VEHICLE ROUTING FOR AERIAL SURVEILLANCE WITH A HOMOGENEOUS FLEET

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

BY

KORAY TARAKÇI

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

DECEMBER 2021

Approval of the thesis:

VEHICLE ROUTING FOR AERIAL SURVEILLANCE WITH A HOMOGENEOUS FLEET

submitted by **KORAY TARAKÇI** in partial fulfillment of the requirements for the degree of **Master of Science** in **Industrial Engineering, Middle East Technical University** by,

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

Prof. Dr. Esra Karasakal
Head of the Department, **Industrial Engineering**

Prof. Dr. Esra Karasakal
Supervisor, **Industrial Engineering, METU**

Prof. Dr. Orhan Karasakal
Co-Supervisor, **Industrial Engineering, Çankaya University**

Examining Committee Members:

Assoc. Prof. Dr. Sedef Meral
Industrial Engineering, METU

Prof. Dr. Esra Karasakal
Industrial Engineering, METU

Assist. Prof. Dr. Özgen Karaer
Industrial Engineering, METU

Assoc. Prof. Dr. İsmail S. Bakal
Industrial Engineering, METU

Assoc. Prof. Dr. M. Alp Ertem
Industrial Engineering, Çankaya University

Date: 10.12.2021

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 Last name : Koray Tarakçı

Signature :

ABSTRACT

VEHICLE ROUTING FOR AERIAL SURVEILLANCE WITH A HOMOGENEOUS FLEET

Tarakçı, Koray
Master of Science, Industrial Engineering
Supervisor: Prof. Dr. Esra Karasakal
Co-Supervisor: Prof. Dr. Orhan Karasakal

December 2021, 87 pages

In this study, we develop models and solution approaches for planning the surveillance mission of a homogeneous fleet of Unmanned Aerial Vehicles (UAVs). Predefined areas are to be observed while satisfying a minimum probability of target detection. Areas are assumed to be rectangular and discrete. UAVs with electro-optical sensors take off from a base and fly through predefined routes. The endurance of UAVs is limited by the maximum flight distance. The proposed models minimize the total travel distance of UAVs to meet the mission requirements. Each UAV starts its tour from the base and, after performing its mission over one or more areas, returns to the base. We developed a mathematical model to solve the route planning for UAVs. Since the problem is NP-Hard, we propose two constructive heuristic algorithms. In the proposed solution approach, we initially transform our problem into an m-TSP problem. Heuristic algorithms start with m-TSP solution and improve the solution to reach the best feasible one. An extensive computational study on the problems taken from the literature shows that the proposed heuristics produce efficient solutions in a reasonable time.

Keywords: UAV Mission Planning, Vehicle Routing, Algorithms

ÖZ

HAVADAN GÖZETLEME İÇİN HOMOJEN BİR FİLO İLE ARAÇ ROTALAMA

Tarakçı, Koray
Yüksek Lisans, Endüstri Mühendisliği
Tez Yöneticisi: Prof. Dr. Esra Karasakal
Ortak Tez Yöneticisi: Prof. Dr. Orhan Karasakal

Aralık 2021, 87 sayfa

Bu çalışmada, özdeş İnsansız Hava Araçlarının (İHA) gözetleme ve keşif amacıyla gerçekleştirdikleri görevlerin planlanması için model ve çözüm yöntemleri geliştirdik. Dikdörtgen şeklinde ve kesikli olduğu varsayılan hedef sahalarının belirli bir hedef tespiti oranını sağlayacak şekilde taranması istenmektedir. Sensörlerle donatılmış birden çok özdeş İHA, bir üsten başlayarak hedeflere belirli bir rotada hareket eder. Her bir İHA, belirli bir maksimum uçuş menzilini geçemeyecek şekilde sınırlandırılmıştır. Problemde amaç, her İHA'lar için oluşturulan turların toplam mesafesini en küçüklemektir. Hedefteki tarama sürecinin sonlanması sonrasında rota planına uygun şekilde ya başka bir dikdörtgen sahaya giderek tura devam eder, ya da üsse giderek turunu sonlandırır. Problemin çözümü için yeni bir matematiksel model geliştirdik. Matematiksel model Np-Hard olduğu için, iki yeni sezgisel algoritma önerdik. Söz konusu problem m-TSP yapısına çevrilmiştir. Bu algoritmalar m-TSP sonucunu başlangıç sonucu olarak ele alır ve sonucu iyileştirirler. Çalışmanın sonucunda makul sürelerde verimli çözümler sunuyoruz.

Anahtar Kelimeler: İHA Görev Planlaması, Araç Rotalama, Algoritmalar

To my precious family and my beloved girlfriend

ACKNOWLEDGMENTS

I’d like to express my deepest gratitude to my supervisor Prof. Dr. Esra Karasakal, and co-supervisor, Prof. Dr. Orhan Karasakal for their guidance, advice, constructive criticism, encouragement, and insight throughout the research. They shared their valuable time, unmeasurable support, and precious knowledge openhandedly in the current pandemic era. If only there were enough words to convey the immense gratitude and indebtedness I carry in me for them.

I would also like to thank Dr. Barış Keçeci for his suggestions and comments. Not only about this particular study, but also about all the technical and ethical advices. From the beginning until today, he has made a series of contributions that had such a positive influence on me.

The technical assistance of Mr. Mengü Nazlı, Mr. Furkan Taylan Tarakçı, and Ms. Melis Boran is gratefully acknowledged. When all the hard times and lack of sufficient sources are reminisced, their support surely made a great difference and deserves endless amount of praise.

Also, throughout the study, the emotional support of friends and the people along the way is worth mentioning in this part. Especially, Ms. İrem Çelebi shared her patience and love unconditionally.

Last but not least, I would like to give my most delicate appreciation and love to my family and loved ones. Their love and assistance were the very thing that was crucial to accomplish this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

An Uncrewed Aerial Vehicle (UAV) is an air vehicle which does not carry a pilot and is operated autonomously or remotely by a human operator. Depending on the field of operation, a UAV is regarded as expandable or recoverable.

Drones invented in the 1840s as balloon carriers, used for offensive purposes in military, are the first examples of UAVs (Buckley 2003). During World War I era, Hewitt-Sperry Automatic Airplane is produced; an unmanned airplane filled with explosives, using gyroscopes to maintain balance (Sahm V. H. & Werrell 1987).

Then radio controlled pilotless aircrafts emerge. Between 1927 and 1936, the British Royal Navy and the United States Navy organize experiments for radio controlled aircrafts. Succesful experiments provoke the use of radio control to increase and the term drone begins to be used for such aircrafts (Howeth 1963).

In the early World War II era, the first large scale aircraft production is commenced. Afterward, the Radioplane Company is established, and various models are designed. The production and utilization of UAVs accelerates and they are used in World War II (Naughton 2005).

Today, the concept of UAV is still improving and ever-evolving both in terms of technical developments and popularity.

UAVs are used in various fields with several different roles. Their purpose is diversified according to the need, and vehicles serve one or more objectives.

UAVs are used for:

- Military combat,
- As baits to reveal the targets,
- Surveillance and observation,
- Search and rescue missions,
- Map drawing,
- Logistic operations.

Vehicles are classified according to their purposes, technical capabilities and features. Classification criteria is based on weight, maximum achievable altitude, degree of autonomy and maximum achievable distance. When a UAV is to be used for a mission, its technical capabilities must be considered carefully in order to choose the most suitable one for the requirements of that mission.

UAVs, adopted into several fields, are highly demanded aircrafts. However, this study solely focuses on surveillance and observation missions. Surveillance and observation are significant elements, primarily for military purposes. Such missions are organized for intelligence gathering, maintaining border security and locating targets. Thus, multiple missions are coordinated in a short time period consistently. Depending on the mission, it may have to be repeated multiple times. Naturally, these missions are costly due to fuel consumption, maintenance and organizational factors. While it is vital to protect and maintain national security, it is also essential the operation is executed efficiently. Therefore, mission planning is the blueprint for successfully lowering total cost and productively using existing resources. Creating feasible solutions in a reasonable period of time is a challenge to for these missions. This thesis aspires to provide satisfying solutions that will eliminate or minimize unused capacity at hand.

In this study, observation of the predefined rectangular areas is discussed via the proposed model. Identical UAVs equipped with electro-optical sensors take off from a base station, and visit rectangular areas without violating the distance capacity limits. The UAVs surveil the regions with strips and return to the base station after all rectangles are visited. The objective is the minimization of total distance taken by UAVs.

To be able to solve the problem, a new mathematical model is proposed. Since the problem is NP-Hard, two alternative heuristic algorithms are introduced. One of the heuristics uses an efficient TSP solver, and the other uses a neighborhood search approach.

To the best of our knowledge, the UAV mission planning concept is studied with different extensions in the literature. Metaheuristic algorithms are adopted in order to provide reasonable solutions generally. Also, some of the produced heuristics are exclusive to the problem at hand. This study contributes to the literature by developing unique algorithms and providing an extension to a new problem.

The organization of this thesis as follows. In chapter 2, literature review on the UAV Surveillance Mission Planning is given. In chapter 3, the problem is explained in detail, and the mathematical model is introduced. Two heuristic solution algorithms and their application to the problem are described in chapter 4. The computational results are presented in chapter 5. This study concludes with a discussion on the conclusions and the future research in chapter 6.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the development of UAV surveillance mission planning is examined, and research findings are shared in four subsections. These are single UAV, multiple UAVs, swarm and mixed vehicles mission planning. The findings are explained in chronological order. Further, a classification table of articles in the literature is shared in Table 2.1. Finally, the last section of this chapter focuses on the articles that inspired this study. The contribution of this thesis and its place in the literature is also mentioned.

A study of taxonomy on UAV routing and trajectory optimization problems (UAVRTOP) in Coutinho, Battarra, and Fliege (2018) is considerably useful and valuable for this thesis to examine the literature. In the study, a review of UAVRTOP is shared and UAV literature is explained while seventy articles are examined. UAV articles are classified with respect to the topic of problems. These are trajectory optimization, path planning, routing, and task assignment. In these topics, different kinds of solution methods are introduced. Mainly Mixed Integer Linear Programing (MILP), rapidly-exploring random trees, metaheuristics and heuristics are used to reach the solutions. The least used methods are simulation, queue theory, enumeration and geometric methods.

2.1 Single UAV Mission Planning

According to our research, the first study for aerial surveillance planning is Panton and Elbers (1999). Aerial vehicles with high-resolution cameras are used by Defense

Sciences and Technology Organization (DSTO) to take pictures of twenty rectangular areas (targets) called swaths in different sizes.

The vehicle flies near the swaths in four different patterns and collects the pictures. First, possible tracks are listed visually between a start base and an end base. Among these tracks, a sequence of targets is selected manually. This method is highly time-consuming and cannot guarantee optimality. Optimization software tools that include an Integer Programming (IP) model similar to Travelling Salesman Problem (TSP) are designed in Panton and Elbers (1999). The objective is to minimize the total distance traveled by aerial vehicle. Compared to former procedure, the results are found to be satisfactory. The model solves twenty rectangles within seconds.

In the following year, the same problem of Panton and Elbers (1999) is named the Regional Surveillance (RS) problem and studied in John, Panton, and White (2001). A metaheuristic procedure as an alternative solution technique is suggested. Three genetic algorithm models called Regional Surveillance Genetic Algorithm (RSGA) are produced. The methods use the steady state without duplicates crossover technique. If a new child is born out of a crossover and is not unique, it is then replaced with the parent with the lowest fitting value. Three RSGA models are compared with the IP model, and the results show that a significant amount of CPU time is consumed by the IP model. Nonetheless, even if the RSGA solutions are sub-optimal, they are reasonable according to CPU time.

A multi-objective study is investigated in Tezcaner and Köksalan (2011). A generalized multi-objective TSP (MOTSP) is defined to decide the route of UAV search missions. The scenario of the problem is that a UAV visits a set of targets and arrives back to base. The problem is composed of two different sub-problems. Firstly, arcs to be followed between each target pair are determined by solving the shortest path problem (SPP). Later, the order of targets to be visited must be considered. Thus, it is handled by solving a MOTSP model. Finally, a Pareto optimal set is generated where an interactive approach helps find an appropriate solution for the decision-maker (DM) under the assumption that the utility function is linear.

UAV route planning problems under refueling constraints are investigated in Sundar and Rathinam (2012). One UAV must visit all targets without violating the fuel capacity. When fuel is not enough to reach the next target, the vessel visits the closest refueling area. A MILP formulation is produced. For complex problems, construction and improvement type heuristics are developed. Heuristics outputs are compared with the optimal solutions, which proves the heuristics to be fast and efficient.

Frequently, objective functions concentrate on distance minimization. Nevertheless, new variants of objective functions emerge while the literature diversifies. A new model based on the search and rescue problem is published in Guitoni and Masri (2013). The focus is a single air vehicle restricted by the maximum flight time. The objective is to maximize the detection probability of targets on the assumption that target has a limited amount of time to stay alive. The model uses a probability map of containment which is a result of an information fusion process. Based on the historical data and the expert judgments, this process identifies possible areas where target may be situated. A map is divided into small sections (i.e., cells) and the probability of detection values are calculated for cells. Relying on an orienteering problem structure, the total probability of target detection is maximized by the model.

In Babel (2017), an aerial surveillance problem with obstacles in which an aerial vehicle follows curvature paths is studied. An aerial vehicle must visit a set of landmarks and avoid obstacles on the map. The vehicle uses curve paths to fly to the landmark without passing through the obstacles. In the meantime, the vessel is assumed to have a constant speed, cannot move backward, and turn with minimum radius. Algorithms which add different landmarks to the network and optimize paths using a discrete routing model are offered.

In Vasquez-Gomez et al. (2020), the focus is on a two-dimensional path coverage problem with a single UAV. In this study, the concentration is on starting and ending point of flight route. An efficient path planner is recommended, and the output is compared with the output of previous methods. The path planner generates same or better solutions.

Finally, a recent study acknowledges UAVs in a way to minimize their carbon footprint. Yi, Sutrisna, and Wang (2021) optimizes the velocity of an aerial vehicle at certain areas of an already-constituted path. The goal is to sustain adequate velocity values in certain areas of the path to ensure effective usage of drone's battery.

2.2 Multiple UAV Mission Planning

Mission planning for multiple UAVs is extensively studied in surveillance mission planning. In NG and Ghanmi (2002), a Decision Support System (DSS) is developed to help find and follow unauthorized vessels that approach Canadian coasts. Insufficient information about the vessels makes the problem more complex. The system helps decision-makers find the sequence of scanned areas while maximizing the probability of target detection. The system consists of three sub-systems. The first sub-system aims to create a probability map of illegal vessels. While the area is divided into cells, the approximate location information is assumed to be obtained from intelligence reports. Thus, the vessels' possible location in a predetermined amount of time is obtained with its probabilities. Considering the probability map, if the vessel is close to the coast, sub-system three suggests barrier patrol missions. Otherwise, sub-system two administers the way of scanning the regular and irregular-shaped areas.

Some problems in literature refer to missions as tasks. A task allocation problem for a UAV fleet, introduced in Alighanbari, Kuwata, and How (2003), is an example to this. Multiple types of vehicles are used in the problem and, while some of the tasks can be satisfied by only one type of vehicle, some other tasks can be satisfied by multiple vehicle types. The problem demands strict time limitations for the tasks. In order to comply with such strict demands, the loitering concept is adopted, which is the act of flying of a vehicle to spend a reasonable amount of time on the mission area to wait for another type of UAV. Additionally, there are no-fly zones in which vehicles are restricted in the considered area. For small-scale problems, a MILP model is developed. Alternatively, an approximate decomposition solution procedure to overcome complexity is created. A tabu search method for larger instances of the problem is proposed.

Reconnaissance, surveillance, search and rescue missions with a limited number of vehicles are studied in Jacobson et al. (2006). These vehicles have different characteristics. The problem can be solved in the same manner as that of the Multiple Travelling Salesman Problem (M-TSP), and the related problem is solved by generalized hill climbing (GHC) and simultaneous generalized hill climbing (SGHC) algorithms.

In Simonin, Le Cadre, and Dambreville (2009), a multilayered problem is defined. An area search problem for moving targets with multiple sensors is solved. The problem is divided into two as upper and lower levels. The objective of the problem is to maximize the probability of locating a hidden target in a large. First, assigning the sensors to the zones is attempted by the model, which is defined as a 2D assignment problem. Then in the lower level, the model performs resource distribution for the search areas to maximize the probability of target detection.

A new variant of the min-max vehicle routing problem is defined in Yakıcı and Karasakal (2012). Demands of various types of customers are met via a heterogeneous fleet of vehicles. The problem is defined under the assumption that vehicles ensure one or more service types, maintain unlimited service capacity, have variable delivery and service times, and allow split delivery of services. For larger problems, a heuristic approach is introduced.

Two metaheuristic procedures for autonomous UAV path planning are compared in Robergo, Tarbouchi, and Labonté (2013). A cost function, which evaluates dynamic features and limitations of a UAV such as path length, fuel consumption and altitude of the vehicle, is introduced to use with the metaheuristic procedures. First, a Genetic

Algorithm (GA) is introduced with four crossover operators. As for the second metaheuristic, a Particle Swarm Optimization (PSO) model is suggested and both metaheuristics are solved using parallel computing. Computational runs show that GA is superior to PSO.

Multiple UAV routing for minimum time area coverage is analyzed in Avellar et al. (2015). A group of UAVs with image sensors cover an area. The solution consists of two sub-methods: First, a graph-based method is used to determine how a single UAV covers an area. Second, the output of the graph-based technique is used in a MILP formula which composes the UAV fleet’s route. It is quite similar to Vehicle Routing Problem (VRP), however some features of the problem are decided during task execution; such as the number of vehicles to be utilized to accomplish the task.

A data collection and transfer problem is addressed in Macharet et al. (2017). A Wireless Sensor Network (WSN) contains sensors in nodes. Due to limited communication range, a mobile sink node travels within sensors and collects the accumulated data. The sink node behaves like a dubins vehicle which turns with curve patterns. When length and time of collection is considered, efficient routes are created. The bi-objective problem is solved by Nondominated Sorting Genetic Algorithm (NSGA)-II, a Multi Objective Evolutionary Algorithm (MOEA).

A heterogeneous UAV fleet routing problem is presented in Coelho et al. (2017). Concept of the problem is the demand satisfaction with UAVs. Packages are expected to be delivered via UAVs while considering battery life and weight capacity of vessels. The model focuses on several minimization type objective functions: such as total distance, time spent on a package, number of UAVs used, maximum velocity and charging requirements. A MILP formulation is proposed. A matheuristic based solution technique with the weighted sum method to find a Pareto optimal is set to tackle the complexity.

A MILP model is created in Alotaibi et al. (2017) which examines optimal routes for multiple UAVs under threat exposure and flight time limits while maximizing the total number of visited targets. For each edge, flight time and threat level are calculated. Several methods are used to generate waypoints in order to reduce the high risk of threat. These obtained candidate waypoints are later used to optimize the UAV routes with branch and cut and price (BCP) algorithm, supported by minimum dependent sets and a basic path heuristic.

