ENERGY EFFICIENT SIMULTANEOUS DRONE AND MOBILE CHARGER ROUTING PROBLEM WITH TIME RESTRICTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALPTUĞ AYBERK CANPOLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

AUGUST 2023

Approval of the thesis:

ENERGY EFFICIENT SIMULTANEOUS DRONE AND MOBILE CHARGER ROUTING PROBLEM WITH TIME RESTRICTIONS

submitted by **ALPTUĞ AYBERK CANPOLAT** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Pelin Bayındır
Head of Department, **Industrial Engineering** _____

Assoc. Prof. Dr. Mustafa Kemal Tural
Supervisor, **Industrial Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Sinan Gürel
Industrial Engineering, METU _____

Assoc. Prof. Dr. Mustafa Kemal Tural
Industrial Engineering, METU _____

Prof. Dr. Cem İyigün
Industrial Engineering, METU _____

Assist. Prof. Dr. Nader Ghaffarinasab
Industrial Engineering, METU _____

Assist. Prof. Dr. Derya Dinler
Industrial Engineering, Hacettepe University _____

Date:09.08.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Alptuğ Ayberk Canpolat

Signature :

ABSTRACT

ENERGY EFFICIENT SIMULTANEOUS DRONE AND MOBILE CHARGER ROUTING PROBLEM WITH TIME RESTRICTIONS

Canpolat, Alptuğ Ayberk
M.S., Department of Industrial Engineering
Supervisor: Assoc. Prof. Dr. Mustafa Kemal Tural

AUGUST 2023, 126 pages

This study focuses on simultaneous routing of a drone and a mobile charging station given a specific time constraint to complete a mission, based on energy efficiency. The drone is expected to visit certain service points within a certain period of time in the problem. During this operation, the energy needed by the drone is provided by the mobile charging station moving on the ground. The aim is to find the most energy-efficient (or time-efficient) route and the velocities of both vehicles during the operation. The route is obtained by operating a matheuristic that integrates various heuristic algorithms and a mathematical model based on Second Order Cone Programming. The aim of this study is to determine the visiting sequence of service points by the drone, the meeting sequence with the mobile charging station, the meeting points (locations), and vehicles' speeds during the operation. These outputs can be obtained under two different objective functions. The first objective function aims to minimize the total energy consumption of the two vehicles, and the second objective function aims to minimize the total system time. In the model implemented with the first objective function, the total system time is one of the most critical constraints.

At the end of the study, different experimental setups are created to analyze the impact

of variations in parameters and algorithms on the results. Experiments are conducted to examine the impact of different energy consumption parameters of the drone and mobile charging station and total system times on the solution. Additionally, these experiments investigate the effect of different algorithmic setups, such as running the 2-Opt Algorithm with or without a speed-up method and moving on to a new iteration after the first improvement during the trial of alternative routes obtained by 2-Opt or testing all alternatives. The aim is to analyze the impact of these setups on the results. These experiments yield results from instances with varying numbers of service points. The experiments' results and durations are compared and analyzed at the end of the study.

Keywords: Energy Efficiency, Second Order Cone Programming, Drones, Mobile Charging Station, Routing, Rendezvous Problem

ÖZ

ENERJİ VERİMLİLİĞİ ESASLI ZAMAN KISITLI VE EŞ ZAMANLI DRON VE MOBİL ŞARJ İSTASYONU ROTALAMA

Canpolat, Alptuğ Ayberk
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Mustafa Kemal Tural

Ağustos 2023 , 126 sayfa

Bu çalışma, enerji verimliliğine dayalı olarak bir görevi tamamlamak için belirli bir zaman kısıtlaması verilen bir drone ve bir mobil şarj istasyonunun eş zamanlı olarak rotalanmasına odaklanmaktadır. Drone'nun problemde belirli bir süre içerisinde belirli servis noktalarını ziyaret etmesi bekleniyor. Bu operasyon sırasında dronun ihtiyaç duyduğu enerji, yerde hareket eden mobil şarj istasyonu tarafından sağlanıyor. Amaç, operasyon sırasında her iki aracın enerji açısından en verimli (veya zaman açısından verimli) rotasını ve hızlarını bulmaktır. Güzergah, çeşitli buluşsal algoritmaları ve İkinci Dereceden Konik Programlamaya dayalı bir matematiksel modeli birleştiren bir matematik işletilerek elde edilir. Bu çalışmanın amacı, drone ile servis noktalarının ziyaret sırası, mobil şarj istasyonu ile buluşma sırası, buluşma noktaları (lokasyonları) ve operasyon sırasında araçların hızlarının belirlenmesidir. Bu çıktılar iki farklı amaç fonksiyonu altında elde edilebilir. Birinci amaç fonksiyonu, iki aracın toplam enerji tüketimini en aza indirmeyi, ikinci amaç fonksiyonu ise toplam sistem süresini en aza indirmeyi amaçlar. Birinci amaç fonksiyonu ile kurulan modelde toplam sistem süresi en kritik kısıtlardan biridir.

Çalışmanın sonunda, parametrelerdeki ve algoritmalardaki değişimlerin sonuçlar üzerindeki etkisini analiz etmek için farklı deney düzenekleri oluşturulmuştur. Drone ve mobil şarj istasyonunun farklı enerji tüketim parametrelerinin ve toplam sistem sürelerinin çözüm üzerindeki etkisini incelemek için deneyler yapılmıştır. Ayrıca bu deneyler, 2-Opt ile elde edilen alternatif rotaların denenmesi sırasında, 2-Opt Algoritmasını hızlandırma yöntemiyle veya hızlandırma yöntemi olmadan çalıştırmak ve ilk iyileştirmeden sonra yeni bir iterasyona geçmek ya da 2-Opt ile elde edilen bütün alternatifleri test etmek gibi farklı algoritmik kurulumların etkisini araştırır. Amaç, bu kurulumların sonuçlar üzerindeki etkisini analiz etmektir. Bu deneyler, değişen sayıda hizmet noktasına sahip örneklerden sonuçlar verir. Çalışma sonunda deneylerin sonuçları ve süreleri karşılaştırılarak analiz edilir.

Anahtar Kelimeler: Enerji Verimliliği, İkinci Dereceden Konik Programlama, Dronelar, Mobil Şarj İstasyonu, Rotalama, Buluşma Problemi

To all my loved ones

ACKNOWLEDGMENTS

First, I am deeply grateful to Assoc. Prof. Dr. Mustafa Kemal Tural for his guidance, supervision, and patience during the study. With his support during graduate studies, he always kept me motivated for my work.

I would like to thank all my colleagues and my dear manager Sümeyra Günel who supported me and never doubted me during this process.

I would also like to thank Aslı Çıngıl, who was not only my colleague but also my lovely friend during all difficult times in this period, who listened to all my complaints with great patience and supported me endlessly.

I would like to thank my dear friend Mahmut Anıl Özcan, with whom we overcame this program together by supporting each other.

I would like to thank all my family elders who have supported me throughout this process.

Endless thanks to my family, my mother Nedret Canpolat, my father Yakup Canpolat, my brother Alper Afşin Canpolat, my sister-in-law Emel Canpolat, and my little cat Akın Bulut Canpolat. Their loving support, words of encouragement, and the feeling that they are always with me are priceless. It is a great honor and happiness for me to share this success with them.

I would like to express my endless gratitude to my dear friend and brother Muhammed Mihca Erikel, who has never left me alone and helped me overcome all kinds of difficulties in my entire life.

I would like to thank Elif Erçelik, one of my greatest blessings in this life, for her endless trust and love, for her endless support, and for standing by me not only in this period but also in all difficulties. I certainly could not have completed this process without her support.

This study was supported by ASELSAN A.Ş.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

CHAPTER 1

INTRODUCTION

Today, unmanned aerial vehicles (drones) significantly contribute to humanity's many aspects. Drones are indispensable tools for deliveries, surveillance (such as in traffic), reconnaissance, and military operations. In furtherance, Eitan Frachtenberg [1] emphasized the importance of drones in merchandise delivery, courier delivery, food delivery, humanitarian aid delivery, and passenger delivery. Drone delivery systems have been improved considerably with the contributions of technology and food companies such as Amazon, Alphabet, Domino's Pizza, and McDonald's [2].

Most of the research offers solutions for systems delivered by drones. However, unmanned aerial vehicles are essential for monitoring and mapping, researching biodiversity, establishing routes for road and railway, and military operations. Drones can easily carry out these operations without requiring a high workforce [3]. Therefore, we concentrated on offering solutions for such systems in this study. Unfortunately, drone systems still have some critical limitations. The most important are limited payload, energy capacity, weather conditions, and location of charging stations [4]. We aim to provide solutions that minimize energy consumption by meeting the energy needs of drones with mobile charging stations during operations for constant or no payload systems mentioned above, thereby facilitating the energy capacity limitation.

Drone operations refer to the activities where drones visit certain service points sequentially for a specific purpose. During these operations, energy resources are limited, and energy depletion can prevent drones from completing their mission. Therefore, energy constraints are an essential factor in drone operations. During operations, energy consumption must be constantly monitored, and energy resources must be used efficiently. Especially in long missions, good management of energy con-

sumption enables drones to complete their tasks successfully.

In drone operations, it is crucial to maintain the energy levels of the drones. Two methods are proposed to supply energy to drones in the literature. One of them is to charge the drone by connecting it to a charging station. Charging stations placed at strategic points to do this or using mobile charging stations allow drones to charge their batteries while continuing their mission. The other one is that drones can provide their energy by replacing a discharged battery with a fully charged one. In this method, the drone can continue its mission without waiting for long [5]. This method is beneficial for missions that require long flight time, as it allows them to travel greater distances without the need for additional battery packs. However, due to the high costs of these operations because of the need for a special process for battery replacement or the need for an operator to replace the battery, charging the drone by simply connecting it to a charging station is the most preferred method of providing energy. We solved the charging problem in our system with Mobile Charging Stations to reduce the adverse effects of the limited energy capacity and to increase the flight range and time of the drone on operations.

This study offers a solution that will optimize energy efficiency while completing operations in a limited time for a system that includes a drone and a mobile charging station. We choose a system with constant or no payload to expand the literature knowledge besides delivery systems. Therefore, a system in which drone and mobile charging stations work integrated and a model that minimizes energy consumption are designed for the above-mentioned operations. In this context, we focus on determining the optimal routing of the system, the optimal speeds of the drone and the charging station, and the optimal rendezvous points where the drone and the charging station meet for charging.

This study could be an essential step to increase the use of drones in operations such as surveillance, mapping, ecological surveys, reconnaissance, and observation flights before highway and railroad constructions, and military operations. In other words, this study provides solutions that will increase energy and time efficiency for drone operations with constant or non-payload (non-delivery).

In this thesis, we aim to find an integrated route for a system including one drone

and one mobile charger. This route can be found for two different objective functions. These are minimizing the total energy consumption and minimizing the completion time of the operation. We developed a matheuristic to find these routes. This matheuristic consists of a mathematical model and some heuristic algorithms. With this solution approach, we carried out some computational studies at the end of the study and completed our study by discussing these results.

In the next chapters, we will briefly touch on the following: In Chapter 2, the literature related to this field is explained. In Chapter 3, we describe the problem. In Chapter 4, the solution methods are explained. In Chapter 5, results of the computational studies are given. In Chapter 6, the conclusion is provided with some future research directions.

CHAPTER 2

LITERATURE REVIEW

Drones have been used for various purposes from the past to the present. In recent years, their popularity and usage have significantly increased, not only for military purposes but also in civilian areas. Large companies such as Amazon and DHL have pioneered in using drones in delivery systems. The areas where drones are used are discussed by Eitan in the literature [1]. In addition, the author highlights the technological challenges encountered in the usage scenarios for delivery by drones, such as capability, efficiency, and coordination. The payload capacity of drones and the limited distance they can fly due to their battery capacity are challenges that need to be resolved. Eitan has argued that it is necessary for drones to work in coordination with other vehicles in a suitable plan to overcome constraints such as time, cost, and energy. Some studies in the literature aim to reduce the delivery time of drone delivery systems [6, 7, 8]. The systems involved in these studies consist of delivery systems in which drones and trucks work collaboratively. The aim is to find the routes of the truck and drone that will minimize the service time of the system. Murray and Chu [6] first determine the route of the truck and then create the route of the drone according to this route. This heuristic approach is called "Truck First, Drone Second". Crişana and Nechita [7] focus on a different objective function. This new objective function approach includes the flying time of the drone. This approach, which directly aims to minimize the total service time, indirectly aims to minimize the energy consumption of drones. Jeong et al. [8] consider the return-to-depot time while minimizing energy consumption in their model after serving all customers. In doing so, they consider restricted areas (no-fly zones) in drone operations.

As seen in articles of this kind, numerous studies have been conducted on drone deliv-

ery systems in the literature. Rather than focusing on drones as a standalone delivery system, these studies have generally centered on systems that integrate drones with trucks. While energy consumption has occasionally been considered in these studies, the primary aim has typically been to minimize the time.

One way to minimize service time in drone delivery systems is to ensure the continuity of drone flights. One of the most significant factors limiting the continuity of drone flights is the limited battery capacity. At this point, energy supply is of great importance for the healthy functioning of the system. Two methods for providing energy to drones have been mentioned in the literature: recharging directly and swapping batteries. Yurek and Ozmutlu study a truck-drone delivery system that provides energy to the drone using these two types of battery recharging policies (flexible recharging policy) [5]. These two types of recharging policies are full (swapping) and partial recharging. The truck has two roles in this system: delivery task and providing energy to the drone with one of the two specified energy supply methods. In the traveling salesman problem with drones (TSP-D) framework, this study gives the first empirical analysis of the impact of various battery policies, recharging rates, and battery life durations on delivery times. The effect of swapping and recharging the battery during operation on delivery time was also investigated for the first time. The study showed that the coordinated and integrated operation of the truck and drone significantly reduces the delivery time under both recharging policies. The battery swapping policy shows shorter delivery times for uniform distributed service points under the recharging policy with 1-minute swapping time. However, the recharging policy may reduce delivery times depending on the recharging rate (recharging time/-battery life) when the swapping duration is extended. The partial recharging policy outperforms the swapping policy for medium-sized centered and clustered distributed service points with a recharging rate (recharging time/battery life) of 1 or less.

Park et al. [9] and Sawadsitang et al. [10] carry out essential studies on the energy supply of drones during operation. The battery and charger assignment problem is handled in a discrete event simulation framework by Park et al. [9]. The authors suggest a solution method that computes charging and job dispatching schedules, as well as decide where to assign batteries and chargers by looking for the optimal location. Additionally, a scheduling model was developed to pinpoint the precise

moment a battery starts or ends discharging or charging.

A supplier cooperation approach is studied by Sawadsitang et al. [10] to split the cost of a drone delivery service. In order to serve as many consumers as possible, suppliers can work together to service their customers and share drones; after delivering cargo to the intended location, a drone can return to any supplier depot that is a part of the collaboration system. The suggested optimization model solves the package assignment problem as mixed integer programming.

Recently, there has been a focus on supporting drone operations with mobile charging stations to prevent drones from wasting time by returning to a stationary charging station for recharging during operation. As a result, there has been an increase in research on routing systems for drones and mobile charging stations that work together. Yu et al. developed algorithms to find the optimal route for systems integrating a mobile charging station and a drone [11]. By enabling the drone to be recharged while traveling by landing on stationary or mobile charging stations, they were able to solve the problem of the drone coverage range. The suggested algorithm calculates the optimal path for the drone to travel while also deciding when and where to land on charging stations. Additionally, it chooses the routes of unmanned ground vehicles and optimal locations for charging stations. Three scenarios which are many stationary stations, one mobile station, and many mobile stations are investigated. The first two problems are addressed using a generalized traveling salesman problem-based approach, while the third problem is addressed using an integer linear programming algorithm.

The Energy Minimizing and Range Constrained Drone Delivery Problem (ERDDP) is studied by Dukkanci et al. [12]. In the problem, drones are utilized to deliver goods to consumers, but the drones are transported by trucks that serve as launch points. The ERDDP contains three steps. These are choosing the launch points from a possible list of locations from which drones will visit to service some customers, assigning consumers to the launch points, and determining the speed at which drones will move between the customers and the launch points. In the research, a nonlinear model for the ERDDP is presented, which minimizes the total cost of operation by explicitly calculating the drone's energy consumption as a function of speed. The

restrictions of the delivery are a service time limit and the drone's range. The model is reformulated as a mixed-integer second order cone programming formulation.

Drones are used not only for supply purposes but also for various other purposes such as surveillance and mapping, exploring biological diversity, screening for creating road and railway routes, and military operations [3, 13]. The common features of these usage areas are that they do not need to return to a certain point during the operation, as in returning to a truck or warehouse during product delivery. The drone is expected to go to a charging station only when it needs to charge energy during its operation. Other than that, it should only move to the next target point (service point). In both [3] and [13], the routing of cooperative drone and mobile charging stations during such operations are investigated. Routing and determining appointment points for this system is an NP-hard problem. In [13], a two-stage strategy has been developed for combined route planning for unmanned aerial vehicle (UAV) and ground vehicle (GV), which charges the drone, to accomplish this task. The first stage calculates the recharging points that provide all points of interest in reaching distance by the UAV and feasible routes for both UAV and GV. In the second stage, they develop precise methods based on Mixed Integer Linear Programming (MILP) to plan optimal routes for the UAV and GV. Since the problem is NP-Hard, the authors also develop heuristic methods that give good, feasible results in a limited time. In short, to finish the operation, the UAV must visit each of the service points. It must recharge to visit every destination because of its limited energy capacity. In order to recharge as necessary, it can run into a GV that is only allowed to move on the road network.

The study of [3] is a following study of [13] with different methods to find a solution for operations involving an integrated system of a mobile charging station and a drone. In the study, the problem is modeled as a variant of the geometric hitting set problem. The aim is to select optimal rendezvous points for charging through certain candidate rendezvous points determined at intervals along the route where the unmanned ground vehicle (UGV) can move in a restricted area. As a result, these two articles are significant works that shed light on our study regarding content and solution approach.

In the system we are working on, drones are required to visit all predetermined service

points within a limited time and return to the starting point. While performing these operations, the drone must obtain the required energy by meeting with the mobile charging station in the same operation area. In this study, we aim to find a solution for both vehicles' combined optimal routes and determine optimal rendezvous points for charging in a system where drones and mobile charging stations work cooperatively, which has recently become very popular in the literature. A comprehensive literature review has been conducted on drone-based delivery systems and their applications, highlighting the difficulties and shortcomings of the system [2]. This review demonstrates how challenging it is to provide delivery services with even a small number of drones to a small number of customers. The complexity increases as the number of components in the system increases. We worked with a single drone, a single mobile charging station, and many service points in our system. In our problem, created with the environment and components mentioned above, finding the optimal routes where the drone and mobile charger will work integratedly and finding the order and location of these vehicles' rendezvous points make the problem very complex. We provided a solution for our problem using matheuristic methods. Our literature review shows the need for additional studies on cooperatively routing drones and MCs. Additionally, energy-efficient routing of drones and MCs and determination of their optimal speeds have received little attention in the literature. To fill these gaps in the literature, we aim to develop a matheuristic approach comprising heuristic methods and mathematical models to determine speeds and routes that minimize energy consumption for drones and mobile charging stations.

CHAPTER 3

PROBLEM DESCRIPTION

In previously studied drone and mobile charging station routing problems, it is seen that the simultaneous routing of the drone and the mobile charging station (MC), and the determination of their speeds during operations have not yet been studied much together. This study aims to find an integrated route for drone and MC and their optimal speeds during operation to minimize total energy consumption (or total operation time).

The system we designed in this study is as follows. Let there be s service points that the drone must visit on an obstacle-free 2D plane, one drone, and one mobile charging station. The drone should visit each service point in a limited system time (S^{Total}). The drone receives the energy needed to visit all these points within a given time from the MC, which moves on the same plane. The drone has a limited charging capacity. The MC's capacity to charge the drone is unlimited. It is also assumed that the MC has unlimited fuel. These two vehicles work in an integrated way in this system.

We work on two different objectives in this study. The first objective is to minimize the total energy consumption of the drone and the MC while completing the drone visits within the given time. In the scenario studied with this objective, it is aimed to find an integrated route and vehicles' speed that will minimize the total energy while the drone completes the operation in a limited system time. The second objective is to minimize the total system time within which the drone has to visit all service points. In this way, it is aimed to minimize the total system time under other constraints by ignoring the energy consumption in time-critical scenarios such as military operations. The final expected result is to obtain the optimal route and speeds of the drone and MC given one of the objectives.

We have a system with s service points. The drone and MC start their movement from the same starting point and finish at the same ending point. However, our solution approach is designed to allow the starting and ending points to be different as well. The system setup regarding the starting and ending points is as follows: The drone and MC are brought to the starting point by a vehicle such as a big truck. The drone initiates its visits to the service points from this point, while the MC starts moving towards the point to meet the drone from the same starting point. During its journey, if the drone's battery level is insufficient to reach the next service point, it meets with the MC to supply its energy needs. The drone charges at a specific rate in its charging station, determined by the coefficient σ. This coefficient indicates the time it takes for drone to receive a unit charge. So, the smaller the value of σ, the faster the drone charges. When the drone arrives at the charging station, it not only spends time charging but also landing on the station, connecting to the dock, detaching from the dock, and leaving the mobile charging station. The total time the drone spends on tasks other than charging in MC is specified as a constant (D) in the problem. After recharging, the drone departs from the MC and continues its movement towards the next service point, while the MC moves towards the next point where it will meet the drone when it requires energy. This process continues as the drone moves between service points and meets with the MC whenever it needs energy. The operation must be completed within a specified system time. Both vehicles return to their initial starting point at the end of the operation. During this operation, the drone and MC move at different speeds. The energy consumption varies based on the respective speeds, as illustrated in Figure 3.2. The rendezvous points and the speeds of the vehicles are significant determinants the total energy consumption and completion time of the operation.

Note that the drone and MC are assumed to consume energy while only in travel. In other words, it is assumed that they do not consume energy during their presence at service and rendezvous points for the drone and at rendezvous points for the MC. Therefore, in all cases where energy consumption is mentioned, only the energy consumed during travel is considered.

An example of the problem's output, which are the routes and battery levels of the drone and MC, can be seen in Figure 3.1.

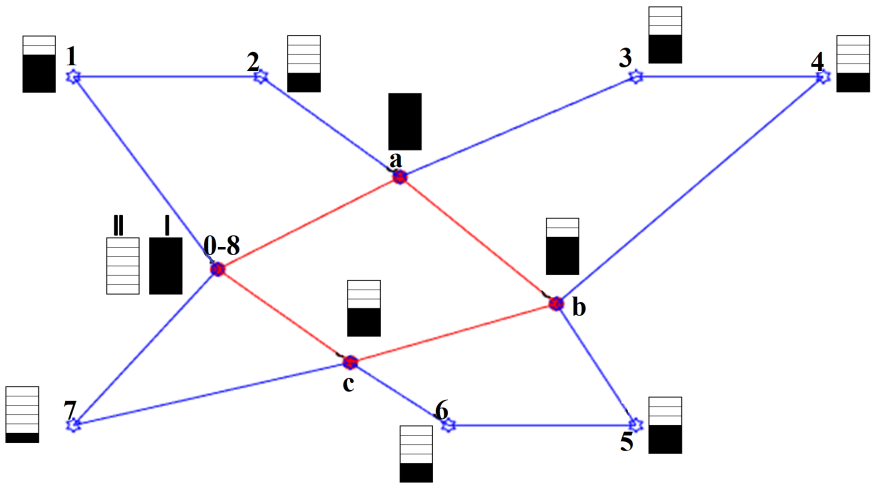

Figure 3.1: An illustration of the system with the optimal route of drone and MC and drone's battery level when departing from nodes

In this system operated by a single drone and a single MC, the blue route belongs to the drone, and the red route belongs to the MC. The points indicated by numbers which are 0 to 8 are given in the problem. These points are service points. The points indicated by the letters which are a, b, and c are the points we will obtain as the solution of the problem. These points are rendezvous points. In this example, there are 8 service points, 3 rendezvous points. The starting (0) and the ending (8) points are the same for both the drone and the MC, and they start and end their movements at this point. The points where the routes of the drone and the MC intersect are the rendezvous points (charge nodes), and the other points visited by the drone are the service points. The scales on each point in Figure 3.1 indicate the drone's battery level when leaving that point. At 0-8 points, the scale denoted by I indicates the drone's charge level at the starting point, and the scale denoted by II indicates the charge level at the ending point. Although the MC's energy consumption is considered in the problem, it has yet to be visualized as it has unlimited energy capacity. Indeed, the MC's energy consumption, like the drone's, affects the location of the rendezvous points. In this example, the drone starts from point 0, and visits points 1 – 2 – a – 3 – 4 – b – 5 – 6 – c – 7 in sequence before returning to point 8, the ending (starting) point. The MC starts from point 0 and visits points a – b – c before returning to point 8, the ending (starting) point. At {a b c}, the drone meets with the MC to meet its energy needs. The drone does not have to be fully charged when it meets the

MC in this system. It continues its travel by taking the required amount of charge to complete the operation. Because charging takes a while, unnecessary charging steals the operation time. The speeds of the drone and MC also significantly impact the location of the rendezvous points and completion time of the operation within the given time (S^{Total}). Therefore, one of the goals of this problem is to calculate the optimal speeds of the drone and MC within the specified limits in order to minimize the total energy consumption and complete the operation within the given time frame (S^{Total}).

This routing problem is a generalization of the Traveling Salesman Problem which is an NP-Hard problem. If the energy capacity of the drone is large enough, upper and lower speed limits of the drone are the same, energy consumption parameters (to be discussed later) of the MC are zero and there is no limit for the total system time (S^{Total}) of the system, our problem turns into TSP. Our model is a more comprehensive and complex version of the TSP. Therefore, our problem is also an NP-Hard problem.

The optimal rendezvous points of the drone and the MC, and their speeds to complete the system within S^{Total} are the output of the model described in Section 4.1.1.

