UNMANNED AIR VEHICLE ROUTING WITH MULTIPLE OBJECTIVES

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

BY

ERDİ DAŞDEMİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

JULY 2021

Approval of the thesis:

UNMANNED AIR VEHICLE ROUTING WITH MULTIPLE OBJECTIVES

submitted by **ERDİ DAŞDEMİR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** i**n 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. Meral Azizoğlu
Supervisor, **Industrial Engineering, METU** _____

Assist. Prof. Dr. Diclehan Tezcaner Öztürk
Co-Supervisor, **Industrial Eng., Hacettepe University** _____

Examining Committee Members:

Prof. Dr. Sinan Gürel
Industrial Engineering, METU _____

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU _____

Prof. Dr. Murat Köksalan
Ross School of Business, UM Ann Arbor _____

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

Assist. Prof. Dr. Vedat Bayram
Industrial Engineering, TED University _____

Date: 29.07.2021

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 : Erdi Daşdemir

Signature :

ABSTRACT

UNMANNED AIR VEHICLE ROUTING WITH MULTIPLE OBJECTIVES

Daşdemir, Erdi
Doctor of Philosophy, Industrial Engineering
Supervisor: Prof. Dr. Meral Azizoğlu
Co-Supervisor: Assist. Prof. Dr. Diclehan Tezcaner Öztürk

July 2021, 147 pages

Unmanned Aerial Vehicles (UAVs) are special types of aircraft operating without a human pilot on board. In this thesis, the multi-objective UAV route planning is addressed by studying three different problems. In the first study, the bi-objective route planning problem of a UAV that is tasked with visiting all targets located in an enemy region monitored by radars is addressed. The aim is to determine both the visiting order of the targets and the specific trajectories to use between consecutive target pairs that minimize the total flight duration and risk of being detected. The terrain the UAV moves is considered as continuous terrain allowing infinitely many trajectory options between targets. We develop a generic preference-based multi-objective evolutionary algorithm (MOEA) that can be used for any multi-objective optimization problem, and adapt it to the UAV route planning problem. In the second study, the first study is extended by considering an additional objective that is maximizing the collected information. We address this problem adapting the MOEA developed in the first study. In both studies, the results show that the developed algorithm is able to converge to preferred regions of the Pareto-optimal frontier,

provide a general idea on the structure and positioning of the entire Pareto-optimal frontier, and adapt to the changes in preferences quickly.

In the third study, the UAV route planning problem is considered as a multi-objective, multi-connection orienteering problem with time dependent prizes. A route plan involves the decisions of which targets to visit, the order of visit to the selected targets, and the trajectories to follow between consecutively-visited targets. The objectives are maximizing the collected information from the operation area while minimizing the mission time and the radar detection threat on the entire route. We discretize the continuous movement space of the UAV by representing infinitely many efficient trajectories between target pairs with a finite trajectory set. We first formulate a mixed integer programming (MIP) model and then develop a hybrid algorithm to decrease the computational requirements of the model. The hybrid algorithm consists of a heuristic route search algorithm that approximates the optimal solution and a more manageable MIP model that finds some bounds for the optimal objective function value. The results show that the developed algorithm is able to reduce the computational requirements of the exact model. A demonstration on the Colorado state of the U.S. is performed as a case study.

Keywords: Multi-objective Programming, Unmanned Air Vehicles, UAV Routing, Evolutionary Optimization, Mixed Integer Programming

ÖZ

ÇOK AMAÇLI İNSANSIZ HAVA ARACI ROTALAMA

Daşdemir, Erdi
Doktora, Endüstri Mühendisliği
Tez Yöneticisi: Prof. Dr. Meral Azizoğlu
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi Diclehan Tezcaner Öztürk

Temmuz 2021, 147 sayfa

İnsansız Hava Araçları (İHA), içinde insan pilot olmadan çalışan özel uçak türleridir. Bu tezde, çok amaçlı İHA rota planlaması üç farklı problem çalışılarak ele alınmaktadır. İlk çalışmada, radarlar tarafından izlenen bir düşman bölgesinde bulunan tüm hedefleri ziyaret etmekle görevlendirilen bir İHA'nın iki amaçlı rota planlama problemi ele alınmıştır. Problemin amacı toplam uçuş süresini ve tespit edilme riskini en aza indirecek şekilde hedeflerin ziyaret sıralarına ve ardışık hedefler arasında kullanılan güzergahlara karar vermektir. İHA'nın hareket ettiği arazi, hedefler arasında sonsuz sayıda bağlantı seçeneğine izin veren sürekli uzay olarak kabul edilmektedir. Problemin çözümü için, herhangi birçok amaçlı optimizasyon problemi için de kullanılabilecek genel bir tercih tabanlı çok amaçlı evrimsel algoritma geliştirmekte ve bunu İHA rota planlama problemine uyarlamaktayız. İkinci çalışmada, hedef bölgelerden bilgi toplamayı maksimize etme ek bir amaç fonksiyonu olarak göz önünde bulundurulmakta ve ilk çalışmadaki problem genişletilmektedir. Problemin çözümü için ilk çalışmada geliştirilen evrimsel algoritma problemin özelliklerine göre uyarlanmaktadır. Her iki çalışmada

da sonuçlar, geliştirilen algoritmanın Pareto-optimal yüzeyin tercih edilen bölgelerine yakınsayabildiğini, tüm Pareto-optimal yüzeyin yapısı ve konumu hakkında genel bir fikir sağladığını ve tercihlerdeki değişikliklere hızlı bir şekilde uyum sağlayabildiğini göstermektedir.

Üçüncü çalışmada, İHA rota planlama problemi, zamana bağlı ödülleri olan çok amaçlı, çok bağlantılı bir oryantiring problemi olarak ele alınmıştır. Bu problem için bir rota planı, hangi hedeflerin ziyaret edileceğine, seçilen hedeflere ziyaret sırasına ve ardışık olarak ziyaret edilen hedefler arasında izlenecek güzergahlara ilişkin kararları içerir. Amaçlar, toplam uçuş süresini ve radar algılama tehdidini en aza indirirken, operasyon alanından toplanan bilgiyi en üst düzeye çıkarmaktır. Bu problem ele alınırken, hedefler arasındaki sonsuz sayıda bağlantı seçeneğini temsil edecek bir bağlantı seti ile İHA'nın sürekli hareket alanı ayrıklaştırılmaktadır. Çözüm için ilk olarak bir karma tam sayılı programlama modeli formüle edilmekte ve ardından modelin hesaplama gereksinimlerini azaltmak için bir hibrit algoritma geliştirilmektedir. Hibrit algoritma, optimal çözüme yaklaşan bir sezgisel rota arama algoritmasından ve optimal amaç fonksiyonu değeri için bazı sınırlar bulan ve hesaplama gereksinimleri açısından daha yönetilebilir bir MIP modelinden oluşmaktadır. Sonuçlar, geliştirilen algoritmanın kesin modelin hesaplama gereksinimlerini azalttığını göstermektedir. Geliştirilen çözüm yöntemleri ABD'nin Colorado eyaletindeki bir olay incelemesi ile örnek olarak gösterilmektedir.

Anahtar Kelimeler: Çok Amaçlı Programlama, İnsansız Hava Araçları, İHA Rotalama, Evrimsel Optimizasyon, Karma Tam Sayılı Programlama

To my beloved family, professors, friends,

and everyone who has been a part of my journey.

Thank you for being constant sources of support and encouragement.

ACKNOWLEDGMENTS

My Ph.D. years has been an incredible evolutionary experience and I am indebted to many people for it. I apologize in advance to the ones whose names are not mentioned in this limited page.

First and foremost, I would like to express my deepest appreciation to my family members. They mean the world to me and without them, neither this study nor my life would be complete. I have to start by thanking my wonderful wife and best friend Sevinç Daşdemir for her endless support, patience and tolerance over the years. I would like to extend my thanks to my parents Prof. Dr. İsmet Daşdemir and Ayşe Daşdemir, and my sister Sevgi Daşdemir for their invaluable and constant support and love.

I extend my sincerest appreciation to Prof. Dr. Murat Köksalan and Assist. Prof. Dr. Diclehan Tezcaner Öztürk for their guidance, support and contributions throughout my Ph.D. study. Their exceptional mentorship enabled me not only write this thesis and contribute to my research area, but also establish strong theoretical, practical and ethical foundations as a young candidate planning a successful academic career. I also would like to express my sincere thanks to Prof. Dr. Rajan Batta from Department of Industrial Engineering, The State University of New York at Buffalo, for inviting me to his lab and giving me the opportunity to collaborate with him on this thesis throughout 2019. I am very fortunate to be able to work with these three professors, who are figures of ideal academicians.

I am so very thankful to Prof. Dr. Meral Azizoğlu for her time, valuable feedbacks, and dealing with all the procedures of my thesis. She has always had a positive attitude and been very helpful in resolving all of my problems. I would like to thank Assoc. Prof. Dr. Mustafa Kemal Tural too for the time and insightful feedbacks he has given throughout my thesis. Finally, I am also grateful to Prof. Dr. Murat Caner

Testik for his guidance on my life, friendship and valuable conversations we have had over the years.

I would like to express my gratitude to the faculty and staff of the Industrial Engineering Departments of Middle East Technical University and Hacettepe University. Very high-quality education is offered in both universities. I really mean it. When I went to U.S. to do research on my thesis, I understood better how well I was educated. I hope the students of these departments are aware of how lucky they are.

I presented earlier versions of the chapters of this thesis at the Brazil Summer School, 2016, Ottowa, 2017 and İstanbul, 2019 conferences of International Society on MCDM; İstanbul, 2017, Eskişehir, 2018 and Ankara, 2019 YAEM conferences of Operational Research Society of Turkey, and Nashville, 2016 and Seattle, 2019 conferences of INFORMS. At these events, I have met very valuable people and made good friends. I am thankful for the committees, moderators, and conference attendees for their feedbacks on my thesis and making those events inspiring.

Several grants helped me throughout this thesis. I would like to thank Council of Higher Education (YÖK) for providing me YUDAB scholarship so I could work as a visiting scholar at University at Buffalo. I would like to thank Scientific and Technological Research Council of Turkey (TUBITAK) for the funding they provided for my Ph.D. study under grant 2211. The third chapter of the thesis is based upon work supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF. Finally, I would like to thank Hacettepe University Scientific Research Projects Office for the funding they provided for me to attend the conferences.

Last but not the least, I would like to close with thanking to my cousins and friends. I will never forget our camping adventures, one of the rare places where I took a break from thinking about my studies. Thank you all for letting me spend my limited free time productively and be more motivated for this thesis.

TABLE OF CONTENTS

ABSTRACT v

ÖZ… vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS x

LIST OF TABLES xvii

LIST OF FIGURES xviii

CHAPTERS

1 INTRODUCTION 1

1.1 First Study: Bi-Objective UAV Route Planning in Continuous Terrain and Development of a Preference-Based MOEA 2

1.2 Second Study: Three-Objective UAV Route Planning in Continuous Terrain and Adaptation of FREA 3

1.3 Third Study: Multi-Objective UAV Route Planning: An Orienteering Problem with Time-Dependent Prizes and Multiple Trajectories 4

2 LITERATURE REVIEW 7

2.1 Basic Definitions 7

2.2 UAV Route Planning Problem 8

3 A PREFERENCE-BASED MOEA AND ITS IMPLEMENTATION TO BI-OBJECTIVE UAV ROUTE PLANNING IN CONTINUOUS TERRAIN 13

3.1 Problem Structure of the Bi-Objective UAV Route Planning Problem ... 15

3.1.1 Terrain Structure of Bi-Objective UAV Route Planning Problem 15

3.1.2 Objectives of the Bi-Objective UAV Route Planning Problem 16

3.1.3 Movement Between Targets 16

3.2 Flexible Reference Point-Based Evolutionary Algorithm (FREA)..... 20
3.2.1 Preference-Based Multi-objective Evolutionary Algorithms 20
3.2.2 Details and Steps of FREA 22
3.2.2.1 Nondominated Sorting, Preference Ranking, and ε-*niching* 24
3.2.2.2 Selection 25
3.2.2.3 Archive Mechanism 26
3.2.2.4 Flexible Reference Points 28
3.2.3 Comparison of FREA with R-NSGA-II 28
3.2.3.1 Comparison on ZDT3 Problem 31
3.2.3.2 Changing Reference Points on 3-objective DTLZ2 32
3.3 Implementing FREA on UAV Route Planning 33
3.3.1 Representation of Solutions 33
3.3.2 Assigning Fitness Values 33
3.3.2.1 Fitting L_q Function to the Pareto-optimal Frontier of Tour π ... 34
3.3.2.2 Finding z_{ijCP} 35
3.3.2.3 Selecting *s* Representative Points on the Pareto-Optimal Frontier of Tour π 38
3.3.3 Crossover and Mutation 38
3.4 Finding Trajectories of the Nondominated Solutions 40
3.5 Results of the Bi-Objective UAV Route Planning Problem 41
3.5.1 5-Target Bi-Objective UAV Route Planning Problem 42
3.5.1.1 Effects of Different ε Values 42
3.5.1.2 Demonstration of FREA_UAV on the 5-Target Problem 44

3.5.1.3 Comparison FREA_UAV and R-NSGA-II on the 5-Target Problem 46

3.5.2 9-Target Bi-Objective UAV Route Planning Problem 47

3.5.2.1 Demonstration of FREA_UAV on the 9-Target Problem 48

3.5.2.2 Comparison of FREA_UAV and R-NSGA-II on the 9-Target Problem 50

3.5.3 15-Target Bi-Objective UAV Route Planning Problem 51

3.5.4 Updating Reference Points 52

3.5.4.1 9-Target Scenario 1 52

3.5.4.2 9-Target Scenario 2 54

3.6 Discussions for Chapter 3 55

4 THREE-OBJECTIVE UAV ROUTE PLANNING PROBLEM WITH TIME-DEPENDENT PRIZES AND IMPLEMENTATION OF FREA 57

4.1 Problem Structure of Three-Objective UAV Route Planning Problem 59

4.2 Implementing FREA_UAV on the 3-Objective UAV Route Planning Problem 61

4.2.1 Assigning Fitness Values with Three Objectives 62

4.2.1.1 Fitting an L_q Function to the Pareto-optimal Frontier of Tour π with Three Objectives 63

4.2.1.2 Heuristic Approximating $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$ for Tour π 64

4.2.1.3 Finding a Central Point with Three Objectives 66

4.2.1.4 Selecting *s* Representative Points on the Pareto-Optimal Frontier of Tour π with Three Objectives 67

4.2.2 Finding Trajectories of the Nondominated Solutions of Three-Objective Problem 68

4.3 Results of the Three-Objective UAV Route Planning Problem 69
4.3.1 DM's Reference Point: *(D=1800, RDT=50, I=15)* 70
4.3.2 DM's Reference Point: *(D=2200, RDT=200, I=5)* 71
4.3.3 Updating Reference Points 72
4.4 Discussions for Chapter 4 74
5 UAV ROUTE PLANNING AS A MULTI-OBJECTIVE ORIENTEERING PROBLEM 77
5.1 Orienteering Problem 79
5.2 Problem Context 81
5.2.1 Problem Definition and Terrain Structure 81
5.2.2 Time-dependency in Collected Information 82
5.2.3 Movement Between Target Pairs 83
5.2.4 Demonstration on a Small Example 83
5.3 MIP Formulation 84
5.3.1 Finding Efficient Routes 87
5.3.2 Properties of the Model 88
5.3.3 Complexity of the Problem 90
5.4 A Hybrid Algorithm 90
5.4.1 Probabilistic Route Search Heuristic 92
5.4.1.1 Construction Heuristic 93
5.4.1.2 Trajectory Exchange Improvement 94
5.4.1.3 Node Insertion Improvement 96
5.4.2 Performance Analysis of PRSH 97
5.4.3 Finding an Alternative Upper Bound 100

5.5 Computational Experiments 103
5.5.1 Problem Instances and Parameter Settings 104
5.5.2 Limits of the Exact Model 105
5.5.3 Experimenting with HA 105
5.5.3.1 Experiments on the 15-node Problem 106
5.5.3.2 Experiments on the 30-node Problem 108
5.5.4 Generating the Pareto-optimal Frontiers of the Problem Instances. 110
5.5.4.1 The 15-node Problem 110
5.5.4.2 The 30-node Problem 111
5.6 Demonstration on the Colorado Case Study 112
5.6.1 Problem Instance 1 114
5.6.2 Problem Instance 2 117
5.7 Discussions for Chapter 5 119
6 CONCLUSIONS 123
REFERENCES 127
APPENDICES
A. Radar Detection Threat Measure 137
B. Nonlinear Mathematical Model to Find the Trajectories Used Between Consecutively Visited Targets 139
B.1 Finding Trajectories Used Between Consecutively Visited Targets for a Given Tour for the Bi-Objective UAV Route Planning Problem 139
B.2 Finding Trajectories Used Between Consecutively Visited Targets for a Given Tour for the Three-Objective UAV Route Planning Problem 141
CURRICULUM VITAE 145

LIST OF TABLES

Table 3.1: Comparison of FREA and R-NSGA-II on test problems 30
Table 3.2: Trajectories between target pairs of the closest solutions to reference points (*D*=56.0, *RDT*=2.0) and (*D*=52.0, *RDT*=10.0) 46
Table 3.3: Trajectories between target pairs of the closest solutions to reference points (*D*=60.0, *RDT*=5.0) and (*D*=68.0, *RDT*=1.5) 49
Table 4.1: Trajectories between target pairs of the closest solutions to reference point *(D=1800, RDT=50, I=15)* 71
Table 4.2: Trajectories between target pairs of the closest solution to the reference point *(D=2200, RDT=200, I=5)* 72
Table 5.1: Mathematical notation 85
Table 5.2: Mathematical notation for HA 92
Table 5.3: Average performance of PRSH on the 15-node problem 99
Table 5.4: Average performance of PRSH on the 30-node problem 100
Table 5.5: Run times of the GUROBI solver for Model *S* on 15 and 30-node problems 105
Table 5.6: Results of Model *S* and HA for the 15-node problem 107
Table 5.7: Further investigation of HA for the 15-node problem 108
Table 5.8: Results of Model *S* and HA on the 30-node problem 109
Table 5.9: Further investigation of HA for the 30-node problem 109
Table 5.10: Results of Model *S* and HA on the Colorado Problem Instance 1 115
Table 5.11: Further investigation of HA for the Colorado Problem Instance 1 ... 116
Table 5.12: Results of Model *S* and HA on the Colorado Problem Instance 2 117
Table 5.13: Further investigation of HA for the Colorado Problem Instance 2 ... 118

LIST OF FIGURES

Figure 3.1: An example terrain structure 16
Figure 3.2: Movement Type 2 17
Figure 3.3: Movement Type 3 18
Figure 3.4: Results for ZDT3 31
Figure 3.5: Changing reference points in FREA for DTLZ2 32
Figure 3.6: Threshold weight 37
Figure 3.7: Finding z_{ijCP} 37
Figure 3.8: Selecting representative points 38
Figure 3.9: POS crossover 39
Figure 3.10: Mutation operator 39
Figure 3.11: The effects of different ε values on the distribution of the final set for the 5-Target UAV route planning problem 43
Figure 3.12: The effects of using different ε values for archive for the 5-Target UAV route planning problem 44
Figure 3.13: Results for the 5-Target UAV route planning problem 45
Figure 3.14: Results of R-NSGA-II for 5-Target UAV route planning problem 47
Figure 3.15: Results for the 9-Target UAV route planning problem 49
Figure 3.16: Comparison of R-NSGA-II and FREA_UAV on the 9-Target UAV route planning problem 50
Figure 3.17: Results for the 15-Target UAV route planning problem 52
Figure 3.18: Changing reference points - First scenario 53
Figure 3.19: Changing reference points - Second scenario 54
Figure 4.1: An example showing information change over time 60
Figure 4.2: The terrain of the 3-objective 5-target UAV route planning problem 70
Figure 4.3: Results of the 3-objective 5-Target UAV route planning problem for the reference point *(D=1800, RDT=50, I=15)* 71
Figure 4.4: Results of the 3-objective 5-Target UAV route planning problem for the reference point (D=2200, RDT=200, I=5) 72

Figure 4.5: Changing reference points: 3-Objective 5-Target UAV route planning problem 73
Figure 5.1: Problem context: (a) Radars and targets, (b) Information vs visit time, (c) Movement types 82
Figure 5.2: Several routes and the Pareto-optimal set of an example problem 84
Figure 5.3: An example of the modified 2-opt improvement heuristic 95
Figure 5.4: An example for node insertion improvement 97
Figure 5.5: Problem instances: (a) 15-node problem, (b) 30-node problem..... 104
Figure 5.6: Approximation of the Pareto-optimal set of the 15-node problem 111
Figure 5.7: Approximation of the Pareto-optimal set of the 30-node problem 112
Figure 5.8: The map of Colorado's 58 fourteeners and 41 water reservoirs 114
Figure 5.9: Colorado Problem Instance 1: (a) Map, (b) Route of the selected solution, (c) Pareto-optimal set. 116
Figure 5.10: Colorado Problem Instance 2: (a) Map, (b) Route of the selected solution, (c) Pareto-optimal set. 118
Figure 5.11: Discrete vs continuous information change 121

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are special types of aircraft operating without a human pilot on board. They have been used for military, civilian, and commercial applications since they are able to perform low risk and cost-efficient operations compared to the manned aircrafts. In the civilian context, they are used for surveillance operations such as inspection of wildlife, search and rescue, crowd monitoring, and border patrolling. In the military context, they are often used in aerial reconnaissance operations to collect intelligence at regions that are outside the control of friendly forces. Since these vehicles fly unmanned, they follow the routes stored in their memories and remote interventions on their routes during flights are limited. Therefore, efficient route plans that reduce the operational costs and failure risks need to be prepared.

In this thesis, we address the route planning problem of a UAV that is responsible from exploration of a radar-monitored terrain to collect information about natural features and adversarial activities in the area. Variants of this problem are studied throughout the thesis. First, we address a bi-objective UAV route planning problem considering the movement terrain of the UAV as continuous terrain. We develop a generic preference-based multi-objective evolutionary algorithm (MOEA) and adapt it to the UAV routing problem. Second, we extend the first study to a three-objective UAV route planning problem by considering an additional objective function. We modify the MOEA we developed in the first study to address the problem. In these two studies, all targets must be visited in a mission. Hence, the visiting order of targets and the trajectories to be used between targets are determined in a route plan. Third, we study the UAV route planning problem as a multi-objective orienteering problem where a subset of targets can be visited in a mission. Hence, a route plan

now requires determining the subset of targets to be visited, order of visit to the selected targets, and the trajectories to be used between targets. Time-dependent prizes, multiple connections and three objectives are considered while searching for the optimal route plans. In the following subsections, we summarize these studies, each of which will be described as a separate chapter later.

This thesis is organized as follows. In Chapter 2, the literature is reviewed and some definitions that will be used throughout the thesis are provided. In Chapter 3, the bi-objective UAV route planning problem is explained, a preference-based MOEA is developed, and its implementation on the bi-objective UAV routing problem is explained. In Chapter 4, the bi-objective UAV routing problem is extended to a three-objective problem considering an additional routing objective, and the developed preference-based MOEA is adapted for the three-objective problem. In Chapter 5, a multi-objective orienteering problem with an application to the multi-objective UAV route planning problem is studied. Finally, the thesis is concluded in Chapter 6.

1.1 First Study: Bi-Objective UAV Route Planning in Continuous Terrain and Development of a Preference-Based MOEA

In this study, the route planning problem of a UAV tasked with visiting all targets located in an enemy region monitored by radars is addressed. The problem can be classified as a traveling salesperson problem (TSP) in the broader sense, and a route plan requires to determine both the visiting order (tour) of the targets and the specific trajectories to be used between consecutive target pairs under two objectives: (1) minimization of the length of the tour, and (2) minimization of the total radar detection threat the UAV is exposed to during the tour. The major distinction of this work from the UAV route planning literature is modeling the movement terrain of the UAV as continuous terrain and allowing infinitely many trajectory options between targets. Two conflicting objectives and the continuous terrain lead the Pareto-optimal frontier of the problem to have infinitely many efficient solutions. Generating all those solutions is neither computationally practical nor meaningful.

As a solution, we develop a preference-based MOEA, Flexible Reference Point Based Multi-Objective Evolutionary Algorithm (FREA), that decreases the computational requirements by focusing to the preferred regions of the Pareto-optimal frontier. In addition to the preferred solutions, the algorithm is able to find an approximation of the whole Pareto-optimal frontier, which provides a general idea on the structure and positioning of the Pareto-optimal frontier. Users have flexibility to change their reference points throughout the algorithm.