A surveillance mission planning problem where multiple aerial vehicles gather information by covering areas of interest in an extended time horizon is promoted in Wang et al. (2018). The two objectives of the problem are defined as the maximization of the minimum number of non-repeatedly covered cells and the maximization of the total number of covered cells. A MOEA method is proposed. A specific chromosome representation that uses three mutation operators is generated and a weighted sum method is used to aid the decision-making process.

Multiple UAV routing problems are investigated in Zhen et al. (2019). Areas of interest with different accuracy requirements are monitored by UAVs in a sequence. While monitoring, a UAV at a specific height related to accuracy level and the service time is required for each area. UAVs are capacitated by maximum flight time and the objective is to minimize total flight time. An integer linear programming model and a tabu search metaheuristic approach are used to solve the problem.

A MILP model that examines multi-period, multi-vehicle coverage routing, and scheduling is defined in Zuo et al. (2020). The problem is defined as persistent surveillance which means visited areas are revisited within a certain interval. Zero aerial vehicle collusions and assaults during flight are assumed while the altitude of vehicles is constant. In the meantime, air vehicles can take off and land multiple times. All air vehicles are limited by a maximum operation time. Also, fuel constraints are defined, which limits the flight time. In order to extend the flight time restricted by fuel consumption, the aerial vehicles must visit a base node to refuel. The air vehicles must visit areas of interest (AOI), and these areas consist of points of interest (POI). Two conflicting objective functions are reviewed in the problem—the simultaneous maximization of coverage area which is the number of visited POI and coverage time which means the time spent in an AOI. The problem is overseen by considering both POI paths and AOI paths separately. First, the POI path planning problem is solved by a heuristic method. Then, search patterns and paths in each AOI are constructed. Finally, the AOI path planning problem with generated search patterns and paths is solved by a MILP solver.

Recent articles disclose the aspect of battery life. The study of Li et al. (2020) is based on WSN to master detecting or monitoring tasks. According to it, sweep coverage is an essential element for WSNs and the idea of using UAVs as sensor nodes is recommended. Bearing the limits of battery life in mind, multiple UAVs are dispatched, aiming to achieve maximum sweep coverage in minimum time. A mathematical problem is established, and a new heuristic algorithm, called weighted targets sweep coverage, is presented.

Another study is presented in Kiam, Besada-Portas, and Schulte (2021). Solar-powered High Altitude Pseudo-Satellites (HAPSs) are found suitable to use multiple UAVs. A new planner is put forward based on MOEA, which operates the mission plan taking weather conditions into account. The GA-optimizer is tested under past weather data and realistic mission settings.

2.3 Swarm Mission Planning

Up to this point, single and multiple vessels are discussed as assets, yet there are some studies worth mentioning that use drone swarms.

The first example of a swarm of UAVs is assembled in Lamont, Slear, and Melendez (2007). A comprehensive UAV swarm mission planning system is established and swarms of autonomous aerial vehicles are formed with two objectives: cost and risk of the given path. The cost of a path is the total time of the used path and fuel consumption. Whereas the other objective consists of terrain, detect and kill formulations. A path planning algorithm is submitted, and the output is given as an input to a MOEA. The metaheuristic objectives minimize the encompassing distance traveled, the amount of climbing a vehicle realizes, and the risk resulting from threat areas.

Pohl and Lamont (2008) study continues with the swarms and presents a new kind of problem called the Swarm Routing Problem (SRP). It provides mission routing of multiple autonomous UAVs. It is stated that UAVs move as swarms on the battlefield. During the combat, some of the vessels must meet enemy targets right on time. Moreover, the swarms can divide into sub-groups and visit the related targets. After the combat, the subgroups merge and continue the mission. A MOEA is developed as a solution method. Since the problem is a variant of the Vehicle Routing Problem with Time Windows (VRPTW), the benchmark problems are converted to SRP to measure the capability of MOEA.

2.4 Mixed Vehicles Mission Planning

A study of maritime routing is conducted in Grob (2006), in which a mathematical formulation is given for maritime surface surveillance routing problems for different vehicles such as helicopters, frigates, and maritime patrol aircrafts. It is implied that the proposed problem is related to the on-line TSP (OLTSP), a TSP on a dynamic network where the network changes in time. Six different algorithms are suggested to handle the problem. One of the defined heuristics is the nearest neighbor rule which is a greedy method. The task that takes the least amount of time is executed first. Another heuristic (n, k) does not consider a single vessel, instead it considers n vessels, calculates the shortest sequence of the tasks, and executes the first k of the sequence. The score per time rule and (n, k) score per time rule are similar to the former two heuristics. Rather than consumption of time of the task, importance score is used. Hence, the score per time rule demonstrates effective solutions with a reasonable computation time.

Refueling problem is solved in Maini et al. (2019) with a ground vehicle (GV) which is used as a fueling station for a single UAV. A two-stage strategy generates the coupled route. While the first stage generates feasible sites for both UAV and GV, the second stage optimizes a route for both vehicles with a MILP formulation. Also, a heuristic solution is proposed.

Same problem is approached in Fesenko et al. (2020) too and it is settled by planting automated aerial battery replacing systems and locating them in specified places. According to the case, a UAV is expected to visit eleven monitoring stations and gather data. This route is fabricated by solving the TSP. Locations of the route points are determined via an algorithm in order to place the battery replacement systems. Therefore, both the route of the UAV and the system are obtained.

Unmanned aerial and ground vehicles in urban environments for persistent surveillance are analyzed in Wu, Wu, and Hu (2021). UAVs perform circular coverings for specific areas. If UAVs cannot complete the tasks, unmanned ground vehicles (UAG) step in to resume. As for the solution, estimation of distribution algorithm (EDA) and GA are integrated into a heuristic procedure. From this point on, some of the studies that inspire the relevant thesis statement are described.

Instead of UAV types, a risk assessment method for military surveillance applications is explored in Buyurgan and Lehlou (2015). The method is expected to improve the effectiveness of UAV used in combat and surveillance missions. Five risk factors are formulated while nine geographical parameters are taken into account. These parameters are foliage, distance, population, altitude, line of sight, slope, trafficability, past events and experience. The risk factors are foliage camouflage, proximity to population, long-range attacks, accessibility to regions, input from experts. By using these factors, risk values for relevant terrains are calculated.

Table 2.1 Literature classification according to the number of vehicles and vehicle types.

Article	Single UAV	Multiple UAVs	Swarms	Mixed Vehicles
Panton and Elbers (1999)	X			
John, Panton, and White (2001)	X			
NG and Ghanmi (2002)		X		
Alighanbari, Kuwata, and How (2003)		X		
Grob (2006)				X
Jacobson, McLay, Hall, Henderson, and Vaughan (2006)		X		
Lamont, Slear, and Melendez (2007)			X	
Pohl and Lamont (2008)			X	
Simonin, Le Cadre, and Dambreville (2009)		X		
Tezcaner and Köksalan (2011)	X			
Sundar and Rathinam (2012)	X			
Yakıcı and Karasakal (2012)		X		
Robergo, Tarbouchi, and Labonté (2013)		X		
Guitoni and Masri (2013)	X			
Avellar, Pereira, Pimenta, and Iscold (2015)		X		
Buyurgan and Lehlou (2015)	X	X	X	X
Macharet, Monteiro, Mateus, and Campos (2017)		X		
Coelho, Coelho, Coelho, Ochi, Haghnazar, Zuidema, Lima, and da Costa (2017)		X		
Alotaibi, Rosenberger, Mattingly, Punugu, and Visoldilokpun (2017)		X		
Babel (2017)	X			
Coutinho, Battarra, and Fliege (2018)	X	X	X	X
Wang, Kirubarajan, and Tharmarasa, Jassemi-Zargani and Kashyap (2018)		X		
Zhen, Li, Laporte, and Wang (2019)		X		
Vasquez-Gomez, Marciano-Melchor, Valentin, and Herrera-Lozada (2019)	X			
Maini, Sundar, Singh, Rathinam, and Sujit (2019)				X
Zuo, Tharmarasa, Jassemi-Zargani, Kashyap, Thiyagalingam, and Kirubarajan (2020)		X		
Li, Xiong, She, and Wu (2020)		X		
Fesenko, Kliushnikov, Kharchenko, Rudakov, and Odarushchenko (2020)				X
Yi, Sutrisna, and Wang (2021)	X			
Kiam, Besada-Portas, and Schulte (2021)		X		
Wu, Wu, and Hu (2021)				X
Ng and Sancho (2009)	X			
Karasakal (2016)	X			
Karasakal, Karasakal, and Maraş (2020)	X			

In the rest of this section, the Aerial Surveillance Problem (ASP) is explained, and the branches of ASP are examined.

ASP is introduced in Ng and Sancho (2009). Regional surveillance is studied with the aim of planning the search sequence of a set of given areas and the inner search pattern while minimizing the distance of the mission route. The mission plan must be limited to a specified mission duration, and the selected route must satisfy a minimum probability of target detection. The regions are assumed to be in the shape of rectangles or squares and areas to be searched have different exit and entry points. As a result, each and every combination of entry and exit points differs concerning distance and time. There is a correlation between these two metrics. While distance decreases, the probability of detection decreases as expected. The problem is considered as a variant of classical TSP. As a solution method, a dynamic programming procedure is developed.

Karasakal (2016), proposes efficient solution methods for ASP. A new formulation for ASP is introduced in this study. One UAV must take off from a base and surveil target areas in stripes to gather intelligence. The goal is to find the minimum distance tours while reassuring the target detection probability constraint. It is claimed that the formulation provides more efficient use than older versions and also a new model based on TSP formulation is introduced. While a more advanced formulation is presented, a max-min version of ASP is put forward to maximize the probability of minimum target detection.

The bi-objective ASP is studied in Karasakal, Karasakal, and Maraş (2020). A multi-objective mission planning model is defined using two conflicting objectives of Karasakal (2016). Multi-objective ASP (MASP) is solved with the Ɛ-constraint method to generate the whole Pareto optimal set of the related problem. Also, heuristic methods are constructed to tackle more complex instances. Different initial solutions are generated by construction methods and an efficient solution set is found by the improvement method using the initial solution. Appropriate solutions for missions are demanded to be decided by a decision-maker. An interactive solution procedure is introduced so DM is not lost in enormous solution sets.

In this thesis, an extension to ASP is introduced. Karasakal (2016) studies distance minimization while taking target detection probability for single UAV into consideration. The same objective is adopted in this thesis to handle the multiple UAV case of ASP. Karasakal (2016) makes it possible for target detection probability to be actively decided by the model, whereas the new problem in this thesis solely focuses on distance values. Hence, this extension is a contribution to multiple UAV mission planning, and it is placed under ASP.

The proposed problem has resemblance Capacitated Vehicle Routing Problem (CVRP). The objective of the CVRP is to generate minimum cost routes to a fleet or a set of vehicles where all vehicles are bounded by a capacity limit. In order to deliberate further on it, some of the recent studies for CVRP are mentioned below.

One of the latest studies is conducted in Altabeeb et al. (2021). CVRP is tackled using cooperative hybrid firefly algorithms (CHFA) with multiple firefly algorithm populations. In this method, the solutions are assumed as fireflies. A local search strategy and genetic operators are used by algorithms. Multiple firefly population, and genetic operators provide diversity and help the method avoid local optima.

Another study is introduced in Sitek Wikarek et al. (2021). The proposed problem is a mixture of various VRP variants such as CVRP, VRP with Pickup and Delivery (VRPPD), and VRP with Time Windows (VRPTW). The main difference of the related model from older variants is that alternative delivery points and parcel lockers are submitted by the model and inserted into the existing network. In order to obtain solutions, a Binary Integer Programing (BIP) model is created. Alternatively, a hybrid approach is used which integrates Constraint Programing (CP), GA and mathematical programing.

For further information about CVRP and variants of VRP, the taxonomy of Breakers, Ramaekers, and Van Nieuwenhuyse (2016) may be examined.

CHAPTER 3

PROBLEM DEFINITION AND MATHEMATICAL MODEL

ASP is defined in Karasakal (2016). One UAV takes off from a base node and flies through the predefined regions, which are assumed to be rectangular or square. The areas are searched by the UAV equipped with electro-optical sensors, and the vessel uses preplanned search along strips, called patterns. Karasakal, Karasakal, and Maraş (2020) proposes MASP by combining two objective functions from Karasakal (2016).

A more detailed problem definition and its features about ASP and MASP are presented in Section 3.1. In Section 3.2, it is focused on the study of this thesis, which is called Aerial Surveillance Problem with a Homogenous Fleet (ASP-H).

3.1 Aerial Surveillance Problem: Problem Definition and the Existing Models

In this section, mathematical models for ASP are examined and in the subsections, Minisum ASP, Maximin ASP, and MASP are explained respectively.

Some of the features are mutual for all three problems. The ASP models are similar to well-known TSP (Flood 1956). In TSP, a salesman starts from a node, visits a set of nodes, and returns to the starting node. ASP adopts the same behavior. ASP has no nodes to visit, but rectangular region. UAV flies over the rectangular region for surveillance systemeticaly.

In ASP, one UAV equipped with electro optic or electronic sensors takes off from a base and is sent to predefined regions for surveillance. When the UAV reaches a target region, it must enter the region and surveil it in strips. The region is assumed to be a rectangle or a square, although in real-life experience, the region may not be necessarily rectangular. In that case, the region to be observed can be fit into a rectangle that is wide enough to cover the area but also tight enough to avoid unnecessary long pattern distances.

There can be irregular regions, for example, a region in the shape of a line. Instead of placing it into a wide rectangular region, it can be divided into small rectangles. Consequently, UAVs avoid unnecessary travel.

Each rectangular region has eight notional entry/exit points placed on the sides of each edge of the region. The entry/exit points are shown in Figure 3.1. One of the eight points is used to enter the region. Depending on the entry point, the UAV departs the area from one of two possible exit points. After the UAV is search over the area, it flies to the next area or to the base and lands if all the predefined target areas are already visited.

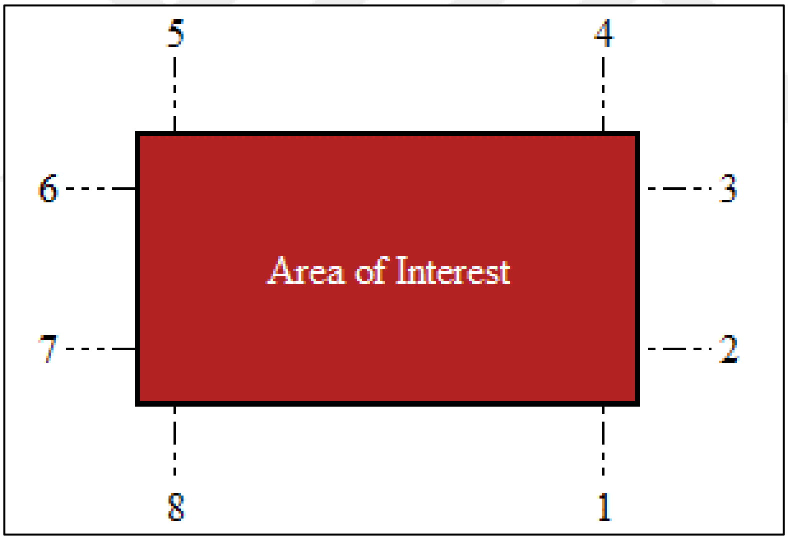


Figure 3.1. An Illustration of eight notional entry/exit points for a rectangular area.

A notional point is used to enter the area and then a strip is followed. When the opposite side of the strip is reached, UAV performs two ninety-degree turns and changes its direction. Turning distances are ignored in the model.

When a UAV reaches a target area, it enters the area using an entry point. The exit point is determined according to the pattern followed. If the pattern consists of an even number of strips, UAV leaves the area from the same side of the rectangle but in the opposite direction of the entry. Otherwise, it departs in the same direction of the entry but on the opposite side of the rectangle. The visual illustration of the entry and the exit points is shown in Figure 3.2.

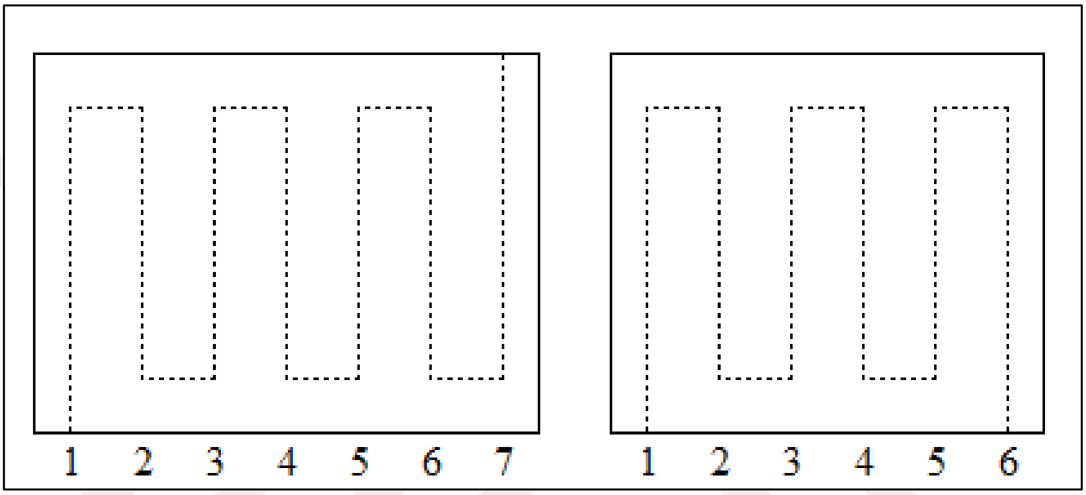


Figure 3.2. An illustration of the change of exit and entry point regarding the number of strips.

The patterns of target areas guarantee a certain amount of minimum target detection probability. There is a correlation between the minimum target detection probability and the distance of the related pattern. As the pattern distance increases, more area is scanned by the UAV and more time is spent inside the target. Thus, the minimum probability of target detection increases.

The calculation of target detection probability in the rectangles is based on Koopman's area search equation (Wagner, 1999). In ASP, the targets are assumed to be steady and uniformly distributed over the area. Since that is the case, equation (1) is used to calculating the values.

$$P = 1 - e^{-\frac{w}{s}} \quad (1)$$

In Equation (1), P is the probability of target detection where w is the sweep width, and s is the track spacing while the target is uniformly distributed over the area.

Sweep width is an aggregated measure for the electro-optical sensor's capability to detect targets. In other words, it is the surveillance range of the sensors. The number of targets not detected inside the sweep width is equal to the number of targets detected outside the sweep width. (Koester, Cooper, 2004). In Figure 3.3, an illustration of sweep width logic on a lateral range curve is shown. The sum of areas A and the sum of areas B are equal.