The energy consumption function provided by Tokekar et al. in [14] is used to calculate the energy consumption of the drone and MC in this study. This energy consumption function given in (3.1) is a function that varies depending on the distance traveled by the vehicle specified by d and the speed during travel specified by v. α, β and γ are energy consumption parameters.

$$E = \alpha d v + \beta d + \gamma \frac{d}{v} \tag{3.1}$$

Figure 3.2 depicts the energy consumption function rate in terms of Joules per meter, as a function of speed in meters per second, according to (3.1). The parameter values used to obtain this figure are α = 0.5, β = 0.25, and γ = 0.3. As observed in Figure 3.2, the energy consumption initially decreases, but it starts to increase after a certain speed (optimal speed). The extent of this decrease and increase can vary with different parameter values.

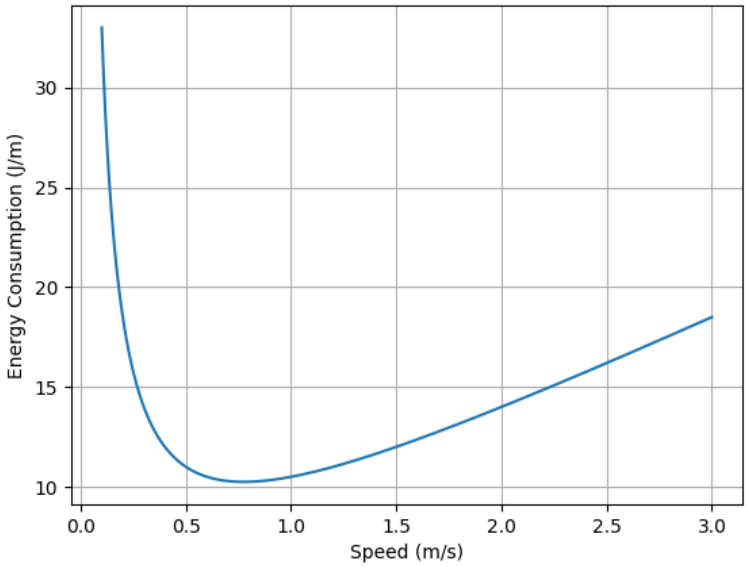

Figure 3.2: Energy consumption in Joule per meter as a function of the drone's and MC's speed

The optimal speed value that minimize the energy consumption according to (3.1) is determined as follows:

$$E'(v) = \alpha d - \gamma \frac{d}{v^2} = 0 \tag{3.2}$$

$$\alpha d = \gamma \frac{d}{v^2} \tag{3.3}$$

$$v^2 = \frac{\gamma}{\alpha} \tag{3.4}$$

$$v = \sqrt{\frac{\gamma}{\alpha}} \tag{3.5}$$

As shown in (3.2), the speed that minimizes the energy consumption is determined by setting the derivative of the energy consumption function with respect to velocity (v) equal to zero. According to the calculations in (3.2), (3.3), and (3.4), the constant velocity we will use is the square root of γ/α, as seen in (3.5).

These functions can be modified when working with drones and MC with different energy consumption functions. As in this study, energy consumption values for drones and MC can be determined using a common function with parameter adjustments for

both vehicles. Alternatively, two different functions can be used for the two vehicles. The model allows for this flexibility.

In this chapter, we have described the problem and mentioned its outputs. In this study, we are tackling an NP-hard problem, which is challenging to solve. Our goal is to find a route for the drone that minimizes energy consumption or total operation time. The energy required for the operation will be provided by an MC. In addition to determining the route between service points for the drone, we also aim to find the rendezvous points where the drone will meet the MC for recharging. These rendezvous points will be located between two service points at specific locations. Moreover, we aim to find the optimal travel speeds for both the drone and the MC during the operation, which will minimize their energy consumption (or total operation time). To achieve these solutions, we are adopting a solution approach called matheuristic that combines different heuristics and mathematical models.

In the next chapter, we will discuss our solution approach and its components in detail.

CHAPTER 4

SOLUTION METHODS

This study presents a multi-stage solution approach to achieve a comprehensive solution for the simultaneous routing of a drone and an MC. We develop a matheuristic consisting of an SOCP-based mathematical model and some heuristic methods. To the mathematical model, the order in which the drone visits service and rendezvous points is given as input. Similarly, the order in which the MC will visit the rendezvous points and the number of rendezvous points are given as input to the mathematical model. We use different methods to obtain and improve these orders. Different methods are employed to obtain various results, such as the number and order of rendezvous points and the best integrated route for the drone and the MC for the given inputs. Therefore, in this study, different heuristic algorithms and mathematical models were designed and used to reach a route that is hopefully close to the optimal solution. Heuristic algorithms and mathematical models used are briefly explained next.

1. Mathematical Models to Use When the Drone's Route and Rendezvous Points between Service Points are given

2. Initial Route Construction Algorithm (based on Nearest Neighbor Algorithm)

3. 2-Opt Algorithm

4. Rendezvous (Charging) Point Order Algorithms

   (a) Appointment Rendezvous Point Order Algorithm

   (b) Best Charge (Rendezvous Point) Sequence Algorithm

Let us briefly explain the logic of these algorithms and the mathematical model working together and then move on to the detailed explanations in this chapter. We will explain the example for the case where the objective is to minimize the total energy consumption. The same method applies to the objective where the total operation time is minimized.

There are s service points with known locations on the obstacle-free 2D plane. We can use a single mathematical model to find the integrated route of the drone and MC to minimize energy consumption. However, as explained before, this problem is an NP-hard problem. Therefore, we aim to solve this problem with a matheuristic in which the mathematical model and heuristics work together. The mathematical model we have created finds the locations of the rendezvous points that minimize the total energy and the optimal speeds of the drone and MC in this operation. The drone's route and intervals of two service points where the rendezvous points will be located (order of the rendezvous points) are given to the model. We do this with the help of other heuristic algorithms we have created. First, we start by determining an initial route and it is assigned as Current Route. We improve this route to find a good integrated route. Alternative routes to be used to improve the initial route are found by 2-Opt algorithms. For each of the initial and alternative routes, the following operations are applied. We need to find the order of the rendezvous points. To do this, candidate rendezvous points are identified for the drone's route. The mathematical model is run with this route and order of the rendezvous points. Then these candidates are eliminated one by one to find the final order of the rendezvous points. After completing all these processes, the integrated route is obtained. If there is a route that improves the initial route from the alternative routes, the route that provides the most improvement is assigned as the Current Route. The same operations are applied sequentially. If there is no improvement, the last determined Current Route is assigned as the Best Route and the matheuristic is completed.

For clarification purposes, a flowchart is provided in Figure 4.1, which represents how the matheuristic works.

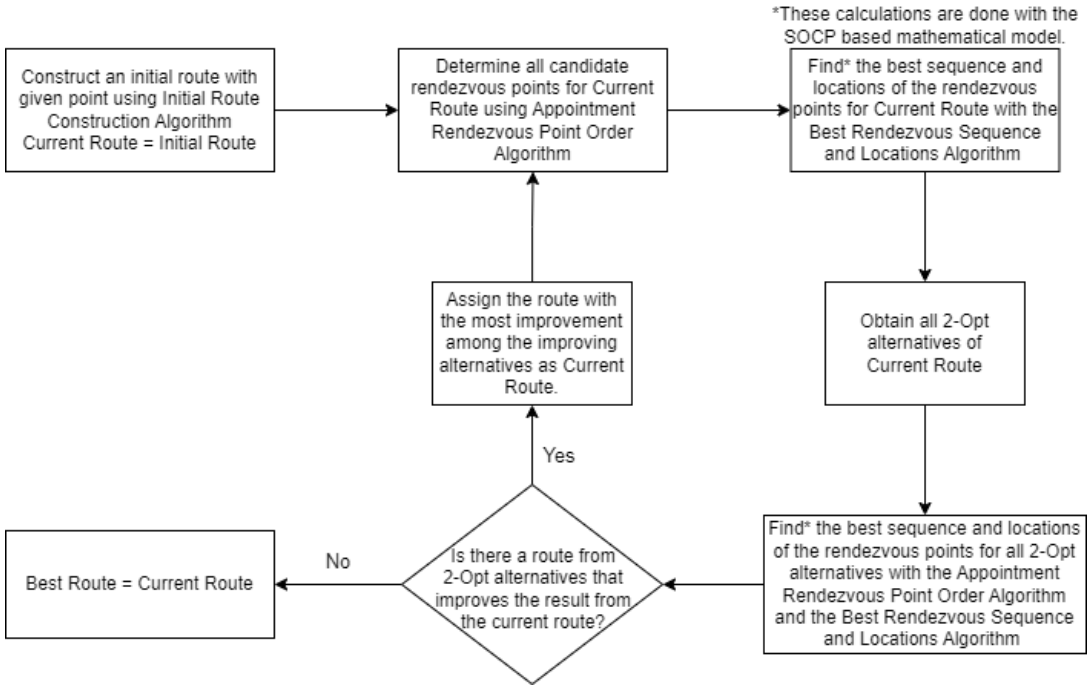

Figure 4.1: Flowchart for matheuristic

In this chapter, the mathematical models and heuristics that lead us to the solution is explained in order.

4.1 Mathematical Models to Use When the Drone's Route and Rendezvous Points between Service Points are given

In this study, to address both the complexity of finding the optimal drone route and the sequence of rendezvous points, we create mathematical models to solve the problem when the drone's route and the order of rendezvous points (information between which two service points it will be) are given as input. We explain these models in this section. The models aim to determine optimal locations for the rendezvous points under different objective functions. Finding these locations means finding the optimal integrated routes of the drone and MC for the given information. Additionally, the other outputs are the optimal speeds of both the drone and the MC during the operation under the given information for different objectives.

Let us explain briefly with an example. The drone's route is 0 - 1 - 2 - a - 3 - 4 - b - 5 - c - 6 - 7. 0 and 7 are the starting and ending points, and they are the same points. These points are the first visited service point. In this route, the points expressed by numbers are service points, and the points expressed by letters are rendezvous points. The locations of the service points are known, but the locations of the rendezvous points are unknown. Only the order of the rendezvous points is known. It means that rendezvous point a is known to be visited between the 2^{nd} and 3^{rd} service points. This route is given to the model, and the model is expected to find the locations of the rendezvous points.

The model can work with two different objective functions. The first objective is to minimize energy consumption, while the second objective is to minimize the total system time. We will describe models that include these two objective functions. These models contain nonlinear constraints. We reformulated the models second order cone programming (SOCP) problems which are convex optimization problems. We explain this SOCP-based mathematical model in this section.

4.1.1 Mathematical Model that Minimizes the Total Energy Consumption

The energy consumption function described in (4.1) is used in our model which is equivalent to (3.1), but uses travel time t instead of speed v.

$$E = \alpha \frac{d^2}{t} + \beta d + \gamma t \tag{4.1}$$

The system consists of n+1 nodes which are elements of N. S is the set of service points to be visited by the drone, C is the set of rendezvous points to be visited by the drone and MC. In addition, there are starting and ending points indicated by 0, n+1. The set N consists of S, C, and 0, n+1. The drone's route includes all the points in the problem, while MC's route only includes rendezvous points, the starting and ending points. The set of arcs (i, j) used in MC's route forms the set M. The set MN comprises all the nodes visited by MC. The set MN consists of C and 0, n+1. ML is the set containing the last visited rendezvous point by MC. This set is defined to calculate the completion time of MC's operation.

In our model, the locations of all service points, the energy consumption function parameters of the drone and MC, the time limit of the system, the drone's charging parameter (which determines the speed at which the drone charges), and the max-min speeds that the drone and MC can travel are provided as parameters. The drone's route and intervals of two service points where the rendezvous points will be located are given to the mathematical model. In other words, the sequence in which the drone will visit all points is known, and the route is fixed in the model. Therefore, the route is defined as 0 - 1 - 2 - 3 - ..., where the specified numbers represent the order in which the drone will visit the points. Only the locations of the rendezvous points are determined by the model. Some decision variables in our model are used only to reach a solution, while others provide us with the outputs we seek. The values we expect as output from this model are the locations of rendezvous points, the distances traveled and travel times by the drone and MC between nodes, which we will use to calculate the speed of the drone and MC.

The drone and MC move in a 2D obstacle-free plane in our model. Euclidean distance is used to find the distance between points in this plane, and this makes the energy consumption constraints for the drone and MC nonlinear. Therefore, our model is nonlinear. To solve the model efficiently, we reformulate it as a second order cone programming which is explained in Section 4.1.3.

Sets and Subsets:

N: all nodes visited and visit order of the drone $\rightarrow \{0, 1, 2, \ldots, n+1\} (0 \equiv n+1)$

C: Rendezvous (Charge) Nodes: Rendezvous nodes of Drone and MC for the purpose of charging

S: Service Nodes: Nodes visited by the drone to serve $\rightarrow \{1, \ldots, n\} \setminus C$

M: Arcs used by MC in the path

MN: Nodes in MC Path $\rightarrow MN = C \cup \{0, n+1\}$

ML: Last rendezvous node (Last element of C)

Parameters:

SX_i: x coordinate of service node $i \in S$

SY_i: y coordinate of service node $i \in S$

FX_0: x coordinate of starting node ($FX_0 = SX_1$)

FY_0: y coordinate of starting node ($FY_0 = SY_1$)

α^d , β^d , γ^d: energy consumption function parameters of the drone

α^{MC} , β^{MC} , γ^{MC}: energy consumption function parameters of the MC

S^{Total}: Maximum time to complete the operation (time limit of the system)

B_{Max}: Battery capacity of the drone

B_{Min}: Minimum possible battery level of the drone

σ: battery recharge function parameter of the drone

D: A constant time taken for the drone to dock at the charger and leave from the charger

T_{Max}: The full energy level of the MC

V_{Max}^{MC}: Maximum speed of the MC

V_{Min}^{MC}: Minimum speed of the MC

V_{Max}^{d}: Maximum speed of the drone

V_{Min}^{d}: Minimum speed of the drone

Decision Variables:

x_i: x coordinate of node i, $i \in N$

y_i: y coordinate of node i, $i \in N$

E_i: the energy consumed by the drone during its travel from node i to node i+1, $i \in N \setminus \{n+1\}$

B_i^d: Battery (energy) level of the drone when leaving node i, $i \in N \setminus \{n+1\}$

B_{n+1}^d: Battery (energy) level of the drone when entering node n+1

d_i^d: Distance travelled by the drone while travelling from node i to node i+1, $i \in N \setminus \{n+1\}$

s_i: The time the drone leaves node i, $i \in N$ (the time MC leaves rendezvous point i, $i \in C$)

c_i: Time spent by the drone while being charged at node i, $i \in C$

w_i^d: Waiting time of the drone without being charged at node i, $i \in C$

t_i^d: Time spent by the drone while traveling from node i to node i+1, $i \in N$

d_i^{MC}: Distance travelled by the MC from node i to the next visiting node, $i \in MN$

B_i^{MC}: Battery (energy) level of the MC when leaving node i, $i \in C \cup \{0\}$

B_{n+1}^{MC}: Battery (energy) level of the MC when entering node n+1

t_i^{MC}: Time spent by the MC while traveling from node i to the next rendezvous node, $i \in MN$

w_i^{MC}: Waiting time of the MC at node i, $i \in MN$

$$\text{min} \qquad \sum_{i=0}^{n} E_i + (B_0^{MC} - B_{n+1}^{MC}) \tag{4.2}$$

subject to

$$x_i = SX_i \qquad \forall i \in S \tag{4.3}$$

$$y_i = SY_i \qquad \forall i \in S \tag{4.4}$$

$$x_0 = FX_0 \tag{4.5}$$

$$y_0 = FY_0 \tag{4.6}$$

$$x_{n+1} = x_0 \tag{4.7}$$

$$y_{n+1} = y_0 \tag{4.8}$$

$$E_i = \alpha^d \frac{(d_i^d)^2}{t_i^d} + \beta^d d_i^d + \gamma^d t_i^d \qquad \forall i \in N \setminus \{n+1\} \tag{4.9}$$

$$(d_i^d)^2 \geq (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \qquad \forall i \in N \setminus \{n+1\} \tag{4.10}$$

$$B_{i-1}^d - B_i^d \geq E_{i-1} \qquad \forall i \in S \cup \{n+1\} \tag{4.11}$$

$$B_i^d \geq B_{Min} \qquad \forall i \in N \setminus \{0\} \tag{4.12}$$

$$B_0^d = B_{Max} \tag{4.13}$$

$$B_i^d \leq B_{Max} \qquad \forall i \in N \setminus \{0\} \tag{4.14}$$

$$B_{i-1}^d \geq E_{i-1} \qquad \forall i \in C \tag{4.15}$$

$$c_i = (B_i^d - (B_{i-1}^d - E_{i-1}))\sigma \qquad \forall i \in C \tag{4.16}$$

$$s_0 = 0 \tag{4.17}$$

$$c_0 = 0 \tag{4.18}$$

$$s_i = s_{i-1} + t_{i-1}^d + c_i + w_i^d + D \qquad \forall i \in N \setminus \{0\} \tag{4.19}$$

$$s_{n+1} \leq S^{Total} \tag{4.20}$$

$$d_i^d \leq t_i^d V_{Max}^d \qquad \forall i \in N \setminus \{n+1\} \tag{4.21}$$

$$d_i^d \geq t_i^d V_{Min}^d \qquad \forall i \in N \setminus \{n+1\} \tag{4.22}$$

$$B_i^{MC} - B_j^{MC} \geq \alpha^{MC} \frac{(d_i^{MC})^2}{t_i^{MC}} + \beta^{MC} d_i^{MC} + \gamma^{MC} t_i^{MC} \qquad \forall (i,j) \in M \tag{4.23}$$

$$(d_i^{MC})^2 \geq (x_j - x_i)^2 + (y_j - y_i)^2 \qquad \forall (i,j) \in M \tag{4.24}$$

$$B_0^{MC} = T^{Max} \tag{4.25}$$

$$d_i^{MC} \leq t_i^{MC} V_{Max}^{MC} \qquad \forall i \in MN \setminus \{n+1\} \quad (4.26)$$

$$d_i^{MC} \geq t_i^{MC} V_{Min}^{MC} \qquad \forall i \in MN \setminus \{n+1\} \quad (4.27)$$

$$s_i - c_i - w_i^{MC} = \sum_{\substack{j \in MN \\ j < i}} (t_j^{MC} + c_j + w_j^{MC}) \qquad \forall i \in C \quad (4.28)$$

$$s_i + t_i^{MC} \leq S^{Total} \qquad i \in ML \quad (4.29)$$

$$E_i, t_i^d, d_i^d, s_i, c_i, w_i^d, B_i^d, \geq 0 \qquad \forall i \in N \quad (4.30)$$

$$d_i^{MC}, B_i^{MC}, t_i^{MC}, w_i^{MC} \geq 0 \qquad \forall i \in N \quad (4.31)$$

The explanations about the constraints can be seen below.

Thanks to the constraints (4.3) and (4.4), the locations of the service nodes are fixed and only the charging appointment points remain as decision variables. Constraints (4.5) and (4.6) determine that the starting point and the first visited service point are the same. Constraints (4.7) and (4.8) determine that the ending point and the starting point are the same. However, starting and ending points can easily be set to different service points. We can ensure that these points are found by the model by removing constraints (4.5), (4.6), (4.7), and (4.8) from the model. Constraint (4.9) computes the energy consumption of the drone during its travel from node i to i+1. Constraint (4.10) states that the distance the drone travels between nodes i and i+1 cannot be shorter than the distance between these two nodes. Constraints (4.11) ensure that the difference in battery level between two nodes with non rendezvous points will be greater than or equal to the energy consumed during the movement. Constraint (4.13) makes sure that the battery level of the drone is full (100% in this problem) at the starting point. Constraints (4.12) and (4.14) guarantee that the charge level of the drone while leaving every node is within the allowable limits. Constraints (4.15) ensure that the difference in battery level between the service point visited before the rendezvous point and the rendezvous point will be greater than or equal to the energy consumed during the movement. Constraints (4.16) calculate the amount of time the drone spends while meeting its charging needs (it indicates only charging time) at MC at rendezvous points. Thanks to this constraint, the drone is charged as much as it needs, it does not steal from the limited system time with overcharging (charging more than necessary). Constraints (4.17) and (4.18) make sure that the departure time

from the starting point (s_0) and the time at the starting point (c_0) are 0. The relation of the leaving time from node i with other times which are time to leave the previous node, travel time between these two nodes, charging time, waiting time without charging and constant spent time (D) for works related to charging is specified in constraint (4.19). Constraint (4.20) limits the completion time of the drone's operation s_{n+1} by S_{Total}. Constraints (4.21) and (4.22) ensure that the distance between points is constrained by the distance the drone will travel at a speed between its speed limits. Constraints (4.23) make sure that the difference in battery level between the two rendezvous points of the MC will be greater than or equal to the energy consumed during movement. Constraints (4.24) state that the distance the MC travels between node i and the next visiting node cannot be shorter than the distance between these two nodes. Constraint (4.25) specifies the energy level of the MC at the starting point. Constraints (4.26) and (4.27) restrict the distance between two rendezvous points by the distance the MC will travel at a speed between its speed limits. The relation of the time that the MC comes to the rendezvous point with other times is specified in constraints (4.28). The left-hand side of these constraints represents MC's arrival time at the rendezvous point. The right-hand side equals the total time spent before arriving at the current rendezvous point. Constraint (4.29) limits the completion time of the MC's operation by S_{Total}. (s_i is the time drone and MC leave the last rendezvous point, t_i^{MC} is the time that MC travels from the last visited rendezvous point to the ending point, where i is the last visited rendezvous point .) Constraints (4.30) and (4.31) are the nonnegativity constraints.

With this model, we obtain the necessary outputs for the drone to provide service to all given service points within a limited time when the order of the points the drone will visit is given.

4.1.2 Mathematical Model that Minimizes the Total System Time (S^{Total})

Minimizing the Total System Time is a top priority for many operations. In operations where time is of utmost importance, such as military operations, it is crucial to not only minimize energy consumption but also the operation time. In such situations, this model can be used to provide us with the shortest possible time to complete the

operation based on the given parameters. The outputs of this model are the integrated route that ensures the operation is completed in the shortest time and the speeds of the drone and MC during the operation.

In our previous model, the time limit of the system is provided as a parameter shown as in constraint (4.20). Accordingly, outputs that minimize the total energy consumption function in objective function (4.2) are obtained. Additionally, we have worked on developing a different objective function to obtain the minimum possible time to complete the operation. As can be seen in objective function (4.2), we aimed to minimize S_{Total}. As a result, S_{Total} was converted from a parameter to a decision variable and all constraints remain the same as in (4.33).

$$\text{min} \quad S^{Total} \tag{4.32}$$

subject to

$$\text{(4.3) - (4.31)} \tag{4.33}$$

4.1.3 SOCP Reformulations of the Mathematical Models

SOCP (Second Order Cone Programming) problem is a convex optimization problem that can be used to solve various real life problems [15].

The mathematically representation of SOCP is as follows:

$$\text{minimize} \quad k^T x \tag{4.34}$$

$$\text{subject to} \quad \|D_i x + f_i\| \leq a_i^T x + b_i, \forall i = 1, 2, 3, ..., n \tag{4.35}$$

In the formulation given by 4.34 and 4.35, $x \in \mathbb{R}^n$ is the decision variable vector, $k \in \mathbb{R}^n$, $D_i \in \mathbb{R}^{n_i \cdot n}$, $f_i \in \mathbb{R}^{n_i}$, $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ are parameters of the problem of appropriate sizes. Constraints (4.35) are called second order cone constraints. Note that any linear inequality can be written in the form given in (4.35).

In our problem, the constraints related to energy consumption functions and distances

between traveled nodes are nonlinear constraints. All the remaining constraints in the model are linear. Even though the distance-related constraints are nonlinear, their forms are the same as the constraint in (4.35). Therefore, we only reformulate energy consumption functions and write them as second order cone constraints.

Thus, (4.9) and (4.23) are needed to be converted to second order cone constraints to reformulate the problem as an SOCP. To facilitate this transformation, we need to introduce separate decision variables for the two constraints. We denote the decision variable for the energy consumption function of the drone in (4.9) as z_i^d and the decision variable for the energy consumption function of the MC in (4.23) as z_i^{MC}. First, let's create an second order cone constraint for (4.9). In (4.9), $\frac{(d_i^d)^2}{t_i^d}$ is replaced with z_i^d. Instead of this constraint, the constraints specified in (4.36) and (4.37) are obtained.

$$E_i = \alpha^d z_i^d + \beta^d d_i^d + \gamma^d t_i^d \qquad \forall i \in N \setminus \{n+1\} \quad (4.36)$$

$$z_i^d \geq \frac{(d_i^d)^2}{t_i^d} \quad (4.37)$$

We can rewrite the (4.37) as follows:

$$z_i^d t_i^d \geq (d_i^d)^2 \quad (4.38)$$

and it can be written as second order cone constraint as follows:

$$\left\| \begin{pmatrix} d_i^d \\ \frac{z_i^d - t_i^d}{2} \end{pmatrix} \right\| \leq \frac{z_i^d + t_i^d}{2} \quad (4.39)$$

Open form of this second order cone constraint is seen in (4.40).

$$(d_i^d)^2 + 0.25(z_i^d - t_i^d)^2 \leq 0.25(z_i^d + t_i^d)^2 \qquad \forall i \in N \setminus \{n+1\} \quad (4.40)$$

$$z_i^d \geq 0 \qquad \forall i \in N \quad (4.41)$$

Let us apply the same procedures for the constraint (4.23), which includes the MC's energy consumption function, to convert second order cone constraint. Constraints in (4.42) and (4.43) are obtained by replacing $\frac{(d_i^{MC})^2}{t_i^{MC}}$ with z_i^{MC} in Constraint (4.23).

$$B_i^{MC} - B_j^{MC} \geq \alpha^{MC} z_i^{MC} + \beta^{MC} d_i^{MC} + \gamma^{MC} t_i^{MC} \qquad \forall \{i,j\} \in M \quad (4.42)$$

$$z_i^{MC} \geq \frac{(d_i^{MC})^2}{t_i^{MC}} \qquad \forall i \in MN \setminus \{n+1\} \quad (4.43)$$

We can write constraint in (4.43) as SOCP constraint as follows:

$$\left\| \begin{pmatrix} d_i^{MC} \\ \frac{z_i^{MC} - t_i^{MC}}{2} \end{pmatrix} \right\| \leq \frac{z_i^{MC} + t_i^{MC}}{2} \tag{4.44}$$

Open form of this SOCP constraint is as follows:

$$(d_i^{MC})^2 + 0.25(z_i^{MC} - t_i^{MC})^2 \leq 0.25(z_i^{MC} + t_i^{MC})^2 \qquad \forall i \in MN \setminus \{n+1\} \tag{4.45}$$

$$z_i^{MC} \geq 0 \qquad \forall i \in N \tag{4.46}$$

The SOCP model is created by deleting the constraints (4.9) and (4.23) from the mathematical model and adding constraints (4.36), (4.40), (4.42), and (4.45) instead. z_i^d and z_i^{MC} are nonnegative variables. Thus, constraints (4.41) and (4.46) are also added to the model.