The research contributes to the literatures of MOEA and UAV route planning. FREA is a generic preference-based MOEA, and can be adapted for any multi-objective optimization problem. In terms of UAV route planning, multiple-objectives and continuous movement terrain are considered, and a preference-based MOEA is implemented. This study is based upon work supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF under Award No. FA9550-16-1-0 0 05. We also published a paper from this work in Computers & Operations Research (Dasdemir et al., 2020). Details of this study are presented in Chapter 3.

1.2 Second Study: Three-Objective UAV Route Planning in Continuous Terrain and Adaptation of FREA

In this work, we extend the first study by introducing a new objective function that is maximizing the collected information. Route plans are now prepared considering three objectives: (1) minimization of the length of the tour, (2) minimization of the total radar detection threat the UAV is exposed during the tour, (3) maximization of the collected information. As all targets must be visited in a mission, the total collected information would be the same in all plans if the available information at targets is fixed. However, we consider time-dependent information at the targets. The information at a target varies throughout a day, and therefore the information collected from a target depends on the time the target is visited. Time-dependent information concept represents the practice better, is a more generalized version of

the fixed prizes case, and make information collection objective meaningful for a TSP-type problem. Similar to the previous study, the terrain that the UAV moves is modeled as continuous terrain. Three-objectives and continuous terrain lead to infinitely many trajectory options. We modify FREA to account for the additional objective and use it to approximate the preferred regions of the Pareto-optimal frontier of the problem. The results show that FREA works well for 3-objective UAV routing problem too.

Contributions of this work is similar to those of the first study. It contributes to the literature of UAV route planning problem considering continuous terrain, three-objectives, and time-dependent prizes. It contributes to the literature of MOEA as an implementation of a preference-based MOEA on a complex optimization problem with three objectives. The details of this work are reported in Chapter 4.

1.3 Third Study: Multi-Objective UAV Route Planning: An Orienteering Problem with Time-Dependent Prizes and Multiple Trajectories

In this problem, we address the route optimization problem of a UAV that is tasked with the exploration of a radar-monitored environment to maximize the collected information while minimizing the mission time and the radar detection threat on the entire route. Unlike the previous two studies where the route planning problems are modeled as variants of TSP, now the problem is modeled as an orienteering problem (OP) where a subset of targets may be visited in an operation due to time and radar threat considerations. The UAV starts its movement from a home base, visits a subset of targets and finishes its movement at a destination base. The problem is a combination of selecting the targets to be visited, determining the sequence of visited targets and the trajectories to follow between target pairs. The information that can be obtained at each target varies with time. We discretize the movement terrain of the UAV by representing infinitely many efficient trajectories between two targets with a finite trajectory set. For the solution, we first develop a mixed integer programming (MIP) model and then employ the ε −constraint method to generate

the Pareto-optimal frontier. We then develop a hybrid algorithm that consists of a heuristic route search algorithm that approximates the optimal solution and an MIP model that finds some bounds for the optimal objective function value with less computational burden. In our computational experiments, HA decreases the computational requirements of the initial MIP model significantly. The algorithm Finally, the solution approaches are demonstrated on a case study developed according to the terrain properties of the Colorado state of the U.S.

This research contributes to the literatures of UAV route planning problem and orienteering problem. From the UAV routing perspective, a UAV prize collection problem in a radar-monitored terrain with time-dependent prizes and multiple connections is studied. From the OP literature perspective, the OP with multiple objectives, time-dependent prizes and multiple connections is addressed.

This is a joint work with Dr. Rajan Batta, with whom we have studied during Erdi Dasdemir's research visit at State University of New York at Buffalo from January, 2019 to December, 2019. Details of this work are presented in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

In this section, we first review some basic definitions of multi-objective optimization that are used throughout the thesis. We then review the UAV route planning problem literature that is relevant to the entire thesis. In addition to the information in this section, the relevant literatures of MOEA and OP are reviewed in Chapters 3 and 5, respectively.

2.1 Basic Definitions

Following definitions are adapted from Köksalan and Tezcaner Öztürk (2017). Let $\boldsymbol{x}$ denote the decision variable vector and $\boldsymbol{z}(\boldsymbol{x}) = \left(z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x})\right)$ be the objective function vector corresponding to the decision vector $\boldsymbol{x}$, where $z_k(\boldsymbol{x})$ is the performance of solution $\boldsymbol{x}$ in the *k*th objective, and *p* is the number of objectives. *X* and *Z* denote the corresponding feasible sets in the solution and objective spaces, respectively. We assume, without loss of generality, that all objectives are minimized.

- A solution $\boldsymbol{x} \in X$ is said to be efficient if there does not exist $\boldsymbol{x}' \in X$ such that $z_k(\boldsymbol{x}') \leq z_k(\boldsymbol{x})$ for all objectives and $z_k(\boldsymbol{x}') < z_k(\boldsymbol{x})$ for at least one objective. If there exist such an $\boldsymbol{x}'$, $\boldsymbol{x}$ is said to be inefficient. Efficient set consists of all the efficient solutions.
- If $\boldsymbol{x}$ is efficient, then $\boldsymbol{z}(\boldsymbol{x})$ is said to be nondominated or Pareto-optimal, and if $\boldsymbol{x}$ is inefficient, then $\boldsymbol{z}(\boldsymbol{x})$ is said to be dominated. The Pareto-optimal set (frontier) consists of all Pareto-optimal points.

- An efficient solution (a Pareto-optimal point) that has the minimum possible value in at least one of the objectives is called an extreme efficient solution (an extreme Pareto-optimal point).
- We refer to the solutions that are nondominated relative to the other population members at a generation of a MOEA as relatively nondominated solutions. A relatively nondominated solution may not be in the Pareto-optimal set of the problem, i.e., a relatively nondominated solution in a generation can be dominated by new solutions in the following generations.

2.2 UAV Route Planning Problem

The idea of developing UAVs originated from the military field a long time ago (see, Keane and Carr, 2013 for a brief history). Today's UAV technology allows them to be frequently and effectively used in the practice. Although UAVs are still usually controlled by ground pilots, they are expected to follow the routes stored in their memories with limited remote interventions on their routes during flights in near future. These uncover the problem of finding optimum route plans.

The UAV route planning literature can be considered as in its early stages. As optimization researchers have advance knowledge on the routing of ground vehicles, they tend to model their UAV routing problems similar to well-known problems developed for ground vehicle routing. For example, Mittal and Deb (2007), Moskal and Batta (2017) and Ergezer and Leblebicioglu (2013) model their UAV routing problems as shortest path problem, orienteering problem, and traveling salesperson problem, respectively. Yet, the UAV routing problem has distinctive characteristics. For example, UAVs operate in 3D continuous space, have various physical limitations, and are subject to flight dynamics. In missions performed in enemy terrains, they are subject to the risk of being detected which causes either the UAV to be shot down or enemies to take actions. Current research stream assumes certain simplifications and focus on the routing problem from an optimization perspective.

We refer to Coutinho et al. (2018) for a review of the existing literature of the UAV route planning problem, and focus here on the relevant studies to this thesis.

The literature can be classified from different perspectives. First issue in the UAV routing is modeling the movement of UAVs. Since UAVs move in continuous terrain, they have infinitely many movement options when moving from one point to another. There are three main approaches in the literature to model the movement of UAVs. In the first approach, mission terrains are discretized using a grid structure where the movement is assumed on the edges. This allows UAVs to have a limited number of movement options between two points. Since this approach simplifies the terrain and allows researchers to address the UAV routing problem similar to the routing of ground vehicles, it has been popular in the literature. Some early studies use a grid structure when routing between only two points (see, for example, Olsan, 1993; Gudaitis, 1994) . Some recent studies generalized these approaches to routing with multiple targets (see, for example, Tezcaner and Köksalan, 2011; Tezcaner Öztürk and Köksalan, 2016). In the second approach, the movement terrain is considered as continuous terrain and UAVs are allowed to have infinite movement options between any two locations. As this approach increases the complexity of the problem, it has been rarely used in the literature. Pfeiffer et al. (2009) consider continuous terrain for a bi-objective UAV routing problem but the movement is limited to between only two bases. Tezcaner Öztürk and Köksalan (2018) study a bi-objective UAV route planning problem with multiple targets in continuous terrain and develop exact and heuristic approaches to approximate the efficient sets of the movement between target pairs. Finally, there is a third approach that is between the first two extremes. In this approach, the waypoints where UAVs change direction of their movements on their routes are used. There are different approaches in using waypoints as well. Some studies assume that the locations of the waypoints are known. For example, Moskal and Batta (2017) and Moskal and Batta (2019) divide their mission terrain into equally-sized target regions, the centers of which are used as the waypoints that the UAVs are allowed to move between. On the other hand, some studies search for the optimal locations of waypoints. The segments connecting

waypoints are also an issue that need to be addressed. As there are infinitely many locations in continuous terrain to locate a waypoint, the studies often focus on routing only between two points and consider heuristic approaches to determine the locations of the waypoints. For example, Nikolos et al. (2003) use Evolutionary Algorithms (EA) to search waypoints and line segments for connection, Zheng et al. (2005) use EA too but consider B-spline curves for connections, and Foo et al. (2009) use Particle Swarm Optimization and B-spline curves to determine the sequence of waypoint and their connections. The common approach in the studies using waypoints is to consider a single objective, or to transform multiple objectives into a single objective using linear aggregation. There are a few studies that consider multiple objectives simultaneously. For example, Mittal and Deb (2007) consider minimizing the route length and crash risk while searching waypoints and develop a MOEA.

All problems we study in this thesis consider routing with multiple targets and multiple objectives. The UAV routing problems that we study in Chapters 3 and 4, model the movement terrain of the UAV as continuous terrain and allow infinitely many trajectory alternatives between targets. In the problem presented in Chapter 5, an approximation of the continuous terrain is used. Although we do not use a grid structure or waypoints explicitly, we use a finite set of trajectories that represent the infinitely many trajectory alternatives. The UAV can use only one of the trajectories in this set when moving from one target to another.

Second classification of the literature can be made according to the mission objectives considered. The UAVs are often used in military reconnaissance and civilian surveillance missions to collect information at battlefields. Hence, maximizing the collected information is an important objective of actual missions. Prize collection as applied to UAV routing has been considered in the literature. In general, the UAV routing literature maximizes the collected information under a flight duration/distance restriction, and fixed prizes and non-monitored terrains are assumed (see, for example, Ergezer and Leblebicioglu, 2013; Ergezer and Leblebicioglu, 2014; Moskal and Batta, 2017, 2019). However, mission safety and

duration are also important considerations in actual missions. There are studies in the literature that consider flight duration and mission safety as routing objectives (see, for example, Pfeiffer et al., 2009; Tezcaner and Köksalan, 2011; Tezcaner Öztürk and Köksalan, 2018). Route planners in practice often try to explore the trade-offs between these three considerations to decide on the best route to take. Hence, we consider these three routing objectives simultaneously throughout the problems we study in this thesis.

Another important issue in the UAV routing problem is the stochastic characteristics of actual mission terrains. The studies in the literature address different aspects of the uncertainty. For example, Mufalli et al. (2012) consider sensor selection, Evers, Barros, et al. (2014) consider stochastic travel times and pop-up targets, Evers, Dollevoet, et al. (2014) consider uncertainties in fuel usage, and Cho and Batta (2020) consider emerging targets in space and time during a mission. It is generally assumed that prizes at targets are fixed, there is no threat at mission terrains, and there is a single trajectory that the UAVs can follow between a target pair. In practice, the available information at target regions varies with time. For example, there may be prior intelligence about an important activity at a target during a certain time period. Hence, visiting the target during that specific time period is more valuable. Xia et al. (2017) consider routing with a set of target regions with uncertain information. Availability of information at target regions are based on predetermined probabilities. Guerriero et al. (2014) study the problem of filming a live sport event by using multiple UAVs equipped with cameras. An event that cannot be recorded on time is penalized with a lower satisfaction of the audience. They consider three objectives simultaneously and develop exact and heuristic approaches for the solution. In this thesis, we address the uncertainty in information collection by considering time-dependent information at targets.

To sum up, the majority of earlier work on UAV routing consider the route planning problem between an origin and a destination, and the UAV routing with multiple targets has been addressed recently. In general, the movement terrain of UAVs is modeled as a discretized terrain, either allowing movement on the edges of a grid

structure or using waypoints that UAVs are allowed to move between. The literature either consider maximizing the collected information as a single objective function or aggregate multiple objectives in a single objective function and employ heuristic approaches for the solution. Majority of the studies ignore enemy threat at mission terrains and the risk of being detected. Many practical issues such as time-dependent information at target regions are ignored. We address these gaps of the literature in this thesis considering routing with multiple targets, movement in the continuous terrain, multiple objectives simultaneously, radar-monitored mission terrains, multiple connection option between targets, and time-dependent prizes.

CHAPTER 3

A PREFERENCE-BASED MOEA AND ITS IMPLEMENTATION TO BI-OBJECTIVE UAV ROUTE PLANNING IN CONTINUOUS TERRAIN

In this chapter, we study a bi-objective UAV route planning problem considering the terrain the UAV moves as continuous terrain. The UAV takes off from a base, visits all targets in an enemy terrain and returns to the base at the end. The problem can be classified as a TSP in the broader sense. Both the order of targets to be visited and the trajectories to follow between consecutively-visited target pairs need to be determined. The mission terrain is protected by radars. In a terrain that is monitored by radars, minimizing distance traveled and minimizing radar detection threat are two commonly used criteria in many applications (Tezcaner and Köksalan, 2011). In this problem, we consider minimization of total distance traveled and radar detection threat as the routing objectives. The length of the route of the UAV is minimized since fuel or battery consumption is an important consideration. Avoiding detection is another important objective as being detected may result in being shot down and not being able to finalize the mission. When the routes are constructed considering the distance objective only, the UAV is likely to pass through the threat regions more and become more vulnerable. Avoiding the threat regions, on the other hand, is likely to increase the total travel distance. Few studies in the literature consider the trade-offs between these two objectives (see, for example, Pfeiffer et al., 2009; Tezcaner and Köksalan, 2011).

The movement terrain of UAVs in practice is a continuous terrain as they move in the air. As we discussed in Chapter 2, the literature often assumes discretized terrains and limits trajectory options between target pairs due to the difficulties of modeling the movement in continuous terrain. However, continuous terrain is a better representation of reality and has the potential to produce solutions that perform better in both objectives. Tezcaner Öztürk and Köksalan (2018) demonstrate both terrain

structures and discuss their relative characteristics in detail. In this study, we consider continuous terrain allowing for infinitely many numbers of trajectories between target pairs.

The route planning problem in continuous terrain with two objectives and multiple targets is a complex problem with infinitely many efficient solutions. Generating the entire Pareto-optimal frontier is neither meaningful (as we will still need to reduce the solutions to preferred regions) nor computationally practical for this problem. Moreover, a decision maker (DM) is often not interested in the whole solution space. We address this problem by developing a preference-based multi-objective evolutionary algorithm (MOEA) that utilizes reference points. The algorithm searches for the efficient solutions that are close to the reference points of the DM. The developed algorithm is a generic MOEA that can be used for other optimization problems as well. We also make several modifications to the developed algorithm for adapting it to the UAV routing problem. Our main contributions are three-fold:

1. A new generic preference-based MOEA that can be adapted for any multi-objective optimization problem is developed. The algorithm approximates the Pareto-optimal solutions that are close to the reference points defined by DMs. The algorithm provides flexibility to DMs for defining initial reference points and making changes in their preferences during the search process.

2. UAV route planning problem is studied considering multiple targets, two conflicting objective functions, and continuous terrain with infinitely many trajectory options between targets.

3. Problem-specific mechanisms are established to adapt the developed MOEA for the UAV route planning problem. To the best of our knowledge, this is the first study that adapts a preference-based MOEA to the UAV route planning problem.

This chapter is organized as follows: In Section 3.1, the properties of the UAV route planning problem in continuous terrain is explained. In Section 3.2, the developed

preference-based MOEA is presented. In Section 3.3, the implementation of the developed algorithm on the bi-objective UAV route planning problem is explained. In Section 3.4, we explain how we determine the true trajectories used in the final solution of the developed MOEA. The computational demonstrations and their results are reported in Section 3.5. Finally, this chapter is concluded with a discussion in Section 3.6.

3.1 Problem Structure of the Bi-Objective UAV Route Planning Problem

In this section, we explain the problem structure that is adapted from Tezcaner Öztürk and Köksalan (2018).

3.1.1 Terrain Structure of the Bi-Objective UAV Route Planning Problem

We present an example terrain structure with five targets (triangles) and four radar regions (circles) in Figure 3.1. The UAV visits all targets while trying to minimize both the distance traveled and the radar detection threat at radar-covered territories. Radars are located at the centers of the circular regions and they are ineffective in the white regions. Inside the radar regions, there are two parts: the core region (darker) where the detection probability is 1.0 and the outer region (lighter) where the detection probability decreases as we get closer to the outer boundary of the light region. We assume that UAV is within the range of at most one radar region when travelling between any two targets. Hence, if the trajectory with shortest distance between two targets is subject to the threats from more than one radar region, the radar region that poses the most threat is considered as the effective radar region.

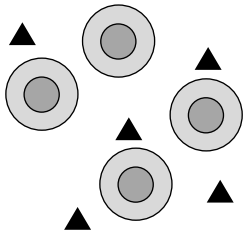


Figure 3.1: An example terrain structure

3.1.2 Objectives of the Bi-Objective UAV Route Planning Problem

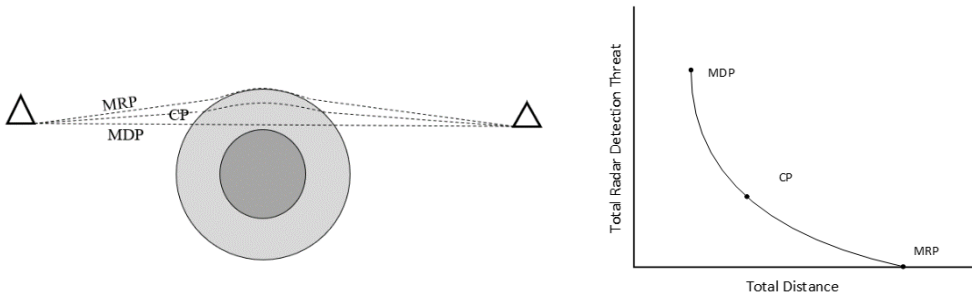
We consider two objectives; minimizing distance traveled and minimizing radar detection threat, which are the main considerations in many applications. Distance traveled (*D*) corresponds to the length of the tour the UAV follows. Radar detection threat (*RDT*) is the total radar threat the UAV is exposed to during its tour. We calculate the radar detection threat inside a radar region using the measure developed by Gudaitis (1994). This measure was used by Tezcaner and Köksalan (2011) and Tezcaner Öztürk and Köksalan (2018) as well. It sums up all radar detection probabilities the UAV is exposed to during the route. It is equivalent to approximating the duration the UAV is exposed to each detection probability if we assume that the UAV moves at a constant speed. Details of this measure is reported in Appendix A.

3.1.3 Movement Between Targets

Tezcaner Öztürk and Köksalan (2018) develop approaches that find all efficient trajectories between target pairs. They enumerate many possible trajectories that pass through a radar region between target pairs. Out of those trajectories, they eliminate the dominated ones and find the structure of the Pareto-optimal frontier. Depending on whether or not the shortest distance between two targets passes through a radar region, they define three types of movements: (1) The shortest distance between the target pair does not pass through any radar region, (2) The shortest distance between

the target pair passes through only the outer radar region, (3) The shortest distance between the target pair passes through both the outer and the inner radar regions.

Type 1 movement has a single trajectory option that is the trajectory with the shortest travel duration and no radar detection threat. Movement type 2 has a curved Pareto-optimal frontier (Figure 3.2). The minimum distance point (MDP), and the minimum radar point (MRP) are the extreme efficient trajectories with minimum distance and minimum radar detection threat, respectively. Central point (CP) is an approximate Pareto-optimal point that has intermediate values in both objectives. Examples of three efficient trajectories (MDP, MRP, and CP) and their images in the objective function space corresponding to movement type 2 are demonstrated in Figure 3.2(a) and (b), respectively.

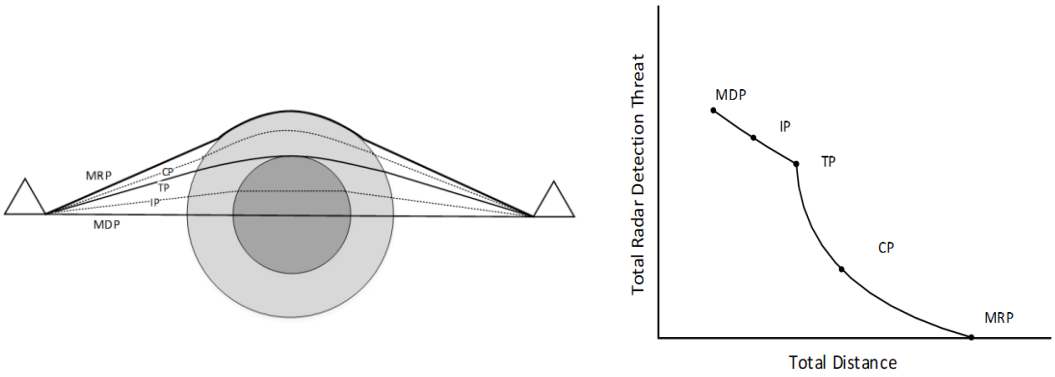


(a) Decision Space (b) Corresponding Pareto-optimal Frontier

Figure 3.2: Movement Type 2

Movement type 3 has a two-piece (a linear and a curved) Pareto-optimal frontier (Figure 3.3). The linear piece corresponds to the efficient trajectories that pass through both the inner radar region (where the detection probability is 1.0) and the outer radar region. The curved piece, similar to movement type 2, corresponds to the efficient trajectories that pass through only the outer radar region. For this movement type, we additionally define the tangent point (TP), that is the efficient trajectory with minimum distance among the trajectories passing through only the outer radar region, and an inner point (IP), that is, a point that corresponds to a trajectory passing through both the outer and the inner radar regions. Examples of five efficient

trajectories (MDP, IP, TP, CP, and MRP) and their images in the objective function space are demonstrated in Figure 3.3(a) and (b), respectively. Since the probability of detection is 1.0 in the inner region, the UAV follows a path that minimizes its distance (a straight-line path) passing through the inner region for any efficient trajectory, as is the case for IP. We allow movement in the inner radar region where the probability of detection is 1.0 as we assume that there is no one-to-one correspondence between probability of being detected and being shutdown. This shortcut structure of the trajectories passing through the inner region causes the approximate straight-line portion of the Pareto-optimal frontier of movement type 3 in Figure 3.3(b).



(a) Decision Space (b) Corresponding Pareto-optimal Frontier

Figure 3.3: Movement Type 3

The trajectories connecting each target pair with the type 2 and 3 movements have infinitely many Pareto-optimal points. Generating actual Pareto-optimal frontiers of these movements is computationally expensive. Tezcaner Öztürk and Köksalan (2018) observe that their enumerated Pareto-optimal frontier is similar in structure to the curves that result from L_q functions that produce a variety of curvatures based on their q values. Hence, instead of generating all efficient solutions for movement types 2 and 3, they approximate their Pareto-optimal frontiers using L_q functions. This function was first developed by Köksalan (1999) in a scheduling context. Köksalan and Lokman (2009) generalized the L_q function for any number of objectives and showed that it is capable of approximating the Pareto-optimal

frontiers of combinatorial problems well. Köksalan and Soylu (2010) used these functions in an evolutionary algorithm context for a bi-criteria hub location problem, and recently Köksalan and Tezcaner Öztürk (2017) utilized them to approximate the Pareto-optimal frontier of a tour for the generalized MOTSP.

Similar to Tezcaner Öztürk and Köksalan (2018), we approximate the Pareto-optimal frontiers of the movement types 2 and 3 by L_q function. Tezcaner Öztürk and Köksalan (2018) fit an L_q function using three points: two extreme Pareto-optimal points (MDP and MRP) and one point approximating the central part of the Pareto-optimal frontier (CP). The CP can be found by solving a nonlinear mathematical model that minimizes the radar detection threat for a given distance value (for example the mid-point of the two extreme distance values MDP and MRP). The corresponding CP value characterizes the efficient solution corresponding to the central distance value among the two extreme efficient solutions. In our approach, rather than solving the mathematical model, we use the approximate procedure developed by Tezcaner Öztürk and Köksalan (2018) to find a central point that has similar properties. In order to see the sensitivity of the q value to different central points, we experimented with different distance values within the range of efficient solutions. We observed that the resulting q values are robust to the selected central point. Even when the distance value of the central point changes by 30% within the range of distance values of efficient solutions, the q value changes less than 3.0%. This is expected since the Pareto-optimal frontier is quite smooth when it is densely populated as in many multi-objective combinatorial optimization problems (see, Köksalan and Lokman, 2009).