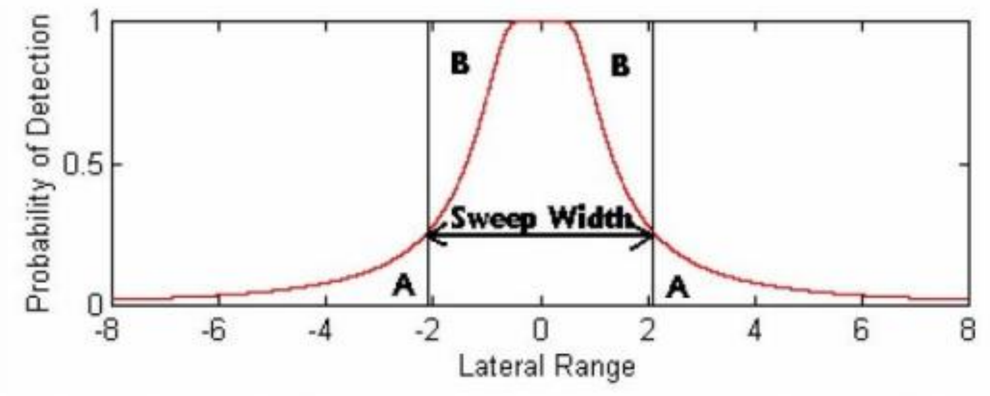


Figure 3.3. A graph shows the relation between lateral range and sweep width.

Track spacing is the width of a strip's area. Also, it defines the distance between two consecutive strips' centers, i.e., distance between the parallel flight paths of UAV.

In Karasakal (2016), both distance and target detection probability are taken into consideration. The models Minisum and Maximin ASP evaluates multiple patterns that use same entry and exit points. These patterns are created by changing track spacing values while sweep width remains constant. While track spacing decreases the target detection probability increases. In return, because track spacing value decreases, the number of strips increases for the relevant pattern. For this reason, the distance taken by air vehicles gets larger.

In this study, the target detection probability that must be ensured and the sweep width value remain constant. The shortest pattern that ensures the minimum target detection probability is generated for each exit and entry point combination.

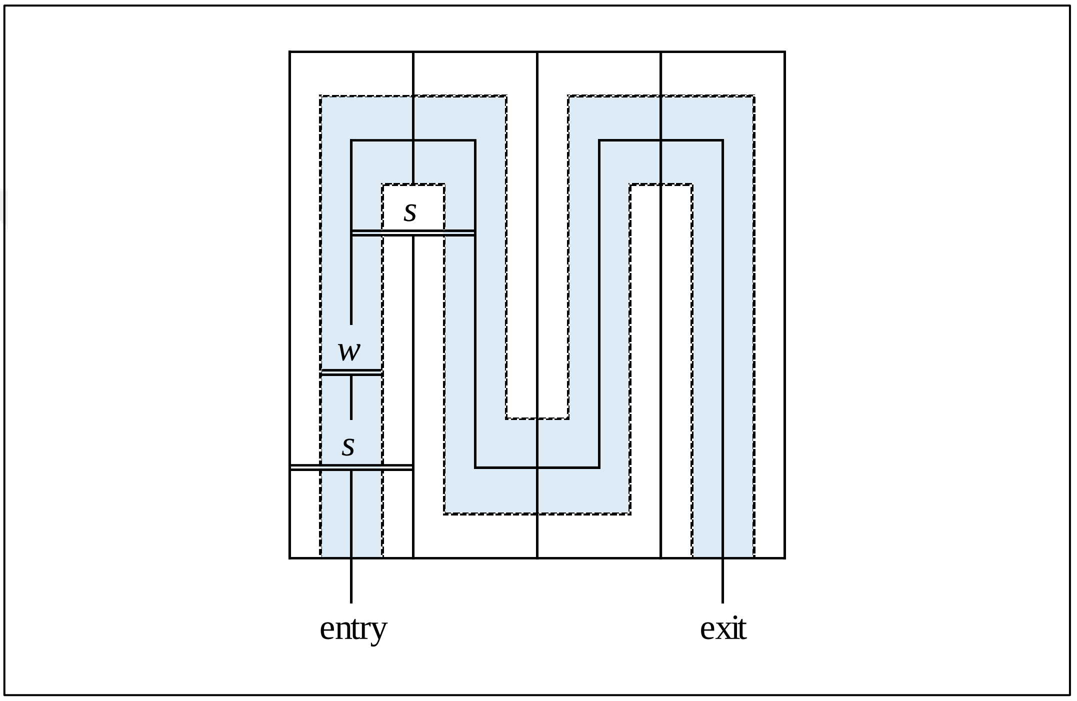


Figure 3.4. An illustration of w and s parameters on a rectangular area.

Assuming that there is a minimum target detection probability to be aimed, the track spacing must ensure the specified amount of probability by taking the following limitations into account:

- The spacing must be larger than or equal to the sweep width. ($s \geq w$).

- If the aircraft travels vertically with respect to coordinate system, the width of the rectangular region must be divisible by s.

- If the aircraft travels horizontally with respect to coordinate system, the length of the rectangular region must be divisible by s.

ASP variants such as Minisum ASP, Maximin ASP, and MASP calculate multiple patterns for one entry and exit point combination. Consequently, the model provides more option to surveil regions with different probability of target detection and distance values. However, ASP-H only minimizes the total distance taken by the fleet. ASP-H calculates shortest pattern that guarantees a constant target detection probability for each entry and exit point combination. Therefore, the model considers only the distance taken by UAVs.

3.1.1 Mathematical Model: Minisum ASP

Surveillance missions are costly because of high fuel consumption, long mission times, and maintenance charges. Therefore, reducing the mission time lowers the amount of consumed resources. Since such missions constantly repeat to ensure border control, the mission plans require optimization. Time is also a critical resource since the life of a person depends on the success of the mission. Minimization of the total distance serves both factors simultaneously.

Minisum ASP (Karasakal 2016) intends to find the shortest tour to realize the mission successfully. In the meantime, the patterns of the rectangular areas are calculated in a way to ensure the minimum probability of target detection. The mathematical model of Minisum ASP is described below: (Karasakal, 2016)

Indices

i, j	:	indices of disjoint areas, i =1 and j=1 represent the base station.
l, k	:	indices of entry and exit points of areas, l=1, … ,8 and k=1, … ,8.
s	:	index of the pattern within an area for each feasible combination of entry and exit points of the area.

Parameters

N	:	total number of disjoint areas and the base station.
D_{iljk}	:	distance between point *l* of area *i* to point *k* of area *j*.
D'_{ilks}	:	distance between point *l* to point *k* of area *i* using pattern s.
P_{ilks}	:	probability of detecting target by flying from point *l* to point *k* in area *i* using pattern *s*.

Decision Variables

$$y_{iljk} = \begin{cases} 1, \text{ if the UAV flies from point } l \text{ of area } i \text{ to point } k \text{ of area } j. \\ 0, \text{ otherwise.} \end{cases}$$

$$z_{ilks} = \begin{cases} 1, \text{ if the UAV flies from point } l \text{ to point } k \text{ of area } i \text{ using pattern } s. \\ 0, \text{ otherwise.} \end{cases}$$

u_i : node potential of area *i* that indicates the order of the corresponding area in the tour.

Model

(Minisum ASP)

$$\boldsymbol{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \tag{2}$$

Subject to

$$\sum_{ilk} y_{iljk} = 1 \qquad \forall j = 1, \dots, N \tag{3}$$

$$\sum_{jlk} y_{iljk} = 1 \qquad \forall i = 1, \dots, N \tag{4}$$

$$\sum_{lks} z_{ilks} = 1 \qquad \forall i = 2, \dots, N \tag{5}$$

$$\sum_{ks} z_{ilks} \le \sum_{jk} y_{jkil} \qquad \forall l = 1, \dots, 8, i = 2, \dots, N \tag{6}$$

$$\sum_{ls} z_{ilks} \le \sum_{jl} y_{ikjl} \qquad \forall k = 1, \dots, 8, i = 2, \dots, N \tag{7}$$

$$u_i - u_j + N\sum_{lk} y_{iljk} \leq N - 1 \qquad \forall i, j = 2, \dots, N, i \neq j \qquad (8)$$

$$u_i \geq 0 \qquad \forall i = 1, \dots, N \qquad (9)$$

$$y_{iljk} \in \{0,1\} \ \ \forall iljk \in \{(i, l, j, k) | i \neq j, i = 1 \wedge l = 1, j = 1 \wedge k = 1\} \qquad (10)$$

$$z_{ilks} \in \{0,1\} \ \ \forall ilks \in \{(i, l, k, s) | i \neq 1, l \neq k\} \qquad (11)$$

The objective function consists of two parts. The first part reflects the total distance taken between the rectangular areas, and the second part implies the distance traveled inside the target regions. Constraint set (3) guarantees that the air vehicle only leaves from one exit point from a rectangle or base. Similarly, Constraint set (4) guarantees that the air vehicle only enters the rectangle from one entry point to a rectangle or base. Constraint set (5) ensures that for each rectangle, only one pattern is used. Constraint sets (6) and (7) indicate that the rectangles' patterns match the exit and entry points used by the air vehicle. Constraint (8) is used for the sub tour elimination. Constraint (9) ensures to have positive sub tour elimination variables. The rest of the constraints (10) and (11) are the binary constraints for decision variables.

3.1.2 Mathematical Model: Maximin ASP

ASP can also be modified to maximize the efficiency of the concerning mission. The target may constitute great importance which eventually makes the success of the mission vital. In such cases, the minimum target detection probability is intended to be maximized by Maximin ASP. Since distance is not acknowledged by the objective, a distance capacity limit is applied to the model in pursuance of restraining the flight distance of the air vehicle with fuel consumption factor kept in sight. The mathematical model of Maximin ASP is described below (Karasakal, 2016):

M : maximum distance the UAV can fly.

(Maximin ASP)

$$\boldsymbol{Max}\ \alpha \tag{12}$$

Subject to

Constraints (3) − (11)

$$\sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \leq M \tag{13}$$

$$\sum_{ljk} P_{ilks} z_{ilks} \geq \alpha \qquad \forall i = 2, \dots, N \tag{14}$$

$$\alpha \geq 0 \tag{15}$$

The Maximin ASP employs all constraints from (3) to (11). Additionally, the objective function consists of a single variable. It is connected with the constraint set (14). Thus, the minimum target detection probability becomes the highest probability value. Constraint (13) limits the total distance traveled by the air vehicle. Finally, constraint (15) is a nonnegativity restriction for added α variable.

3.1.3 Mathematical Model: MASP

The same problem is viewed by MASP from a multi objective perspective. Two conflicting objectives of Minisum ASP and Maximin ASP are studied at once. MASP model is given below (Karasakal, Karasakal, and Maraş 2020):

(MASP)

$$\boldsymbol{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \tag{2}$$

$$\boldsymbol{Max}\ \alpha \tag{12}$$

Subject to

Constraints $(3) - (11)$

Constraints $(13) - (15)$

3.2 The Proposed Model: Aerial Surveillance for a Homogenous Fleet (ASP-H)

In Minisum ASP, only a single vessel is taken into consideration. Even though it is an efficient model, a fleet of UAVs demands a more complex model. Since that is the case, this study sides with the ASP-H model. As the area to be searched grows larger, accomplishing the mission with a single UAV becomes more challenging and even infeasible. The assumptions of the proposed model are listed below.

- The UAV fleet is identical, and each has the same flight distance limit.
- The mission plan is considered on a 2D plane. Therefore, the altitude of UAVs is ignored.
- The distance taken by the turn of air vehicles is ignored.
- The UAVs are assumed to take off and start the mission simultaneously.
- The collision risk of air vehicles is ignored.

The ASP-H model is given below:

Indices

t : index of the vehicle used in the mission.

Parameters

M : total number of UAVs.

C	:	maximum flight range of each UAV
D'_{ilk}	:	distance between point *l* to point *k* of area *i*.

Decision Variables

$$y_{iljkt} = \begin{cases} 1, \text{ if UAV } t \text{ flies from point } l \text{ of area } i \text{ to point } k \text{ of area } j. \\ 0, \text{ otherwise.} \end{cases}$$

$$z_{ilkt} = \begin{cases} 1, \text{ if UAV } t \text{ flies from point } l \text{ to point } k \text{ of area } i \text{ by a pattern.} \\ 0, \text{ otherwise.} \end{cases}$$

Model

$$\boldsymbol{Min} \sum_{iljkt} D_{iljk} y_{iljkt} + \sum_{ilkt} D'_{ilk} z_{ilkt} \tag{16}$$

Subject to

$$\sum_{j=2}^{N} \sum_{lkt} y_{1ljkt} = M \tag{17}$$

$$\sum_{i=2}^{N} \sum_{lkt} y_{il1kt} = M \tag{18}$$

$$\sum_{lkt} y_{1ljkt} \leq 1 \qquad \forall j, j \neq 1 \tag{19}$$

$$\sum_{j=2}^{N} \sum_{lk} y_{1ljkt} \leq 1 \qquad \forall t \tag{20}$$

$$\sum_{lkt} y_{il1kt} \leq 1 \qquad \forall i, i \neq 1 \tag{21}$$

$$\sum_{i=2}^{N} \sum_{lk} y_{il1kt} \leq 1 \qquad \forall t \tag{22}$$

$$\sum_{ilkt} y_{iljkt} = 1 \qquad \forall j, j \neq 1 \tag{23}$$

$$\sum_{jlkt} y_{iljkt} = 1 \qquad \forall i, i \neq 1 \tag{24}$$

$$u_i - u_j + (N-M)\sum_{lkt} y_{iljkt} \leq N-M-1 \qquad \forall i,j, i \neq j \neq 1 \qquad (25)$$

$$\sum_{lkt} z_{ilkt} = 1 \qquad \forall i, i \neq 1 \qquad (26)$$

$$\sum_{k} z_{ilkt} \leq \sum_{jk} y_{jkilt} \qquad \forall l,i,t, i \neq 1 \qquad (27)$$

$$\sum_{l} z_{ilkt} \leq \sum_{jl} y_{ikjlt} \qquad \forall k,i,t, i \neq 1 \qquad (28)$$

$$\sum_{iljk} D_{iljk} y_{iljkt} + \sum_{ilk} D'_{ilk} z_{ilkt} \leq C \qquad \forall t \qquad (29)$$

$$u_i \geq 0 \qquad \forall i \qquad (30)$$

$$y_{iljkt} \in \{0,1\} \;\; \forall iljkt \in \{(i,l,j,k,t) | i \neq j, i = 1 \wedge l = 1, j = 1 \wedge k = 1\} \qquad (31)$$

$$z_{ilkt} \in \{0,1\} \;\; \forall ilkt \in \{(i,l,k,t) | i \neq 1, l \neq k\} \qquad (32)$$

A route is created with ASP-H model by minimizing the total distance traveled by UAVs in regard to the distance capacity restrictions of UAVs. The model's first and second constraints (17) and (18) define the exits and entries to the base station, node 1. There are *M* air vehicles that take off from the base, and *M* air vehicles land at the base. Constraints (19) and (20) make sure assigning the take-off of different UAVs to avoid assigning one UAV to multiple take-offs. Constraints (21) and (22) apply the same logic to the landing process to avoid assigning multiple landings to a single UAV. Constraints (23) and (24) are typical assignment constraints that push the model to create tours to visit the rectangular areas. Constraint (23) ensures that only one UAV can reach a rectangular area using only one entry point from only one exit point of another rectangular area. Constraint (24) ensures that only one UAV can leave a rectangular area using only one exit point and reach another rectangle using only one entry point. Constraint (25) is sub tour elimination constraints (Miller, Tucker, and Zemlin, 1960). These constraints can sketch the outer tour. After that point, the model focuses on inner pattern assignment. Constraint (26) is the assignment of inner patterns of the rectangular areas. Each rectangle must be scanned with a specified pattern. Constraints (27) and (28) allow the model only to choose the patterns that use same entry and exit point with the outer tour defined by the same model. Therefore, the inner pattern’s entry and exit points are aligned with the outer tour's entry and exit points. For example, if the UAV enters the area through entry point number 1, patterns that start from other entry points must be eliminated. Distance restrictions for each air vehicle are defined by constraint (29). Constraint (30) ensures nonnegativity of sub tour elimination variables. Constraints (31) and (32) are binary constraints for decision variables. The objective function can be seen in Equation (16) which minimizes the total distance taken by UAVs.

CHAPTER 4

SOLUTION APPROACH

In this chapter, two heuristic algorithms are proposed for ASP-H. Since ASP-H is NP-Hard, one cannot solve any instance of the problem to optimality. Thus, efficient and fast solution approaches are designed and proposed.

The proposed algorithms use Concorde TSP solver (Applegate et al. 1998) to generate an initial solution. Some modifications are applied to the initial solution step by step to reach a good feasible solution. Concorde TSP Solver is described in Section 4.1 in order to depict the proposed solution approach and the way to construct a structure to apply the solver to ASP-H. In Section 4.2, heuristic algorithms are explained.

4.1 Concorde TSP Solver and Application to ASP-H

In this section, two subsections are introduced. The first subsection explains the Concorde TSP solver, and the second subsection describes the application of the Concorde solver to ASP-H.

4.1.1 Concorde TSP Solver

Concorde TSP solver is first introduced by Applegate et al. (1998) as an algorithm. It is written in ANSI C programing language, which is shared online for academic applications.

Concorde TSP solver uses the cutting plane method introduced by Dantzig, Fulkerson, and Johnson (1954). A relaxed TSP Linear Programming (LP) model is solved by the simplex algorithm, and a feasible solution is generated for that relaxed

TSP. This feasible solution is called a fractional tour since it is an optimal solution to the relaxed TSP. Then, elements of the fractional tour that cause infeasibility to the actual TSP structure are added to the relaxed TSP LP model as a constraint to avoid the current infeasibility. The relaxed TSP LP with additional constraints is solved again. This procedure continues until an optimal solution is reached for the actual TSP. In that process, the optimality is controlled due to weak and strong duality theorems. These additional constraints are called cutting planes, and the method is called the cutting plane method. The Concorde solver adopts the cutting plane method and makes it applicable to the problems that are symmetric TSP and similar problem structures.

The performance of the Concorde TSP solver is measured by solving TSPLIB instance library which consists of one hundred ten problems with various numbers of cities (Reinelt 1991). The computer has the following hardware features: a single processor of a dual-processor 2.8 GHz Intel Xeon PC with a 533 MHz front-side-bus and 2 gigabytes of RAM. The ILOG CPLEX (Version 6.5) linear programming solver is used in these computations. The results of Concorde solver are very promising. For example, it solves the instance pr2392 with 2392 nodes in 35.04 seconds.

The proposed algorithms use Concorde solver as a tool. For that reason, the ASP structure must be reflected as a symmetrical TSP. In the following subsection, this transformation is described.

