other two considered as basic operations (CSPNS-D-3), or combining the Link-Node-Link

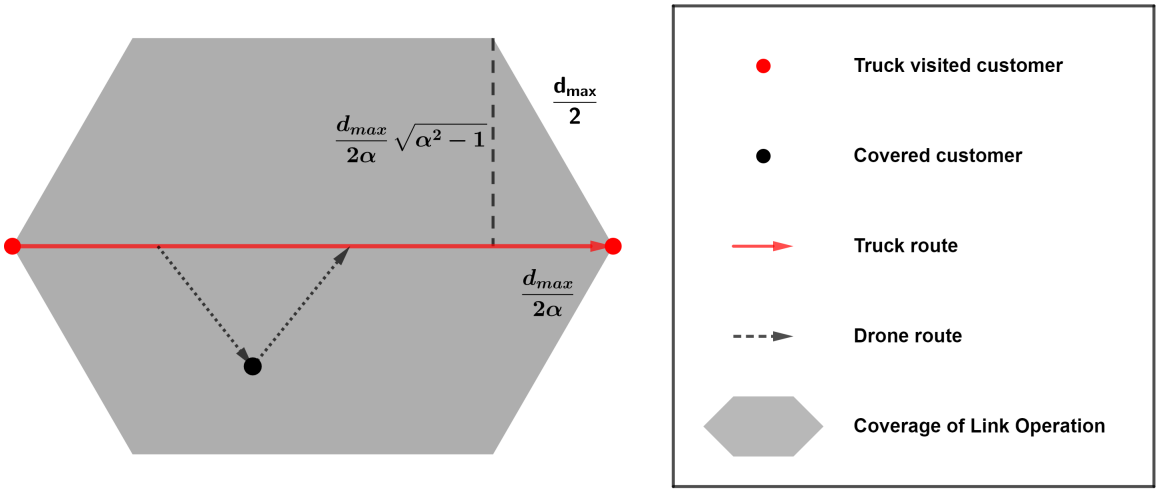


Figure 3.2. Link Operation and its covering area

Operation with the basic Operations (CSPNS-D-2) or only the basic Operations (CSPNS-D-1).

3.2.1 Link Operation

Link Operation is defined as the drone operation which has both its taking-off and landing nodes on an edge of the truck route. While the truck drives from one customer to another, multiple drones can individually perform deliveries within the hexagonal area represented in Figure 3.2.

In order to ensure synchronisation with the truck, at most half of the battery can be consumed to reach a customer, while the other half should be kept to come back to the truck. Therefore the drone routes in Link Operation form isosceles triangles together with the truck route. Figure 3.2 shows the formula for the maximum perpendicular distance r that a drone can cover from the truck driving route, calculated from d_{max} and α. Customer locations closer than $\frac{d_{max}}{2\alpha}\sqrt{\alpha^2-1}$ from

the truck route, excluding the first and the last $\frac{d_{max}}{2\alpha}$ length towards the ends, can be served by Link Operation without incurring extra waiting time of the truck.

3.2.2 Link-Node Operation

The Link-Node Operation is designed to fill in the gap areas which are hardly covered by Link Operations near the truck visited nodes. This operation works in node and link pairs, where a drone takes-off at a node (link) and lands on a link (node). Remember that the maximum flying distance of a drone is defined as d_{max}. In this operation we need to choose the departure/landing point on the link to ensure that the total flying distance of a drone service is not violating this limit.

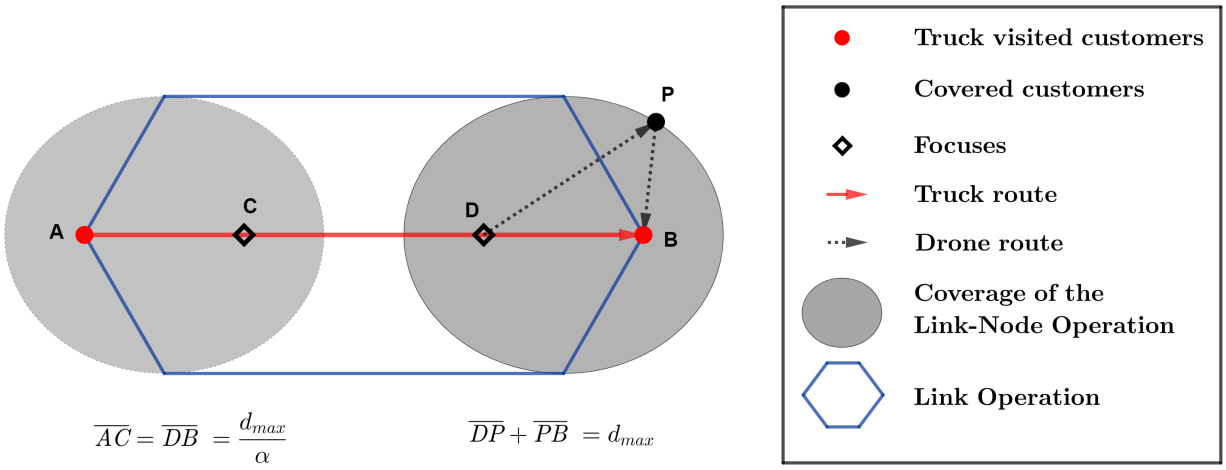


Figure 3.3. Link-Node Operation and its covering area

As shown in Figure 3.3, we consider truck route (A, B). Let P be a customer node on the boundary of the covering range of the Link-Node Operation around node B. Let D be the drone taking-off point on link (A, B), we must have $\overline{DP} + \overline{PB} \leq d_{max}$. Suppose we consider points D and B as two foci, it is not hard to see that the boundary point P form an ellipse (the shaded area in Figure 3.3). Any points within this area can be serviced at no extra waiting time by Link-Node Operation,

with a taking-off (landing) point on the link that is $(\overline{DP} + \overline{PB})/\alpha$ away from the end nodes of the link.

Proposition 1: The largest coverage area of the Link-Node Operation around node B is achieved by an ellipse with foci at B and D where $\overline{BD} = d_{max}/\alpha$.

Proof: To begin, any foci D that satisfies $\overline{BD} > d_{max}/\alpha$ violates the drone's maximum flying distance. Secondly, for foci D closer to B, the drone's flying distance is reduced to $\alpha\overline{BD} < d_{max}$ for synchronization with the truck. This reduction also diminishes the coverage of the ellipse. For a more detailed proof of the second point, please refer to **Appendix A**.

Like the Link Operation, the services inside the Link-Node Operation covering area do not require any extra time in addition to truck driving time. It follows that our objective is to minimize the driving times of truck. However, the extended area of the Link-Node Operation is relatively small and is highly depending on the value of α. To enlarge the covering area further, we also propose the Link-Node-Link Operation in next sub-section.

3.2.3 Link-Node-Link Operation

In addition to the single-link operations, drones can also launch and land on different truck edges. In this subsection, we consider drone operations ranging through two edges that are connected to the same node. In this case the angle between the two edges matters.

Let us consider the example shown in Figure 3.4. Let AB, BC be the two links under consideration which connect at customer node B, and θ $(0 \leq \theta \leq \pi)$ be the (non-reflex) angle between them. Our aim here is to identify the maximum cover

range of all feasible drone operations that have one of the launch and land point on link AB and the other on link BC or vice versa. Let P be a customer node on the boundary of the drone cover range, D and E be the taking-off and landing points on each link, we must have $\overline{DP} + \overline{PE} = d_{max}$ and $\overline{DB} + \overline{BE} = \frac{1}{\alpha}(\overline{DP} + \overline{PE}) = \frac{d_{max}}{\alpha}$. To obtain the maximum cover range we must have $\overline{DB} = \overline{BE}$, as otherwise the cover range will lean to one side of the link and overlap with the cover range of (single) Link Operation.

As discussed before, D and E can be seen as the foci of the ellipse that defines the cover range, with the length of the semi-major axis equal to $d_{max}/2$, distance between two foci equal to $\frac{d_{max}}{\alpha} sin(\frac{\theta}{2})$. The cover range is then defined as the shaded area in Figure 3.4

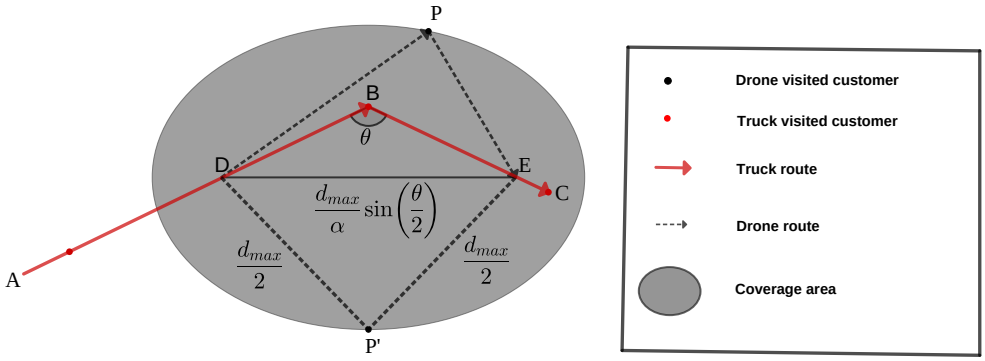


Figure 3.4. Link-Node-Link Operation and its covering area

Proposition 2: The Link-Node-Link Operation extends the coverage area of Link-Node Operation for all $\alpha > 1$. For formal proof please refer to **Appendix B**.

3.2.4 Node Operation

In this section we propose the Node Operation, which allows the truck to wait at the customer node while the drone set off to perform its own task. Unlike all the other operations proposed in this work, the Node Operation does incur waiting time of the truck. The reason we suggest the Node Operation at extra cost (waiting time) lies in the fact that, this operation enlarges the drone covering area and larger covering area at each node can potentially reduce the number of nodes the truck need to visit, which may ultimately reduce total operational time. This is justified by the superiority of CSPNS-D-3 (which allows Node Operation) over the other two models (which don't). Please refer to Section 3.4 for detailed numerical experiments and results.

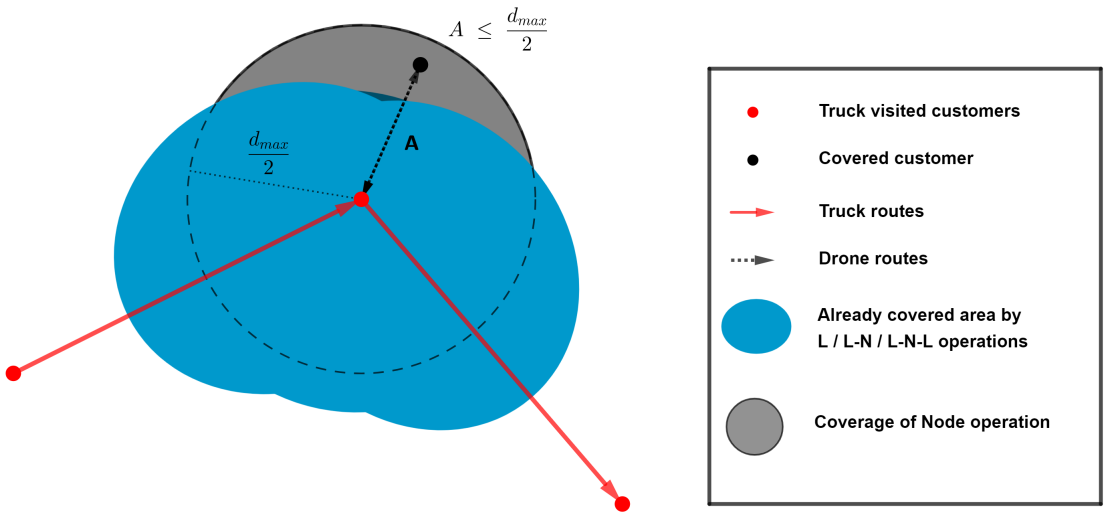


Figure 3.5. Node Operation and its covering area

The covering area of the Node Operation forms a circle with radius $\frac{d_{max}}{2}$ to guarantee returning of drones. Note that the waiting time should only incur when

a customer node falls into the grey area of Figure 3.5, as otherwise the node can be covered by other means without extra cost.

3.3 Mathematical Formulation of CSPNS-D Models

In this section, we formulate three mixed integer linear programming models for the CSPNS-D by allowing and disabling the Link-Node-Link and Node Operations at extra costs (waiting time) as discussed in Section 3.2.4. The CSPNS-D-1 minimizes the truck travelling time of servicing all customers with Link Operation and Link-Node Operation; CSPNS-D-2 does it with Link Operation, Link-Node Operation and Link-Node-Link Operation; CSPNS-D-3, on the other hand, minimize the on-duty time of the truck (including both driving and waiting time) of servicing all customers with Link Operation, Link-Node Operation, Link-Node-Link Operation and Node Operation.

3.3.1 Assumptions

- Multiple drones are carried by the truck and they can take-off and land on the moving truck independently, automatically and instantly.
- The velocities of drones and truck are constant. i.e., the truck is travelling on traffic-free roads and drones' flying speed are not influenced by the weight of parcel they are carrying or other factors such as wind speed.
- Roads can be well approximated by straight line.
- Drone starts with fully charged battery, i.e., d_{max} is a given constant.

- All parcels can be delivered by drones. Multiple demands of parcels for one customer can be serviced by several drones.
- The service times for the hand-over of parcels are negligible as assumed by [31].
- There's no limit on the number of drones that can be carried by the truck.

3.3.2 Notations

Sets and indices

$V = \{1, 2, ..., n\}$	Set of nodes, 1 denotes the depot and the rest the customer nodes.
$i, j, h, k, l \in V \setminus \{1\}$	Index of customer nodes.

Parameters

$d_{ij}, i, j \in V$	Euclidean distance between i and j.
$a_{ijk}, i, j, k \in V$	Binary parameter; equals to one if and only if customer i can be covered by Link Operation or Link-Node Operation of link (j,k).
$b_{ikjl}, i, k, j, l \in V$	Binary parameter; equals to one if and only if customer i can be covered by Link-Node-Link Operation of links (k, j) and (j,l).
$c_{ij}, i, j \in V$	Binary parameter; equals to one if and only if $d_{ij} \leq \frac{d_{max}}{2}$, i.e., customer i can be covered by Node Operation from j.

Decision variables

$x_{ij}, i \neq j \in V$	Binary variable; equals to one if and only if link (i, j) is travelled by truck.
$y_i, i \in V$	Binary variable; equals to one if and only if customer i is serviced by truck.
$u_{ij}, i \neq j \in V$	Non-negative continuous variable; single commodity flow variable for sub-tour elimination constraints.
$\delta_{kjl}, j \neq k \neq l \in V$	Binary variable (CSPNS-D-2 and CSPNS-D-3 only); equal to one, if and only if both links (k, j) and (j, l) are travelled by truck.
$\eta_{ih}, i \neq h \in V$	Binary variable (CSPNS-D-3 only); equal to one, if and only if customer i is covered by the Node Operation from node h.
$v_h, h \in V$	Non-negative continuous variable (CSPNS-D-3 only); the longest waiting time at node h if any Node Operations are initiated at this node.

3.3.3 The CSPNS-D Models

Three models are developed in this work to capture different operations settings.

CSPNS-D-1 Only Link Operation and Link-Node Operation are considered. This is a standard Covering Salesman Problem with Node and Segments (CSPNS) on drone assisted delivery application.

CSPNS-D-2 Link-Node-Link Operation is considered together with Link Operation and Link-Node Operation. This is an extension of the CSPNS by adding cover range of combined links.

CSPNS-D-3 Node Operation (at extra waiting time) is considered together with Link Operation, Link-Node Operation and Link-Node-Link Operation. This is an extension of the CSPNS by adding cover range of combined links and allowing enlarged cover range at extra costs.

3.3.3.1 CSPNS-D-1

When we allow the Link Operation and the Link-Node Operation as defined in Section 3.2.1 and 3.2.2 respectively. This is a standard Covering Salesman Problem with Nodes and Segment.

$$\min \sum_{i \in V} \sum_{j \in V \setminus \{i\}} d_{ij} x_{ij} \tag{3.1}$$

Subject to :

$$y_1 = 1, \tag{3.2}$$

$$\sum_{h \in V \setminus \{i\}} x_{hi} = y_i, \qquad \forall i \in V \tag{3.3}$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} = y_i, \qquad \forall i \in V \tag{3.4}$$

$$\sum_{j \in V} \sum_{k \in V \setminus \{j\}} a_{ijk} x_{jk} \geq 1, \qquad \forall i \in V \tag{3.5}$$

$$\sum_{j \in V \setminus \{i\}} u_{ij} - \sum_{h \in V \setminus \{i\}} u_{hi} = y_i, \qquad \forall i \in V \setminus \{1\} \tag{3.6}$$

$$u_{ij} \leq (n-1) x_{ij}, \qquad \forall i \in V, \forall j \in V \setminus \{i\} \tag{3.7}$$

$$\sum_{j \in V \setminus \{1\}} u_{1j} = 0, \tag{3.8}$$

$$\sum_{h \in V \setminus \{1\}} u_{h1} = \sum_{k \in V} y_k - 1, \tag{3.9}$$

$$u_{ij} \geq 0, \qquad \forall i \in V, \forall j \in V \setminus \{i\} \tag{3.10}$$

$$y_i \in \{0,1\}, \qquad \forall i \in V \tag{3.11}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in V, \forall j \in V \setminus \{i\} \tag{3.12}$$

Equation (3.1) is the objective function which minimizes the total truck driving time, as the the Link Operation and Link-Node Operation do not interrupt truck driving. Constraint (3.2) ensures that the truck route starts from the depot. Constraints (3.3) and (3.4) are the network flow constraints, where if node i is visited by the truck there must be exactly two links connected to it. Constraint (3.5) ensures that every node i is covered by the truck route with either Link Operation or Link-Node Operation. Constraints (3.6, 3.7, 3.8, 3.9 and 3.10) are sub-tour elimination constraints revised from Gavish and Graves [59] (single commodity flow formulation). Finally, constraints (3.11) and (3.12) define the variables as binary value. Continuous variable u_{ij} represents the amount of flow from node i to node j.

3.3.3.2 CSPNS-D-2

When we allow the Link-Node-Link Operation as defined in Section 3.2.3, customers located in the ellipse around a truck visited node can be serviced without additional waiting time. To enable this operation, additional decision variables. i.e., $\delta_{kjl}, j \neq k \neq l \in V$, are introduced to indicate the availability of Link-Node-Link Operation.

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} d_{ij} x_{ij} \tag{3.13}$$

Subject to :

$$(3.2) - (3.4) \text{ and } (3.6) - (3.12), \tag{3.14}$$

$$\sum_{j \in V} \sum_{k \in V \setminus \{j\}} a_{ijk} x_{jk} + \sum_{j \in V} \sum_{k \in V \setminus \{j\}} \sum_{l \in V \setminus \{j,k\}} b_{ikjl} \delta_{kjl} \geq 1, \qquad \forall i \in V \tag{3.15}$$

$$\delta_{kjl} \leq x_{kj}, \qquad \forall j \neq k \neq l \in V \tag{3.16}$$

$$\delta_{kjl} \leq x_{jl}, \qquad \forall j \neq k \neq l \in V \tag{3.17}$$

$$\delta_{kjl} \in \{0, 1\}, \qquad \forall j \neq k \neq l \in V \tag{3.18}$$

CSPNS-D-2 holds most constraints of **CSPNS-D-1** except constraint (3.5), which is replaced by constraint (3.15) as here we allow Link-Node-Link Operation as well as others to service each node. Constraints (3.16) and (3.17) ensure that a Link-Node-Link Operation can be performed only when both links involved in the operation are travelled by the truck. Constraints (3.18) set the domain for the new decision variable.

3.3.3.3 CSPNS-D-3

When we allow the Node Operation as defined in Section 3.2.4, customers located in the circle around a truck visited node can be serviced with additional truck waiting time. To enable this operation, additional decision variables. i.e., $\eta_{ih}, i \neq h \in V$ and

$v_h, h \in V$ are introduced to indicate the Node Operation and to calculate the extra waiting time of the truck.

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} d_{ij} x_{ij} \quad + \sum_{h \in V} v_h \tag{3.19}$$

Subject to:

$$(3.2) - (3.4) \text{ and } (3.6) - (3.12) \text{ and } (3.16) - (3.18) \tag{3.20}$$

$$\sum_{j \in V} \sum_{k \in V \setminus \{j\}} a_{ijk} x_{jk} + \sum_{j \in V} \sum_{k \in V \setminus \{j\}} \sum_{l \in V \setminus \{j,k\}} b_{ikjl} \delta_{kjl} + \sum_{h \in V} \eta_{ih} \geq 1, \qquad \forall i \in V \tag{3.21}$$

$$\eta_{ih} \leq c_{ih} y_h, \qquad \forall i, h \in V \tag{3.22}$$

$$v_h \geq \frac{2 d_{ih}}{\alpha} \eta_{ih}, \qquad \forall i, h \in V \setminus \{1\} \tag{3.23}$$

$$v_h \geq 0, \qquad \forall h \in V \tag{3.24}$$

$$\eta_{ih} \in \{0, 1\}, \qquad \forall i \in V, \forall h \in V \setminus \{i\} \tag{3.25}$$

CSPNS-D-3 holds most constraints of **CSPNS-D-2** except constraint (3.15), which is replaced by constraint (3.21) as here we allow Node Operation as well as others to service each node. Constraint (3.22) ensures that node i can only be covered by node h if it is in the cover range of node h and node h is visited by truck. Constraint (3.23) calculates the maximum waiting time at node h suppose some Node Operations from node h are performed. Constraints (3.24) and (3.25) set the domain for the new decision variables.

3.4 Numerical Results

This section presents the numerical results of the CSPNS-D and their corresponding TSP solutions. The models are coded in MATLAB R2020a and solved by CPLEX 12.10.0 on a CPU with an Intel(R)Core(TM)i5-7300U processor.

3.4.1 TSP and CSPNS-D Results on Random Instances

We conducted numerical experiments on a set of 10 randomly generated instances of varying sizes, up to a maximum of 40 orders. The drone covering distance was consistently set to 30 km across all instances, aligning with the current flying capability of drones in use [60]. The drone speed was selected as 60 km/h, while the truck speed was set at 40 km/h. For each instance, we determined the optimal solution of CSPNS-D models and compared them with the optimal TSP routes. The results are summarized in Table 3.1

It can be seen that CSPNS-D models significantly improve delivery efficiency in all test examples. Operated jointly with drones, truck travelling times reduces by 17%, 20.11% and 20.24% (experiment 40 not included because of large gaps for CSPNS-D-2 and CSPNS-D-3) with CSPNS-D-1, CSPNS-D-2 and CSPNS-D-3 respectively. Except for 40 orders, CSPNSD-3 always performs better than CSPNSD-2, which is better or the same as CSPNSD-1. This is in line with our expectation as we consider more and more operation types in the model design. Due to larger search space of CSPNSD-3 and CSPNSD-2 on 40 orders, worse solutions are found with these models compared to CSPND-1 within the limited computational time allowed (7200 seconds).

Optimality gaps are reported in the last three columns for all CSPNS-D models. CPLEX, like many other solvers, calculates these gaps to assess the quality of the solutions it produces. The gaps measure how far the current best solution is from the optimal solution and are defined as:

$$\frac{|\hat{f} - \overline{f}|}{|\overline{f}| + 10^{-10}} \tag{3.26}$$

Where:

- $\hat{f}$ is the objective value of the incumbent (best known feasible solution).
- $\overline{f}$ is the best known bound for the objective function.

CPLEX converts it to a percentage in the node log. By monitoring the optimality gap, users can understand how close the solver is to finding the optimal solution. A small gap indicates that the current solution is likely close to optimal, while a larger gap suggests that the solver may need more time to improve the solution further. For optimum solutions, gaps of 0.00% are reported, indicating that the incumbent solution and the best bound are the same. A gap of 2.48% for the CSPNS-D-2 solution for 30 orders in Table 3.1 indicates that the best bound is 2.48% smaller than the incumbent solution (**247.64 sec**). The gaps are high for the 40 orders example, i.e., 50.57% for CSPNS-D-2 and 59.27% for CSPNS-D-3, indicating that CPLEX needs more time to further improve these solutions. For this reason, heuristic acceleration approaches are presented in Chapter 5 to make larger size instances more tractable.