Once MDP, MRP, and CP between targets i and j are found, the L_q function can be characterized. Let the first (distance) and second (radar detection threat) objective values of MDP, MRP, and CP of target pair (i, j) be, $\left(z^1_{ijMDP}, z^2_{ijMDP}\right)$, $\left(z^1_{ijMRP}, z^2_{ijMRP}\right)$, and $\left(z^1_{ijCP}, z^2_{ijCP}\right)$, respectively. Then, we fit the L_q function using Equation (3.1), where $\left(d_{ijCP}, r_{ijCP}\right)$ are the normalized first and second objective values of CP of target pair (i, j), respectively.

$$\left(1 - d_{ijCP}\right)^q + \left(1 - r_{ijCP}\right)^q = 1$$
$$\text{where, } \left(d_{ijCP}, r_{ijCP}\right) = \left(\frac{z^1_{ijCP} - z^1_{ijMDP}}{z^1_{ijMRP} - z^1_{ijMDP}}, \frac{z^2_{ijCP} - z^2_{ijMRP}}{z^2_{ijMDP} - z^2_{ijMRP}}\right) \quad (3.1)$$

The only unknown in Equation (3.1) is the q value, which can be found solving a nonlinear programming problem (with any objective function) satisfying Equation (3.1) as the only constraint. Once the q value is found, for any trajectory XP between MDP and MRP of target pair (i,j), r_{ijXP} corresponding to d_{ijXP} can be calculated by replacing d_{ijCP} and r_{ijCP} with d_{ijXP} and r_{ijXP} in Equation (3.1).

3.2 Flexible Reference Point-Based Evolutionary Algorithm (FREA)

The trajectories connecting each target pair with the second and third movement types have infinitely many Pareto-optimal points. This leads to infinitely many Pareto-optimal points for the tours. Generating all of them is neither practical nor computationally possible. As a solution, we develop a general preference-based MOEA, FREA, together with its specific mechanisms for UAV route planning. DM's preferences are captured with reference points and Pareto-optimal solutions that are close to the reference points are approximated. The developed algorithm is a generic MOEA that can be used for other multi-objective optimization problems as well. In the following, first the relevant literature of MOEA is reviewed and then the details of FREA are presented.

3.2.1 Preference-Based Multi-objective Evolutionary Algorithms

MOEA methods are population-based approaches mimicking the natural evolution process. An initial population of solutions are exposed to several genetic operations, e.g. selection, crossover, mutation, and it evolves through the generations. The population members with better objective values survive at the end. We refer Fonseca and Fleming (1995) and Zhou et al. (2011) for more details on MOEAs.

There has been a realization in evolutionary multi-objective optimization that generating all efficient solutions is not very useful in many situations and trying to converge to preferred solutions by incorporating DM's preference information is more practical. Deb and Koksalan (2010) argue the importance of incorporating the DM's preferences from three perspectives. Firstly, it is not practical to generate all Pareto-optimal solutions in complex problems. Secondly, DMs are often not interested in the whole Pareto-optimal set. Finally, concentrating on preferred regions could help obtaining better approximations of the Pareto-optimal solutions in those regions.

There are several ways of incorporating preferences into the solutions process in the literature. One popular approach is employing interactive methods that progressively elicit preference information from DMs through solution or objective comparisons, and direct the search to the preferred regions of the Pareto-optimal frontier (see, Phelps and Köksalan, 2003; Köksalan and Karahan, 2010, for some examples). Another popular approach is to employ reference point-based approaches, where the search process is guided by the reference points or reference directions towards preferred regions of the Pareto-optimal frontier. Such approaches are widely used in the multi-criteria decision making (MCDM) literature and are suitable if the DM is comfortable with reflecting his/her preferences through reference points. An early example of reference point-based approaches in MCDM is the study of Wierzbicki (1980) that projects reference points to the Pareto-optimal frontier using achievement scalarizing functions. Following this study, many reference point-based algorithms, including Jaszkiewicz and Słowiński (1999)'s Light Beam Search, and Korhonen and Wallenius (1988)'s Pareto Race are developed. Since reference point-based approaches are easy to understand and implement, they became popular in the field of MOEA as well. Deb and Kumar (2007a) adapt reference direction approach to converge to preferred solutions using the DM's reference directions. Deb and Kumar (2007b) combined light beam search idea with evolutionary algorithms to find a preferred set of solutions. Thiele et al. (2009) directly used reference point information in fitness evaluation of an evolutionary algorithm. Our preference-based

MOEA utilizes reference points by searching for efficient solutions that are close to the reference points of the DM. The idea behind our algorithm is similar to that of R-NSGA-II developed by Deb and Sundar (2006) to converge to nondominated solutions close to the DM's reference points. However, our algorithm has important distinctive features. It overcomes the deficiencies of their algorithm, provides more flexibility to users and can present a set of solutions representing the Pareto-optimal frontier at the end in addition to the preferred solutions close to DM's reference points.

3.2.2 Details and Steps of FREA

The idea behind our algorithm resembles to that of another algorithm called R-NSGA-II. It is a reference point-based MOEA developed by Deb and Sundar (2006). The DM specifies his/her reference points a-priori, and the algorithm searches for a set of Pareto-optimal solutions close to those points. The algorithm replaces NSGA-II's (Deb et al., 2002) crowding distance operator with a preference distance operator. The preference distance operator uses normalized Euclidean distance to measure the distance between a solution and a reference point. The solutions closer to the reference points are assigned smaller ranks and are favored in both tournament selection and in reducing the population size for the next generation. In order to maintain some diversity, preference ranks are also manipulated with an ε-niching procedure. The idea is to select a representative among the solutions that are very close to each other (within ε distance) and favor only this solution within its neighborhood in the genetic operators.

An important shortcoming of R-NSGA-II is that some of its final points at termination may be dominated when the Pareto-optimal frontier is discontinuous. When presented, the DM may choose one of the dominated solutions, in the absence of information that there are Pareto-optimal solutions that dominate these solutions. Had the DM known the Pareto-optimal solutions dominating these solutions, he/she

would have never chosen the dominated solutions. Therefore, it is important to avoid ending up with dominated solutions in the final population.

The 30-variable, 2-objective ZDT3 test problem has a discontinuous Pareto-optimal frontier. Deb and Sundar (2006)'s method ends up with dominated solutions when it starts with a dominated reference point.

We develop an algorithm that implements new mechanisms that aim to avoid dominated solutions. Specifically, we keep a fixed-capacity archive population throughout the algorithm. We update this archive population in each generation with the nondominated solutions of the most recent generation replacing the solutions that become dominated with those that dominate them. We use the ε-niching procedure by selecting only one solution within an ε-neighborhood, to maintain the capacity of the archive population fixed. Keeping an archive population is not unusual for MOEAs. Various studies resort to archive populations for various purposes (Köksalan and Karahan, 2010; Sindhya et al., 2011; Chen et al., 2015).

We first present the steps of FREA below and later discuss the specifics of some of the steps. Let G denote the number of generations obtained before a set of solutions are presented to the DM. Let P_g, Q_g, and R_g denote the parent, offspring, and global populations at generation g, respectively. Let *Archive* denote the archive population.

Step 0. Initialization: Generate an initial population (P_0) of size *N*.

Step 1. Evaluation: Apply nondominated sorting, preference ranking, and ε-niching to P_0.

Step 2. Initialize archive population: Fill the *Archive.*

Step 3. Set generation counter $(g = 0)$.

Step 4. Generating offspring population: Generate offspring population (Q_g) of size *N* using genetic operations (selection, crossover, and mutation).

Step 5. Filling global population (R_g): Combine parent and offspring populations, $R_g = P_g \cup Q_g$.

Step 6. Filling new population (P_{g+1})**:** Apply nondominated sorting to $R_g \cup Archive$. Apply preference ranking and ε-niching to R_g. Form new population P_{g+1} from R_g. Let $g \leftarrow g + 1$.

Step 7. **Updating archive population:** Update *Archive*.

Step 8. Check termination: If $g = G$, go to Step 9. Otherwise, go to Step 4.

Step 9. Updating reference points: Present the final population and *Archive* to the DM. If the DM updates his/her reference point(s), add *Archive* to the final population, set the value of G, and go to Step 3. Otherwise, terminate the algorithm.

3.2.2.1 Nondominated Sorting, Preference Ranking, and *ε-niching*

To evaluate the original population and the global population, we apply nondominated sorting (Deb et al., 2002) and preference ranking (Deb and Sundar, 2006). Let R be the set of reference points. In the preference ranking measure, we first calculate the normalized Euclidean distance (Equation (3.2)) that measures the distance (e_j^i) between solution $\boldsymbol{x^i}$ and reference point $\boldsymbol{r_j}$. Here, $z_k(\boldsymbol{x^i})$ and r_{kj} are the kth objective value of solution $\boldsymbol{x^i}$ and reference point $\boldsymbol{r_j}$, respectively, and z_k^{max} and z_k^{min} are the maximum and minimum values of objective k among all relatively nondominated solutions found until the current generation, respectively.

$$e_j^i = \sqrt{\sum_{k=1}^{p} \left(\frac{z_k(\boldsymbol{x^i}) - r_{kj}}{z_k^{max} - z_k^{min}} \right)^2}, \quad i = 1, \dots, n, \qquad j \in R \tag{3.2}$$

Solutions are then sorted based on their e_j^i values for each $j \in R$. Let pr_j^i be the preference rank of solution $\boldsymbol{x^i}$ for reference point $\boldsymbol{r_j}$. The solutions that are closer to the reference points, i.e., solutions with smaller e_j^i values, are assigned smaller ranks. We then assign an overall preference rank, pr^i, to each solution $\boldsymbol{x^i}$ such that $pr^i = \min_{j \in R} \{pr_j^i\}$. Employing the ε-niching procedure, we update these preference ranks (see, Deb and Sundar, 2006). In this procedure, we first sort the solutions in

descending order by their preference ranks. We start with the first solution in the sorted list, say $\boldsymbol{x^a}$, whose preference rank is 1. We then find the normalized Euclidean distance between $\boldsymbol{x^a}$ and each solution that has a preference rank worse than that of $\boldsymbol{x^a}$. For all solutions that have a normalized Euclidean distance smaller than ε to $\boldsymbol{x^a}$, we assign an arbitrarily large preference rank. We then select the next solution from the top of the list whose preference rank has not been updated yet and repeat the same procedure with the solutions that are within its ε normalized Euclidean distance. We continue updating the preference ranks until we evaluate all solutions in the list. In the rest of the algorithm, the solutions with smaller preference ranks are favored in both tournament selection and population size reduction operations.

When we need to reduce the population size in Step 6, we use the latest-assigned nondominated ranks of solutions, select solutions beginning from the nondominated front, and continue with dominated fronts sequentially. If the size of a front is larger than the remaining slots, we select the solutions in that front in the order of their preference ranks to fill the remaining slots.

3.2.2.2 Selection

In Step 4, we use the tournament selection operator (Deb et al., 2002) to select the parents that undergo the crossover operation. This operator matches and compares randomly selected pairs of solutions. The solution having a better nondominated rank wins and ties are broken based on their preference rank values. Winners are placed in a mating pool. Each solution participates in the tournament twice. Based on whether it loses both, wins once, or wins twice, 0, 1, or 2 copies of it are placed in the mating pool, respectively. Each parent in the mating pool is randomly matched with another parent and the crossover operator is applied to all parent pairs to obtain offspring. Then the mutation operator is applied to offspring to obtain the final offspring population.

3.2.2.3 Archive Mechanism

In Steps 2 and 7, we form and update the archive population, respectively. We next give the details of updating the archive population. Let NDF_g be the set of nondominated points of the population at generation g, *Capacity* be the capacity of *Archive*, and *TempCapacity* be the capacity of the set selected from the relatively nondominated solutions of the population at generation g, $NDF\text{-}Temp_g$.

Updating the Archive

a. Initialize an empty temporary set for generation g ($NDF\text{-}Temp_g$). Select at most *TempCapacity* solutions from NDF_g (using the representative selection algorithm described below) and place them in set $NDF\text{-}Temp_g$.

b. Combine *Archive* and $NDF\text{-}Temp_g$. Select at most *Capacity* solutions from the combined population (using nondominated sorting and ε-niching), and place them in *Archive*.

We use two approaches in filling the temporary and archive sets. We use the representative selection algorithm to fill the temporary set, where we represent a set of nondominated solutions with a diverse subset having fewer elements. After filling the archive in the first generation, we update it in subsequent generations. In updating, we employ nondominated sorting to find the relatively nondominated solutions of the combined members of the current temporary and archive sets. If the number of the relatively nondominated solutions is larger than *Capacity*, we implement ε-niching among them to reduce the set size below *Capacity*. Implementation of the ε-niching procedure to the archive is slightly different from that of the regular population. Rather than assigning large preference ranks to those that are within ε distance of the considered solution, we keep them out of the archive set. Smaller ε values place more solutions in the *Archive*. We first start with a small ε value and eliminate solutions. If more solutions than the *Capacity* are retained, we double the ε value and keep repeating until the *Archive's* size is within *Capacity*.

Representative Selection Algorithm

While updating the archive, we select representatives from NDF_g and place them in set *NDF-Temp*$_g$. We use the procedure of Sylva and Crema (2007) where they generate a representative subset of nondominated points for multi-objective mixed integer linear programs. The idea is to sequentially select representative points, each time adding the point that is at maximum (Tchebycheff) distance (the worst-represented point) to current representative points. We select the first representative point as the solution that is farthest to its farthest solution in Tchebycheff distance.

We next give the details of the selection procedure. Let $\boldsymbol{RP_j}$ be the *j*th representative point, $j = 1, \dots, CurrentCapacity$, SRP_j be the set of first *j*-1 representative points found, $SRP_j = \{\boldsymbol{RP_1}, \boldsymbol{RP_2}, \dots, \boldsymbol{RP_{j-1}}\}$, and $z_k(\boldsymbol{x})$ be the *k*th objective value of solution $\boldsymbol{x}$.

Selecting $\boldsymbol{RP_1}$

For each solution $\boldsymbol{x^i} \in NDF_g$,

a. Calculate the Tchebycheff distance, $d(\boldsymbol{x^i}, \boldsymbol{x^j})$ between the objective values of solution $\boldsymbol{x^i}$ and each solution $\boldsymbol{x^j} \in NDF_g/\{\boldsymbol{x^i}\}$, $d(\boldsymbol{x^i}, \boldsymbol{x^j}) = \max_k\{|z_k(\boldsymbol{x^i}) - z_k(\boldsymbol{x^j})|\}$.

b. Assign a worst-representation value $wRP(\boldsymbol{x^i}) = \max_{\boldsymbol{x^j} \in NDF_g \setminus \boldsymbol{x^i}} d(\boldsymbol{x^i}, \boldsymbol{x^j})$ to each solution $\boldsymbol{x^i}$.

Set $\boldsymbol{RP_1} = \underset{\boldsymbol{x^i} \in NDF_g}{argmax}\ wRP(\boldsymbol{x^i})$.

Selecting $\boldsymbol{RP_j}$, $j = 2, \dots, CurrentCapacity$

For each $\boldsymbol{x^i} \in NDF_g \setminus SRP_j$,

a. Calculate $d(\boldsymbol{x^i}, \boldsymbol{RP_t}) = \max_k\{|z_k(\boldsymbol{x^i}) - z_k(\boldsymbol{RP_t})|\}$, for $t = 1, \dots, j-1$.

b. Let $d(\boldsymbol{x^i}) = \min_{t=1,\dots,j-1} d(\boldsymbol{x^i}, \boldsymbol{RP_t})$.

Set $\boldsymbol{RP_j} = \underset{\boldsymbol{x^i} \in NDF_g \setminus SRP_j}{argmax}\ d(\boldsymbol{x^i})$.

This archive mechanism has three important roles in the algorithm. First, it helps prevent ending up with dominated solutions by storing a representative set of nondominated solutions found throughout the algorithm. Second, the archive population gives the DM a general idea for the position and spread of the Pareto-optimal frontier in addition to the solutions found close to reference points. Third, it helps in converging to a different region quickly if the DM decides to change the reference point during the algorithm.

3.2.2.4 Flexible Reference Points

The DM is allowed to define multiple reference points at the same time. The algorithm results in approximately nondominated solutions regardless of whether the DM provides nondominated or dominated reference points. The DMs are also allowed to change their reference point(s) during the algorithm. Allowing the DM to change his/her reference points is another aspect in which our algorithm is different from R-NSGA-II. We allow for changes in the reference points in Step 9, where the final set of solutions are presented to the DM. This reduces the burden on the DM in setting the reference point(s), since with more information on the spread and location of the Pareto-optimal frontier, the DM has the flexibility to modify his/her reference point(s). This acknowledges that the solution process may serve as a learning process for the DM. With more information on the available solutions, the DM may make more informed decisions regarding his/her preferred regions.

3.2.3 Comparison of FREA with R-NSGA-II

We compare our algorithm with R-NSGA-II on the same test problems solved in Deb and Sundar (2006). The test problems ZDT1, ZDT2, ZDT3 are developed by Zitzler et al. (2000). ZDT1 has two minimization-type objectives, 30 variables, and a convex Pareto-optimal frontier. ZDT2 is similar, with the difference of a non-convex Pareto-optimal frontier. ZDT3 also has the same settings in the objective

functions and variables but has a disconnected Pareto-optimal frontier. DTLZ2 test problem is developed by (Deb et al., 2005). As in Deb and Sundar (2006), we solve 3-objective and 11-variable, 5-objective and 14-variable, and 10-objective and 19-variable versions of these problems. DTLZ2 problems have the property that the sum of squares of the objective function values of a Pareto-optimal solution is equal to 1. Finally, the two engineering design problems we solved have two objectives to be minimized. One of them has four variables and four constraints and the other has three variables and eight constraints. Both problems have convex Pareto-optimal frontiers.

We solve each instance using both FREA and R-NSGA-II, starting both algorithms with the same seed. We use the parameter settings given in Deb and Sundar (2006). We perform our tests using R (R Core Team, 2021) on a PC with 6-core Intel Core i7-4770S CPU, 3.10Ghz, 16 GB RAM. We set *Capacity* to 50 (50% of the population) and *TempCapacity* to 10.

For ZDT1, ZDT2, 3, 5, and 10-objective DTLZ2, and the two engineering design problems, the performances of FREA and R-NSGA-II are similar and very good. Both algorithms are able to converge to the Pareto-optimal solution sets around the given reference points by the route planners. The closest solutions in the final solution sets of both algorithms for the same reference points and their Euclidean distances to the reference points can be seen in Table 3.1. The points they converge to at the end of the algorithms for the same reference points are almost identical and these points are either on or arbitrarily close to their respective Pareto-optimal frontiers. FREA also finds additional solutions representing the Pareto-optimal front.

The solution times of FREA are slightly higher than those of R-NSGA-II as expected since FREA maintains and updates an archive throughout the algorithm. FREA takes roughly 50% longer (around 100 vs 70 CPU seconds) for the 2-objective problems (ZDT1, ZDT2, ZDT3, and the two engineering design problems). The difference is less for problems with more objectives (e.g., the CPU times of R-NSGA-II and

FREA are about 135 and 162 seconds, respectively, for the 10-objective DTLZ2 problem).

Although the two algorithms perform similarly on these problems, R-NSGA-II only finds solutions that are close to the reference points. FREA, on the other hand, provides information on the spread of the Pareto-optimal frontier as well. This gives the DM an insight about the spectrum of solutions and guides him/her well if he/she wishes to explore different parts of the solution space. Exploring different parts of the solution space is facilitated by the additional capability of FREA that allows for defining new reference points progressively.

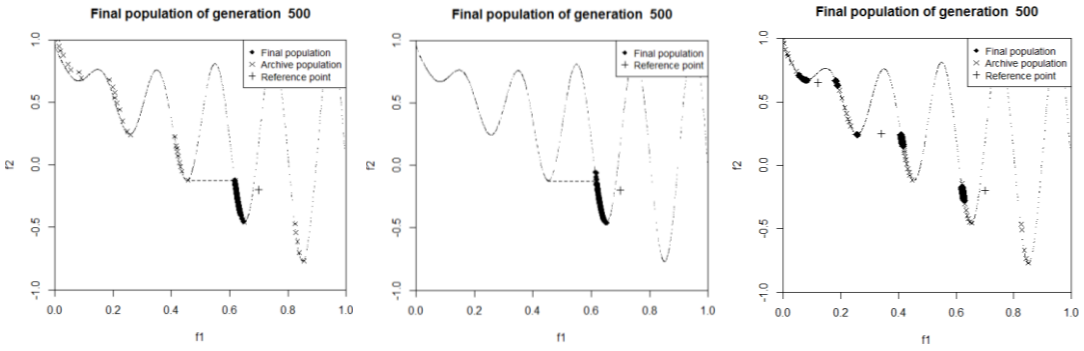
In our experiments, FREA is superior to R-NSGA-II in cases where the Pareto-optimal frontiers are discontinuous. ZDT3 is such a problem and we demonstrate next how FREA outperforms R-NSGA-II on this problem. We will also compare the two algorithms later on another problem that may have a discontinuous Pareto-optimal front, the UAV routing problem.

Table 3.1: Comparison of FREA and R-NSGA-II on test problems

Problem	**Reference Point**	**FREA**		**R-NSGA-II**	
		Final Solution	**Distance to Reference Point**	**Final Solution**	**Distance to Reference Point**
ZDT1	(0.100, 0.600)	(0.134, 0.634)	0.034	(0.143, 0.621)	0.034
	(0.700, 0.200)	(0.687, 0.171)	0.022	(0.678, 0.177)	0.023
ZDT2	(0.100,0.800)	(0.154, 0.976)	0.130	(0.152, 0.977)	0.131
	(0.600, 1.000)	(0.446, 0.801)	0.169	(0.435, 0.811)	0.178
DTLZ2 – 3 obj	(0.200, 0.200, 0.600)	(0.302, 0.302, 0.904)	0.056	(0.301, 0.301, 0.905)	0.056
	(0.600, 0.800, 0.800)	(0.551, 0.581, 0.649)	0.045	(0.446, 0.630, 0.666)	0.044
DTLZ2 – 5 obj	(0.500, 0.500, 0.500, 0.500, 0.500)	(0.502, 0.450, 0.480, 0.449, 0.459)	0.011	(0.510, 0.445, 0.454, 0.459, 0.497)	0.011
	(0.200, 0.200, 0.200, 0.200, 0.800)	(0.223, 0.222, 0.224, 0.224, 0.895)	0.013	(0.223, 0.224, 0.223, 0.224, 0.895)	0.013
DTLZ2 – 10 obj	0.250 for all objectives	(0.315, 0.316, 0.317, 0.316, 0.318, 0.316, 0.315, 0.316, 0.317, 0.316)	0.019	(0.315, 0.316, 0.317, 0.316, 0.318, 0.316, 0.315, 0.316, 0.317, 0.316)	0.019
Welded Beam Design Problem	(4.000, 0.003)	(5.999, 0.003)	0.037	(5.999, 0.003)	0.037
	(20.000, 0.002)	(17.364, 0.001)	0.130	(16.941, 0.001)	0.131
Spring Design Problem	(4.000, 180000.000)	(3.984, 155356.100)	7.935×10^{-6}	(3.952, 152374.100)	2.380×10^{-5}
	(25.000, 20000.000)	(24.996, 59055.461)	1.980×10^{-6}	(25.126, 590069.000)	6.240×10^{-5}

3.2.3.1 Comparison on ZDT3 Problem

30-variable ZDT3 problem has a discontinuous Pareto-optimal frontier. The results given in Deb and Sundar (2006) show that some of the solutions found by R-NSGA-II are dominated for this problem when a dominated reference point is used. To investigate this issue further, we replicate both algorithms 200 times, providing the same dominated reference point, each time changing the random generation seed. In all replications, FREA ended up with Pareto-optimal solutions. We present the result of an arbitrary replication of FREA in Figure 3.4(a). R-NSGA-II, on the other hand, found some dominated solutions in addition to some Pareto-optimal solutions in 120 of the 200 instances (60%). The result of an arbitrarily-selected one of these 120 cases is given in Figure 3.4(b). Terminating with some dominated solutions could be critical. The closest solution to the reference point may correspond to a dominated point. The DM may select a dominated solution as the most preferred solution, unaware of undetected Pareto-optimal solutions that perform better than the selected solution in all objectives. In Figure 3.4(c), we present the results of FREA introducing three dominated reference points. The algorithm again ends up with Pareto-optimal solutions close to each of the reference points.