4.2 Matheuristic and Penalty Approaches

We developed a matheuristic to find the routes of the drone and the mobile charger and rendezvous points in a system where the drone is tasked with visiting service points and works with the MC integratedly to obtain the necessary energy. This matheuristic consists of multiple sub-heuristic algorithms and solves a mathematical model which is an SOCP iteratively. All of these sub-heuristics and the mathematical model work together to ultimately provide us with the routes of the drone and MC that minimize total energy consumption (or the operationt completion time), as well as the order and location of the rendezvous points where the two vehicles meet for charging.

Before explaining the steps of the mathheuristic separately, we explain how the entire solution method works cohesively from start to finish. We mentioned in Chapter 3 that the locations of service points, energy consumption parameters of the drone and MC, the time limit of the system for every minimization problem (which we also examined as a decision variable), maximum and minimum speed limits of the drone and MC, and initial battery level and battery limits of the two vehicles are all known inputs to our problem. Using an *Initial Route Construction Algorithm based on the Nearest Neighbor Algorithm*, an initial route is determined without considering the rendezvous points. In this step, we consider only the drone's route over

service points. Then the best rendezvous points are tried to be determined, and the energy consumption of the resulting solution is recorded. This initial route's solution is improved by considering alternative routes using *the 2-Opt Algorithm*. This improvement heuristic method aims to generate alternative routes that can enhance the solution. The number of alternative routes generated increases with the increase in the number of nodes. To reduce the solution time by reducing the number of alternative routes, we use a *Speed-up Method*. So far, we have only discussed the routing of the drone over service points. Next, we will explain *the Rendezvous (Charging) Point Order Algorithms* and the mathematical model we developed to determine the order and location of the rendezvous points to satisfy the drone's charging needs. Using *the Appointment Rendezvous Point Determination Algorithm*, we determine the rendezvous points for every alternative route constructed.

To generate alternative rendezvous points for any sequence of service points, we create new ones before and/or after them (after visiting the previous service point and/or after visiting the next service point). By doing so, we increase the options and aim to explore solutions that can yield better objective values using *the Appointment Rendezvous Point Order Algorithm*, which incorporates a consistent prediction method. From the list containing the candidate rendezvous points expanded with alternative rendezvous points, we aim to find the list of rendezvous points and their locations that minimize the objective function by eliminating candidates one by one from the list using the SOCP approach in our mathematical model. This algorithm is *the Best Charge (Rendezvous Point) Sequence and Locations Algorithm*.

4.2.1 Initial Route Construction Algorithm (based on Nearest Neighbor Algorithm)

To generate the optimal routes for both the drone and the MC, it is necessary to first create an initial route for the drone according to the given service points. This facilitates the identification of sequences of rendezvous points that will require energy reinforcement during the operation. This algorithm constructs the drone's initial tour based on the Nearest Neighbor Algorithm. The process begins by selecting one of the service points and starting the algorithm. The next visited service point is determined

as the one closest to the current point. This selection is repeated after each service point. In this case, distances are calculated using the Euclidean distance.

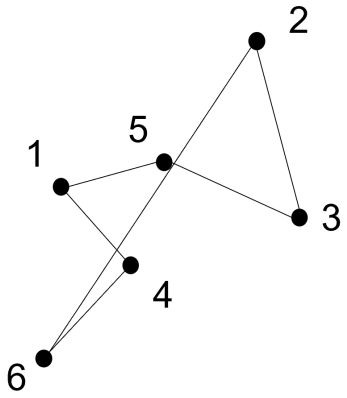

Figure 4.2: Illustration of the route with initial point 3

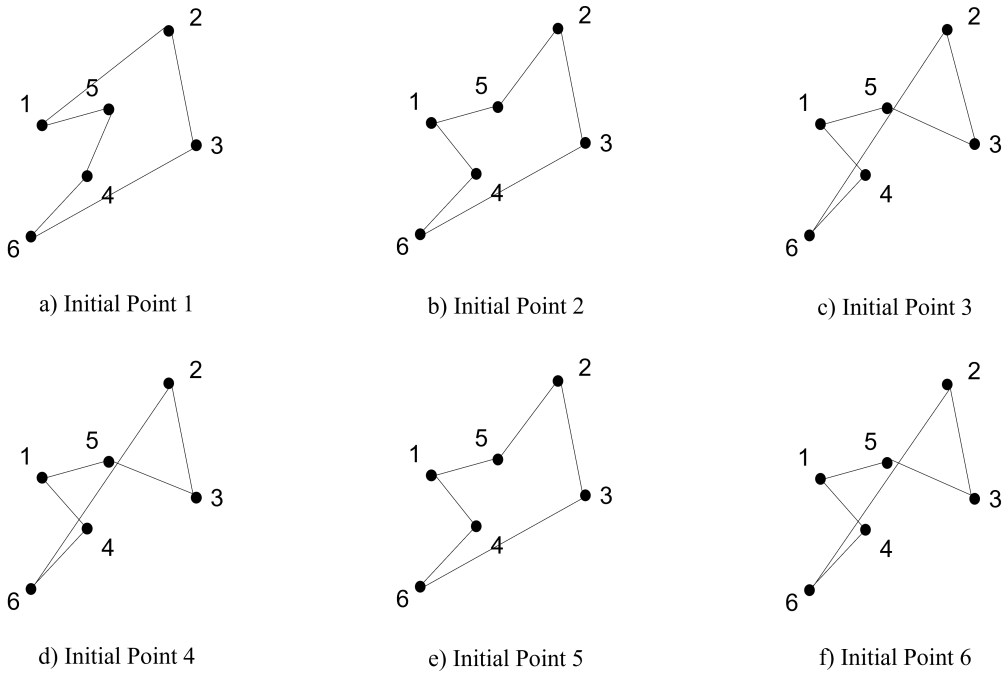

Figure 4.3: The illustration of all routes with different initial points

Considering the example given in Figure 4.2, {1, 2, 3, 4, 5, 6} are the service points. Starting with point 3, which will be the first point visited, the algorithm proceeds by selecting the point closest to it from the remaining set {1, 2, 4, 5, 6} based on the Euclidean distance. Let the second visited point be 5. Sequentially, the algorithm selects point 1, which is the closest to 5, followed by point 4, which is the closest to 1, then point 6, which is the closest to 4, point 2, which is the closest to 4, and finally returns to the starting point, which is point 3, completing the route. The resulting route 3 - 5 - 1 - 4 - 6 - 2 - 3 is the drone's initial route created based on the Nearest Neighbor Algorithm, starting from service point 3. The algorithm continues by designating each service point in the set as the initial service point and creating a route for each of them. Finally, the one with the shortest total distance is chosen as the drone's initial route among all these routes. Figure 4.3 displays all routes with different initial points. The shortest routes from all the generated routes are in Figure 4.3b-c. Algorithm 1 briefly describes the Initial Route Construction Algorithm, which is based on the Nearest Neighbor Algorithm.

Algorithm 1 Initial Route Construction Algorithm based on Nearest Neighbor Algorithm

s: # of the service points

Inputs: List of service points (SP_{List}) and coordinates of these service points

Initial Route = []

for $i = 1 \rightarrow s$ **do**

Path = []

Current Point = i

Add *Current Point to Path.*

for $j = 1 \rightarrow s - 1$ **do**

Find *nearest neighbor point k by Euclidean Distance according to coordinates of the service point to Current Point.*

Append *k to Initial Route.*

Current Point = k

Remove *k from* SP_{List}

end

if *Total Distance of Path* $\leq$ *Total Distance of Initial Route* **then**

Initial Route = Path

else

Continue with the same Initial Route.

end

end

4.2.2 2-Opt Algorithm

The Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) can be solved using mixed-integer linear programming models. However, as the number of visited points increases, the solution of the problem becomes increasingly challenging. These problems are NP-Hard problems due to their inherent complexity. In addition to mathematical modeling, heuristic approaches can be used to solve these problems. Since our problem in this study encompasses both TSP and VRP, solving it solely through mathematical modeling is highly difficult. Therefore, our solution ap-

proach involves using the 2-Opt Algorithm to improve the current route of the drone. In this algorithm, two non-adjacent edges in an existing route are removed, and then two new edges are created to form a new route. It is crucial to ensure that the created edges do not form sub-tours. An example of this is illustrated in Figure 4.4.

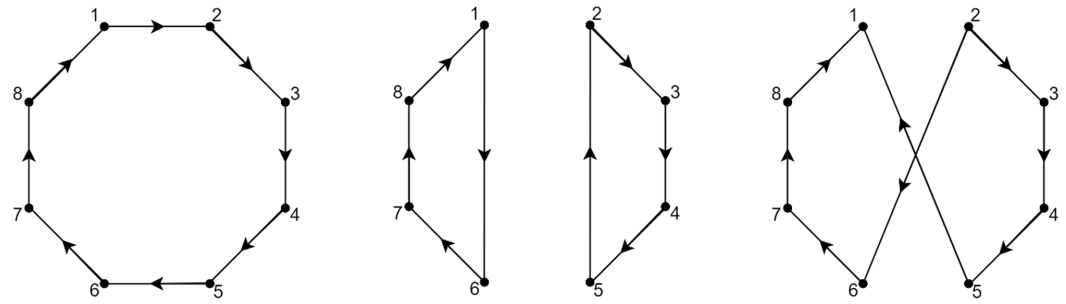

Figure 4.4: The illustration of 2-Opt Algorithm ((a) Initial tour, (b) New tour with subtours, (c) New legit tour)

In the example shown in Figure 4.4, where our initial route is 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 - 1, the edges between 1 – 2 and 5 – 6 are removed. In such a situation, if we create edges 1 – 6 and 2 – 5, we will end up with a route that contains two subtours, as seen in Figure 4.4b. Instead, when we combine edges 1 – 5 and 2 – 6, a legit route is formed, as seen in Figure 4.4c. As observed, when the 2-Opt Algorithm is applied, and two edges are removed, each scenario can result in one legit route and one route that contains a subtour.

In a system containing n service points, there are n edges. This means that $\binom{n}{2}$ pairs of edges can be removed. However, out of these pairs, n of them are adjacent edges, and if removed, they would not result in any different and complete routes. Therefore, when we run the algorithm, there are $\binom{n}{2} - (n)$ 2-Opt moves. In each 2-Opt move, only one legitimate route can be generated. Therefore, the output of the 2-Opt algorithm will yield the same number of alternative routes.

The 2-Opt algorithm is only applied to the drone's route consisting of service points in this thesis. The locations and sequence of rendezvous points are obtained from other heuristics and SOCP based mathematical models.

In this study, the 2-Opt algorithm is used as described in Algorithm 2. Following the generation of the initial route, all 2-Opt moves are obtained using Algorithm 2. Alternative Routes obtained by 2-Opt Algorithm array includes all alternative routes obtained by 2-Opt Algorithm. Subsequently, using the Rendezvous (Charging) Point Order Algorithms explained in the next stage, and the SOCP based mathematical model, routes for the drone and MC, as well as the sequence and locations of rendezvous points, are determined for each route. The alternative that yields the lowest total energy consumption among the 2-Opt moves replaces the initial route. These procedures are then applied to the newly obtained route. This process continues sequentially for the new route until no further improvement can be achieved. Through these iterations, the best possible results that can be obtained using these heuristics are attained.

Algorithm 2 2-Opt Algorithm

n: # of all nonadjacent edge pairs of Initial Route

Inputs: Initial Route

Alternative Routes obtained by 2-Opt Algorithm = [Initial Route]

Nonadjacent Edge Pairs = Combination of all nonadjacent edges of Initial Route

for $i = 1 \rightarrow n$ **do**

> ***Remove*** *Nonadjacent Edge Pairs[i] from Initial Route*
>
> ***Connect*** *idle nodes, whose edges are broken, without creating sub-tours*
>
> ***Append*** *this new route to Alternative Routes obtained by 2-Opt Algorithm*

end

4.2.2.1 Speed – Up Method for 2-Opt Algorithm

As the number of service points in the system increases, the number of routes generated by the 2-Opt algorithm also increases quadratically. For example, in a system with 10 points, there are $\binom{n}{2} - (n) = \binom{10}{2} - (10) = 35$ new routes, while in a system with 30 points, there are 405 new routes, and in a system with 60 points, there are 1710 new routes. That is to say, the number of new routes grows depending on the number of service points, which significantly extends the computational time. To reduce the computational time, it is necessary to eliminate alternatives generated by

the 2-Opt algorithm that is not believed to improve the solution. To overcome this challenge, a speed-up method is employed in this study.

In the literature, various methods have been proposed to accelerate k-Opt algorithms. Nilsson [16] attempts to eliminate routes by comparing the sum of the lengths of the removed edges with the sum of the lengths of the newly created edges. In a scenario where the edges $(a1, a2)$ and $(a3, a4)$ are removed and replaced by the edges $(a1, a4)$ and $(a2, a3)$, the sums $dist(a1, a2) + dist(a3, a4)$ and $dist(a1, a4) + dist(a2, a3)$ are compared. If $dist(a1, a2) + dist(a3, a4) > dist(a1, a4) + dist(a2, a3)$, it is considered as a route that can potentially improve the solution and is added to the list of alternatives. On the other hand, if it does not satisfy this condition, the route is not added to the list.

According to Steiglitz and Weiner [17], the situations that do not satisfy $dist(a1, a2) > dist(a2, a3)$ can be removed from the list of routes generated by the 2-Opt algorithm. This condition allows for a significant elimination.

In this study, we use the speed up method of Nilsson to generate our list of 2-Opt routes, as seen in Algorithm 3.

Based on this observation, in this study, we compare all routes generated by the 2-Opt algorithm with the initial route. Routes that are shorter than the initial route are considered as routes that can potentially improve the solution. Therefore, we only apply the remaining steps on routes that are shorter than the initial route rather than on all 2-Opt moves. We select the route that provides the most improvement among these routes and generate new 2-Opt moves based on this route in the next step. We continue the process by applying the elimination through the same speed-up method.

Algorithm 3 Speed-Up Method for 2-Opt Algorithm

a: # of alternative routes obtained by 2-Opt Algorithm (size of the array)

Inputs: Initial Route and Alternative Routes obtained by 2-Opt Algorithm

Alternative Routes = Alternative Routes obtained by 2-Opt Algorithm

Routes obtained by Speed-Up Method = [Initial Route]

Calculate *Initial Route's Distance*

for $i = 1 \rightarrow a$ **do**

  Calculate *Alternative Route[i]'s Distance*

  if *Alternative Route[i]'s Distance < Initial Route's Distance* **then**

    Append *Alternative Route[i] to Routes obtained by Speed-Up Method*

  end

end

4.2.3 Rendezvous (Charging) Points Determination Algorithms

We have constructed the initial route for the drone using *the Initial Route Construction Algorithm*. We obtained alternative routes that could potentially improve the solution by applying *the 2-Opt Algorithm* on this route. Now, we need to determine the sequence and locations of rendezvous points where the drone will meet the MC to charge on each of these routes in a way that tries to minimize the total energy consumption. To achieve this, we will use different algorithms sequentially.

Firstly, we will use *the Appointment Rendezvous Point Order Algorithm* to identify the visit sequence where the drone requires charging based on certain assumptions (which will be detailed in Section 4.2.3.1). This algorithm determines the visit order where the drone's energy demand arises, i.e., when its charge drops below zero. Subsequently, new rendezvous points are determined as alternatives to these identified points, either in front of or behind them. This topic will be further illustrated with examples in the following section.

Next, with the enriched set of alternatives, we will use *the Best Charge (Rendezvous Point) Sequence and Locations Algorithm* to select the most suitable sequence and determine the locations of these rendezvous points. This algorithm utilizes the SOCP-based mathematical model. Detailed examples of this algorithm will be provided in Section 4.2.3.2.

4.2.3.1 Appointment Rendezvous Point Order Algorithm

In order to determine the final sequence and location of rendezvous points for the drone and the MC, we need first to identify some candidate rendezvous points. We need to calculate the drone's energy consumption between service points to determine these points. To calculate the drone's energy consumption, we utilize the Energy Consumption Function given in (3.1) in Chapter 3. α, β, and γ are energy consumption parameters, where d represents the distance covered by the drone, and v represents the drone's velocity.

In this algorithm, the drone's velocity is assumed to be constant. This constant velocity is determined by calculating the value that minimizes the drone's energy consumption. Thus, it is the optimal speed of the drone, as seen in (3.5).

In this algorithm, based on the energy consumption function in (3.1) and the optimal speed determined in (3.5) in Chapter 3, the energy consumption between each visited point along the route that does not have any rendezvous points is calculated. It is assumed that the initial service point is the starting point of the drone's operation, and its energy is assumed to be fully charged (100%) at the beginning. The points at which the drone's energy will be depleted are determined based on the calculated energy consumption. Under the given assumptions, it is assumed that the drone needs to recharge at one of the candidate rendezvous points within this interval. The location of this rendezvous point is not determined in this algorithm. Therefore, it is assumed that the drone charges at the point visited before from these two visited points and departs from that point with a full charge. The next location where the fully charged drone will require a recharge is determined similarly. This process is repeated until all the necessary rendezvous points for the drone to complete the route are determined.

To illustrate this process with an example as in Figure 4.5, let us consider a route without any rendezvous points: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12. This route is seen in Figure 4.5a. All the points are service points. The battery capacity of the drone is 100%. Table 4.1 shows the energy consumption between the points. According to these consumption values, the energy consumption for route 1 – 2 – 3 – 4 is 75%. The energy consumption between 4 and 5 is 35%. It appears that

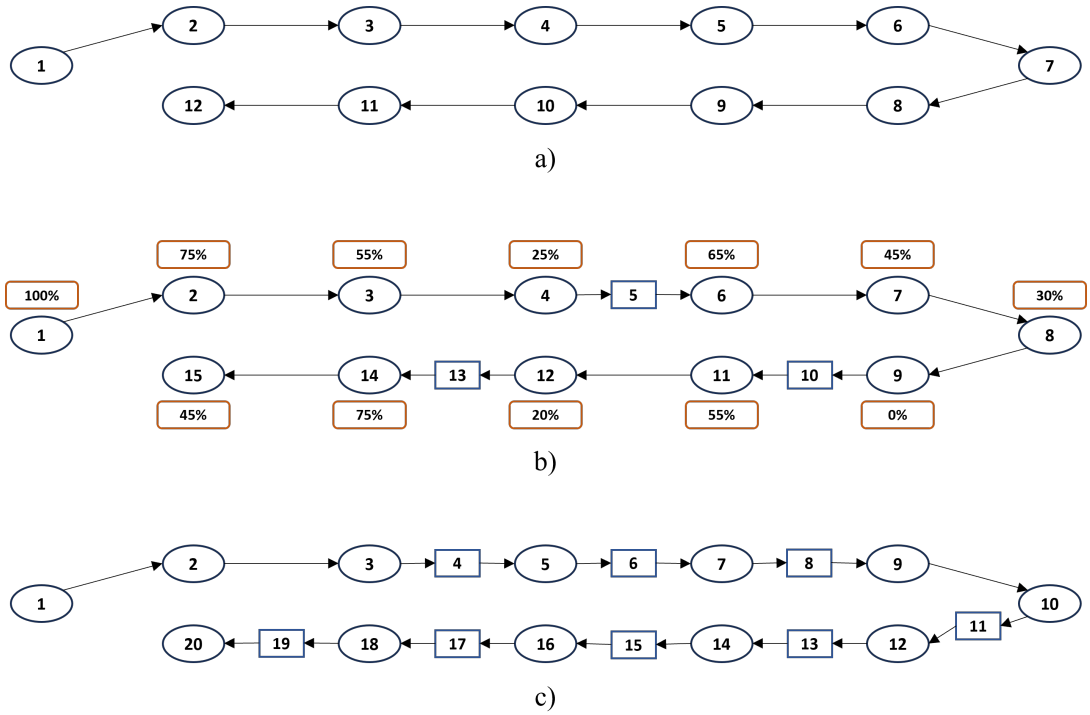

Figure 4.5: The illustration of Appointment Rendezvous Point Order Algorithm ((a) Initial tour, (b) The route with initial candidate rendezvous points, (c) The route with all candidate rendezvous points) (The percentages above nodes are the battery level of the drone at the service points)

the drone will not have enough energy to travel from 4 to 5. Therefore, our first candidate rendezvous point should be between points 4 and 5. Since the location of the point is not determined yet, we assume that this rendezvous point is at the same location as point 4. Similarly, when these steps continue, we observe that the drone's energy depletes at point 8. Therefore, our second candidate rendezvous point should be between points 8 and 9. Another rendezvous point is between points 10 and 11. Since we have named the points based on the drone's visiting order, the updated route becomes 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15. From these points, {1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 14, 15} are service points, and {5, 10, 13} are assigned as rendezvous points.

The identified rendezvous points may differ from the correct points that minimize energy consumption and enable the completion of the drone's operation, as they are determined under certain assumptions. Therefore, additional candidate rendezvous

Table 4.1: Energy consumptions between edges in the example

Edges	**Energy Consumptions (%)**
1 - 2	25
2 - 3	20
3 - 4	30
4 - 5	35
5 - 6	20
6 - 7	15
7 - 8	30
8 - 9	45
9 - 10	35
10 - 11	25
11 - 12	30

points need to be determined. For this purpose, new rendezvous points are assigned to the intervals before and/or after the initially determined rendezvous points. Different combinations of assignments can be tried, such as assigning 1 point before and 1 point after, or 2 points before, or only 2 points before. These alternatives are explored to find the most suitable alternative for solving our problem by considering computational time and evaluating the results of the problem. Among the alternatives, the most suitable option is assigning 2 candidate points before, 1 candidate point after. This alternative is used in this algorithm.

Let us explain the alternative of assigning 1 point before and 1 point after using the above-mentioned example. Here, "before" refers to the point before the service point visited before the rendezvous point. So, the interval before the rendezvous point located between the 3^{rd} and 4^{th} service points is between the 2^{nd} and 3^{rd} service points.

The initial candidate rendezvous points are indicated as bolts in the route 1(1) – 2(2) – 3(3) – 4(4) – **5** – 6(5) – 7(6) – 8(7) – 9(8) – **10** – 11(9) – 12(10) – **13** – 14(11) – 15(12). It can be seen in Figure 4.5b. The numbers given in parentheses represent the numbers assigned to the points in the initial route. When alternative candidate

rendezvous points are added before and after these points, the resulting route is as follows: 1(1) – 2(2) – 3(3) – **4** – 5(4) – **6** – 7(5) – **8** – 9(6) – 10(7) – **11** – 12(8) – **13** – 14(9) – **15** – 16(10) – **17** – 18(11) – **19** – 20(12). It is seen in Figure 4.5c.

The starting and ending points are not considered in *the Appointment Rendezvous Point Order Algorithm*. Therefore, since the exact distance between the last visited service point and the ending (or starting) point cannot be known, the drone may run out of charge while returning to the starting point. Hence, an additional candidate rendezvous point is assigned before the last visited service point. In this example, there is no need for such an assignment.

4.2.3.2 Best Charge (Rendezvous Point) Sequence and Locations Algorithm

This algorithm, which encompasses and integrates the other algorithms described above, can be referred to as the main algorithm of the study. First, *the Initial Route Construction Algorithm* determines the initial route for the drone's visit based on the given service points.

Taking this route as input, Algorithm 4 starts execution.

The energy consumption for the current route is calculated by the SOCP based mathematical model. Then we try to improve this route with *the 2-Opt Algorithm*. This algorithm produces alternative routes from the initial route. In this study, the algorithm is run both with and without the speed-up technique.

For each generated alternative route, the following steps are performed.

Firstly, *the Appointment Rendezvous Point Order Algorithm* is applied on the selected route to create a new route that includes all candidate rendezvous points.

Through elimination on the list containing all candidate rendezvous points, the best sequence and location of the rendezvous point that yields the best result (minimum total energy consumption) for that route are determined. The sub-algorithm performing this task operates as follows:
The algorithm is given a list of candidate rendezvous points, i.e., {3, 7, 9, 14}. Starting with eliminating only the first rendezvous point, i.e., {7, 9, 14}, the remaining

list is used to solve the problem using the SOCP-based mathematical model. Since the number of appointment points is high in the initial solutions, the operation may not be completed within the given time limit or the charge may not be sufficient for travels between two points on the routes created, which may lead to infeasible results. In these cases, in order to eliminate the infeasible, it is aimed to first reduce the penalty values and finally to find the most feasible. In order to eliminate initial infeasible solutions, penalty variables are introduced to the model by incorporating the constraints described in 4.47, 4.48, 4.49, and 4.50. In Section 4.2.4, the penalty approach is explain in detail.

$$s_{n+1} \leq S_{Total} + p^1 \tag{4.47}$$

$$s_i + t_i^{MC} \leq S_{Total} + p^1 \qquad i \in ML \tag{4.48}$$

$$B_i^d \geq B_{Min} + p_i^2 \qquad \forall i \in N \setminus \{0\} \tag{4.49}$$

$$B_i^d \leq B_{Max} + p_i^2 \qquad \forall i \in N \setminus \{0\} \tag{4.50}$$

The initial result generated through this solution is recorded as the Minimum Objective Value. Subsequently, only the second rendezvous point is removed from the list, i.e., {3, 9, 14}. The remaining list is again used to solve the route using the SOCP-based mathematical model. If the resulting objective value is smaller than the current Minimum Objective Value, it is assigned as the new Minimum Objective Value. Otherwise, the Minimum Objective Value remains the previous value. These steps continue for the lists created by sequentially eliminating each candidate rendezvous point. The goal of this sub-algorithm is to determine which candidate rendezvous point, when removed, provides a feasible solution or infeasible solution with the least penalty with the minimum total energy consumption. Thus, at the end of this loop, we determine the elimination that achieves the minimum total energy consumption among the candidate rendezvous points.