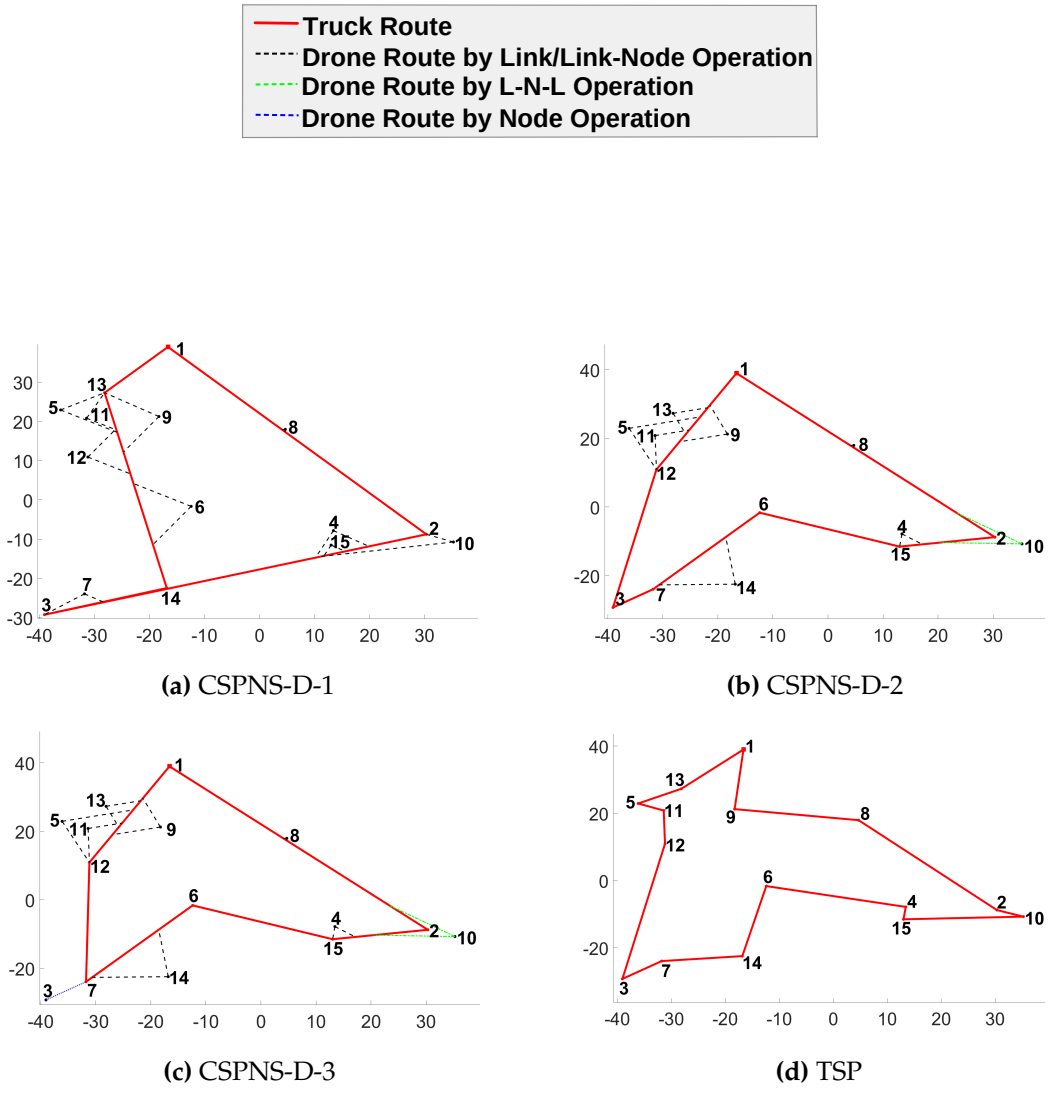
In more details, two sets of sample solutions are presented graphically in Figures 3.6 & 3.7. Figure 3.6 is for instance with 15 orders, where sub-figures (a),

Table 3.1: TSP and CSPNS-D Solutions of randomly generated instances with fixed map range and drone flying distance

No. of	TSP Deliv-ery	Delivery Time			Savings Over TSP			CPU Time (sec)			No. of orders served by drone			Optimality gap		
Orders	time (s)	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3
10	208.17	187.56	187.56	187.56	9.90	9.90	9.90	0.08	0.19	0.16	5	5	5	0.00%	0.00%	0.00%
15	262.68	229.80	222.86	219.80	12.52	15.16	16.32	0.41	1.13	2.11	10	8	9	0.00%	0.00%	0.00%
18	269.29	244.06	227.24	227.24	9.37	15.62	15.62	1.63	2.36	2.80	11	11	11	0.00%	0.00%	0.00%
20	306.37	260.95	260.95	260.95	14.83	14.83	14.83	1.72	3.16	4.28	13	13	13	0.00%	0.00%	0.00%
22	320.88	255.45	251.99	251.99	20.39	21.47	21.47	1.02	3.75	5.70	16	15	15	0.00%	0.00%	0.00%
25	361.31	291.16	291.16	291.16	19.41	19.41	19.41	6.11	253.39	267.73	17	17	17	0.00%	0.00%	0.00%
28	358.18	286.30	260.40	260.40	20.07	27.30	27.30	11.08	2457.10	2041.80	19	14	14	0.00%	0.00%	0.00%
30	337.97	261.55	247.64	247.64	22.61	26.73	26.73	17.86	7200.00	5191.60	21	20	20	0.00%	2.48%	0.00%
35	403.78	307.29	280.44	280.44	23.90	30.55	30.55	8.58	3853.10	4906.70	25	26	26	0.00%	0.00%	0.00%
40	367.97	277.56	321.29	332.18	24.57	12.68	9.73	55.16	7200.00	7200.00	31	29	23	0.00%	50.57%	59.22%
Average Savings Over TSP					**17.76**	**20.11**	**20.24**									

(b), (c) and (d) are for CSPNS-D-1, CSPNS-D-2,CSPNS-D-3 and TSP respectively.

In CSPNS-D-1, 5 out of 15 orders are serviced by the truck; in CSPNS-D-2 and



(a) CSPNS-D-1

(b) CSPNS-D-2

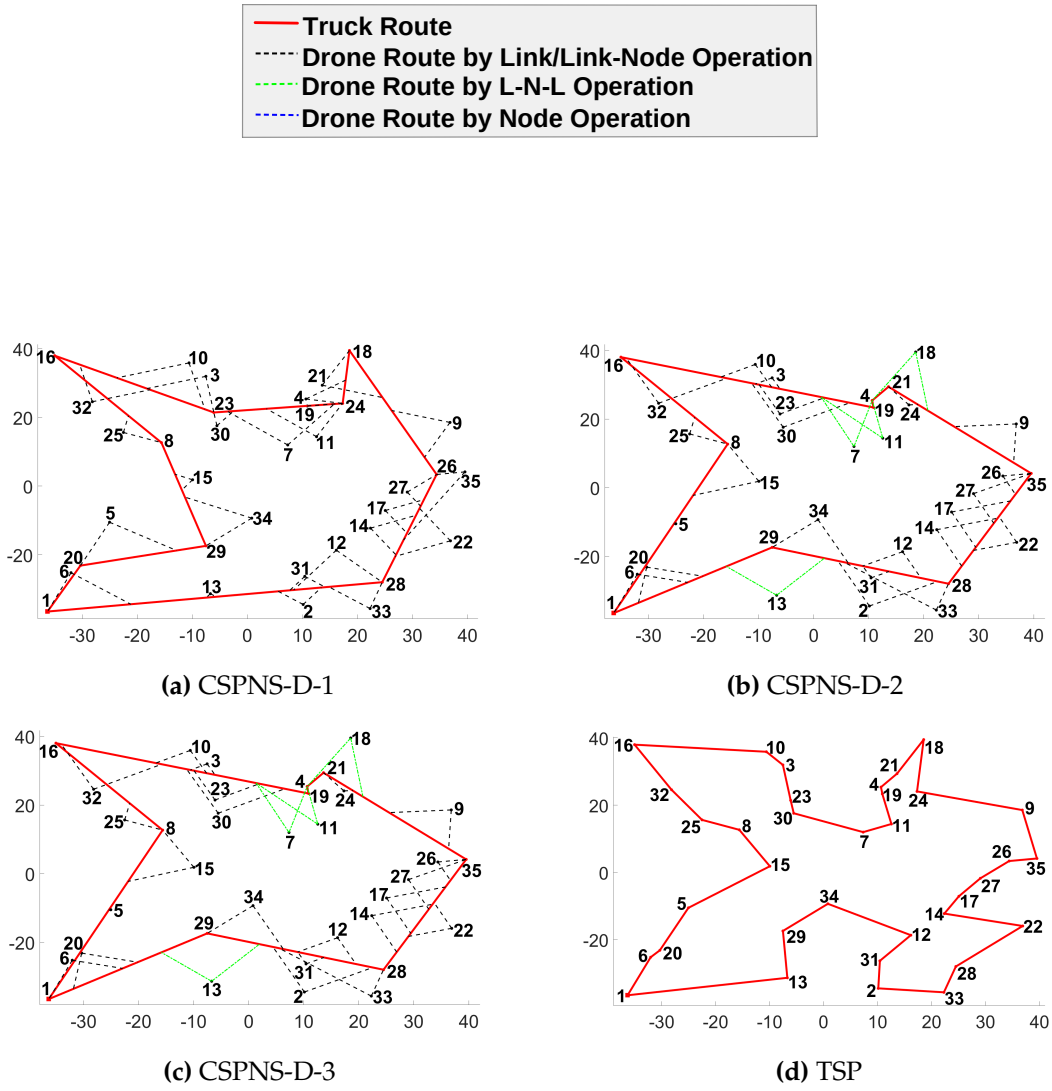
(c) CSPNS-D-3

(d) TSP

Figure 3.6. CPLEX solution of CSPNS-Ds and TSP with 15 Random orders

CSPNS-D-3, this number increases to 7 and 6, respectively. The savings over TSP are 12.52%, 15.16%, and 16.32% for CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3, as shown in Table 3.1. As depicted in Figure 3.6b, when considering the Link-Node-Link operation alongside the Link Operation and Link-Node Operation, node 10 can be covered by a Link-Node-Link Operation from links (1,2) and (2,15).

Consequently, the truck route to node 3 and then to node 14 (although it appears in Figure 3.6a as if node 3 is serviced by the truck with a return trip from node 14, the actual route is (2,3), (3,14), and (4,13))—the most inefficient segment of the CSPNS-D-1 route can be reshaped, saving the total travel time by approximately 3%. Figure 3.6c illustrates the truck route of CSPNS-D-3, where node operations are allowed with extra waiting time. Under this setting, node 3 can now be covered by a node operation from node 7, saving the time it takes for the truck to travel from node 7 to 3 and then to node 12. Although the truck's waiting time at node 7 must be considered, it is shorter than the truck's driving time, as the drone travels at a higher speed. This further reduces the total service time by 1.16%.



(a) CSPNS-D-1 **(b)** CSPNS-D-2

(c) CSPNS-D-3 **(d)** TSP

Figure 3.7. CPLEX solution of CSPNS-Ds and TSP with 35 Random orders

In Figure 3.7, a similar example is presented, associated with the instance containing 35 orders. Both CSPNS-D-2 and CSPNS-D-3 demonstrate more significant improvements over the TSP solution (by 30%) compared to CSPNS-D-1 (by 23%). It is worth noting that in this specific case, CSPNS-D-2 and CSPNS-D-3 yield identical solutions. This occurs because all nodes that can potentially be covered by Node Operation (e.g., node 7 from node 4 and node 18 from node 21) can also be covered by Link-Node-Link Operation with the given route. Given that no

extra waiting time is incurred by Link-Node-Link Operation, the model selects it over Node Operation. Another noteworthy point is the visiting sequence of nodes. For instance, in the CSPNS-D-1 route, if all routes displayed are traveled anti-clockwise, node 29 is visited later in the route. In contrast, in CSPNS-D-2 and CSPNS-D-3 routes, it is visited right at the beginning. Traditionally, drone delivery heuristics construct drone routes based on optimal TSP and follow the initial visiting sequence of nodes suggested by the optimal TSP. However, this observation highlights the possibility of missing better decisions using this type of heuristic. Instead, our model does not rely on TSP solutions but employs optimization models to find real optimal drone and truck routes simultaneously.

3.4.2 Numerical Results on Benchmark Instances

In this section, we test the performance our CSPNS-D models on benchmark instances obtained from [61]. We consider two groups of test with 10 and 20 orders, each consisting of 30 instances. As stated in the benchmark data, customers locations follow uniform, single center and double centre distributions. Following the work of [32], maps for 10 and 20 customers are re-scaled by 15% and 30%, respectively, to make it compatible for drone delivery tests. We consider the use of multiple drones, with drone covering distance 30km, truck speed of 40kmh and drone speed 60kmh throughout the experiment. Summary of results are presented in Table 3.2. For detailed results for every instance please refer to Appendix C.

Table 3.2: Summary of Obtained Results for Benchmark instances

ST/SD, km/h	No. of Orders	Average savings (%)			Standard Deviation		
		CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3
40/60	10	21.55	51.55	52.61	31.55	35.56	33.95
40/60	20	24.26	33.30	33.63	63.45	68.00	67.38

For 10 orders, CSPNS-D-3 obtains better solutions in 5 out of 30 benchmark instances than CSPNS-D-2 and CSPNS-D-1. In the remaining 25 instances, CSPNS-D-3 gives the same solution as CSPNS-D-2, which are better than CSPNS-D-1. Average savings over TSP for CSPNS-D-1, CSPNS-D-2 and CSPNS-D-3 are 21.55%, 51.55% and 52.61% respectively. For 20 orders, CSPNS-D-3 performs better than both CSPNS-D-2 and CSPNS-D-1 on 3 out of 30 instances, while they are the same with CSPNS-D-2 on the remaining 27 instances, same with CSPNS-D-1 on 1 instance, but better on 26 instances. Average savings over TSP are 24.26%, 33.30% and 33.63% for CSPNS-D-1, CSPNS-D-2 and CSPNS-D-3 respectively. Comparing with [32] who reported 10.5% and 10.6% savings on TSP with their Greedy and Enroute heuristics respectively, our models' savings is obviously higher. However, we consider the usage of multiple drones while they used only one. On a related note, their heuristic approach for en-route operations is not easily extendable to cover multiple drone cases. The results presented here highlight the significance of CSPNS-D-3 and CSPNS-D-2 in applications, particularly if they can be solved more efficiently for larger instances. In the next section, we will explore potential approaches to expedite the solution process for all CSPNS-D models.

3.4.3 Link Removal Heuristic for Larger Instances

In this section, we explore a potential approach to expedite CSPNS-D solutions, aiming to solve larger instances within a reasonable time frame. The approach is based on removing links that are highly unlikely to be chosen as part of the truck route, thereby minimizing the number of possible choices for branching in the MILP solution process. If successfully implemented, this link-removal strategy

should effectively reduce computational time without significantly compromising solution quality.

Examining numerous optimal solutions and their routing structures, the following criteria have been identified for link removal:

1. Any link that is shorter than half the drone covering distance ($\frac{d_{max}}{2}$) is removed as in most cases the relevant nodes can be covered by, at least, Node Operation.

2. Any link that is longer than twice the average distance between orders is removed provided that its removal does not affect the connectivity of the network considering the aim of cost/time minimization.

3. Any link covering very few nodes, say less than or equal to 2, is removed as to improve efficiency we should select links that covers as many nodes as possible.

4. If a node is located in the middle of some other nodes, the node will not be serviced by a Node Operation (CSPNS-D-3 only), so we remove the associated variables like η and v for this node.

3.4.3.1 Performance of the Link Removal Heuristic

To assess the performance of the heuristic, we employed benchmark instances comprising 20 orders, with results summarized in Table 3.3. The optimal (MILP) solutions for CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3 are presented in columns 2, 3, and 4 respectively, while their heuristic solutions are displayed in columns 5, 6, and 7 correspondingly. The gaps, which measure the disparity between the

heuristic and MILP solutions, are depicted in columns 8, 9, and 10 for CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3 respectively. Unlike the sub-optimality gaps presented in Table 3.1, which were obtained from the CPLEX solver, these gaps are calculated manually using the formula:

$$\frac{(\theta_{LR} - \theta_{Opt})}{\theta_{Opt}} * 100 \tag{3.27}$$

Where:

- θ_{Opt} is the objective value of the optimum (MILP) solution.
- θ_{LR} is the objective value of the Link-Removal Heuristic.

Table 3.3: Link-Removal Heuristic Solution on Benchmark Instances of Small sizes

Instance	MILP Solution (sec.)			Link-Removal Solution (sec.)			Opt. Gap (%) Btw MILP and Heuristic		
	CSPND-1	CSPND-2	CSPND-3	CSPND-1	CSPND-2	CSPND-3	CSPND-1	CSPND-2	CSPND-3
1	66.06	44.74	44.74	66.06	66.06	63.57	0.00	47.65	42.09
2	76.48	57.12	57.12	76.48	73.25	73.25	0.00	28.24	28.24
3	69.93	64.36	64.36	69.93	69.93	69.93	0.00	8.65	8.65
4	71.22	60.24	60.24	96.85	66.17	66.17	35.99	9.84	9.84
5	81.54	62.11	62.11	81.54	81.54	70.29	0.00	31.28	13.17
6	69.23	69.23	69.23	89.96	89.96	89.96	29.94	29.94	29.94
7	79.6	68.97	68.97	81.14	77.24	77.24	1.93	11.99	11.99
8	82.54	59.4	59.4	82.82	82.54	82.54	0.34	38.96	38.96
9	68.76	51.47	51.47	68.76	68.76	60.96	0.00	33.59	18.44
10	83.58	65.79	65.79	86.82	83.58	83.58	3.88	27.04	27.04
11	132.13	124.73	124.73	135.64	130.71	132.13	2.66	4.79	5.93
12	129.27	98.4	98.4	129.83	129.27	114.3	0.43	31.37	16.16
13	167.71	164.31	164.31	199.26	180.65	177.68	18.81	9.94	8.14
14	152.17	150.11	150.11	190.68	152.17	152.17	25.31	1.37	1.37
15	152.95	136.51	130.36	189.32	153.01	149.49	23.78	12.09	9.51
16	108.65	92.38	92.38	128.23	109.57	109.57	18.02	18.61	18.61
17	135.51	118.5	118.5	145.22	136.46	135.73	7.17	15.16	14.54
18	144.17	126.24	126.24	150.32	145.96	143.28	4.27	15.62	13.50
19	177.83	177.57	177.57	247.47	182.31	177.83	39.16	2.67	0.15
20	115.4	91.19	91.19	148.32	115.4	107.35	28.53	26.55	17.72

The outcomes indicate that the heuristic is capable of achieving optimal or near-optimal solutions, particularly for the CSPNS-D-1 model. However, near-optimal or occasionally inferior solutions are obtained for CSPNS-D-2 and CSPNS-D-3, especially in cases where the optimal solution involves drones being launched and retrieved on two different links (Link-Node-Link).

Table 3.4: Link Removal Method (LRM) of Instances with 50 orders

Instance	TSP Delivery	Delivery Time			Savings Over TSP			CPU Time (sec)			Optimality gap		
	time (s)	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3
U1	282.84	169.18	*	*	40.19	*	*	299.41	7200.00	7200.00	0.00%	-	-
U1_LRM		178.32	259.90	194.31	36.95	8.11	31.30	16.72	7200.00	7200.00	0.00%	74.04%	44.80%
U2	308.48	179.95	*	*	41.67	*	*	847.67	7200.00	7200.00	0.00%	-	-
U2_LRM		179.95	220.28	179.98	41.67	28.59	41.66	31.72	7200.00	7200.00	0.00%	67.54%	36.59%
U3	304.12	159.98	*	*	47.40	*	*	213.39	7200.00	7200.00	0.00%	-	-
U3_LRM		160.20	171.49	159.98	47.32	43.61	47.40	7.58	7200.00	7200.00	0.00%	55.73%	19.30%
D1	716.34	585.74	*	*	18.23	*	*	1510.30	7200.00	7200.00	0.00%	-	-
D1_LRM		632.94	616.35	595.90	11.64	13.96	16.81	35.36	7200.00	7200.00	0.00%	16.69%	11.67%
D2	628.99	508.44	*	*	19.17	*	*	1052.80	7200.00	7200.00	0.00%	-	-
D2_LRM		521.39	536.02	508.44	17.11	19.17	19.17	205.45	7200.00	7200.00	0.00%	27.92%	6.91%
D3	590.72	462.46	*	*	21.71	*	*	7200.00	7200.00	7200.00	8.29%	-	-
D3_LRM		484.39	562.63	451.79	18.00	4.76	23.52	460.56	7200.00	7200.00	0.00%	42.14%	13.50%
D4	568.28	458.23	*	*	19.37	*	*	7200.00	7200.00	7200.00	9.83%	-	-
D4_LRM		458.23	514.24	457.38	19.37	9.51	19.52	106.13	7200.00	7200.00	0.00%	34.55%	17.85%
S1	349.40	221.74	*	*	36.54	*	*	14.50	7200.00	7200.00	0.00%	-	-
S1_LRM		236.10	240.26	223.60	32.43	31.24	36.00	5.56	7200.00	7200.00	0.00%	41.28%	7.64%
S2	404.18	289.55	*	*	28.36	*	*	51.97	7200.00	7200.00	0.00%	-	-
S2_LRM		293.18	298.59	288.69	27.46	26.12	28.57	5.75	7200.00	7200.00	0.00%	31.26%	6.76%
S3	306.22	177.21	*	*	42.13	*	*	6.31	7200.00	7200.00	0.00%	-	-
S3_LRM		188.43	236.77	177.89	38.47	38.47	41.91	2.20	7200.00	7200.00	0.00%	61.11%	8.83%

* indicates a feasible solution not found within 2 hrs, while "-" indicates no gap returned by CPLEX

Table 3.4 provides CSPNS-D solutions on benchmark instances with 50 orders. Recall that in Table 3.1 the optimal solutions of CSPNS-D-2 and CSPNS-D-3 cannot be achieved for 40 orders within 2 hours. For 50 orders, CPLEX can not even complete the pre-processing stage, so a feasible solution is not achievable at all for CSPNS-D-2 and CSPNS-D-3 within 2 hrs time limit. This is why no values are shown in Table 3.4 for these two models. CSPNS-D-1 can produce feasible solutions in all cases. Performing the link-removal steps beforehand, as described above, reduces the number of links (nodes) effectively which makes feasible solutions achievable within the time limit, as shown by the *XX_LRM* in Table 3.4. The symbol "*" in the table indicates that a feasible solution was not found within the 2-hour time limit. The optimality gaps are calculated by the CPLEX solver using the formula in equation 3.26. A symbol "-" for optimality gaps of CSPNS-D-2 and CSPNS-D-3 indicates that optimality gap could not be reported by the CPLEX solver, as no feasible solution was obtained within the 2-hour time limit for these two models.

Examining Table 3.4, it is evident that the link-removal steps significantly reduce solution times without substantially affecting solution quality. In five out of ten instances, CSPNS-D-3 with link removal produces better solutions compared to CSPNS-D-1 without link removal. In instances where sub-optimality gaps are observed with CSPNS-D-1 (e.g., instances D3 and D4), CSPNS-D-3 with link removal slightly improves the CSPNS-D-1 solution within the same computational time. Upon closer inspection, CSPNS-D-1 with link removal appears more promising in terms of computational efficiency. Although slightly worse solutions (or comparable) are obtained by CSPNS-D-1 after link removal,

computation time is dramatically reduced. Therefore, we recommend deploying the CSPNS-D-1 with the link removal heuristic for large-scale implementations.

3.4.4 Drone Covering Range

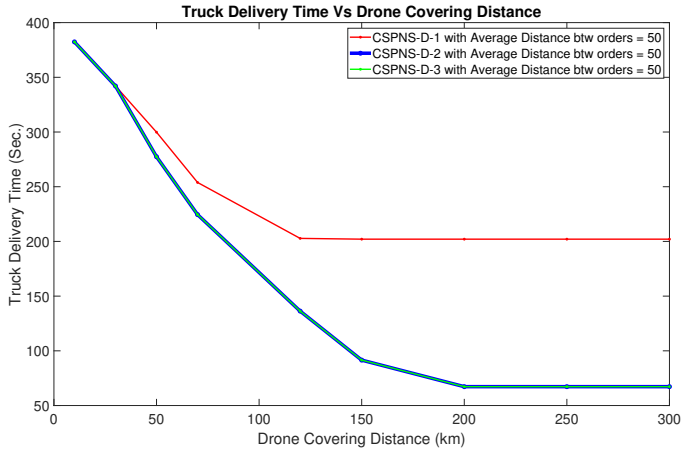
The drone covering distance and the average distance between orders can significantly influence drone usage in making deliveries. When the drone covering distance is large, and the average distance between orders is relatively small, more orders can be serviced by the drone, thereby reducing the total service time. Currently, drones in use have a maximum flying distance of 36 km with a fully charged battery [60]. However, due to technological advancements, we can expect next-generation drones with higher flying capabilities. In this section, we simulate scenarios with different order densities and drone covering ranges to draw insights from the numerical results that can potentially guide the future design and development of drones for home delivery.

In this section, we consider instances with 20 orders, varying the drone covering distance from 10 km to 300 km, across three average order distance levels: 50 km, 100 km, and 150 km. Results are presented in Table 3.5 and Figure 3.8. It is evident that delivery time is influenced by both the average distance between orders and the drone covering distance. Figure 3.8 illustrates that as the drone covering distance (d_{max}) increases, the delivery time by CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3 decreases until it reaches a threshold on the value of d_{max}. After that point, the delivery time remains the same for higher d_{max} values. However, the threshold value occurs earlier for CSPNS-D-1 than for CSPNS-D-2 and CSPNS-D-3, indicating a higher impact of increased d_{max} on the Link-Node-Link Operation and

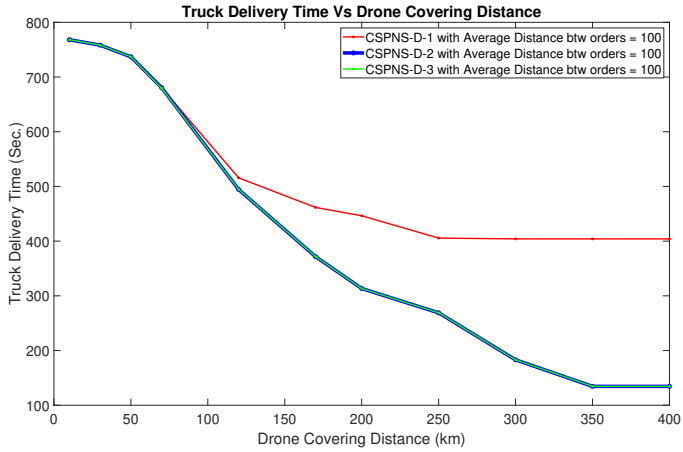
the Node-Operation. Additionally, it can be observed that CSPNS-D-2 performs as well as CSPNS-D-3 in nearly all instances, suggesting that excluding Node Operation won't lead to a cost increase in most cases.