(a) FREA for a Dominated Reference Point

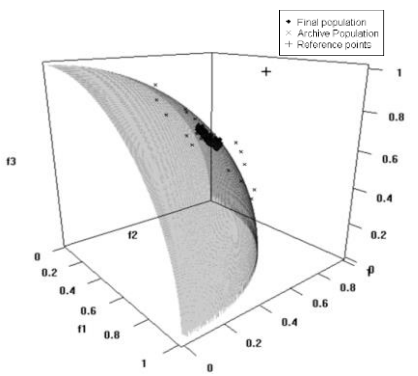
(b) R-NSGA-II for a Dominated Reference Point

(c) FREA for Three Dominated Reference Points

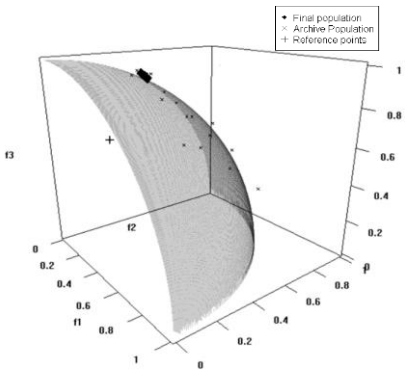
Figure 3.4: Results for ZDT3

3.2.3.2 Changing Reference Points on 3-objective DTLZ2

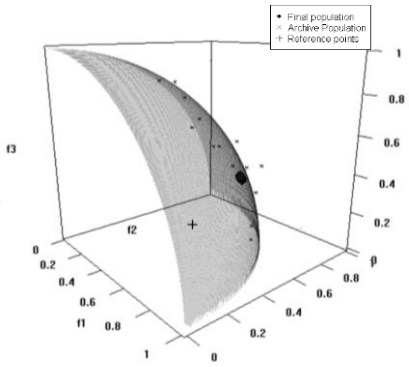
We solve the 3-objective, 11-variable DTLZ2 test problem that has a non-convex Pareto-optimal frontier, assuming that the DM changes reference points. We first run the algorithm for 100 generations using (0.8, 0.6, 0.1). Assume that we show the obtained solutions (close to the reference point) as well as the archive to the DM (Figure 3.5(a)), and the DM modifies his/her reference point to (0.2, 0.2, 0.6). The results after 100 more generations are shown in Figure 3.5(b). Assuming that the DM changes his/her reference point once more to (0.8, 0.2, 0.3), the algorithm again converges to the solutions close to the new reference point. Assuming that the DM does not update the latest reference point, we terminate the algorithm after 100 more generations in Figure 3.5(c). FREA manages to adapt to these changes easily. The solution time was approximately 25 seconds for each reference point.



(a) DM's Initial Reference Point: (0.8, 0.6, 0.1)



(b) DM's Second Reference Point: (0.2, 0.2, 0.6)



(b) DM's Final Reference Point: (0.8, 0.2, 0.3)

Figure 3.5: Changing reference points in FREA for DTLZ2

3.3 Implementing FREA on UAV Route Planning

We develop specific aspects and steps of FREA for the route planning problem. We refer this version of the algorithm as FREA_UAV. We first discuss the representation of solutions and the special fitness assignment procedure we developed. We refer to the DM of this problem as a route planner (RP).

3.3.1 Representation of Solutions

In our algorithm, each individual in the population corresponds to a tour. Different representation schemes are available in the literature to represent a tour (see Larrañaga et al., 1999). We use a scheme called path representation as in Köksalan and Tezcaner Öztürk (2017). If target *i* is the *j*th target to be visited, it is placed in the *j*th position in the chromosome. For example, 1-2-3-4-5-1 is a path representation for a tour with five targets, where the UAV starts its movement from target 1, visits targets 2, 3, 4, and 5 respectively and returns to target 1.

3.3.2 Assigning Fitness Values

There are infinitely many ways to move between two targets that are covered by a radar region. Therefore, each tour having such connections has a continuous Pareto-optimal frontier. We approximate the Pareto-optimal frontiers between each target pair with the L_q function before executing the algorithm, and use these as input to FREA_UAV. During the algorithm, similar to approximating the Pareto-optimal frontier of the movement between target pairs, we approximate the Pareto-optimal frontier of a tour with an L_q function. We then choose several points on the approximated frontier of the tour, each of which is represented with its corresponding total distance traveled and total radar detection threat value. These points become separate members of the population, implicitly representing the same tour. We next

provide the details of fitting an L_q function to a tour, selecting the points on the tour, and assigning fitness vectors to these points.

3.3.2.1 Fitting an L_q Function to the Pareto-optimal Frontier of Tour π

We fit the L_q function to tour π using two extreme and a central Pareto-optimal point of tour π using Equation (3.3).

$$(1 - d_{\pi CP})^q + (1 - r_{\pi CP})^q = 1$$
$$\text{where, } (d_{\pi CP}, r_{\pi CP}) = \left(\frac{Z^1_{\pi CP} - Z^1_{\pi MDP}}{Z^1_{\pi MRP} - Z^1_{\pi MDP}}, \frac{Z^2_{\pi CP} - Z^2_{\pi MRP}}{Z^2_{\pi MDP} - Z^2_{\pi MRP}}\right) \tag{3.3}$$

Let $d_{\pi CP}$ and $r_{\pi CP}$ represent the normalized distance and radar detection threat of the central Pareto-optimal point *CP* of tour π, respectively, $Z^k_{\pi t}$ denote the *k*th objective value of the point t of the Pareto-optimal frontier of tour π, z^k_{ijt} denote the *k*th objective value of the extreme point t of the Pareto-optimal frontier of target pair (i, j). The extreme points MDP and MRP are the extreme Pareto-optimal points with minimum distance and minimum radar detection threat, respectively, and $k = 1, 2$ correspond to (1) distance and (2) radar detection threat objectives. We compute the objective function values of MDP and MRP of tour π in Equation (3.4) using the extreme Pareto-optimal points of the trajectories between the consecutive target pairs in that tour, $(i, j) \in \pi$.

$$Z^k_{\pi t} = \sum_{(i,j)\in\pi} z^k_{ijt} \qquad k = 1, 2, \;\; t = MDP, MRP \tag{3.4}$$

We then find a central point, finding the solution of Equation (3.5) for tour π.

$$\text{Min } U(\boldsymbol{x}) = w.z_1(\boldsymbol{x}) + (1 - w).z_2(\boldsymbol{x}) \tag{3.5}$$

This equation is a composite objective function formed by combining the two objectives for some $0 \leq w \leq 1$. It is possible to use different weights, w. We use

$w = w' = \left(\frac{Z^2_{\pi MDP} - Z^2_{\pi MRP}}{Z^2_{\pi MDP} - Z^2_{\pi MRP} + Z^1_{\pi MRP} - Z^1_{\pi MDP}}\right)$ which creates an objective function contour that passes through the two extreme Pareto-optimal points.

Let z^k_{ijCP} denote the *k*th objective value of the central point of the trajectories connecting targets i and j, for *k=1,2*, optimizing Equation (3.5) for w'. We then find $Z^k_{\pi CP}$ that is the *k*th objective value of the central point of tour π, optimizing Equation (3.5) for w', using Equation (3.4) for $t = CP$. In the next section, we explain the procedure we use to find z^k_{ijCP}.

Once the $Z^k_{\pi CP}$ for *k=1,2* are found, the only unknown in Equation (3.3) is the q value, which can be found solving a nonlinear programming problem (with any objective function) satisfying Equation (3.3) as the only constraint. Once the q value is found, for any point t between the two extreme Pareto-optimal points MDP and MRP of tour π, i.e. the radar detection threat ($r_{\pi t}$) corresponding to distance ($d_{\pi t}$) or vice versa, can be calculated by replacing $d_{\pi CP}$ and $r_{\pi CP}$ with $d_{\pi t}$ and $r_{\pi t}$ in Equation (3.3).

3.3.2.2 Finding z_{ijCP}

For each target pair, we have an approximated Pareto-optimal frontier obtained with L_q function. We use the following procedure developed by Türeci (2017) to find z^k_{ijCP} values.

Türeci (2017) finds the point (d^*, r^*) on the approximated Pareto-optimal frontier between target pairs that optimize Equation (3.5) for any *w*. This point may not correspond to a true trajectory but there exists a trajectory with distance, d^*, and radar detection threat value, possibly, slightly different from r^*. Then, Equation (3.5) reduces to Equation (3.5') that is solved over the corresponding L_q function.

$$\text{Min } U = w.d + (1 - w).r \qquad (3.5')$$

Once we find the solution, we set z^1_{ijCP} and z^2_{ijCP} to d^* and r^*, respectively.

Türeci (2017) takes different actions based on the movement type between target pair (i, j):

- If movement type is 1, there is a single solution, z_{ij}^{k} for these target pairs, so that $z_{ijCP}^{k} = z_{ij}^{k}$, $k = 1,2$.
- If movement type is 2, she optimizes Equation (3.5') subject to Equations (3.6) - (3.8). Let (d^*, r^*) be the optimal values of (d, r).

$$\left(1 - \frac{d - z_{ijMDP}^{1}}{z_{ijMRP}^{1} - z_{ijMDP}^{1}}\right)^{q} + \left(1 - \frac{r - z_{ijMRP}^{2}}{z_{ijMDP}^{2} - z_{ijMRP}^{2}}\right)^{q} = 1 \qquad (3.6)$$

$$z_{ijMDP}^{1} \leq d \leq z_{ijMRP}^{1} \qquad (3.7)$$

$$z_{ijMRP}^{2} \leq r \leq z_{ijMDP}^{2} \qquad (3.8)$$

We set $z_{ijCP}^{1} = d^*$ and $z_{ijCP}^{2} = r^*$.

- If movement type is 3, Türeci (2017) checks all parts of the Pareto-optimal frontier. She finds the threshold weight ($wtan$) using Model ($wtan$), where $wtan$ defines the slope of the line that passes through MDP and is tangent to the L_q function (see Figure 3.6). In addition to MDP and MRP of the trajectories connecting targets *i* and *j*, she also uses another point, tangent point (TP) with objective values $z_{ijTP}^{k}, k = 1,2$ that is the minimum-distance point of the L_q function (Figure 3.6). Let $wtan, d^*$, and r^* be the optimal values of w, d, and r, respectively.

Model ($wtan$)

Max w

s.t.

$$\left(1 - \frac{d - z_{ijTP}^{1}}{z_{ijMRP}^{1} - z_{ijTP}^{1}}\right)^{q} + \left(1 - \frac{r - z_{ijMRP}^{2}}{z_{ijTP}^{2} - z_{ijMRP}^{2}}\right)^{q} = 1 \qquad (3.9)$$

$$w.z^1_{ijMDP} + (1-w).z^2_{ijMDP} = w.d + (1-w).r \quad (3.10)$$

$$z^1_{ijMDP} \le d \le z^1_{ijMRP} \quad (3.11)$$

$$z^2_{ijMRP} \le r \le z^2_{ijMDP} \quad (3.12)$$

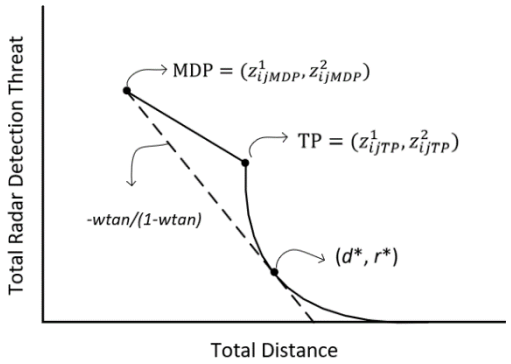
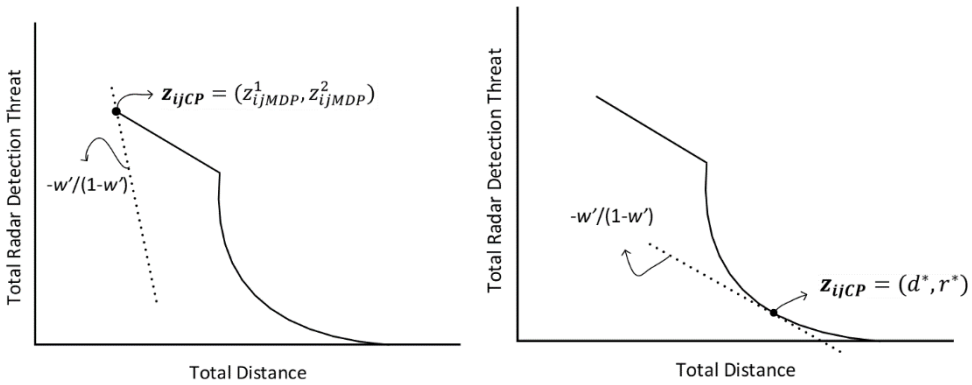


Figure 3.6: Threshold weight

If $|w'| \ge |wtan|$, we set $z^k_{ijCP} = z^k_{ijMDP}$ (Figure 3.7(a)).

If $|w'| < |wtan|$, she solves Equation (3.5') subject to Equations (3.9), (3.11) an (3.12), modifying the lower bound in (3.11) to z^1_{ijTP} and the upper bound in (3.12) to z^2_{ijTP} (Figure 3.7(b)). Let d^* and r^* be the optimal values of d and r, respectively. We set $z^1_{ijCP} = d^*$ and $z^2_{ijCP} = r^*$.



(a) $\mathbf{z_{ijCP}}$ when $|w'| \ge |wtan|$ (b) $\mathbf{z_{ijCP}}$ when $|w'| < |wtan|$

Figure 3.7: Finding z_{ijCP}

3.3.2.3 Selecting *s* Representative Points on the Pareto-Optimal Frontier of Tour π

We choose two extreme Pareto-optimal points, MDP and MRP, and (*s-2*) random points on the Pareto-optimal frontiers as representative points. For selecting the random points, we first find the nadir point of the tour. This point takes its first objective value from MRP ($Z^1_{\pi MRP}$) and the second objective value from MDP ($Z^2_{\pi MDP}$). We then project the nadir point onto the L_q function using different angles. Specifically, we randomly generate *(s-2)* angle values between 0 and 90 degrees and make a projection for each angle. Each tour is then represented with *s* members in the population (see Figure 3.8 for an example with *s*=10).

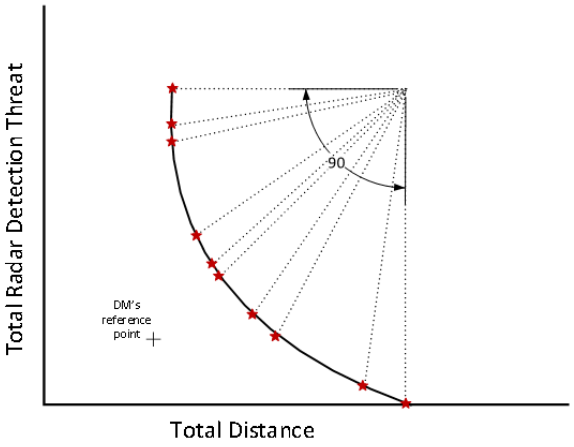


Figure 3.8: Selecting representative points

In Step 0 of FREA_UAV, we generate *M* tours, each having *s* representative points. This results in *N= M.s* solutions

3.3.3 Crossover and Mutation

We use *position-based crossover (POS)* that is developed for path representation (see Larrañaga et al., 1999). Crossover operator is applied to the solutions in the mating pool consisting of the tournament winners. From each of the two parents, *c* positions are randomly selected. The targets at these positions in *parent 1* are transferred

directly to *offspring 2*, and the targets at these positions in *parent 2* are transferred directly to *offspring 1*. The remaining positions in offspring 1 and 2 are filled from parents 1 and 2, respectively, without changing the order of these targets in the parents. Figure 3.9 represents a *POS* operation, where c=2, and randomly selected positions are 3 and 5.

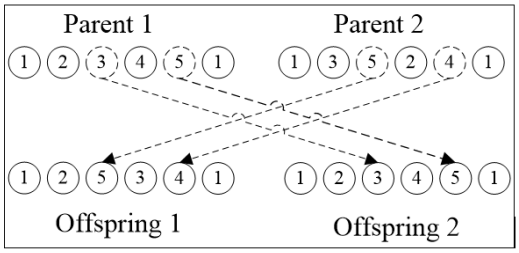


Figure 3.9: POS crossover

For mutation, we use insertion-based mutation operator. After creating an offspring, p random positions are selected from that offspring. The targets at the selected positions are removed and inserted at another randomly selected position. We apply mutation to an offspring with probability *pmut.* We demonstrate the mutation operator in Figure 3.10 for p=1. We randomly select the 4th position, remove the target (target 4) at that position and place it randomly in the 5th position.

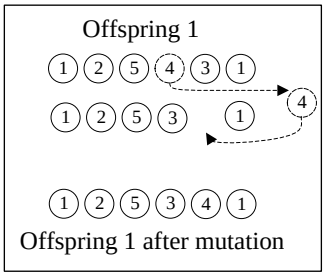


Figure 3.10: Mutation operator

While generating the offspring population in Step 4 in FREA_UAV, we perform crossover and mutation $\frac{M}{2}$ times.

3.4 Finding Trajectories of the Nondominated Solutions

Upon termination of FREA_UAV, we find a set of relatively nondominated solutions close to the reference points of the RP. Each chromosome in the final set contains the tour information (the order of visit to the targets, π) and the values of the two objective functions (distance, D_π, and radar detection threat, RDT_π). Although we know the total values of the two objectives for a tour, we do not know which combination of the trajectories leads to those objective function values.

After finding the relatively nondominated solutions close to the reference points, we find the trajectories to be used between target pairs in those tours. We solve a modified version of the nonlinear mathematical model developed by Tezcaner Öztürk and Köksalan (2018). We present the model in Appendix B.1. This model selects approximately efficient trajectories to be used between consecutive target pairs in a tour, π, making sure that the summation of the first objective values (d_{ij}, between $(i,j) \in \pi$) of the trajectories equals at most the tour's total distance value, D_π. As the objective function, we minimize the summation of the radar detection threat values of the used trajectories (rdt_{ij}, between $(i,j) \in \pi$). The model selects one point on the approximated Pareto-optimal frontier of each consecutive target pair. These constraints make the model nonlinear, as Type 2 and 3 frontiers are approximated by L_q functions. Still, the resulting trajectories are approximations, since they are selected from the approximated Pareto-optimal frontiers.

After finding the objective values of the trajectories to be used between each consecutive target pair, we find the real efficient trajectory between each consecutive target pair using the heuristic developed by Tezcaner Öztürk and Köksalan (2018). This heuristic finds the efficient trajectory between a target pair with a distance value, d_{ij}, using Golden Section Search. In our application, we omit the rdt_{ij} information obtained after solving the nonlinear mathematical model, and proceed with the d_{ij} values and solve the heuristic. We then obtain the rdt_{ij} value of

trajectory. The resulting trajectories with objective values d_{ij} and rdt_{ij} that are approximately efficient, are presented to the RP.

3.5 Results of the Bi-Objective UAV Route Planning Problem

We programmed FREA_UAV in R (R Core Team, 2021) and tested it on 5, 9, and 15-target problems on a PC with Intel Core i7-4770S CPU, 3.10Ghz, 16 GB RAM. We set the population size to a small value to keep the computational effort low. We thus set the number of tours to M=20 and we represent each tour by 10 points, resulting in a population size of $M.s$ = 200. We tried several generation numbers, observed that the results were robust to those changes, and decided to continue with a small generation number of 50. All population members participate in a tournament twice, hence 200 tournaments are realized with 200 winners. During crossover, c=2, 5, and 8 random target positions directly pass from the parents for 5, 9, and 15-target problems, respectively. In choosing these values, we tried to keep the c to number of targets ratios similar to those of Köksalan and Tezcaner Öztürk (2017). We also use mutation and crossover probabilities as in Köksalan and Tezcaner Öztürk (2017). If the two parents of an offspring are identical, mutation is applied with probability $pmut$=1.0. Otherwise, mutation is applied with probability $pmut$=0.1. We use a crossover probability of 0.9. After each crossover and mutation, two offspring are generated. For each offspring tour, s=10 representative points are selected. Therefore, to produce 200 offspring, M/2=10 crossover and mutation operations are applied to the winners. In all the figures for the results, the reference points are indicated with the plus signs (+). The archive population is represented with (x). We give the spread of the initial, final, and the archive populations in the same figure. All reported solution times are in terms of CPU seconds.

For the evaluation of the results, we find 100 Pareto-optimal points on the approximated Pareto-optimal frontiers of the problems using the nonlinear mathematical model developed by Tezcaner Öztürk and Köksalan (2018). We use BARON solver in GAMS Optimization Package (Tawarmalani and Sahinidis, 2005;

Sahinidis, 2017). We then find the closest of these 100 Pareto-optimal points to the reference point in terms of the normalized Euclidean distance and compare this closest solution with the closest solution found with FREA_UAV. The closer the distances to the reference points of the solutions found by FREA_UAV and by the mathematical model, the better the performance of FREA_UAV.

3.5.1 5-Target Bi-Objective UAV Route Planning Problem

This instance is developed by Tezcaner Öztürk and Köksalan (2018). The placements of the targets and the radars on a 400 km^2 terrain can be seen in Figure 3.13(b-c). The targets are placed at coordinates (3.0, 17.0), (10.0, 9.0), (6.0, 1.0), (15.0, 3.0), and (16.0, 14.0) and the radars are located at coordinates (5.0, 12.0), (9.0, 5.0), (12.0, 16.0), and (16.0, 8.0). The radii of outer (2.9108 km) and inner (1.2274 km) radar areas are as in Tezcaner and Köksalan (2011), based on the radar detection threat formulation of Gudaitis (1994) (see Appendix A). Different terrain sizes can be represented easily by rescaling the dimensions.

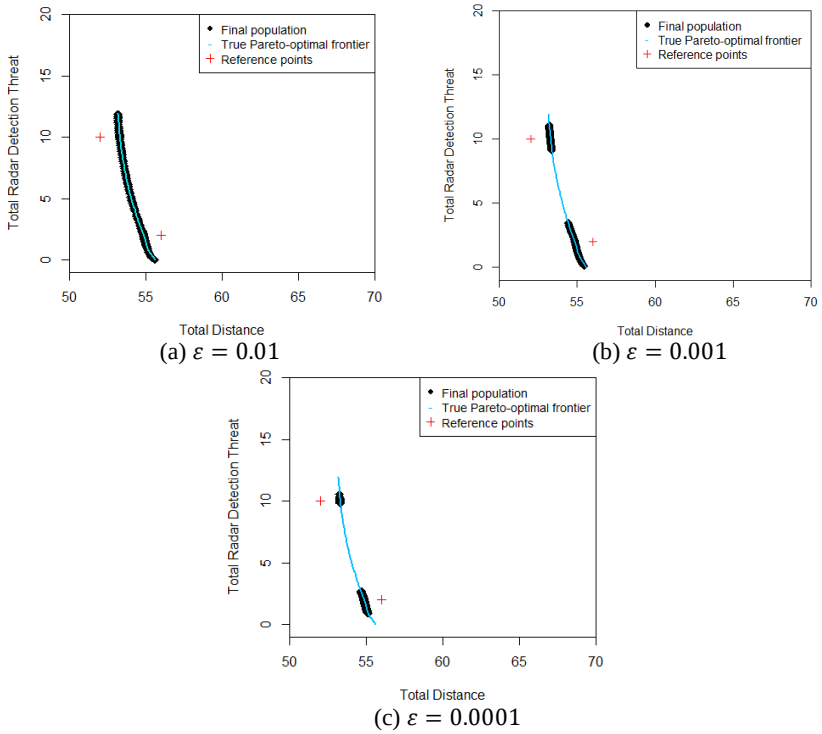
We first explain the effects of using different ε values for the ε-niching procedure. We then give our computational results. We experiment with different reference points; but due to the similarity of the results, we only present one case with two reference points.

3.5.1.1 Effects of Different ε Values

We experiment with different ε values in the ε-niching procedure in evaluating the population for this problem. The results can be seen in Figure 3.11. As ε gets larger, we obtain a more diverse set of solutions that are close to the reference points. Thus, the value of ε can be set depending on the preferences of the DM on the spread of the final solutions.

The computational times increase only slightly with decreasing ε. In all our remaining experiments, we use ε as 0.001 as in Deb and Sundar (2006) in the ε-niching procedure in evaluating the population.

We also experiment with different initial values for ε in forming *Archive*. We give the results for this problem in Figure 3.12. We set *Capacity* to 50 (that is, 25% of the population), and *TempCapacity* to 10. As the ε value increases, the size of *Archive* fluctuates more. For a small ε value, *Archive* size stays stable and close to 50. Again, the computational times increase only slightly with decreasing ε. In all our experiments, we set the initial ε to 0.001 to keep *Archive* at a stable size.