The remaining list re-enters the same loop. For example, removing the first candidate rendezvous point in the initial scenario minimizes the total energy consumption. The remaining list is {7, 9, 14}. This list is again subjected to the same loop to determine which remaining points should be eliminated to achieve a best solution. Suppose that the scenario that yields the desired result is the elimination of the rendezvous point specified as 9. The remaining candidate rendezvous point list is {7, 14}.

This sub algorithm continues by updating the Minimum Objective Value in each iteration. If removing any point from the current list of candidate rendezvous points does not yield a better result than the current Minimum Objective Value, the loop is completed. In other words, if no improvement in the Objective Value is achieved in the final iteration, the loop is terminated. This means that the loop is completed when no candidate rendezvous point can be removed from the list.

Ultimately, the remaining list represents the order of the final rendezvous points for the currently processed route among the 2-Opt alternatives. When the SOCP-based mathematical model is solved with the route containing these rendezvous points in the list, it provides the locations of the rendezvous points that yield the best objective function value.

These processes are performed for all alternative routes generated by *the 2-Opt Algorithm*. The route and rendezvous point order that yield the best result among these routes are selected as the new route, and the algorithm returns to the step referred to as "Algorithm 4 starts execution". This outermost loop continues in this manner.

This loop is completed when all alternatives generated by *the 2-Opt Algorithm* have been tested for the last determined route, and no better result is achieved. In other words, the algorithm is completed if there is no improvement.

Algorithm 4 Best Charge (Rendezvous Point) Sequence and Locations Algorithm

Input: Initial Route

Initial Route for Each Improvement = Initial Route

while *No Improvement* **do**

Execute *the 2-Opt Algorithm with Speed-Up or without Speed-Up for Initial Route for Each Improvement, and* ***obtain*** *All Alternative Routes*

for $i \in AllAlternativeRoutes$ **do**

Execute *the Appointment Rendezvous Point Order Algorithm, and* ***obtain*** *Drone's Route with All Candidate Rendezvous Points and Candidate Rendezvous Points Order*

c = length of Candidate Rendezvous Points Order (size of array)

Execute *Candidate Rendezvous Point Elimination Algorithm in Algorithm 5*

Initial Route for Each Improvement = Min Route with Rendezvous Points

Optional(It can be inactivated when it is not wanted to stop at the first improvement.)

if $i \neq 0$ *and the first improvement is achieved* **then**

break

end

end

end

Note That: *The last value of Min Objective Value is the best solution value, and the last Min Route with Rendezvous Points and their locations obtain by SOCP-based math model is the best integrated route and locations.*

Algorithm 5 Candidate Rendezvous Point Elimination Algorithm

Inputs = Drone's Route with All Candidate Rendezvous Points and Candidate Rendezvous Points Order Min Objective Value = BigM

Min Route with Rendezvous Points = []

for $j = 0 \rightarrow c$ **do**

  Eliminated Candidate = 0

  m = length of Candidate Rendezvous Points Order (size of array)

  if *j = 0* **then**

    Solve *SOCP-based Math Model with Drone's Route with All Candidate Rendezvous Points, and* ***obtain*** *Objective Value.*

    Min Route with Rendezvous Points = Drone's Route with All Candidate Rendezvous Points

  else

    for $k = 1 \rightarrow m$ **do**

      Remove *Candidate Rendezvous Points Order[k] from Drone's Route with All Candidate Rendezvous Points*

      Solve *SOCP-based Math Model with Current Drone's Route, and* ***obtain*** *Objective Value.*

      if *Objective Value < Min Objective Value* **then**

        Min Objective Value = Objective Value

        Eliminated Candidate = k

        Min Route with Rendezvous Points = Current Drone's Route

      end

    end

    Remove *Candidate Rendezvous Points Order[Eliminated Candidate] from Candidate Rendezvous Points Order*

  end

end

4.2.4 Penalty Approach

Since we solve our problem using improvement heuristic methods, there is a high probability of infeasible outcomes in the initial steps. Therefore, relaxing some constraints in the mathematical model for initial scenarios helps to eliminate infeasible

situations and prevents the problem solution from getting stuck at the beginning, allowing us to reach feasible solutions in the subsequent steps. These penalties are not used throughout the entire solution process in the mathematical model. As the process progresses, the penalty values in the results gradually decrease. When both penalties reach zero, indicating a feasible solution, they are removed from the model, and the subsequent solutions are searched among penalty-free feasible solutions to find the best result.

We explain penalty approach for model that minimize the total energy consumption in Sections 4.2.4.1 and 4.2.4.2. The explanation of penalty approach for model that minimize the total system time is given in Section 4.2.4.3.

4.2.4.1 Relaxing Time Limit of the System Time

One of the hardest constraints that restricts our solution is the total system time (S^{Total}) constraints, as seen in (4.20) and (4.29) in Section 4.1.1. In these constraints, s_{n+1} represents the time the operation is completed. S^{Total} represents the total service time that the system needs to complete.

The solution approach involves finding the final rendezvous points by eliminating alternatives from a list that contains multiple candidates for rendezvous points. Initially, the total system time can be significantly high because there are many rendezvous points on the route or the initial route may be bad. This total time easily exceeds the feasible limit which is S^{Total}. Consequently, the solution becomes infeasible. To address this issue and make the model more flexible, we updated the constraints as shown in (4.51) and (4.52), and modified the objective function presented in (4.53).

$$s_{n+1} \leq S^{Total} + p^1 \tag{4.51}$$

$$s_i + t_i^{MC} \leq S^{Total} + p^1 \qquad i \in ML \tag{4.52}$$

$$\text{min} \qquad \sum_{i=0}^{n} E_i + (B_0^{MC} - B_{n+1}^{MC}) + BigM \times p^1 \tag{4.53}$$

4.2.4.2 Relaxing Battery Level of the Drone

In addition to the total system time, there are some other critical constraints of the problem, which determine the limits of the drone's battery level. These constraints are represented in (4.54) and (4.55). According to these constraints, the battery levels of the drone can only be the range of B^{Min} to B^{Max} during its operation. In other words, if there is no flexibility in these constraints, its battery level must be within this range when the drone departs from the service point.

$$B_i^d \geq B_{Min} \qquad \forall i \in N \setminus \{0\} \quad (4.54)$$

$$B_i^d \leq B_{Max} \qquad \forall i \in N \setminus \{0\} \quad (4.55)$$

In the solution approach, bad initial route or removing certain candidate rendezvous points between some service points lead to situations where the drone does not have enough energy to cover that distance. As a result, the solution becomes infeasible. To overcome this issue and make the model more flexible in the early stages of the solution process, we have updated the constraints as shown in (4.56) and (4.57), and the objective function as shown in (4.58).

$$B_i^d \geq B_{Min} - p_i^2 \qquad \forall i \in N \setminus \{0\} \quad (4.56)$$

$$B_i^d \leq B_{Max} + p_i^2 \qquad \forall i \in N \setminus \{0\} \quad (4.57)$$

$$\min \qquad \sum_{i=0}^{n} E_i + (B_0^{MC} - B_{n+1}^{MC}) + BigM \times p^1 + BigM \times \sum_{i=1}^{n+1} p_i^2 \qquad (4.58)$$

4.2.4.3 Penalty Approach in Model That Minimizes the Total System Time (S^{Total})

In the model that minimize the total energy consumption, we consider the total system time as a parameter. The objective of that model is to minimize the total energy consumption, and we aim to obtain the proper results by appropriately determining

the S^{Total} value as parameter. Our second aim in this study is to minimize the completion time of the operation. By converting S^{Total} into a variable and making the necessary adjustments in the model, we can achieve a model that minimizes the total system time.

One of the primary changes that need to be made in the model is to revise the parameter S^{Total} as a decision variable, as indicated below.

S^{Total}: Total time spent in the system (the completion time of the operation)

In (4.59) and (4.60), it is not necessary to make any changes in the constraint in which this variable is contained. Since S^{Total} is now a decision variable, there is no need for any penalty to relax the constraint. However, p^2 is still present in the model.

$$s_{n+1} \leq S^{Total} \tag{4.59}$$

$$s_i + t_i^{MC} \leq S^{Total} \qquad i \in ML \tag{4.60}$$

The main change in the model is indeed in the objective function. As seen in (4.61), S^{Total} is added to the objective function, and total energy consumption is removed. This allows for the minimization of the total system time.

$$\min \qquad S_{Total} + BigM \times \sum_{i=1}^{n+1} p_i^2 \tag{4.61}$$

Indeed, with the model we have established, we can both define S^{Total} as a parameter and receive it as an external input, and also create a system design that minimizes the total system time with minor modifications.

CHAPTER 5

COMPUTATIONAL STUDIES

In this chapter, we discuss illustrative examples, preliminary experiments, and computational experiments, along with their results. All of these studies were conducted on random problem instances that we generate, because there is no benchmark data set available that is suitable for our problem. The instances were created by generating random points in Excel based on the number of service points to be considered. Different instances were generated, and more detailed information about the instances will be provided in the following subsections. For the experiments, we used a computer equipped with an Intel Xeon E2246G processor running at 3.6 GHz and 16 GB of RAM. We chose to utilize Python version 3.8.10, IBM ILOG CPLEX Optimization Studio 12.10.0, and Microsoft Office Excel, with seamless integration between them. We developed our algorithms and main code in Python, and executed our SOCP mathematical model using the IBM ILOG CPLEX infrastructure. The instances were created in Microsoft Excel, and inputs to the algorithms and model were performed via Microsoft Excel. Furthermore, we recorded our results in Microsoft Excel, conducting our analyses using these Excel files. The following paragraph briefly outlines the computational processes carried out in each section.

Section 5.1 presents an illustrative example that was created to provide a simple explanation of the outputs generated by the matheuristic. Section 5.2 includes some preliminary experiments. The first experiment explores the relationship between the total system time (S^{Total}) and total energy consumption. The second experiment explains the impact of the energy consumption parameters of the drone and the MC on the route. Finally, Section 5.3 contains the computational experiments and their results.

5.1 Illustrative Example

In order to enhance the comprehensibility of the study, we provide an illustrative example with 30 service points. The distribution of service points, generated randomly in Microsoft Excel, is visualized in Figure 5.1, representing a two-dimensional plane with dimensions of 100 m x 100 m.

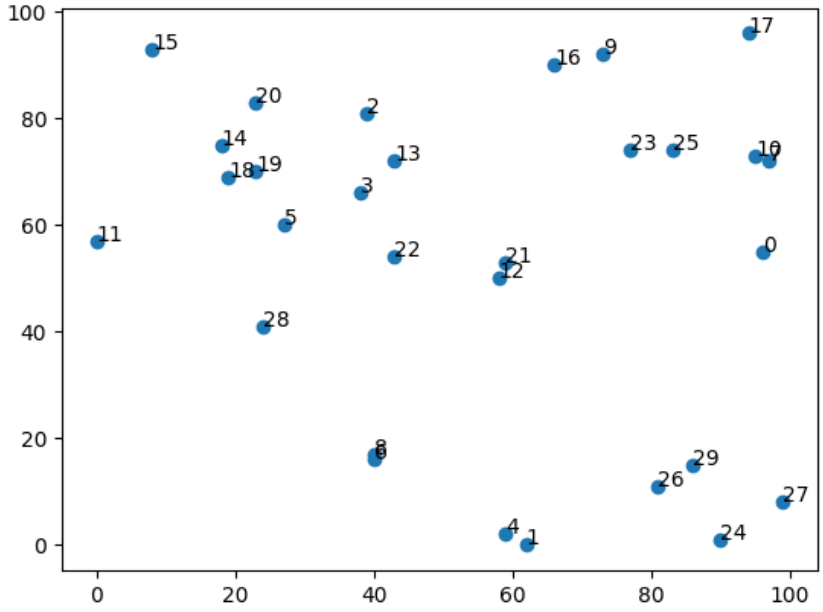

Figure 5.1: Service points of the illustrative example

It is expected that the drone will visit these points within a specified Total System Time (S^{Total}). Furthermore, during this visiting process, the drone is expected to obtain the required energy from the MC located on the same plane. Our objective is to find an integrated route that minimizes the total energy consumption of both the MC and the drone.

We executed our matheuristic based on the parameters specified in Tables 5.1 and 5.2.

Before determining the appointment points for the drone and the MC, it is necessary to find an initial route for the drone. Based on the "*Initial Route Construction Algorithm*

Table 5.1: Parameters of the illustrative examples

S^{Total}	B^{Max}	σ	T^{Max}	V^{MC}_{Max}	V^{MC}_{Min}	B^{Min}	V^{d}_{Max}	V^{d}_{Min}	D
370	100	0.005	100	3	0.001	0	3	0.01	1

Table 5.2: Drone's and MC's energy consumption parameters of the illustrative examples

α^{d}	β^{d}	γ^{d}	α^{MC}	β^{MC}	γ^{MC}
0.15	0.25	0.4	0.15	0.25	0.4

based on the Nearest Neighbor Algorithm", an initial route was constructed using the service points shown in Figure 5.2. As depicted in this figure, the route starts from the initial point marked by a circle and continues until it reaches the endpoint, which is also the same point as the initial point within the circle.

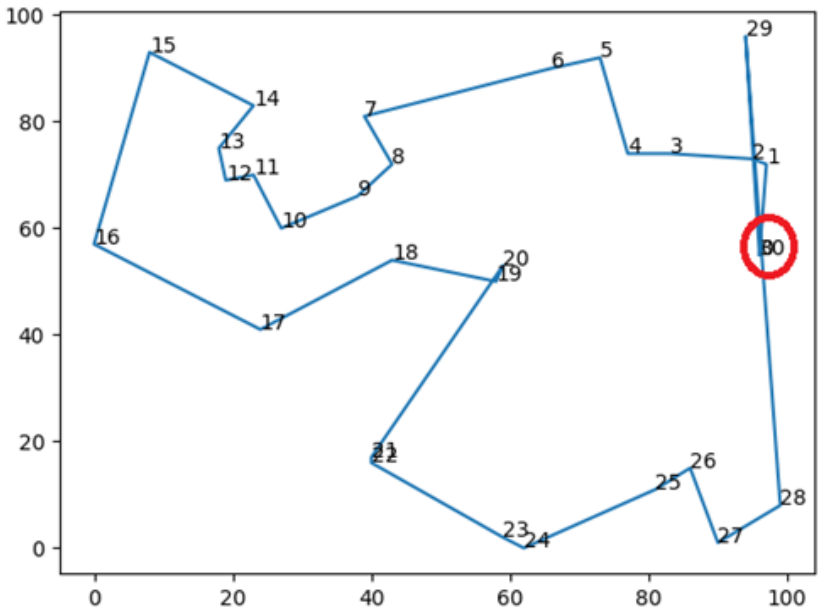

Figure 5.2: Illustration of the initial route of the drone

Using the initial route, the candidate rendezvous points, where the drone's energy

needs can be fulfilled, are assigned through the "*Rendezvous (Charging) Point Determination Algorithms" and "Appointment Rendezvous Point Order Algorithm*". Throughout the entire process, the first integrated route of the drone and the MC is established based on the candidate rendezvous points initially determined. Figure 5.3 presents the resulting integrated route.

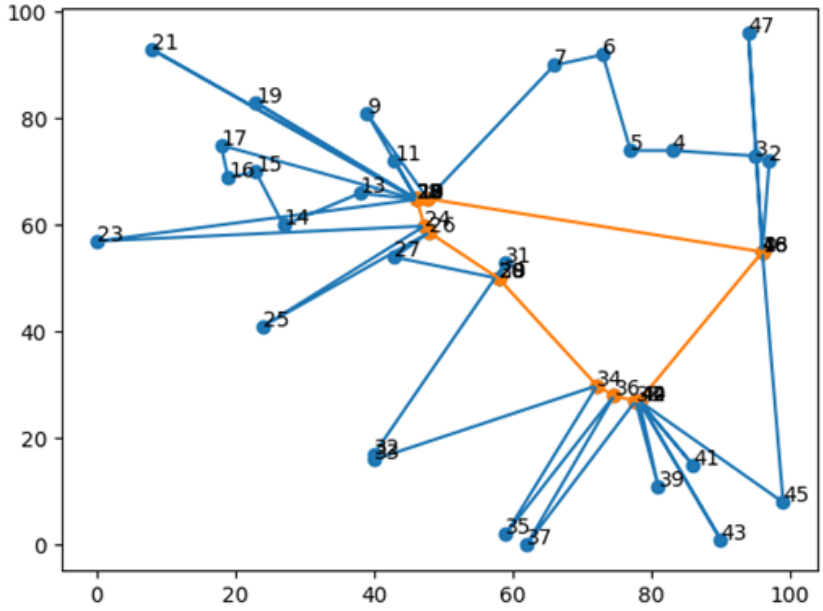

Figure 5.3: Illustration of the first integrated route of the drone and MC before improvements. In this illustration, the route with blue edges belongs to drone, the route with orange edge belongs to MC. Intersections of these two routes are the rendezvous points.

In Figure 5.3, all candidate rendezvous points are included. As a result, the drone charges at all of these points based on the construction of the model. Consequently, there are situations where the system time exceeds the Total System Time constraint.

Using the initial integrated route shown in Figure 5.3, we performed improvements through the "*Best Charge (Rendezvous Point) Sequence and Locations Algorithm*" and "*2-Opt Algorithm*". After implementing all the improvements, the best result obtained is presented in Figure 5.4.

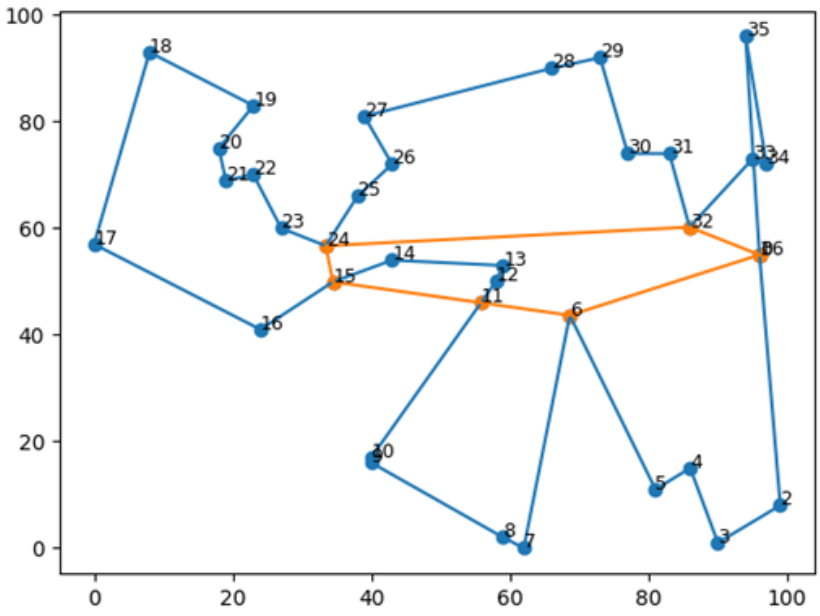

Figure 5.4: Final integrated route of the drone and MC in the illustrative example

During the process of obtaining the final route, we observed situations where the battery level of the drone became insufficient between two service points or when travelling from the service point to a rendezvous point, especially when eliminating some candidate rendezvous points. This resulted in initially finding infeasible solutions. We have overcome this situation with the penalty approach.

5.1.1 Computational Solutions of the Illustrative Example

In this section, we will present the numerical results of the problem using tables and graphs to explain the results which is investigated from the illustrations in the previous section. During the solution of this problem, the 2-Opt Algorithm is implemented with Speed Up Method. The algorithm considers alternative routes and selects the route where the first improvement is achieved. Without trying other routes, the algorithm proceeds to the next iteration. The locations of the points in the drone's route, indicated by blue edges in Figure 5.4, and the speed of the drone between these points are provided in Tables 5.3 and 5.4. Similarly, the locations of the points in the MC's

route, marked in orange, and the speed of the MC between these points are given in Table 5.5.

According to the optimal speed formula in 3.5 in Chapter 4, specifically in Section 4.2.3.1, the optimal speed for both the drone and the MC is determined to be 1.633 m/s when considering their energy consumption parameters.

The points highlighted in yellow in Table 5.3 and Table 5.4 represent the rendezvous points. Upon examining the table, it is shown that the drone is not moving at the optimal speed. The reason is that the drone needs to complete the operation within the given time constraint imposed by the Total System Time (S^{Total} = 370). Thus, the drone's speed must be adjusted to ensure that it completes the operation within the specified time-frame. It is observed that drones travel at different speeds between various rendezvous points. The distance covered by the drone varies between these rendezvous points. The energy consumption of the drone also varies depending on the distance traveled. The drone's energy consumption increases as the distance increases due to the longer travel distance. The drone reduces its speed to ensure the battery charge lasts until the next rendezvous point. Specifically, between two rendezvous points where the distance is longer, it travels at a slower speed (closer to the optimal speed which is 1.633 m/s) to prevent the battery from depleting. On the other hand, in intervals where the distance is shorter and the battery charge is sufficient, the drone moves faster to complete its operation within the Total System Time.

The MC, operating only between rendezvous points, covers a shorter distance, resulting in a completion time for its operation that remains lower than the Total System Time (S^{Total}). Consequently, the Total System Time constraint does not restrict the operation of the MC. As a result, the MC can move at its optimal speed to minimize energy consumption. The route of the MC and its speeds are given in Table 5.5.

Table 5.3: Route of the drone and speed of the drone on this route

Path of the Drone			
Nodes	**X Coordinate**	**Y Coordinate**	**Drone's Speed (m/s)**
0	96.000	55.000	-
1	96.000	55.000	1.748
2	99.000	8.000	1.748
3	90.000	1.000	1.748
4	86.000	15.000	1.748
5	81.000	11.000	1.748
6	68.630	43.665	1.748
7	62.000	0.000	1.748
8	59.000	2.000	1.748
9	40.000	16.000	1.748
10	40.000	17.000	1.748
11	55.825	46.013	1.748
12	58.000	50.000	1.748
13	59.000	53.000	1.748
14	43.000	54.000	1.748
15	34.468	49.928	1.656
16	24.000	41.000	1.656
17	0.000	57.000	1.656
18	8.000	93.000	1.656
19	23.000	83.000	1.656
20	18.000	75.000	1.656
21	19.000	69.000	1.656
22	23.000	70.000	1.656
23	27.000	60.000	1.656
24	33.424	56.692	1.748
25	38.000	66.000	1.748
26	43.000	72.000	1.748

Table 5.4: Continuation of Table 5.3 (Route of the drone and speed of the drone on this route)

Path of the Drone			
Nodes	**X Coordinate**	**Y Coordinate**	**Drone's Speed (m/s)**
27	39.000	81.000	1.748
28	66.000	90.000	1.748
29	73.000	92.000	1.748
30	77.000	74.000	1.748
31	83.000	74.000	1.748
32	85.913	60.200	1.748
33	95.000	73.000	1.748
34	97.000	72.000	1.748
35	94.000	96.000	1.748
36	96.000	55.000	-

Table 5.5: The route of the MC and speed of the MC on this route

Path of the MC			
Nodes	**X Coordinate**	**Y Coordinate**	**MC's Speed (m/s)**
0	96.000	55.000	1.633
6	68.630	43.665	1.633
11	55.825	46.013	1.633
15	34.468	49.928	1.633
24	33.424	56.692	1.633
32	85.913	60.200	1.633
36	96.000	55.000	-

Table 5.6: Best objective function value and number of iterations of the problem

Best OFV (Joule)	**# of Iterations**
524.934	4

In Table 5.6, the first column represents the objective function value of the problem. Each step in which the route is improved with the help of the 2-Opt algorithm is called iteration. The last column specifies the iteration at which the most recent improvement was obtained. In this table, it is seen that the number of iterations is 4. This means that there is no improvement in the 5^{th} iteration, where the last improvement was in the 4^{th} iteration.

Table 5.7: Energy consumptions in the problem

Drone's Energy Consumption (Joule)	**MC's Energy Consumption (Joule)**	**Total Energy Consumption (Joule)**
424.934	100.000	524.934

Table 5.7 provides the energy consumption of both the drone and the MC in the best result obtained by the matheuristic. The last column represents the total energy consumed by the two vehicles.

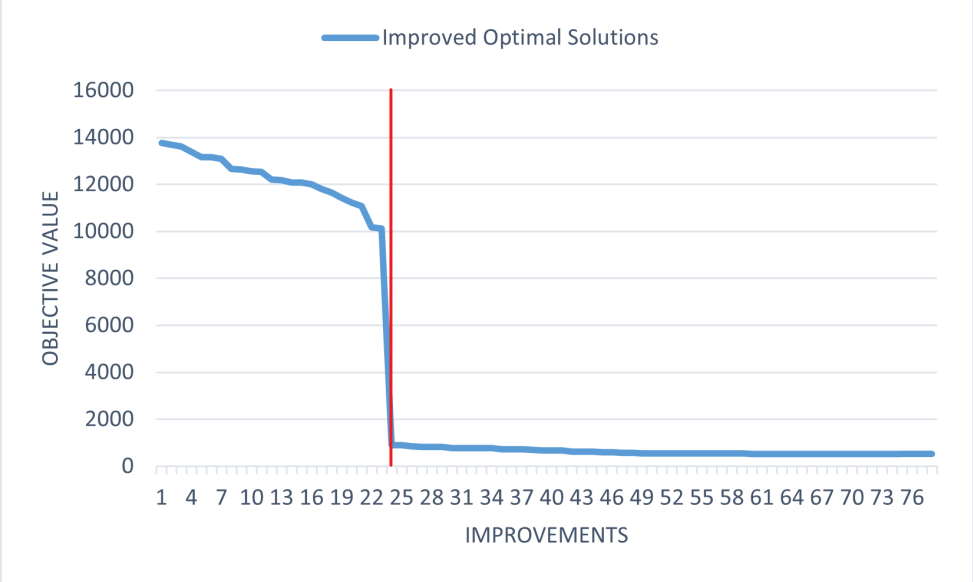

Figure 5.5: All improvements from the first solution with penalty to best solution

Figure 5.5 provides us with crucial information. In this figure, improvement axis indicate all the improvements obtained from the first solution to the final solution. Our approach aims to eliminate penalties first and then improve our solution. As seen in the graph, the left side of the red line represents the presence of penalties.