4.1.2 Transformation of ASP-H to Symmetrical m-TSP

Concorde solver is a symmetric TSP solver. Thus, ASP-H must be defined in a symmetric m-TSP matrix form. A multilayered transformation procedure is applied to meet the demand of the solver. Firstly, ASP is defined as an Asymmetric Generalized TSP (AGTSP). Noon and Bean (1993) introduce a methodology that turns AGTSP into a Clustered TSP (CTSP). Then, CTSP is converted into an

Asymmetric TSP (ATSP). A technique obtaining Symmetric TSP (STSP) from ATSP is proposed in Ben-Arieh, et al. (2003). Consequently, the new structure satisfies the need for the Concorde solver. Finally, ASP-H is a problem that uses multiple aerial vehicles. Due to that, STSP is transformed into an m-TSP structure using a method suggested in Gorenstein (1970).

Before describing the transformation, different problem types used in the procedure are defined below:

In TSP, it is assumed that there is a travelling salesperson visiting several cities. The objective of the problem is to find the minimum distance Hamiltonian cycle. The salesperson must visit all the cities. If arcs between the nodes have direction, the problem is defined as ATSP. Otherwise, it is STSP.

Being a variant of TSP, GTSP is explained as follows: There is a travelling salesperson assumed to visit multiple cities. These cities are gathered in areas and the salesperson must visit all the areas by visiting only one city in each area. The salesperson starts from a point and must find the shortest tour that visits all areas once. The problem's structure may lead to some differences, and this changes the definition of the problem. If the graph's arcs have a direction, the problem is Asymmetric GTSP. Otherwise, it is Symmetric GTSP.

Another extension is CTSP. There are clusters that include several nodes to visit. One salesperson must visit all the clusters and the nodes in the clusters consecutively. The salesperson does not visit another cluster without visiting all nodes of the current cluster.

In the beginning, ASP must be reflected as GTSP. There are areas to be observed, which can be assumed as clusters. The UAV must leave an area from an exit/entry point and go to another area. After that, it enters and departs from two different exit/entry points. If the travel between two nodes is defined as follows, ASP can be reflected as GTSP.

- A UAV departs an area from an exit point and goes to another area
- The area is surveilled by the UAV using a pattern.
- The UAV reaches the exit point

Thus, if the travel between two nodes is defined from one exit point to another the problem can be reflected as AGTSP. The graph of the problem has directions because the nodes do not explain each other in the same way in both directions.

I is an AGTSP instance and defined as a directed graph $G(N, A)$, where N is the set of nodes, S is the set of areas, A is the set of arcs and the cost vector is c. Set A defines the directed arcs between nodes that have nonnegative costs $c_{ij} \neq c_{ji} \geq 0, (i, j) \in A$. The nodes are in m non-intersecting clusters. $i.e.$ $N = S_1 \cup S_2 \cup \ldots \cup S_m$ $with$ $S_i \cap S_j = \emptyset$, $\forall$ $i, j, i \neq j$.

In ASP, sixteen nodes are defined for each rectangular area except the base. Each node represents the possible entry and exit points that the UAV can use. Thus, if there is a problem with $n - 1$ rectangular areas, the graph of the problem has $16 * (n - 1) + 1$ nodes in AGTSP. Because one node is base, the rest of the nodes are entry/exit point combinations inside the rectangular areas. The node representation is as follows: the base node is the starting point and reflected as index 1, the rest of the regions would be shown as $2(1), 2(2), \ldots, 2(16), 3(1), 3(2), \ldots, n(16)$ respectively. In that sense, the first number reflects the cluster (rectangle), and the number inside parentheses shows the exit and the entry points combination used by the UAV. The references for inner indexes are shown in Table 4.1.

Table 4.1 Reference table for indexes of entry/exit point combinations

Index	Entry/Exit	Index	Entry/Exit
(1)	1-5	(9)	5-1
(2)	1-8	(10)	5-4
(3)	2-3	(11)	6-2
(4)	2-6	(12)	6-7
(5)	3-2	(13)	7-3
(6)	3-7	(14)	7-6
(7)	4-5	(15)	8-1
(8)	4-8	(16)	8-4

For example, moving from node $2(5)$ to $3(8)$ means that the UAV exits area 2 from entry/exit point 2 and goes to area 3's entry/exit point 4, finishes surveillance in the area, and exits from area 3's entry/exit point 8. From the other side, node $3(8)$ to $2(5)$ implies that the UAV exits area 3 from entry/exit point 8 and goes to area 2's entry/exit point 3, finishes surveillance in the area, and exits from area 2's entry/exit point 2. Consequently, both representations use different paths. Therefore, the problem is asymmetric.

In Figure 4.1, the dash dotted line refers to moving from node $2(5)$ to $3(8)$, the dashed line refers to moving from node $3(8)$ to $2(5)$.

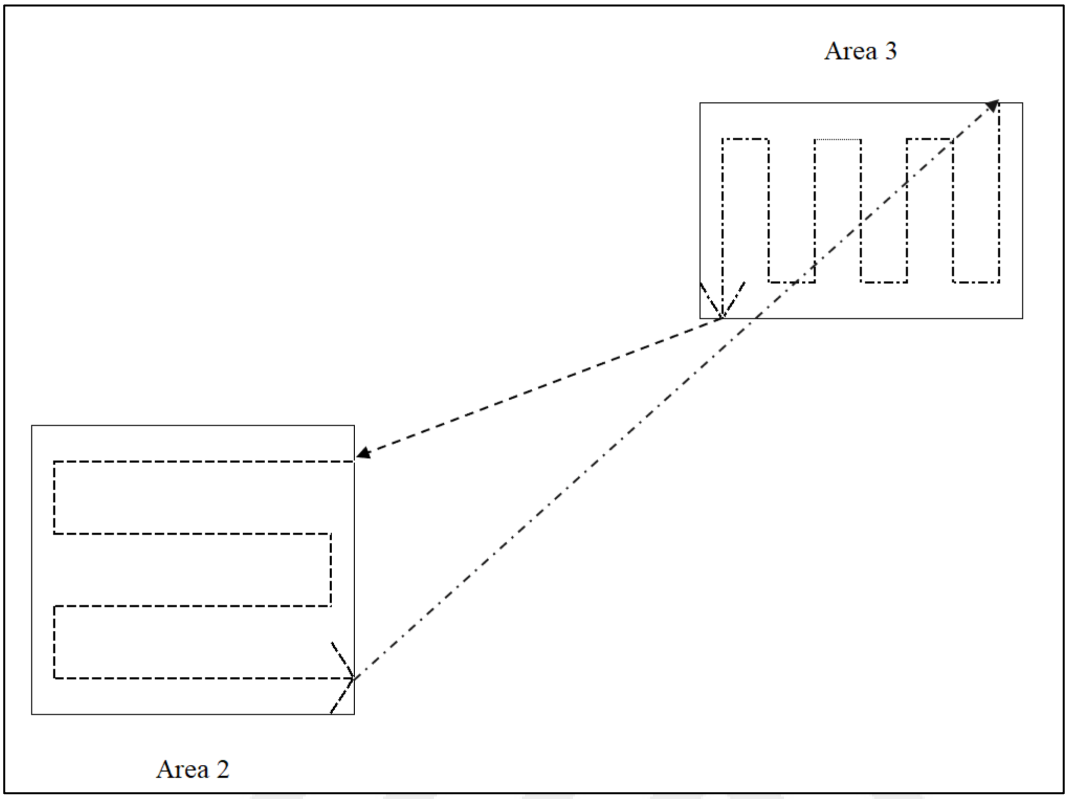


Figure 4.1. Two path examples that show asymmetric structure of the defined GTSP.

Noon and Bean (1991) describes a transformation technique to reflect an AGTSP as a CTSP. The study shows that the transformation is possible when the cost between the nodes in the same cluster is set to zero in a respective manner. However, if the vehicle comes from another group (cluster), the problem must reflect the exact cost value. So, in a CTSP, when a vehicle gets inside a cluster, the vehicle pays a fee for entering. The vehicle visits each node respectively in the cluster without suffering any cost.

The CTSP can be described using the same notation:

I' is a CTSP instance that is on a graph $G(N', A')$ where N' is the set of nodes, A' is the set of arcs and a vector with nonnegative arc costs c'. Additionally, N' is a union of m non-intersecting clusters, $N' = \bigcup_{i=1}^{m} S_i$. For each cluster, it is assumed that the member nodes have an index order, $i^1, i^2, \dots, i^r \in S_i$ and $|S_i| = r$.

The cost data transformation from AGTSP to CTSP, which is construction of arc set A', is below:

The nodes in each cluster with $|S_i| = r > 1$ construct directed arcs according to the index ordering at the former paragraph. Consequently, the following arcs are constructed: $(i^1, i^2), (i^2, i^3), (i^3, i^4), \dots, (i^{r-1}, i^r), (i^r, i^1)$ in A' with a cost of zero. That is $c'_{i^1 i^2} = c'_{i^2 i^3} = \cdots = c'_{i^{r-1} i^r} = c'_{i^r i^1} = 0$. For intercluster movement, each arc $(i^j, k^l) \in A$ is an identical arc for $(i^{j-1}, k^l) \in A'$that is $c'_{i^{j-1} k^l} = c_{i^j k^l}$. In addition, each arc $(i^j, k^l) \in A, r = |C_i|$ creates an arc $(i^j, k^l) \in A'$ with cost $c'_{i^r k^l} = c_{i^1 k^l}$.

In short, while transferring the structure of the problem from AGTSP to CTSP, inner-cluster arcs reflect no cost. For intercluster arcs, if the salesperson goes from node index $(i-1)$ of cluster f to index j of cluster k, the model must reflect the cost between the index i of cluster f to index j of cluster k. The related example is visually presented in Figure 4.2. In Figure 4.2, dotted circles are clusters, and regular circles are inner nodes. While dashed arrows are reflecting costs, dotted arrows are reflecting no cost. So, the intercluster movement has a cost, but the internal cluster movement is cost-free.

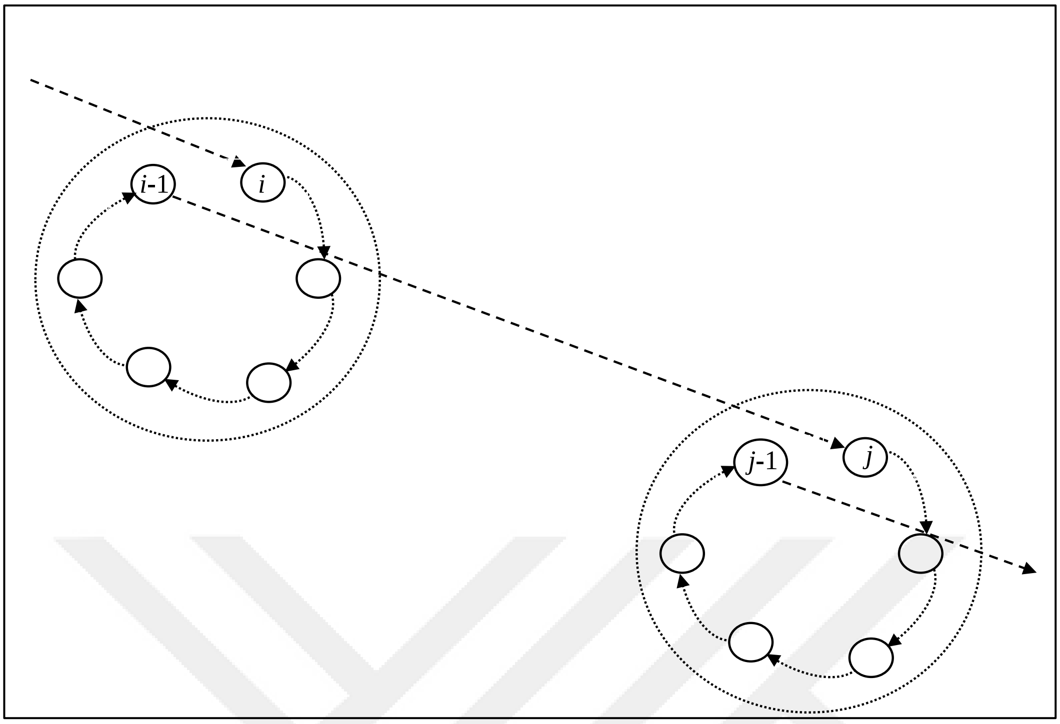


Figure 4.2. A visual representation of air vehicle's movement in and between clusters.

In Noon and Bean (1991) it is also stated that the transformation from CTSP to ATSP is handled by adding a *Big M* value to intercluster arcs. Thus, the problem structure is transformed into ATSP. Another transformation is applied to the current structure of the problem In Ben-Arieh et al. (2003) a standard reduction is used to transform ATSP into a symmetric one. Each node x_i becomes three nodes that are denoted as ${x_i}^+$, ${x_i}^0$ and ${x_i}^-$. The edge costs of ${x_i}^-$ to ${x_i}^0$, and ${x_i}^0$ to ${x_i}^+$, are set to zero. Each arc that is denoted as ${x_i}^+$ to ${x_j}^-$ has the cost of x_i to x_j. The rest of the arc values take a *Big M* value greater than the total cost of each arc.

For n areas of interest (including the base node), ATSP has the same number of nodes of AGTSP that is $16 * (n-1) + 1$. Therefore, ATSP to TSP transformation for n rectangular areas (including the base node) is equal to $3 * (16 * (n-1) + 1)$.

Finally, ASP–H is transformed into a TSP. After that point, it is possible to solve ASP with Concorde.

Under these conditions, Concorde TSP solver becomes easily applicable to ASP with a single UAV. Since ASP-H considers multiple platforms, a final transformation must be applied. The current ATSP must be transformed into a m-TSP.

m-TSP aims to find the minimum total distance traveled among a predefined number of cities using m salesmen.

In Gorenstein (1970), TSP problems are transformed into a m-TSP matrix by adding $m - 1$ home city nodes to the existing graph. Arcs that connect the home cities are set to have a $Big\ M$ cost value which prevents the model to use these specific arcs in the solution. Therefore, a tour starts from actual home city, and it cannot end without visiting the other home cities. An illustration of a full graph example for five cities and two salesmen is shared in Figure 4.3.

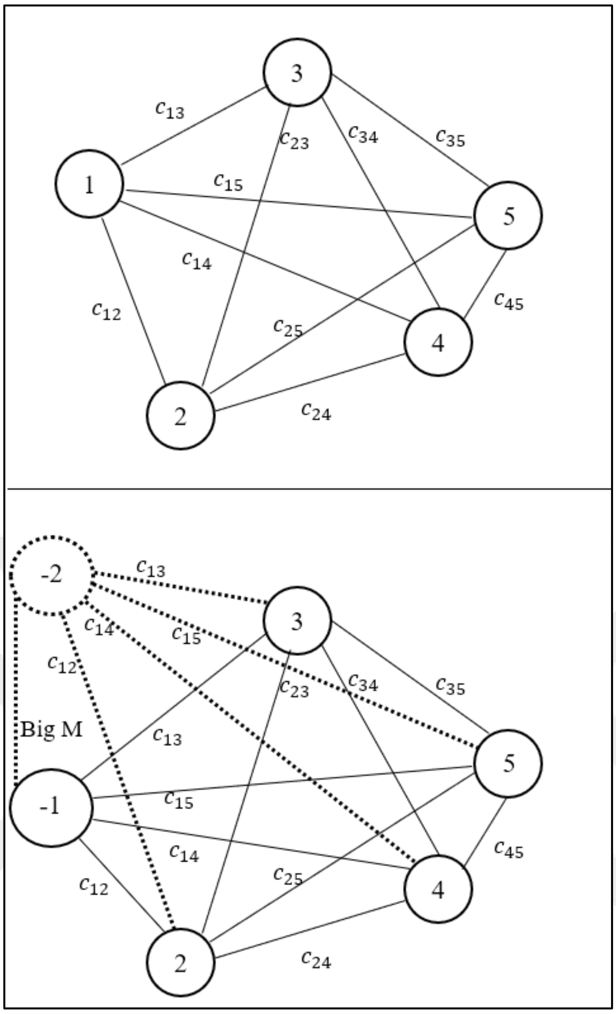


Figure 4.3. Transformation of a TSP graph to 2-TSP graph.

In Figure 4.3, a TSP with five cities is shown at the upper side. An additional home city node is added to the graph to transform the problem into a 2 TSP at the bottom. The dotted circle and arcs are added to the graph. The home cities are denoted with negative index. A feasible solution for 2-TSP such as (-1, 4, 2, 3, -2, 5, -1) is decoded into two tours, (1, 4, 2, 3, 1) for salesperson 1 and (1, 5, 1) for salesperson 2.

4.2 Proposed Heuristic Methods

In this section, two different procedures are described. Then, two heuristic algorithms that utilize different combinations of the procedures are presented.

At the beginning, the subsection 4.2.1 starts with explanation of some terms that are used to clarify the features of the procedures. A general structure for the procedures is explained. Accordingly, the two procedures are described in subsections 4.2.1.1 and 4.2.1.2 which are Change and Exchange procedures respectively. Subsections 4.2.2 and 4.2.3 describe the heuristic methods, Heuristic Based on Concorde (HBC) and Heuristic based on Neighborhood Search (HBN).

4.2.1 Procedures

In this subsection, two procedures are described. These procedures are designed considering computational burden and increasing chance of obtaining good solutions. The procedures adopt different search strategies. While one procedure uses Concorde solver for new solutions, the other adopts Neighborhood Search.

Before describing the procedures, used terms are explained below.

- **Current Solution:** The solution that changes at the end of any iteration of a procedure. It is used to produce new solutions. The current solution is denoted as *CS*.
- **Basic Move**: Basic move is the set of rules adopted by the procedure to produce new solutions over the *CS*.
- **Possible Solutions:** Possible solutions are the newly produced solutions after applying the basic move to the *CS* in an iteration. They are denoted as *PS*.
- **Best Solution:** It is the best solution found at the end of the execution. The best solution is denoted as *BS*.
- **Termination Conditions:** Conditions that stop the procedure. If the procedure reaches the defined state, it stops the execution.

The procedures employ an initial solution in the beginning. Then the initial solution is set as *CS*. The defined basic move for the procedures is applied and produces a set of new solutions which are denoted as *PS*. If there are one or more feasible solutions in *PS,* the best one is chosen to be *BS* assuming that it dominates the current best. If the procedure reaches the defined state, the execution stops. An illustration of the generic procedure flow is depicted in Figure 4.4.

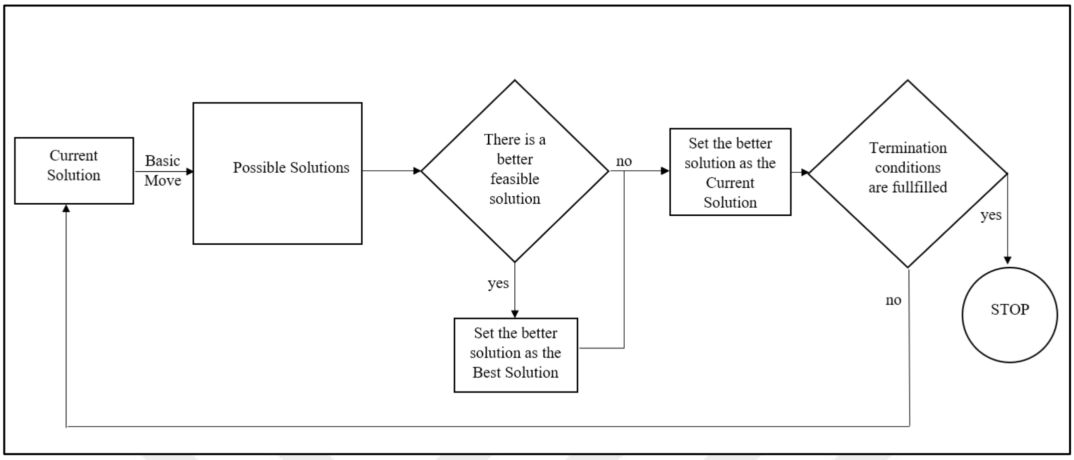


Figure 4.4. Flow chart of the procedures.