Table 3.5: Effects of Drone Covering Distance and Average Distance between orders on Delivery Times

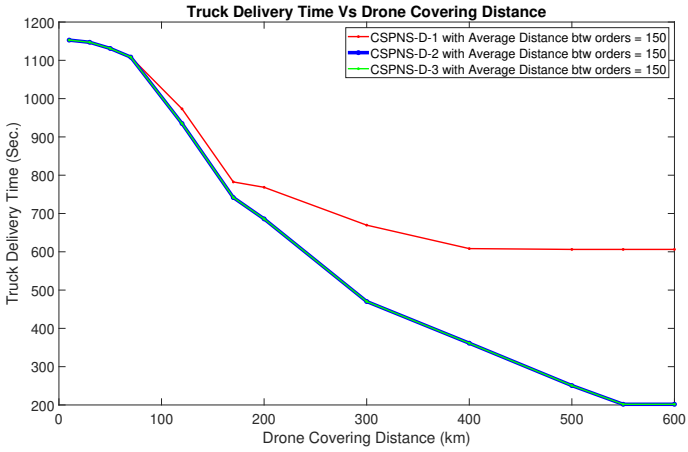
No. of	Av. Distance	Drone Covering	TSP delivery	CSPNS-D Delivery time			No. of orders served by drone			CSPNS-D CPU Time (sec)		
Orders	between orders	distance (d_{max})	time (s)	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3	CSPNS-D-1	CSPNS-D-2	CSPNS-D-3
		10		382.37	382.37	382.37	06	06	06	0.52	0.42	0.25
		30		342.05	342.05	342.05	11	11	11	0.34	2.24	2.36
		50		299.84	277.40	277.40	14	13	13	1.41	164.00	158.77
20		70		253.91	224.51	224.51	16	13	13	1.63	766.56	899.05
	50	120	384.290	202.77	136.31	136.31	17	16	16	1.38	697.98	879.16
		150		202.06	91.51	91.51	18	16	16	1.56	88.41	147.81
		200		202.06	67.25	67.25	18	17	17	1.66	36.64	41.81
		250		202.06	67.25	67.25	18	17	17	1.52	35.41	50.24
		300		202.06	67.25	67.25	18	17	17	1.50	37.11	51.92
		10		768.22	768.22	768.22	03	03	03	0.50	0.45	0.28
		30		758.25	758.25	758.25	07	07	07	0.66	0.48	0.30
		50		736.98	736.98	736.98	10	10	10	1.33	1.19	1.33
		70		680.50	680.50	679.54	13	13	12	1.42	18.17	12.61
		120		515.70	494.45	494.45	15	14	14	1.16	199.88	664.30
20	100	170	768.58	461.60	371.70	371.70	15	13	13	1.05	1889.60	2207.60
		200		446.36	313.42	313.42	16	16	16	1.16	1312.90	512.19
		250		405.54	268.90	268.90	15	17	17	2.08	595.95	497.41
		300		404.12	183.03	183.03	18	16	16	1.55	166.41	116.77
		350		404.12	134.50	134.50	18	17	17	1.70	39.92	58.49
		400		404.12	134.50	134.50	18	17	17	1.56	35.50	50.56
		10		1152.80	1152.80	1152.80	02	02	02	0.55	0.50	0.28
		30		1147.10	1147.10	1147.10	06	06	06	0.58	0.56	0.29
		50		1131.00	1131.00	1131.00	07	07	07	0.91	0.63	0.50
		70		1108.40	1108.40	1108.40	10	10	10	0.88	1.03	0.66
		120		973.27	934.98	934.98	12	12	12	2.34	12.34	13.30
20	150	170	1152.90	782.40	741.68	741.68	15	14	14	0.77	196.75	146.39
		200		768.42	685.89	685.89	15	12	12	1.44	661.45	701.25
		300		669.55	470.14	470.14	16	16	16	2.20	1516.50	1054.50
		400		608.31	361.29	361.29	17	16	16	1.67	543.17	661.50
		500		606.19	250.65	250.65	18	16	16	1.77	77.8	77.75
		550		606.19	201.75	201.75	18	17	17	1.75	35.17	63.95
		600		606.19	201.75	201.75	18	17	17	1.84	36.73	55.83



(a) Average distance = 50km



(b) Average distance = 100km



(c) Average distance = 150km

Figure 3.8. Relationship between delivery time and drone covering distance

In terms of suggestions on ideal drone covering range for home delivery services, we can see from Figure 3.8 that the delivery time decreases with a sharper

slope until drone covering distance is broadly the same as the average distance between orders. After this point, increasing the drone covering distance will lead to relatively marginal savings.

3.5 Summary

This chapter introduces the Covering Salesman Problem with Nodes and Segments Using Drones (CSPNS-D), a problem integrating a single truck with multiple drones to efficiently service customer orders. The approach involves determining the maximum coverage area of nodes or links serviced by the drones as the truck traverses the corresponding node or link. Three distinct Mixed Integer Linear Programming (MILP) models—CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3—are proposed, each considering different drone operations. CSPNS-D-1 seeks to minimize the truck's travel time while servicing all customers, considering Link and Link-Node operations. CSPNS-D-2 minimizes the truck's travel time, accounting for Link, Link-Node, and Link-Node-Link operations. Meanwhile, CSPNS-D-3 aims to minimize the total on-duty time of the truck, encompassing both driving and waiting time, while servicing all customers. Drone operations considered in CSPNS-D-3 include Link, Link-Node, Link-Node-Link, and Node operations. These models provide decision-making capabilities regarding which orders can be serviced by the truck and drones, as well as the optimal sequence for service. Experimental testing using both random and benchmark data sets from [61] demonstrates the efficiency of the models, with all three capable of solving problems with up to 20 orders efficiently to optimality using CPLEX. Results

indicate that the average savings over the Traveling Salesman Problem (TSP) for our models, particularly on benchmark instances of size 10 and 20, surpass those achieved by [32]. However, our models consider the use of multiple drones, while the reference utilizes only one drone. Further investigations reveal a sharp decrease in delivery time as drone flying capabilities increase, emphasizing the importance of determining the optimal covering distance for drones in areas with varying order density.

To deal with problems with larger instances, a simple approach based on removing links that are less likely to be chosen as part of the optimal solution is proposed, so as to reduce the possible number of choices for branching in the solution process. The approach works effectively in reducing computational time without affecting the solution quality significantly. We tested the proposed approach on ten benchmark instances with size 50. In all the cases, CSPNS-D-1 is solved efficiently exploring the entire B&B tree. The search process cannot be completed for CSPNS-D-2 and CSPNS-D-3 within 2 hours, however, there is a significant improvement when compared to solutions without removing links. Better solutions can be achieved faster especially on larger instances.

Chapter 4

The Robot-Assisted Delivery Problems (RADP)

4.1 Introduction

While Automated Delivery Robots (ADRs) are relatively new in academic research, they have captured the attention of researchers and logistics companies due to their promising practical applications. ADRs have become a focal point of discussion in the realms of innovation and freight distribution [62]. The integration of autonomous delivery robots for freight distribution is particularly noteworthy in light of the challenges posed by traffic congestion in city centers, often resulting in high delivery costs and delays. ADRs strategically navigate pedestrian routes to bypass traffic issues, effectively minimizing delivery times. Forecasts for the ADR market suggest significant growth, with expectations of reaching US$55 billion by 2026 and a robust annual growth rate of 20.4% [40]. This underscores the increasing importance and potential impact of ADRs in addressing contemporary

challenges in the logistics and delivery sectors. Companies such as Starship Technologies [63] and JD.Com [64], among others, have implemented the usage of autonomous robots for small parcel delivery on campuses and in residential areas in the United States and Beijing, respectively. Most robots operate on electric power and encounter limitations in travel distance due to their constrained battery capacity. Additionally, their travel speed is relatively slow, with a maximum of 4 mph, making them less efficient for addressing long-distance delivery challenges [65] . In response to this limitation, researchers have introduced the concept of the mother-ship model which is applied in our study. This innovative approach involves transporting robots using trucks to suitable customer areas, optimizing their usage, and overcoming the constraint of limited travel distance. Despite robots having a slower travel speed than drones, they present a more practical solution, especially in environments characterized by numerous stopping points within relatively short distances. The storage configuration of robots allows for multiple compartments, enabling a single trip to serve at least one customer at a time. Furthermore, it is argued that one of the major challenges associated with drone deliveries revolves around safety and security concerns. Government regulations impose restrictions on the commercial use of drones, and adverse weather conditions can impact drone operations [66]. In contrast, slow-moving parcel delivery robots offer easier management and control, minimizing the risk of devastating damage [67][68].

Both trucks and robots come with their respective limitations and advantages, as outlined in Table 1.1. However, leveraging these two entities in conjunction can yield several advantages, thanks to their complementarity. The combination

of trucks and robots allows for a more comprehensive and synergistic approach, mitigating the individual shortcomings of each and enhancing overall operational efficiency.

In this chapter, we introduce a framework called the "Robot-Assisted Delivery Problem (RADP)." This problem encompasses the integration of a truck, an autonomous robot (transported on the truck), and local depots into a unified model. This integration implies that customer orders can be fulfilled either by the truck directly, a robot deployed from the truck, or by a local depot. The operational process involves the truck departing from the distribution center and travelling to customer and/or depot locations to fulfill orders. The truck is equipped with a specific quantity of robots, which can be launched at either customer locations or local depot nodes to meet customer demands. To enable the simultaneous fulfillment of orders by both vehicles, the robots are collected at a distinct location from the drop-off point. Furthermore, the local depot is equipped with facilities such as localized robots, responsible for serving customers within the depot's coverage area.

We have developed two distinct yet consistent models to address the Robot-Assisted Delivery Problem (RADP): RADP-1 and RADP-2. In RADP-1, our modeling approach involves the exclusive consideration of each node and arc within the system. Conversely, RADP-2 adopts an operational concept, where each operation is a combination of both arcs and nodes. This fundamental difference in modeling approaches distinguishes the two variants. To provide a comprehensive understanding of their effectiveness, we proceed to compare the performance of RADP-1 and RADP-2. This comparative analysis will shed light on the strengths

and limitations of each model, facilitating a nuanced evaluation of their respective contributions to addressing the Robot-Assisted Delivery Problem.

4.1.1 Problem Description

Suppose we have a set of trucks $\mathcal{K}$, a set of robots $\mathcal{R}$, and a set of local depots $\mathcal{N}_\mathcal{D}$ to serve a set of customers $\mathcal{N}_C$. A truck starts from the distribution center, which is loaded with robots and customers orders, and returns to the distribution center after completing its tour. Although there is only one physical distribution center, we assign it two unique node numbers for notation purposes. Thus, the truck departs from the distribution center at node 1 and returns to the distribution center at node $N_C + N_D + 2$, where N_C and N_D are the numbers of customers and local depots, respectively. Therefore, $\mathcal{N} = \{1, \cdots, N_C + N_D + 2\}$ represents the set of all nodes in the network. In practice, the distribution center is located in rural areas, far from customer-dense areas. Robots, however, are not designed for long-distance travel considering their limited battery capacity and traveling speed. So, in this study, similar to what has been considered in most existing works, we use the truck to carry robots to suitable areas and launch/collect them from/to the truck. Instead of using the truck solely as mobile depots for robot delivery, as considered in some literature, they can also serve customers as standard delivery vans. The storage of each robot is separated into compartments for multiple shipments, and after delivery, the robot returns to a pickup point, which could be a customer node or a local depot node. In addition to serving customers, the truck can also serve multiple local depots, which are equipped with other types of customer service facilities, such as lockers, convenience stores, and/or even local

robots. A customer order can be served by either a truck, a robot launched from a truck, or a local depot.

The aim of this study is to optimize the delivery schedules of all facilities involved to minimize total work span of trucks. Specifically, we need to determine the following:

- The route to be traveled by each truck.
- The route to be traveled by the robot launched from the truck to reach each customer.
- The drop-off and collection points of the robot carried on the truck.
- The service mode and visiting time at each customer.

The important assumptions of the model are summarized below:

1. We assume that there are K trucks and R robots, starting and finishing at the distribution centre. We also assume that there are N_D local depots equipped with certain capacity to service local customer orders.
2. There are N_C customers to be served, each with known location, order size q_i and service time $o_T i$.
3. The location of local depots, their coverage areas and service capacity p_i are known.
4. Robots carried by trucks are identical with their storage separated into compartments, which allow multiple deliveries provided that their maximum allowable travel distance and payload capacity is not violated.

5. Each customer is served either by a truck, a localized robot from robot depot or a robot carried on a truck.

6. The drop-off and pick-up locations of robots launched from trucks could be different to allow the trucks and the robots serve customers simultaneously, rather than one vehicle wait while the other one is servicing.

7. Multiple robots can be launched and retrieved at the same customer node.

8. The robots must be picked up by the truck from which it was dropped.

Figure 4.1 provides a simple illustration of the Robot-Assisted Delivery Problem (RADP) involving 1 truck, 2 robots, 11 customers, and 1 local depot. Node 1 denotes the distribution center where the truck initiates and terminates its route. Node 2 represents a local depot, while nodes 3 through 13 correspond to customer nodes. The truck follows a route that includes nodes 3, 5, 7, 2, and 4. At node 3, two robots are launched. The first robot serves customers 8 and 9, later being picked up by the truck at node 4. Simultaneously, the second robot serves customers 13 and 12 and is picked up by the truck at local depot node 2. Nodes 6, 10, and 11 fall within the coverage range of local depot 2 and are consequently serviced by the local depot. This configuration optimizes the delivery process by efficiently utilizing both the truck and the robots.

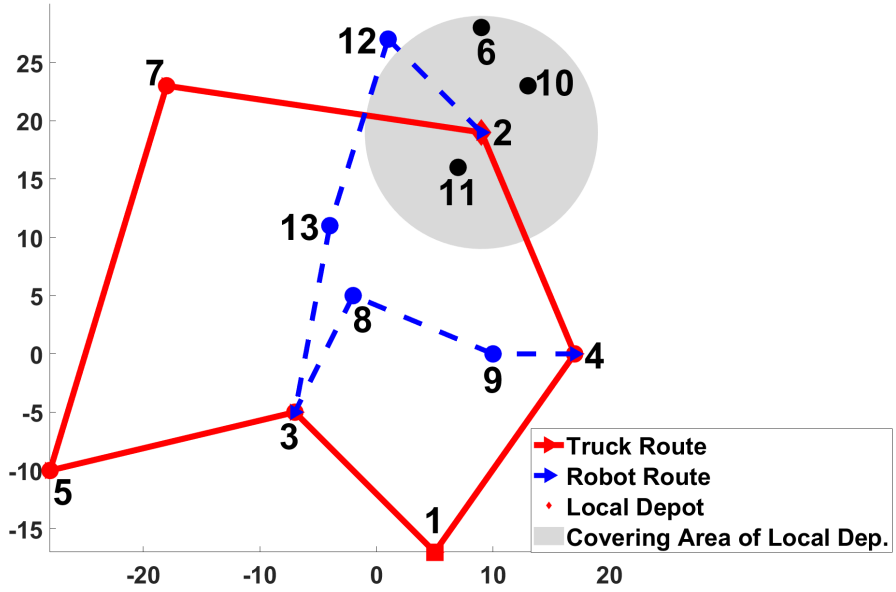


Figure 4.1. A simple illustration of RADP

4.2 Mathematical Formulation of RADP-1

This section outlines the mathematical formulation of the RADP-1 model. In this formulation, each arc (i, j) and each node i are treated independently during the modeling process.

4.2.1 Notations, Parameters and Variables of RADP-1

Sets

$\mathcal{K}$:= set of trucks.

$\mathcal{R}$:= set of robots.

$\mathcal{N}_C := \{2, \cdots, N_C + 1\}$ - set of customer nodes

$\mathcal{N}_\mathcal{D} := \{N_C + 2, \cdots N_C + N_D + 1\}$ - set of local depot nodes

$\mathcal{N}_{\mathcal{DC}} := \{1, N_C + N_D + 2\}$ - set of nodes representing distribution centre.

$\mathcal{N} := \{1, \cdots, N_C + N_D + 2\}$ - set of all nodes.

Parameters

- N_C : number of Customers
- N_D : number of local depots
- q_i : order size of customer node $i \in \mathcal{N}_C$
- $o_T i$: service time of truck at node $i \in \mathcal{N}_C \cup \mathcal{N}_D$
- $o_R i$: service time of robot at node $i \in \mathcal{N}_C$
- d_{ij} : distance between node i and node j, $\forall i \neq j \in \mathcal{N}$
- p_i : maximum capacity that can be served by the local depot.
- s : capacity occupation of a robot
- D_{max} : maximum travel distance of robot
- C_R : capacity of robot
- C_T : capacity of truck
- v_R : speed of robot
- v_T : speed of truck
- h : setup time of robot which include time to fill the robot with orders and setup its traveling routes

- $\lambda_{ij} = \begin{cases} 1, & \text{if node } i \text{ is in the cover range of local depot } j, \\ 0, & \text{otherwise} \end{cases} \quad i \in \mathcal{N}_C, j \in \mathcal{N}_D$

Discrete Variables

- $x_{ijk} = \begin{cases} 1, & \text{if truck } k \text{ travels } (i,j), \\ 0, & \text{otherwise} \end{cases} \quad i \neq j \in \mathcal{N}, \text{k} \in \mathcal{K}.$
- $y_{ijr} = \begin{cases} 1, & \text{if robot } r \text{ travels } (i,j), \\ 0, & \text{otherwise} \end{cases} \quad i \neq j \in \mathcal{N}_D \cup \mathcal{N}_C, r \in \mathcal{R}.$
- $\delta_{kr} = \begin{cases} 1, & \text{if robot } r \text{ is placed on truck } k, \\ 0, & \text{otherwise} \end{cases} \quad r \in \mathcal{R}, k \in \mathcal{K}.$
- $\gamma_{ikr} = \begin{cases} 1, & \text{if robot } r \text{ is launched from truck } k \text{ at node } i, \\ 0, & \text{otherwise} \end{cases} \quad r \in \mathcal{R}, k \in \mathcal{K}, i \in \mathcal{N}_D \cup \mathcal{N}_C.$
- $\eta_{ikr} = \begin{cases} 1, & \text{if robot } r \text{ is collected by truck } k \text{ at node } i, \\ 0, & \text{otherwise} \end{cases} \quad r \in \mathcal{R}, k \in \mathcal{K}, i \in \mathcal{N}_D \cup \mathcal{N}_C.$
- $z_{ij} = \begin{cases} 1, & \text{if node } i \text{ is served by local depot } j, \\ 0, & \text{otherwise} \end{cases} \quad i \in \mathcal{N}_C, j \in \mathcal{N}_D.$

Continous Variables

- t_{ik} = the visiting time at node i by truck k, $\forall i \in \mathcal{N}, \forall k \in \mathcal{K}$.
- τ_{ir} = the visiting time at node i by robot r, $\forall i \in \mathcal{N}_C \cup \mathcal{N}_D, \forall r \in R$.

Auxiliary Variables

- π_k = the total amount of shipments to be served by robots carried on truck k, $\forall k \in \mathcal{K}$.
- ϕ_k = the total amount of shipments to be served from local depots carried on truck k, $\forall k \in \mathcal{K}$.

4.2.2 RADP-1

A Mixed Integer Linear Programming (MILP) model (RADP-1) for the problem is formulated to find the best route to travel by all trucks so as to minimize the total working time of the trucks.

The RADP-1 Model

$$\min \sum_{k} \left(t_{(N_C+N_D+2,k)} - t_{1k} \right) \tag{4.1}$$

subject to:

$$\sum_{j \in \mathcal{N} \setminus \{1\}} x_{1jk} = 1, \qquad \forall k \in \mathcal{K}. \tag{4.2}$$

$$\sum_{i \in \mathcal{N_C} \cup \mathcal{N_D}} x_{i,N_C+N_D+2,k} = 1, \qquad \forall k \in \mathcal{K}. \tag{4.3}$$

$$\sum_{j \in \mathcal{N}} x_{ijk} = \sum_{j \in \mathcal{N}} x_{jik}, \qquad \forall i \in \mathcal{N_C} \cup \mathcal{N_D}, \forall k \in \mathcal{K}. \tag{4.4}$$

$$\sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} y_{ijr} + \sum_{k\in\mathcal{K}} \eta_{ikr} = \sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} y_{jir} + \sum_{k\in\mathcal{K}} \gamma_{ikr}, \qquad \forall i \in \mathcal{N}_C \cup \mathcal{N}_D, \forall r \in \mathcal{R}. \quad (4.5)$$

$$2\gamma_{ikr} \leq \delta_{kr} + \sum_{j\in\mathcal{N}} x_{ijk}, \quad \forall k \in \mathcal{K}, \forall r \in \mathcal{R}, \forall i \in \mathcal{N}_C \cup \mathcal{N}_D. \quad (4.6)$$

$$2\eta_{ikr} \leq \delta_{kr} + \sum_{j\in\mathcal{N}} x_{ijk}, \quad \forall k \in \mathcal{K}, \forall r \in \mathcal{R}, \forall i \in \mathcal{N}_C \cup \mathcal{N}_D. \quad (4.7)$$

$$\sum_{k\in\mathcal{K}} \delta_{kr} \leq 1, \qquad \forall r \in \mathcal{R}, \quad (4.8)$$

$$z_{ij} \leq \lambda_{ij}, \quad \forall i \in \mathcal{N}_C, j \in \mathcal{N}_D. \quad (4.9)$$

$$\sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{ijk} \geq \frac{1}{M} \sum_{i\in\mathcal{N}_C} q_i z_{ij}, \qquad \forall j \in \mathcal{N}_D. \quad (4.10)$$

$$\sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{K}} x_{ijk} + \sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} \sum_{r\in\mathcal{R}} y_{ijr} + \sum_{j\in\mathcal{N}_D} z_{ij} = 1 + \sum_{r\in\mathcal{R}} \sum_{k\in\mathcal{K}} \gamma_{ikr}, \qquad \forall i \in \mathcal{N}_C \quad (4.11)$$

$$\sum_{i\in\mathcal{N}_C} q_i z_{ij} \leq p_j, \qquad \forall j \in \mathcal{N}_D. \quad (4.12)$$

$$\sum_{i\in\mathcal{N}_C\cup\mathcal{N}_D} \sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} d_{ij} y_{ijr} \leq D_{max}, \qquad \forall r \in \mathcal{R}. \quad (4.13)$$

$$\sum_{i\in\mathcal{N}_C} q_i \left(\sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} y_{ijr} - \sum_{k\in\mathcal{K}} \gamma_{ikr} \right) \leq C_R, \qquad \forall r \in \mathcal{R}. \quad (4.14)$$

$$\sum_{i\in\mathcal{N}_C} \sum_{j\in\mathcal{N}} q_i x_{ijk} + \pi_k + \phi_k + \sum_{r\in\mathcal{R}} s.\delta_{kr} \leq C_T, \qquad \forall k \in \mathcal{K}. \quad (4.15)$$

$$\pi_k \geq \sum_{i\in\mathcal{N}_C} q_i \left(\sum_{j\in\mathcal{N}_C\cup\mathcal{N}_D} y_{ijr} - \sum_{k} \gamma_{ikr} \right) - M(1-\delta_{kr}), \qquad \forall k, \forall r. \quad (4.16)$$

$$\phi_k \geq \sum_{i\in\mathcal{N}_C} q_i z_{ij} - M\left(1 - \sum_{i\in\mathcal{N}} x_{ijk}\right), \qquad \forall k, \forall j. \quad (4.17)$$

$$t_{jk} \geq t_{ik} + o_i + \frac{d_{ij}}{v_T} + h\sum_{r\in\mathcal{R}} \gamma_{ikr} - M(1-x_{ijk}), \qquad i \neq j \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \quad (4.18)$$

$$\tau_{jr} \geq \tau_{ir} + o_i(1-\gamma_{ikr}) + \frac{d_{ij}}{v_R} - M(1-y_{ijr}), \qquad i \neq j \forall i,j \in \mathcal{N}_C \cup \mathcal{N}_D, \forall k \in \mathcal{K}. \quad (4.19)$$

$$t_{ik} \geq \tau_{ir} - M(1-\eta_{ikr}), \qquad \forall i \in \mathcal{N}_C \cup \mathcal{N}_D, \forall k \in \mathcal{K}, \forall r \in \mathcal{R}. \quad (4.20)$$

$$\tau_{ir} \geq t_{ik} + h - M(1-\gamma_{ikr}), \qquad \forall i \in \mathcal{N}_C \cup \mathcal{N}_D, \forall k \in \mathcal{K}, \forall r \in \mathcal{R}. \quad (4.21)$$

$$x_{ijk} \in \{0,1\}, \qquad i \neq j \in \mathcal{N}, k \in \mathcal{K}. \quad (4.22)$$

$$y_{ijr}, \delta_{kr}, \gamma_{ikr}, \eta_{ikr} \in \{0,1\}, \qquad i \in \mathcal{N}_C \cup \mathcal{N}_D, k \in \mathcal{K}, r \in \mathcal{R}. \quad (4.23)$$

$$z_{ij} \in \{0,1\}, \qquad i \in \mathcal{N}_C, j \in \mathcal{N}_D. \quad (4.24)$$

$$\pi_k, \phi_k, t_{ik}, \tau_{ir} \geq 0, \qquad i \in \mathcal{N}_C \cup \mathcal{N}_D, k \in \mathcal{K}, r \in \mathcal{R}. \quad (4.25)$$

Equation (4.1) represents the objective function that minimizes the total working time of all trucks. Constraints in Equations (4.2), (4.3), and (4.4) outline the standard network flow constraints for trucks. Equation (4.5) specifies the network flow constraint for robots. Constraints in Equations (4.6) and (4.7) ensure that a robot can be launched from or collected at a node only if it is carried by a truck that visits the node. Equation (4.8) guarantees that the same robot cannot be carried by more than one truck. Equation (4.9) ensures that a customer node is covered by a depot before it can be served by the depot. Constraint (4.10) ensures that all local depots with positive customer demands are visited by a truck. Equation (4.11) ensures that all customers are either served by a truck, a robot launched from a truck, or by a local depot. Equation (4.12) ensures that the total order size served by a local depot does not exceed its capacity. Equation (4.13) ensures that

the total distance traveled by a robot to serve customer(s) does not exceed its maximum travel capacity, while Equation (4.14) ensures that the total capacity of shipments carried on a robot does not exceed its carrying capacity. Constraint (4.15) ensures that the total capacity of shipments and the capacity occupation of all robots carried on a truck do not violate the truck's capacity. Constraint (4.16) calculates the amount of shipment serviced by a robot, while Constraint (4.17) calculates the amount of shipment serviced by a local depot. Equations (4.18) and (4.19) calculate the visiting time at nodes by trucks and robots, respectively, while Equations (4.20) and (4.21) establish their interconnections. Finally, Equations (4.22), (4.23), (4.24), and (4.25) define the values of the decision variables.

4.3 Mathematical Formulation of RADP-2

In this section, the Robot-Assisted Delivery Problem is further elucidated through a Mixed Integer Linear Programming (MILP) formulation, denoted as RADP-2. This formulation is a modification of the Travelling Salesman Problem with Drones (TSP-D) proposed by [23] and the Travelling Salesman Problem with Robots (TSP-R) by [10]. In this context, the term "operation," represented by $o \in O$, refers to a series of nodes that can be serviced either by a truck with a robot on board or by a truck and a robot that separate at the starting node, visiting other nodes concurrently and reuniting at a pickup node. Local depot nodes are serviced by truck only, but can be used for robot launch and pickup, while customer nodes can be visited by either a truck or a robot, providing flexibility in the delivery process.

Assumptions

The following assumptions are added to the list already discussed in 4.1.1.

1. We assume that robots are allowed to visit at most two nodes per launch.

2. Each operation $o \in O$ requires a completion time, t_o

4.3.1 Operations - Feasible Segments of Solutions

Each operation consists at least two nodes, i.e., the start and end nodes, which are referred to as drop off and pick up nodes of robots when two delivery modes are used in parallel. Depending on the size of the problem, the following operations are considered:

1. **Operations with Two Truck Nodes (Figure 4.2, Algorithm 1):** In this operation, the truck travels from node i to node j. The robot is not utilized; both nodes are served by the truck. Figure 4.2 illustrates the operations, while Algorithm 1 describes the preparation of the operation and its completion time.

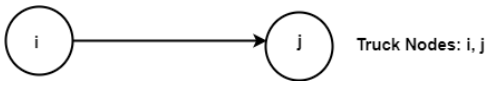


Figure 4.2. Operation with two truck nodes

2. **Operations with Two Truck Nodes and One Robot Node (Figure 4.3, Algorithm 2):** In this type of operation, the truck travels from nodes i to j while the robot is deployed at node i to service node k and is then retrieved at node j by the truck. Figure 4.3 illustrates the operations, while Algorithm 2 describes the preparation of the operation and its completion time.