(a) $\varepsilon = 0.01$ (b) $\varepsilon = 0.001$

(c) $\varepsilon = 0.0001$

Figure 3.11: The effects of different ε values on the distribution of the final set for the 5-Target UAV route planning problem

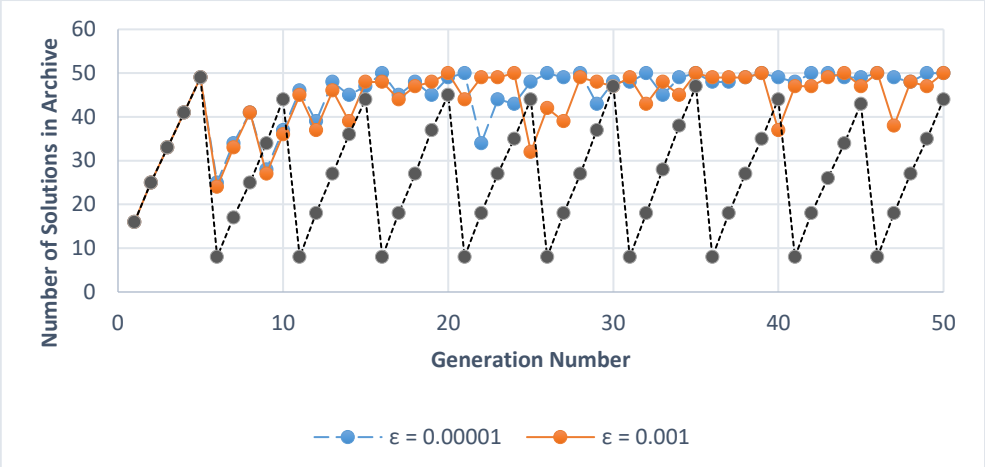


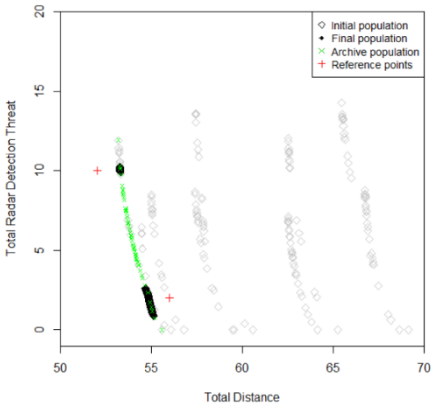
Figure 3.12: The effects of using different ε values for archive for the 5-Target UAV route planning problem

3.5.1.2 Demonstration of FREA_UAV on the 5-Target Problem

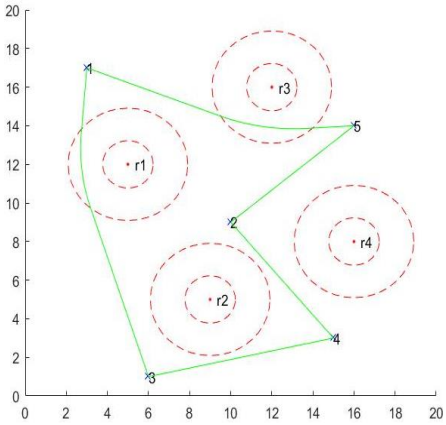
We choose two reference points with objective values (*D*=56.0, *RDT*=2.0) and (*D*=52.0, *RDT*=10.0). The first reference point is dominated by some parts of the Pareto-optimal frontier and the second is a point that dominates some parts of the Pareto-optimal frontier. As can be seen in Figure 3.13(a), the algorithm converges to Pareto-optimal points that are close to the reference points. In this problem, FREA_UAV converges to multiple reference points simultaneously when the RP wants to explore different regions of the Pareto-optimal frontier. The solution duration is 38.71 seconds.

Among the results of FREA_UAV, the closest observation to reference point (56.0, 2.0) is from tour 1-3-4-2-5-1 with *D*=54.896 and *RDT*=1.816. The closest observation to reference point (52.0, 10.0) is from tour 1-2-3-4-5-1 with *D*=53.283 and *RDT*=10.089. From the set of approximated Pareto-optimal points we generated using the nonlinear mathematical model of Tezcaner Öztürk and Köksalan (2018), we find the closest points to the first and second reference points as (*D*=54.900, *RDT*=1.815), and (*D*=53.270, *RDT*=10.112), respectively. The normalized Euclidean distances between the first reference point and the closest points found

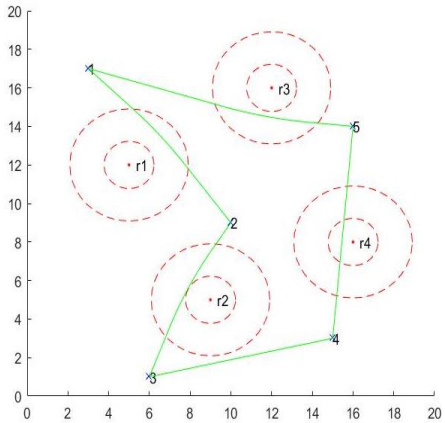
using the nonlinear mathematical model and FREA_UAV are 0.0702 and 0.0704, respectively. The distances between the second reference point and the closest points found using the nonlinear mathematical model and FREA_UAV are 0.0801 and 0.0807, respectively. Considering these distance values, we infer that the solutions found by FREA_UAV are at practically the same distance to their respective reference points as are the exact solutions found by the nonlinear mathematical model.



(a) Results of FREA_UAV



(b) Closest solution to reference point (*D*=56.0, *RDT*=2.0)



(c) Closest solution to reference point (*D*=52.0, *RDT*=10.0)

Figure 3.13: Results for the 5-Target UAV route planning problem

Using the tour, *D*, and *RDT* information obtained with evolutionary algorithm as an input, we find the trajectories to be used between consecutive target pairs using the method explained in Section 3.4. The resulting distance (*d*) and radar detection threat

values (*rdt*) between target pairs are given in Table 3.2. The final routes and the corresponding trajectory information of the solutions that are closest to the first and second reference points are shown in Figure 3.13(b) and (c), respectively.

Table 3.2: Trajectories between target pairs of the closest solutions to reference points (*D*=56.0, *RDT*=2.0) and (*D*=52.0, *RDT*=10.0)

Reference Point: (56.0, 2.0) FREA_UAV Result: (54.896, 1.816)		**1-->3**	**3-->4**	**4-->2**	**2-->5**	**5-->1**	**Total**
Trajectory Information	***d***	16.505	9.220	7.810	7.810	13.550	54.896
	rdt	0.776	0.000	0.000	0.000	1.091	1.867

Reference Point: (52.0, 10.0) FREA_UAV Result: (53.283, 10.089)		**1-->2**	**2-->3**	**3-->4**	**4-->5**	**5-->1**	**Total**
Trajectory Information	***d***	10.658	8.978	9.220	11.045	13.382	53.283
	rdt	0.907	2.918	0.000	3.733	2.265	9.823

3.5.1.3 Comparison FREA_UAV and R-NSGA-II on the 5-Target Problem

We also run R-NSGA-II on the 5-target bi-objective UAV route planning problem for the same reference points, (*D*=56.0, *RDT*=2.0) and (*D*=52.0, *RDT*=10.0), and for the same parameter settings as in FREA_UAV. The final population of R-NSGA-II (Figure 3.14) is very close to that of FREA_UAV (Figure 3.13(a)).

Among the results of R-NSGA-II, the closest observation to reference point (56.0, 2.0) is from tour 1-3-4-2-5-1 with *D*=54.897 and *RDT*=1.811 and the closest observation to reference point (52.0, 10.0) is from tour 1-2-3-4-5-1 with *D*=53.283 and *RDT*=10.089. The normalized Euclidean distances between the first and the second reference points to the closest points found are 0.0705, and 0.0807, respectively. These values are very close to the corresponding results of FREA_UAV. The solution time of R-NSGA-II is around 80% of that of FREA_UAV.

As seen from the results, the two algorithms perform similarly for the 5-target UAV problem. The main reason for this is that this problem has a continuous frontier (see Figure 3.13(a)), and R-NSGA-II does not suffer from the shortcoming we discussed for the cases of discontinuous frontiers. FREA_UAV, unlike R-NSGA-II, also produces solutions that represent the whole spectrum of the Pareto-optimal frontier. This not only puts the obtained solutions in perspective of the efficient frontier, but also provides valuable information if a decision maker wants to further explore other parts of the frontier updating the reference point(s).

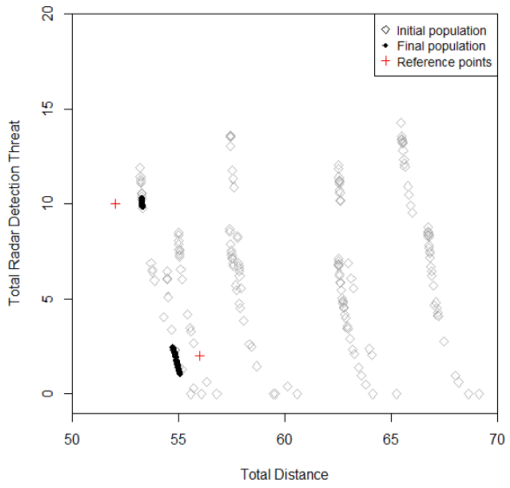


Figure 3.14: Results of R-NSGA-II for the 5-Target UAV route planning problem

3.5.2 9-Target Bi-Objective UAV Route Planning Problem

This problem is developed by Türeci (2017). The placement of the targets and radars on a 400 km^2 terrain can be seen in Figure 15 (b-c). The targets are located at coordinates (16.6, 8.0), (7.0, 13.2), (6.6, 2.2), (2.0, 7.2), (15.4, 1.8), (13.2, 13.4), (0.5, 16.9), (10.6, 7.7), and (0.7, 0.7) and the radars are located at coordinates (13.7, 4.9), (6.6, 6.8), and (3.6, 12.7). The effective regions of the radars are the same as in the 5-target problem discussed above.

3.5.2.1 Demonstration of FREA_UAV on the 9-Target Problem

The results for two reference points (one dominating some of the Pareto-optimal points, and the other is dominated by some of the Pareto-optimal points) are given in Figure 3.15(a). The algorithm finds solutions close to both reference points. The closest observation we find to the reference point (*D*=60.0, *RDT*=5.0) is from tour 1-6-2-7-4-9-3-8-5-1 with *D*=63.516 and *RDT*=5.091. The closest observation to the other reference point (*D*=68.0, *RDT*=1.5) is from tour 1-8-6-2-7-4-9-3-5-1 with *D*=63.619 and *RDT*=1.413. The solution duration is 37.37 seconds.

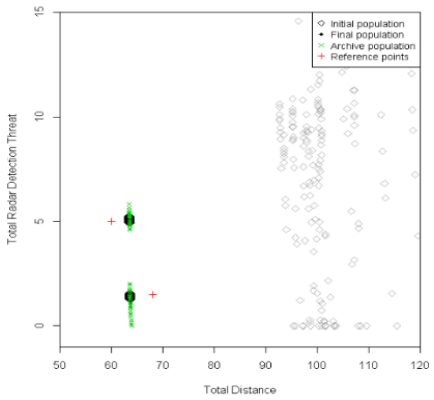
From the set of Pareto-optimal points we generated using the nonlinear mathematical model of Tezcaner Öztürk and Köksalan (2018), the closest points to the first and second reference points are (*D*=63.501, *RDT*=5.051) and (*D*=63.617, *RDT*=1.433), respectively. The normalized Euclidean distances between the closest points found using the nonlinear mathematical model and FREA_UAV are 0.0883 and 0.0888 for the first reference point, and 0.1106 and 0.1105 for the second reference point, respectively. Considering these distance values, we infer that the solutions found by FREA_UAV are at practically the same distance to their respective reference points as are the exact solutions found by the nonlinear mathematical model.

Using the tour, *D*, and *RDT* information obtained with FREA_UAV, we find the trajectories to be used between each consecutive target pair. The distance (*d*) and radar detection threat values (*rdt*) between target pairs for both tours are given in Table 3.3. The routes based on the results of the mathematical model, which are closest to the first and second reference points, are shown in Figure 3.15(b) and (c), respectively.

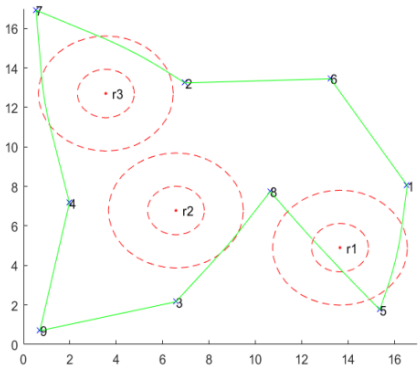
Table 3.3: Trajectories between target pairs of the closest solutions to reference points (D=60.0, RDT=5.0) and (D=68.0, RDT=1.5)

Reference Point: (60.0, 5.0) FREA_UAV: (63.516, 5.091)		**1→6**	**6→2**	**2-→7**	**7→4**	**4→9**	**9→3**	**3→8**	**8→5**	**5→1**	**Total**
Trajectory Information	***d***	6.341	6.278	7.429	9.874	6.596	6.051	6.908	7.627	6.413	63.516
	rdt	0.000	0.000	0.508	0.171	0.000	0.000	0.027	3.656	0.404	4.766

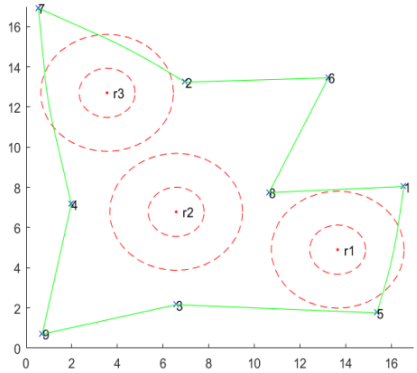
Reference Point: (68.0, 1.5) FREA_UAV: (63.619, 1.413)		**1→8**	**8→6**	**6→2**	**2→7**	**7→4**	**4→9**	**9→3**	**3→5**	**5→1**	**Total**
Trajectory Information	***d***	5.922	6.273	6.278	7.420	9.868	6.596	6.051	8.806	6.406	63.619
	rdt	0.000	0.000	0.000	0.589	0.228	0.000	0.000	0.000	0.468	1.284



(a) Results of FREA_UAV



(b) Closest solution to reference point (D=60.0, RDT=5.0)

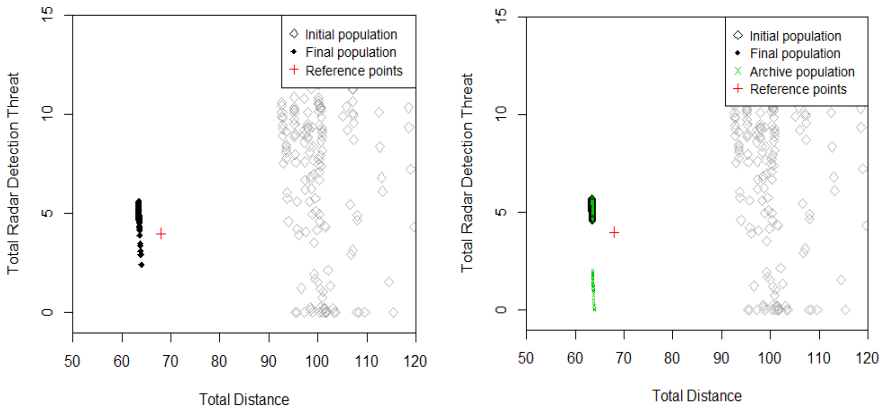


(c) Closest solution to reference point (D=68.0, RDT=1.5)

Figure 3.15: Results for the 9-Target UAV route planning problem

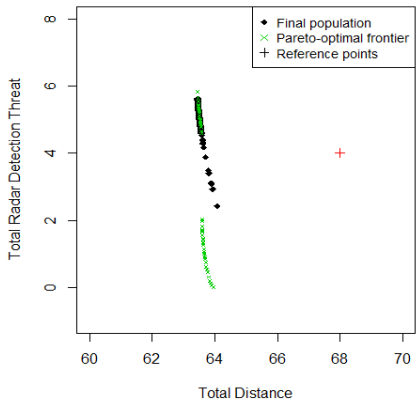
3.5.2.2 Comparison of FREA_UAV and R-NSGA-II on the 9-Target Problem

We compare R-NSGA-II and FREA_UAV on the 9-target bi-objective UAV route planning problem where a dominated reference point is provided by the DM. We use the same parameter setting for both algorithms. Solutions of the final generation of R-NSGA-II and our algorithm are given in Figure 3.16(a) and (b), respectively. Some of the final solutions of R-NSGA-II are dominated by the true Pareto-optimal solutions, which can be seen in Figure 3.16 (c), which is an enlarged view of Figure 3.16(a). The solution durations of FREA_UAV and R-NSGA-II are 34.62 and 25.71 seconds, respectively.



(a) Results of R-NSGA-II

(b) Results of FREA_UAV



(c) Enlarged view of (a)

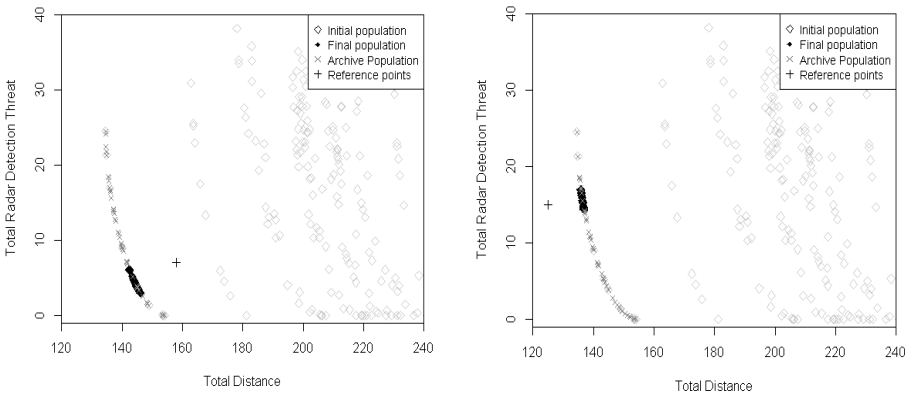
Figure 3.16: Comparison of R-NSGA-II and FREA_UAV on the 9-Target UAV route planning problem

3.5.3 15-Target Bi-Objective UAV Route Planning Problem

We also demonstrate FREA_UAV on a larger problem with 15 targets. For this problem, we did not place the targets or radars in a terrain, but instead, we randomly generated the movement between each target pair as follows: First, we randomly set the movement type (Type 1, 2, or 3) between each target pair. Then we randomly generated corresponding extreme Pareto-optimal points and L_q functions for each target pair. We implemented FREA_UAV on this hypothetical problem and observed that although solution space gets significantly larger than those of 5 and 9-target problems, computational times increase only slightly. We set generation number to 200 in this problem.

The closest solution to the reference point (*D*=158.0, *RDT*=8.0) that is dominated by some of the Pareto-optimal points, is 1-6-9-3-4-2-10-7-13-14-8-12-11-15-5-1 with *D*= 140.251 and *RDT*= 8.795 (Figure 3.17(a)). The closest solution to the reference point (*D*=125.0, *RDT*=15.0) that dominates some of the Pareto-optimal points is 1-6-8-10-15-11-12-3-13-7-4-2-14-9-5-1 with *D*=136.499 and *RDT*=15.567 (Figure 3.17(b)). The solution durations are 224.8 and 232.1 seconds for the first and second reference points, respectively.

From the set of Pareto-optimal solutions we generated using the nonlinear mathematical model of Tezcaner Öztürk and Köksalan (2018), we find the closest points to the first and second reference points as (*D*=141.161, *RDT*=7.317) and (*D*=136.661, *RDT*=16.019), respectively. The normalized Euclidean distances between the closest points found using the nonlinear mathematical model and FREA_UAV are 0.1517 and 0.1601 for the first reference point, and 0.1072 and 0.1039 for the second reference point, respectively. Both methods result in solutions with very close distance values to the reference points.



(a) Reference point (*D*=158.0, *RDT*= 8.0) (b) Reference point (*D*=125.0, *RDT*=15.0)

Figure 3.17: Results for the 15-Target UAV route planning problem

3.5.4 Updating Reference Points

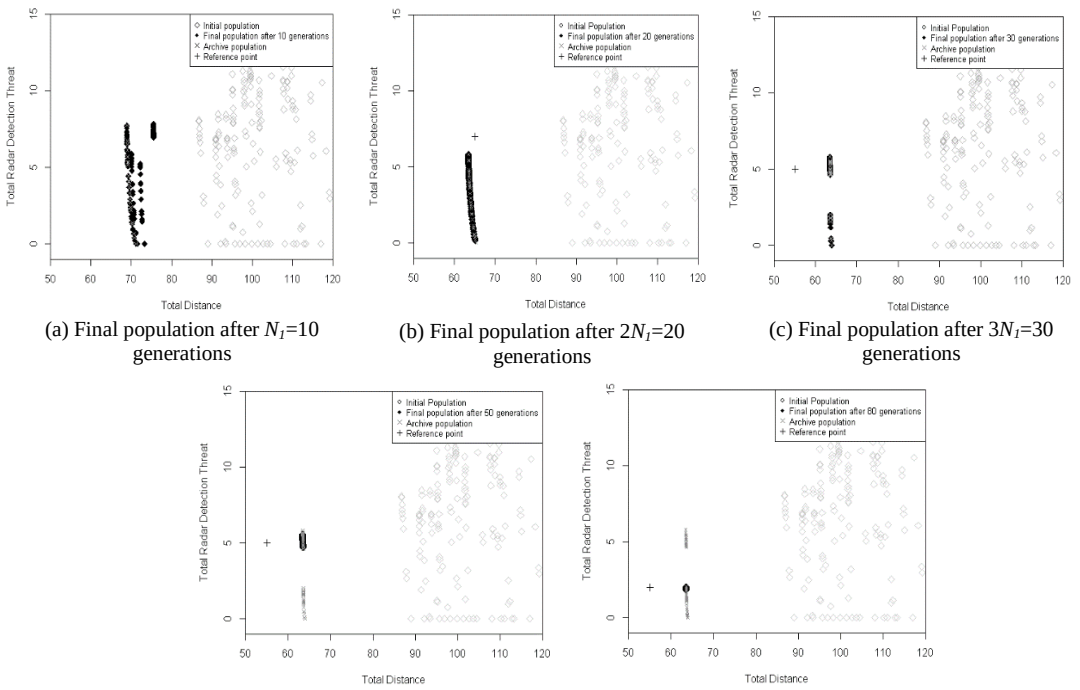
We next consider a case where the RP updates his/her reference points during the search for the 9-target case.

In order to see the effects of changing reference points in different directions, we consider two scenarios. In the first scenario, the RP changes his/her reference points towards the southwest of the search region (to points with better values in both objectives). In the second scenario, the RP changes his/her reference points three times, each time either to southwest (better objective values) or to northwest (better in distance and worse in detection threat objective). After each update of the reference point, we run the algorithm for N_1 more generations. If the RP does not change the reference point, we run N_2 more generations and terminate. In the following two examples, we set N_1 to 10 and N_2 to 20 and were able to obtain satisfactory convergence in so few generations.

3.5.4.1 9-Target Scenario 1

We start with an arbitrary reference point (*D*=0.0, *RDT*= 0.0), run the algorithm for N_1 generations. We present the spread of solutions in the objective space to the RP

(Figure 3.18(a)). The RP then chooses (*D*=65.0, *RDT*=7.0) as his/her initial reference point. We run the algorithm for N_1 more generations, and the reference point turns out to be dominated (Figure 3.18(b)). The RP provides a new reference point (*D*=55.0, *RDT*=5.0). We run the algorithm for N_1 more generations (Figure 3.18(c)). After observing the spread of solutions, the RP does not change his/her reference point. We run the algorithm N_2 more generations and terminate (Figure 3.18(d)). After these ($3N_1+ N_2$) generations, we present the final and archive populations to the RP. To demonstrate an extreme case, we assume that the RP wants to discover another region of the objective space and sets his/her new reference point as (*D*=55.0, *RDT*=2.0). We run the algorithm for N_1 more generations, and he/she does not change his/her final reference point. We run the algorithm N_2 more generations and terminate. Our algorithm converges quickly again to the new region (in approximately 10 seconds for each reference point) as shown in Figure 3.18(e).