The Change procedure aims to find an improved solution by changing the nodes in tours.

The Exchange procedure tries to improve the output of the Change procedure. Hence, one of the procedure's objectives is to investigate and find a solution to the relevant problem where the other aims for improvement. In the following sections, the procedures are explained in more detail.

4.2.1.1 The Change Procedure Based on Concorde and Neighborhood Search

An initial solution must be created as the first step. The Concorde TSP solver is used to generate with an initial solution. In the previous sections, a m-TSP matrix structure is constituted for ASP-H. The Concorde TSP solver obtains an initial solution using the m-TSP matrix. Subsequently, the initial solution is set as *CS*, and the investigation of *BS* starts.

The termination condition and the basic move are defined as follows for the Change procedure.

- **Basic Move:** A rectangle of the largest tour is removed and added to another tour. The new tours are solved by the Concorde Solver or manually inserted into the existing tours by adopting neighborhood search.
- **Termination Condition:** If the initial m-TSP solution is feasible, then the optimal solution is found, hence the procedure terminates itself. Otherwise, the procedure continues to investigate. If the largest tour of *CS* is switched to another air vehicle during the execution, then the procedure terminates itself.

The basic move forces the procedure to find feasible solutions and, if so, better solutions by changing the allocation of the rectangles of the largest tour in the current state. Practically, the procedures produce feasible solutions for test instances. However, there is no guarantee that the procedure generates a feasible solution for all problem instances. A pseudocode of the Change procedure is proposed in Table 4.2:

Table 4.2 Pseudocode of the Change procedure

The Change Procedure

Input : The # of air platform M
The # of rectangular areas N
The set of rectangles for corresponding tours K_i, where $K = K_1 \cup K_2, \dots, \cup K_M$
The rectangular areas visitied by a specific air platform R^j, where $K_i = \{R^1, R^2, \dots, R^{|K_i|}\}$ and $|K| = N$

Initialize : $K_1 = K_2, \dots, = K_M = \emptyset$
Set CS, PS, BS and $banned$ as Null

Begin
  Solve the problem with Concorde Solver as m-TSP
  Set CS as m-TSP solution of the problem and denote the largest tour as K_{max}
  if CS is infeasible **then**
    Do while K_{max} is the largest tour
      for $i \leftarrow 1$ **to** M **do**
        if $K_i \neq K_{max}$ and $K_i \neq banned$ **then**
          for $j \leftarrow 1$ **to** $|K_{max}|$ **do**
            Change the air vehicle of R^j from K_{max} to K_i.
            Solve the new K_i and set it as $K_i{}'$ by ….
            Solve the new K_{max} and set it as $K_{max}{}'$ by … if it has not solved yet.
            Record the new solution as a PS, where $PS \leftarrow K_i{}' \cup K_{max}{}' \bigcup_{f \neq i,max} K_f$.
        else
          Continue.
      if PS contains feasible solution **then**
        if the best feasible solution in PS is better then the current BS **then**
          Set the best feasible solution in PS as BS and CS
        else
          Set the best solution in PS as CS
      else
        if a loop is detected **then**
          Set the smaller tour index that switches the rectangular area as $banned$
        else
          Set the best solution in PS as CS
  else
    Continue.
End

Two different approaches are utilized for generating the feasible solutions: The dotted parts in lines 9 and 10 in Table 4.2 can be filled with in two ways.

- **Change procedure based on Concorde:** The procedure solves the new K_i and K_{max} using Concorde Solver
- **Change procedure based on Neighborhood Search:** The procedure evaluates the best rectangular area one by one and places the rectangular area in the best way possible.

If the procedure uses the Concorde based method, it solves both tours separately by using the Concorde solver. Otherwise, the removed rectangle is inserted into the related tour by evaluating each possible option. The minimum distanced option becomes the new *PS*.

The Neighborhood Search based method evaluates where to insert the removed rectangle to the new tour K_i by examining all possible insertions between the existing rectangular areas.

An illustration of the *PS* generation of the Change procedure based on Concorde for 9 rectangular areas and 3 air vehicles is given in Figure 4.5.

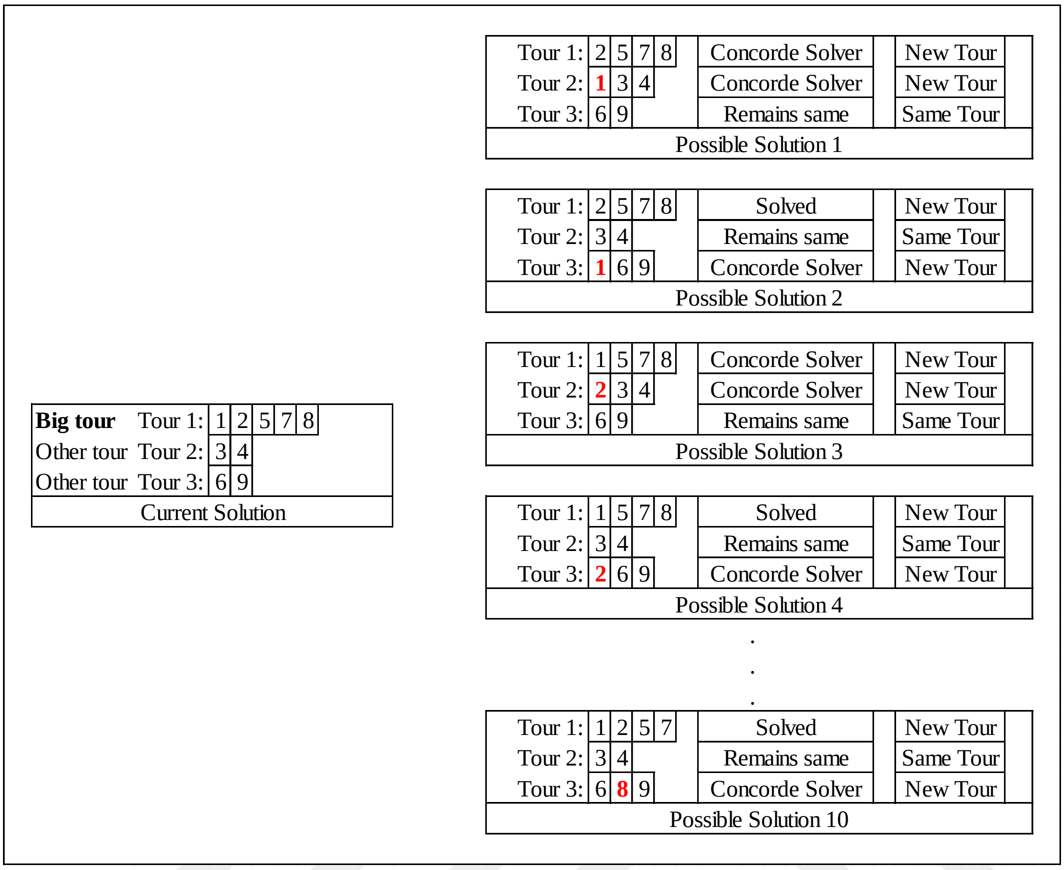


Figure 4.5. An illustration of the *PS* generation of the Change procedure based on Concorde.

An illustration of the *PS* generation of the Change procedure based on Neighborhood Search for 6 rectangular areas and 2 air vehicles is given in Figure 4.6.

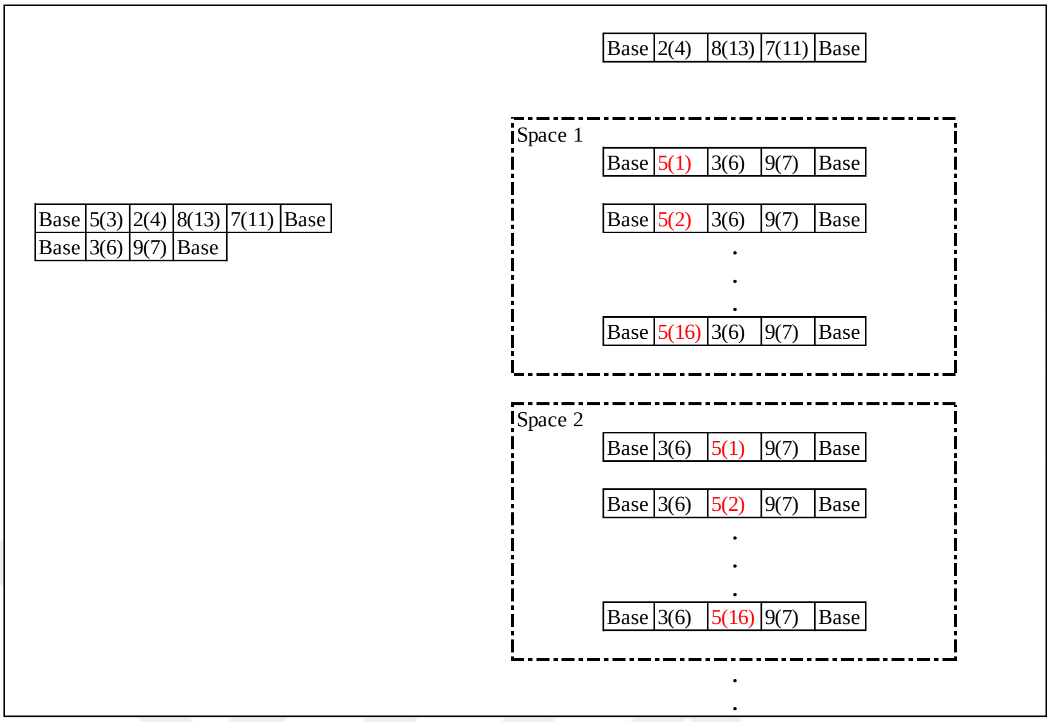


Figure 4.6. An illustration of the *PS* generation of the Change procedure based on Neighborhood Search.

In Figure 4.6, the numbers outside of the parenthesis stand for the rectangular areas, and the numbers in the parenthesis stand for the entry/exit point combinations. Two tours placed on the left side show the largest tour and one of the small tours in *CS*. On the right side, rectangle 5 is removed from the largest tour and placed at the top. Each space is evaluated below side in Figure 4.6 for all entry/exit point combinations. Consequently, the removed rectangle is placed to the space using the entry/exit point combination that provides minimum distance. The related solution is set as a *PS*.

4.2.1.2 The Exchange Procedure

The Exchange procedure improves the solution by exchanging the rectangular areas between the largest tour and smaller tours. The basic move and termination condition for the Exchange procedure is explained below:

- **Basic Move:** One of the rectangles in the largest tour is exchanged with another rectangle in one of the smaller tours. The two newly generated rectangle sets are solved with Concorde solver, and the tours are routed. If there are any other tours, they construct a new solution with the new tours together.
- **Termination Condition:** If *BS* cannot improve in the current iteration, the procedure terminates itself.

The Exchange procedure has a more straightforward composition rather than the Change procedures. The termination is rapid and directly aims for improvement.

Similarly, each exchange of the rectangular areas creates new tours to solve, just like in the Change procedures. The Exchange procedure overcomes this challenge by using the Concorde solver. Despite the time consumption of the Concorde solver, it gives better solutions in comparison to Neighborhood Search. Hence, the Concorde solver becomes ideal for the Exchange procedure.

A pseudocode for the Exchange procedure is shown in Table 4.3 as follows in detail step by step:

Table 4.3 Pseudocode of the Exchange procedure.

The Exchange Procedure

Input : The # of air platform M.
The # of rectangular areas N.
The set of rectangles for corresponding tours K_i, where $K = K_1 \cup K_2, \ldots, \cup K_M$.
The rectangular areas R^j, where $K_i = \{R^1, R^2, \ldots, R^{|K_i|}\}$ and $|K| = N$.
Set CS and BS as the input solution.

Initialize : Set PS as Null.

Begin
  Denote the largest tour in CS as K_{max}.
  Do while if BS is changed
    for $i \leftarrow 1$ **to** M **do**
      if $K_i \neq K_{max}$ **then**
        for $j \leftarrow 1$ **to** $|K_{max}|$ **do**
          for $f \leftarrow 1$ **to** $|K_i|$ **do**
            Exchange the rectangles R^j and R^f between tours K_{max} and K_i.
            Solve the new K_i and set it as K_i' using Concorde Solver.
            Solve the new K_{max} and set it as K_{max}' using Concorde solver if it has not solved yet.
            Record the new solution as a PS, where $PS \leftarrow K_i' \cup K_{max}' \cup_{k \neq i,max} K_k$.
      else
        Continue.
    if PS contains feasible solution **then**
      if the best feasible solution in PS is better then the current BS **then**
        Set the best feasible solution in PS as BS and CS.
      else
        Continue.
    else
      Continue.
End

4.2.2 The Heuristic Based on Concorde (HBC)

The HBC is the combination of two procedures. It uses the Change procedure to investigate the related problem and obtain a good solution. In the improvement phase, the obtained solution from the Change procedure is imported into the Exchange procedure. The Exchange procedure seeks a better solution.

The HBC uses the Change procedure based on Concorde. Despite the Concorde Solver being a fast tool, the repetitive execution consumes significant time. Therefore, the HBC is expected to be an exhausting heuristic in terms of time. However, it is more likely to obtain better solutions.

HBC method is described as follows:

Step 0 : Initialize the BS.

Step 1 : Use the Change procedure based on Concorde to generate a good solution and set the output solution as BS .

Step 2 : Use the Exchange procedure to improve the BS.

4.2.3 The Heuristic Based on Neighborhood Search (HBN)

The HBN uses the Change procedure based on Neighborhood Search. The Change procedure generates a good solution, afterward the Exchange procedure is executed.

The HBN method is shown as follows:

Step 0 **:** Initialize the BS.

Step 1 **:** Use the Change procedure with Neighborhood Search to generate a good solution and set the output solution as BS .

Step 2 **:** Use the Exchange procedure to improve the BS.

The reason behind these algorithms is to see whether it is logical to use exhausting but rapid solvers (Concorde) to obtain reasonable solutions in exchange of high memory usage and significant time consumption. On the other hand, instead of using Concorde solver, is it sufficient to use faster basic moves to obtain new solutions. Therefore, the produced solutions may not show better results, but the solution time significantly decreases.

In short, the proposed heuristics obtain reasonable solutions. However, the performance differences between Concorde solver and the Neighborhood Search method are investigated in the computational results chapter.

CHAPTER 5

COMPUTATIONAL RESULTS

In this chapter, the proposed mathematical model and the heuristics are evaluated. ASP-H is a new problem. Therefore, there is not a benchmark set already generated. A new benchmark set is generated, and distance capacity values are calculated according to a procedure used for generating various benchmark problems is described in Section 5.1. Then, a new methodology is suggested to obtain strict distance capacity values in Section 5.2. Finally, the results are discussed in Section 5.3.

5.1 ASP-H Problem Generation

There is no benchmark instance produced for this specific problem type. Hence, twenty-five test instances are generated using the same test set used in Karasakal (2016) to evaluate the mathematical model and algorithms. The test set used in Karasakal (2016) is shared in APPENDIX A.

There are 60 disjoint rectangles placed on a two-dimensional plane. Various number of rectangles are randomly chosen using a uniform distribution. Including the base station, new benchmark instances are generated for 7, 10, 15, 20, and 25 number of rectanles. While the selection of rectangles is completed, the inner path lengths of rectangles are calculated according to the coordinates of rectangles. For each number of rectangle, 5 instances are created. So, in total, 25 different instances are prepared to be tested.

A test instance for 10 rectangles instances, including base (origin), is shown in Figure 5.1.

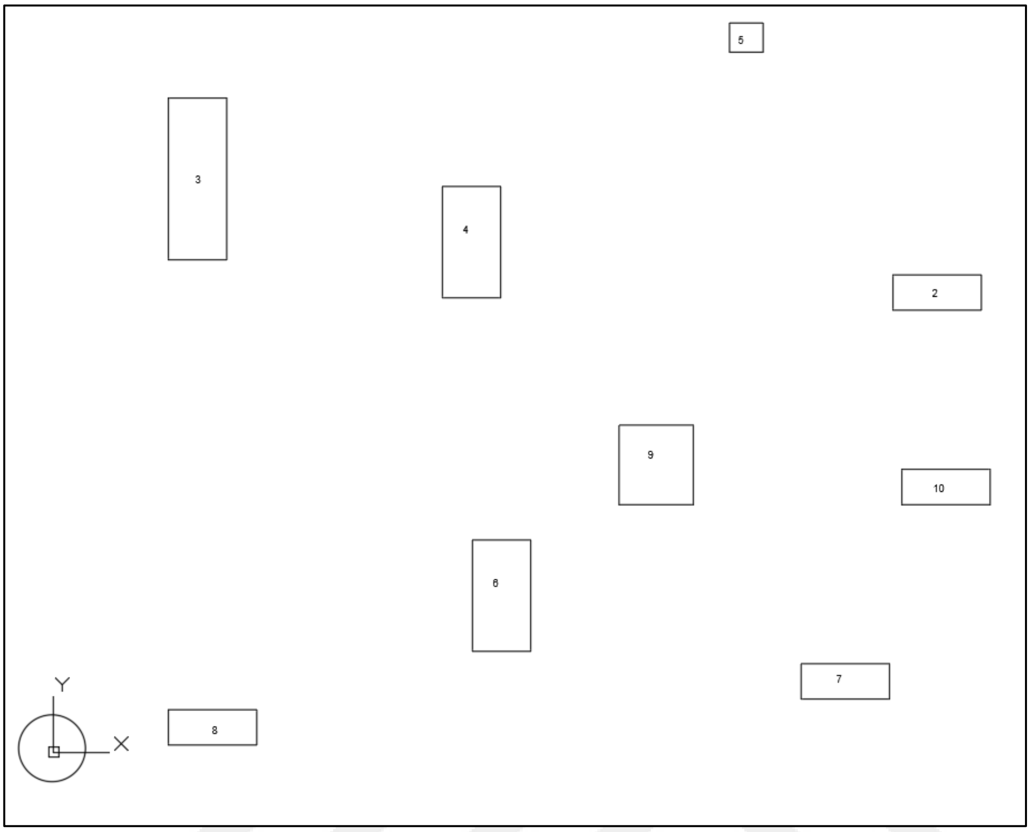


Figure 5.1. A 2-D visualization of 10 rectangles instance including the base (origin)

After the generation of different test instnaces, each instance is considered for 2, 3, 4, and 5 air vehicles. Therefore, one generated instance is used for four other benchmark instances. For that reason, one hundred different benchmark instances are generated for evaluation.