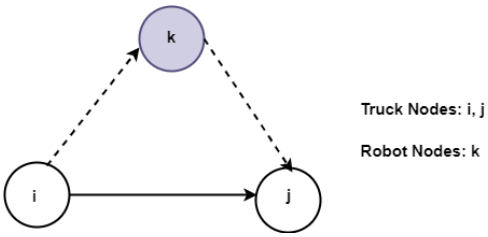


Figure 4.3. Operation with two truck nodes and one robot node

3. **Operations with Three Truck Nodes and One Robot Node (Figure 4.4, Algorithm 3):** In this operation, the truck travels from nodes i to j, then to k. A robot is deployed at i to service l and is retrieved at k. Figure 4.4 illustrates the operations, while Algorithm 3 describes the preparation of the operation and its completion time.

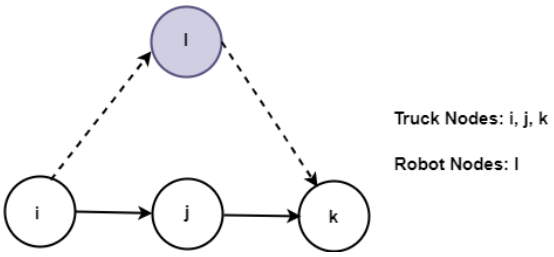


Figure 4.4. Operation with three truck nodes and one robot node

4. **Operations with Two Truck Nodes and Two Robot Nodes (Figure 4.5, Algorithm 2):** In this operation, the truck travels from i to j, while a robot is deployed at i to service both k and l, and is subsequently collected at j. Figure 4.5 illustrates the operations, while Algorithm 2 describes the preparation of the operation and its completion time.

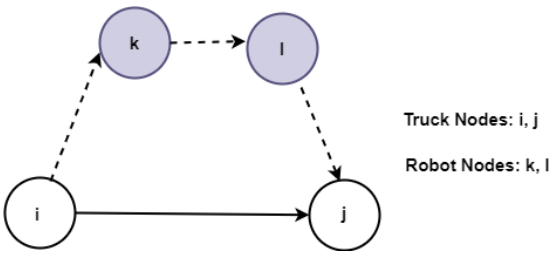


Figure 4.5. Operation with two truck nodes and two robot node

5. **Operations with Three Truck Nodes and Two Robot Nodes (Figure 4.6, Algorithm 3):** In this type of operation, the truck travels from nodes i to j and then to k, while a robot is deployed at node i to service nodes l and m, and is subsequently collected at k. Figure 4.6 illustrates the operations, while Algorithm 3 describes the preparation of the operation and its completion time.

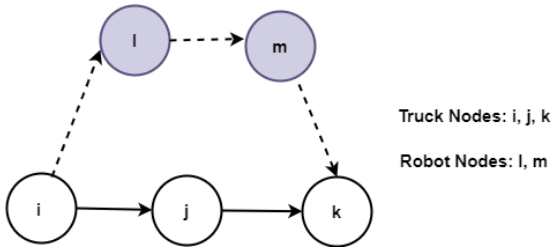


Figure 4.6. Operation with three truck nodes and two robot nodes

6. **Operations with Four Truck Nodes and One Robot Node (Figure 4.7, Algorithm 4):** The truck travels from i to j, then from j to k, and finally from k to l, while a robot is deployed at i to service m and is subsequently collected at l. Figure 4.7 illustrates the operations, while Algorithm 4 describes the preparation of the operation and its completion time.

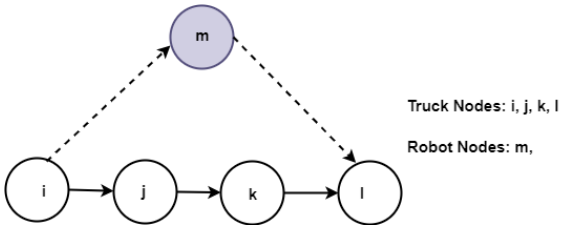


Figure 4.7. Operation with four truck nodes and one robot node

7. **Operations with Four Truck Nodes and Two Robot Nodes (Figure 4.8, Algorithm 4):** In these operations, the truck travels from i to j, then from j to k, and finally from k to l, while a robot is launched at i to service m and n,

and is subsequently collected at l. Figure 4.8 illustrates the operations, while Algorithm 4 describes the preparation of the operation and its completion time.

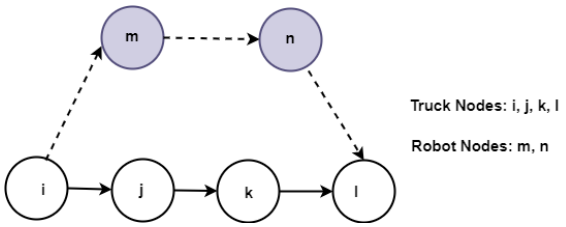


Figure 4.8. Operation with four truck nodes and two robot nodes

8. **Operations with Five Truck Nodes and One Robot Node (Figure 4.9, Algorithm 5):** In these operations, the truck travels from i to j, then from j to k, followed by k to l, and finally from l to m, while a robot is launched at i to service n and is subsequently collected at m. Figure 4.9 illustrates the operations, while Algorithm 5 describes the preparation of the operation and its completion time.

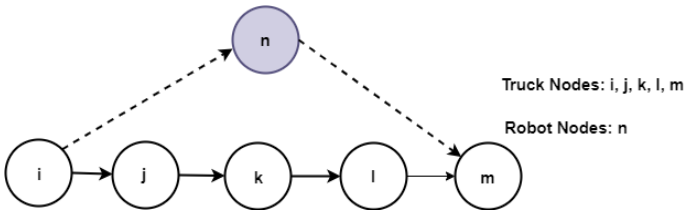


Figure 4.9. Operation with five truck nodes and one robot node

Apart from the operations listed above, we have additionally joined any pair of feasible operations with identical truck node sequences, but having distinct nodes serviced by robots. These joined operations facilitate the concurrent service of customers by two robots carried on the truck. The joined operations can be obtained from combinations of the same type of operations or different combinations of operations types having the same number of truck nodes. Figure 4.10 below is an

example of one joined operation obtained from combination of two operations both with three truck nodes i, j and k. Notably, one of the operations has robot node l while the other has m and n, these robot nodes remains distinct ($m \neq l$ and $n \neq l$). For a clear outline of the process to prepare joined operations please refer to Algorithm 6. List of all operations discussed in items 1-8 above together with the joined operations forms the overall list of operations used in the RADP-2 model.

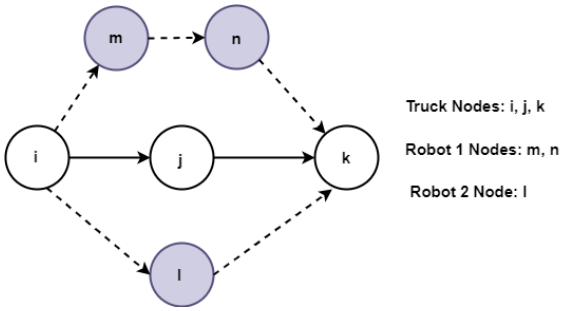


Figure 4.10. Example of Joined Operations

Algorithm 1 Constructing Operations with two truck nodes

Input: $d_{i,j}, v_T, v_R, O_T, O_R, h, D_{max}, nC, nD$
Output: Operations with two truck nodes and their completion times

count = 1
$n = nC + nD + 2$
for $i = 1$ to $n - 1$ **do**
  for $j = 2$ to n **do**
    if $i \neq j$ **then**
      if ($i = 1$ and $j < n$) OR ($i > 1$ and $j \geq 2$) **then**
        Add operation from i to j to $Op2$ list
        Compute $t_2(count) = \frac{d_{i,j}}{v_T} + O_T$
        count = count + 1
      end if
    end if
  end for
end for

Algorithm 2 Constructing Operations with three nodes (2 Trucks and 1 robot) and four nodes (2 trucks and 2 robots)

Input: Algorithm 1, nC nD, h, v_R, D_{max}, $d_{i,j}$, O_R
Output: List of Operations with three nodes(2 trucks and 1 robot nodes) and four nodes(2 trucks and 2 robot nodes) with their corresponding times.

$set(x, y) = size(Op2)$
count = 1
$n = nC + nD + 2$ ▷ All nodes including start and end
for $i = 1$ to x **do**
  if $Op2(i,1) \geq 2$ and $Op2(i,2) \geq 2$ and $Op2(i,2) < n$ **then**
    Compute $T_t = t_2(i) + h$
    for $k = nD + 2$ to $n - 1$ **do** ▷ k is robot node
      if $k \neq Op2(i,1)$ and $k \neq Op2(i,2)$ **then**
        Compute $R_t = \frac{d(Op2(i,1),k)+d(k,Op2(i,2)}{v_R} + O_R$
        if $d(Op2(i,1),k) + d(k,Op2(i,2) \leq D_{max}$ **then**
          Add Operation from $Op2(i,1)$ to $Op2(i,2)$ and from $Op2(i,1)$ to k, then from k to $Op2(i,2)$ to $Op3$ list
          Select $t_3(count) = max\{T_t, R_t\}$
          count = count + 1
        end if
        for $l = nD + 2$ to $n - 1$ **do** ▷ l is robot node
          if $l \neq Op2(i,1)$ and $l \neq Op2(i,2)$ and $l \neq k$ **then**
            Compute $R_t = \frac{d(Op2(i,1),k)+d(k,l)+d(l,op2(i,2)}{v_R} + 2 * O_R$
            if $d(Op2(i,1),k) + d(k,l) + d(l,op2(i,2) \leq D_{max}$ **then**
              Add Operation from $Op2(i,1)$ to $Op2(i,2)$ and from $Op2(i,1)$ to k, from k to l, then from l to $Op2(i,2)$ to $Op4$ list
              Select $t_4(count) = max\{T_t, R_t\}$
              count = count + 1
            end if
          end if
        end for
      end if
    end for
  end if
end for

Algorithm 3 Constructing Operations with four nodes (3 truck nodes and 1 robot node) and five nodes (3 truck nodes and 2 robot nodes)

Input: h, D_{max}, $d_{i,j}$, v_T, v_R, O_T,O_R, nC, nD
Output: List of Operations with four nodes (3 trucks and 1 robot node) and five nodes (3 trucks and 2 robot nodes) with their corresponding times.

count = 1
$n = nC + nD + 2$ ▷ All nodes including start and end
for $i = 2$ to $n-1$ **do**
  for $j = 2$ to $n-1$ **do**
    for $k = 2$ to $n-1$ **do**
      if $k \neq i$ and $i \neq j$ and $j \neq k$ **then**
        Compute $T_t = \frac{(d_{i,j}+d_{j,k})}{v_T} + h + 2 * O_T$
        for $l = nD + 2$ to $n-1$ **do** ▷ l is robot node
          Compute $R_t = \frac{(d_{i,l}+d_{l,k})}{v_R} + O_R$
          if $(d_{i,l} + d_{l,k}) \leq D_{max}$ **then**
            Add operation from i to j, then j to k and from i to l, then l to k to $Op4^{(2)}$ list
            Select $t_4^{(2)}(count) = max\{T_t, R_t\}$
            count = count + 1
          end if
        end for
        for $m = nD + 2$ to $n-1$ **do** ▷ m is robot node
          if $m \neq i$ and $m \neq j$ and $m \neq k$ and $m \neq l$ **then**
            Compute $R_t = \frac{d_{i,l}+d_{l,m}+d_{m,k}}{v_R} + 2 * O_R$
            if $d_{i,l} + d_{l,m} + d_{m,k} \leq D_{max}$ **then**
              Add operation from i to j, then j to k and from i to l, l to m then m to k to $Op5$ list
              Select $t_5(count) = max\{T_t, R_t\}$
            end if
          end if
        end for
      end if
    end for
  end for
end for

Algorithm 4 Constructing Operations with five nodes (4 trucks and 1 robot nodes) and six nodes (4 trucks and 2 robot nodes)

Input: $d_{i,j}$, v_T, v_R, O_T, O_R, D_{max}, h, nC, nD

Output: List of Operations with five nodes (4 trucks and 1 robot nodes) and six nodes (4 trucks and 2 robot nodes) with their corresponding times.

count = 1
$n = nC + nD + 2$ ▷ All nodes including start and end
for $i = 2$ to $n-1$ **do**
  for $j = 2$ to $n-1$ **do**
    for $k = 2$ to $n-1$ **do**
      for $l = 2$ to $n-1$ **do**
        if $i \neq j$ and $i \neq k$ and $i \neq l$ and $j \neq k$ and $j \neq l$ and $k \neq l$ **then**
          Compute $T_t = \frac{\left(d_{i,j}+d_{j,k}+d_{k,l}\right)}{v_T} + h + 3 * O_T$
          for $m = nD + 2$ to $n-1$ **do** ▷ m is robot node
            if $m \neq i$ and $m \neq j$ and $m \neq k$ and $k \neq l$ **then**
              Compute $R_t = \frac{(d_{i,m}+d_{m,l})}{v_R} + O_R$.
              if $d_{i,m} + d_{m,l} \leq D_{max}$ **then**
                Add operation from i to j,j to k, then k to l and from i to m then m to l to $Op5^{(2)}$ list.
                Select $t_5^{(2)}(count) = max\{T_t, R_t\}$
                count = count + 1
              end if
              for $h = nD + 2$ to $n-1$ **do** ▷ h is robot node
                if $h \neq i$ and $h \neq j$ and $h \neq k$ and $h \neq l$ and $h \neq m$ **then**
                  Compute $R_t = \frac{(d_{i,m}+d_{m,h}+dh,l)}{v_R} + 2 * O_R$
                  if $d_{i,m} + d_{m,h} + dh,l \leq D_{max}$ **then**
                    Add operation from i to j,j to k, then k to l and from i to m, m to h then h to l to $Op6$ list.
                    Select $t_6(count) = max\{T_t, R_t\}$
                    count = count + 1
                  end if
                end if
              end for
            end if
          end for
        end if
      end for
    end for
  end for
end for

Algorithm 5 Constructing Operations with six nodes (5 trucks and 1 robot node)

Input: $d_{i,j}$, v_T, v_R, O_T, O_R, h, D_{max}, nC, nD
Output: List of Operations with six nodes (5 trucks and 1 robot nodes) with their corresponding times.

count = 1
$n = nC + nD + 2$ ▷ All nodes including start and ending nodes
for $i = 2$ to $n-1$ **do**
  for $j = 2$ to $n-1$ **do**
    for $k = 2$ to $n-1$ **do**
      for $l = 2$ to $n-1$ **do**
        for $m = 2$ to $n-1$ **do**
          if $i \neq j$ and $i \neq k$ and $i \neq l$ and $i \neq m$ and $j \neq k$ and $j \neq l$ and $j \neq m$ and $k \neq l$ and $k \neq m$ and $l \neq m$ **then**
            Compute $T_t = \frac{\left(d_{i,j}+d_{j,k}+d_{k,l}+d_{l,m}\right)}{v_T} + h + 4 * O_T$
            for $h = nD + 2$ to $n-1$ **do** ▷ h is robot node
              if $h \neq i$ and $h \neq j$ and $h \neq k$ and $h \neq l$ and $h \neq m$ **then**
                Compute $R_t = \frac{(d_{i,h}+d_{h,m})}{v_R} + O_R$
                if $d_{i,h} + d_{h,m} \leq D_{max}$ **then**
                  Add operation from i to j,j to k,k to l then l to m and from i to h then h to m to $Op6^{(2)}$ list.
                  Select $t_6^{(2)}(count) = max\{T_t, R_t\}$
                  count = count + 1
                end if
              end if
            end for
          end if
        end for
      end for
    end for
  end for
end for

Algorithm 6 Merging list of all operations prepared in Algorithms 1, 2, 3, 4 and 5

Input: Algorithms 1,2, 3, 4 and 5
Output: List of all operations with their completion times

Set $\mathcal{H} = Op2 \cup Op3 \cup Op4 \cup Op4^{(2)} \cup Op5 \cup Op5^{(2)} \cup Op6 \cup Op6^{(2)}$
Set $\mathcal{T} = t_2 \cup t_3 \cup t_4 \cup t_4^{(2)} \cup t_5 \cup t_5^{(2)} \cup t_6 \cup t_6^{(2)}$
Set (m,n) = size($\mathcal{H}$)
Set $\mathcal{K}_i$ = set of truck nodes in operation $i, \forall i \in \mathcal{H}$
Set $\mathcal{R}_i$ = set of robot nodes in operation $i, \forall i \in \mathcal{H}$
Set $\mathcal{J} = \emptyset$ ▷ $\mathcal{J}$ is a list of joined operations with same truck nodes
uprow = 1
while $uprow < m - 1$ **do**
    for $botrow = uprow + 1$ to m **do**
        if $\mathcal{K}(uprow) = \mathcal{K}(botrow)$ and in same order **then**
            if $\mathcal{R}(uprow) \neq \mathcal{R}(botrow)$ **then**
                Join operations $\mathcal{H}(uprow)$ and $\mathcal{H}(botrow)$ together and add to $\mathcal{J}$
                Select $\mathcal{T}^{(2)} = max\{\mathcal{T}(uprow), \mathcal{T}(botrow)\}$
            end if
        end if
    end for
    $uprow = uprow + 1$
end while
$OperList = \mathcal{H} \cup \mathcal{J}$ ▷ List of all operations
$t_{OperList} = \mathcal{T} \cup \mathcal{T}^{(2)}$

4.3.2 RADP - 2

Upon the preparation of all operations, the starting nodes, ending nodes, nodes covered, and the duration required to complete all travel and services for the relevant nodes can be calculated in advance and fed into the model as parameters.

- $O^-(i) \subset O$: Set of operations with start node $i \in N$
- $O^+(i) \subset O$: Set of operations with end node $i \in N$
- $O(i) \subset O$: Set of all operations that contain node $i \in N$
- t_o : Duration of operation $o \in O$

Decision Variables

- $x_o = \begin{cases} 1, & \text{if operation } o \text{ is chosen} \\ 0, & \text{otherwise} \end{cases} \quad \forall o \in O$
- $\beta_i = \begin{cases} 1, & \text{if depot } i \text{ has positive demand} \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in N_D$
- v_i : visiting time at node $i \in N$

The RADP-2 Model

$$\min v_{(nC+nD+1)} \tag{4.26}$$

subject to:

$$\sum_{o \in O^-(i)} x_o = \sum_{o \in O^+(i)} x_o, \qquad \forall i \in \mathcal{N}_C \cup \mathcal{N}_D \tag{4.27}$$

$$\sum_{o \in O(i)} x_o + \sum_{j \in \mathcal{N}_D} \lambda_{ij} \beta_j \geq 1, \qquad \forall i \in \mathcal{N}_C \tag{4.28}$$

$$\sum_{o \in O^-(0)} x_o = 1 \tag{4.29}$$

$$\sum_{o \in O^+(N_C+N_D+1)} x_o = 1 \tag{4.30}$$

$$\sum_{o \in O(i)} x_o \geq \beta_i, \qquad \forall i \in \mathcal{N}_D \tag{4.31}$$

$$v_j \geq v_i + \sum_{o \in (O^-(i) \cap O^+(j))} t_o x_o - M(1 - x_o), \qquad \forall i \neq j \in \mathcal{N} \tag{4.32}$$

$$x_o \in \{0,1\}, \qquad \forall o \in O \tag{4.33}$$

$$\beta_i \in \{0,1\}, \qquad \forall i \in \mathcal{N}_D \tag{4.34}$$

$$v_i \geq 0, \qquad \forall i \in \mathcal{N} \tag{4.35}$$

The objective function (4.26) minimizes the total time to complete the tour. Constraint (4.27) represents the standard network flow constraint, and Constraint (4.28) ensures that all customer nodes are covered either by an operation or by a local depot. Constraint (4.29) and (4.30) compel the route to start from and finish at the distribution center. Additionally, Constraint (4.31) ensures that all local depots

with positive demands are covered by an operation. Constraint (4.32) calculates the visiting time at each node, serving as sub-tour elimination constraints. Finally, Constraints (4.33), (4.34), and (4.35) set the domain for variables x_o, β_i, and v_i.

4.4 RADP-1 Vs RADP-2

In this section, we present the computational results for both RADP-1 and RADP-2, taking into account their computational complexity and performance. The MILP solution approach for these models was implemented in MATLAB R2020a and executed on a CPU equipped with an Intel(R) Core(TM) i5-7300U processor. The MILP problems were solved using CPLEX Studio 12.10.0.

4.4.1 CPLEX MILP Solution of RADP-1 and RADP-2

Due to the complexity of the problem, our experiments involved scenarios with six and seven customers, one local depot, one truck, and two robots. In all the experiments, we set the average speed of the truck and robot at 18.64 mph and 6.21 mph, respectively, following the approach outlined in [69]. The local depot's covering range was defined as 500 meters. The service time for the truck was set at 6 minutes, while the robot's service time was 3 minutes. These times include considerations for additional tasks such as parking, unloading, and any necessary interactions with humans at the customer's location. The setup time for a robot was 1 minute, reflecting the time required to prepare and release the robot when it is launched from the truck. The experiments were divided into three distinct groups, with each group containing ten experiments involving six customers and

five experiments with seven customers. The first group contains the results of experiments with an average distance between orders in the range of 420m to 850m, which are displayed in Table 4.1. The second group involves experiments with average distances in the range of 1100m to 2600m, displayed in Table 4.2. The third group focuses on experiments with average distances in the range of 3000m to 4500m, as shown in Table 4.3. In each table, we present the solution, computation time, service facility of customers, TSP solution, savings over TSP, and the optimality gap calculated by the CPLEX solver using equation 3.26 for each model. The results across all tables indicate that RADP-1 is easier to solve than RADP-2 due to the larger number of feasible operations involved in RADP-2. Both models were solved for seven nodes, including six customers and one local depot, to optimality within two hours. Across all the tables, the TSP solutions for each of the experiments are the same for both RADP-1 and RADP-2. Savings over TSP (except experiment 5 of Table 4.3) and the optimality gap for experiments 1-10 are also the same for both models. An optimality gap of 0.00% for experiments 1-10 indicates that an optimal solution is achieved for both models. However, large optimality gaps for RADP-2 in experiments 11-15 indicate that more computational time is required by the solver to further improve the RADP-2 solutions due to the increased number of nodes (by one), resulting in a large number of feasible operations.

The highest Savings over TSP (42.25%) was achieved using RADP-1 in Experiment 11 of Table 4.1, with an optimum solution found within 4514.89 seconds. However, an optimum solution could not be found for RADP-2 within the computational time limit due to the exponential increase in the number of feasible

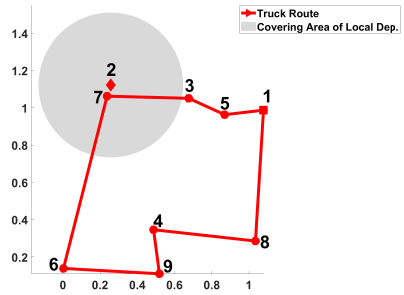
operations, resulting in an optimality gap of 84.88%. A graphical representation of the experiment's solution is displayed in Figure 4.11, with sub-figures (a), (b), and (c) showing the TSP, RADP-1, and RADP-2 solutions, respectively. The optimum solution is achieved using RADP-1, with the truck route following the sequence 1-5-6-4-1. The two robots carried by the truck are both launched at node 5 and collected at node 4. One of the robots travels the path 5-3-7-4, while the other takes the route 5-8-9-4. Notably, although the customer at node 7 falls within the covering range of the local depot, the customer is best served by a robot rather than receiving service from the local depot. Due to computational challenges associated with RADP-2, a sub-optimal solution is found within a 2-hour time limit. In the RADP-2 solution, the truck travels the sequence 1-2-9-4-5-1. The robots are both launched at the local depot and collected at node 4. One of them is then relaunched at the same node to serve customer 3 and collected at node 5. An investigation of the solution reveals that as the truck travels the segment 2-9-4, one of the robots travels 2-6-4. Considering the slower speed of the robot compared to the truck, this arrangement could result in the truck waiting for the robot at node 4 for an extended period. Similarly, as the truck travels the sequence 4-5, the robot traveling 4-3-5 may lead to longer waiting times for the truck at node 5.

The solutions in Experiment 1 of Table 4.1 show that all six customers are equally served by the service facilities in both RADP-1 and RADP-2, resulting in an average savings over TSP of approximately 27.32% for both models. RADP-1 can be solved within 2 minutes, whereas RADP-2 takes almost 2 hours to find the optimal solution. A graphical representation of the solution is also available in Figure 4.12, featuring sub-figures (a), (b), and (c) displaying the TSP, RADP-1, and

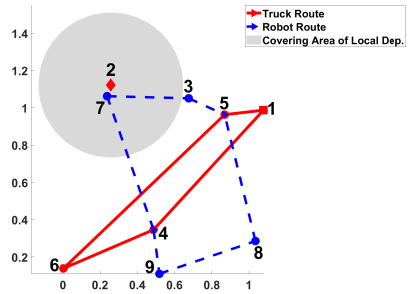
RADP-2 solutions, respectively. Interestingly, both RADP-1 and RADP-2 yield the exact same optimal solution, with identical truck and robot routes. In this solution, the truck follows the sequence 1-8-4-2-1, while each robot takes routes 8-6-2 and 8-5-2. Customers at nodes 3 and 7 are served from the local depot.