Consequently, the optimal solution values are quite high. We can observe the results obtained after eliminating the penalty values on the left side of the red line. The initial 24 units represent the stage where we attempt to reduce and annihilate the penalty values, while the subsequent stage shows the improvement of penalty-free results. Foremost among these is the improvement achieved from the initial feasible solution with penalties to the final solution without penalties. The initial feasible solution without penalties provided by the matheuristic has objective function value 894.191 joule. On the other hand, the final solution with the route depicted in Figure 5.4 has an objective function value of 524.934 joule. This demonstrates that there has been a 41.3% improvement from the first penalty-free feasible solution solution to the final solution for this problem. Taking a general look at the graph, it can be observed that in the solution process where a total of 1394 feasible solutions is obtained, in 78 cases, an improvement is achieved.

5.2 Preliminary Experiments

In this section, two preliminary experiments and their results are mentioned. First, we describe the experiment in which we examined how the change in S^{Total} affects energy consumption. Afterwards, we explain the experiment in which we will examine the effects of the energy consumption parameters of vehicles on the integrated route. With these experiments, it is aimed to examine the effects of some important parameters on the results.

5.2.1 Total System Time (S^{Total}) vs Energy Consumptions

Total System Time is one of the most critical constraints in the problem, as it significantly affects the routes of both the drone and the MC. If the required time for the drone to meet the MC for charging and fulfill its service visits is not provided, it is not possible to reach a feasible solution for the operation. Therefore, an appropriate S^{Total} value must be selected. Among the feasible S^{Total} values, up to a certain threshold, the total energy consumption of the drone and the MC is high. This is because both vehicles are unable to travel at an optimal speed that would minimize their energy consumption.

In this section, we will explain how the energy consumption of the drone and the MC varies with different S^{Total} values. Before calculating the energy consumption, it is necessary to determine between which S^{Total} values these operations should be performed. To determine suitable S^{Total} values, we utilize the mathematical model with penalty approach described in Section 4.2.4.3.

To observe the changes in energy consumption of the drone and the MC, we need to find both the minimum and maximum S^{Total} values. To achieve this, we run the model directly to find the minimum S^{Total}. It is known that the model does not consider the energy consumption, but also consider the total system time. By using this model without any charge, we obtain minimum S^{Total}.

To calculate the maximum S^{Total} value, we consider to minimize only total energy consumption. To do this, we need to use the SOCP-based mathematical model with

penalty approach that minimize total energy consumption. In order to obtain the correct maximum S^{Total} value, it is necessary to make some changes in the model. Since we primarily aim to find the appropriate S^{Total}, it is taken as a variable, not a parameter, in this model. In addition, we need to add S^{Total} to the objective function by multiplying it by a small coefficient, so that the model focuses on minimizing energy consumption without considering S^{Total}; at the same time, S^{Total} should still be included in the objective function to avoid it taking an unnecessarily large value. In this experiment, to calculate the maximum S^{Total} value, we set the coefficient in the objective function to 0.0000001 as shown in (5.1).

$$\text{min} \qquad \sum_{i=0}^{n} E_i + (B_0^{MC} - B_{n+1}^{MC}) + 0.0000001 \times S_{Total} + BigM \times \sum_{i=0}^{n+1} p_i^2 \quad (5.1)$$

We conducted this experiment with two different sets of energy consumption function parameters. In the first case, the drone and MC have equal parameters, as shown in Table 5.8. In the second case, the MC's energy consumption parameters are twice that of the drone's, as indicated in Table 5.9.

Table 5.8: The drone's and MC's energy consumption parameters in the first case

α	β	γ	α^{MC}	β^{MC}	γ^{MC}
0.15	0.25	0.40	0.15	0.25	0.40

Table 5.9: The drone's and MC's energy consumption parameters in the second case

α	β	γ	α^{MC}	β^{MC}	γ^{MC}
0.15	0.25	0.40	0.30	0.50	0.80

The other parameters that we will use when running the matheuristic are provided in Table 5.10.

The minimum (Min) and maximum (Max) S^{Total} values obtained for the first case and the second case are provided in Table 5.11.

Table 5.10: Parameters of the S^{Total} vs energy consumptions experiments

B^{Max}	σ	T^{Max}	V_{Max}^{MC}	V_{Min}^{MC}	B^{Min}	V_{Max}^{d}	V_{Min}^{d}	D
100	0.005	3	3	0.001	0	3	0.01	1

Table 5.11: Min and max S^{Total} values for the first and second cases

	The First Case	**The Second Case**
Min S^{Total} (Min) (sec)	164	200
Max S^{Total} (Max) (sec)	294	331

We divided the range between the Min and Max S^{Total} values into 15 equal intervals in order to observe how energy consumption changes with respect to S^{Total}. We input each S^{Total} value into the mathematical model and minimize only the total energy consumption. In total, we calculated 16 energy consumption values for each case, considering both the minimum and maximum values. The results for the first case are presented in Figure 5.6, and the results for the second case are presented in Figure 5.7.

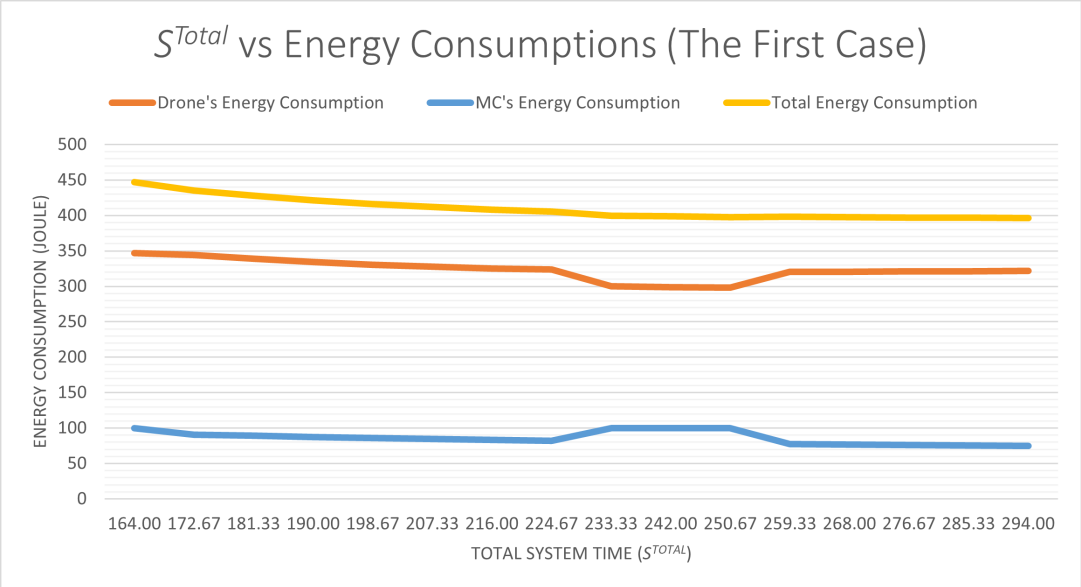

Figure 5.6: S^{Total} vs energy consumption for the first case

As seen in Figure 5.6, the energy consumption of both the Drone and the MC is higher where the S^{Total} value is smaller. As S^{Total} value increases, the total energy consumption can reach a steady state after a certain point. This is because when the total system time restricts the system no longer, the drone and the MC can perform their operations at optimal speeds to minimize their energy consumption. It can be observed that there are specific S^{Total} values where the energy consumption of the drone can be minimized. However, at these values, the energy consumption of the MC tends to increase.

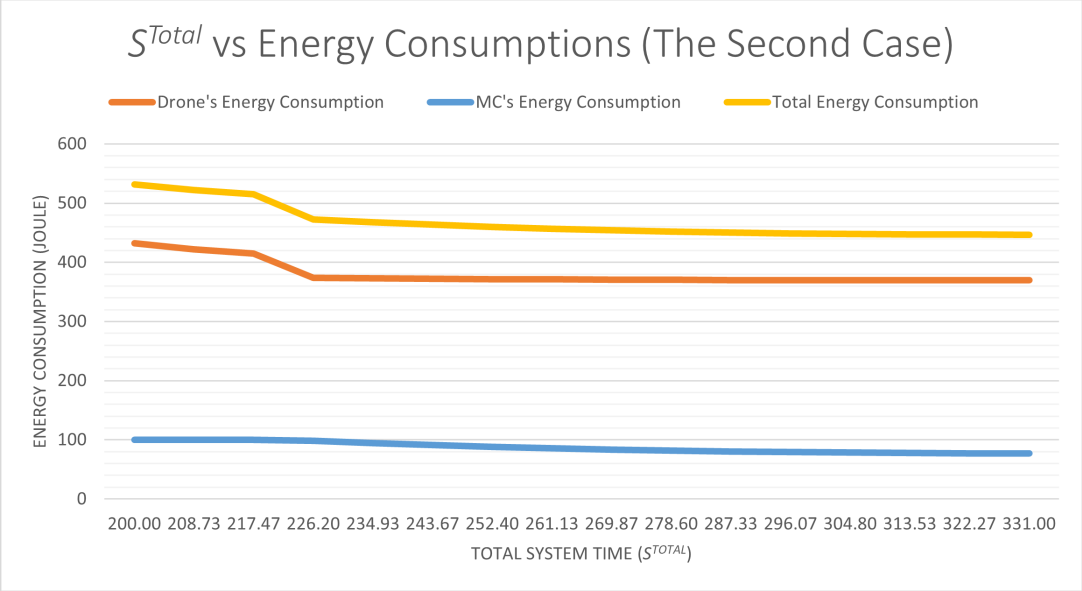

Figure 5.7: S^{Total} vs energy consumption for the second case

Figure 5.7 indeed supports the observation that as S^{Total} increases, the energy consumption decreases. It is clearly seen that the energy consumptions of the drone and MC also decrease continuously as S^{Total} increases. Therefore, the total energy consumption also decreases continuously.

As a result, in this experiment performed with different energy consumption function parameters, it is seen that as the S^{Total} value increases, the total energy consumption decreases.

5.2.2 Change in Integrated Route According to Drone's and MC's Energy Consumption Function Parameters

Some of the most important parameters that determine the integrated route of the drone and mobile charging station (MC) are the energy consumption function parameters. The location of rendezvous points is significantly influenced by these parameters. If the energy consumption function parameters of the MC are larger than drone's ones, it can be said that the location of the rendezvous points will be confined to a narrower range. In other words, the location of the rendezvous points will be designed to minimize the movement of the MC. As the value of the parameters of the MC increases, the location of the rendezvous points will cause the drone to deviate further from its own route. To observe this phenomenon, we conduct experiments using different parameter sets on the same instance containing identical service points. We attempt to interpret this phenomenon based on the resulting graphs. Table 5.12 provides the parameter sets is used in the experiment.

Table 5.12: Parameter sets to be used in the experiment

Sets	α^d	β^d	γ^d	α^{MC}	β^{MC}	γ^{MC}	**MC Parameters / Drone Parameters Ratio**
1	0.1500	0.2500	0.4000	0.1500	0.2500	0.4000	1.00
2	0.1500	0.2500	0.4000	0.1875	0.3125	0.5000	1.25
3	0.1500	0.2500	0.4000	0.2250	0.3750	0.6000	1.50
4	0.1500	0.2500	0.4000	0.2625	0.4375	0.7000	1.75
5	0.1500	0.2500	0.4000	0.3000	0.5000	0.8000	2.00
6	0.1500	0.2500	0.4000	0.3375	0.5625	0.9000	2.25
7	0.1500	0.2500	0.4000	0.3750	0.6250	1.0000	2.50
8	0.1500	0.2500	0.4000	0.4125	0.6875	1.1000	2.75
9	0.1500	0.2500	0.4000	0.4500	0.7500	1.2000	3.00
10	0.1500	0.2500	0.4000	0.4875	0.8125	1.3000	3.25
11	0.1500	0.2500	0.4000	0.5250	0.8750	1.4000	3.50
12	0.1500	0.2500	0.4000	0.7500	1.2500	2.0000	5.00

As shown in Table 5.12, in the initially created parameter set (Parameter Set 1), the energy consumption parameters of the drone and the MC are equal. Subsequently, we generated the sets by gradually increasing the MC's parameter values relative to the drone's parameters by a certain factor. In Parameter Set 12, we set the MC's parameters to be relatively higher compared to the other sets, specifically five times that of the drone's parameters. The aim is to observe the changes based on these parameter sets. The routes obtained by the run with each of these parameter sets are provided in Figure 5.8.

The graphs presented in Figure 5.8 are arranged based on the increasing order of the mobile charging station's (MC) energy consumption function parameters. In the illustration in Figure 5.8a, it can be observed that the route of the MC highlighted in orange is longer compared to the route of the MC in Figure 5.9m. As the energy consumption parameters of the MC increase, its route becomes narrower and eventually deviates from the route where the drone visits the service points.

An alternative interpretation can be made based on this experiment. Figure 5.9 shows the route generated according to a parameter set where the drone's parameters are higher than MC's ones. As the energy consumption parameters of the drone increase, the rendezvous points will be located in a way that minimizes the distance covered by the drone. In other words, the route of the MC will expand, and the rendezvous points will approach the route where the drone visits the service points.

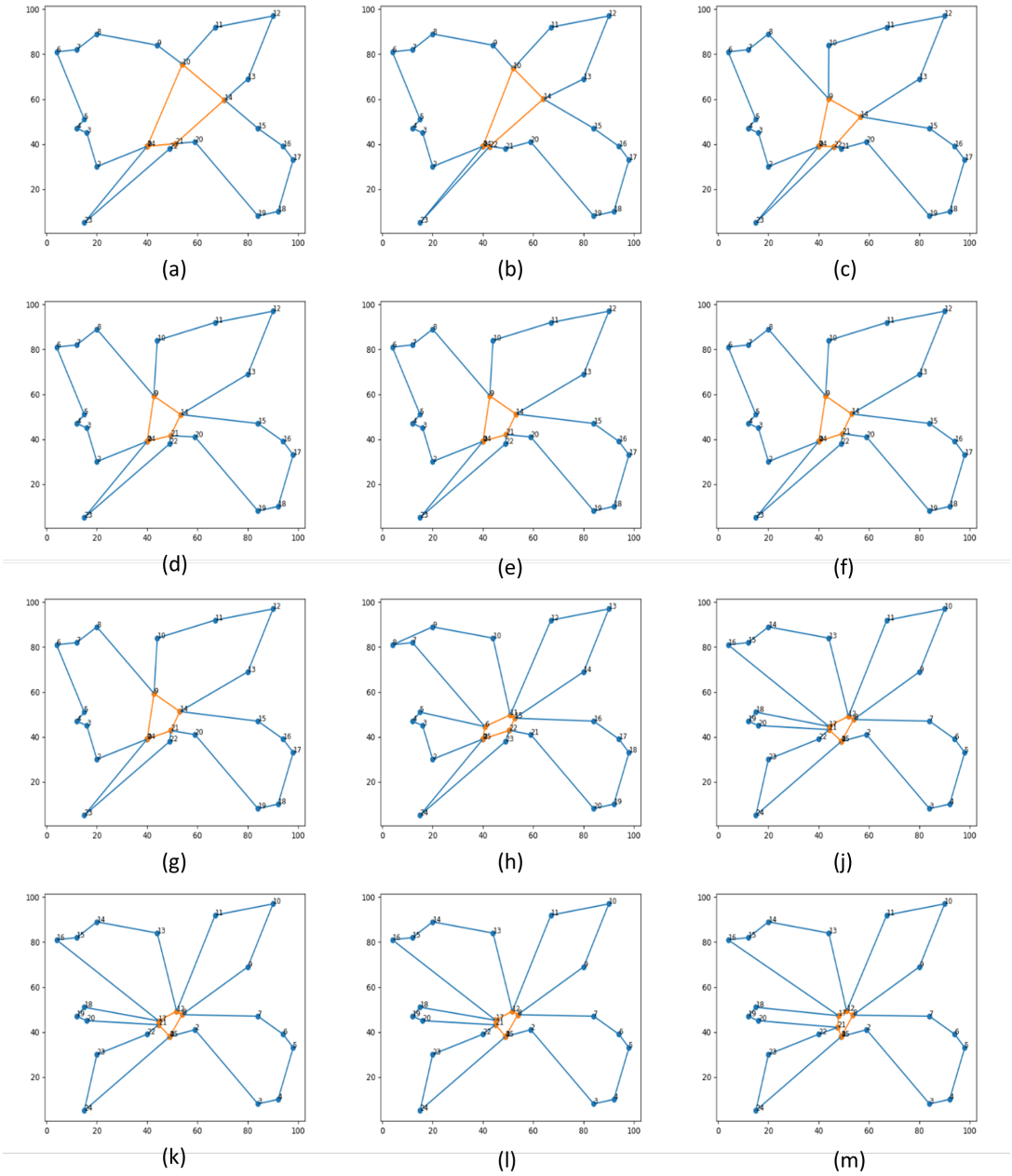

Figure 5.8: Routes According to Parameters Set in Figure 5.2.2.1 ((a) Route with parameter set 1, (b) Route with parameter set 2, (c) Route with parameter set 3, (d) Route with parameter set 4, (e) Route with parameter set 5, (f) Route with parameter set 6, (g) Route with parameter set 7, (h) Route with parameter set 8, (j) Route with parameter set 9, (k) Route with parameter set 10, (l) Route with parameter set 11, (m) Route with parameter set 12)

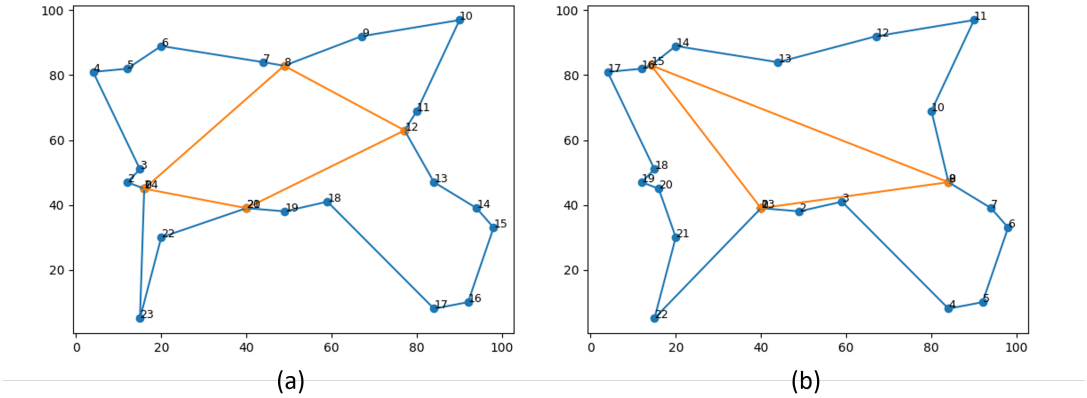

Figure 5.9: The integrated route with low MC energy consumption parameters (a) α^d = 0.15, β^d = 0.25, γ^d = 0.4, α^{MC} = 0.075, β^{MC} = 0.125 and γ^{MC} = 0.2, b) α^d = 0.15, β^d = 0.25, γ^d = 0.4, α^{MC} = 0.03, β^{MC} = 0.05 and γ^{MC} = 0.08)

5.3 Detailed Computational Experiments

5.3.1 Experiment Design

In order to analyze the results of our studies, we need to organize an experimental setup. This experimental setup aims to enable the following analyses:

1. Analysis of the results obtained from the setup operated according to different working principles implemented in our algorithms.

2. Analysis of the results obtained from the setup operated with different energy consumption parameters.

3. Analysis of the results obtained from the setup operated under different S^{Total} values.

4. Analysis of the results obtained from the setup operated with different numbers of service points (instance size).

To perform all these analyses, an experimental setup is designed, and its factors and their levels are described in Table 5.13 and Table 5.14.

Table 5.13: Factors of the experiments and their levels

Levels	**A) 2 Opt - Speed up**	**B) Stop after 1^{st} Improvement in 2 Opt Alternatives**	**C) Energy Consumption Function Parameters**	**D) S^{Total}**	**E) Instance Size**
Lv1	With Speed Up	Stop	Parameter Set 1	Value 1	30
Lv2	Without Speed Up	Do Not Stop	Parameter Set 2	Value 2	45
Lv3	-	-	-	-	60

Table 5.14: Sets of the energy consumption function parameters of both vehicles (F C)

C) Energy Consumption Function Parameters of Both Vehicles						
Levels	α^d	β^d	γ^d	α^{MC}	β^{MC}	γ^{MC}
Parameter Set 1 (Lv1)	0.15	0.25	0.40	0.15	0.25	0.40
Parameter Set 2 (Lv2)	0.15	0.25	0.40	0.30	0.50	0.80

Table 5.13 provides the factors and their levels for the experiment. In the first factor (A), the matheuristic model incorporates the 2-Opt Algorithm with two levels: one where the Speed Up method is applied and one where it is not. Without the Speed Up method, the runtime of the model significantly increases. Therefore, in experiments involving this level, the model's runtime is limited to 2 hours.

In the second factor (B), which focuses on testing alternative routes obtained using the 2-Opt Algorithm, there are two levels. In the first one, we move to the first route the improves the current one. In the second one, we move to the route that is the best out of all possible 2-Opt moves.

The third factor (C) offers two levels for the energy consumption function parameter set used in the model. The values for these two sets are provided in Table 5.14.

In the fourth factor (D), the Total System Time (S^{Total}) value for the model is specified. Two levels are presented, with different values depending on the experimental setup. Different parameter sets and instance sizes lead to variations in the system's service time. Therefore, there are six different S^{Total} values for the six experiment setups created based on the combinations of Factors C and E. The calculation of these values is explained in Section 5.3.2, and the different values of S^{Total} for each setup is provided in Table 5.18.

Finally, the fifth factor (E) offers instance size levels for running the model. These levels allow for observations and comments on the model's solution time and improvement amount for different instance sizes.

The experimental setups created based on these five factors and their levels are presented in Table 5.15, Table 5.16, and Table 5.17. A total of 48 experiments were designed by combining the levels available in each factor. To better understand, let's take Experiment 13 as an example. In this experiment, Level 1 from Factor A (2-Opt Algorithm with Speed Up method) and Level 2 from Factor B (testing all alternative routes without stopping at the first improvement) were selected. Level 2 (Parameter Set 2) for Factor C, Level 1 (Value 1 of S^{Total}) for Factor D, and Level 1 (instance size 30) for Factor E were chosen. Therefore, the experiment was designed using the following combination of factors: 2-Opt with Speed Up, Do Not Stop After 1^{st} Improvement, Parameter Set 2, Value 1 of S^{Total}, and Instance size 30.

Table 5.15: Experiments designed for instance size 30 and 45

EXPERIMENTS	**FACTORS**				
1	A - Lv1	B - Lv1	C - Lv1	D - Lv1	E - Lv1
2	A - Lv1	B - Lv1	C - Lv1	D - Lv1	E - Lv2
3	A - Lv1	B - Lv1	C - Lv1	D - Lv2	E - Lv1
4	A - Lv1	B - Lv1	C - Lv1	D - Lv2	E - Lv2
5	A - Lv1	B - Lv1	C - Lv2	D - Lv1	E - Lv1
6	A - Lv1	B - Lv1	C - Lv2	D - Lv1	E - Lv2
7	A - Lv1	B - Lv1	C - Lv2	D - Lv2	E - Lv1
8	A - Lv1	B - Lv1	C - Lv2	D - Lv2	E - Lv2
9	A - Lv1	B - Lv2	C - Lv1	D - Lv1	E - Lv1
10	A - Lv1	B - Lv2	C - Lv1	D - Lv1	E - Lv2
11	A - Lv1	B - Lv2	C - Lv1	D - Lv2	E - Lv1
12	A - Lv1	B - Lv2	C - Lv1	D - Lv2	E - Lv2
13	A - Lv1	B - Lv2	C - Lv2	D - Lv1	E - Lv1
14	A - Lv1	B - Lv2	C - Lv2	D - Lv1	E - Lv2
15	A - Lv1	B - Lv2	C - Lv2	D - Lv2	E - Lv1
16	A - Lv1	B - Lv2	C - Lv2	D - Lv2	E - Lv2
17	A - Lv2	B - Lv1	C - Lv1	D - Lv1	E - Lv1
18	A - Lv2	B - Lv1	C - Lv1	D - Lv1	E - Lv2
19	A - Lv2	B - Lv1	C - Lv1	D - Lv2	E - Lv1
20	A - Lv2	B - Lv1	C - Lv1	D - Lv2	E - Lv2
21	A - Lv2	B - Lv1	C - Lv2	D - Lv1	E - Lv1
22	A - Lv2	B - Lv1	C - Lv2	D - Lv1	E - Lv2
23	A - Lv2	B - Lv1	C - Lv2	D - Lv2	E - Lv1
24	A - Lv2	B - Lv1	C - Lv2	D - Lv2	E - Lv2
25	A - Lv2	B - Lv2	C - Lv1	D - Lv1	E - Lv1
26	A - Lv2	B - Lv2	C - Lv1	D - Lv1	E - Lv2
27	A - Lv2	B - Lv2	C - Lv1	D - Lv2	E - Lv1
28	A - Lv2	B - Lv2	C - Lv1	D - Lv2	E - Lv2

Table 5.16: Continuation of Table 5.15 (Experiments designed for instance sizes 30 and 45)

EXPERIMENTS	**FACTORS**				
29	A - Lv2	B - Lv2	C - Lv2	D - Lv1	E - Lv1
30	A - Lv2	B - Lv2	C - Lv2	D - Lv1	E - Lv2
31	A - Lv2	B - Lv2	C - Lv2	D - Lv2	E - Lv1
32	A - Lv2	B - Lv2	C - Lv2	D - Lv2	E - Lv2

Table 5.17: Experiments designed for instance size 60

EXPERIMENTS	**FACTORS**				
33	A - Lv1	B - Lv1	C - Lv1	D - Lv1	E - Lv3
34	A - Lv1	B - Lv1	C - Lv1	D - Lv2	E - Lv3
35	A - Lv1	B - Lv1	C - Lv2	D - Lv1	E - Lv3
36	A - Lv1	B - Lv1	C - Lv2	D - Lv2	E - Lv3
37	A - Lv1	B - Lv2	C - Lv1	D - Lv1	E - Lv3
38	A - Lv1	B - Lv2	C - Lv1	D - Lv2	E - Lv3
39	A - Lv1	B - Lv2	C - Lv2	D - Lv1	E - Lv3
40	A - Lv1	B - Lv2	C - Lv2	D - Lv2	E - Lv3
41	A - Lv2	B - Lv1	C - Lv1	D - Lv1	E - Lv3
42	A - Lv2	B - Lv1	C - Lv1	D - Lv2	E - Lv3
43	A - Lv2	B - Lv1	C - Lv2	D - Lv1	E - Lv3
44	A - Lv2	B - Lv1	C - Lv2	D - Lv2	E - Lv3
45	A - Lv2	B - Lv2	C - Lv1	D - Lv1	E - Lv3
46	A - Lv2	B - Lv2	C - Lv1	D - Lv2	E - Lv3
47	A - Lv2	B - Lv2	C - Lv2	D - Lv1	E - Lv3
48	A - Lv2	B - Lv2	C - Lv2	D - Lv2	E - Lv3

You can find the list of experiments designed based on instance sizes 30 and 45 in Tables 5.15 and 5.16. Similarly, the list of experiments designed based on a instance size of 60 can be found in Table 5.17.