The instance names are given in a rule. The following term is an example to explain the rule: "IN_N#_M#_I#". The number after the parameter N explains the number of rectangles used in the problem, including the base. The number after M notation specifies the number of used air vehicles for the corresponding instance. Finally, the number after notation I remarks the index of the related instance type. For example, "IN_N7_M2_I1" means that, instance index 1 for 7 rectangles and 2 air vehicles.

So far, the instances are generated. In the next section, the methodology to determine the distance capacity is given.

5.2 Specification of Distance Capacity Limit

The flight distances for air platforms are calculated for each generated instance specifically. The methodology uses an upper and a lower bound to specify the exact number for the endurance values. If ASP-H model is relaxed by removing the capacity constraint, the model behaves like an m-TSP model. For example, the relaxed model is applied for two UAVs. The solution contains two tours, and the higher value is the maximum capacity value for ASP-H instance. If this value is used as a capacity limit for ASP-H, the same instance's solution does not change. This upper bound value is used to guarantee that the capacity constraints are meaningful for the ASP-H. The capacity limit should be less than the upper bound.

On the other hand, the lower bound aims to produce a strict capacity value. For this reason, a myopic algorithm for a single vehicle is used to make a strict lower bound.

One air vehicle starts from the base station and chooses the closest area of interest at that moment. After creating a Hamiltonian cycle, the output length value is divided by the number of UAVs used for the related problem. Therefore, a reasonable lower bound value is created for the instances. Consequently, a range for endurance is generated. Three different endurance values are found considering these bounds. These numbers are calculated as follow:

$$C_d = \begin{cases} LB + 0.25 * (UB - LB) & , \text{if } d = 1 \\ LB + 0.50 * (UB - LB) & , \text{if } d = 2 \\ LB + 0.75 * (UB - LB) & , \text{if } d = 3 \end{cases} \tag{33}$$

In this formulation, UB and LB stand for upper and lower bounds. Index d is a level indicator that underlines the difficulty level of an instance. While d increases, the difficulty level of the instance decreases.

Finally, the specified numbers for related instances applied for ASP-H. Then, a procedure is adopted to select the capacity limit as follows:

Step 1 : Apply C_1 for ASP-H and solve the instance with Cplex.

Step 2 : Does the model give a solution?

If yes choose C_1 as capacity and STOP, otherwise go to Step 3.

Step 3 : Apply C_2 for ASP-H and solve the instance with Cplex.

Step 4 : Does the model give a solution?

If yes choose C_2 as capacity and STOP, otherwise go to Step 5.

Step 5 : Apply C_3 for ASP-H and solve the instance with Cplex.

Step 6 : Does the model give a solution?

If yes choose C_3 as capacity and STOP, otherwise go to Step 7.

Step 7 : Generate another test instance.

5.3 The Results

Before generating benchmark instances, the problem instance used in Karasakal (2016) is adapted for ASP-H with two UAVs. Also, the HBC and the HBN methods are applied to the corresponding instance and compared with the Cplex solution in Table 5.1.

Table 5.1 The comparison table for the instance problem used in Karasakal (2016).

CPLEX			HBC			HBN		
Gap (%)	Value	Time (Sec.)	Gap (%)	Value	Time (Sec.)	Gap (%)	Value	Time (Sec.)
0.00	92.529	272.43	0.30	92.81	34.719	0.00	92.529	15.124

For this instance, the HBN method dominates the HBC concerning time and gap ratio. The HBN found the optimal solution faster than the Cplex solver. On the other hand, the HBC method is also fast and gives a good solution with a 0.3% GAP ratio.

In Figures 5.2 and 5.3, the solutions are illustrated. The parameters at the sides of the rectangular areas give information about used patterns, where D' is the distance of the corresponding pattern, t is the number of turns along with the pattern, and s is the track spacing.

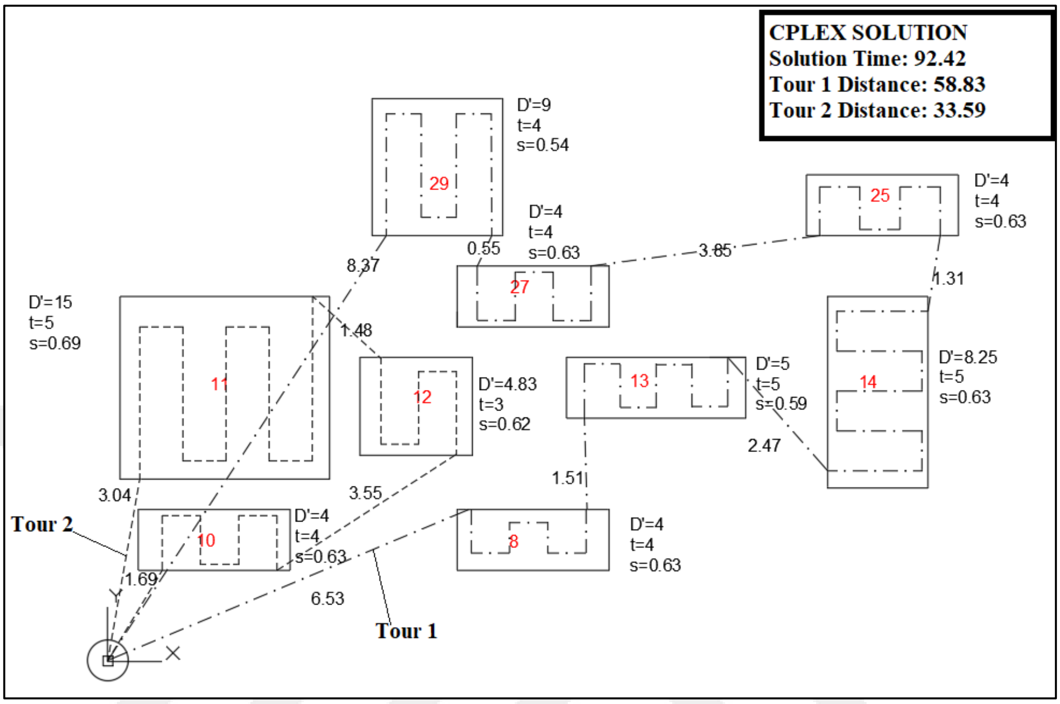


Figure 5.2. An illustration optimal solution for problem instance used in Karasakal (2016).

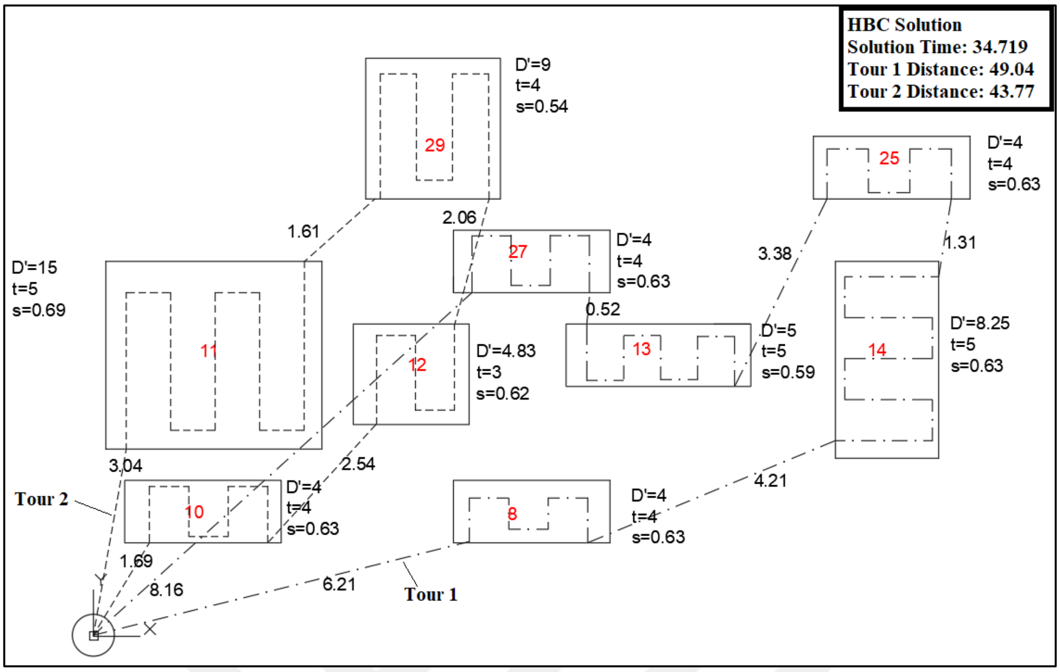


Figure 5.3. An illustration of the solution obtained with the HBC method for problem instance used in Karasakal (2016).

After the generation of benchmark instances and the distance capacity values, it becomes applicable to gather computational results of the model.

Example output series of a ten rectangular region problem including base is shown in Figures 5.4, 5.5, 5.6, 5.7.

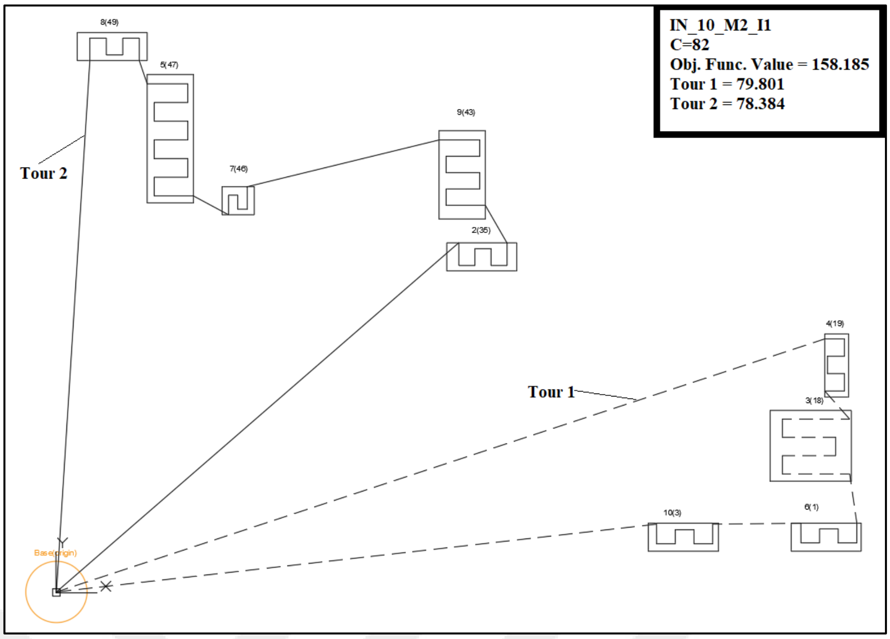


Figure 5.4. An illustration of optimal solution of benchmark instance IN_10_M2_I1.

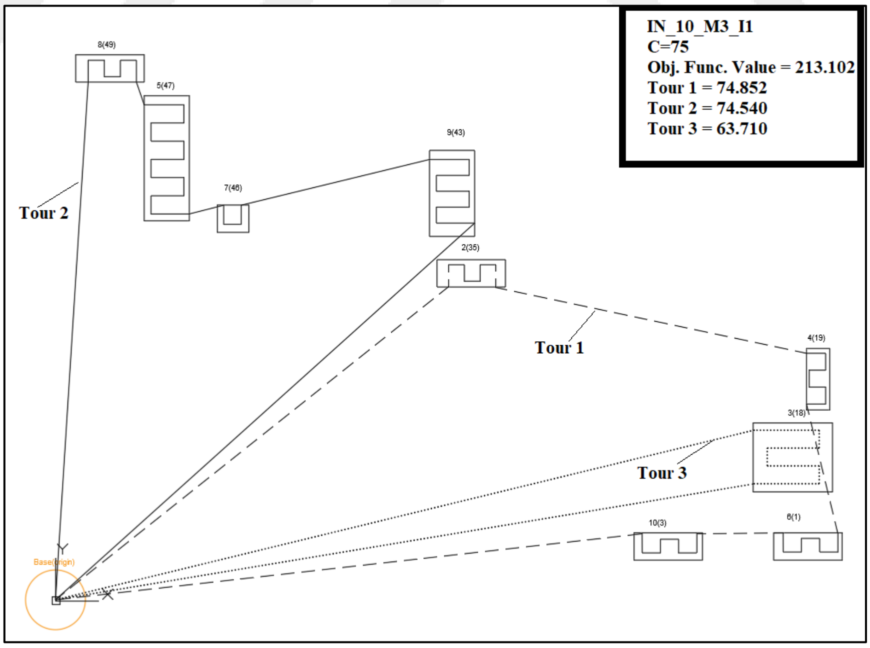


Figure 5.5. An illustration of optimal solution of benchmark instance IN_10_M3_I1.

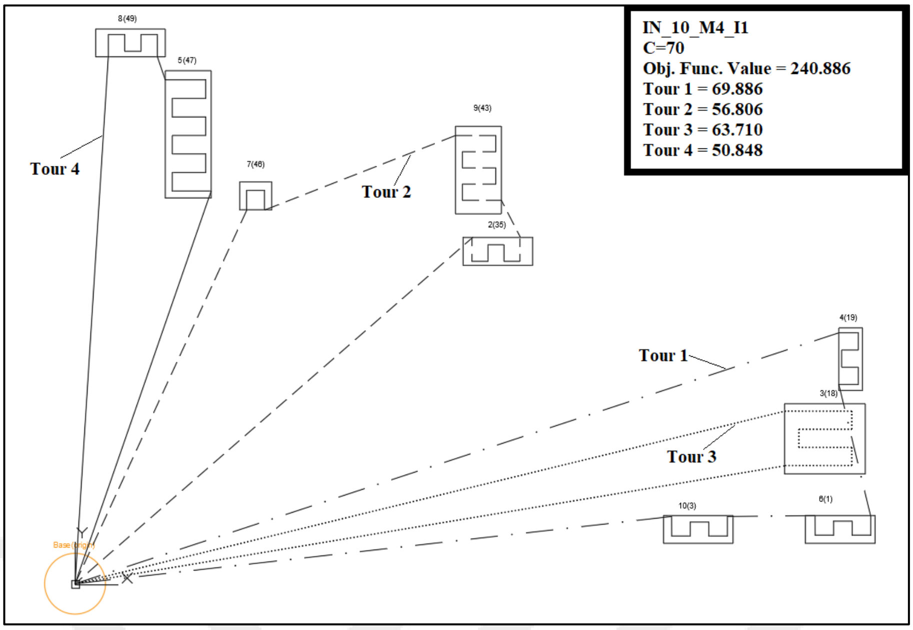


Figure 5.6. An illustration of optimal solution of benchmark instance IN_10_M4_I1.

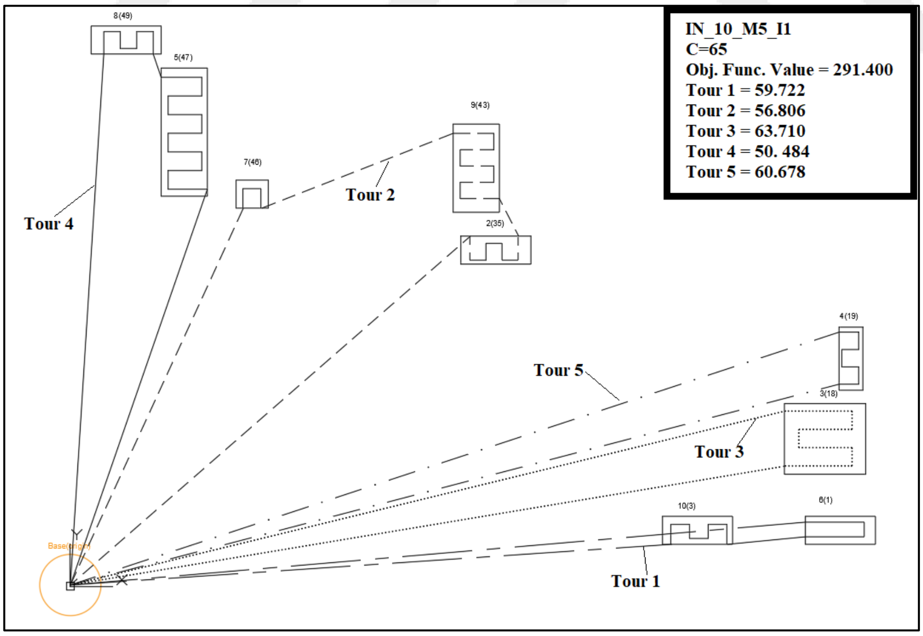


Figure 5.7. An illustration of optimal solution of benchmark instance IN_10_M5_I1.

The performance measures and the parameters that are used for the evaluation of the mathematical model and comparison between the mathematical model and heuristic algorithm results are listed as follows:

- **Time:** The wall clock time consumption of the corresponding problem instance in terms of seconds.

- **Value:** The objective function value of the corresponding problem for the solution found.

- **Status:** If the run does not reach the 24 hours time limit the status is “Optimal”. If the run reaches the time limit, then the status is “Timelimit”.

- **GAP:** The difference between the best-known solution and the best-known bound in optimization software at the moment. When the gap reaches zero percent, it means the optimal solution to the problem is found. The formulation of the gap measure is explained below:

$$GAP = \frac{z_P - z_D}{z_P} * 100 \tag{34}$$

Where z_P is primal model value, and z_D is dual model value.

- **GAP v.2:** The difference between the mathematical model solution and the heuristic algorithm solution for the corresponding problem instance. The formulation is stated below:

$$GAPv.2 = \frac{AO - BK}{BK} * 100 \tag{35}$$

Where AO is the objective function value of the corresponding heuristic algorithm, and BK is the best-known solution of the corresponding problem instance found by optimization software.

- **Time Ratio:** The ratio of the time consumption of the heuristic algorithm to the time consumption of the optimization software. It is used to understand the time difference between the heuristic algorithm and optimization tool.

$$Time\ Ratio = \frac{T_{HA}}{T_{OS}} * 100 \tag{36}$$

Where T_{HA} is the wall clock time consumed by the corresponding heuristic algorithm, and T_{OS} is the wall clock time consumed by the optimization software.

The problem instances are solved with IBM CPLEX optimization software version 12.10. The hardware features on a personal computer with Intel(R) Xeon(R) E-2246G processor, 16384 MB RAM, 6 cores, 12 CPU's at ~ 3.6GHz clock speed.

Since the solution of the problems consumes significant time, a time limit is set for each model run. Therefore, each problem is restricted to 86400 seconds. Additionally, a Cplex Optimization Studio function is used that is *nodefileind*. By default, the software uses RAM to record the solutions found. However, for many instances, the memory consumption increases and exceeds the memory capacity at hand.

Consequently, *nodefileind* function is used at setting number three, which allows the software to use hard disk memory. Also, *workdir* parameter must be connected to a file (directory) to use *nodefileind* function. Otherwise, the software terminates the run because it cannot record the corresponding solution files. The computational results of the mathematical model are contained in APPENDIX B.