Table 4.1: Solutions of RADP-1 Vs RADP-2, Av. Dist. =(420m - 850m)

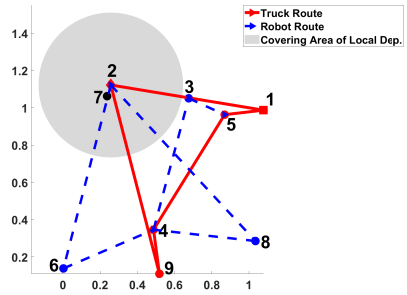
Exp	Model	No. of Cust.	CPLEX MILP Solutions					TSP Sol. (h)	Savings Over TSP (%)	Optimality Gap %
			Obj. Val (h)	Comp.time (Sec.)	Service Facility					
					Depot	Robot	Truck			
1	RADP-1	6	0.5154	129.14	2	2	2	0.7091	27.32	0.00
	RADP-2		0.5154	6820.17	2	2	2			
2	RADP-1	6	0.5113	115.90	0	4	2	0.7051	27.49	0.00
	RADP-2		0.5113	6627.65	0	4	2			
3	RADP-1	6	0.4825	105.09	2	3	1	0.6924	30.31	0.00
	RADP-2		0.4825	7120.29	2	3	1			
4	RADP-1	6	0.4118	46.97	4	1	2	0.4683	12.06	0.00
	RADP-2		0.4118	6718.59	4	1	2			
5	RADP-1	6	0.5054	92.69	0	3	3	0.8040	37.14	0.00
	RADP-2		0.5054	6900.26	0	3	3			
6	RADP-1	6	0.4353	60.94	3	2	1	0.5879	25.96	0.00
	RADP-2		0.4353	5870.47	3	2	1			
7	RADP-1	6	0.5061	294.00	0	3	3	0.7347	31.11	0.00
	RADP-2		0.5061	7123.89	0	3	3			
8	RADP-1	6	0.5336	109.26	0	3	3	0.7224	26.14	0.00
	RADP-2		0.5336	6997.98	0	3	3			
9	RADP-1	6	0.5160	100.29	2	2	2	0.7111	27.44	0.00
	RADP-2		0.5160	6782.15	2	2	2			
10	RADP-1	6	0.5054	79.13	0	3	3	0.8156	38.03	0.00
	RADP-2		0.5054	6300.16	0	3	3			
11	RADP-1	7	**0.5540**	4514.89	0	4	3	0.9641	42.54	0.00
	RADP-2		0.7212	7200.00	1	3	3		25.19	84.88
12	RADP-1	7	**0.5503**	2231.26	2	3	2	0.8303	33.72	0.00
	RADP-2		0.6352	7200.00	2	2	3		23.50	83.10
13	RADP-1	7	**0.5277**	3678.75	2	3	2	0.8054	34.48	0.00
	RADP-2		0.5976	7200.00	2	2	2		25.80	82.52
14	RADP-1	7	**0.4969**	2442.14	2	3	2	0.7131	30.32	0.00
	RADP-2		0.6137	7200.00	2	2	3		13.94	82.17
15	RADP-1	7	**0.5142**	2100.43	1	4	2	0.8131	36.76	0.00
	RADP-2		0.6214	7200.00	2	2	3		23.58	83.18
					Average Savings			**RADP-1**	**28.80**	
								RADP-2	**26.33**	



(a) TSP (Optimum Solution = 0.9641)

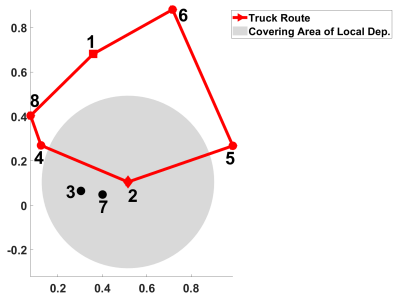


(b) RADP-1 (Optimum Solution = 0.5540)

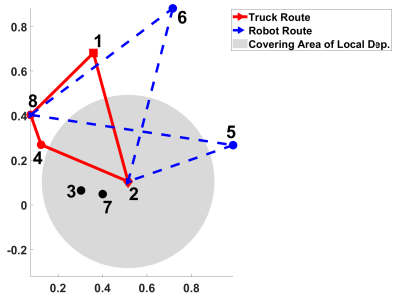


(c) RADP-2 (Sub-Optimal Solution = 0.7212)

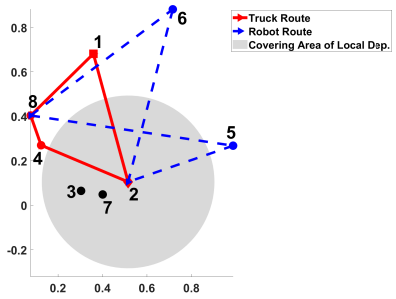
Figure 4.11. Solution of Experiment 11 of Table 4.1



(a) TSP (Optimum Solution = 0.7091)



(b) RADP-1 (Optimum Solution = 0.5154)



(c) RADP-2 (Optimum Solution = 0.5154)

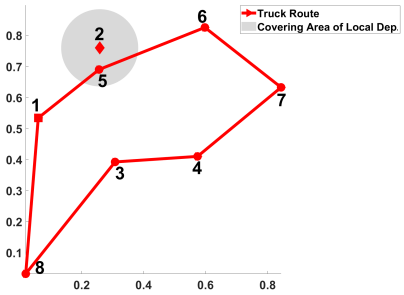
Figure 4.12. Solution of Experiment 1 in Table 4.1

The solutions in Experiment 1 of Table 4.2 indicate that optimal solutions for both RADP-1 and RADP-2 were found within the 2-hour time limit. The graphical representations in Figures 4.13 (a) and 4.13 (b) reveal them as alternative optimal solutions. In RADP-1, the truck route follows the sequence 1-5-6-7-3-1, and the robots are launched at node 5 to fulfill orders for customers 4 and 8. They are then both collected by the truck at node 3. However, in RADP-2, the truck follows the sequence 1-5-7-4-3-1. The robots are both launched at node 5, similar to RADP-1,

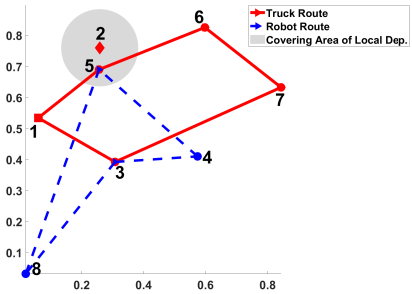
but one of the robots that serves customer 4 in RADP-1 serves customer 6 in RADP-2. RADP-2 has a shorter truck route compared to RADP-1, but the truck has to wait longer at node 3 for the robot that travels the link 5-8-3. In both solutions, the robot travel time across the sequence 5-8-3 determines the duration of the truck working time.

Table 4.2: Solutions of RADP-1 Vs RADP-2, Av. Dist. = (1100m - 2600m)

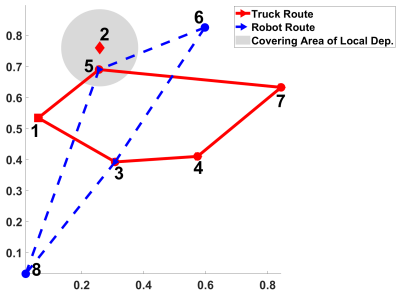
Exp	Model	No. of Cust.	CPLEX MILP Solutions					TSP Sol. (h)	Savings Over TSP (%)	Optimality Gap %
			Obj. Val (h)	Comp.time (Sec.)	Service Facility					
					Depot	Robot	Truck			
1	RADP-1	6	0.8059	259.14	0	2	4	1.0369	22.28	0.00
	RADP-2		0.8059	6862.64	0	2	4			
2	RADP-1	6	0.8561	36.34	2	2	2	1.039	17.60	0.00
	RADP-2		0.8561	1112.20	2	2	2			
3	RADP-1	6	0.7205	91.30	0	3	3	0.9967	27.71	0.00
	RADP-2		0.7205	5602.57	0	3	3			
4	RADP-1	6	0.7291	110.79	0	3	3	1.0281	29.08	0.00
	RADP-2		0.7291	5678.92	0	3	3			
5	RADP-1	6	0.726	258.20	0	3	3	0.9346	22.32	0.00
	RADP-2		0.726	6105.87	0	3	3			
6	RADP-1	6	0.8512	19.57	2	1	3	0.9449	9.92	0.00
	RADP-2		0.8512	431.59	2	1	3			
7	RADP-1	6	0.7853	133.97	0	4	2	1.1003	28.63	0.00
	RADP-2		0.7853	4674.38	1	3	2			
8	RADP-1	6	0.8649	354.60	0	3	3	1.1521	24.93	0.00
	RADP-2		0.8649	6820.85	0	3	3			
9	RADP-1	6	0.8807	45.22	0	2	4	1.1166	21.13	0.00
	RADP-2		0.8807	5783.36	0	2	4			
10	RADP-1	6	0.5893	406.19	0	3	3	0.8712	32.36	0.00
	RADP-2		0.5893	5823.78	0	3	3			
11	RADP-1	7	**0.8571**	572.93	0	3	4	1.1687	26.66	0.00
	RADP-2		0.9059	7200.00	1	2	4		22.49	86.76
12	RADP-1	7	0.8553	727.23	2	2	3	1.1107	22.99	0.00
	RADP-2		0.8553	7200.00	2	2	3		22.99	81.63
13	RADP-1	7	**0.7498**	884.34	2	3	2	1.047	28.39	0.00
	RADP-2		0.8552	7200.00	2	2	3		18.32	81.32
14	RADP-1	7	**0.8187**	219.80	0	3	4	1.2099	32.33	0.00
	RADP-2		0.9371	7200.00	0	2	5		22.55	86.72
15	RADP-1	7	0.8259	266.40	2	2	3	1.0238	19.33	0.00
	RADP-2		0.8259	7200.00	2	2	3		19.33	82.20
					Average Savings			**RADP-1**	**24.38**	
								RADP-2	**22.78**	



(a) TSP (Optimum Solution = 1.0369)

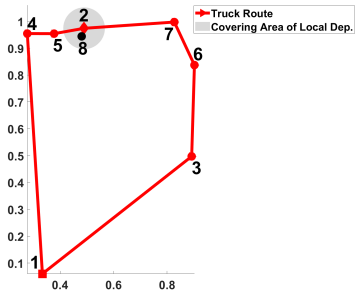


(b) RADP-1 (Optimum Solution = 0.8059)

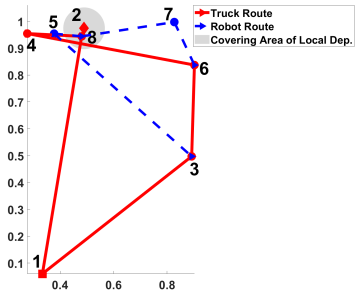


(c) RADP-2 (Optimal Solution = 0.8059)

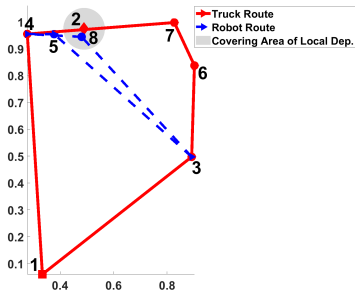
Figure 4.13. Solution of Experiment 1 of Table 4.2



(a) TSP (Optimum Solution = 1.2738)



(b) RADP-1 (Optimum Solution = 1.1307)



(c) RADP-2 (Optimum Solution = 1.1332)

Figure 4.14. Solution of Experiment 5 in Table 4.3

While the optimum solutions for both RADP-1 and RADP-2 in Experiment 5 of Table 4.3 were found within the computational time limit, the solutions themselves are quite distinct. RADP-1 outperforms RADP-2 in this case. The graphical representation of the solution is displayed in Figure 4.14, with sub-figures (a), (b), and (c) displaying the TSP, RADP-1, and RADP-2, respectively. In the RADP-1 solution, the truck route follows the sequence 1-8-4-6-3-1, and both robots are launched at customer node 8 but collected at different nodes (one at node 6 and the

Table 4.3: Solutions of RADP-1 Vs RADP-2, Av. Dist. = (3000m - 4500m)

Exp	Model	No. of Cust.	CPLEX MILP Solutions					TSP Sol. (h)	Savings Over TSP (%)	Optimality Gap %
			Obj. Val (h)	Comp.time (Sec.)	Service Facility					
					Depot	Robot	Truck			
1	RADP-1	6	1.1128	35.76	0	3	3	1.3187	15.61	0.00
	RADP-2		1.1128	875.39	0	3	3			
2	RADP-1	6	1.2832	25.13	1	2	3	1.4003	8.36	0.00
	RADP-2		1.2832	445.16	1	2	3			
3	RADP-1	6	1.2226	22.78	1	2	3	1.3599	10.10	0.00
	RADP-2		1.2226	167.53	1	2	3			
4	RADP-1	6	0.8477	49.18	1	2	3	1.0639	20.32	0.00
	RADP-2		0.8477	7200.00	1	2	3			
5	RADP-1	6	1.1307	100.37	0	2	4	1.2738	11.23	0.00
	RADP-2		1.1332	5673.86	0	2	4		11.04	
6	RADP-1	6	0.9304	81.22	0	3	3	1.2249	24.04	0.00
	RADP-2		0.9304	5934.62	0	3	3			
7	RADP-1	6	1.1345	35.38	0	3	3	1.3162	13.80	0.00
	RADP-2		1.1345	5486.44	0	3	3			
8	RADP-1	6	1.2038	33.65	1	3	2	1.4309	15.87	0.00
	RADP-2		1.2038	444.00	1	3	2			
9	RADP-1	6	1.0372	32.56	1	2	3	1.1961	13.28	0.00
	RADP-2		1.0372	1563.97	1	2	3			
10	RADP-1	6	1.1937	24.13	0	2	4	1.3711	12.94	0.00
	RADP-2		1.1937	451.59	0	2	4			
11	RADP-1	7	**1.3922**	1289.47	0	2	5	1.5206	8.44	0.00
	RADP-2		1.4707	7200.00	0	2	5		3.28	90.70
12	RADP-1	7	**1.0572**	606.48	0	3	4	1.3601	22.27	0.00
	RADP-2		1.1505	7200.00	1	2	4		15.41	90.33
13	RADP-1	7	1.1436	227.86	1	2	4	1.34068	18.71	0.00
	RADP-2		1.1436	7200.00	1	2	4		18.71	81.14
14	RADP-1	7	**1.1153**	432.47	0	4	3	1.4208	21.50	0.00
	RADP-2		1.2197	7200.00	0	2	5		14.15	81.45
15	RADP-1	7	**1.2415**	208.52	0	3	4	1.4837	16.32	0.00
	RADP-2		1.2826	7200.00	0	2	5		13.55	84.05
					Average Savings			**RADP-1**	**15.52**	
								RADP-2	**14.03**	

other at node 3). This highlights the flexibility of RADP-1 in selecting the launch and collection nodes for robots. With RADP-1, robots can be launched at the same node and collected at distinct nodes, or they can be launched at distinct nodes and collected at the same node. In contrast, in the RADP-2 solution, the truck route follows 1-3-6-7-4-1, and robots are both launched at node 3 and collected at node 4. Also, note that the truck does not visit the local depot but has its route pass close to the local depot. The operations in RADP-2 only allow robots to be launched and collected at the same node, potentially excluding operations that could be part of the optimal solution. Furthermore, considering these operations as part of the feasible operations can make the model much more challenging to solve.

4.5 Summary

The Robot-Assisted Delivery problem (RADP) integrates traditional truck logistics with the inclusion of robots and a local depot into a comprehensive model. Two Mixed Integer Linear Programming Models (MILP), RADP-1 and RADP-2, were introduced to optimize delivery schedules with the goal of minimizing the total working span of the truck. Numerical results consistently demonstrate that both models produce identical outcomes when they converge on an optimal solution. However, RADP-1 provides a better solution in cases where the optimum solution involves launching or collection of two robots at different nodes, a scenario not allowed in RADP-2, which only permits such actions at the same node. Additionally, RADP-1 generally proves more computationally tractable than RADP-2, owing to the exponential growth of feasible operations in the latter.

Optimal solutions for problems featuring up to 8 nodes are attainable with RADP-1 using CPLEX, whereas RADP-2 achieves optimal solutions for scenarios involving only 7 nodes within a 2-hour computational time limit.

The operational efficiency of robots exhibits a higher performance in densely populated areas compared to sparser regions due to their travel distance limitations and slower speeds relative to trucks. Conversely, solving problems in dense areas poses computational challenges due to the multitude of possible robot operations that satisfy the distance constraints. As the problem size expands, the feasible number of robot operations grows exponentially, rendering it impractical to solve problems of practical size within reasonable computational time. Consequently, heuristic approaches to expedite solutions for larger instances are proposed in the following chapter.

Chapter 5

Heuristics Approaches To Expedite RADP Solutions

5.1 Heuristics

In the previous chapter, we have discussed solutions for RADP-1 and RADP-2 for small instances. Exact Solutions for both models could not be obtained for practical size problems with CPLEX. As RADP is a derivative of TSP, a problem known to be NP-hard, RADP inherits the NP-hard complexity as well. To tackle real-world instances, heuristics approaches come into play. In this chapter, we introduce two heuristics methods aimed at expediting solutions for practical-scale instances.

5.1.1 Greedy Heuristics By Prioritizing Operations (P-Heur)

To achieve the aim of improving overall efficiency, one consideration is to increase the number of fulfilled orders per unit time. Inspired by this idea, a greedy

heuristic is proposed in this section, centered around the concept of "node-time ratio", which entails dividing the number of nodes covered by an operation by the duration of that operation. The ranking system gives preference to operations with higher ratios. It is important to note that P-Heur is specifically designed for the RADP-2 model.

Algorithm 7 outlines the steps of the greedy heuristic. Firstly, to reduce the problem size (the number of feasible operations) we assume that every node that is within the covering range of the local depot must be serviced by the local depot (line 3-5 of Algorithm 7). Any operation containing the distribution centre has only one node shared by another operation. Its ratio is determined by subtracting 0.5 from the number of its covered nodes and dividing the result by its completion time. In contrast, operations not involving the warehouse have both their start and end nodes shared by other operations. For these operations, the ratio is determined by subtracting 1 from the number of their covered nodes and dividing by their completion time (as described in line 8 of Algorithm 7). Then, we select the most preferred operation, i.e., the one with the highest node-time ratio (line 12-13 of Algorithm 7); any operations containing nodes already covered by the selected operation are removed from the list (line 14-20 of Algorithm 7). If the resulting number of operations in the reduced list is below a specified threshold value, we use CPLEX to solve the reduced RADP-2 with selected operations and remaining ones in the *OperList* (line 22 of Algorithm 7). However, if the number of operations exceed the threshold, we repeat the process to select the next most preferred operation from the reduced list and remove any operations that cover nodes already accounted for by the second selected operation to form the refreshed

reduced list. This step continues until the number of operations in the reduced list fall below the threshold, which, in our experiments in section 5.2.1, is set to 500 for CPLEX to deliver efficient solutions. This value is independent of the size of the problem. For larger problems with thousands of feasible operations, the algorithm continuously selects the best feasible operation and reduces the number of feasible operations by removing operations containing nodes that are already covered by the selected operations until the number of remaining feasible operations is less than a value (500) that can be solved by CPLEX. This value remains the same irrespective of the size of the problem. The only difference is that larger problems with a large number of feasible operations require more iterations of selecting the best operations and reducing the number of operations until the list of operations is fewer than 500, so that it can be solved by CPLEX. Note that, to ensure feasibility, all operations containing only two truck nodes are kept in the reduced operations list (second condition in line 15 of Algorithm 7).

5.1.2 K-Means Decomposition Heuristics (K-Heur)

K-Means is a widely employed clustering algorithm with the primary objective of grouping similar elements or data points into clusters. It addresses the clustering problem by minimizing the sum of squared errors (SSE) [70]. The purpose of clustering orders is to decompose the problem into more manageable subgroups that can be easily solved using commercial solvers. The choice of the number of clusters depends on the scale of the main problem. Clustering is performed in such a way that exact solutions to the sub-problems can be obtained within a reasonable time frame. The K-Means clustering is implemented using PyCharm IDE and

Algorithm 7 A Greedy Heuristic Based on Operations Prioritization

Input: All feasible operations generated by Algorithms 1-6 denoted by *OperList*, with their duration denoted by $t_{OperList}$

Output: A truck and robot tour covering all nodes and its completion time

Define $\mathcal{S}$ = set of nodes that can be covered by any local depot
for $i = 1$ to number of rows of *OperList* **do**
  if $OperList(i) \cap \mathcal{S} \neq \emptyset$ **then** ▷ removes all operations from *OperList* containing nodes that can be covered by local depot
    Remove row i from *OperList*
    Remove row i from $t_{OperList}$
  else ▷ determines the ratio for prioritizing operations
    $n(OperList(i))$ = the number of nodes covered by operation $OperList(i)$
    $r(i) = \begin{cases} \frac{n((OperList(i))-0.5}{t_{OperList(i)}}, & \text{If OperList(i) contains the warehouse,} \\ \frac{n((OperList(i))-1}{t_{OperList(i)}}, & \text{otherwise} \end{cases}$
  end if
end for
while number of rows in *OperList* > 500 **do**
  i^* = index of the maximum value of $r(i)$
  $x_{o(i^*)} = 1$ ▷ operation i^* is selected as a part of the optimal solution
  for $i = 1$ to number of rows in *OperList* **do**
    if $OperList(i) \cap OperList(i^*) \neq \emptyset$ & $length(OperList(i)) \geq 3$ **then**
      Remove row i from *OperList*
      Remove row i from $t_{OperList}$
      Remove row i from r
    end if
  end for
end while
Fix x values of selected operations to 1 in the RADP-2 model with reduced *OperList* and solve using CPLEX
Return the final truck and robot tour and its completion time

the Anaconda Python distribution with Python 3.8. As K-Heur necessitates exact solutions for clusters, the more efficient RADP-1 is employed instead of RADP-2 to reduce computational times.

Algorithm 8 outlines the steps of the heuristic based on decomposition and solution of sub-problems. We first begin by performing K-Means clustering on our data points to decompose the problem into K clusters (line 1-3 of Algorithm 8). Numerical experiments conducted in Section 4.4.1 revealed that exact solutions for RADP-1 can be obtained for seven nodes within a short computational time. Therefore, if a cluster contains more than seven nodes, we redistribute the excess to the nearest cluster with fewer than seven elements (lines 5-10 of Algorithm 8). The next step is to use CPLEX to solve the RADP-1 model for each cluster to find the best combinations and sequences of servicing the nodes involved (line 12-17 of Algorithm 8). Finally, set the x values corresponding to solutions of clusters to 1 in the RADP-2 model and solve using CPLEX with operations containing only two truck nodes obtained in Algorithm 1 (line 18 of Algorithm 8).

This final step will connect, in the best possible way, the solution segments obtained for each cluster together and force the route to start and finish at the central depot {0}.

Algorithm 8 Heuristic Based on Decomposition (K-Heur) and re-connecting solution segments

Input: nC,nD,nR,O_T,O_R,D_{max},V_T,V_R,h,λ_{ij}, Customer location sets $X \subseteq \mathbb{R}^2$; all feasible operations generated by Algorithm 1 denoted by $Op2$, with their duration denoted by t_2.

Output: A truck and robot tour covering all nodes and its completion time

1: $n_{max} \longleftarrow 7$ ▷ maximum number of elements in every cluster = 7
2: Set $K = \lceil \frac{|nC+nD|}{n_{max}} \rceil$ ▷ calculate the number of clusters needed
3: Perform K-Means clustering algorithm on X to decompose the problem into K clusters, $X_1, X_2, ...X_K$
4: Calculate $n(X_i)$ = number of elements in cluster, $i = 1, ..., K$
5: **for** $i = 1$ to K **do**
6: **while** $n(X_i) > n_{max}$ **do** ▷ Ensures that the number of data points in each cluster does not exceed 7
7: Move a data point from cluster i that is nearest to the closest cluster whose $n(X_i) < n_{max}$
8: Update $n(X)$ of both clusters
9: **end while**
10: **end for**
11: **for** $k = 1$ to K **do**
12: **for** each x_i in X_k **do**
13: Calculate Euclidean Distance, $d_{i,j} = ||x_j - x_i||^2$
14: **end for**
15: Call RADP-1 model to find the optimal solution of cluster, i
16: **end for**
17: Fix the corresponding x values of the solutions to 1 in the RADP-2 model with two-nodes operations ($Op2$) prepared in Algorithm 1 and solve using CPLEX.
18: Return the final truck and robot tour and its completion time

5.1.3 Demonstration of the Heuristic Approaches with an Example

In this section, we illustrate the solution approaches of P-Heur and K-Heur using an example instance comprising a single local depot and 15 customers, with an average distance of 996 meters between orders. The graphical representation of the service area for the experiment is shown in Figure 5.1. Node 1 functions as the distribution center, serving as both the starting and terminating point for the truck's route, while node 2 designates the local depot. All other nodes represent customers.

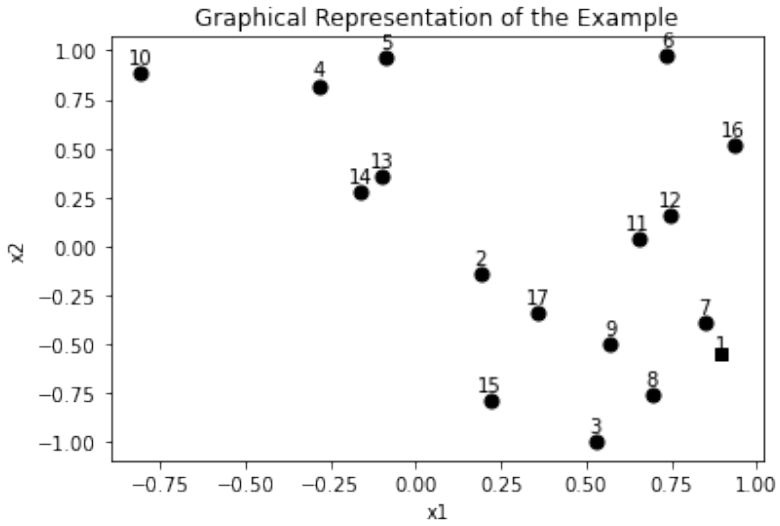


Figure 5.1. Graphical Representation of the Service Area

5.1.3.1 Using Greedy Heuristic By Prioritizing Operations (P-Heur)

We begin the process by generating a list of all feasible operations, denoted as *OperList*, following the steps outlined in Algorithms 1 - 6. To ensure that every node within the covering range of the local depot is serviced by the depot, we remove all operations containing nodes covered by the depot (lines 3-5 of Algorithm 7). In the specific example presented, none of the customers fall within the local

depot's range. Next, we determine the node-time ratio as explained in line 8 of Algorithm 7. Operations containing the warehouse have only one node shared with their subsequent or preceding operation, and the total number of nodes for such operations is determined by subtracting 0.5 from the count of their covered nodes. In contrast, operations not involving the warehouse have both their start and end nodes shared with their preceding and succeeding operations, respectively. For these operations, the total number of nodes is determined by subtracting 1 from the count of their covered nodes. Subsequently, we select the most preferred operation, determined by the highest node-time ratio, from the list of operations (lines 12-13 of Algorithm 7). In this case, the most preferred operation encompasses truck nodes 7, 11, and 15, along with robot nodes 3, 8, 9, and 17, as depicted in Figure 5.2 (highlighted in red). We then streamline the list of operations by removing any operation containing nodes already covered by the selected operation (lines 14-20 of Algorithm 7). Given that the number of operations in the reduced list exceeds 500, we proceed to select the next most preferred operation from this reduced list (lines 12-13 of Algorithm 7). The second selected operation encompasses truck nodes 5, 6, 16, and 7, as well as robot nodes 4, 12, 13, and 14, illustrated in Figure 5.2 (highlighted in blue). Subsequently, we remove any operations from the list that contain nodes already covered by the second selected operation. With the resulting number of operations in the updated list falling below 500, we employ CPLEX to solve the reduced RADP using the selected operations and the remaining ones in *OperList* (line 22 of Algorithm 7). The resulting solutions are visually represented in Figure 5.2. This approach allows us to determine the solution to the example in less than 10 seconds.