(a) Final population after N_1=10 generations

(b) Final population after $2N_1$=20 generations

(c) Final population after $3N_1$=30 generations

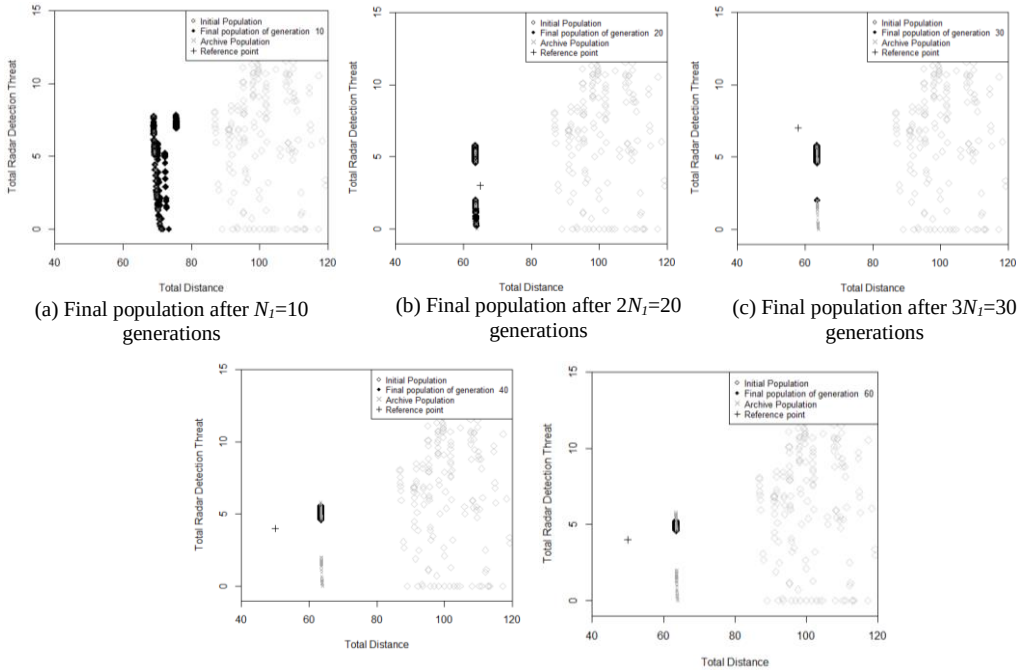
(d) Final population after $3N_1+N_2$=50 generations

(e) Final population after $4N_1+2N_2$=80 generations

Figure 3.18: Changing reference points - First scenario

3.5.4.2 9-Target Scenario 2

We start with an arbitrary reference point (D=0.0, RDT= 0.0). RP's first, second, and third reference points are (D=65.0, RDT= 3.0), (D=58.0, RDT= 7.0), and (D=50.0, RDT= 4.0), respectively. Each time the RP provides a new reference point, we run the algorithm N_1 more generations. When the RP does not change his/her reference point, we run the algorithm N_2 more generations before termination. We present the results in Figure 3.19. In this scenario too, the algorithm quickly converges to a new region (in approximately 10 seconds for each reference point), even when the new reference point is considerably different from the current reference point. Notice that these computational durations are short.



(a) Final population after N_1=10 generations

(b) Final population after $2N_1$=20 generations

(c) Final population after $3N_1$=30 generations

(d) Final population after $4N_1$=40 generations

(e) Final population after $4N_1+N_2$=60 generations

Figure 3.19: Changing reference points - Second scenario

3.6 Discussions for Chapter 3

In this chapter, we develop a preference-based MOEA, FREA. FREA uses the reference points defined by DMs to converge to the preferred regions of Pareto-optimal frontiers. It eliminates an important shortcoming of R-NSGA-II, which occasionally ends up with solutions that are dominated for problems having discontinuous Pareto-frontiers, if the reference point is dominated. Our archive mechanism prevents obtaining dominated solutions, and creates a robust environment in case the DM wishes to change the reference points throughout the algorithm.

We also further develop our algorithm for the complex multi-objective UAV route planning problem in continuous terrain, where there are multiple targets to be visited and each target pair has infinitely many connections. We find the order of visit to the targets and the trajectories to use between target pairs considering two objectives: minimizing distance traveled and minimizing radar detection threat. Infinitely many connections between target pairs and two conflicting objectives result in infinitely many efficient solutions. We develop the specifics of FREA for the UAV route planning problem and demonstrate it on randomly generated UAV route planning problems with 5, 9, and 15 targets. The results show fast convergence to most preferred regions of the DM. The results on the UAV instances and other well-known problems also show that even when the DM changes his/her reference points to explore completely different regions of the Pareto-optimal frontier, the algorithm quickly converges to new regions. Although there is no guarantee for heuristic algorithms to converge to the Pareto-optimal frontier, FREA always ended with solutions that are either on or arbitrarily close to the Pareto-optimal frontiers in our computational experiments.

As a future work, the UAV route planning problem studied here can be enriched with considerations of additional meaningful objectives and practical issues such as ground obstacles, moving radars and targets, unknown radar threats and etc. Studying the problem with multiple UAVs is another future extension direction. The

number of UAVs to be used can be considered as an objective function since it may have significant impacts on the total flight duration the fleet spends and total radar detection threat the fleet is exposed to.

FREA is a generic MOEA that can be used for other multi-objective optimization problems as well. FREA can be adapted for the problems from different fields to investigate the capabilities and performance of the algorithm. Different approaches for solution representation, fitness assignment, evaluation of the population, etc. can be developed for FREA in order to improve the performance of the algorithm.

FREA considers the preferences of the DM through reference points, and finds solutions that are close to the reference points. The point closest to the reference point can be considered as a desirable solution for the DM. Another fruitful extension of this study may be consideration of an interactive preference-based MOEA. Although FREA provides the flexibility of changing reference point(s) during the algorithm, interactions of the user with the algorithm are limited. Besides, providing reference points in the objective space may not be an easy task for a DM. As an alternative approach, an interactive preference-based MOEA such as Phelps and Köksalan (2003) and Köksalan and Karahan (2010) can be developed. In interactive algorithms, the preference information is obtained from a DM through his/her comparisons among objectives and/or solutions progressively throughout the algorithm.

CHAPTER 4

THREE-OBJECTIVE UAV ROUTE PLANNING PROBLEM WITH TIME-DEPENDENT PRIZES AND IMPLEMENTATION OF FREA

In this chapter, we extend the bi-objective UAV route planning problem presented in Chapter 3 by introducing "maximization of the collected information" as an additional objective. Now there are three routing objectives: (1) minimization of the length of the tour, (2) minimization of the radar detection threat of the tour, and (3) maximization of the information collected from the mission terrain. In this study, similar to the bi-objective version, the terrain the UAV moves is considered as continuous terrain allowing for infinitely many trajectory options between targets. The UAV starts its movement from a base, visits all targets in an enemy terrain monitored by radars, and returns to the initial base. The aim is to determine both the visiting order of the targets and the specific trajectories to use between consecutive target pairs. The structure of the problem is similar to TSP in the broader sense. Specifically, the problem can be classified as a variant of TSP with Profits (Feillet et al., 2005).

The length of the route of the UAV is minimized since fuel or battery consumption is an important consideration. Avoiding detection is another important objective as getting detected may cause the mission to fail. Information collection is maximized since the majority of the surveillance and reconnaissance operations are performed to collect information about the movements and assets of enemies. Prize collection is one of the major considerations in recent UAV route planning studies as well (see, for example, Ergezer and Leblebicioglu, 2013; Moskal and Batta, 2017; Xia et al., 2017). However, maximizing the collected information is a meaningful objective for the missions where only some targets can be visited due to the mission restrictions. Hence, in the literature, the problem is often modelled as an orienteering problem (OP) (Golden et al., 1987) where the subset of targets whose visits will give the

largest benefit in terms of information collection is determined. On the other hand, for the missions where all targets must be visited, as in the one considered in this chapter, the problem turns to be a TSP, and maximization of the information collected becomes a meaningless objective as the total collected information is the same in all feasible solutions. In order to make information collection a meaningful objective for this problem, we consider time-dependent prizes. The amount of information at the targets varies throughout a day, and the collected information from a target depends on the time that the target is visited. Indeed, time-dependent information at enemy targets is often the case in practical reconnaissance operations. For example, the prior knowledge about the time that enemy assets will likely be moved into or away from a target region may have substantial impacts on the available information at the target.

A similar problem in the literature is Vehicle Routing Problem with Time Windows (VRPTW) (Braysy and Gendreau, 2005). In VRPTW, time windows are generally defined as strict constraints such that a vehicle has to arrive at the target on a given time interval. A variant of VRPTW where the strict time window constraints are relaxed is Vehicle Routing Problem with Soft Time Windows (VRPSTW) (Figliozzi, 2010). In this problem, time windows are not strict constraints; i.e. not reaching to the target on time does not make the problem infeasible, but often a penalty is introduced to penalize the arrivals that are outside the time windows. VRPSTW has been generally studied for the ground vehicle routing (see, for example, Calvete et al., 2007; Fu et al., 2008; Figliozzi, 2010). To the best of our knowledge, the only study in the literature considering soft time windows concept for the UAV route planning problem is Guerriero et al. (2014). They develop exact and heuristic solution approaches for the multi-objective route planning problem of a fleet of UAVs filming a sport event. In our problem, the time-dependency in information collection is similar to the soft time window concept. The UAV is allowed to collect information from a target at any time but a penalty in terms of lower information gain is introduced if the UAV does not reach the target when the available information is highest.

Similar to the bi-objective version presented in Chapter 3, the UAV route planning problem in continuous terrain with three objectives, multiple targets and time-dependent prizes is a complex problem with infinitely many efficient solutions. We address this problem adapting FREA_UAV, the preference-based MOEA developed for the bi-objective version of the problem in Chapter 3, to account for the additional objective and time-dependent information collection. We demonstrate the adapted algorithm on several problem instances.

The contributions of this work to the literature are similar to those of Chapter 3:

1. UAV route planning problem is studied considering multiple targets, three objective functions, continuous terrain with infinitely many trajectory options between targets. The trade-offs among information collection, flight duration and risk of being detected have not been considered in the literature.
2. Time-dependency in information collection, which may be the case in many practical reconnaissance and surveillance missions, is considered.
3. A heuristic solution approach is provided by adapting FREA_UAV (the preference-based MOEA developed in Chapter 3).

This chapter is organized as follows. In Section 4.1, we explain the problem structure of the three-objective UAV route planning problem studied in this chapter. In Section 4.2, implementation of FREA_UAV on the problem is presented. In Section 4.3, the solution approach is demonstrated on some example problems and results are reported. In Section 4.4, the chapter is concluded with a discussion.

4.1 Problem Structure of the Three-Objective UAV Route Planning Problem

The problem structure of the three-objective UAV route planning problem is the same as that of the bi-objective version presented in the previous chapter (see

Chapter 3.1), except that each target now has its own soft time windows presenting the change in the available information over time.

The UAV starts its movement from a base, visits all targets and returns to the initial location. The route plan requires determining both the visiting order of the targets and the specific trajectories to use between consecutive target pairs. Three routing objectives are considered. The two objectives, minimizing the distance (*D*) and the radar detection threat (*RDT*) are the objectives used in the bi-objective version. In addition to those two, we now consider maximizing the collected information *(I)* from the mission terrain. The maximum time the UAV is allowed to operate is assumed to be 24 hours, and each target has its own time-dependent information structure where the available information varies throughout 24 hours. The UAV is allowed to collect information at any time within 24 hours at a target, but the collected information depends on the time of the visit. We assume that the information takes integer values between 1 and 5, where the former is the lowest and the latter is the highest level. An example time-dependent information structure is presented in Figure 4.1. The horizontal axis indicates the arrival time of the UAV to the target and the vertical axis indicates the corresponding information level.

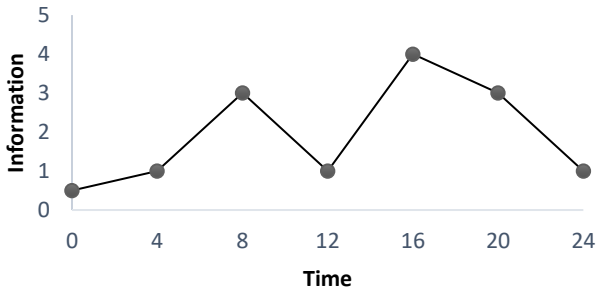


Figure 4.1: An example showing information change over time

Similar to the bi-objective problem, we consider continuous terrain when modeling the movement of the UAV. To model the movement of the UAV in the continuous terrain, we use the same approach that we used for the bi-objective version of the problem (see Section 3.1.3). According to that approach, the UAV has three types of

possible movements when moving between a target pair: The movement is type 1 if the shortest-distance path between the target pair does not pass through any radar region, type 2 if the shortest-distance path between the target pair passes through only the outer radar region, and type 3 if the shortest-distance path between the target pair passes through both the outer and the inner radar regions. We approximate the Pareto-optimal frontiers of type 2 and 3 movements using Köksalan (1999)'s L_q functions. Tezcaner Öztürk and Köksalan (2018) show that Pareto-optimal frontiers of type 2 and 3 movements are similar in structure to the curves that result from L_q functions. We refer readers to see Section 3.1.3. for more details of the approximation of the Pareto-optimal frontiers of type 2 and 3 movements with L_q functions.

4.2 Implementing FREA_UAV on the 3-Objective UAV Route Planning Problem

In Chapter 3, we develop a preference-based MOEA (FREA) and adapt it for the UAV route planning problem (FREA_UAV). Here, we adapt FREA_UAV for the 3-objective version of the problem.

In FREA_UAV, each solution in the population corresponds to a tour. We use the path representation scheme for the solutions, i.e. if target i is the j^{th} target to be visited, it is placed in the j^{th} position in the chromosome. This is the chromosome structure generally used for TSP (see, for example, Köksalan and Tezcaner Öztürk, 2017). Each solution corresponds to a tour and is evaluated according to its performance on three objectives, (1) distance traveled, (2) radar detection threat exposed and (3) information collected, throughout generations of FREA_UAV. However, each tour has a continuous Pareto-optimal frontier having trade-offs among three objectives rather than a single fitness value due to the infinitely many trajectory options between consecutively visited targets in that tour. Therefore, it is not meaningful to assign a unique fitness value to a solution. As the Pareto-optimal

frontier of a tour has infinitely many points, it is difficult and not practical to generate all those points. Similar to the solution approach we used in the bi-objective case in the previous chapter, here we first approximate the Pareto-optimal frontier of a tour using L_q function, and then choose several points on the approximated frontier, each of which becomes a separate member of the population implicitly representing the same tour. Köksalan and Lokman (2009) show that L_q functions are able to successfully approximate the Pareto-optimal frontiers of problems with three and four objectives. We next provide the details of fitting the L_q function to a tour, selecting the points on the tour, and assigning fitness vectors to these points.

4.2.1 Assigning Fitness Values with Three Objectives

We first approximate the Pareto-optimal frontier of a tour with an L_q function, and then choose several points on the approximated frontier of the tour. Each selected point becomes a separate member of the population of the evolutionary algorithm, and is represented with its corresponding total distance traveled, total radar and information collection values. To select the representative points for a tour and determine their fitness values, we follow the following steps:

Step 1. Approximate the Pareto-optimal frontier of the tour with L_q function

1.1. Find the minimum and maximum possible values of each objective

1.2. Find a central point

1.3. Fit an L_q function

Step 2. Select representative points on the approximated Pareto-optimal frontier

We next provide the details of these steps.

4.2.1.1 Fitting an L_q Function to the Pareto-optimal Frontier of Tour π with Three Objectives

We fit the L_q function to a tour π using the three extreme and a central Pareto-optimal solution of the tour using Equation (4.1). Let $d_{\pi t},\ r_{\pi t},\ i_{\pi t}$ represent the normalized distance, radar detection threat, and the collected information collected values of a point t of tour π, respectively. In Equation (4.1), the normalized objective function values of the central point (CP) of tour π are used to fit the L_q function. To normalize the k^{th} objective value of CP, the minimum and maximum values of the k^{th} objective in the Pareto-optimal frontier of tour π is used. Let $Z^k_{\pi Min}$ and $Z^k_{\pi Max}$ represent the minimum and maximum values in the Pareto-optimal frontier of tour π for the k^{th} objective, respectively, and $Z^k_{\pi CP}$ denote the k^{th} objective value of the central point CP , where $k = 1,2,3$ corresponds to (1) distance, (2) radar detection threat, and (3) information collection, objectives.

$$(1 - d_{\pi CP})^q + (1 - r_{\pi CP})^q + (1 - i_{\pi CP})^q = 1$$
$$\text{where, } (d_{\pi CP}, r_{\pi CP}, i_{\pi CP}) = \left(\frac{Z^1_{\pi CP} - Z^1_{\pi Min}}{Z^1_{\pi Max} - Z^1_{\pi Min}}, \frac{Z^2_{\pi CP} - Z^2_{\pi Min}}{Z^2_{\pi Max} - Z^2_{\pi Min}}, \frac{Z^3_{\pi CP} - Z^3_{\pi Max}}{-(Z^3_{\pi Max} - Z^3_{\pi Min})}\right) \quad (4.1)$$

Fitting an L_q function means finding the q value in Equation (4.1). To find the q value, first we need to compute the values of $d_{\pi CP},\ r_{\pi CP}$ and $i_{\pi CP}$. The distance and radar threat are two conflicting objectives (see Section 3.1.3). Hence, the Pareto-optimal point of tour π with the minimum distance value (MDP) has the maximum radar detection threat value, and the extreme Pareto-optimal of tour π with the minimum radar threat value (MRP) has the maximum distance value. Let $Z^k_{\pi MDP}$ and $Z^k_{\pi MRP}$ denote the performances of MDP and MRP of tour π on k^{th} objective, respectively. Then we have $(Z^1_{\pi Min}, Z^1_{\pi Max}) = (Z^1_{\pi MDP},\ Z^1_{\pi MRP})$, and $(Z^2_{\pi Min}, Z^2_{\pi Max}) = (Z^2_{\pi MRP}, Z^2_{\pi MDP})$ for the (*min*, *max*) values of (1) distance and (2) radar detection threat objectives, respectively. $Z^k_{\pi t}$ for $k = 1, 2$ and $t = MDP, MRP$ can be calculated using Equation (3.4). In other words, the distance and radar

detection threat values of MDP of tour π is calculated summing up the distance and radar detection threat values of the trajectories with the shortest distances between consecutive targets visited in tour π. On the other hand, the distance and radar detection threat values of MRP of tour π is calculated by summing up the distance and radar detection threat values of the trajectories with the largest distances between consecutive targets visited in that tour.

However, finding $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$, for the third objective function (information collection), is not straightforward as it is for the other two objectives. An option to find the maximum and minimum collected information for tour π, is solving mathematical models that determine the trajectories to be used between consecutively visited targets in tour π to maximize and minimize the total collected information, respectively. However, using exact models are computationally demanding as the models need to be run for each population member at each generation. As a solution approach, we develop a heuristic to approximate $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$.

4.2.1.2 Heuristic Approximating $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$ for Tour π

We have the approximated Pareto-optimal frontiers of the movements between target pairs in terms of distance and radar detection threat. Recall that, z^k_{ijt} represents the *k*th objective value of the extreme point *t* of the Pareto-optimal frontier of target pair (i, j), where $t = MDP, MRP$ and $k = 1, 2$, where $k = 1, 2$ correspond to (1) distance and (2) radar detection threat objectives. The extreme points MDP and MRP are the Pareto-optimal trajectories with the shortest and longest distances, respectively (see Figure 3.2 and Figure 3.3).

Let N_π be the set of visited targets (excluding the base h) in tour π, t_j be arrival time to target j, $I_j(t_j)$ be the available information in target j at arrival time t_j, and j^- be the preceding target of target j in tour π. We assume that at time 0 the UAV is at the

base, i.e. $t_h = 0$, and the UAV flies with a constant speed, v. The following heuristic approximates $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$ for tour π.

Step 0. Set $t_h^{min} = t_h^{max} = 0$.

Step 1. Find t_j^{min} and t_j^{max} for all $j \in N_\pi$, where t_j^{min} and t_j^{max} are the earliest and latest possible arrival times at target j, respectively.

$$t_j^{min} = t_{j^-}^{min} + \frac{Z^1_{j^- jMDP}}{v} \tag{4.2}$$

$$t_j^{max} = t_{j^-}^{max} + \frac{Z^1_{j^- jMRP}}{v} \tag{4.3}$$

Step 2. Find I_j^{min} and I_j^{max} for all $j \in N_\pi$, where I_j^{min} and I_j^{max} are the minimum and maximum possible information collection values from target j, respectively.

$$I_j^{min} = \min_{t_j^{min} \le t_j \le t_j^{max}} I_j(t_j) \tag{4.4}$$

$$I_j^{max} = \max_{t_j^{min} \le t_j \le t_j^{max}} I_j(t_j) \tag{4.5}$$

Step 3. Find $Z^3_{\pi Min}$ and $Z^3_{\pi Max}$:

$$Z^3_{\pi Min} = \sum_{j \in N_\pi} I_j^{min} \tag{4.6}$$

$$Z^3_{\pi Max} = \sum_{j \in N_\pi} I_j^{max} \tag{4.7}$$

In Step 1, for a given tour, we find the earliest and latest possible arrival times to each visited target. To find them, we use the minimum and maximum possible flight durations between consecutively visited target pairs. Note that the UAV is at the base target at time 0. In Step 2, for each target, we find the smallest and largest available information between the earliest and the latest arrival times to the target. For example, consider the example information structure given for a target in Figure 4.1. Assume that the earliest and latest possible arrival times to this target are 8 am and

4 pm, respectively. Then lowest and highest information levels are 1 and 4 in this interval. After finding these values for each target visited in the tour, we sum them in Step 3 to find the minimum and maximum values for the information collected for that tour. Consequently, this heuristic approximates the minimum and maximum values of the collected information objective for a given tour. With this approximation, we may end up with less tight bounds but obtain solutions in much shorter durations.

4.2.1.3 Finding a Central Point with Three Objectives

Finally, a central point (CP) is required to fit the L_q function in Equation (4.1). We find a CP of a tour using the approach presented in Section 3.3.2. This approach selects central trajectories on the approximated Pareto-optimal frontiers (in terms of distance and radar detection threat) of the consecutive target pairs of a tour. Then $Z^1_{\pi CP}$ and $Z^2_{\pi CP}$ are found summing up the distance and radar detection threat values of these selected trajectories. To find the collected information of the CP of tour π ($Z^3_{\pi CP}$), we first use the distance values of these central trajectories to find the arrival times to the targets using Equation (4.8). We then use the arrival times to the targets to find the collected information, and finally sum up the collected information from the visited targets using Equation (4.9).

$$t_j^{CP} = t_{j^-}^{CP} + \frac{Z^1_{j^- jCP}}{v} \tag{4.8}$$

$$Z^3_{\pi CP} = \sum_{j \in N_\pi} I_j\left(t_j^{CP}\right) \tag{4.9}$$

Once the extreme values of each objective function and the CP is found, the only unknown in Equation (4.1) is the q value, which can be found solving a nonlinear programming problem (with any objective function) satisfying Equation (4.1) as the only constraint.

4.2.1.4 Selecting *s* Representative Points on the Pareto-Optimal Frontier of Tour π with Three Objectives

Recall that in FREA_UAV, each tour is represented by *s* representative points in the population. For example, if there are *M* tours in the population, the population size becomes $N = M \cdot s$. Once the q value is found in Equation (4.1), we know the L_q curve that approximates the Pareto-optimal frontier of tour π. We use this approximated Pareto-optimal frontier to generate *s* representative points four tour π. These points become separate members of the population, implicitly representing the same tour.

We employ the following approach to generate *s* representative Pareto-optimal points for tour π. The extreme points MDP and MRP of tour π are selected as the first two representatives. Distance and radar detection threat objective values of these two points, $Z_{\pi t}^{k}$ for $k = 1, 2$ and $\mathrm{t} = \mathrm{MDP}, \mathrm{MRP}$, can be calculated using Equation (3.4). To find information collection values of these points, $Z_{\pi MDP}^{3}$ and $Z_{\pi MDP}^{3}$, we use Equations (4.10) and (4.11) as MDP and MRP of tour π are the points when the shortest and largest distances between consecutive targets in tour π are summed up, respectively.

$$Z_{\pi MDP}^{3} = \sum_{j \in N_{\pi}} I_j\left(t_j^{min}\right) \tag{4.10}$$

$$Z_{\pi MRP}^{3} = \sum_{j \in N_{\pi}} I_j\left(t_j^{max}\right) \tag{4.11}$$

We do not find the extreme Pareto-optimal point in terms of information collection as this requires solving a model that maximizes the collected information of the tour. In addition to the two points, we generate (*s-2*) random points on the approximated Pareto-optimal frontier of tour π as representative points. To generate a random point t, we do the following. We first replace CP with t in Equation (4.1). The we generate an information collection value $Z_{\pi t}^{3}$ randomly between $Z_{\pi Min}^{3}$ and $Z_{\pi Max}^{3}$. The remaining unknowns in Equation (4.1) are now $Z_{\pi t}^{1}$ and $Z_{\pi t}^{2}$. To find them, we

employ a similar approach that we employed in Section 3.3.2.3. We first find the nadir point of the tour π considering only the distance and radar detection threat objectives. This point takes its first objective value from MRP ($Z^1_{\pi MRP}$) and the second objective value from MDP ($Z^2_{\pi MDP}$). We then project the nadir point onto the L_q function using a randomly generated angle between 0 and 90. Distance and radar detection threat values of the projected point are set to $Z^1_{\pi t}$ and $Z^2_{\pi t}$, respectively.