The heuristic methods are written in C Programming Language, and the Concorde TSP solver callable library is used. The algorithm runs are executed in the same computers stated above. The computational results of the heuristic methods are shared in APPENDIX C.

Since all outputs are obtained, comparison tables are formed to evaluate the heuristic methods. Average and median values are calculated and analyzed together to comment on a general perspective.

In Table 5.2, the total number of instances can be examined where x is the Gapv.2 values.

Table 5.2 The total number of outputs grouped by Gapv.2 values.

GAPv.2	HBC	HBN
$x \leq 5\%$	76	64
$5\% < x \leq 10\%$	16	22
$10\% < x \leq 15\%$	6	10
$x > 15\%$	2	4

Performances of the proposed algorithms are compared concerning the number of rectangular areas. For that reason, the instances are grouped into three subgroups. Hence, instances with 7 and 10 rectangles are denoted as small size, instances with 15 and 20 rectangles are considered medium-size, and instances with 25 rectangles are denoted as large size.

The average values for performance parameters are calculated and shared for corresponding problem instances in Table 5.3.

Table 5.3 The average values for the performance parameters.

Problem Info.	Cplex		HBC			HBN		
Instance Group	Gap (%)	Time (Sec.)	Gapv.2 (%)	Time (Sec.)	Time Gap (%)	Gapv.2 (%)	Time (Sec.)	Time Gap (%)
Small N=7, 10	0.81	14702.30	1.51	11.97	0.08	3.44	6.52	0.04
Medium N= 15, 20	4.39	76940.00	4.26	494.54	0.64	6.08	225.96	0.29
Large N=25	4.29	83560.88	3.53	3818.00	4.57	5.04	1069.03	1.28

According to the average values, HBC and HBN are superior in time consumption compared to the Cplex. On the other hand, comparing Cplex and heuristic algorithms can be misleading since most of the Cplex solutions are obtained without reaching the optimal solution.

Due to Gapv.2 values, the HBC algorithm finds better solutions compared to the HBN method. Since the HBC uses Concorde solver repeatedly, it finds solutions with low gap values quickly. Therefore, the HBC becomes an appropriate option for good objective function values.

On the other hand, the HBN method dominates the HBC in terms of solution time. Because the HBN uses fewer Concorde solvers compared to the HBC, for that reason, it is expected to observe less solution time for the HBN. Consequently, the HBN is a better option for short solution times.

The average values show both of the heuristics provide reasonable solutions, and they are swift. Still, median values give another perspective. In Table 5.4, the median values are given.

Table 5.4 The median values for the performance parameters.

Problem Info.	Cplex		HBC			HBN		
Instance Group	Gap (%)	Time (Sec.)	Gapv.2 (%)	Time (Sec.)	Time Gap (%)	Gapv.2 (%)	Time (Sec.)	Time Gap (%)
Small N=7, 10	0.00	302.51	0.00	9.17	3.03	1.28	5.67	1.88
Medium N= 15, 20	3.95	86411.63	2.23	301.11	0.35	4.99	138.30	0.16
Large N=25	3.64	86414.00	2.09	2523.79	2.92	3.27	1031.90	1.19

Due to median values, the performance measures show significant differences compared to the average values. The small number of extreme outputs negatively affects the performance parameters. Especially, median values are quite different for small-size Cplex values. However, it does not change the comments based on average values. The median values support and show that the heuristic methods provide better performance while considering Tables 5.2 and 5.4.

The heuristics are examined to understand the contribution of the Exchange procedure.

The Exchange procedure contributed 19 problem instances with average values of 1.74% marginal contribution with 707.93 seconds for the HBC. The Change procedure carries a big portion of the workload.

On the other hand, 54 instances improved by the Exchange procedure for the HBN with a 5.2% marginal contribution.

The Exchange procedure is more beneficial for the HBN. Still, it is advantageous for both methods since the Cplex solutions consume significant amount of time.

The memory usage of the exact method is extensive. Accordingly, 48 problem instances give errors because of the memory overflow. Therefore, the Cplex uses hard disk, and the heuristic methods only use RAM. Consequently, the heuristic methods are also helpful for efficient memory usage.

Finally, the previous parts are mentioned that both heuristics give reasonable solutions. Nevertheless, the methods found better solutions for some problem instances than Cplex with 24 hours run time. For the six problems listed in Table 5.5, the heuristic algorithms show excellent performance.

Table 5.5 The outputs of six problem instances which the heuristics show excellent performance.

Problem Info.		Cplex			HBC				HBN			
Instance Name	C	Gap (%)	Value	Time (Sec.)	Gapv.2 (%)	Value	Time (Sec.)	Time Gap (%)	Gapv.2 (%)	Value	Time (Sec.)	Time Gap (%)
IN_N25_M3_I3	140	7.32	293.83	86411.22	-2.05	287.82	1094.574	1.27	1.13	297.14	778.909	0.90
IN_N25_M2_I3	177	6.68	266.502	86420.03	-0.27	265.77	2902.476	3.36	0.44	267.68	601.131	0.70
IN_N15_M2_I5	114	6.72	187.854	86409.75	-0.15	187.58	345.532	0.40	0.00	187.854	150.781	0.17
IN_N25_M4_I4	143	5.55	290.312	86410.39	0.55	291.91	1944.686	2.25	-0.82	287.93	975.879	1.13
IN_N25_M2_I2	160	9.10	235.976	86423.59	1.04	238.44	10042.668	11.62	-0.65	234.44	902.489	1.04
IN_N25_M3_I4	150	6.64	267.783	86413.41	2.33	274.02	3672.828	4.25	-0.45	266.57	1330.254	1.54

The HBC and the HBN are effective and rapid. Moreover, the excellent performance over the Cplex is the effect of the Concorde solver. So, both methods can find the optimal solution swiftly. Still, the heuristics are open for improvement.

CHAPTER 6

CONCLUSION

In this thesis study, the concept of ASP is remembered, and a new variant of ASP is defined, that is, ASP-H as an extension of models provided in Karasakal (2016). To the best of our knowledge, ASP-H is the first study that examines the multiple UAV routing under the circumstances of ASP.

A homogenous UAV fleet must visit a set of rectangular areas, and all UAVs must visit at least one of the areas. All air vehicles are assumed to be the same. A capacity value restricts them in terms of distance. The air vehicles take off from the base, visit the rectangular areas, and return to the base as long as all the rectangular areas are visited.

The solution methods are produced and evaluated using the newly created instance problems. A new mathematical model is defined and programmed in the IBM CPLEX Linear Optimization tool to solve the problem optimally. One hundred generated instance problems are solved with twenty-four hours wall clock time limit. In terms of memory usage and time consumption, the mathematical model is significantly costly. While the size of the problem increases and the capacity limit is getting more strict, the solution time starts to become more likely to hit the time limit. Besides, the memory usage of the model is exhausting since the optimization tool can solve the problems only if it uses the HDD as the storage instead of RAM.

Two heuristic algorithms are proposed to tackle the problem efficiently. The heuristics use Concorde TSP solver. Hence, ASP-H is converted into TSP and m-TSP matrix structure with a multilayered transformation method to use Concorde TSP solver. Both heuristics consist of two procedures, named change and exchange procedures. Where one of the heuristics uses Concorde solver multiple times, the other heuristic uses Concorde solver limitedly.

The ASP-H can extend to various research topics. As for the future work, the heterogeneous fleet of UAVs can be considered. The capacity limits of the UAV vary due to the different kinds of UAVs used. Also, ASP can be considered with time windows. The rectangular areas can be visited only in a predefined time range. Additionally, Concorde solver algorithms are improvable. Instead of focussing on improving the solution at hand, focusing on allocating the rectangles to the air vehicles may give better solutions.

REFERENCES

Alighanbari, M., Kuwata, Y., & How, J. P. (2003). Coordination and Control of Multiple UAVs with Timing Constraints and Loitering. *Proceedings of the American Control Conference*, *6*. https://doi.org/10.1109/acc.2003.1242572

Alotaibi, K. A., Rosenberger, J. M., Mattingly, S. P., Punugu, R. K., & Visoldilokpun, S. (2018). Unmanned aerial vehicle routing in the presence of threats. *Computers and Industrial Engineering*, *115*. https://doi.org/10.1016/j.cie.2017.10.030

Altabeeb, A. M., Mohsen, A. M., Abualigah, L., & Ghallab, A. (2021). Solving capacitated vehicle routing problem using cooperative firefly algorithm. *Applied Soft Computing*, *108*. https://doi.org/10.1016/j.asoc.2021.107403

Applegate, D., Bixby, R., Cook, W., & Chvátal, V. (1998). On the solution of traveling salesman problems. Rheinische Friedrich-Wilhelms-Universität Bonn.

Avellar, G. S. C., Pereira, G. A. S., Pimenta, L. C. A., & Iscold, P. (2015). Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors (Switzerland)*, *15*(11). https://doi.org/10.3390/s151127783

Babel, L. (2017). Curvature-constrained traveling salesman tours for aerial surveillance in scenarios with obstacles. *European Journal of Operational Research*, *262*(1). https://doi.org/10.1016/j.ejor.2017.03.067

Ben-Arieh, D., Gutin, G., Penn, M., Yeo, A., & Zverovitch, A. (2003). Transformations of generalized ATSP into ATSP. *Operations Research Letters*, *31*(5). https://doi.org/10.1016/S0167-6377(03)00031-2

Braekers, K., Ramaekers, K., & van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. In *Computers and Industrial Engineering* (Vol. 99). https://doi.org/10.1016/j.cie.2015.12.007

Buyurgan, N., & Lehlou, N. (2015). A terrain risk assessment method for military surveillance applications for mobile assets. *Computers and Industrial Engineering*, *88*. https://doi.org/10.1016/j.cie.2015.06.025

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Haghnazar K., R., Zuidema, D., Lima, M. S. F., & da Costa, A. R. (2017). A multi-objective green UAV routing problem. *Computers and Operations Research*, *88*. https://doi.org/10.1016/j.cor.2017.04.011

Coutinho, W. P., Battarra, M., & Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. In *Computers and Industrial Engineering* (Vol. 120). https://doi.org/10.1016/j.cie.2018.04.037

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, *2*(4). https://doi.org/10.1287/opre.2.4.393

Fesenko, H., Kliushnikov, I., Kharchenko, V., Rudakov, S., & Odarushchenko, E. (2020). Routing an Unmanned Aerial Vehicle during NPP Monitoring in the Presence of an Automatic Battery Replacement Aerial System. *Proceedings - 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies, DESSERT 2020*. https://doi.org/10.1109/DESSERT50317.2020.9125080

Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, *4*(1), 61–75. https://doi.org/10.1287/opre.4.1.61

GORENSTEIN S. (1970). PRINTING PRESS SCHEDULING FOR MULTI-EDITION PERIODICALS. *Management Science*, *16*(6). https://doi.org/10.1287/mnsc.16.6.b373

Grob, M. J. H. B. (2006). Routing of platforms in a maritime surface surveillance operation. *European Journal of Operational Research*, *170*(2). https://doi.org/10.1016/j.ejor.2004.02.029

Guitouni, A., & Masri, H. (2014). An orienteering model for the search and rescue problem. *Computational Management Science*, *11*(4). https://doi.org/10.1007/s10287-013-0179-1

Howeth, L. S. (1963). Radio Controlled Aircraft. Howeth: Chapter XL (1963). http://earlyradiohistory.us/1963hw40.htm.

Jacobson, S. H., McLay, L. A., Hall, S. N., Henderson, D., & Vaughan, D. E. (2006). Optimal search strategies using simultaneous generalized hill climbing algorithms. *Mathematical and Computer Modelling*, *43*(9–10). https://doi.org/10.1016/j.mcm.2005.05.025

John, M., Panton, D., & White, K. (2001). Mission Planning for Regional Surveillance. *Annals of Operations Research, 108*(1–4). https://doi.org/10.1023/A:1016063129217

Karasakal, O. (2016). Minisum and maximin aerial surveillance over disjoint rectangles. *TOP*, *24*(3). https://doi.org/10.1007/s11750-016-0416-1

Karasakal, O., Karasakal, E., & Maraş, G. (2020). Multiobjective aerial surveillance over disjoint rectangles. *Computers and Industrial Engineering*, *148*. https://doi.org/10.1016/j.cie.2020.106732

Kiam, J. J., Besada-Portas, E., & Schulte, A. (2021). Hierarchical mission planning with a ga-optimizer for unmanned high altitude pseudo-satellites. *Sensors*, *21*(5). https://doi.org/10.3390/s21051630

Koester, R. J., Cooper, D. C., Frost, J. R., & Robe, R. Q. (2004). Sweep width estimation for ground search and rescue.

Lamont, G. B., Slear, J. N., & Melendez, K. (2007). UAV swarm mission planning and routing using multi-objective evolutionary algorithms. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, MCDM 2007*. https://doi.org/10.1109/MCDM.2007.369410

Li, J., Xiong, Y., She, J., & Wu, M. (2020). A Path Planning Method for Sweep Coverage with Multiple UAVs. *IEEE Internet of Things Journal*, *7*(9). https://doi.org/10.1109/JIOT.2020.2999083

Macharet, D. G., Monteiro, J. W. G., Mateus, G. R., & Campos, M. F. M. (2017). Bi-objective data gathering path planning for vehicles with bounded curvature. *Computers and Operations Research*, *84*. https://doi.org/10.1016/j.cor.2016.07.004

Maini, P., Sundar, K., Singh, M., Rathinam, S., & Sujit, P. B. (2019). Cooperative Aerial-Ground Vehicle Route Planning with Fuel Constraints for Coverage Applications. *IEEE Transactions on Aerospace and Electronic Systems*, *55*(6). https://doi.org/10.1109/TAES.2019.2917578

Miller, C. E., Zemlin, R. A., & Tucker, A. W. (1960). Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM (JACM)*, *7*(4). https://doi.org/10.1145/321043.321046

Naughton, R. (2005). Reginald Denny (1891-1967) - Aviation Pioneer. Reginald Denny (1891-1967) - aviation pioneer. https://www.ctie.monash.edu.au/hargrave/dennyplane.html.

Ng, K. Y. K., & Ghanmi, A. (2002). An automated surface surveillance system. *Journal of the Operational Research Society*, *53*(7). https://doi.org/10.1057/palgrave.jors.2601363

Ng, K. Y. K., & Sancho, N. G. F. (2009). Regional surveillance of disjoint rectangles: A travelling salesman formulation. *Journal of the Operational Research Society*, *60*(2). https://doi.org/10.1057/palgrave.jors.2602507

Noon, C. E., & Bean, J. C. (1993). An Efficient Transformation Of The Generalized Traveling Salesman Problem. *INFOR: Information Systems and Operational Research*, *31*(1). https://doi.org/10.1080/03155986.1993.11732212

Panton, D. M., & Elbers, A. W. (1999). Mission planning for synthetic aperture radar surveillance. *Interfaces*, *29*(2). https://doi.org/10.1287/inte.29.2.73

Pohl, A. J., & Lamont, G. B. (2008). Multi-objective UAV mission planning using evolutionary computation. *Proceedings - Winter Simulation Conference*. https://doi.org/10.1109/WSC.2008.4736199

Reinelt, G. (1991). TSPLIB. A traveling salesman problem library. *ORSA Journal on Computing*, *3*(4). https://doi.org/10.1287/ijoc.3.4.376

Roberge, V., Tarbouchi, M., & Labonte, G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, *9*(1). https://doi.org/10.1109/TII.2012.2198665

Simonin, C., le Cadre, J. P., & Dambreville, F. (2009). A hierarchical approach for planning a multisensor multizone search for a moving target. *Computers and Operations Research*, *36*(7). https://doi.org/10.1016/j.cor.2008.08.007

Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., & Banaszak, Z. (2021). Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. *Neurocomputing*, *423*. https://doi.org/10.1016/j.neucom.2020.02.126

Sundar, K., & Rathinam, S. (2012). Route planning algorithms for unmanned aerial vehicles with refueling constraints. *Proceedings of the American Control Conference*. https://doi.org/10.1109/acc.2012.6315620

Tezcaner, D., & Köksalan, M. (2011). An Interactive Algorithm for Multi-objective Route Planning. *Journal of Optimization Theory and Applications*, *150*(2). https://doi.org/10.1007/s10957-011-9838-y

Vasquez-Gomez, J. I., Marciano-Melchor, M., Valentin, L., & Herrera-Lozada, J. C. (2020). Coverage Path Planning for 2D Convex Regions. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *97*(1). https://doi.org/10.1007/s10846-019-01024-y

Wagner, D. H. (1999). Naval operations analysis. Naval Inst. Press.