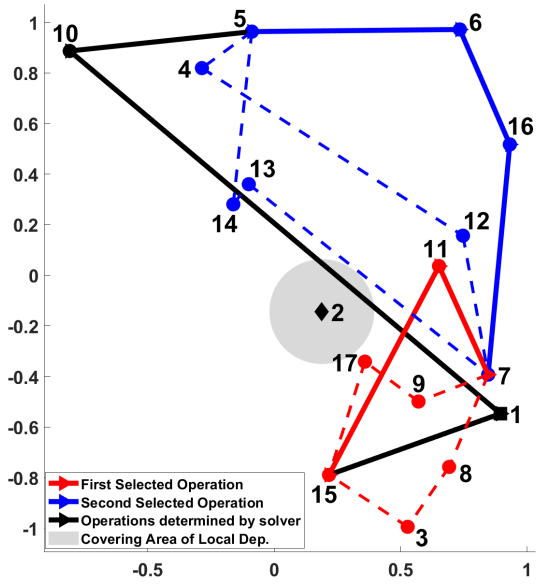


Figure 5.2. P-Heur Solution With Objective Value = 1.4353

5.1.3.2 Using The Clustering Approach (K-Heur)

Based on the computational experiments conducted in Section 4.4, optimal solutions for instances with up to 8 nodes can be solved reasonably fast for RADP-1 using CPLEX. Therefore, in our approach, we set the maximum number of nodes in each cluster to 7 (Line 1 of Algorithm 8). We then calculate the number of clusters required by dividing the total number of nodes by 7 and rounding up the result to the nearest integer (Line 2 of Algorithm 8). Given that there are 15 customers and 1 local depot in this particular scenario, the number of clusters is calculated as 3. We proceed by performing the K-Means clustering algorithm on the problem to decompose it into 3 clusters (Line 3 of Algorithm 8). The composition of these clusters is as follows: Cluster 1 = {4, 5, 10, 13, 14}, Cluster 2 = {6, 11, 12, 16} and Cluster 3 = {2, 3, 7, 8, 9, 15, 17} as illustrated in Figure 5.3. The mark "**x**" in the figure indicates the centroid (centre point) of each cluster. For each of the three

clusters, we calculate the Euclidean distance between every pair of data points, and subsequently, the RADP-1 model is utilized to find the optimal solutions (Lines 12-17 of Algorithm 8). The solutions for each cluster are presented in Figure 5.4a. In the final step, we set the corresponding x-values of all cluster solutions to 1 in the RADP-2 model and solve it with two-node operations as outlined in Algorithm 3 (Line 15 of Algorithm 8). The complete solution is presented in Figure 5.4b. Although this approach necessitates exact solutions for three independent clusters, the solution time is remarkably quick, taking only a few seconds due to the utilization of RADP-1 in determining the optimal solutions of the clusters.

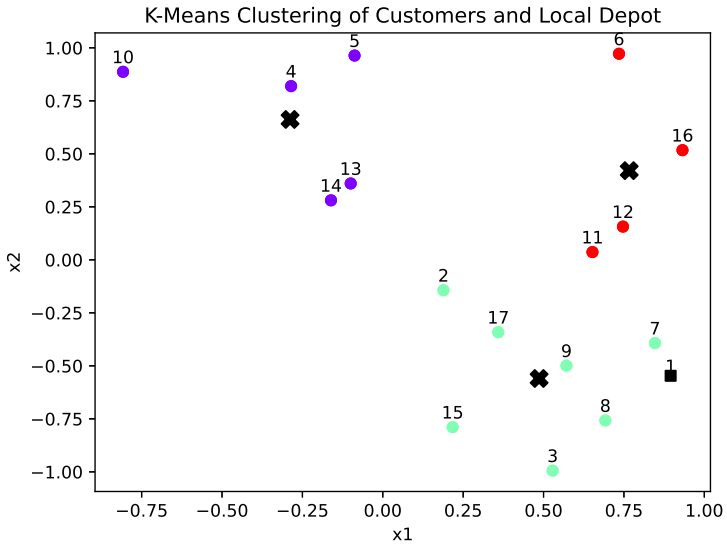
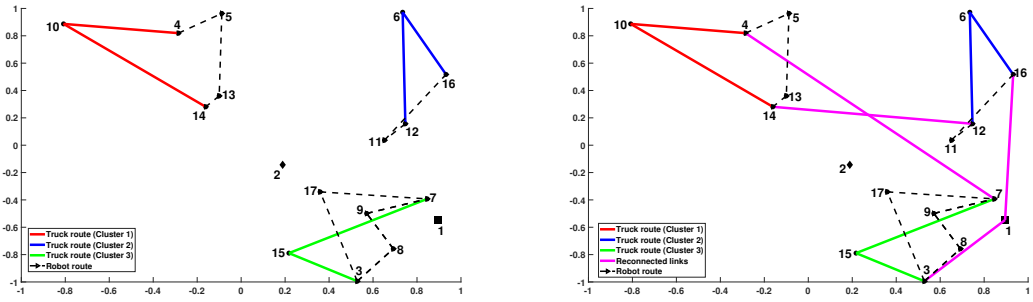


Figure 5.3. K-Means Clustering



(a) Cluster Solutions **(b)** Connecting Clusters and the Warehouse

Figure 5.4. K-Heur Solution With Objective Value = 1.4997

5.2 Results

In this section, we present a comprehensive overview of the results derived from our numerical experiments. To evaluate the performance of the heuristic, we conducted a series of experiments utilizing both the Mixed Integer Linear Programming (MILP) model (RADP-2) and the heuristic approaches outlined in the preceding section. Furthermore, we expand our analysis to encompass larger instances, providing results for the heuristics and drawing comparisons between the two approaches. The implementation of these heuristics is carried out in MATLAB, and their solutions are obtained using CPLEX.

5.2.1 MILP and Greedy Heuristics Solutions of the RADP

In this subsection, we conduct experiments using the MILP model and the Greedy heuristic approach to test the performance of the heuristic. Similar to section 4.4.1, the experiments are categorized into three groups: The first group contains instances with an average distance between orders ranging from 420m to 850m, the second group from 1100m to 2600m, and the third group from 3000m to 4500m. Our

experiments consist of 6 customers and 1 local depot; other parameters remain the same as discussed in Section 4.4.1. The results of these experiments are presented in Tables 5.1, 5.2, and 5.3 for the first, second, and third groups, respectively.

Column 2 of these tables displays the solution for the Traveling Salesman Problem (TSP), where robots are not used, but local depots are employed alongside the truck. Columns 3 to 8 represent the solution, computation time, percentage savings over TSP, the number of customers served by the local depot, by robots, and by the truck, respectively, for the MILP model (RADP-2). Columns 9 to 14 provide the same information for the Operations Prioritization Heuristic (P-Heur), as discussed in the previous section. The last column displays the optimality gap between the Prioritization Heuristic and the MILP (RADP-2) model which measures the performance of the heuristic. The gap is manually calculated using the formula:

$$\frac{(\theta_{PHeur} - \theta_{RADP-2})}{\theta_{RADP-2}} * 100 \tag{5.1}$$

Where:

- θ_{RADP-2} is optimum objective value of the RADP-2 model.
- θ_{PHeur} is the objective value of the P-Heur.

As previously discussed in Section 3.4.1, smaller optimality gap indicates that the current solution is closer to being optimal. It is also noted from Section 4.4.1 that the RADP-2 model effectively handles problems with up to 7 nodes, including the local depot, within a two-hour computation time window. The solution time is closely related to the average distance between customers. Specifically, instances in the first group require longer computation times due to a higher number

of feasible robot operations compared to the second and third groups, where robot travel between customers is often infeasible, considering the limited travel distance of robots. Savings are also more significant in the first and second groups, as robots are employed more intensively. With the P-Heur heuristic approach, solutions matching the optimum RADP-2 results were identified in 5 out of the 30 experiments (experiment 1, 4, and 6 of Table 5.1, experiment 6 of Table 5.2, and experiment 2 of Table 5.3). Remarkably, these optimal solutions were found in less than a second, while the RADP-2 took hours. However, notable sub-optimality gaps were observed in some of these small-scale instances compared to the RADP-2-optimal solutions. Nevertheless, it delivered significant savings compared to the TSP in the majority of experiments, with the highest level of savings (35.26%) achieved in experiment 5 of Table 5.1 and an optimality gap of (2.90%) compared to the optimum RADP-2 solution.

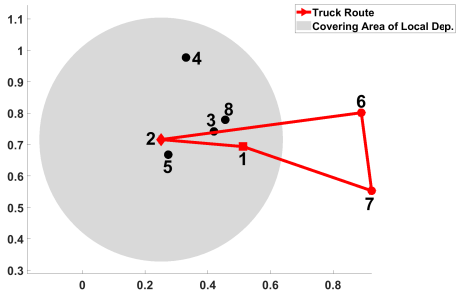
Robots were extensively utilized in experiments 5 of Table 5.1, experiment 8 of Table 5.2 and experiment 2 of Table 5.3, yielding solutions with small optimality gaps compared to the optimum RADP-2 solutions. More customers are served by trucks in the experiments presented in Table 5.3 than in Table 5.2 and Table 5.1. This could be attributed to a larger spacing between customers, leading to numerous infeasible robot operations.

Figure 5.5 provides a graphical representation of Experiment 4 in Table 5.1, with sub-figures (a), (b), and (c) displaying TSP, RADP-2, and P-Heur solutions, respectively. In this specific example, the spacing between customers and the local depot is close, resulting in several customers (3, 4, 5, and 8) being serviced from the local depot. Only one customer is served by the robot using both RADP-2 and

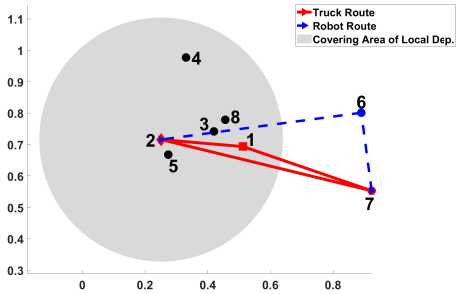
P-Heur, and the majority of customers are served from the local depot, leading to low savings (12.06%). The heuristic solution initiates by selecting the truck route segment 1-2 and eventually converges to a solution that matches the optimum RADP-2 solution. Figure 5.6 provides a graphical representation of Experiment 5 in Table 5.1, with sub-figures (a), (b), and (c) displaying the TSP, RADP-2, and P-Heur solutions, respectively. In the RADP-2 solution, the truck route follows the sequence 1-8-5-6-1, and the robots are launched at customer node 8 to follow the sequences 8-3-4-6 and 8-7-6. Despite the customer at node 7 being within the covering range of the local depot (node 2), it is better served by the robot than by the truck visiting the local depot. The heuristic solution initiates by selecting the operation containing the truck route segment 2-5-6, along with robot operations 2-4-8-6 and 2-3-6 as part of its solution. Subsequently, the solver determines the operations that best connect to the selected operation to complete the tour.

Table 5.1: Solutions of RADP-2 and P-Heur, Av. Dist. = (420m - 850m)

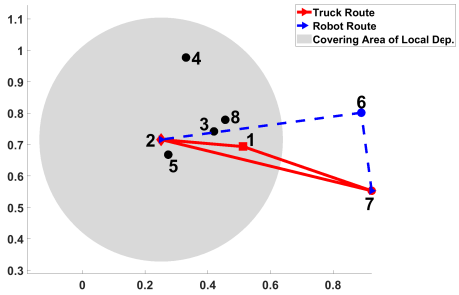
Exp	TSP	RADP-2 Solution						P-Heur Solution						Opt.
	Sol.	Obj.	Comp.	Savings	Service Facility			Obj.	Comp.	Savings	Service Facility			Gap
	(hrs)	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(%)
1	0.7091	0.5154	6820.17	27.32	2	2	2	0.5154	0.17	27.32	2	2	2	0.00
2	0.7051	0.5113	6627.65	27.49	0	4	2	0.5382	0.15	23.67	2	2	2	5.00
3	0.6924	0.4825	7120.29	30.31	2	3	1	0.519	0.21	25.04	2	2	2	7.03
4	0.4683	0.4118	6718.59	12.06	4	1	1	0.4118	0.28	12.06	4	1	1	0.00
5	0.804	0.5054	6900.26	37.14	0	3	3	0.5205	0.15	35.26	1	3	2	2.90
6	0.5879	0.4353	5970.47	25.96	3	2	1	0.4353	0.26	25.96	3	2	1	0.00
7	0.7374	0.5061	7123.89	31.37	0	3	3	0.6051	0.18	17.94	2	2	1	16.36
8	0.7224	0.5336	6997.98	26.14	0	3	3	0.5437	0.15	24.74	2	2	2	1.86
9	0.7111	0.516	6782.15	27.44	2	2	2	0.5338	0.24	24.93	2	2	2	3.33
10	0.8156	0.5054	6300.16	38.03	0	3	3	0.6549	0.30	19.70	1	2	3	22.83
													Average Optimality gap	**5.93**



(a) TSP (Optimum Solution = 0.4683)

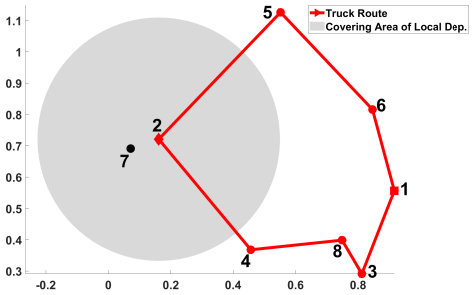


(b) RADP-2 (Optimum Solution = 0.4118)

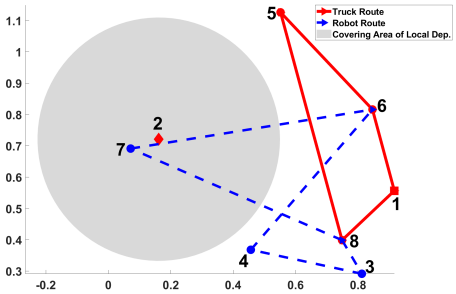


(c) P-Heur (Solution = 0.4118)

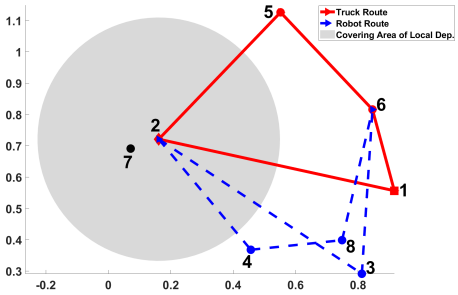
Figure 5.5. Solution of Experiment 4 of Table 5.1



(a) TSP (Optimum Solution = 0.8040)



(b) RADP-2 (Optimum Solution = 0.5054)



(c) P-Heur (Solution = 0.5205)

Figure 5.6. Solution of Experiment 5 of Table 5.1

The largest optimality gap (22.83%) is observed in Experiment 10 of Table 5.1. The graphical representation of the solution for this experiment is provided in Figure 5.7, with sub-figures (a), (b), and (c) displaying the TSP, RADP-2, and P-Heur solutions, respectively. In the RADP-2 solution, the truck route follows the sequence 1-3-5-6-1, and the robots are launched at customer node 3 to follow routes 3-8-6 and 3-7-4-6, after which they are collected at node 6. In contrast, P-Heur begins by selecting the operation with the truck route 4-5 and robot routes

Table 5.2: Solutions of RADP-2 and P-Heur, Av. Dist. = (1100m - 2600m)

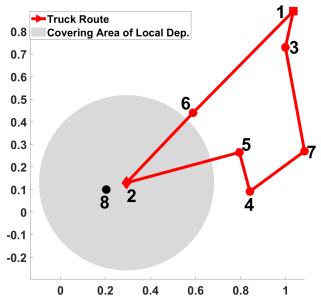
Exp	TSP	RADP-2 Solution						P-Heur Solution						Opt.
	Sol.	Obj.	Comp.	Savings	Service Facility			Obj.	Comp.	Savings	Service Facility			Gap
	(hrs)	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(%)
1	1.0369	0.8059	5362.64	22.28	0	2	4	0.8319	2.78	19.77	1	2	3	3.13
2	1.039	0.8561	1112.20	17.60	2	2	2	0.9482	0.22	8.74	2	1	3	9.71
3	0.9967	0.7205	5602.57	27.71	0	3	3	0.8589	0.32	13.83	1	2	2	16.11
4	1.0281	0.7291	5117.45	29.08	0	3	3	0.8618	0.41	16.18	1	2	3	15.40
5	0.9346	0.726	4718.37	22.32	0	3	3	0.7877	0.28	15.72	1	1	3	7.83
6	0.9449	0.8512	431.59	9.92	2	1	3	0.8512	0.25	9.92	2	1	3	0.00
7	1.1003	0.7853	5689.34	28.63	2	1	3	0.9927	0.15	9.78	1	2	3	20.89
8	1.1521	0.8649	5420.85	24.93	0	3	3	0.8704	1.77	24.45	1	3	2	0.63
9	1.1166	0.8807	5783.36	21.13	0	2	4	0.914	1.62	18.14	1	2	3	3.64
10	0.8712	0.5893	5823.78	32.36	0	3	3	0.6152	0.27	29.38	1	2	3	4.21
													Average Optimality gap	**8.16**

Table 5.3: Solutions of RADP-2 and P-Heur, Av. Dist. = (3000m - 4500m)

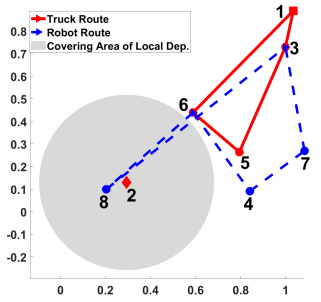
Exp	TSP	RADP-2 Solution						P-Heur Solution						Opt.
	Sol.	Obj.	Comp.	Savings	Service Facility			Obj.	Comp.	Savings	Service Facility			Gap
	(hrs)	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(hrs)	Time(s)	(%)	Dep.	Rob.	Tru.	(%)
1	1.3187	1.1128	875.39	15.61	0	3	3	1.1281	0.90	14.45	1	2	3	1.36
2	1.4003	1.2832	445.16	8.36	1	2	3	1.2832	0.30	8.36	1	3	3	0.00
3	1.3599	1.2226	167.53	10.10	1	2	3	1.3188	0.27	3.02	1	1	4	7.29
4	1.3356	1.1498	3727.80	13.91	1	2	3	1.2401	0.62	7.15	1	2	3	7.28
5	1.2738	1.1332	5673.86	11.04	0	2	4	1.2324	0.25	3.25	1	1	4	8.05
6	1.2249	0.9304	4934.62	24.04	0	3	3	1.1681	0.23	4.64	1	2	3	20.35
7	1.3162	1.1345	5486.44	13.80	0	3	3	1.2747	0.16	3.15	1	1	4	11.00
8	1.4309	1.2038	444.00	15.87	1	3	2	1.2422	0.76	13.19	1	2	3	3.09
9	1.1961	1.0372	1563.97	13.28	1	2	3	1.0659	0.94	10.89	1	2	3	2.69
10	1.3711	1.1937	451.59	12.94	0	2	4	1.262	0.67	7.96	1	1	4	5.41
													Average Optimality gap	**6.65**

4-6-5 and 4-7-5. Unlike RADP-2, which considers the best possible combination of operations, P-Heur starts by choosing the least expensive operation without considering the composition of potential operations that make up the tour.

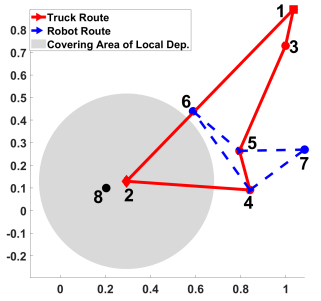
Similarly, the least optimality gap (0.63%) is observed in Experiment 8 of Table 5.2. Figure 5.8 provides the graphical representation of the solution, with sub-figures (a), (b), and (c) displaying the TSP, RADP-2, and P-Heur solutions, respectively. The RADP-2 and P-Heur solutions are highly comparable. The major difference lies in the waiting time of the truck at node 3; the truck waits longer in the P-Heur solution due to the robot traveling a greater distance along the sequence 2-7-6-3 compared to the RADP-2 solution where the same robot travels a shorter distance along the sequence 4-7-6-3. Additionally, in P-Heur, any customer within the local depot's covering range must be served from the local depot without considering the cost of the tour, whereas in the RADP-2, a depot is only used when it provides the best solution.



(a) TSP (Optimum Solution = 0.8156)

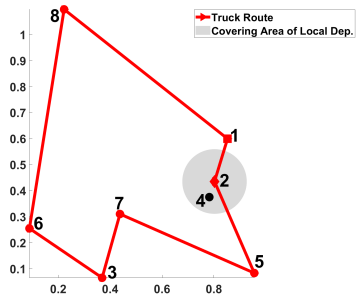


(b) RADP-2 (Optimum Solution = 0.5054)

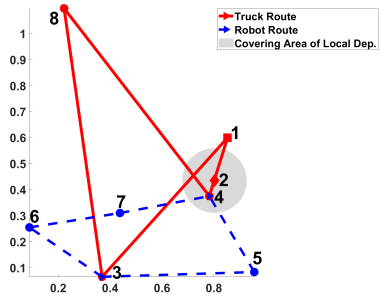


(c) P-Heur (Solution = 0.6549)

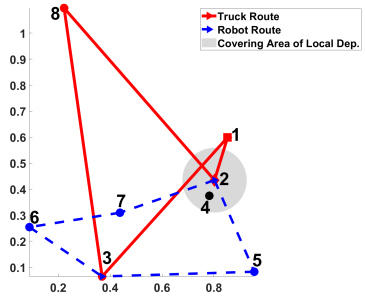
Figure 5.7. Solution of Experiment 10 of Table 5.1



(a) TSP (Optimum Solution = 1.1521)



(b) RADP-2 (Optimum Solution = 0.8649)



(c) P-Heur (Solution = 0.8704)

Figure 5.8. Solution of Experiment 8 of Table 5.2

5.2.2 P-Heur and K-Heur Solutions for Larger Instances

In this subsection, we present the solutions for RADP-1 and RADP-2 models in larger instances using the two heuristic approaches discussed in Section 5.1. The experiments were conducted with 17 customers, 1 local depot, and 2 robots, while maintaining other parameters as discussed in Section 4.4.1. The instances are categorized into three groups, each comprising 10 instances. The first group

includes instances with average distances between orders ranging from 800m to 1100m, the second group from 2400m to 3200m, and the third group from 3500m to 5000m. The results of these experiments are detailed in Tables 5.4, 5.5, and 5.6 for the first, second, and third groups, respectively. For each instance, we provide information on the total working time of the truck, computation time, the number of nodes served by the depot, the number of nodes served by robots, and the number of nodes served by the truck for both P-Heur and K-Heur. This comprehensive analysis offers insights into the performance and efficiency of the heuristic approaches across varied instances and distance scenarios.

Table 5.4: P-Heur and K-Heur Solutions, Av. Dist. = (800m - 1100m)

Exp	P-Heur Solution					K-Heur Solution				
	Obj.	Comp.	Service Facility			Obj.	Comp.	Service Facility		
	(hrs)	Time(s)	Dep.	Rob.	Tru.	(hrs)	Time(s)	Dep.	Rob.	Tru.
1	**1.0989**	6.12	2	9	6	1.2059	53.23	0	9	8
2	**1.0831**	23.75	4	7	6	1.2091	67.12	3	9	5
3	**1.1098**	193.57	6	5	6	1.1675	336.01	4	7	6
4	**0.9088**	132.69	6	6	5	1.1649	209.95	4	6	7
5	**1.0888**	78.71	4	7	6	1.0922	71.29	4	7	6
6	**0.9756**	637.75	6	6	5	1.0100	24.38	6	6	5
7	**0.9777**	136.23	4	8	5	0.9841	53.82	4	8	5
8	**1.1418**	4.82	2	9	6	1.2147	51.94	2	9	6
9	**1.0447**	263.44	4	7	6	1.0791	67.33	4	7	6
10	**0.9553**	295.70	4	8	5	1.1199	36.03	4	8	5

P-Heur consistently demonstrates superior solutions across all experiments conducted on instances belonging to the first group, as well as in four instances from the second group (Exp 1, 4, 7, and 8). However, it exhibits less effectiveness in seven instances within the third group. A noteworthy trend is observed: K-Heur tends to perform less efficiently in scenarios with shorter average distances between orders. In these situations, there is a higher likelihood of data points that

Table 5.5: P-Heur and K-Heur Solutions, Av. Dist. = (2400m - 3200m)

Exp	P-Heur Solution					K-Heur Solution				
	Obj.	Comp.	Service Facility			Obj.	Comp.	Service Facility		
	(hrs)	Time(s)	Dep.	Rob.	Tru.	(hrs)	Time(s)	Dep.	Rob.	Tru.
1	**1.8590**	9.40	2	8	7	1.8687	54.10	0	8	9
2	1.7839	54.77	2	6	9	**1.7717**	203.72	2	7	8
3	1.7049	507.14	3	6	8	**1.6778**	105.39	3	6	8
4	**1.6598**	273.68	3	7	7	1.8574	91.86	0	8	9
5	1.9211	53.23	2	7	8	**1.8395**	282.85	0	8	9
6	2.0285	23.49	2	6	9	**1.9039**	71.05	0	8	9
7	**1.6975**	13.30	2	8	7	1.8241	209.66	2	7	8
8	**1.5025**	419.28	3	8	6	1.7649	58.91	0	9	8
9	1.8965	67.52	3	6	8	**1.8260**	50.90	3	5	9
10	1.8891	51.84	2	7	8	**1.8326**	84.04	2	6	9

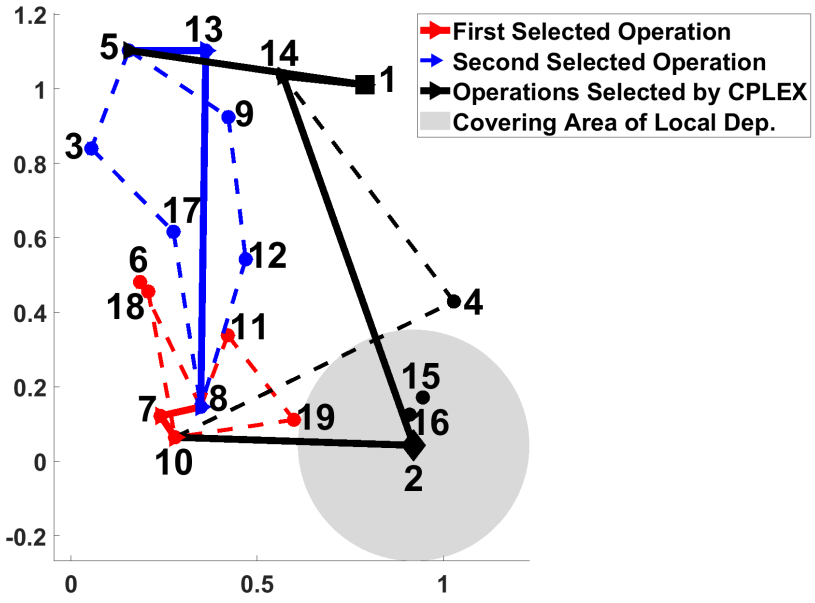
Table 5.6: P-Heur and K-Heur Solutions, Av. Dist. = (3500m - 5000m)

Exp	P-Heur Solution					K-Heur Solution				
	Obj.	Comp.	Service Facility			Obj.	Comp.	Service Facility		
	(hrs)	Time(s)	Dep.	Rob.	Tru.	(hrs)	Time(s)	Dep.	Rob.	Tru.
1	2.3142	1.20	2	6	9	**2.2572**	69.66	2	6	9
2	2.4395	17.71	2	6	9	**2.2670**	54.77	2	6	9
3	2.6173	737.51	1	5	11	**2.4598**	114.24	1	7	9
4	2.3599	11.53	2	5	10	**2.2888**	64.87	1	8	8
5	**2.2848**	18.26	2	6	9	2.3193	54.25	0	8	9
6	**2.2978**	40.47	3	5	9	2.4190	61.58	0	7	10
7	2.6843	356.20	1	3	13	**2.4974**	36.82	0	7	10
8	2.2647	407.98	2	7	8	**2.0941**	48.98	2	6	9
9	**2.2889**	1430.11	3	6	8	2.3306	99.03	2	7	8
10	2.4943	828.20	2	6	9	**2.3501**	19.58	0	7	10

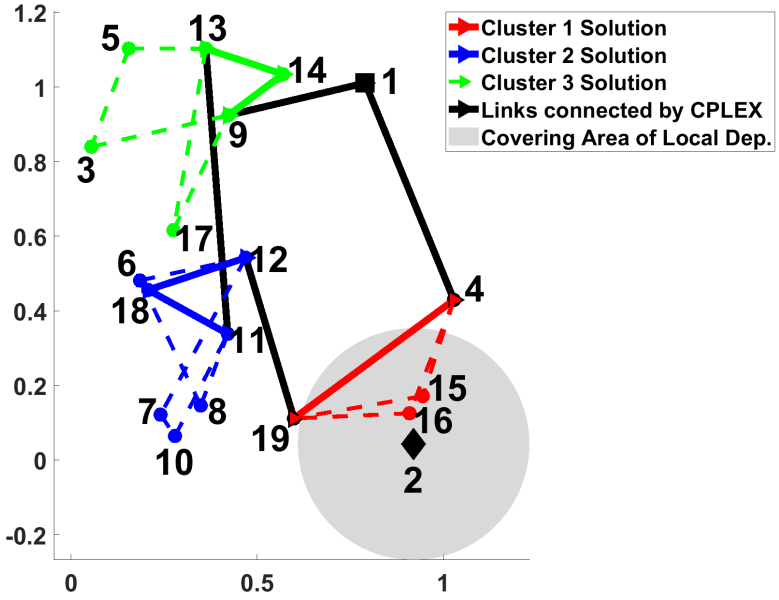
could potentially yield improved solutions being part of different clusters. This observation suggests that K-Heur is better suited for scenarios characterized by higher average distances between orders. Despite differences in solutions, both P-Heur and K-Heur demonstrate faster computational times, efficiently solving problems for which solutions cannot be found for several hours using commercial solvers. Occasionally, P-Heur may have higher computational times, possibly due to the heuristic selecting operations covering only a few nodes, resulting in a large number of operations in the end that may require longer time to solve. Conversely, K-Heur is solved reasonably fast, owing to the efficiency of RADP-1.