5.3.2 Determining Proper S^{Total} Values for Computational Experiments

The appropriate determination of the S^{Total} parameter values in the experiment design created in Section 5.3.1 is crucial for obtaining accurate results. Arbitrarily setting the time frame for the system to complete its service would lead to infeasible outcomes. Therefore, it is necessary to determine suitable S^{Total} values for the experiments.

Systems with different service points will have varying total system times. While systems with 30 service points require relatively less time on average, the required total time to complete operation will increase as the number of service points increases. In addition to the number of service points, the energy consumption parameters of both the drone and the MC also play a significant role in determining S^{Total}. As explained in Section 5.2.2, if the MC's parameters are larger than those of the drone's, the MC's route will be shorter. Consequently, the drone will deviate more from its direct path between service points. This will increase the distance covered by the drone and thus prolong the operation time. Hence, we conducted this study to find suitable S^{Total} values under these two situations.

According to the experimental design formulated in Section 5.3.1, experiments will be conducted on two different energy consumption parameter sets and three different service point numbers (instance size with 30, 45, 60). This requires a total of six different S^{Total} values. To determine these S^{Total} values, matheuristic that minimizes the total energy consumption and the total system time together needs to be executed. The objective function of this model can be seen in (5.2). The coefficient of the S_{Total} is the smaller than the coefficient of energy consumption. Because the main purpose is not to minimize the S_{Total}, but to determine the S_{Total} value in cases where we minimize energy consumption.

$$\min \quad \sum_{i=0}^{n} E_i + (B_0^{MC} - B_{n+1}^{MC}) + 0.1 \times S_{Total} + BigM \times \sum_{i=0}^{n+1} p_i^2 \qquad (5.2)$$

Table 5.18: All S^{Total} values according to instance sizes and parameter sets (defined in Section 5.3.1 Experimental Design)

S^{Total} **(sec)**						
	Parameter Set 1			**Parameter Set 2**		
Levels	**Instance size 30**	**Instance size 45**	**Instance size 60**	**Instance size 30**	**Instance size 45**	**Instance size 60**
Set1	243	308	514	408	408	821
Set2	265	320	-	298	353	-
Set3	286	352	-	380	531	-
Average	265	327	514	362	431	821
Value1 (%110)	292	360	565	398	474	903
Value2 (%120)	318	392	617	434	517	985

The method we employed to find the appropriate S^{Total} is as follows: Randomly generate three instances in Excel for each instance size considered. Run this model for these generated instances for the two different energy consumption parameter sets. Thus, obtain three S^{Total} values for each combination of instance sizes and parameter sets. We take the average of these three values and obtain an S^{Total} value for each experiment. Table 5.18 presents the calculation of the S^{Total} values to be used in the computational experiments.

Let's illustrate with an example using Table 5.18. To determine the S^{Total} value to be used in an experiment with instance size 45 and Parameter Set 1, matheuristic that minimizes the total energy consumption and the total system time together was executed for three different random instances with instance size 45 and Parameter Set 1. The obtained results are 308 sec, 320 sec, 352 sec. The average of these results is 327 sec. The experiments are conducted based on the newly derived values by adding a certain margin to the obtained average value. The reason behind this is that these values are obtained by taking the average of the optimal S^{Total} values for three different instances. Since 327 sec is an average value, the S^{Total} value may be higher than this for some instances. To avoid creating excessively tight constraints in such cases, the experiments are conducted by the S^{Total} value with an additional 10%

and 20% of the average value. Therefore, for the experiment with instance size 45 - Parameter Set 1, the S^{Total} values to be used are 360 sec ($327 \times 110\%$) and 392 sec ($327 \times 120\%$).

5.3.3 Results of the Experiments

We conducted our matheuristic model on multiple instances according to the experimental setup. We created five random instances with instance sizes of 30 and 45, and one instance with an instance size of 60. We ran all experiments with these five instances with instance sizes of 30 and 45. We performed all experiments on the same instances to analyze the results by comparing them. For the experiments with a instance size of 60, we ran them on this single randomly generated instance.

We will discuss the experimental results in three different groups. In the first group, we will examine the impact of the differences in algorithms in Factors A and B on the results. In the second group, we will analyze the effects of different parameter sets of the energy consumption function and different S^{Total} values determined by these parameter sets and instance size on the results. In these two parts, we will discuss the results based on instances with instance sizes of 30 and 45. In the third part, we will focus on the experiments with a instance size of 60 and the impact of problem size on time.

5.3.3.1 Discussion of the Results According to Factors A and B

This section will focus on the first analysis mentioned at the beginning of Section 5.3.1. We will examine the impact of the different setups created with minor algorithm differences on the results. Let us briefly summarize the levels provided by Factors A and B, which indicate these algorithm differences. In Factor A, the levels are as follows: running the 2-Opt Algorithm with the Speed Up Method or running it without the Speed Up Method.

In Factor B, the levels are as follows: while sequentially searching for the best route among the alternative routes generated by the 2-Opt Algorithm, either stopping when the first improvement is achieved or continuing the search without stopping (trying all levels).

Tables 5.19, 5.20, 5.21, and 5.22 present the results for experiments conducted on instances with a instance size of 30, while Tables 5.23, 5.24, 5.25, and 5.26 present the results for experiments conducted on instances with a instance size of 45.

We compare the experiments with the same levels selected from Factors C and D and the different levels in Factors A and B from these tables. For example, in Table 5.19, the results of experiments using Parameter Set 1 in Factor C and Value 1 of S^{Total} in Factor D are given according to the combinations of Factor A's and B's levels. In Experiment 1, the levels are Factor A: With Speed Up and Factor B: Stop After 1^{st} Improvement. In Experiment 9, the levels are Factor A: With Speed Up and Factor B: Do Not Stop After 1^{st} Improvement. In Experiment 17, the levels are Factor A: Without Speed Up and Factor B: Stop After 1^{st} Improvement. In Experiment 25, the levels are Factor A: Without Speed Up and Factor B: Do Not Stop After 1^{st} Improvement. The results obtained with these setups will be evaluated according to these tables. The "Best Objective Function Value (OFV)" row in these tables represents the best solution obtained from the mat-heuristic model. The "Number of Iterations" row indicates the number of iterations required to reach this best solution. The "Computational Time" row provides the run time of the model.

The results in Table 5.19 show that better results are obtained from the setups with With Speed Up, except for Sample 3. Based on these results, the With Speed Up level yields better results. To understand the effect of Factor B, the results of Experiments 1 and 17 need to be compared with the results of Experiments 9 and 25. However, upon comparing the results, it is not possible to claim that one level consistently or generally outperforms the other. In Experiment 1, the results for Sample 2 and Sample 3 are better than Experiment 9, while the results for Sample 4 and Sample 5 are worse. For Sample 1, the results are equal in both experiments. Looking at Experiment 17, the results for Sample 1 are better than Experiment 25, while the results for Sample 3, 4, and 5 are worse. The results are equal for Sample 2 in both experiments. Based on the results, it is not possible to make a clear inference about the superiority of one level over the other.

Table 5.19: Experiments with different levels in Factor A and B and the same levels Factor C, D and E (Factor C: Parameter Set 1, Factor D: Value 1 of S^{Total}, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 1	**Best OFV (Joule)**	423.590	488.864	490.540	466.842	460.804
With Speed Up	**Number of Iterations**	2	5	4	3	9
Stop	**Computational Time (sec)**	601.174	1095.075	729.157	606.675	1733.088
Experiment 9	**Best OFV (Joule)**	423.590	522.080	491.777	462.047	451.943
With Speed Up	**Number of Iterations**	2	3	4	3	4
Do Not Stop	**Computational Time (sec)**	484.860	2029.018	1296.698	1038.566	1755.576
Experiment 17	**Best OFV (Joule)**	428.884	563.025	489.784	493.938	566.726
Without Speed Up	**Number of Iterations**	1	0	6	1	2
Stop	**Computational Time (sec)**	7235.643	7258.957	7248.598	7275.828	7263.506
Experiment 25	**Best OFV (Joule)**	432.367	563.025	485.846	489.789	545.389
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7230.050	7235.502	7228.842	7262.512	7241.962

From the results in Table 5.20, we can infer that setups with With Speed Up generally yield better results, except for Sample 3. Based on these results, the With Speed Up level provides better outcomes. To reach a conclusion from the perspective of Factor B, we need to compare the results of Experiments 3 and 19 with the results of Experiments 11 and 27. Upon comparing the results, in Experiment 3, the results for Sample 1 and Sample 2 are better than Experiment 11, while only the results for Sample 5 are worse. The results for Samples 3 and 4 are equal in both experiments. Looking at Experiment 19, the results for Sample 3 and Sample 4 are worse than Experiment 27, while the result for Sample 5 is better. The results for Samples 1 and 2 are equal in both experiments. Based on the results, it is not possible to make a clear inference about the superiority of one level over the other.

Table 5.20: Experiments with different levels in Factors A and B and the same levels Factors C, D and E (Factor C: Parameter Set 1, Factor D: Value 2 of S^{Total}, Factor E: Instance size 30))

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 3	**Best OFV (Joule)**	422.790	486.408	507.977	459.394	485.108
With Speed Up	**Number of Iterations**	2	5	0	3	7
Stop	**Computational Time (sec)**	382.810	1030.328	337.501	589.307	1665.878
Experiment 11	**Best OFV (Joule)**	422.790	513.803	507.977	459.394	449.862
With Speed Up	**Number of Iterations**	2	3	0	3	4
Do Not Stop	**Computational Time (sec)**	517.370	2183.151	384.138	1122.445	1911.621
Experiment 19	**Best OFV (Joule)**	431.556	552.964	485.601	489.008	530.779
Without Speed Up	**Number of Iterations**	0	0	6	1	1
Stop	**Computational Time (sec)**	7263.003	7238.837	7227.455	7225.481	7264.801
Experiment 27	**Best OFV (Joule)**	431.556	552.964	481.910	485.280	535.766
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7238.346	7254.555	7250.285	7252.233	7231.998

From the results in Table 5.21, setups with With Speed Up generally provide significantly better results, except for Sample 3. Only in Experiment 13, the result for Sample 4 is higher than in Experiment 21. The With Speed Up level yields relatively good outcomes based on these results. To evaluate the levels of Factor B, let us compare the results of Experiments 5 and 21 with the results of Experiments 13 and 29. Upon examining the results, in Experiments 5 and 13, the results for Samples 1 and 3 are equal in both experiments, while for Sample 2 and Sample 5, Experiment 13 has better results. For Sample 4, the result in Experiment 5 is better. Looking at Experiment 21, the results for Sample 3 and Sample 4 are better than Experiment 29, while for Samples 2 and 5, the results are worse. The result for Sample 1 is equal in both experiments. Based on the results, it is not possible to make a clear inference about the superiority of one level over the other.

Table 5.21: Experiments with different levels in Factor A and B and the same levels Factor C, D and E (Factor C: Parameter Set 2, Factor D: Value 1 of S^{Total}, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 5	**Best OFV (Joule)**	502.490	579.829	602.416	554.545	586.704
With Speed Up	**Number of Iterations**	1	7	0	4	6
Stop	**Computational Time (sec)**	342.365	1113.609	332.943	565.802	1869.628
Experiment 13	**Best OFV (Joule)**	502.490	556.968	602.416	605.076	573.776
With Speed Up	**Number of Iterations**	1	2	0	2	1
Do Not Stop	**Computational Time (sec)**	412.145	1715.231	338.955	984.165	1230.938
Experiment 21	**Best OFV (Joule)**	517.431	594.088	566.147	556.089	623.655
Without Speed Up	**Number of Iterations**	0	5	3	2	1
Stop	**Computational Time (sec)**	7310.584	7264.246	7214.771	7278.784	7265.718
Experiment 29	**Best OFV (Joule)**	517.431	578.194	567.228	577.661	617.460
Without Speed Up	**Number of Iterations** er	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7296.954	7271.257	7248.758	7287.284	7280.641

From the results in Table 5.22, we can observe that setups with With Speed Up generally provide significantly better results, except for Sample 3 in Experiment 15, where it is higher than in Experiment 31. Based on these results, the With Speed Up level generally yields better outcomes. To evaluate the levels of Factor B, let us compare the results of Experiments 7 and 23 with the results of Experiments 15 and 31. Upon examining the results, in Experiments 7 and 23, the results for Samples 1, 3, and 4 are equal in both experiments, while for Samples 2, Experiment 15 has better results. For Sample 5, the result in Experiment 5 is better. In Experiment 31, the results for Samples 2,3 and 5 are better. Based on these results, it can be suggested that in setups with the Without Speed Up level, the Do Not Stop after 1^{st} improvement level tends to yield better results. However, for setups with the With Speed Up level, it is not easy to make a definitive comment regarding the levels in Factor B.

Table 5.22: Experiments with different levels in Factor A and B and the same levels Factor C, D and E (Factor C: Parameter Set 2, Factor D: Value 2 of S^{Total}, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 7	**Best OFV (Joule)**	502.191	611.521	598.714	724.110	555.988
With Speed Up	**Number of Iterations**	2	4	0	0	8
Stop	**Computational Time (sec)**	400.145	1016.825	334.025	468.665	1854.832
Experiment 15	**Best OFV (Joule)**	502.191	556.615	598.714	724.110	572.540
With Speed Up	**Number of Iterations**	2	2	0	0	1
Do Not Stop	**Computational Time (sec)**	475.236	1771.683	349.722	517.980	1307.299
Experiment 23	**Best OFV (Joule)**	517.430	584.090	577.451	554.940	615.943
Without Speed Up	**Number of Iterations**	0	4	4	2	1
Stop	**Computational Time (sec)**	7235.340	7208.648	7273.410	7231.415	7285.301
Experiment 31	**Best OFV (Joule)**	517.430	572.949	561.511	575.699	610.535
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7311.582	7230.398	7216.537	7259.432	7232.648

When examining the results obtained from the instances with a instance size of 30, it can be observed that alternative setups yield different outcomes across different instances. While it can be said that the With Speed Up level in Factor A tends to provide relatively better results, the presence of cases where the Without Speed Up level yields better outcomes in different instances demonstrates that it is possible to obtain better results with Without Speed Up in certain scenarios.

Based on these experimental results, no definitive inference can be made for Factor B. It has been observed that both levels can yield better results in different instances. When facing the choice between these two levels, Computational Time can be a good criterion to consider. It can be noted that the Computational Time for the Stop 1^{st} Improvement level is considerably lower than that for the Do Not Stop level. This makes the first level more attractive in limited times.

In general, when examining the results in all tables, it can be said that With Speed Up - Stop and With Speed Up - Do Not Stop setups from the combinations of Factor A's and B's levels tend to yield better outcomes than other setups from combinations. However, making a direct comparison between these levels seems challenging. So far, we have examined the results of experiments conducted on instances with a instance size of 30. In the following tables, we will analyze the results of experiments conducted on instances with a instance size of 45.

Based on the results in Table 5.23, to determine the best level for Factor A, we need to evaluate Experiments 2 and 18, and Experiments 10 and 26. It can be observed that, within the pairs of experiments, although not significantly, better results are obtained in the experiments where the With Speed Up level is chosen. For Factor B, we should consider Experiments 2 and 10 in itself and Experiments 18 and 26 in itself. While in itself many results are equal, it can be said that the Do Not Stop level tends to provide better outcomes among the remaining results. Therefore, according to Table 5.23, the With Speed Up level can be preferred for Factor A, and the Do Not Stop level can be preferred for Factor B.

Table 5.23: Experiments with different levels in Factor A and B and the same levels Factor C, D and E (Factor C: Parameter Set 1, Factor D: Value 1 of S^{Total}, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 2	**Best OFV (Joule)**	549.076	563.392	599.094	599.991	206.858
With Speed Up	**Number of Iterations**	4	4	0	0	9
Stop	**Computational Time (sec)**	975.722	1066.235	537.749	560.585	305.533
Experiment 10	**Best OFV (Joule)**	502.779	557.820	599.094	599.991	206.858
With Speed Up	**Number of Iterations**	3	4	0	0	7
Do Not Stop	**Computational Time (sec)**	1773.770	1293.907	684.675	606.631	592.092
Experiment 18	**Best OFV (Joule)**	530.580	546.943	607.867	600.470	229.632
Without Speed Up	**Number of Iterations**	1	2	4	0	4
Stop	**Computational Time (sec)**	7237.073	7230.306	7248.225	7266.962	7222.877
Experiment 26	**Best OFV (Joule)**	524.129	569.887	599.094	600.470	232.125
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7263.364	7283.917	7255.895	7287.535	7231.180

According to the results in Table 5.24, by comparing Experiment 4 with 12 and Experiment 20 with 28, it can be concluded that the Do Not Stop level in Factor B yields better results only when the Without Speed Up level is chosen in Factor A. Similarly, for Factor A, it does not seem possible to determine the best level for all scenarios in the table. However, it can be clearly stated that the With Speed Up level in Factor A provides better results when the Do Not Stop level is chosen in Factor B.

Table 5.24: Experiments with different levels in Factor A and B and the same levels Factor C, D and E (Factor C: Parameter Set 1, Factor D: Value 2 of S^{Total}, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 4	**Best OFV (Joule)**	544.086	562.721	589.048	592.117	206.858
With Speed Up	**Number of Iterations**	4	3	1	0	9
Stop	**Computational Time (sec)**	1078.963	962.363	843.187	812.885	324.748
Experiment 12	**Best OFV (Joule)**	490.398	553.429	589.048	592.117	206.858
With Speed Up	**Number of Iterations**	3	4	1	0	7
Do Not Stop	**Computational Time (sec)**	1805.153	1244.062	954.778	742.555	598.578
Experiment 20	**Best OFV (Joule)**	526.292	546.328	599.492	592.117	229.632
Without Speed Up	**Number of Iterations**	1	1	5	0	4
Stop	**Computational Time (sec)**	7238.842	7239.428	7328.783	7318.851	7215.118
Experiment 28	**Best OFV (Joule)**	520.610	566.285	591.273	592.117	232.125
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7266.981	7257.727	7259.201	7307.049	7233.933

According to Table 5.25, we can say that the best level for Factor A is With Speed Up. For Factor B, when comparing the results between Experiment 6 and 14, it can be seen that they have an equal number of good outcomes. Similarly, between Experiments 22 and 30, there is an equal number of good results as well. This implies that, according to Table 5.25, no level in Factor B can be deemed better under the specified conditions.

Table 5.25: Experiments with different levels in Factors A and B and the same levels Factors C, D and E (Factor C: Parameter Set 2, Factor D: Value 1 of S^{Total}, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 6	**Best OFV (Joule)**	585.823	750.511	712.464	683.623	207.512
With Speed Up	**Number of Iterations**	5	4	2	0	9
Stop	**Computational Time (sec)**	1880.042	902.994	529.673	747.662	290.331
Experiment 14	**Best OFV (Joule)**	660.492	668.492	692.310	683.623	207.512
With Speed Up	**Number of Iterations**	2	4	1	0	7
Do Not Stop	**Computational Time (sec)**	1574.539	1321.498	748.114	742.355	627.825
Experiment 22	**Best OFV (Joule)**	596.103	680.378	771.479	683.623	245.377
Without Speed Up	**Number of Iterations**	8	3	5	0	3
Stop	**Computational Time (sec)**	7235.495	7330.650	7379.510	7249.994	7237.798
Experiment 30	**Best OFV (Joule)**	680.885	690.046	744.774	683.623	234.084
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7263.641	7225.692	7228.341	7237.766	7234.009

From the results in Table 5.26, it is evident that setups with With Speed Up yield significantly better results. However, for Factor B, we cannot make a clear inference regarding which level provides better results. This is because the experiments with the Stop level and the Do Not Stop level have an equal number of good results.

Table 5.26: Experiments with different levels in Factors A and B and the same levels Factors C, D and E (Factor C: Parameter Set 2, Factor D: Value 2 of S^{Total}, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 8	**Best OFV (Joule)**	582.053	663.912	684.445	662.357	207.512
With Speed Up	**Number of Iterations**	7	5	2	0	9
Stop	**Computational Time (sec)**	2467.931	1192.681	1002.364	792.774	314.304
Experiment 16	**Best OFV (Joule)**	668.581	663.912	684.445	662.357	207.512
With Speed Up	**Number of Iterations**	2	5	2	0	7
Do Not Stop	**Computational Time (sec)**	1744.802	1516.666	1174.126	788.384	672.009
Experiment 24	**Best OFV (Joule)**	595.560	672.337	766.244	662.357	245.377
Without Speed Up	**Number of Iterations**	5	4	4	0	3
Stop	**Computational Time (sec)**	7336.421	7226.820	7296.573	7243.883	7231.372
Experiment 32	**Best OFV (Joule)**	674.278	685.844	736.784	662.357	234.084
Without Speed Up	**Number of Iterations**	0	0	0	0	0
Do Not Stop	**Computational Time (sec)**	7240.339	7292.850	7236.665	7237.376	7225.664

From the tables generated for instances with a instance size of 45, it can be observed that the experiments with the With Speed Up level in Factor A generally yield better results. However, there are scenarios where the Without Speed Up level performs better. Therefore, we cannot definitively infer that With Speed Up always provides the best results for Factor A. Factor B has no superiority of one level. Different combinations of the factor's levels may lead to both levels being preferred.

When examining the results of experiments conducted on instances with a instance size of 45, there is no significant difference among the results to determine the best combination of Factors A and B. Therefore, it is not possible to conclude that experiments with the With Speed Up level outperform the Stop or Do Not Stop level, as it depends on the specific scenario.

According to all the results, it can be inferred that the 2-Opt Algorithm performs better when combined with the Speed Up Method. The most significant factor behind this conclusion in our experiments is the limited time. In the experiments where the 2-Opt Algorithm is run without the Speed Up Method, the computational time becomes very high due to the large number of levels being explored. In this study, due to limited time, we had to restrict these experiments to 2 hours. This reduces the likelihood of finding alternative routes that could improve the results within the given time frame. By using the Speed Up Method to select alternative routes with the potential for improvement, we increase the chances of improving the results.

For example, in a 30-point problem, as mentioned in Section 4.2.2, there are 434 alternative routes according to the formula $\binom{n+1}{n} - (n+1)$. Since we construct the initial route based on the Nearest Neighbor Algorithm, few routes among the alternatives can significantly improve upon this initial route. In this study, there were cases where, in experiments with the Speed Up Method, only a maximum of 10 routes out of the 434 alternatives had the potential to yield better results. From this, we can make the following inference: the computational time for experiments with 10 alternative routes (i.e., experiments with Speed Up) is significantly shorter than the computational time for experiments with 434 alternative routes (i.e., experiments without Speed Up). This allows for faster attainment of better results. It can be said that if there were no time constraints for the 2-Opt Algorithm without the Speed Up Method,

even better results could be achieved by trying a larger number of alternatives.

When examining the results for Factor B, we cannot claim that there is a significant difference between the two levels. However, stopping after the 1^{st} improvement is better regarding computational time and allows for reaching the final result more quickly. The reason why many results are the same for both levels is as follows: In the first level, when exploring alternatives, the route where the first improvement is achieved is selected. In the second level, despite trying all alternative routes after the first improvement, there is no improvement. As a result, the same route is selected in both cases. Stopping after the first improvement sometimes means that better levels are overlooked. However, there is also the possibility that the selected route could be further improved in subsequent iterations by generating alternative 2-Opt routes. Therefore, stopping or not stopping after the first improvement cannot be compared directly; they provide alternative approaches to each other.

We can see no direct correlation between the number of iterations and the results when examining the tables. For example, in Table 5.21, Experiments 6 and 14 have both equal and different numbers of iterations. For Sample 1, Experiment 6 has a higher number of iterations, and the obtained result is better than Experiment 14. For Sample 2, even though the number of iterations is the same, the result of Experiment 14 is better. For Sample 4, the number of iterations is equal, and the results are the same. For Sample 3, despite Experiment 6 having a higher number of iterations, the result is higher, meaning worse. For Sample 5, the results are the same despite having different numbers of iterations.