Wang, Y., Kirubarajan, T., Tharmarasa, R., Jassemi-Zargani, R., & Kashyap, N. (2018). Multiperiod Coverage Path Planning and Scheduling for Airborne Surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, *54*(5). https://doi.org/10.1109/TAES.2018.2812538

Sahm, V. H., & Werrell, K. P. (1987). The Evolution of the Cruise Missile. *Technology and Culture*, *28*(2). https://doi.org/10.2307/3105602

Wu, Y., Wu, S., & Hu, X. (2021). Cooperative Path Planning of UAVs UGVs for a Persistent Surveillance Task in Urban Environments. *IEEE Internet of Things Journal*, *8*(6). https://doi.org/10.1109/JIOT.2020.3030240

Yakıcı, E., & Karasakal, O. (2013). A min-max vehicle routing problem with split delivery and heterogeneous demand. *Optimization Letters*, *7*(7). https://doi.org/10.1007/s11590-012-0571-8

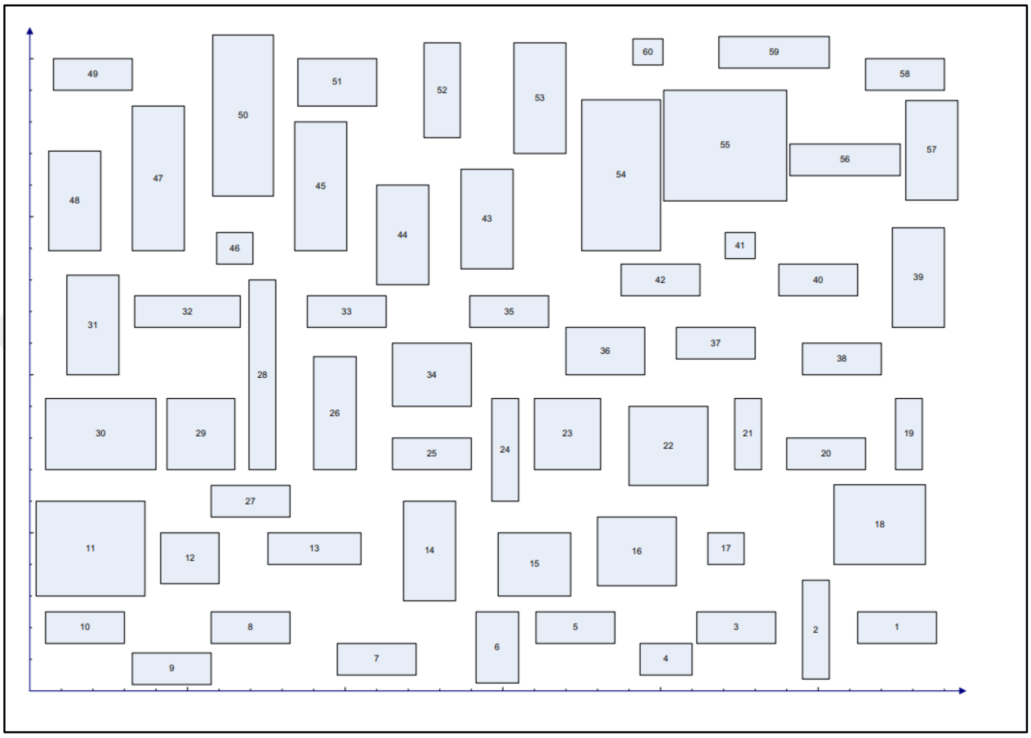
Yi, W., Sutrisna, M., & Wang, H. (2021). Unmanned aerial vehicle based low carbon monitoring planning. *Advanced Engineering Informatics*, *48*. https://doi.org/10.1016/j.aei.2021.101277

Zhen, L., Li, M., Laporte, G., & Wang, W. (2019). A vehicle routing problem arising in unmanned aerial monitoring. *Computers and Operations Research*, *105*. https://doi.org/10.1016/j.cor.2019.01.001

Zuo, Y., Tharmarasa, R., Jassemi-Zargani, R., Kashyap, N., Thiyagalingam, J., & Kirubarajan, T. T. (2020). MILP formulation for aircraft path planning in persistent surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(5). https://doi.org/10.1109/TAES.2020.2983532

APPENDICES

A. The rectangle set for generating problem instances (Karasakal 2016).



B. The Mathematical Model Results for Benchmark Instances.

Instance Name	C	Gap (%)	Value	Time (Sec.)	Status
IN_N7_M2_I1	79	0.00	149.729	7488.16	Optimal
IN_N7_M2_I2	58	0.00	107.932	9.46	Optimal
IN_N7_M2_I3	72	0.00	138.006	3.79	Optimal
IN_N7_M2_I4	87	0.00	168.406	9.66	Optimal
IN_N7_M2_I5	82	0.00	150.65	37.53	Optimal
IN_N10_M2_I1	82	0.00	158.185	65.13	Optimal
IN_N10_M2_I2	93	0.00	184.939	4703.91	Optimal
IN_N10_M2_I3	107	6.74	197.003	86410.95	Timelimit
IN_N10_M2_I4	103	0.00	132.677	2022.45	Optimal
IN_N10_M2_I5	100	6.46	156.027	86412.28	Timelimit
IN_N15_M2_I1	125	5.65	183.687	86410.27	Timelimit
IN_N15_M2_I2	98	4.36	179.951	86417.78	Timelimit
IN_N15_M2_I3	120	0.40	179.226	86407.3	Timelimit
IN_N15_M2_I4	126	3.23	149.747	86412.97	Timelimit
IN_N15_M2_I5	114	6.72	187.854	86409.75	Timelimit
IN_N20_M2_I1	123	0.00	230.046	22105.47	Optimal
IN_N20_M2_I2	136	4.52	238.431	86412.83	Timelimit
IN_N20_M2_I3	140	12.07	218.222	86413.53	Timelimit
IN_N20_M2_I4	147	0.00	226.037	16272.27	Optimal
IN_N20_M2_I5	150	5.11	220.725	86410.94	Timelimit
IN_N25_M2_I1	186	3.75	287.028	86424.33	Timelimit
IN_N25_M2_I2	160	9.10	235.976	86423.59	Timelimit
IN_N25_M2_I3	177	6.68	266.502	86420.03	Timelimit
IN_N25_M2_I4	170	8.08	250.858	86411.11	Timelimit
IN_N25_M2_I5	174	3.09	254.41	86219.27	Timelimit
IN_N7_M3_I1	78	0.00	155.937	294.81	Optimal
IN_N7_M3_I2	55	0.00	123.01	2.8	Optimal
IN_N7_M3_I3	68	0.00	182.121	5.61	Optimal
IN_N7_M3_I4	75	0.00	204.709	159.67	Optimal
IN_N7_M3_I5	80	0.00	164.066	5.01	Optimal
IN_N10_M3_I1	76	0.00	213.102	5719.78	Optimal
IN_N10_M3_I2	86	0.00	217.39	1147.66	Optimal
IN_N10_M3_I3	92	8.63	237.581	86410.06	Timelimit
IN_N10_M3_I4	80	0.00	161.149	75172.66	Optimal
IN_N10_M3_I5	89	0.00	168.3	519.05	Optimal

Appendix B Continued

IN_N15_M3_I1	96	14.67	217.867	86409.44	Timelimit
IN_N15_M3_I2	102	0.00	185.082	5274.27	Optimal
IN_N15_M3_I3	104	3.97	203.185	86409.78	Timelimit
IN_N15_M3_I4	104	2.96	165.123	86411.34	Timelimit
IN_N15_M3_I5	101	4.30	201.405	86409.22	Timelimit
IN_N20_M3_I1	118	0.00	259.052	68442.09	Optimal
IN_N20_M3_I2	130	3.93	258.05	86416.39	Timelimit
IN_N20_M3_I3	130	7.92	225.513	86417.77	Timelimit
IN_N20_M3_I4	117	0.00	249.9	74957.14	Optimal
IN_N20_M3_I5	135	0.56	226.916	86408.61	Timelimit
IN_N25_M3_I1	155	3.21	310.572	86415.81	Timelimit
IN_N25_M3_I2	148	4.98	240.607	86416.09	Timelimit
IN_N25_M3_I3	140	7.32	293.83	86411.22	Timelimit
IN_N25_M3_I4	150	6.64	267.783	86413.41	Timelimit
IN_N25_M3_I5	125	5.03	282.88	86415.2	Timelimit
IN_N7_M4_I1	75	0.00	186.096	23.21	Optimal
IN_N7_M4_I2	59	0.00	152.481	3.76	Optimal
IN_N7_M4_I3	63	0.00	223.453	2.79	Optimal
IN_N7_M4_I4	71	0.00	245.487	131.38	Optimal
IN_N7_M4_I5	75	0.00	225.651	40.05	Optimal
IN_N10_M4_I1	74	0.00	240.886	1138.92	Optimal
IN_N10_M4_I2	82	0.00	252.975	1228.59	Optimal
IN_N10_M4_I3	82	8.42	284.035	86410.77	Timelimit
IN_N10_M4_I4	76	0.00	170.757	32489.63	Optimal
IN_N10_M4_I5	78	0.00	197.616	310.2	Optimal
IN_N15_M4_I1	94	8.46	224.077	86412.01	Timelimit
IN_N15_M4_I2	101	0.00	207.679	33253.48	Optimal
IN_N15_M4_I3	104	0.00	224.054	15175.5	Optimal
IN_N15_M4_I4	97	8.96	193.814	86412.03	Timelimit
IN_N15_M4_I5	95	2.55	221.314	86410.78	Timelimit
IN_N20_M4_I1	105	3.31	292.402	86417.75	Timelimit
IN_N20_M4_I2	128	1.81	280.03	86416.98	Timelimit
IN_N20_M4_I3	130	5.67	239.883	86421.27	Timelimit
IN_N20_M4_I4	101	5.34	285.312	86216.08	Timelimit
IN_N20_M4_I5	125	0.25	245.636	86407.53	Timelimit
IN_N25_M4_I1	150	3.12	339.245	86415.83	Timelimit
IN_N25_M4_I2	131	4.06	255.13	86415.36	Timelimit
IN_N25_M4_I3	135	3.53	311.227	86412.16	Timelimit

Appendix B Continued

IN_N25_M4_I4	143	5.55	290.312	86410.39	Timelimit
IN_N25_M4_I5	130	0.89	294.987	86218.59	Timelimit
IN_N7_M5_I1	72	0.00	229.816	12.03	Optimal
IN_N7_M5_I2	55	0.00	182.63	2.95	Optimal
IN_N7_M5_I3	61	0.00	269.226	4.3	Optimal
IN_N7_M5_I4	68	0.00	275.713	60.95	Optimal
IN_N7_M5_I5	75	0.00	253.538	7.58	Optimal
IN_N10_M5_I1	74	0.00	278.54	1596.41	Optimal
IN_N10_M5_I2	79	0.00	280.869	1497.33	Optimal
IN_N10_M5_I3	75	1.95	319.563	86408.56	Timelimit
IN_N10_M5_I4	74	0.00	197.39	19516.09	Optimal
IN_N10_M5_I5	71	0.00	232.689	606.08	Optimal
IN_N15_M5_I1	105	5.85	238.754	86423.63	Timelimit
IN_N15_M5_I2	85	5.06	236.247	86413.98	Timelimit
IN_N15_M5_I3	87	0.00	256.121	45408.41	Optimal
IN_N15_M5_I4	85	15.80	235.74	86414.91	Timelimit
IN_N15_M5_I5	86	1.22	245.921	86413.34	Timelimit
IN_N20_M5_I1	92	6.02	333.398	86411.91	Timelimit
IN_N20_M5_I2	126	2.17	308.585	86426.58	Timelimit
IN_N20_M5_I3	123	3.92	262.379	86427.11	Timelimit
IN_N20_M5_I4	101	3.41	310.333	86215.67	Timelimit
IN_N20_M5_I5	115	7.76	282.72	86415.36	Timelimit
IN_N25_M5_I1	140	1.50	366.531	86414.98	Timelimit
IN_N25_M5_I2	126	3.52	276.005	86413.31	Timelimit
IN_N25_M5_I3	132	3.28	342.508	86414.59	Timelimit
IN_N25_M5_I4	135	2.41	308.965	86411.94	Timelimit
IN_N25_M5_I5	127	0.00	320.597	29720.36	Optimal

C. The Heuristic Method Results for Benchmark Instances

Problem Info.	Cplex		HBC		HBN	
Instance Name	Gap (%)	Time (Sec.)	Gapv.2 (%)	Time (Sec.)	Gapv.2 (%)	Time (Sec.)
IN_N7_M2_I1	0.00	7488.16	0	5.185	0.05	5.46
IN_N7_M2_I2	0.00	9.46	0	4.826	5.21	2.861
IN_N7_M2_I3	0.00	3.79	0	6.952	1.60	5.849
IN_N7_M2_I4	0.00	9.66	0	8.04	0.00	3.344
IN_N7_M2_I5	0.00	37.53	0	9.112	1.72	11.945
IN_N10_M2_I1	0.00	65.13	0	18.74	0.72	11.59
IN_N10_M2_I2	0.00	4703.91	0	23.818	0.49	10.689
IN_N10_M2_I3	6.74	86410.95	3.28	17.185	1.04	16.822
IN_N10_M2_I4	0.00	2022.45	3.55	12.946	3.60	5.497
IN_N10_M2_I5	6.46	86412.28	0.87	36.789	0.87	23.976
IN_N15_M2_I1	5.65	86410.27	18.9	529.969	8.61	179.437
IN_N15_M2_I2	4.36	86417.78	1.51	554.959	8.12	173.094
IN_N15_M2_I3	0.40	86407.3	5.27	39.619	4.41	21.12
IN_N15_M2_I4	3.23	86412.97	1.33	293.707	1.33	239.072
IN_N15_M2_I5	6.72	86409.75	-0.15	345.532	0.00	150.781
IN_N20_M2_I1	0.00	22105.47	0.53	325.897	1.69	344.129
IN_N20_M2_I2	4.52	86412.83	2.08	308.514	4.81	256.201
IN_N20_M2_I3	12.07	86413.53	2.92	593.992	2.56	184.344
IN_N20_M2_I4	0.00	16272.27	1.78	1862.109	3.16	497.713
IN_N20_M2_I5	5.11	86410.94	0.23	2182.167	5.16	1255.383
IN_N25_M2_I1	3.75	86424.33	0	1989.978	4.96	1622.182
IN_N25_M2_I2	9.10	86423.59	1.04	10042.668	-0.65	902.489
IN_N25_M2_I3	6.68	86420.03	-0.27	2902.476	0.44	601.131
IN_N25_M2_I4	8.08	86411.11	1.09	7076.514	2.50	1451.744
IN_N25_M2_I5	3.09	86219.27	6.77	11408.954	3.36	1548.481
IN_N7_M3_I1	0.00	294.81	0	4.209	26.68	8.682
IN_N7_M3_I2	0.00	2.8	0	0.155	0.00	0.259
IN_N7_M3_I3	0.00	5.61	0	2.613	0.12	1.442
IN_N7_M3_I4	0.00	159.67	0	3.26	0.23	1.282
IN_N7_M3_I5	0.00	5.01	0	6.359	1.72	4.839
IN_N10_M3_I1	0.00	5719.78	2.76	20.209	6.39	11.88
IN_N10_M3_I2	0.00	1147.66	6.44	11.085	6.87	10.137
IN_N10_M3_I3	8.63	86410.06	6.49	20.801	12.07	7.64

Appendix C Continued

IN_N10_M3_I4	0.00	75172.66	1.31	14.644	16.46	14.287
IN_N10_M3_I5	0.00	519.05	0	23.816	1.52	11.378
IN_N15_M3_I1	14.67	86409.44	0.59	271.387	3.97	50.702
IN_N15_M3_I2	0.00	5274.27	8.9	237.377	3.09	500.925
IN_N15_M3_I3	3.97	86409.78	2.91	50.752	2.82	31.912
IN_N15_M3_I4	2.96	86411.34	0.84	112.808	0.83	61.593
IN_N15_M3_I5	4.30	86409.22	0	270.091	2.07	96.673
IN_N20_M3_I1	0.00	68442.09	0.08	204.963	11.74	125.817
IN_N20_M3_I2	3.93	86416.39	5.86	869.599	10.23	670.607
IN_N20_M3_I3	7.92	86417.77	3.17	457.013	0.85	352.97
IN_N20_M3_I4	0.00	74957.14	12.47	807.884	12.66	174.695
IN_N20_M3_I5	0.56	86408.61	0	2019.609	23.92	769.918
IN_N25_M3_I1	3.21	86415.81	4.12	2145.108	6.00	2230.868
IN_N25_M3_I2	4.98	86416.09	4.49	7948.654	9.23	567.315
IN_N25_M3_I3	7.32	86411.22	-2.05	1094.574	1.13	778.909
IN_N25_M3_I4	6.64	86413.41	2.33	3672.828	-0.45	1330.254
IN_N25_M3_I5	5.03	86415.2	11.95	1614.813	14.82	848.264
IN_N7_M4_I1	0.00	23.21	0	2.653	0.74	1.544
IN_N7_M4_I2	0.00	3.76	0	0.273	0.00	0.55
IN_N7_M4_I3	0.00	2.79	0	0.184	0.00	0.285
IN_N7_M4_I4	0.00	131.38	0	5.106	1.56	1.528
IN_N7_M4_I5	0.00	40.05	0.58	2.904	0.61	1.144
IN_N10_M4_I1	0.00	1138.92	0	24.601	1.52	9.514
IN_N10_M4_I2	0.00	1228.59	0	9.232	0.66	4.164
IN_N10_M4_I3	8.42	86410.77	2.75	15.567	1.93	11.66
IN_N10_M4_I4	0.00	32489.63	16.67	14.611	18.57	7.296
IN_N10_M4_I5	0.00	310.2	2.78	74.662	6.50	12.109
IN_N15_M4_I1	8.46	86412.01	11.1	205.52	4.85	53.503
IN_N15_M4_I2	0.00	33253.48	6.16	167.117	0.75	186.518
IN_N15_M4_I3	0.00	15175.5	1.14	36.123	1.14	16.954
IN_N15_M4_I4	8.96	86412.03	3.42	136.623	7.49	38.575
IN_N15_M4_I5	2.55	86410.78	1.4	120.019	4.68	32.196
IN_N20_M4_I1	3.31	86417.75	7.11	221.402	9.80	73.881
IN_N20_M4_I2	1.81	86416.98	7.17	255.136	8.40	833.11
IN_N20_M4_I3	5.67	86421.27	3.28	350.017	0.68	366.542
IN_N20_M4_I4	5.34	86216.08	8.49	810.869	8.31	457.525
IN_N20_M4_I5	0.25	86407.53	0	391.563	0.00	783.427
IN_N25_M4_I1	3.12	86415.83	1.84	3625.733	3.18	1270.812
IN_N25_M4_I2	4.06	86415.36	6.5	5912.472	10.38	1087.926
IN_N25_M4_I3	3.53	86412.16	0.83	954.66	1.76	470.193

Appendix C Continued

IN_N25_M4_I4	5.55	86410.39	0.55	1944.686	-0.82	975.879
IN_N25_M4_I5	0.89	86218.59	12.23	1321.931	14.06	1730.291
IN_N7_M5_I1	0.00	12.03	0	2.231	0.00	1.247
IN_N7_M5_I2	0.00	2.95	0	0.395	0.00	0.311
IN_N7_M5_I3	0.00	4.3	0	5.757	0.85	2.363
IN_N7_M5_I4	0.00	60.95	0	2.079	0.00	2.377
IN_N7_M5_I5	0.00	7.58	3.28	1.958	3.28	0.738
IN_N10_M5_I1	0.00	1596.41	0	10.005	0.99	8.419
IN_N10_M5_I2	0.00	1497.33	0.33	11.278	0.28	2.612
IN_N10_M5_I3	1.95	86408.56	0	19.807	3.40	6.066
IN_N10_M5_I4	0.00	19516.09	5.77	14.083	5.81	9.081
IN_N10_M5_I5	0.00	606.08	3.62	10.792	3.62	7.877
IN_N15_M5_I1	5.85	86423.63	8.75	247.427	11.19	124.949
IN_N15_M5_I2	5.06	86413.98	1.97	91.611	7.59	26.707
IN_N15_M5_I3	0.00	45408.41	1.2	30.758	3.90	24.096
IN_N15_M5_I4	15.80	86414.91	9.68	120.211	9.61	16.957
IN_N15_M5_I5	1.22	86413.34	7.84	64.535	8.25	35.902
IN_N20_M5_I1	6.02	86411.91	13.14	202.055	6.46	9.656
IN_N20_M5_I2	2.17	86426.58	3.69	178.801	7.80	3.578
IN_N20_M5_I3	3.92	86427.11	2.37	445.184	7.66	6.32
IN_N20_M5_I4	3.41	86215.67	9.61	436.085	7.12	17.218
IN_N20_M5_I5	7.76	86415.36	0.26	681.499	0.75	376.387
IN_N25_M5_I1	1.50	86414.98	3.77	3286.715	5.45	245.462
IN_N25_M5_I2	3.52	86413.31	4.45	5146.819	12.85	1295.574
IN_N25_M5_I3	3.28	86414.59	0.06	948	0.26	499.713
IN_N25_M5_I4	2.41	86411.94	0.86	1642.166	1.88	1329.855
IN_N25_M5_I5	0.00	29720.36	10.1	1680.321	10.55	593.239