To conclude, as in the implementation of FREA_UAV on the bi-objective case, here we first approximate the Pareto-optimal frontier of the movement between each target pair before executing the algorithm, and use these as input to FREA_UAV. During FREA_UAV, each population member corresponds to a tour. As each target pair in a tour has a continuous Pareto-optimal frontier, the tour has a continuous Pareto-optimal frontier, where each point offers different trade-offs among the three objectives. Hence, each tour is represented with multiple points selected from the Pareto-optimal frontier of the tour.

To approximate the Pareto-optimal frontier of a tour, we use L_q functions, and to represent the tour in the population, we select *s* representative points. Consequently, the population has *M* tours, each having *s* representative points. This results in *N*= *M.s* solutions in the population. Whenever a new tour is generated during the generations of FREA_UAV, we use the same approach to generate *s* points for the new tour. All the rest of the implementation of FREA_UAV for the three-objective UAV route planning problem described in this chapter is the same as those presented for the bi-objective case in the previous chapter.

4.2.2 Finding Trajectories of the Nondominated Solutions of Three-Objective Problem

Upon termination of FREA_UAV, we find a set of relatively nondominated solutions close to the reference points of the RP. Each chromosome in the final set contains the tour information, i.e. the order of visit to the targets, (π), and the values of the

three objective functions (distance, D_π, radar detection threat, RDT_π, and information collection, I_π). Although we know the total values of the three objectives for a tour, we do not know which combination of the trajectories leads to those objective function values.

To find the trajectories to use between consecutive target pairs in tour π, we use a similar approach as in the bi-objective case and solve a modified version of the nonlinear mathematical model developed by Tezcaner Öztürk and Köksalan (2018). This model selects approximately efficient trajectories to be used between consecutive target pairs in a tour, π, making sure that the summation of the objective function values of the selected trajectories are as close as possible to the objective function values (D_π, RDT_π, and I_π) of the solution of FREA_UAV. The model selects one point on the approximated Pareto-optimal frontier of each consecutive target pair. This makes the model nonlinear, as Type 2 and 3 frontiers are approximated by L_q functions. According to the distance values of the selected trajectories, the model calculates the arrival times to targets and determines the information collected from the targets. This nonlinear model is presented in Appendix B.2. As the trajectories are selected from the approximated Pareto-optimal frontiers by the model, they are still approximations. After finding the objective values of the trajectories to be used between each consecutive target pair with the nonlinear model, we find the real efficient trajectory between each consecutive target pair using the approach developed by Tezcaner Öztürk and Köksalan (2018). This approach finds the efficient trajectory between a target pair with a distance value using Golden Section Search. In our application, we use their approach to find the actual radar detection threat values of the trajectories between consecutively visited targets.

4.3 Results of the Three-Objective UAV Route Planning Problem

We experiment on a 5-target UAV route planning problem using R (R Core Team, 2021) on a PC with 6-core Intel Core i7-4770S CPU, 3.10Ghz, 16 GB RAM. We set population size, M = 20, number of representative points, s = 10, resulting in a parent

population of size $M.s$ =200. All population members participate in a tournament twice, hence 200 tournaments are realized with 200 winners. During crossover, c=2 random target positions directly pass from the parents. If both of the parents of an offspring are the same tour, mutation is applied with probability $pmut$=1.0. Otherwise, mutation is applied with probability $pmut$=0.1. After each crossover and mutation, two offspring are generated. For each offspring tour, s=10 representative points are selected. Therefore, to produce 200 offspring, $M/2$=10 crossover and mutation operations are applied to the winners. The algorithm terminates after 100 generations.

We assume that the UAV flies with a constant speed of 100 km/hour. The terrains we considered for demonstrating the bi-objective problem is too small for a UAV flying at such a speed. Therefore, we enlarged the terrain to 700x700 = 490,000 km^2, which is about 0.62 times the surface area of Turkey. We set the radius of the inner and outer radar regions to 30 and 70 km, respectively. The placement of radars and targets on the new terrain is given in Figure 4.2.

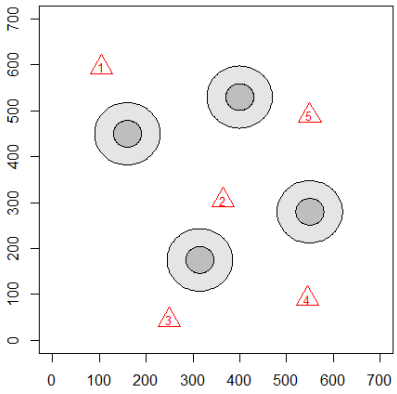


Figure 4.2: The terrain of the 3-objective 5-target UAV route planning problem

4.3.1 DM's Reference Point: *(D=1800, RDT=50, I=15)*

We first run the algorithm for a reference point which is nondominated with respect to the Pareto-optimal frontier of the problem. As can be seen in Figure 4.3, the algorithm converges to Pareto-optimal points that are close to the reference point.

FREA_UAV results in the tour 1-5-3-4-2-1 with *D*= 1982.769, *RDT* = 77.586, and *I* = 12.670 as the closest observation to the reference point (*D*=1800, *RDT*=50, *I*=15). We also present the archive population that approximates the Pareto-optimal frontier. The solution duration is 59.12 seconds.

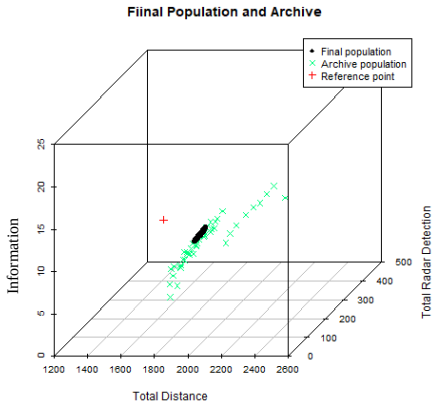


Figure 4.3: Results of the 3-objective 5-Target UAV route planning problem for the reference point *(D=1800, RDT=50, I=15)*

Using the tour, *D, RDT, and I* information obtained with FREA_UAV, we find the actual trajectories to be used between each consecutive target pair using the method explained in 4.2.2. The resulting distance (*d*), radar detection threat (*rdt*), and information collection (*i*) values between target pairs are given in Table 4.1.

Table 4.1: Trajectories between target pairs of the closest solutions to reference point *(D=1800, RDT=50, I=15)*

Reference Point: (1800, 50, 15) FREA_UAV: (1982.769, 77.586, 12.670)		**1→5**	**5→3**	**3→4**	**4→2**	**2→1**	**Total**
Trajectory Information	***d***	477.695	537.128	298.410	280.400	389.525	1983.158
	rdt	0.047	74.129	0.000	0.000	3.800	77.975
	i	1.583	2.463	4.283	3.952	0.000	12.281

4.3.2 DM's Reference Point: *(D=2200, RDT=200, I=5)*

In this demonstration, we assume that the DM provides a reference point which is dominated by the Pareto-optimal frontier of the problem. As can be seen in Figure

4.4, the algorithm converges to Pareto-optimal points that are close to the reference point. FREA_UAV results in the tour 1-5-4-3-2-1 with *D*= 1831.800, *RDT* = 197.155, and *I* = 8.023 as the closest observation to the reference point (*D*=2200, *RDT*=200, *I*=5). The solution duration is 62.21 seconds.

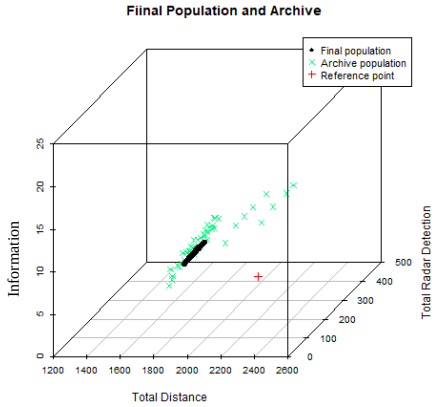


Figure 4.4: Results of the 3-objective 5-Target UAV route planning problem for the reference point (D=2200, RDT=200, I=5)

Using the tour, *D, RDT, and I* information obtained with FREA_UAV, we find the actual trajectories to be used between each consecutive target pair using the method explained in 4.2.2. The resulting distance (*d*), radar detection threat (*rdt*), and information collection (*i*) values between target pairs are given in Table 4.2.

Table 4.2: Trajectories between target pairs of the closest solution to the reference point *(D=2200, RDT=200, I=5)*

Reference Point: (2200, 200, 5) FREA_UAV: (1831.800, 197.155, 8.023)		**1→5**	**5→4**	**4→3**	**3→2**	**2→1**	**Total**
Trajectory Information	***d***	461.056	404.398	298.410	284.322	389.607	1837.793
	rdt	53.287	55.616	0.000	92.497	1.749	203.149
	i	1.458	2.327	2.090	2.861	0.000	8.737

4.3.3 Updating Reference Points

In this experiment, we let the DM update his/her reference points during the search. We start with an arbitrary reference point (*D*=0, *RDT*= 0, *I*=0) and run the algorithm for 5 generations. We show the spread of solutions in the objective space to the DM

(Figure 4.5(a)). The DM selects (D=2100, RDT=150, I=11) as his/her initial reference point. We run the algorithm for 20 more generations and the reference point turns out to be dominated Figure 4.5(b)). The DM provides a new reference point (D=1900, RDT=130, I=13) and we run the algorithm for 20 more generations (Figure 4.5(c)). After observing the spread of solutions in the archive, the DM does not change his/her reference point. We terminate the algorithm after 100 more generations (Figure 4.5(d)). The total solution time is approximately 350 seconds. We then present the final and archive populations to the DM. To demonstrate an extreme case, we assume that the DM wants to discover another region of the objective space and sets his/her new reference point as (D=1700, RDT=160, I=8). We run the algorithm for 20 more generations, and he/she does not change his/her final reference point (Figure 4.5(e)). To finalize the algorithm, we run 20 more generations. Our algorithm quickly converges to the new region as shown in Figure 4.5(f).

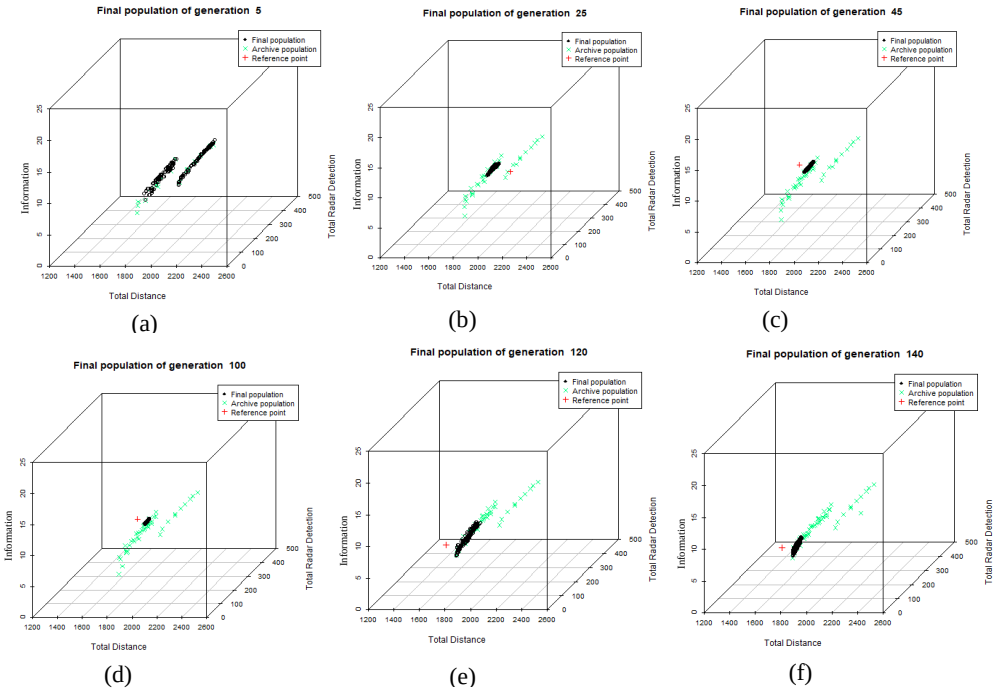


Figure 4.5: Changing reference points: 3-Objective 5-Target UAV route planning problem

4.4 Discussions for Chapter 4

In this chapter, we extend the bi-objective UAV routing problem we study in the previous chapter by introducing a third objective. Now route plans are prepared considering three objectives: (1) minimization of the length of the tour, (2) minimization of the radar detection threat of the tour, and (3) maximization of the information gain.

Similar to the bi-objective UAV route planning problem, the UAV starts its movement from a base, visits all targets in the area of operation and returns to the base. The aim is to determine both the visiting order of the targets and the specific trajectories to use between consecutive target pairs. We consider continuous terrain by allowing infinitely many trajectory options between target pairs. To make the maximization of the collected information a reasonable objective function for a mission where all targets are visited, we consider time-dependent prizes. The available information at a target is time-dependent and the value of the information that can be collected varies throughout a day.

Continuous terrain and three-objectives lead to infinitely many trajectory options. We adapt FREA_UAV to account for the additional objective. We demonstrate the modified FREA_UAV on a mission with 5-targets. The results show that our modified algorithm is able to handle the additional objective and can be used to find the preferred Pareto-optimal solutions of the route planner. Moreover, similar to the results of bi-objective problem, we observe that even if the route planner changes his/her reference points to explore completely different regions of the Pareto-optimal frontier, the algorithm can quickly converge to new regions. This work contributes to the literature of UAV route planning problem considering continuous terrain, three-objectives, and time-dependent prizes.

Future extension directions of this study are similar to those of the previous chapter. Additional practical issues such as ground obstacles and flight dynamics can be considered to enrich the UAV route planning problem. The UAV may be allowed to

loiter in the mission terrain in order to take advantage of higher information collection by arriving at a target later. The problem can be studied with multiple UAVs. FREA can be adapted for different three-objective optimization problems. Different genetic operators and parameters can be used in FREA to improve its performance. An interactive version of FREA can be developed to elicit the preference information of the DM through his/her comparisons among objectives and/or solutions progressively throughout the algorithm.

CHAPTER 5

UAV ROUTE PLANNING AS A MULTI-OBJECTIVE ORIENTEERING PROBLEM

In this chapter, we address a variant of the multi-objective UAV route planning problems studied in the previous chapters. In the previous two studies, all targets in the mission terrain must be visited by the UAV. Hence, the problems are modeled as TSP in the broader sense, and solution approaches are developed to find efficient route plans that involves the order of visits to the targets, and the trajectories used between consecutively visited target pairs. Unlike the previous two problems, in this chapter, we present a problem where visiting all targets in the mission fields is not necessary. Three objectives (1) minimization of the length of the tour, (2) minimization of the radar detection threat of the tour, and (3) maximization of the collected information from the mission terrain, are considered simultaneously, and subset of targets may be visited in a mission in order to gain advantage in terms of distance and radar detection threat objectives. In addition, now the UAV finishes its route at a different base than the initial one. The UAV takes off from a home base, visits a subset of targets, and finishes its movement at a final base. The structure of the problem is similar to the Orienteering Problem (OP) framework. In a typical OP, the objective is to create a path along a network that is restricted by a travel cost to maximize the total prize collection (Golden et al., 1987). However, our problem has important distinctions from the traditional OP.

First, we consider three-objectives simultaneously. A route planner usually wishes to explore the trade-offs among these three objectives elements to decide on the best route to take. Some combinations of these objectives have been considered for UAV routing by other studies as well. For example, the collected information is maximized by Moskal and Batta (2017) and Xia et al. (2017), and mission duration and mission

safety are considered by Pfeiffer et al. (2009) and Tezcaner Öztürk and Köksalan (2018).

Second, we consider time-dependent prizes. In reconnaissance mission planning, the amount of information that can be collected from a target often depends on the time the target is visited. This may be due to, for example, the weather or light conditions at the target's location, or the prior knowledge about the time that assets of interest from an information gathering perspective will likely be moved into or away from the target area.

Third, we consider multiple trajectory options between two points in the mission terrain. UAVs operate on a continuous terrain in practice. When the flight duration is the only consideration, a UAV will take the shortest path. On the other hand, if radar detection threat is the main consideration, a UAV will follow the shortest path that avoids threat regions. Tezcaner Öztürk and Köksalan (2018) show that there are infinitely many meaningful trajectories considering both objectives. In Chapters 3 and 4, we consider the movement terrain of the UAV as continuous terrain. In this chapter, we use a representative subset of meaningful trajectories to create a practical optimization framework.

Our problem can be classified as an OP with multiple objectives, time dependent prizes, and multiple connections. We develop a mixed integer programming (MIP) model and generate an approximate Pareto-optimal frontier. Our experiments show that the exact solution approach is computationally demanding for large problem instances. We thus develop a hybrid algorithm to decrease the computational requirements. This algorithm involves heuristics that approximate the optimal solution of the information-maximizing problem and another MIP model that provides an upper bound for the maximum information. Our computational experiments show that the hybrid algorithm substantially improves the solution times. We approximate the Pareto-optimal frontiers of the problem instances using the hybrid algorithm. We also conduct a case study experimenting on two new problem instances generated based on the terrain properties of the Colorado state of

the U.S. Finally, we provide a discussion on some practical issues related with UAV routing.

This study contributes to both the OP and the UAV route planning literature as follows:

1. We introduce the OP with multiple objectives, time-dependent prizes, and multiple connections to the literature. The vast majority of the literature study OP considering a single objective, a single connection, and fixed prizes.
2. We consider a UAV prize collection problem in a radar-monitored terrain and represent the continuous terrain with multiple trajectories. UAV prize collection problem is generally studied in non-monitored terrains with single trajectories between targets in the literature.
3. We develop an MIP model that can be used to solve small-sized problem instances to optimality. We also develop a hybrid solution algorithm involving heuristics and exact approaches to solve larger problem instances.

The rest of this chapter is organized as follows: In Section 5.1, we review the relevant literature of orienteering problem. In Section 5.2, we explain the specifications of the problem, and provide a demonstration of the problem on a small problem instance. In Section 5.3, MIP formulations are developed, and in Section 5.4, Hybrid Algorithm is presented. In Section 5.5, computational experiments and their results are reported. In Section 5.6, a case study is presented. In Section 5.7 the chapter is concluded and future research directions are provided.

5.1 Orienteering Problem

As we model the UAV route planning problem studied in this chapter as an OP, we review the relevant literature in the following. We refer readers to Chapter 2 for the literature of UAV route planning problem with prize collection.

The OP originates from orienteering which is the sport of racing and navigation played in unfamiliar terrains, and has been addressed as an optimization problem by several researchers (see, for example, Tsiligirides, 1984; Golden et al., 1987; Chao et al., 1996). Given a set of nodes with prizes and given travel costs between nodes, total prize is maximized subject to a constraint on the travel cost.

There are three other problems that are very similar to OP: (a) Maximum Collection Problem (MCP) (Kataoka and Morito, 1988) or Selective Traveling Salesperson Problem (Selective TSP) (Laporte and Martello, 1990), (b) Prize-Collecting Traveling Salesperson Problem (PCTSP) (Balas, 1989) and (c) Profitable Tour Problem (PTP) (Dell'Amico et al., 1995). Feillet et al. (2005) define a generic name for these problems as "Traveling Salesperson Problems with Profits (TSP with Profits)" and provide a comprehensive survey. Any of these problems can be converted to each other or to OP easily. Therefore, we consider the relevant literature of these problems as well.

OP is NP-hard (Golden et al., 1987) and few studies consider exact solution approaches (see, for example, Laporte and Martello, 1990; Leifer and Rosenwein, 1994). Many authors focus on developing heuristic solution approaches (see, for example, Tsiligirides, 1984; Golden et al., 1987; Chao et al., 1996). In general, studies assume a single objective function, fixed prizes, and single arc option between node pairs. We refer readers to Vansteenwegen et al. (2011) and Gunawan et al. (2016) for surveys on OP.

OP is rarely considered with multiple objectives. Schilde et al. (2009) develop heuristic solution approaches for a bi-objective Selective TSP to design tourist routes. Bérubé et al. (2009) develop exact and heuristic approaches for the bi-objective combinatorial problems with integer objective values, and implement the approaches on the bi-objective PCTSP. Mei et al. (2016) consider an OP with a time-dependent travel time and a single trajectory between each node pair. They allow for multiple prize categories at nodes and develop meta-heuristics to maximize a combination of prizes collected from different categories.

OP literature that consider time dependency is also quite scarce. Erkut and Zhang (1996) address MCP considering linearly decreasing rewards and ignoring the travel cost restriction. Their MIP formulation schedules visits to all nodes even if the prize is zero. Ekici et al. (2009) and Ekici and Retharekar (2013) study MCP with multiple agents and linearly decreasing rewards. They search for tours that maximize the total reward minus travel costs. Blum et al. (2007) study a PCTSP where the rewards decrease by a factor at each time unit. These studies combine prizes and costs into a single measure and employ a simplifying prize structure that linearly reduces with time. Recently, Yu et al. (2019) consider time-dependent prizes for OP. Peng et al. (2019) and Peng et al. (2020) address a satellite scheduling problem as an OP with time-dependent prizes with heuristic and exact approaches, respectively.

A concept similar to time-dependent prizes is considered in Vehicle Routing Problem with Time Windows (VRPTW) (Braysy and Gendreau, 2005). In VRPTW, the prize of a node can only be collected in a given time interval. This concept has also been used for the OP as OP with Time Windows (OPTW) (Vansteenwegen et al., 2009). The strict time window constraint is relaxed in VRP with Soft Time Windows (VRPSTW), where targets can be visited any time but a penalty is introduced if the arrival is not in the given time window (Figliozzi, 2010). We benefit from the VRPSTW literature while developing our mathematical model.

5.2 Problem Context

In this section, we explain the structure of the problem.

5.2.1 Problem Definition and Terrain Structure

In this problem, the UAV is tasked to collect information from predetermined targets in an enemy terrain. An example terrain structure where two bases, 13 targets, and four radar zones are located in a 250,000 km^2 area is illustrated in Figure 5.1(a). The radars are effective in two circular regions as in the previous two sections and in

(Tezcaner Öztürk and Köksalan, 2018). The UAV is exposed to an additional radar detection threat at the target points.

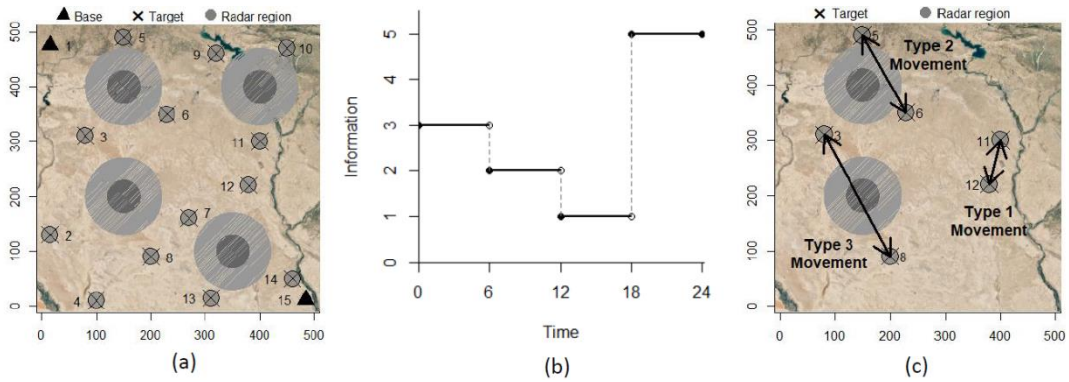


Figure 5.1: Problem context: (a) Radars and targets, (b) Information vs visit time, (c) Movement types

A route plan involves the decisions for which targets to visit, the order of visit to the selected targets, and the trajectories to follow between targets. The UAV starts from the home base (node 1) and ends at the destination base (node 15). The route planners aim to maximize the collected information while minimizing the flight duration and the radar detection threat. A subset of the targets can be visited as the UAV moves from its home base to its destination base, and different routes may be advantageous in terms of one of the three objectives.

5.2.2 Time-dependency in Collected Information

We assume that the UAV has a limited flight duration in each mission, flies at a constant speed, and each of its targets has several time periods, each providing a different level of information availability. Figure 5.1(b) demonstrates an example where the UAV can fly up to 24 hours, and each of the four time periods corresponds to a different level of information availability, where larger is better.