As a result, no direct relationship can be established between the number of iterations and the results in these experiments. The reason why the results in setups with Without Speed Up and Do Not Stop after 1^{st} improvement levels reach the solution in the first iteration is due to the large number of 2-Opt alternatives and the fact that all these alternatives are tried because it does not stop after the first improvement. It is resulted in the model's 2-hour solution time limit being reached without trying all the alternatives, i.e., without completing the first iteration.

5.3.3.2 Discussion of the Results According to Factors C and D

This section will examine the results regarding the second and third items mentioned in Section 5.3.1. We will analyze the effects of different Factors C and D levels on the results using experiment setups created by their combinations. Table 5.14 in Section 5.3.1 shows that Factor C consists of two different Energy Consumption Parameter Sets. In the first level, the energy consumption parameters of the drone and MC are equal. In the second level, the MC's parameters are twice that of the drone's. While analyzing this factor, we will examine how different parameter sets affect the route and energy consumption. In Factor D, the levels are two different values of S^{Total}. We explained how the values of S^{Total} are determined for different factor combinations in Section 5.3.2. We will examine the effects of S^{Total} values on the route, energy consumption, and the speeds of the drone and MC.

Tables 5.27, 5.28, 5.29, and 5.30 present experiments with a instance size of 30 instances, while Tables 5.31, 5.32, 5.33, and 5.34 present experiments with a instance size of 45 instances. In these tables, the selected levels from Factor A and B remain the same, while the levels in Factors C and D differ. For example, in Table 5.27, the results of experiments with "With Speed Up" for Factor A and "Stop after 1^{st} improvement" for Factor B are given in accordance with the combinations of Factors C and D. In Experiment 1, Factor C is Parameter Set 1, and Factor D is Value 1 of S^{Total}. In Experiment 7, Factor C is Parameter Set 2, and Factor D is Value 2 of S^{Total}.

Firstly, we will evaluate the effects of Factors C and D on Energy Consumption. Subsequently, we will examine the routes and vehicle speeds obtained from the experiments regarding the influences of these two factors.

We recognize that the most significant impact of these two factors will be on the routes and vehicle speeds. We will examine the effects on these two outcomes using graphs and tables. Additionally, we will analyze all the tables that include energy consumption, taking into account the instance sizes, in a single evaluation.

Table 5.27: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: With Speed Up, Factor B:Stop after 1^{st} Improvements, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 1	**Best OFV (Joule)**	423.590	488.864	490.540	466.842	460.804
Parameter Set 1	**Number of Iterations**	2	5	4	3	9
Value 1 of S^{Total}	**Computational Time (sec)**	601.174	1095.075	729.157	606.675	1733.088
Experiment 3	**Best OFV (Joule)**	422.790	486.408	507.977	459.394	485.108
Parameter Set 1	**Number of Iterations**	2	5	0	3	7
Value 2 of S^{Total}	**Computational Time (sec)**	382.810	1030.328	337.501	589.307	1665.878
Experiment 5	**Best OFV (Joule)**	502.490	579.829	602.416	554.545	586.704
Parameter Set 2	**Number of Iterations**	1	7	0	4	6
Value 1 of S^{Total}	**Computational Time (sec)**	342.365	1113.609	332.943	565.802	1869.628
Experiment 7	**Best OFV (Joule)**	502.191	611.521	598.714	724.110	555.988
Parameter Set 2	**Number of Iterations**	2	4	0	0	8
Value 2 of S^{Total}	**Computational Time (sec)**	400.145	1016.825	334.025	468.665	1854.832

Table 5.28: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: With Speed Up, Factor B: Do Not Stop after 1^{st} Improvements, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 9	**Best OFV (Joule)**	423.590	522.080	491.777	462.047	451.943
Parameter Set 1	**Number of Iterations**	2	3	4	3	4
Value 1 of S^{Total}	**Computational Time (sec)**	484.860	2029.018	1296.698	1038.566	1755.576
Experiment 11	**Best OFV (Joule)**	422.790	513.803	507.977	459.394	449.862
Parameter Set 1	**Number of Iterations**	2	3	0	3	4
Value 2 of S^{Total}	**Computational Time (sec)**	517.370	2183.151	384.138	1122.445	1911.621
Experiment 13	**Best OFV (Joule)**	502.490	556.968	602.416	605.076	573.776
Parameter Set 2	**Number of Iterations**	1	2	0	2	1
Value 1 of S^{Total}	**Computational Time (sec)**	412.145	1715.231	338.955	984.165	1230.938
Experiment 15	**Best OFV (Joule)**	502.191	556.615	598.714	724.110	572.540
Parameter Set 2	**Number of Iterations**	2	2	0	0	1
Value 2 of S^{Total}	**Computational Time (sec)**	475.236	1771.683	349.722	517.980	1307.299

Table 5.29: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: Without Speed Up, Factor B: Stop after 1^{st} Improvements, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 17	**Best OFV (Joule)**	428.884	563.025	489.784	493.938	566.726
Parameter Set 1	**Number of Iterations**	1	0	6	1	2
Value 1 of S^{Total}	**Computational Time (sec)**	7235.643	7258.957	7248.598	7275.828	7263.506
Experiment 19	**Best OFV (Joule)**	431.556	552.964	485.601	489.008	530.779
Parameter Set 1	**Number of Iterations**	0	0	6	1	1
Value 2 of S^{Total}	**Computational Time (sec)**	7263.003	7238.837	7227.455	7225.481	7264.801
Experiment 21	**Best OFV (Joule)**	517.431	594.088	566.147	556.089	623.655
Parameter Set 2	**Number of Iterations**	0	5	3	2	1
Value 1 of S^{Total}	**Computational Time (sec)**	7310.584	7264.246	7214.771	7278.784	7265.718
Experiment 23	**Best OFV (Joule)**	517.430	584.089	577.451	554.940	615.943
Parameter Set 2	**Number of Iterations**	0	4	4	2	1
Value 2 of S^{Total}	**Computational Time (sec)**	7235.340	7208.648	7273.410	7231.414	7285.301

Table 5.30: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: Without Speed Up, Factor B: Do Not Stop after 1^{st} Improvements, Factor E: Instance size 30)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 25	**Best OFV (Joule)**	432.367	563.025	485.846	489.789	545.389
Parameter Set 1	**Number of Iterations**	0	0	0	0	0
Value 1 of S^{Total}	**Computational Time (sec)**	7230.050	7235.502	7228.842	7262.512	7241.962
Experiment 27	**Best OFV (Joule)**	431.556	552.964	481.910	485.280	535.766
Parameter Set 1	**Number of Iterations**	0	0	0	0	0
Value 2 of S^{Total}	**Computational Time (sec)**	7238.346	7254.555	7250.285	7252.233	7231.998
Experiment 29	**Best OFV (Joule)**	517.431	578.194	567.228	577.661	617.460
Parameter Set 2	**Number of Iterations**	0	0	0	0	0
Value 1 of S^{Total}	**Computational Time (sec)**	7296.954	7271.257	7248.758	7287.284	7280.641
Experiment 31	**Best OFV (Joule)**	517.430	572.949	561.511	575.699	610.535
Parameter Set 2	**Number of Iterations**	0	0	0	0	0
Value 2 of S^{Total}	**Computational Time (sec)**	7311.582	7230.398	7216.537	7259.432	7232.648

When examining the results of these experiments conducted for Instance size 30, we observe that in Parameter Set 1 and Parameter Set 2, the energy consumption parameters of the drone are equal. However, in Parameter Set 2, the MC parameters are higher than in Parameter Set 1. Therefore, a reduction in the MC's route is expected. The results of experiments conducted with the same factor combinations and different Parameter Sets were compared among themselves (e.g., Experiment 1 with 3, Experiment 13 with 15, Experiment 25 with 27, and so on). It was found that the results of experiments conducted with Parameter Set 1 are all smaller than the results of experiments conducted with Parameter Set 2. This indicates that although the route of the MC is shortened, the total energy consumption has increased due to both the elongation of the drone's route and the fact that the MC's route has not been shortened by more than half.

When analyzing the effects of the System's Time Limit from the tables, we can observe that in most experiments where Value 2 of S^{Total} is used, the energy consumption decreases or remains the same. In some experimental results, even though the S^{Total} value is high, the total energy consumption is lower. The reason is that the final result is reached through minor improvements in iterations, which may result in better alternatives not being explored. In experiments with a lower S^{Total}, alternative routes with a lower energy consumption could be possible, considering the order and location of rendezvous points and the model having solved this route within the relevant factor context. This indicates that better results can be obtained even in experiments conducted with a smaller S^{Total}. Many experiments encountering such situations utilize setups that stop after the 1^{st} improvement. Therefore, it is likely that the expected improvements are not realized due to not exploring all alternative routes.

So far, we have examined the results of experiments conducted with a instance size of 30 instances. The subsequent tables present the results of experiments conducted with a instance size of 45 instances.

Table 5.31: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: With Speed Up, Factor B: Stop after 1^{st} Improvements, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 2	**Best OFV (Joule)**	549.076	563.392	599.094	599.991	206.858
Parameter Set 1	**Number of Iterations**	4	4	0	0	9
Value 1 of S^{Total}	**Computational Time (sec)**	975.722	1066.235	537.749	560.585	305.533
Experiment 4	**Best OFV (Joule)**	544.086	562.721	589.048	592.117	206.858
Parameter Set 1	**Number of Iterations**	4	3	1	0	9
Value 2 of S^{Total}	**Computational Time (sec)**	1078.963	962.363	843.187	812.885	324.748
Experiment 6	**Best OFV (Joule)**	585.823	750.511	712.464	683.623	207.512
Parameter Set 2	**Number of Iterations**	5	4	2	0	9
Value 1 of S^{Total}	**Computational Time (sec)**	1880.042	902.994	529.673	747.662	290.331
Experiment 8	**Best OFV (Joule)**	582.053	663.912	684.445	662.357	207.512
Parameter Set 2	**Number of Iterations**	7	5	2	0	9
Value 2 of S^{Total}	**Computational Time (sec)**	2467.931	1192.681	1002.364	792.774	314.304

Table 5.32: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: With Speed Up, Factor B: Do Not Stop after 1^{st} Improvements, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 10	**Best OFV (Joule)**	502.779	557.820	599.094	599.991	206.858
Parameter Set 1	**Number of Iterations**	3	4	0	0	7
Value 1 of S^{Total}	**Computational Time (sec)**	1773.770	1293.907	684.675	606.631	592.092
Experiment 12	**Best OFV (Joule)**	490.398	553.429	589.048	592.117	206.858
Parameter Set 1	**Number of Iterations**	3	4	1	0	7
Value 2 of S^{Total}	**Computational Time (sec)**	1805.153	1244.062	954.778	742.555	598.578
Experiment 14	**Best OFV (Joule)**	660.492	668.492	692.310	683.623	207.512
Parameter Set 2	**Number of Iterations**	2	4	1	0	7
Value 1 of S^{Total}	**Computational Time (sec)**	1574.539	1321.498	748.114	742.355	627.825
Experiment 16	**Best OFV (Joule)**	668.581	663.912	684.445	662.357	207.512
Parameter Set 2	**Number of Iterations**	2	5	2	0	7
Value 2 of S^{Total}	**Computational Time (sec)**	1744.802	1516.666	1174.126	788.384	672.009

Table 5.33: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: Without Speed Up, Factor B: Do Not Stop after 1^{st} Improvements, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 18	**Best OFV (Joule)**	530.580	546.943	607.867	600.470	229.632
Parameter Set 1	**Number of Iterations**	1	2	4	0	4
Value 1 of S^{Total}	**Computational Time (sec)**	7237.073	7230.306	7248.225	7266.962	7222.877
Experiment 20	**Best OFV (Joule)**	526.292	546.328	599.492	592.117	229.632
Parameter Set 1	**Number of Iterations**	1	1	5	0	4
Value 2 of S^{Total}	**Computational Time (sec)**	7238.842	7239.428	7328.783	7318.851	7215.118
Experiment 22	**Best OFV (Joule)**	596.103	680.378	771.479	683.623	245.377
Parameter Set 2	**Number of Iterations**	8	3	5	0	3
Value 1 of S^{Total}	**Computational Time (sec)**	7235.495	7330.650	7379.510	7249.994	7237.798
Experiment 24	**Best OFV (Joule)**	595.560	672.337	766.244	662.357	245.377
Parameter Set 2	**Number of Iterations**	5	4	4	0	3
Value 2 of S^{Total}	**Computational Time (sec)**	7336.421	7226.820	7296.573	7243.883	7231.372

Table 5.34: Experiments with different levels in Factors C and D and the same levels Factors A, B and E (Factor A: Without Speed Up, Factor B: Do Not Stop after 1^{st} Improvements, Factor E: Instance size 45)

		Sample 1	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**
Experiment 26	**Best OFV (Joule)**	524.129	569.887	599.094	600.470	232.125
Parameter Set 1	**Number of Iterations**	0	0	0	0	0
Value 1 of S^{Total}	**Computational Time (sec)**	7263.364	7283.917	7255.895	7287.535	7231.180
Experiment 28	**Best OFV (Joule)**	520.610	566.285	591.273	592.117	232.125
Parameter Set 1	**Number of Iterations**	0	0	0	0	0
Value 2 of S^{Total}	**Computational Time (sec)**	7266.981	7257.727	7259.201	7307.049	7233.933
Experiment 30	**Best OFV (Joule)**	680.885	690.046	744.774	683.623	234.084
Parameter Set 2	**Number of Iterations**	0	0	0	0	0
Value 1 of S^{Total}	**Computational Time (sec)**	7263.641	7225.692	7228.341	7237.766	7234.009
Experiment 32	**Best OFV (Joule)**	674.278	685.844	736.784	662.357	234.084
Parameter Set 2	**Number of Iterations**	0	0	0	0	0
Value 2 of S^{Total}	**Computational Time (sec)**	7240.339	7292.850	7236.665	7237.376	7225.664

When examining the experiments with a instance size of 45 for Factor B, it is evident that the total energy consumption is higher in experiments with Parameter Set 2. This aligns with our previous explanation, as the experiments with higher energy consumption parameters are expected to yield higher results.

To observe the effect of the System's Time Limit, pairs of experiments were compared where all factors were the same, but only the S^{Total} values differed (for example, Experiment 2 - 4). In contrast to the experiments with a instance size of 30, the energy consumption in almost all of these experiment pairs is lower in experiments where S^{Total} is higher. Based on these results, vehicles can adjust their speeds to further reduce energy consumption when the Total System Time is relaxed (i.e., when the System's Time Limit, S^{Total}, is larger). This can be observed by referring to Tables 5.35, 5.36, 5.37 and 5.38.

Table 5.35: Order and coordinates of all visited points by drone and speed of drone in the best solution of Sample 2 in Experiments 6 and 8

Experiment 6 - Sample 2				**Experiment 8 - Sample 2**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
0	58.000	37.000	-	0	58.000	37.000	-
1	58.000	37.000	2.023	1	58.000	37.000	1.753
2	62.000	45.000	2.023	2	62.000	45.000	1.753
3	47.000	51.000	2.023	3	47.000	51.000	1.753
4	51.000	57.000	2.023	4	51.000	57.000	1.753
5	50.000	60.000	2.023	5	50.000	60.000	1.753
6	61.000	71.000	2.023	6	61.000	71.000	1.753
7	67.000	59.000	2.023	7	67.000	59.000	1.753
8	85.000	54.000	2.023	8	58.980	45.262	1.691
9	59.048	42.113	1.947	9	90.000	44.000	1.691
10	89.000	47.000	1.947	10	89.000	47.000	1.691
11	90.000	44.000	1.947	11	85.000	54.000	1.691
12	95.000	59.000	1.947	12	95.000	59.000	1.691
13	97.000	68.000	1.947	13	97.000	68.000	1.691
14	97.000	70.000	1.947	14	97.000	70.000	1.691
15	84.000	85.000	1.947	15	84.000	85.000	1.691
16	72.000	84.000	1.947	16	72.000	84.000	1.691
17	66.000	85.000	1.947	17	66.000	85.000	1.691
18	45.230	56.498	1.894	18	48.020	57.980	1.684
19	42.000	97.000	1.894	19	42.000	97.000	1.684
20	31.000	88.000	1.894	20	31.000	88.000	1.684
21	17.000	98.000	1.894	21	17.000	98.000	1.684
22	9.000	76.000	1.894	22	9.000	76.000	1.684
23	22.000	70.000	1.894	23	22.000	70.000	1.684
24	30.000	67.000	1.894	24	30.000	67.000	1.684

Table 5.36: Order and coordinates of all visited points by drone and speed of drone in the best solution of Sample 2 in Experiments 6 and 8 (continuation of Table 5.35)

Experiment 6 - Sample 2				**Experiment 8 - Sample 2**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
25	40.123	54.797	2.023	25	44.147	55.965	1.753
26	24.000	63.000	2.023	26	24.000	63.000	1.753
27	17.000	62.000	2.023	27	17.000	62.000	1.753
28	14.000	41.000	2.023	28	14.000	41.000	1.753
29	38.440	45.510	2.023	29	43.866	49.015	1.713
30	25.000	41.000	2.023	30	25.000	41.000	1.713
31	23.000	29.000	2.023	31	23.000	29.000	1.713
32	13.000	18.000	2.023	32	13.000	18.000	1.713
33	0.000	18.000	2.023	33	0.000	18.000	1.713
34	17.000	3.000	2.023	34	17.000	3.000	1.713
35	49.204	32.349	1.814	35	44.000	17.000	1.713
36	50.000	4.000	1.814	36	58.296	32.975	1.753
37	59.000	3.000	1.814	37	57.000	16.000	1.753
38	70.000	4.000	1.814	38	65.000	19.000	1.753
39	84.000	0.000	1.814	39	71.000	28.000	1.753
40	99.000	1.000	1.814	40	66.000	31.000	1.753
41	94.000	15.000	1.814	41	72.000	39.000	1.753
42	81.000	21.000	1.814	42	76.000	39.000	1.753
43	58.186	36.212	2.023	43	83.000	31.000	1.753
44	83.000	31.000	2.023	44	61.937	32.594	1.656
45	76.000	39.000	2.023	45	81.000	21.000	1.656
46	72.000	39.000	2.023	46	94.000	15.000	1.656
47	66.000	31.000	2.023	47	99.000	1.000	1.656
48	71.000	28.000	2.023	48	84.000	0.000	1.656

Table 5.37: Order and coordinates of all visited points by drone and speed of drone in the best solution of Sample 2 in Experiments 6 and 8 (continuation of Table 5.36)

Experiment 6 - Sample 2				**Experiment 8 - Sample 2**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
49	65.000	19.000	2.023	49	70.000	4.000	1.656
50	57.000	16.000	2.023	50	59.000	3.000	1.656
51	44.000	17.000	2.023	51	50.000	4.000	1.656
52	58.000	37.000	-	52	58.000	37.000	-

Table 5.38: Order and coordinates of all visited points by MC and speed of MC in the best solution of Sample 2 in Experiments 6 and 8

Experiment 6 - Sample 2				**Experiment 8 - Sample 2**			
Path of the MC				**Path of the MC**			
Nodes	**X Coord**	**Y Coord**	**MC's Speed**	**Nodes**	**X Coord**	**Y Coord**	**MC's Speed**
0	58.000	37.000	1.633	0	58.000	37.000	1.633
9	59.048	42.113	1.633	8	58.980	45.262	1.633
18	45.230	56.498	1.633	18	48.020	57.980	1.633
25	40.123	54.797	1.633	25	44.147	55.965	1.633
29	38.440	45.510	1.633	29	43.866	49.015	1.633
35	49.204	32.349	1.633	36	58.296	32.975	1.633
43	58.186	36.212	1.633	44	61.937	32.594	1.633
52	58.000	37.000	-	52	58.000	37.000	-

The results for Sample 2 in Experiment 6 and Experiment 8 are provided in Tables 5.35, 5.36, 5.37, and 5.38. These results include the order and coordinates of the points forming the drone's route and the velocities between these points. In Experiment 6, the S^{Total} is 398 seconds, while in Experiment 8, it is 434 seconds. A lower Total System Time in Experiment 6 implies that the drone performs its operations faster. Therefore, it should move at a higher velocity during operations. However, this is clearly seen from the table. Under a broader system time frame, the drone traveled at a velocity closer to the optimal speed of 1.633. In Experiment 6, on the other hand, it moved at a faster velocity under a more constrained system time frame. Consequently, in Experiment 8, the total energy consumption for Sample 2 is lower compared to Experiment 6.

Let us examine the effects of energy consumption parameters on the results using visualizations, as demonstrated in Section 5.2.2, to observe these effects more comprehensively.

Figure 5.10 presents the routes obtained from experiments with different parameter sets for the same problem instance. Figures 5.10a-b display the routes obtained from experiments using Parameter Set 1, while Figures 5.10c-d show the routes obtained from experiments using Parameter Set 2. In Figures 5.10a-c, experiments with Value 1 were conducted for S^{Total}, while in Figures 5.10b-d, experiments with Value 2 were conducted.

Each experiment has a different S^{Total} value, but this does not hinder the interpretation of the figures based on the parameter sets.

In Figures 5.10c-d, where Parameter Set 2 with higher values is used, it can be observed that the route of MC is considerably shorter compared to Figures 5.10a-b. From these visualizations, an increase in the parameter values of MC leads to a shorter route.

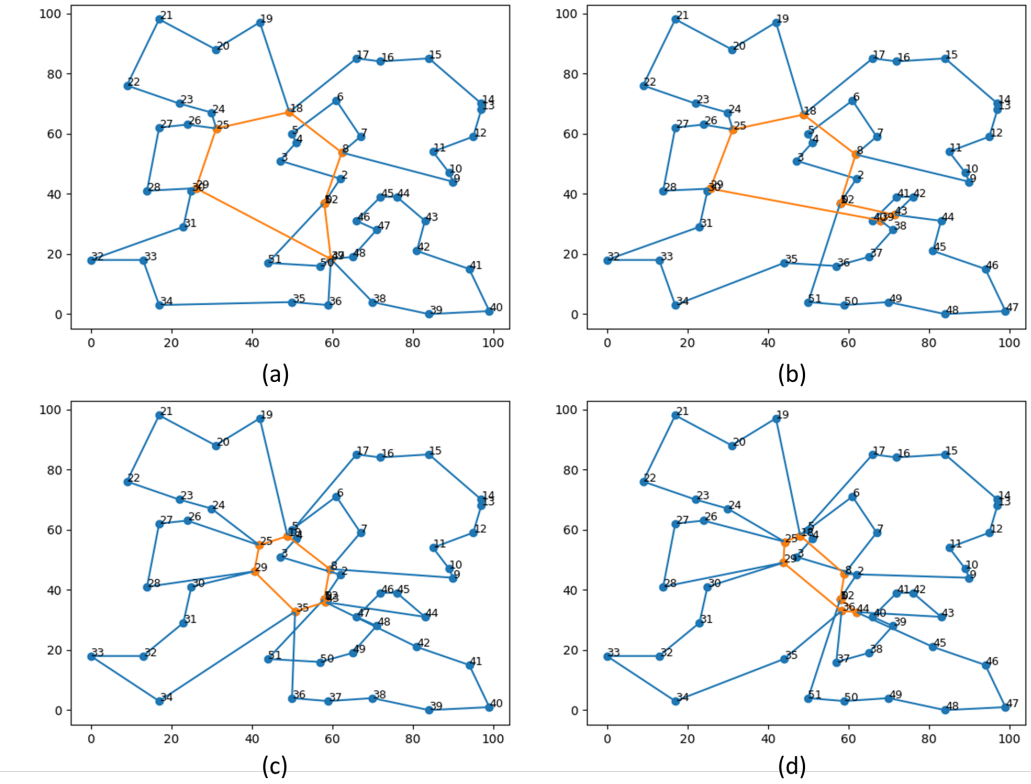

Figure 5.10: Illustrations of routes of Sample 3's solutions in Experiments 10, 12, 14, and 16 (a) route of Sample 3's solution in Experiment 10, b) route of Sample 3's solution in Experiment 12, c) route of Sample 3's solution in Experiment 14 and d) route of Sample 3's solution in Experiment 16)

The route of MC may vary for different S^{Total} values. In scenarios where S^{Total} is high, we encounter a more flexible constraint, allowing more time for the drone's travel. Therefore, there are scenarios where the route of MC can be reduced. Indeed, when comparing Figures 5.10b-d to Figures 5.10a-c, it can be observed that the route of MC is slightly smaller.

5.3.3.3 Discussion of the Results of Experiments with Instance size 60

Up to this section, we conducted studies on relatively smaller instances in Sections 5.1, 5.2, 5.3.3.1, and 5.3.3.2. To examine larger instances' solution times and outcomes, we performed experiments using a 60-point instance, as described in Table

5.17 in Section 5.3.1. In previous experiments, we extensively discussed the effects of Factors A, B, C, and D. In these experiments, we will focus on the influence of instance size and experimental setups on the results and computational time. Finally, we will analyze the routes and speeds of drones and MCs based on the results obtained from the system containing 60 service points. Through illustrations, we will examine the different routes obtained from these experiments.

First, let us examine the results and computational times of the experiments conducted with the 60-point instance. According to the findings discussed in Section 5.3.1, experiments with the "With Speed Up" level yielded better results among the setup variations based on algorithmic differences. However, no distinction could be made between the "Do Not Stop" and "Stop" levels. When looking at the results in Table 5.39, it can be observed that the best results for Parameter Set 1 were obtained in Experiments 33, 34, 37, and 38. The common characteristic of these experiments is that they were conducted using setups that include the "With Speed Up" level. Experiments 33 and 34 were performed with setups using the "Stop" level, while Experiments 37 and 38 were conducted using the "Do Not Stop" level. As can be seen, there is no significant difference among the results. Therefore, we cannot conclude that one of these levels is better than the other.