The numerical solutions further reveal that K-Heur does not utilize local depots in 10 out of the 30 experiments, even in instances where multiple nodes fall within the coverage range of the local depot. This non utilization of local depots occurs because K-Heur only considers feasible operations within a cluster containing the local depot, and the optimal combination of operations within that cluster may not necessarily involve a visit to the local depot.

In addition, robots are more extensively utilized in instances characterized by smaller average distances between orders compared to those with larger spacing. As the average distance between orders increases, the utilization of robots and local depots tends to decrease. This decrease is attributed to the limited coverage range of the local depot and the limited travel distance of the robots. In scenarios with larger average distances between orders, the operational efficiency of robots and local depots diminishes, highlighting the impact of spatial factors on their effective utilization in logistics or operational contexts.



(a) P-Heur Solution (Obj. Val. = 1.0989)



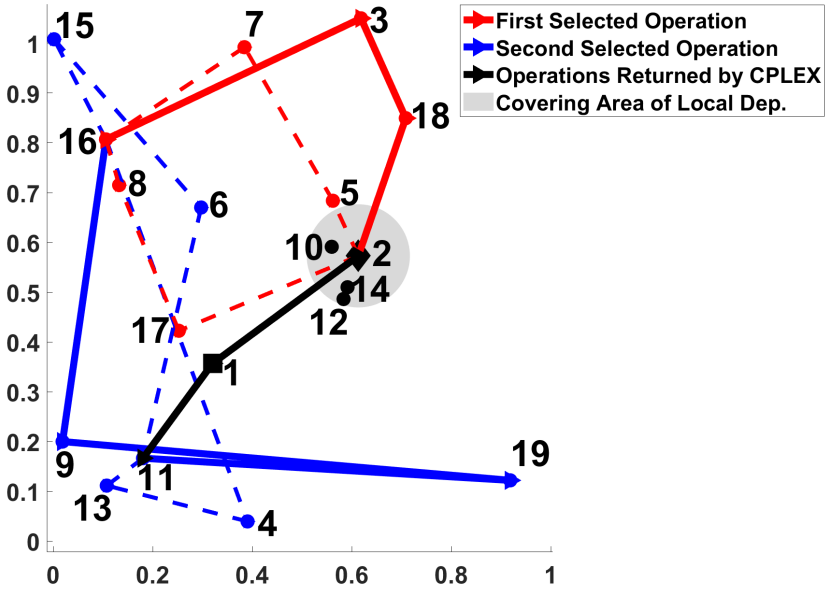
(b) K-Heur Solution (Obj. Val. = 1.2059)

Figure 5.9. K-Heur and P-Heur Solutions of Experiment 1 of Table 5.4

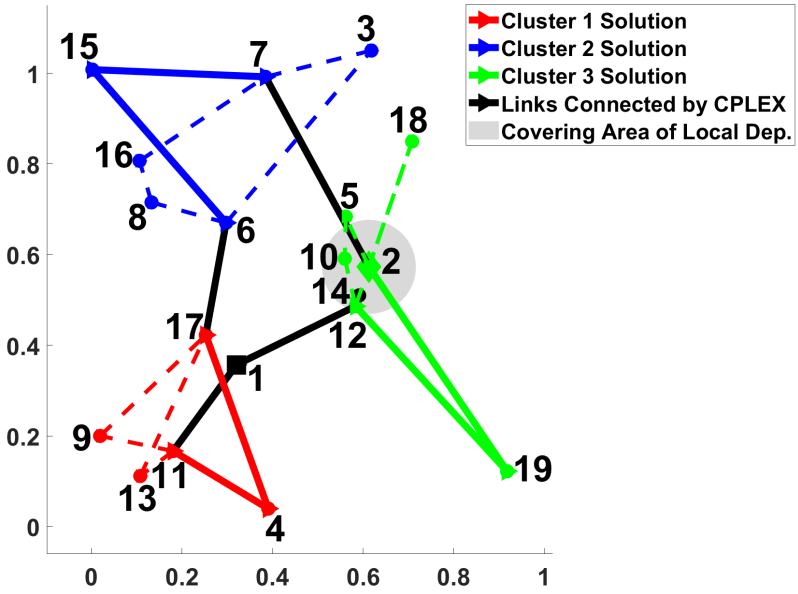
Figure 5.9 provides a graphical representation of experiment 1 from Table 5.4, with sub-figures (a) and (b) displaying the solutions obtained using P-Heur and K-Heur, respectively. In the P-Heur solution, the truck route follows the sequence 1-5-13-8-7-10-2-14-1, and links 8-7-10 constitute the truck route for the first selected

operation involving robot nodes 6, 18, 11, and 19. The second selected operation contains truck link 5-13-8, which is associated with robot nodes 3, 9, 12, and 17. Notably, customers at nodes 15 and 16 fall within the coverage range of the local depot (node 2) and are serviced by the depot. In contrast, for K-Heur, solutions are determined for each cluster independently, without considering the problem as a whole. Consequently, despite being within the coverage range of the local depot, nodes 15 and 16 are served by a robot in cluster 1, considered the optimal solution for that specific cluster. The two solutions in this scenario are entirely different, with P-Heur providing a better solution within a few seconds.

Figure 5.10 illustrates another example, showcasing the solution of experiment 8 from Table 5.5. Sub-figures (a) and (b) display solutions for P-Heur and K-Heur, respectively. In this specific instance, the average distance between customers is larger than the instance discussed earlier (Exp 1 of Table 5.4). For P-Heur, the truck route follows the sequence 1-2-18-3-16-9-19-11-1. Nodes 10, 12, and 14 receive service from the local depot. The first selected operation involves the truck link 2-18-3-16 with robot nodes 5, 7, 8, and 17. The second selected operation contains the truck link 16-9-19-11 with robot nodes 4, 13, 6, and 15. It is evident that the truck link 9-19-11 is considered as an inferior segment of the solution. For K-Heur, the truck route follows 1-11-4-17-6-15-7-2-19-12-1. Although nodes 10, 12, and 14 fall within the coverage range of the local depot, only 14 is served by the local depot, while 10 and 12 are served by the robot and truck, respectively. Similarly, P-Heur gives a better solution, but has higher computational time compared to K-Heur.



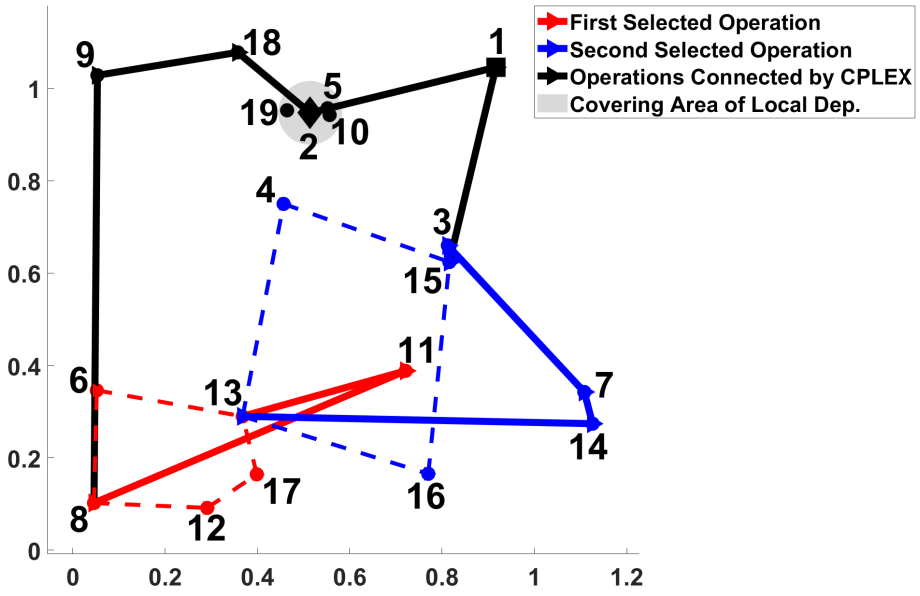
(a) P-Heur Solution (Obj. Val. = 1.5025)



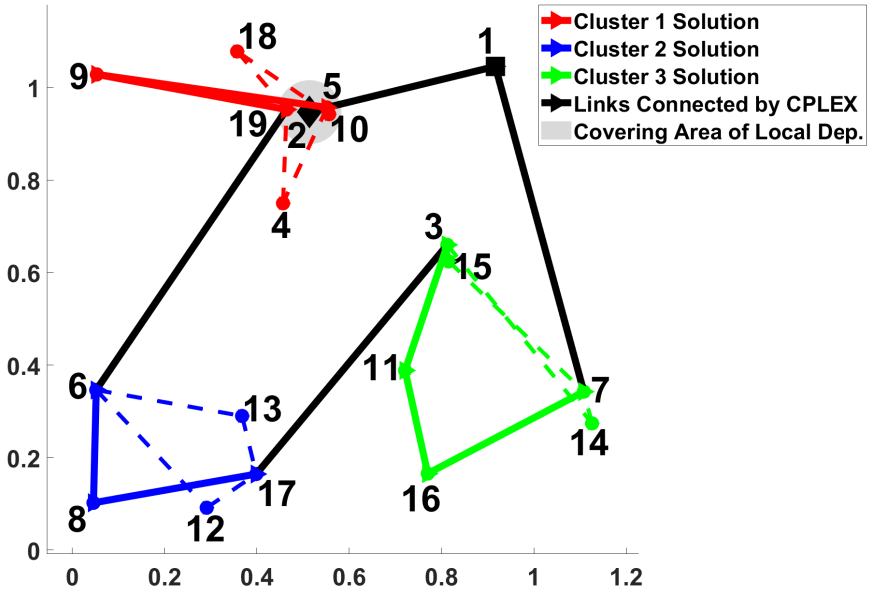
(b) K-Heur Solution (Obj. Val. = 1.7469)

Figure 5.10. K-Heur and P-Heur Solutions of Experiment 8 of Table 5.5

Finally, Figure 5.11 presents the solution for experiment 6 from Table 5.6, with sub-figures (a) and (b) depicting the solutions obtained by P-Heur and K-Heur, respectively. In the P-Heur solution, the truck follows the route 1-2-18-9-8-11-13-14-7-3-15-1. The first selected operation based on the heuristic involves the truck



(a) P-Heur Solution (Obj. Val. = 2.2978)



(b) K-Heur Solution (Obj. Val. = 2.4190)

Figure 5.11. K-Heur and P-Heur Solutions of Experiment 6 of Table 5.6

sequence 8-11-13 with robot nodes 6, 12 and 17.The second selected operation follows the truck sequence 13-14-7-3-15 with robot nodes 4 and 16. Customers at nodes 5, 10, and 19 are served by the local depot. For K-Heur, the truck route follows the sequence 1-5-9-19-6-8-17-3-11-16-7-1. The truck and the robots are used

simultaneously in each of the three clusters. Unlike P-Heur, the local depot is not used to service any customer. In this case, though with larger average distance between orders, P-Heur provides a better solution than K-Heur.

5.3 Summary

Heuristic approaches, specifically P-Heur and K-Heur, have been proposed to expedite the solution of practical-sized problems. P-Heur, a greedy heuristic, prioritizes operations based on the 'time-nodes ratio,' primarily working with RADP-2. In contrast, K-Heur employs a decomposition-based strategy: it clusters nodes using K-Means, determines optimal routes within each cluster via RADP-1, and initializes a reduced TSP with RADP-2 to link the optimal route segments.

The performance of P-Heur is assessed by comparing its solutions with the MILP solutions for instances of size 7 (including the local depot). In 5 out of the 30 experiments, P-Heur yields identical results to MILP, while in the remaining cases, there are small to moderate optimality gaps.

The evaluation of heuristic approaches on larger instances revealed that, in general, P-Heur outperformed K-Heur, especially in scenarios with shorter average distances between orders because K-Heur does not allow orders that are not far away to be grouped together. On the contrary, K-Heur showed superiority in cases characterized by larger average distances between orders. Importantly, P-Heur achieves less computational time in scenarios where the operations selected by the heuristic cover many nodes, leading to the removal of numerous operations from the list of operations under consideration. On the other hand, in scenarios

where the operations selected by P-Heur cover only a few nodes, it results in a larger pool of operations from which the model needs to make selections. For the K-Heur, the operational times are less because the solution of the clusters are found using RADP-1 which is more efficient than RADP-2 used in P-Heur.

Chapter 6

Conclusion

The transformative impact of online shopping, accelerated by the global COVID-19 pandemic, has reshaped the landscape of retail industries, particularly in developed countries. This shift, from traditional in-store shopping to the virtual realm, has not only boosted sales for retail companies but has also necessitated innovative solutions for last-mile deliveries. In this thesis, we explored two significant dimensions of this evolving paradigm: the integration of drones and robots into truck delivery systems and the optimization of delivery schedules involving them.

Delivery drones and robots serve as assisted delivery vehicles, complementing trucks due to their limited payload capacity and travel distance. In the majority of studies available in the literature, drones depart from and return to the truck at customer locations or depots after servicing customers. However, there are few studies that consider drones departing from or joining a truck while the truck is in transit along its route. Unlike existing approaches that rely on the optimal (sub-optimal) Traveling Salesman Problem (TSP) to design en-route operations, our novel approach explores the maximum cover range of nodes and

links serviced by drones as trucks traverse their routes. Three Mixed-Integer Linear Programming (MILP) models, derived from the Covering Salesman Problem (CSP), were introduced, each catering to different sets of drone operations.

The contributions of our study on drone-assisted deliveries beyond existing methodologies are:

- **Innovative Approach:** Unlike prior en-route operation models, which build upon the optimal TSP, our method starts by exploring the maximum cover range of nodes and links that can be serviced by a drone when the corresponding node or link is travelled by a truck, offering a fresh perspective.

- **Model Development:** Three MILP models, CSPNS-D-1, CSPNS-D-2, and CSPNS-D-3 addressing various drone operation scenarios, were developed and compared, providing managerial insights.

- **Heuristic Enhancement:** To tackle larger instances efficiently, a Link-removal heuristic was proposed, by strategically removing links unlikely to be chosen as part of the truck route, this heuristic significantly reduces solution times without compromising quality. Results demonstrate its effectiveness in obtaining better solutions for larger instances.

Numerical experiments conducted on both randomly generated and benchmark instances demonstrated the effectiveness of our approach, showing substantial improvements in delivery efficiency. For the random instances, we conducted 10 experiments of varying sizes up to a maximum of 40 orders, except for the 40-order scenario where optimum solutions for CSPNSD-3 and CSPNS-D-2 were not found. CSPNSD-3 consistently outperformed CSPNSD-2 and CSPNSD-1, aligning with

our expectations as it considers a broader covering range in the model. In the benchmark datasets, comprising 10 and 20 orders, each with 30 instances sourced from [61], results indicated that CSPNS-D-3, although computationally expensive compared to CSPNS-D-2 and CSPNS-D-1, consistently outperformed or at least equaled solutions by the latter two. Average savings were more pronounced in CSPNS-D-3 than in CSPNS-D-2 and CSPNS-D-1. Moreover, in the majority of experiments, CSPNSD-3 and CSPNS-D-2 produced identical solutions since all nodes that could potentially be covered by the node operation could also be covered by the Link-Node-Link Operation without incurring extra waiting time.

The Covering Salesman Problem (CSP) is inherently NP-hard, and our extended problem inherits this complexity. Solutions for larger instances necessitate heuristic approaches. Therefore, we propose the link removal heuristic, which involves eliminating links unlikely to be part of the truck route, thereby reducing the potential number of branches in the MILP solution process. This heuristic was applied to benchmark instances with 50 orders. Before its application, obtaining feasible solutions for 50 orders with CSPNS-D-2 and CSPNS-D-3 within 2 hours was challenging, with the CPLEX pre-processing stage unable to complete in that time frame. However, after applying the heuristic, we successfully obtained feasible solutions for CSPNSD-2 and CSPNS-D-3, which, in some cases, surpassed the optimum solution with CSPNS-D-1.

Lastly, it is well-established that using a drone with a large covering distance (d_{max}) in a dense service area allows for the efficient fulfillment of more orders, ultimately reducing delivery time. Conversely, when employing a drone with a small d_{max} in a sparse service area, only a few or no orders can be served by

the drone, potentially leading to higher delivery times. We simulated scenarios with different drone covering ranges (d_{max}) and order densities to extract insights from the numerical results, offering valuable guidance for future designs and the development of drones for home delivery.

In Robot-Assisted delivery systems, two existing frameworks are commonly used in existing studies. One involves using a truck to transport customer shipments to localized robot hubs for onward delivery to customers by localized robots, while the other entails transporting both customer shipments and robots, releasing the robots to serve customers, and then rejoining the truck. Each of these setups has its advantages. In our Robot-Assisted Delivery Problem (RADP), we propose a framework that combines the standard truck, robot (transported on truck), and local depot into one model. This integrated framework allows for the simultaneous service of customers by all facilities. The primary aim is to optimize the delivery schedules of all facilities to minimize the total work span of trucks.

The contributions of the study on the Robot-Assisted Delivery Problem include:

- **Innovative Framework:** By merging the concepts of using local depots and standard trucks carrying robots and shipments, RADP addresses a significant gap in the literature, presenting a comprehensive model for optimized delivery scheduling.

- **Model Development:** Two distinct but consistent MILP models, RADP-1 and RADP-2, are introduced.

- **Heuristic Solutions:** Two heuristic approaches, P-Heur and K-Heur, based on operation prioritization and decomposition, respectively, are proposed.

Numerical solutions revealed that RADP-1 can solve up to 8 nodes optimally, whereas RADP-2 can only solve 7 nodes optimally. While the two models are consistent, RADP-1 tends to give better results when the optimum solution involves launching robots at the same node and collecting them at different nodes or launching at different nodes and collecting them at the same node. This is due to RADP-2 operations only considering both launching and collecting of robots at the same node. Additionally, RADP-1 is consistently easier to solve than RADP-2 due to the large number of operations in RADP-2.

Due to the computational complexity of RADP models, heuristic approaches (P-Heur & K-Heur) were proposed to accelerate solutions for larger instances. Experiments on randomly generated data with a total of 7 nodes tested the performance of the prioritization heuristic (P-Heur) on RADP-2. Results show that the heuristic matches the MILP solutions in 5 out of the 30 experiments, with less than a 10% optimality gap in 18 experiments and a greater than 10% optimality gap in 7 experiments. These solutions were found in less than a second. Finally, the two heuristic approaches were applied to larger instances, producing solutions in a few seconds that would otherwise be impossible to find with commercial solver like CPLEX.

Bibliography

[1] S. Srinivas, S. Ramachandiran, and S. Rajendran, "Autonomous robot-driven deliveries: A review of recent developments and future directions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 165, no. 102834, 2022.

[2] ONS, "Retail sales, great britain : November 2022. available at: https:https://www.ons.gov.uk/.," 2022. Accessed: 14 Oct 2023.

[3] C. Waite, "2020 uk greenhouse gas emmissons, department for business, energy and industrial strategy. available at: https:https://assets.publishing.service.gov.uk/media/61f7fb228fa8f5388f444e85/2020-final-emissions-statistics-one-page-summary.pdf," 2022. Accessed: 10 Oct 2023.

[4] M. Savelsbergh and T. Van Woensel, "50th anniversary invited article—city logistics: Challenges and opportunities," *Transportation Science*, vol. 50, no. 2, pp. 579–590, 2016.

[5] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[6] N. Boysen, D. Briskorn, S. Fedtke, and S. Schwerdfeger, "Drone delivery from trucks: Drone scheduling for given truck routes," *Networks*, vol. 72, no. 4, pp. 506–527, 2018.

[7] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2016.

[8] N. Boysen, S. Schwerdfeger, and F. Weidinger, "Scheduling last-mile deliveries with truck-based autonomous robots," *European Journal of Operational Research*, vol. 271, no. 3, pp. 1085–1099, 2018.

[9] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198–212, 2018.

[10] M. D. Simoni, E. Kutanoglu, and C. G. Claudel, "Optimization and analysis of a robot-assisted last mile delivery system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, no. 102049, 2020.

[11] M. Ostermeier, A. Heimfarth, and A. Hübner, "Cost-optimal truck-and-robot routing for last-mile delivery," *Networks*, vol. 79, no. 3, pp. 364–389, 2022.

[12] T. Kirschstein, "Comparison of energy demands of drone-based and ground-based parcel delivery services," *Transportation Research Part D: Transport and Environment*, vol. 78, no. 102209, 2020.

[13] A. J. Torija and C. Clark, "A psychoacoustic approach to building knowledge about human response to noise of unmanned aerial vehicles," *International Journal of Environmental Research and Public Health*, vol. 18, no. 682, 2021.

[14] M. G. Speranza, "Trends in transportation and logistics," *European Journal of Operational Research*, vol. 264, no. 3, pp. 830–836, 2018.

[15] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton university press, 2006.

[16] W. J. Cook, *In pursuit of the traveling salesman: mathematics at the limits of computation*. Princeton University Press, 2011.

[17] M. Drexl, "Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.

[18] I.-M. Chao, "A tabu search method for the truck and trailer routing problem," *Computers & Operations Research*, vol. 29, no. 1, pp. 33–51, 2002.

[19] S. Scheuerer, "A tabu search heuristic for the truck and trailer routing problem," *Computers & Operations Research*, vol. 33, no. 4, pp. 894–909, 2006.

[20] S.-W. Lin, F. Y. Vincent, and S.-Y. Chou, "Solving the truck and trailer routing problem based on a simulated annealing heuristic," *Computers & Operations Research*, vol. 36, no. 5, pp. 1683–1692, 2009.

[21] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, 2020.

[22] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.

[23] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.

[24] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," *Networks*, vol. 72, no. 4, pp. 528–542, 2018.

[25] S. Poikonen, B. Golden, and E. A. Wasil, "A branch-and-bound approach to the traveling salesman problem with a drone," *INFORMS Journal on Computing*, vol. 31, no. 2, pp. 335–346, 2019.

[26] R. Roberti and M. Ruthmair, "Exact methods for the traveling salesman problem with drone," *Transportation Science*, vol. 55, no. 2, pp. 315–335, 2021.

[27] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optimization Letters*, vol. 11, pp. 679–697, 2017.

[28] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, 2017.

[29] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *International Journal of Production Economics*, vol. 225, no. 107598, 2020.

[30] Y. Liu, Z. Liu, J. Shi, G. Wu, and W. Pedrycz, "Two-echelon routing problem for parcel delivery by cooperated truck and drone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 12, pp. 7450–7465, 2020.

[31] D. Schermer, M. Moeini, and O. Wendt, "A hybrid vns/tabu search algorithm for solving the vehicle routing problem with drones and en route operations," *Computers & Operations Research*, vol. 109, pp. 134–158, 2019.

[32] M. Marinelli, L. Caggiani, M. Ottomanelli, and M. Dell'Orco, "En route truck–drone parcel delivery for optimal vehicle routing strategies," *IET Intelligent Transport Systems*, vol. 12, no. 4, pp. 253–261, 2018.

[33] S. Lin, "An efficient heuristic algorithm for the traveling salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[34] J. R. Current and D. A. Schilling, "The covering salesman problem," *Transportation science*, vol. 23, no. 3, pp. 208–213, 1989.