Similarly, for Parameter Set 2, the best results are obtained in Experiments 35 and 39. Looking at the common characteristic of these two experiments, both utilize Factor A with the "With Speed Up" level. The difference is that Experiment 35 uses Factor B with the "Stop" level, while Experiment 39 uses the "Do Not Stop" level. The same conclusion can be drawn by examining the results among the experiments using Parameter Set 2.

When making inferences based on computational times, the durations of experiments conducted with the "Without Speed Up" level will not be considered. This is because the computational times of these experiments are normally extremely high, so a time limit of 2 hours has been set for them. The solution times of all experiments conducted with the "Without Speed Up" level are slightly over 2 hours and very close to each other. Therefore, any time-related conclusions are made solely based on the solution times of experiments conducted with the "With Speed Up" level.

Table 5.39: Results and computational times of experiments with instance size 60

Experiments	Best OFV (Joule)	Number of Iterations	Computational Times (sec)
Experiment33	689.354	2	2874.031
Experiment34	688.981	3	3316.888
Experiment35	910.784	7	3887.313
Experiment36	923.999	5	3576.516
Experiment37	689.354	2	6528.341
Experiment38	688.859	3	8028.945
Experiment39	910.784	5	7691.769
Experiment40	961.430	6	7342.749
Experiment41	757.926	0	7318.190
Experiment42	783.365	0	7335.799
Experiment43	1093.094	0	7319.100
Experiment44	1056.697	0	7326.016
Experiment45	753.708	0	7300.016
Experiment46	751.990	0	7229.575
Experiment47	921.066	0	7237.227
Experiment48	922.742	0	7285.377

According to Table 5.39, the solution times vary significantly. Some experiments solve in 2874 seconds, while others take over 8000 seconds. The experiments with the highest computational times are Experiment 38 and Experiment 39. Experiment 38 is conducted with Parameter Set 1, while Experiment 39 is conducted with Parameter Set 2. These two experiments yield the best results for Parameter Set 1 and Parameter Set 2, respectively. According to these results, the experiments with the longest durations, as indicated in Table 5.39, have achieved the best outcomes.

The computational times of experiments conducted using instances with instance sizes of 30, 45, and 60, along with the setups provided in Table 5.40, are given in Table 5.41. Since the experiments conducted with the "Without Speed Up" level of Factor A have a time limit of 2 hours, only the durations of experiments conducted

with the setups using the "With Speed Up" level from Table 5.40 are examined.

These tables will analyze the impact of increasing instance size on computational times.

Table 5.40: All Setups with With Speed Up Level of Factor A

SETUPS	**LEVELS**			
Setup 1	A - Lv1	B - Lv1	C - Lv1	D - Lv1
Setup 2	A - Lv1	B - Lv1	C - Lv1	D - Lv2
Setup 3	A - Lv1	B - Lv1	C - Lv2	D - Lv1
Setup 4	A - Lv1	B - Lv1	C - Lv2	D - Lv2
Setup 5	A - Lv1	B - Lv2	C - Lv1	D - Lv1
Setup 6	A - Lv1	B - Lv2	C - Lv1	D - Lv2
Setup 7	A - Lv1	B - Lv2	C - Lv2	D - Lv1
Setup 8	A - Lv1	B - Lv2	C - Lv2	D - Lv2

Table 5.41: All computational times according to setups for all samples grouped by instance size

	SETUPS	**Setup 1**	**Setup 2**	**Setup 3**	**Setup 4**	**Setup 5**	**Setup 6**	**Setup 7**	**Setup 8**
Instance size 30	**Sample1**	601.17	382.81	342.36	400.15	484.86	517.37	412.15	475.24
	Sample2	1095.07	1030.33	1113.61	1016.82	2029.02	2183.15	1715.23	1771.68
	Sample3	729.16	337.50	332.94	334.02	1296.70	384.14	338.96	349.72
	Sample4	606.67	589.31	565.80	468.66	1038.57	1122.44	984.17	517.98
	Sample5	1733.09	1665.88	1869.63	1854.83	1755.58	1911.62	1230.94	1307.30
Instance size 45	**Sample1**	975.72	1078.96	1880.04	2467.93	1773.77	1805.15	1574.54	1744.80
	Sample2	1066.23	962.36	902.99	1192.68	1293.91	1244.06	1321.50	1516.67
	Sample3	537.75	843.19	529.67	1002.36	684.67	954.78	748.11	1174.13
	Sample4	560.58	812.89	747.66	792.77	606.63	742.55	742.35	788.38
	Sample5	305.53	324.75	290.33	314.30	592.09	598.58	627.83	672.01
Instance size 60	**Sample1**	2874.03	3316.89	3887.31	3576.52	6528.34	8028.94	7691.77	7342.75

When examining the results separately based on instance size, it can be observed that they reach solutions at significantly different durations. In the experiments conducted with Setup 1 - Instance size 30, Sample 1 solves in 601.17 seconds, while Sample 5 solves in 1733.23 seconds. Similarly, when examining the experiments conducted with Setup 8 - Instance size 30, it is clear that there is a more significant gap. Sample 3 completes in 349.72 seconds, while Sample 2 completes in 1771.68 seconds. When looking at the durations of experiments conducted with different setups for the same sample, it can be seen that the experiments also vary significantly in completion times. For example, for Sample 2 with a size of 30, Setup 1 solves in 1095.07 seconds, while Setup 6 reaches a solution in 2183.15 seconds for the same sample.

Overall, there is a relationship between setups and computational times. Setup 5 and Setup 6 generally have higher computational times than other setups. Similarly, Setup 2 and Setup 3 tend to have lower computational times than other setups. There is also a clear connection between samples and computational times. When considering instances with a instance size of 30, the computational time for experiments conducted with Sample 5 is considerably higher in all setups, while Sample 1 consistently has the lowest computational time.

The same conclusions can be made when examining the results based on instances with a instance size of 45.

When examining the results among instance sizes with different sizes, it can be observed that no clear inference can be made. When looking at the results among instances with a instance size of 45, it is evident that Sample 5 has consistently lower solution times across all setups. The lowest computational time of 290.33 seconds was achieved in Instance size 45 - Sample 5 - Setup 3. The highest computational time of 2467.93 seconds was observed in Instance size 45 - Sample 1 - Setup 4. Thus, when analyzing the results between instance sizes of 30 and 45, it can be concluded that these two sizes have no significant impact on computational time.

In conclusion, it is evident that samples significantly influence computational time regardless of their size. From these experiments conducted with randomly generated samples, it can be deduced that different samples have different improvement paths, and the time to reach a solution increases based on the extent of improvement

achieved. Similarly, with different setups, the computational time varies depending on the length of the improvement path. Therefore, the solution times are directly associated with the specific improvement path that is achieved, which is determined by the generated sample and the improvements provided by the chosen setups.

In experiments with a instance size of 60, we can observe a significant increase in the duration compared to experiments with instance sizes of 30 and 45. Since the durations of experiments with instance sizes of 30 and 45 vary significantly among themselves, it is not very meaningful to compare the average durations calculated based on setups with the durations of experiments with a instance size of 60 directly. However, when we individually compare the durations of them calculated based on setups with the durations of the experiments with a instance size of 60, we can clearly see that an increase in instance size significantly impacts the duration.

It can be observed that the durations of experiments with a instance size of 60 vary greatly depending on the setups. Experiments conducted with Setup 1, 2, 3, and 4 tend to be completed in shorter durations, while experiments conducted with Setup 5, 6, 7, and 8 take longer. From this observation, we can infer that the "Stop after 1^{st} improvement" level of Factor B yields quicker results as the instance size increases.

We examined the results obtained for the random instance with 60 service points in Table 5.39. Due to the length of the tables, we will focus on only two experiments. These experiments are 45 and 48. The results are provided in Tables 5.42, 5.43, 5.44 and 5.45.

Table 5.42: Order and coordinates of all visited points by drone and speed of drone in the solution for instance with instance size 60 in Experiments 45 and 48

Experiment 45				**Experiment 48**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
0	48.000	64.000	-	0	48.000	64.000	-
1	48.000	64.000	1.788	1	48.000	64.000	1.633
2	59.000	67.000	1.788	2	59.000	67.000	1.633
3	59.000	66.000	1.788	3	59.000	66.000	1.633
4	66.000	64.000	1.788	4	66.000	64.000	1.633
5	74.000	58.000	1.788	5	74.000	58.000	1.633
6	78.000	66.000	1.788	6	78.000	66.000	1.633
7	81.000	72.000	1.788	7	81.000	72.000	1.633
8	86.000	75.000	1.788	8	86.000	75.000	1.633
9	98.000	70.000	1.788	9	98.000	70.000	1.633
10	97.000	71.000	1.788	10	97.000	71.000	1.633
11	93.000	69.000	1.788	11	96.000	69.000	1.633
12	88.000	56.000	1.788	12	88.000	56.000	1.633
13	83.000	56.000	1.788	13	83.000	56.000	1.633
14	78.000	53.000	1.788	14	78.000	53.000	1.633
15	77.000	49.000	1.788	15	61.434	42.522	1.633
16	74.036	46.358	1.788	16	77.000	49.000	1.633
17	81.000	47.000	1.788	17	81.000	47.000	1.633
18	80.000	42.000	1.788	18	80.000	42.000	1.633
19	94.000	6.000	1.788	19	94.000	6.000	1.633
20	100.000	10.000	1.788	20	100.000	10.000	1.633
21	100.000	28.000	1.788	21	100.000	28.000	1.633
22	88.000	27.000	1.788	22	88.000	27.000	1.633
23	78.000	31.000	1.788	23	78.000	31.000	1.633

Table 5.43: Order and coordinates of all visited points by drone and speed of drone in the solution for instance with instance size 60 in Experiments 45 and 48 (continuation of Table 5.42)

Experiment 45				**Experiment 48**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
24	71.644	31.095	1.788	24	61.360	47.306	1.633
25	73.000	27.000	1.788	25	73.000	27.000	1.633
26	71.000	11.000	1.788	26	71.000	11.000	1.633
27	58.000	7.000	1.788	27	58.000	7.000	1.633
28	52.426	21.571	1.788	28	47.000	12.000	1.633
29	47.000	12.000	1.788	29	24.000	2.000	1.633
30	24.000	2.000	1.788	30	38.078	44.595	1.633
31	9.000	2.000	1.788	31	9.000	2.000	1.633
32	0.000	14.000	1.788	32	0.000	14.000	1.633
33	11.000	16.000	1.788	33	11.000	16.000	1.633
34	22.000	15.000	1.788	34	22.000	15.000	1.633
35	22.000	19.000	1.788	35	22.000	19.000	1.633
36	15.000	24.000	1.788	36	15.000	24.000	1.633
37	34.032	34.442	1.759	37	38.078	44.595	1.633
38	7.000	31.000	1.759	38	21.000	76.000	1.633
39	1.000	46.000	1.759	39	1.000	46.000	1.633
40	12.000	63.000	1.759	40	12.000	63.000	1.633
41	11.000	66.000	1.759	41	11.000	66.000	1.633
42	21.000	76.000	1.759	42	21.000	76.000	1.633
43	27.000	68.000	1.759	43	27.000	68.000	1.633
44	41.000	69.000	1.759	44	41.000	69.000	1.633
45	36.000	63.000	1.759	45	36.000	63.000	1.633
46	31.000	51.000	1.759	46	37.822	49.617	1.633

Table 5.44: Order and coordinates of all visited points by drone and speed of drone in the solution for instance with instance size 60 in Experiments 45 and 48 (continuation of Table 5.43)

Experiment 45				**Experiment 48**			
Path of the Drone				**Path of the Drone**			
Nodes	**X Coord**	**Y Coord**	**Drone's Speed**	**Nodes**	**X Coord**	**Y Coord**	**Drone's Speed**
47	37.469	43.288	1.788	47	32.000	51.000	1.633
48	45.000	29.000	1.788	48	37.822	49.617	1.633
49	56.000	29.000	1.788	49	54.000	29.000	1.633
50	60.000	32.000	1.788	50	56.000	29.000	1.633
51	65.000	50.000	1.788	51	60.000	32.000	1.633
52	74.000	75.000	1.788	52	65.000	50.000	1.633
53	78.000	79.000	1.788	53	48.000	64.000	1.633
54	67.000	77.000	1.788	54	74.000	75.000	1.633
55	51.950	68.378	1.788	55	78.000	79.000	1.633
56	67.000	90.000	1.788	56	67.000	77.000	1.633
57	66.000	91.000	1.788	57	67.000	90.000	1.633
58	61.000	99.000	1.788	58	66.000	91.000	1.633
59	71.000	100.000	1.788	59	61.000	99.000	1.633
60	90.000	98.000	1.788	60	71.000	100.000	1.633
61	98.000	99.000	1.788	61	48.000	64.000	1.633
62	51.981	68.378	1.788	62	90.000	98.000	1.633
63	48.000	88.000	1.788	63	98.000	99.000	1.633
64	31.000	91.000	1.788	64	48.000	64.000	1.633
65	30.000	99.000	1.788	65	48.000	88.000	1.633
66	11.000	88.000	1.788	66	31.000	91.000	1.633
67	5.000	91.000	1.788	67	30.000	99.000	1.633
68	48.000	64.000	-	68	11.000	88.000	1.633
				69	5.000	91.000	1.633
				70	48.000	64.000	-

Table 5.45: Order and coordinates of all visited points by MC and speed of MC in the solution for instance with instance size 60 in Experiments 45 and 48

Experiment 45				**Experiment 48**			
Path of the MC				**Path of the MC**			
Nodes	**X Coord**	**Y Coord**	**MC's Speed**	**Nodes**	**X Coord**	**Y Coord**	**MC's Speed**
0	48.000	64.000	1.633	0	48.000	64.000	1.633
16	74.036	43.358	1.633	15	61.434	47.522	1.633
24	71.644	31.095	1.633	24	61.360	47.306	1.633
28	52.426	21.571	1.633	30	38.078	44.595	1.633
37	34.032	34.442	1.633	37	38.078	44.595	1.633
47	37.469	43.288	1.633	46	37.822	49.617	1.633
55	51.951	68.378	1.633	48	37.822	49.617	1.633
62	51.951	68.378	1.633	53	48.000	64.000	1.633
68	48.000	64.000	-	61	48.000	64.000	1.633
				64	48.000	64.000	1.633
				70	48.000	64.000	-

Experiment 45 has the following experimental setup: Factor A: Without Speed Up, Factor B: Do Not Stop after 1^{st} improvement, Factor C: Parameter Set 1 and Factor D: Value 1 of S^{Total}. Experiment 48, on the other hand, has the following experimental setup: Factor A: Without Speed Up, Factor B: Do Not Stop after 1^{st} improvement, Factor C: Parameter Set 2 and Factor D: Value 2 of S^{Total}.

In Experiment 48, the drone and MC travel at optimal speeds, while in Experiment 45, the drone travels faster than the optimal speed. This is because the drone visits an additional rendezvous points in its route, which requires more time. The reason for this is that in Experiment 45, it is conducted with the small one of the S^{Total} values. Therefore, the drone aims to finish the operation within the given time and it is faster during the operation. Due to its speed, it consumes more energy and requires more charging. Because of this, the drone meet more with MC.

It can be observed that solutions obtained from setups with different levels for Fac-

tors A and B have significantly different routes. The solution obtained with the "With Speed Up" level is better. Figure 5.11 illustrates the routes of the best and worst results obtained among the experiments conducted for Parameter Set 1. In Figure 5.11b, the integrated route has more rendezvous points than Figure 5.11a. As a result, it requires more movement, and the drone needs to accelerate more, leading to increased energy consumption.

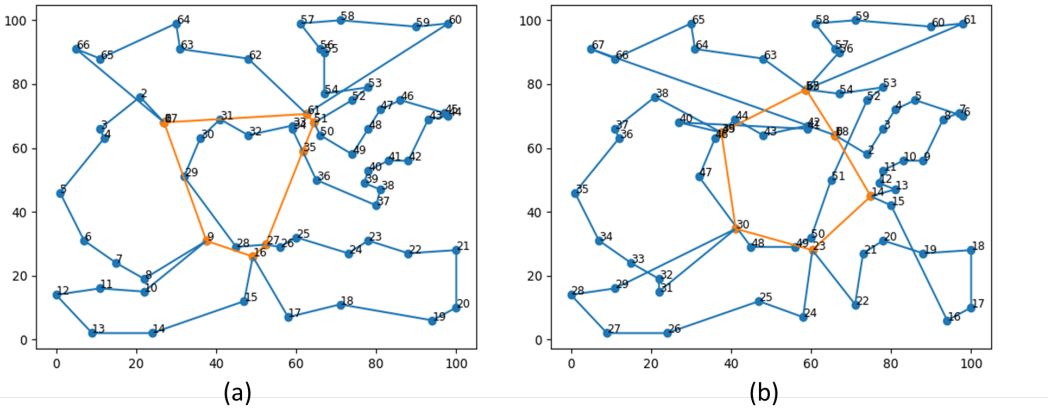

Figure 5.11: Illustration of the best route solution for instance with instance size 60 in Experiments 38 and 42 (a) The result of Experiment 38 which is the best; b) the result of Experiment 42 which is the worst.)

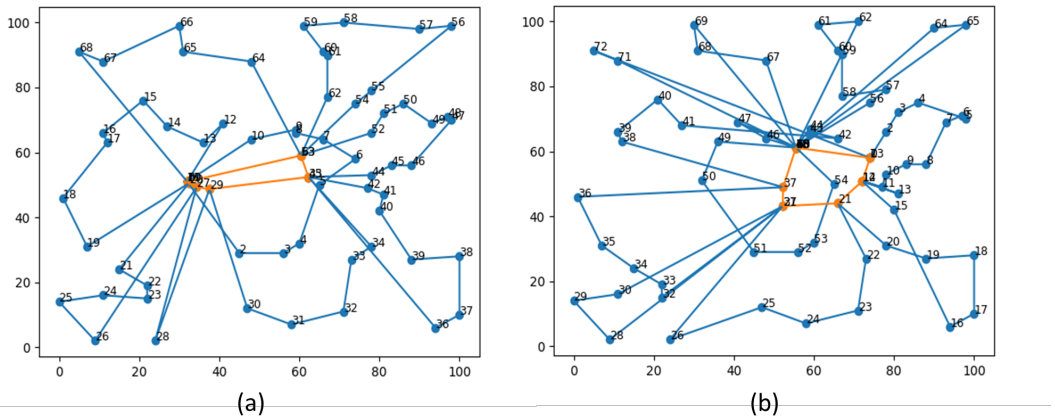

Figure 5.12: Illustration of the best route solution for instance with instance size 60 in Experiments 35 and 43 (a) The result of Experiment 35 which is the best; b) the result of Experiment 43 which is the worst.)

Figure 5.12 illustrates the routes of the best and worst results obtained among the experiments conducted for Parameter Set 2. When examining these routes, it can be observed that increasing the parameter values of the MC leads to a shorter route. The routes in Figure 5.12 are more complex and longer than those in Figure 5.11. This is because as the MC's route becomes shorter, the drone needs to move more within the same time period, resulting in faster movement. This increased movement speed of the drone leads to higher energy consumption and therefore requires more charging. Consequently, it implies that the MC will have more rendezvous points in their routes, indicating more meets with the drone.

CHAPTER 6

CONCLUSION

This thesis focuses on energy-efficient, time-constrained, and simultaneous routing of drones and mobile charging stations. The outcome of this study provides routing solutions for fixed-payload or payload-free drone operations, such as military operations, surveillance flights in traffic, ecological research, and reconnaissance flights for road construction. Instead of drones obtaining their energy supply from fixed charging stations or specific fixed centers during operations, the aim is to determine suitable routes for mobile charging stations and drones to acquire energy from mobile charging stations within the same operational area. This allows drones to meet and charge with a mobile charging station between visits to service points without interrupting their operations under time constraints and without drastically altering their routes. In our research, a heuristic model has been developed to find an integrated route for drones and mobile charging stations. This model calculates the necessary speed for vehicles to travel in order to minimize energy consumption on the route while completing the service within the restricted system time. To solve the Energy Efficient Simultaneous Drone and Mobile Charger Routing Problem with Time Restrictions, a mat-heuristic model is created by integrating different heuristics and an SOCP-based mathematical model. The heuristics determine the drone's route and the order of rendezvous points, while the SOCP-based mathematical model calculates the locations of rendezvous points to form the integrated route of the drone and mobile charging station. The best possible route for the problem is searched through this matheuristic.

In this study, two models with different objectives are developed. The first objective is to minimize the total energy consumption of the vehicles. The second objective is

minimizing the total system time while minimizing energy consumption. Under the first objective, the model is designed with Total System Time being one of the most critical constraints. The drone is expected to complete the operation within the given time frame. Under the second objective, the model aims to minimize energy consumption while completing the operation in the shortest possible time. The output of this model is of great importance in operations where time is crucial, such as military operations, as it enables the completion of the operation in the shortest time possible.

The illustrative example in this study demonstrates a significant improvement of 41.3% between the initial feasible solution found by the model and the final solution. Preliminary experiments examined how Total System Time and energy consumption change. According to the results obtained from these experiments, as the time constraint of the system is relaxed, the drone and mobile charging station can travel at an optimal speed to minimize energy consumption. Therefore, as Total System Time increases, the total energy consumption decreases. Additionally, the effect of the energy consumption parameter values of the mobile charging station on the route was examined. Different parameter values were experimented with, and it was observed that as the energy consumption of the mobile charging station increases, its route becomes smaller while the drone's route becomes larger.

In order to examine the results of our study, various experiments were designed. These experiments aimed to investigate the effects of different setups by modifying the algorithms used in the heuristics and parameter values on the results. The algorithmic differences include running the 2-Opt Algorithm with or without the Speed Up Method and determining the best route among the routes obtained by the 2-Opt Algorithm either by moving to the next iteration after the 1^{st} improvement or by thoroughly testing all alternative routes after the 1^{st} improvement. Regarding the parameters, the impact of different energy consumption parameters for drones and mobile charging stations, as well as different Total System Times, were examined.

The experiments showed that using the "Speed Up Method" led to obtaining good results in a shorter period of time. Due to the significant increase in time without using Speed Up Method, a time constraint of 2 hours was imposed. When comparing the results obtained within this constrained time with the experiments that utilized the

Speed Up Method, it was observed that the use of the Speed Up Method positively impacted the results. On the other hand, no significant difference was observed between stopping or continuing after the 1^{st} improvement. It was clearly evident that the energy consumption parameters had a significant effect on the route. Different energy consumption values were measured in experiments conducted under different Total System Times. As the given time for the drone to complete the operation increased, it was able to move at a speed closer to the optimal speed that minimizes energy consumption. Therefore, it was observed that energy consumption was lower in experiments where Total System Time was higher.

In the experiments conducted with sample sizes of 30, 45, and 60, it was observed that as the dimension increased, the computational time also increased rapidly. While the durations were relatively low in the experiments with sample sizes of 30 and 45, it was seen that the duration significantly increased in the experiments with a sample size of 60.

This study addresses the simultaneous routing of drone and mobile charging station under a specific time constraint, aiming to minimize energy consumption, which has yet to be previously explored at the same time in the literature. One of the significant contributions brought by this research is the calculation of the speeds at which drone and mobile charging station need to operate on this route in order to minimize energy while completing the operation within the given time frame.

This study used the Nearest Neighbor algorithm to determine the initial routes. We focused on the 2-Opt Algorithm to improve the routes. The study was conducted with a single drone and mobile charging station on a 2D plane without obstacles. In future research, different initial route construction algorithms and improvement heuristics, such as 3-Opt, can be used to improve the solutions. By increasing the number of drones and mobile charging stations, more complex problems can be studied for larger systems. Our study has paved the way for future comprehensive research encompassing broader concepts.

REFERENCES

[1] E. Frachtenberg, "Practical drone delivery," *Computer*, vol. 52, no. 12, pp. 53–57, 2019.

[2] T. Benarbia and K. Kyamakya, "A literature review of drone-based package delivery logistics systems and their implementation feasibility," *Sustainability*, vol. 14, no. 1, p. 360, 2022.

[3] P. Maini and P. Sujit, "On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1370–1377, IEEE, 2015.

[4] D. Sun, X. Peng, R. Qiu, and Y. Huang, "The traveling salesman problem: Route planning of recharging station-assisted drone delivery," in *Proceedings of the Fourteenth International Conference on Management Science and Engineering Management: Volume 2*, pp. 13–23, Springer, 2021.

[5] E. E. Yurek and H. C. Ozmutlu, "Traveling salesman problem with drone under recharging policy," *Computer Communications*, vol. 179, pp. 35–49, 2021.

[6] 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.

[7] G. C. Crişan and E. Nechita, "On a cooperative truck-and-drone delivery system," *Procedia Computer Science*, vol. 159, pp. 38–47, 2019.

[8] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[9] S. Park, L. Zhang, and S. Chakraborty, "Battery assignment and scheduling for drone delivery businesses," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6, IEEE, 2017.

[10] S. Sawadsitang, D. Niyato, P. S. Tan, and P. Wang, "Supplier cooperation in drone delivery," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, IEEE, 2018.

[11] K. Yu, A. K. Budhiraja, and P. Tokekar, "Algorithms for routing of unmanned aerial vehicles with mobile recharging stations," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 5720–5725, IEEE, 2018.

[12] O. Dukkanci, B. Kara, and T. Bektaş, "Minimizing energy and cost in range-limited drone deliveries with speed optimization," *Transportation Research Part C: Emerging Technologies*, vol. 125, p. 102985, 04 2021.

[13] P. Maini, K. Sundar, M. Singh, S. Rathinam, and P. Sujit, "Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3016–3028, 2019.

[14] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal velocity profiles for car-like robots," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1457–1462, 2011.

[15] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical programming*, vol. 95, no. 1, pp. 3–51, 2003.

[16] C. Nilsson, "Heuristics for the traveling salesman problem," *Linkoping University*, vol. 38, pp. 00085–9, 2003.

[17] K. Steiglitz, P. Weiner, and D. Kleitman, "The design of minimum-cost survivable networks," *IEEE Transactions on Circuit Theory*, vol. 16, no. 4, pp. 455–460, 1969.