[35] M. Gendreau, G. Laporte, and F. Semet, "The covering tour problem," *Operations Research*, vol. 45, no. 4, pp. 568–576, 1997.

[36] B. Golden, Z. Naji-Azimi, S. Raghavan, M. Salari, and P. Toth, "The generalized covering salesman problem," *INFORMS Journal on Computing*, vol. 24, no. 4, pp. 534–553, 2012.

[37] T. Matsuura, T. Kimura, *et al.*, "Covering salesman problem with nodes and segments," *American Journal of Operations Research*, vol. 7, no. 4, pp. 249–264, 2017.

[38] Z. Naji-Azimi and M. Salari, "The time constrained maximal covering salesman problem," *Applied Mathematical Modelling*, vol. 38, no. 15-16, pp. 3945–3957, 2014.

[39] G. Ozbaygin, H. Yaman, and O. E. Karasan, "Time constrained maximal covering salesman problem with weighted demands and partial coverage," *Computers & Operations Research*, vol. 76, pp. 226–237, 2016.

[40] M. Hossain, "Autonomous delivery robots: A literature review," *IEEE Engineering Management Review*, vol. 51, no. 4, pp. 77–89, 2023.

[41] M. Poeting, S. Schaudt, and U. Clausen, "Simulation of an optimized last-mile parcel delivery network involving delivery robots," in *Interdisciplinary Conference on Production, Logistics and Traffic*, pp. 1–19, Lecture Notes in Logistics, Springer, 2019.

[42] D. Jennings and M. Figliozzi, "Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel," *Transportation Research Record*, vol. 2673, no. 6, pp. 317–326, 2019.

[43] M.-O. Sonneberg, M. Leyerer, A. Kleinschmidt, F. Knigge, and M. H. Breitner, "Autonomous unmanned ground vehicles for urban logistics: Optimization of last mile delivery operations," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[44] S. Yu, J. Puchinger, and S. Sun, "Van-based robot hybrid pickup and delivery routing problem," *European Journal of Operational Research*, vol. 298, no. 3, pp. 894–914, 2022.

[45] I. Bakach, A. M. Campbell, and J. F. Ehmke, "A two-tier urban delivery network with robot-based deliveries," *Networks*, vol. 78, no. 4, pp. 461–483, 2021.

[46] T. G. Crainic, "City logistics," in *State-of-the-art decision-making tools in the information-intensive age*, pp. 181–212, INFORMS, 2008.

[47] J. Olsson, D. Hellström, and H. Pålsson, "Framework of last mile logistics research: A systematic review of the literature," *Sustainability*, vol. 11, no. 7131, 2019.

[48] R. Gevaers, E. Van de Voorde, and T. Vanelslander, "Characteristics and typology of last-mile logistics from an innovation perspective in an urban context," *City Distribution and Urban Freight Transport: Multiple Perspectives, Edward Elgar Publishing*, pp. 56–71, 2011.

[49] A. H. Hübner, H. Kuhn, J. Wollenburg, N. Towers, and H. Kotzab, "Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework," *International Journal of Retail & Distribution Management*, vol. 44, no. 3, 2016.

[50] M. Xu, B. Ferrand, and M. Roberts, "The last mile of e-commerce–unattended delivery from the consumers and etailers' perspectives," *International Journal of Electronic Marketing and Retailing*, vol. 2, no. 1, pp. 20–38, 2008.

[51] N. Tiwapat, C. Pomsing, and P. Jomthong, "Last mile delivery: Modes, efficiencies, sustainability, and trends," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pp. 313–317, IEEE, 2018.

[52] M. Punakivi, H. Yrjölä, and J. Holmström, "Solving the last mile issue: reception box or delivery box?," *International Journal of Physical Distribution & Logistics Management*, vol. 31, no. 6, pp. 427–439, 2001.

[53] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, 2018.

[54] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: an exploration into natural human-drone interaction," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pp. 361–365, 2015.

[55] P. M. Asaro, "The labor of surveillance and bureaucratized killing: new subjectivities of military drone operators," *Social semiotics*, vol. 23, no. 2, pp. 196–224, 2013.

[56] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 597–621, 2018.

[57] D. German Aerospace Center DLR, "Landing on a moving car / landung auf einem fahrenden auto," Jan 18. 2016. [Accessed: 24 September 2020].

[58] ManBehindtheCanon, "Drone Out of a Moving Car-Behind the Scenes." https://www.youtube.com/watch?v=aYw0PmN0irY, 2017.

[59] B. Gavish and S. C. Graves, "The travelling salesman problem and related problems," 1978. Operations Research Center Working Paper;OR 078-78.

[60] DJI, "Phantom 4 pro v2.0: Unboxing and highlights.[online]. available at: https://store.dji.com/guides/phantom-4-pro-v2-unboxing-and-highlights/." [Accessed: 04 August 2021].

[61] P. Bouman, N. Agatz, and M. Schmidt, "nstances for the TSP with Drone and some solutions." https://zenodo.org/record/1204676#.YmAX_IXMJyw, 2018.

[62] V. N. Lu, J. Wirtz, W. H. Kunz, S. Paluch, T. Gruber, A. Martins, and P. G. Patterson, "Service robots, customers and service employees: what can we learn from the academic literature and where are the gaps?," *Journal of Service Theory and Practice*, vol. 30, no. 3, pp. 361–391, 2020.

[63] J. H. Andrew, "Thousands of autonomous delivery robots are about to descend on us college campuses. available at: https://www.theverge.com/2019/8/20/20812184/starship-delivery-robot-expansion-college-campus," 2019. Accessed: 08 Nov 2022.

[64] G. Liping, "Delivery robots hit the road in beijing. available at: http://www.ecns.cn/news/sci-tech/2018-06-20/detail-ifyvmiee7350792.shtml," 2018. Accessed: 08 Nov 2022.

[65] Y. Li, "Business plan for autonomous delivery robot," *Intelligent Control and Automation*, vol. 11, no. 2, pp. 33–46, 2020.

[66] I. Kottasova, "Forget drones, here come delivery robots. available at: https://money.cnn.com/2015/11/03/technology/starship-delivery-robots/," 2015. Accessed: 12 Oct 2022.

[67] S. Yu, J. Puchinger, and S. Sun, "Two-echelon urban deliveries using autonomous vehicles," *Transportation Research Part E: Logistics and Transportation Review*, vol. 141, no. 102018, 2020.

[68] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch, *et al.*, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[69] C. Chen, E. Demir, Y. Huang, and R. Qiu, "The adoption of self-driving delivery robots in last mile logistics," *Transportation Research part E: Logistics and Transportation Review*, vol. 146, no. 102214, 2021.

[70] J. Blömer, C. Lammersen, M. Schmidt, and C. Sohler, "Theoretical analysis of the k-means algorithm–a survey," *Algorithm Engineering: Selected Results and Surveys*, pp. 81–116, 2016.

Appendix A

To proof Proposition 1, we need to proof that "Any point P that is on the boundary of the ellipse defined by foci (E,B) is inside the ellipse defined by (D,B) where $\overline{DB} = d_{max}/\alpha$, as shown in Figure 1.

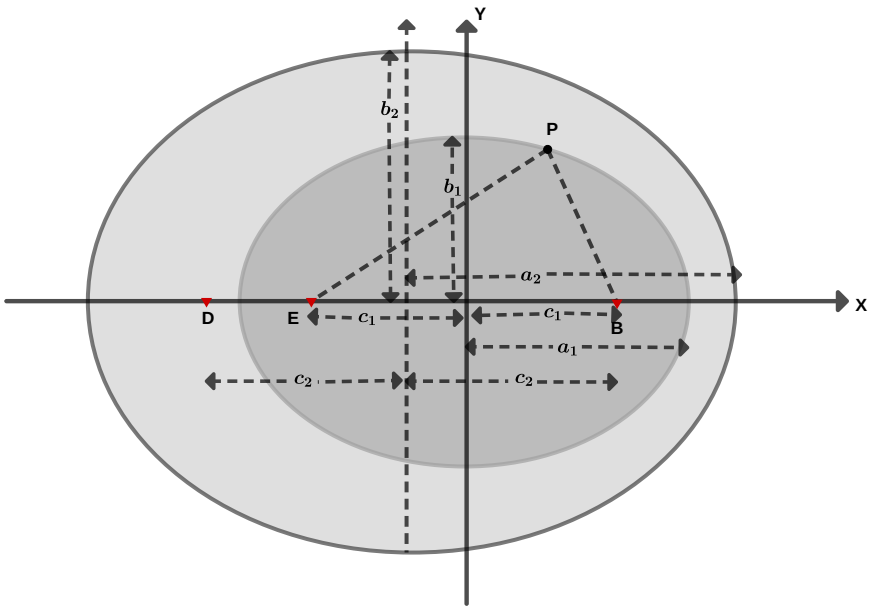


Figure 1. Ellipse defined by foci (E,B) enclosed by ellipse defined foci (D,B)

Proof

Let λ $(0 < \lambda < 1)$ be the ratio between $\overline{EB}$ and $\overline{DB}$, we have $\overline{EB} = \lambda\overline{DB} = \lambda d_{max}/\alpha$. $\overline{EB}$ is the distance between the foci of ellipse $\mathcal{E}_1(E,B)$, so we have $c_1 = \overline{EB}/2 = \lambda d_{max}/2\alpha$. Let P be a point on the boundary of $\mathcal{E}_1(E,B)$. To allow the drone to synchronise with the truck we must have $\overline{EP} + \overline{PB} = \lambda d_{max}$, so the

semimajor axis of $\mathcal{E}_1(E,B)$ is given by $a_1 = \lambda d_{max}/2$, Therefore the semiminor axis $b_1 = \sqrt{a_1^2 - c_1^2} = \frac{\lambda d_{max}}{2\alpha}\sqrt{\alpha^2 - 1}$. Placing the ellipse $\mathcal{E}_1(E,B)$ into the coordinate system, centred at the origin, a point P on its boundary must satisfy the equation:

$$\frac{x^2}{(\frac{\lambda^2 d_{max}^2}{4})} + \frac{y^2}{(\frac{\lambda^2 d_{max}^2(\alpha^2-1)}{4\alpha^2})} = 1 \tag{1}$$

On the other hand, the ellipse $\mathcal{E}_2(D,B)$ has $c_2 = \overline{DB}/2 = d_{max}/2\alpha$, $a_2 = d_{max}/2$, $b_2 = \sqrt{a_2^2 - c_2^2} = \frac{d_{max}}{2\alpha}\sqrt{\alpha^2 - 1}$, and centred at $((\lambda-1)d_{max}/2\alpha, 0)$ in the same coordinate system. Therefore our aim is to show that P satisfies the equation:

$$\frac{(x - (\lambda-1)\frac{d_{max}}{2\alpha}))^2}{(\frac{d_{max}^2}{4})} + \frac{y^2}{(\frac{d_{max}^2(\alpha^2-1)}{4\alpha^2})} \leq 1 \tag{2}$$

Replacing $\frac{y^2}{(\frac{d_{max}^2(\alpha^2-1)}{4\alpha^2})}$ by that from equation (18),

$$\frac{(x - (\lambda-1)\frac{d_{max}}{2\alpha}))^2}{(\frac{d_{max}^2}{4})} + \frac{y^2}{(\frac{d_{max}^2(\alpha^2-1)}{4\alpha^2})} \tag{3}$$

$$= \frac{(x - (\lambda-1)\frac{d_{max}}{2\alpha}))^2}{(\frac{d_{max}^2}{4})} + \lambda^2 - \frac{x^2}{(\frac{d_{max}^2}{4})} \tag{4}$$

$$= \frac{x^2 - 2x(\lambda-1)\frac{d_{max}}{2\alpha} + (\lambda-1)^2(\frac{d_{max}}{2\alpha})^2 + \frac{\lambda^2 d_{max}^2}{4} - x^2}{(\frac{d_{max}^2}{4})} \tag{5}$$

$$= \frac{-2x(\lambda-1)\frac{d_{max}}{2\alpha} + (\lambda-1)^2(\frac{d_{max}}{2\alpha})^2 + \frac{\lambda^2 d_{max}^2}{4}}{(\frac{d_{max}^2}{4})} \tag{6}$$

As $\lambda < 1$, so the coefficient $-2(\lambda-1)\frac{d_{max}}{2\alpha} > 0$. Therefore the equation 6 is increasing with x. P is a point on ellipse $\mathcal{E}_1(E,B)$, so $-\lambda d_{max}/2 \leq x \leq \lambda d_{max}/2$. So we have:

$$\leq \frac{-2\frac{\lambda d_{max}}{2}(\lambda-1)\frac{d_{max}}{2\alpha}+(\lambda-1)^2(\frac{d_{max}}{2\alpha})^2+\frac{\lambda^2 d_{max}^2}{4}}{(\frac{d_{max}^2}{4})} \tag{7}$$

$$= \frac{-\lambda(\lambda-1)\frac{d_{max}^2}{2\alpha}+(\lambda-1)^2(\frac{d_{max}}{2\alpha})^2+\frac{\lambda^2 d_{max}^2}{4}}{(\frac{d_{max}^2}{4})} \tag{8}$$

$$= \frac{((\lambda-1)\frac{1}{\alpha}-\lambda)^2\frac{d_{max}^2}{4}}{(\frac{d_{max}^2}{4})} \tag{9}$$

$$= ((\lambda-1)\frac{1}{\alpha}-\lambda)^2 \tag{10}$$

$$= ((1-\lambda)\frac{1}{\alpha}+\lambda)^2 \tag{11}$$

$$\leq ((1-\lambda)+\lambda)^2 \tag{12}$$

$$= 1 \tag{13}$$

So we proved that P is inside the ellipse $\mathcal{E}_2(D,B)$.

Appendix B

To proof proposition 2, let P be the "highest" point that can be covered by Link-Node Operation from link AB. We aim to prove that P is inside the ellipse with foci DE and semi-major axis , $a = \frac{1}{2}d_{max}$, i.e the covering area of Link-Node-Link Operation with links AB and BC.

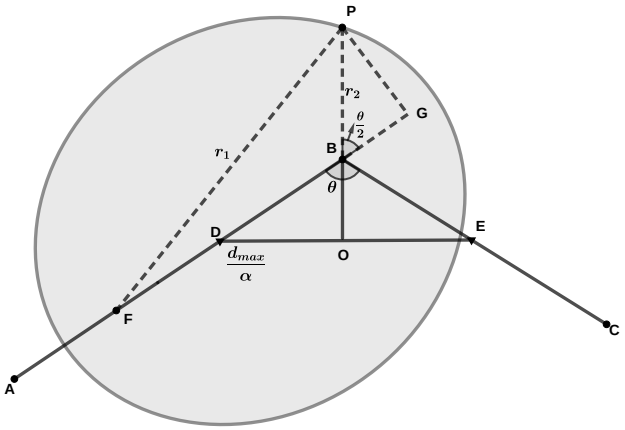


Figure 2. Ellipse with foci (F,B)

Proof

Let θ denote the angle between links AB and BC, with $0 \leq \theta \leq \pi$. (The other case is related to the discussion here). Let $\overline{FP} = r_1$ and $\overline{PB} = r_2$, as P is on ellipse, $\mathcal{E}$ with foci (F, B) , then by defination we have $r_1 + r_2 = d_{max}$. $PG = r_2 \sin\left(\frac{\theta}{2}\right)$, $BG = r_2 \cos\left(\frac{\theta}{2}\right)$,

$FB = \frac{d_{max}}{\alpha}$.

So, for the right angle triangle PGF:

$$r_1^2 = r_2^2 \sin^2\left(\frac{\theta}{2}\right) + \left(\frac{d_{max}}{\alpha} + r_2 \cos\left(\frac{\theta}{2}\right)\right)^2 \tag{1}$$

$$\left(d_{max} - r_2^2\right) = r_2^2 \sin^2\left(\frac{\theta}{2}\right) + \frac{d_{max}^2}{\alpha^2} + \frac{2}{\alpha} r_2 d_{max} \cos\left(\frac{\theta}{2}\right) + r_2^2 \cos^2\left(\frac{\theta}{2}\right) \tag{2}$$

$$d_{max}^2 - 2 d_{max} r_2 + r_2^2 = r_2^2 + \frac{d_{max}^2}{\alpha^2} + \frac{2}{\alpha} r_2 d_{max} \cos\left(\frac{\theta}{2}\right) \tag{3}$$

$$r_2 = \frac{\left(1 - \frac{1}{\alpha^2}\right) d_{max}}{\frac{2}{\alpha}\cos\left(\frac{\theta}{2}\right) + 2} \tag{4}$$

$$= \frac{\left(\alpha^2 - 1\right) d_{max}}{2\alpha \cos\left(\frac{\theta}{2}\right) + 2\alpha^2} = PB. \tag{5}$$

So, if we set point O on DE as the origin, P is $\left(0, \frac{d_{max}}{2\alpha}\cos\left(\frac{\theta}{2}\right) + \frac{\left(\alpha^2-1\right)d_{max}}{2\alpha\cos\left(\frac{\theta}{2}\right)+2\alpha^2}\right)$. Now we prove that P is inside ellipse centred at O. For the ellipse, $a = \frac{d_{max}}{2}$, $c = \frac{d_{max}}{2\alpha}\sin\left(\frac{\theta}{2}\right)$, and $b = \sqrt{\frac{d_{max}^2}{4} - \frac{d_{max}^2}{4\alpha^2}\sin^2\left(\frac{\theta}{2}\right)}$

$$= \frac{d_{max}}{2\alpha}\sqrt{\alpha^2 - \sin^2\left(\frac{\theta}{2}\right)} \tag{6}$$

So, the ellipse is:

$$\frac{x^2}{\frac{d_{max}^2}{4}} + \frac{y^2}{\frac{d_{max}^2}{4\alpha^2}\left(\alpha^2 - \sin^2\left(\frac{\theta}{2}\right)\right)} = 1 \tag{7}$$

Substituting P we have:

$$0 + \frac{\left(\frac{d_{max}}{2\alpha}\cos\left(\frac{\theta}{2}\right) + \frac{(\alpha^2-1)d_{max}}{2\alpha\cos\left(\frac{\theta}{2}\right)+2\alpha^2}\right)^2}{\frac{d_{max}^2}{4\alpha^2}\left(\alpha^2 - \sin^2\left(\frac{\theta}{2}\right)\right)} \tag{8}$$

$$= \frac{\cos^2\left(\frac{\theta}{2}\right) + 2\cos\left(\frac{\theta}{2}\right)\left(\frac{\alpha^2-1}{\cos\left(\frac{\theta}{2}\right)+\alpha}\right) + \left(\frac{(\alpha^2-1)}{\cos\left(\frac{\theta}{2}\right)+\alpha}\right)^2}{\alpha^2 - \sin^2\left(\frac{\theta}{2}\right)} \tag{9}$$

$$\leq \frac{\cos^2\left(\frac{\theta}{2}\right) + 2\cos\left(\frac{\theta}{2}\right)\frac{\alpha^2-1}{\cos\left(\frac{\theta}{2}\right)+\alpha} + \left(\frac{\alpha^2-1}{\cos\left(\frac{\theta}{2}\right)+\alpha}\right)^2}{1 - \sin^2\left(\frac{\theta}{2}\right)} \tag{10}$$

$$\leq \frac{\cos^2\left(\frac{\theta}{2}\right)}{\cos^2\left(\frac{\theta}{2}\right)} = 1 \tag{11}$$

Appendix C

Table 1: Statistical Comparison of CSPNS-D's Results on Benchmark instances with 10 orders

Instance	TSP Delivery time (s)	Delivery Time (s)			No. of drones Used			Savings over TSP(%)		
		CSPNSD-1	CSPNSD-2	CSPNSD-3	CSPNSD-1	CSPNSD-2	CSPNSD-3	CSPNSD-1	CSPNSD-2	CSPNSD-3
Uniform_10_1	45.18	32.99	17.39	17.39	4	7	7	26.97	61.51	61.51
Uniform_10_2	45.58	34.96	13.96	13.96	5	6	6	23.31	69.37	69.37
Uniform_10_3	42.70	31.14	4.68	4.68	6	4	4	27.06	89.03	89.03
Uniform_10_4	46.66	31.20	15.36	15.36	6	6	6	33.13	67.08	67.08
Uniform_10_5	50.71	33.71	19.84	19.84	4	6	6	33.52	60.88	60.88
Uniform_10_6	48.40	29.52	13.49	13.49	7	5	5	39.00	72.12	72.12
Uniform_10_7	38.34	27.60	7.70	7.70	6	5	5	28.01	79.93	79.93
Uniform_10_8	45.33	38.93	15.35	15.35	5	5	5	14.12	66.14	66.14
Uniform_10_9	52.79	38.57	16.66	16.66	5	6	6	26.93	68.44	68.44
Uniform_10_10	42.04	33.23	16.09	16.09	6	5	5	20.94	61.73	61.73
Single_Center_10_1	65.86	48.87	28.18	28.18	7	5	5	25.79	57.21	57.21
Single_Center_10_2	58.42	37.87	34.39	34.39	4	5	5	35.19	41.14	41.14
Single_Center_10_3	76.57	62.41	27.09	27.09	4	4	4	18.49	64.62	64.62
Single_Center_10_4	54.75	48.62	13.49	13.49	2	5	5	11.20	75.36	75.36
Single_Center_10_5	80.50	48.11	9.45	9.45	4	5	5	40.23	88.26	88.26
Single_Center_10_6	62.46	42.19	29.17	29.17	5	4	4	32.46	53.30	53.30
Single_Center_10_7	79.83	59.50	37.16	37.16	4	5	5	25.46	53.45	53.45
Single_Center_10_8	79.82	67.39	44.00	44.00	4	5	5	15.58	44.88	44.88
Single_Center_10_9	57.56	39.96	16.53	16.53	3	5	5	30.57	71.28	71.28
Single_Center_10_10	54.75	48.62	13.49	13.49	2	6	6	11.20	75.36	75.36
Double_Center_10_1	92.70	74.34	64.21	56.47	3	4	3	19.80	30.73	39.08
Double_Center_10_2	71.12	45.80	45.69	45.69	6	7	7	35.61	35.76	35.76
Double_Center_10_3	119.69	105.83	100.98	87.96	3	4	5	11.58	15.64	26.51
Double_Center_10_4	89.52	75.07	59.48	59.48	3	4	4	16.14	33.55	33.55
Double_Center_10_5	136.70	134.47	126.39	126.39	3	2	2	1.63	7.54	7.54
Double_Center_10_6	100.72	93.16	87.25	87.25	4	4	4	7.51	13.37	13.37
Double_Center_10_7	113.09	108.23	84.21	80.61	5	5	3	4.30	25.54	28.73
Double_Center_10_8	116.49	110.18	101.05	96.48	4	4	4	5.42	13.26	17.18
Double_Center_10_9	126.33	117.82	99.52	92.66	3	2	2	6.74	21.22	26.65
Double_Center_10_10	131.23	106.87	93.41	93.41	3	2	2	18.56	28.82	28.82
Average Savings Over TSP								**21.55**	**51.55**	**52.61**
Min								**1.63**	**7.54**	**7.54**
Max								**40.23**	**89.03**	**89.03**

Table 2: Statistical Comparison of CSPNS-D's Results on Benchmark instances with 20 orders

Instance	TSP Delivery	Delivery Time			No. of drones Used			Savings over TSP		
	time (s)	CSPNSD-1	CSPNSD-2	CSPNSD-3	CSPNSD-1	CSPNSD-2	CSPNSD-3	CSPNSD-1	CSPNSD-2	CSPNSD-3
Uniform_20_1	106.87	66.06	44.74	44.74	8	7	7	38.19	58.14	58.14
Uniform_20_2	112.70	76.48	57.12	57.12	7	11	11	32.14	49.32	49.32
Uniform_20_3	118.35	69.93	64.36	64.36	8	7	7	40.91	45.62	45.62
Uniform_20_4	110.83	71.22	60.24	60.24	7	8	8	35.74	45.65	45.65
Uniform_20_5	119.65	81.54	62.11	62.11	8	7	7	31.85	48.10	48.10
Uniform_20_6	130.93	89.96	69.23	69.23	7	8	8	31.29	47.13	47.13
Uniform_20_7	117.48	79.60	68.97	68.97	5	7	7	32.25	41.29	41.29
Uniform_20_8	130.96	82.54	59.40	59.40	7	7	6	36.97	54.64	54.64
Uniform_20_9	114.13	68.76	51.47	51.47	8	10	10	39.75	54.91	54.91
Uniform_20_10	127.00	83.58	65.79	65.79	6	6	6	34.19	48.20	48.20
Single_Center_20_1	177.34	132.13	124.73	124.73	6	7	7	25.50	29.67	29.67
Single_Center_20_2	162.05	129.27	98.40	98.40	9	7	5	20.23	39.28	39.28
Single_Center_20_3	223.44	167.71	164.31	164.31	4	7	7	24.94	26.46	26.46
Single_Center_20_4	189.71	152.17	150.11	150.11	10	8	8	19.79	20.87	20.87
Single_Center_20_5	192.92	152.95	136.51	130.36	5	5	5	20.72	29.24	32.43
Single_Center_20_6	148.43	108.65	92.38	92.38	8	7	4	26.80	37.76	37.76
Single_Center_20_7	170.63	135.51	118.50	118.50	8	7	7	20.59	30.55	30.55
Single_Center_20_8	178.74	144.17	126.24	126.24	9	7	9	19.35	29.37	29.37
Single_Center_20_9	213.28	177.83	177.57	177.57	5	4	4	16.62	16.74	16.74
Single_Center_20_10	155.57	115.40	91.19	91.19	6	4	4	25.82	41.38	41.38
Double_Center_20_1	265.79	239.79	229.74	227.17	3	3	3	9.78	13.56	14.53
Double_Center_20_2	250.53	212.73	212.73	212.73	7	7	7	15.09	15.09	15.09
Double_Center_20_3	294.39	268.85	263.67	263.67	4	4	4	8.67	10.43	10.43
Double_Center_20_4	336.34	296.95	296.35	296.35	5	5	3	11.71	11.89	11.89
Double_Center_20_5	227.25	177.16	158.70	158.70	5	5	5	22.04	30.17	30.17
Double_Center_20_6	226.99	196.74	190.73	190.73	9	4	4	13.33	15.97	15.97
Double_Center_20_7	237.38	202.25	189.82	189.82	5	4	4	14.80	20.04	20.04
Double_Center_20_8	224.30	183.04	137.19	137.19	6	7	7	18.39	38.84	38.84
Double_Center_20_9	232.12	183.81	166.21	166.21	5	6	6	20.81	28.39	28.39
Double_Center_20_10	254.43	204.82	202.85	188.10	4	4	6	19.50	20.27	26.07
Average Savings Over TSP								**24.26**	**33.30**	**33.63**
Min								**8.67**	**10.43**	**10.43**
Max								**40.91**	**58.14**	**58.14**