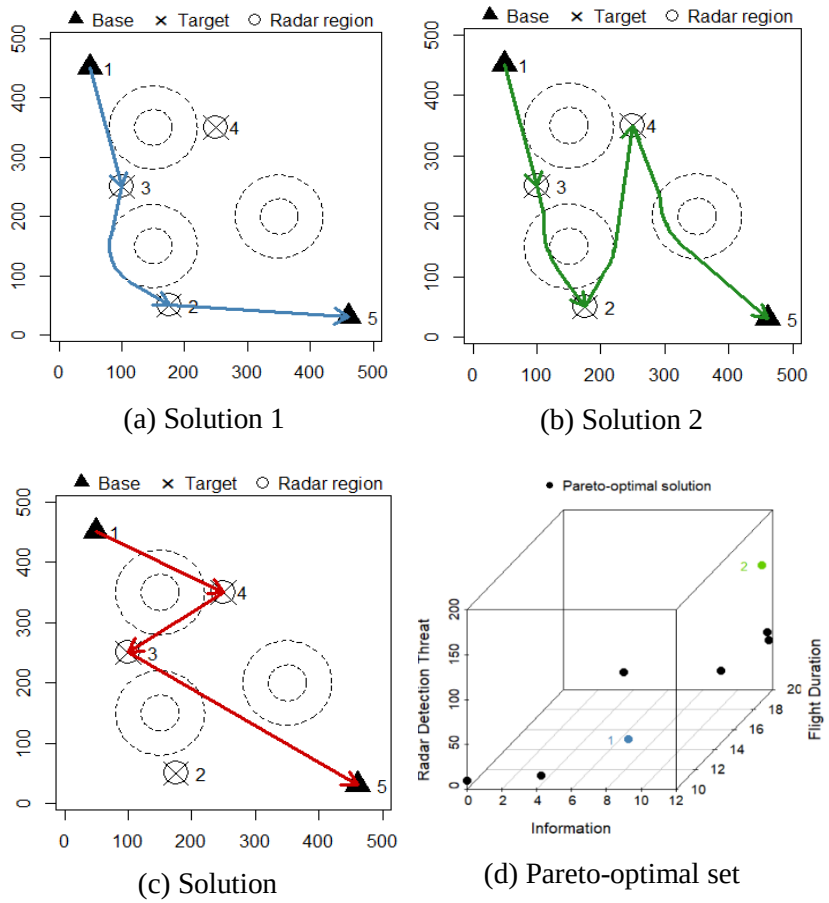
5.2.3 Movement Between Target Pairs

Similar to the first two problems studied in Chapters 3 and 4, we consider three types of movements depending on whether or not the shortest-distance between two targets passes through a radar region. We illustrated an example for each of the three movement types in Figure 5.1(c). The structures of the efficient frontiers of these movement types are illustrated previously in Figure 3.2 and Figure 3.3.

In this study, rather than using the whole efficient frontier, we select a representative subset of efficient points on the frontier. We approximate the curved Pareto frontier of Type 2 movement with three trajectories, MDP, CP and MRP (see Figure 3.2), and the more distinct Pareto frontier of Type 3 movement with four trajectories, MDP, TP, CP, and MRP (see Figure 3.3). As in Tezcaner and Köksalan (2011) and previous chapters of this paper, we calculate the radar detection threat inside a radar region using the measure developed by Gudaitis (1994) (see Appendix A).

5.2.4 Demonstration on a Small Example

To demonstrate we consider a small problem that has two bases and three targets, located in a terrain that is monitored by three radars. Let $I,\ FD,\ and\ RDT$ represent the total information collected, the total flight duration, and the total radar detection threat that the UAV is exposed to, respectively. In Solution 1 (Figure 5.2(a)), $I = 8$, $FD = 12.31$, $RDT = 30$, and in Solution 2 (Figure 5.2(b)), $I = 12$, $FD = 18.88$, $RDT = 150.6$. Solution 2 outperforms Solution 1 in terms of only I, and Solution 1 is better in terms of the other two objectives. Both solutions are in the Pareto-optimal set and it is not possible to choose one over the other without further preference information from the route planner. Solution 3 (Figure 5.2(c)), on the other hand, has objective function values $I = 4,\ FD = 13.76, RDT = 87.97$, and this solution is dominated by Solution 1. Figure 5.2(d) shows the Pareto-optimal set.



(a) Solution 1 (b) Solution 2

(c) Solution (d) Pareto-optimal set

Figure 5.2: Several routes and the Pareto-optimal set of an example problem

5.3 MIP Formulation

Let the routing problem be defined on a complete graph $G = (N, A)$ with the node set $N = \{1,2, \dots, n\}$ and the trajectory set $A = \{(i,j)^k | \, i \in N \backslash \{n\}, j \in N \backslash \{1\}, i \neq j, k \in K_{ij}\}$. Using the notation in Table 5.1, we formulate the problem with a multi-objective MIP model (Model O). The model selects the trajectories the UAV follows, determines arrival times to the targets, and finds the amount of information collected from targets.

Table 5.1: Mathematical notation

Parameters	
$G = (N, A)$	Graph with node set N and trajectory set A.
N	Node set, $N = \{1,2, \dots, n\}$.
n	Destination base and number of nodes.
A	Trajectory set $A = \{(i,j)^k \mid i \in N\backslash\{n\}, j \in N\backslash\{1\}, i \neq j,\ k \in K_{ij}\}$.
K_{ij}	Index set of the trajectories between nodes i and j, $i \neq j$, $i \neq n$, $j \neq 1$.
$(i,j)^k$	Trajectory k connecting node i to node j.
t_{ij}^k	Travel duration of $(i,j)^k \in A$.
r_{ij}^k	Radar detection threat on $(i,j)^k \in A$.
r_j	Radar detection threat at target $j \in N\ \{1,n\}$.
P	Set of time periods, $P = \{1,2, \dots, l\}$.
s^p	Starting time of the time period $p \in P$.
e^l	Ending time of the time period l.
g_j^p	Available information at target $j \in N\ \{1,n\}$ in time period p.
C	The maximum time the UAV can operate.
T	Time restriction for the overall route.
R	Radar detection threat restriction for the overall route.
MG_j	Sufficiently large scalar for target $j \in N\ \{1,n\}$.
MT, MY	Sufficiently large scalars.
α	Augmentation parameter.
Objectives	
I	Total information collected.
FD	Total flight duration of the route.
RDT	Total radar detection threat that the UAV is exposed on the route.
Decision Variables	
t_j	Arrival time at node $j \in N\backslash\{1\}$.
I_j	Information gain from target $j \in N\backslash\{1,n\}$.
x_{ij}^k	$= 1$, if $(i,j)^k \in A$ is used; 0, otherwise.
y_j^p	$= 1$, if t_j belongs to the time period p; 0, otherwise.

Model O:

$$\max I = \sum_{j \in N\backslash\{1,n\}} I_j \tag{5.1}$$

$$\min FD = \sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(t_{ij}^k\, x_{ij}^k\right) \tag{5.2}$$

$$\min RDT = \sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(r_{ij}^k x_{ij}^k\right) + \sum_{j \in N\backslash\{1,n\}} r_j \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k \tag{5.3}$$

s.t.

$$t_j \leq t_i + \sum_{k \in K_{ij}} x_{ij}^k\, t_{ij}^k + MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \tag{5.4}$$

$$t_j \geq t_i + \sum_{k \in K_{ij}} x_{ij}^k \, t_{ij}^k - MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.5)$$

$$t_j \leq MT \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k \qquad \forall j \in N\backslash\{1, n\} \qquad (5.6)$$

$$t_1 = 0 \qquad (5.7)$$

$$s^p y_j^p \leq t_j < s^{p+1} y_j^p + MY\left(1 - y_j^p\right) \qquad \forall j \in N\backslash\{1, n\}, \forall p \in P\backslash\{l\} \qquad (5.8)$$

$$s^l y_j^l \leq t_j < e^l y_j^l + MY\left(1 - y_j^l\right) \qquad \forall j \in N\backslash\{1, n\} \qquad (5.9)$$

$$\sum_{p \in P} y_p^j = 1 \qquad \forall j \in N\backslash\{1, n\} \qquad (5.10)$$

$$I_j \leq MG_j \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k \qquad \forall j \in N\backslash\{1, n\} \qquad (5.11)$$

$$I_j \leq \sum_{p \in P} g_j^p y_j^p \qquad \forall j \in N\backslash\{1, n\} \qquad (5.12)$$

$$I_j \geq \sum_{p \in P} g_j^p y_j^p - MG_j \left(1 - \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall j \in N\backslash\{1, n\} \qquad (5.13)$$

$$\sum_{j \in N\backslash\{1\}} \sum_{k \in K_{1j}} x_{1j}^k = 1 \qquad (5.14)$$

$$\sum_{i \in N\backslash\{n\}} \sum_{k \in K_{in}} x_{in}^k = 1 \qquad (5.15)$$

$$\sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k - \sum_{\substack{i \in N\backslash\{1\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ji}^k = 0 \qquad \forall j \in N\backslash\{1, n\} \qquad (5.16)$$

$$x_{ij}^k \in \{0,1\} \qquad \forall i \in N\backslash\{n\}, \forall j \in N\backslash\{1\}\ i \neq j, k \in K_{ij} \qquad (5.17)$$

$$y_j^p \in \{0,1\} \qquad \forall j \in N\backslash\{1, n\}, \forall p \in P \qquad (5.18)$$

$$0 \leq I_j \leq \max_{p \in P}\left(g_j^p\right) \qquad \forall j \in N\backslash\{1, n\} \qquad (5.19)$$

$$0 \leq t_j \leq C \qquad \forall j \in N\backslash\{1\} \qquad (5.20)$$

Objective functions (5.1) - (5.3) correspond to the three objectives of maximizing total information, minimizing total flight duration, and minimizing total radar detection threat, respectively. Constraints (5.4) - (5.7) together represent the arrival times to the targets. Note that the UAV does not loiter in the mission terrain to delay its arrival to the nodes (to obtain higher rewards) considering the unexpected threats in enemy terrains that can endanger the safety of the UAV. Constraints (5.8) - (5.10)

determine the time periods at which the UAV arrives at targets. The scalars MY and MT can be set to C. Constraints (5.11) - (5.13) capture the information gains for each of the targets. MG_j can be set to $\max_{p \in P}(g_j^p)$ for each target $j \in N\backslash\{1, n\}$. Constraints (5.14) - (5.16) are flow balance equations and (5.17) - (5.20) specify the decision variables.

5.3.1 Finding Efficient Routes

We use the $\varepsilon - constraint$ method (Haimes et al., 1971) treating the objectives FD and RDT of Model O as constraints, and solve Model S. We solve the model iteratively, changing the restrictions, to investigate the trade-offs and approximate the Pareto-optimal frontier.

Model S:

$$\max \sum_{j \in N\backslash\{1,n\}} I_j - \alpha \cdot (\overline{FD} + \overline{RDT}) \tag{5.21}$$

s.t.

$$FD \leq T \tag{5.22}$$

$$RDT \leq R \tag{5.23}$$

$$\sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(t_{ij}^k x_{ij}^k\right) - FD = 0 \tag{5.24}$$

$$\sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(r_{ij}^k x_{ij}^k\right) + \sum_{j \in N\backslash\{1,n\}} r_j \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k - RDT = 0 \tag{5.25}$$

Constraints (5.4) - (5.20)

The objective function (5.21)'s first term is to maximize the sum of collected information from the visited targets. The scaled total flight duration $(\overline{FD})$ and the scaled total radar detection threat $(\overline{RDT})$, both multiplied with a sufficiently small positive constant, α, are added to the objective function to break ties for solutions having the same maximum I value, in order to guarantee that the resulting solution is Pareto-optimal. Constraints (5.22) and (5.23) limit FD and RDT to T and R,

respectively. Constraints (5.24) and (5.25) compute FD and RDT, respectively. In this model, MT and MY can be set to T.

5.3.2 Properties of the Model

In this section, we discuss several properties of our model.

(I) Our formulation does not require subtour elimination constraints:

Although sub-tour elimination constraints are necessary in OP formulations (see, (Vansteenwegen et al., 2011) for a typical OP formulation), our formulation eliminates this requirement.

Proposition 5.1: The arrival time Constraints (5.4) - (5.7) together with the flow balance Constraints (5.14) - (5.16) and binary restrictions in Constraint (5.17) eliminate any subtours.

Proof of Proposition 5.1: Consider a subtour with three targets i, j, and m such that j is the successor of i, m is the successor of j, and i is the successor of m. Our formulation assigns values such that $t_j > t_i$, $t_m > t_j$, and $t_i > t_m$. Here, the implied inequality $t_j > t_m$ from the first and last terms contradicts with the second term, $t_m > t_j$.

(II) Augmentation parameter α has the following property:

Proposition 5.2: Setting $0 < \alpha < 0.5$ in the objective function of Model S guarantees finding a Pareto-optimal point that has the maximum I.

Proof of Proposition 5.2: $\sum_{j \in N \setminus \{1,n\}} I_j$ in Objective (5.21) is integer valued. If the maximum value of the augmentation $\alpha \cdot \left(\overline{FD} + \overline{RDT}\right) < 1$, then the model will not sacrifice 1 unit from $\sum_{j \in N \setminus \{1,n\}} I_j$ regardless of the $\alpha \cdot \left(\overline{FD} + \overline{RDT}\right)$. Since FD and RDT are scaled in the interval [0,1], $\alpha < 0.5$ suffice. To guarantee Pareto optimality, $\alpha > 0$ is required.

Corollary 5.1: Rounding up Objective (5.21) gives the value of I.

Proof of Corollary 5.1: Directly follows from the property that $0 < \alpha \cdot (FD + RDT) < 1$.

We next present additional discussion on several issues we encountered when formulating the model and experimenting with the GUROBI solver.

(III) Our initial arrival time constraints were nonlinear:

The exact arrival times to each target can be found using Equation (5.26). Since this constraint is nonlinear, we linearized it using Constraints (5.4) - (5.6) with the help of the Constraints (5.14) - (5.17). Equation (5.26) results in $|N| - 1$ constraints whereas with linearization, we have $2|N|^2 - 6|N| + 6$ constraints.

$$t_j = \sum_{\substack{i \in N \setminus \{1\} \\ i \neq j}} \sum_{k \in K_{ij}} (t_i + t_{ij}^k) x_{ij}^k \quad \forall\, j \in N \setminus \{1\}, i \neq j \tag{5.26}$$

We linearize arrival time constraints similar to the linearization used in the Time Dependent Vehicle Routing Problem (see, Malandraki and Daskin, 1992, for an example). Note that, in a typical VRP formulation, the objective function minimizes the total travel time. Therefore, in the main paper, Constraint (5.4) is redundant; because, the model always selects the smallest t_j satisfying Constraint (5.5). In our model, however, we need both Constraints (5.4) - (5.5) to avoid setting t_j to a time period with more information.

(IV) Alternative for the calculation of FD:

In our initial MIP formulation and experiments, we defined the FD using Equation (5.27) instead of the Constraint (5.24).

$$t_n - FD = 0 \tag{5.27}$$

However, the GUROBI solver could not return a solution in meaningful times when T and R allow more targets to be visited. This is due to the fact that the GUROBI solver cannot obtain tight bounds that enable fathoming nodes when Equation (5.27) is used.

(V) We observe slightly better performance with Constraint (5.6):

Constraint (5.6) sets t_j of the unvisited targets j to 0. It can be removed because Constraint (5.11) guarantees that the collected information is 0 for the unvisited targets. However, we observed that the GUROBI solves the model slightly faster when Constraint (5.6) is included. Our guess for this without any proof is that the solver spends extra time to assign arbitrary arrival times to the unvisited targets in Constraints (5.4) - (5.5), which requires it to assign time periods to the arrival times in Constraints (5.8) - (5.10).

5.3.3 Complexity of the Problem

Golden et al. (1987) prove that OP is NP-hard by showing that TSP is a special case of OP.

Proposition 5.3: Model S is NP-hard.

Proof of Proposition 5.3: Consider our problem where $|K_{ij}| = 1$ for each target pair i and $j, i, j \in N$, I_j for all $j \in N\backslash\{1, n\}$ is the same in all time periods, and R is large enough so that Constraint (5.23) is redundant. The problem then turns into a traditional OP where the information collected is maximized under time restriction T.

5.4 A Hybrid Algorithm

We develop Hybrid Algorithm (HA) that can approximate a solution faster than Model S, guaranteeing that the approximation is within the optimality gap route

planner desires. HA consists of two approaches: (a) The Probabilistic Route Search Heuristic (PRSH) that approximates the optimal solution (see Section 5.4.1), and (b) The MIP Model Q that provides an upper bound on the optimal solution (see Section 5.4.3). In addition to the notation presented in Table 5.1, we introduce the notation in Table 5.2 to develop HA and PRSH.

The steps of HA are as follows:

Step 1. Run PRSH: Let f_P be the PRSH's final solution. Set $Z_P = Z(f_P)$ and $LB = Z(f_P)$.

Step 2. Solve Model Q: Solve Model Q introducing f_P and LB to the solver as the initial solution and initial lower bound, respectively. Find Z_Q^* and set $UB = Z_Q^*$.

Step 3. Check Approximation Quality: Calculate gap using Equation (5.28). If $gap \leq U$, terminate the algorithm with f_P as the final solution, and $Z(f_P)$ as the final objective function value. Otherwise, go to Step 4.

$$gap = \frac{UB - LB}{LB} \tag{5.28}$$

Step 4. Solve Model *S*: Solve Model S introducing f_P and LB to the solver as the initial solution and initial lower bound, respectively. Let the solver find its own upper bound (UB_S^H), improve LB and UB_S^H, set $UB = \min(Z_Q^*, UB_S^H)$, and terminate when $gap \leq U$ is satisfied. Let f_P and $Z(f_P)$ denote the final solution and the objective function value, respectively, upon termination.

Table 5.2: Mathematical notation for HA

m	Iteration counter.
M	A user-defined parameter defining the number of routes that will be generated.
U	A user-defined parameter defining the maximum deviation from the optimal solution.
q	A parameter that augments the desirability of a trajectory.
θ	A parameter defining the relative weight of a resource with respect to the other one.
T_r	Remaining flight duration resource.
R_r	Remaining radar detection threat resource.
D_{ij}^k	Fitness value of the trajectory $(i,j)^k$.
P_{ij}^k	Selection probability of the trajectory $(i,j)^k$.
c_{ij}^k	Weighted sum of the resources consumed when the trajectory $(i,j)^k$ is used.
Z_Q^*	Optimal objective function value of Model Q.
f_m	Route m, defined by a set of visited targets and followed trajectories.
F_i	The set of feasible trajectories leaving from node i.
F_i'	The set of feasible trajectories connecting node i to the destination base (node n).
A_m	The set of trajectories in f_m (visited trajectories).
V_m	The set of targets in f_m (visited targets).
U_m	The set of targets not in f_m and feasible to be visited in f_m.
S_m	Route set m.
$Z(f_m)$	Objective function value of route f_m, calculated using Equation (5.21).
$RDT(f_m)$	Total radar detection threat on the route f_m.
$FD(f_m)$	Total flight duration of the route f_m.
UB	Upper bound on the objective function value.
LB	Lower bound on the objective function value.

5.4.1 Probabilistic Route Search Heuristic

Probabilistic Route Search Heuristic (PRSH) consists of two phases: (a) construction and (b) improvement. In the construction phase, we generate routes using a probabilistic approach similar to the stochastic algorithm developed by Tsiligirides (1984) for OP to determine the next node to be added to the path. Here, instead of nodes, we select the trajectories to be added to the path. Whenever the UAV reaches a node, a probability is assigned to each of the feasible trajectories leaving the node, based on the information gain per unit resource consumed. Then, the next trajectory to be traversed is selected probabilistically. This selection procedure is repeated until there is no feasible trajectory that can be included in the route. Many routes are generated in this fashion and the one with the highest objective function value is selected at the end as the final solution. In the second phase, we implement improvement heuristics to improve the final solution of the construction heuristic.

Suppose that the heuristic is run on a problem with node set N and trajectory set A, where 1 and n are the initial and final bases, respectively. Let route f_m be a set of sequence of visited nodes and used trajectories $\{i_1, a_1, i_2, \dots, i_{n-1}, a_{n-1}, i_n\}$ where $i_l \in N,\ a_l \in A,\ i_1 = 1,\ i_n = n,\ a_l = (i_l, i_l + 1)^k,\ k \in K_{i_l i_{l+1}}$. The objective function of a route is calculated using Equation (5.21) throughout PRSH. We next explain the steps of the construction and improvement heuristics.

5.4.1.1 Construction Heuristic

The steps of the construction heuristic are as follows:

Step 1. Initialization: Set $S_c = \emptyset, m = 1, T_r = T, R_r = R$.

Step 2. Starting a new route: If $m > M$, go to Step 7. Otherwise, set $f_m = \{1\}$, $i = 1, V_m = \{1, n\}$, and go to Step 3.

Step 3. Determining feasible trajectories: Find $F_i = \{(i,j)^k \mid (r_{ij}^k + r_j + r_{jn}^{k'}) \leq R_r,\ (t_{ij}^k + t_{jn}^{k'}) \leq T_r,\ j \in N \backslash V_m,\ k \in K_{ij},\ (j,n)^{k'} \in A,\ k' \in K_{jn}\}$. If $F_i \neq \emptyset$, go to Step 4. Otherwise, find $F_i' = \{(i,n)^k \mid r_{in}^k \leq R_r,\ t_{in}^k \leq T_r,\ k \in K_{in}\}$. If $F_i' \neq \emptyset$, set $f_m = f_m \cup \{(i,n)^k, n\}$, where $(i,n)^k$ is arbitrarily selected from F_i', $S_c = S_c \cup \{f_m\}$, $m = m + 1$, and return to Step 2. Otherwise, terminate the algorithm establishing that the problem is infeasible.

Step 4. Fitness assignment: Compute D_{ij}^k for each trajectory $(i,j)^k \in F_i$ using Equation (5.29).

$$D_{ij}^k = \left(\frac{I_j}{c_{ij}^k}\right)^q, \text{where } c_{ijk}^k = \left(\frac{r_j + r_{ij}^k}{R_r} + \theta \frac{t_{ij}^k}{T_r}\right) \tag{5.29}$$

Step 5. Probability assignment: Assign the probability P_{ij}^k to each trajectory $(i,j)^k \in F_i$ using Equation (5.30):

$$P_{ij}^{k} = \frac{D_{ij}^{k}}{\sum_{(i,j)^k \in F_i} D_{ij}^{k}} \tag{5.30}$$

Step 6. Selection and updating: Generate a random number between $0-1$ to select $(i,j)^k \in F_i$ to traverse, using the probabilities computed in Step 5. Set $f_m = f_m \cup \{(i,j)^k, j\}$, $V_m = V_m \cup \{j\}$, $T_r = T_r - t_{ij}^k$, $R_r = R_r - \left(r_{ij}^k + r_j\right), i = j,$ and return to Step 3.

Step 7. Termination: Terminate the heuristic with the route $f_c = \underset{f_m \in S_c}{\operatorname{argmax}}\{Z(f_m)\}$.

We construct at most M routes in the construction heuristic, each route initiating at base node 1 (Step 2). Each time we execute Steps 3-6, we add a new trajectory and a new node to the route while maintaining feasibility. We check feasibility when we construct set F_i for the last visited node i in Step 3. If there are no feasible trajectories from node i to an intermediate node j, then we arbitrarily select a feasible trajectory connecting node i to node n. If $i \neq 1$, then there is always a feasible trajectory connecting node i to node n by the design of F_i. However, if $i = 1$, and there is no feasible trajectory connecting node i to node n, then the problem is terminated as infeasible. We use $q \in \{1,4\}$ and $\theta \in \{0.5, 1, 1.5\}$ to increase the diversity of the generated routes in Step 4. We generate M routes for each (q, θ) combination and select the best route in our computational experiments.

5.4.1.2 Trajectory Exchange Improvement

Let route f_c be the final route of the construction heuristic. We next explain the trajectory exchange algorithm and demonstrate an example exchange.

Step 1. New route generation: Set $A_c = \{(i,j)^k \in A \mid (i,j)^k \in f_c\}$. For every trajectory $a \in A_c$, perform all possible 2-opt moves. While building new connections in a 2-opt move, consider all combinations of the trajectory options. Generate all possible routes and let these new routes constitute the set S_t.

Step 2. Feasibility check: Set $S_e = \{f_m \in S_t : FD(f_m) \leq T, RDT(f_m) \leq R\}$.

Step 3. Improvement and termination: Set $S_e = S_e \cup \{f_c\}$ and terminate the algorithm with the route $f_e = \underset{f_m \in S_e}{\operatorname{argmax}} Z(f_m)$.

In Step 1, a 2-opt move at trajectory $(i,j)^k$ removes trajectory $(i,j)^k$ and another trajectory that is not an immediate predecessor or successor $(i,j)^k$. The order of visit between the nodes in the central subgraph is reversed, keeping their trajectories. The removed trajectories are replaced with two new trajectories to reconnect the two disconnected subgraphs based on the new order. When connecting the subgraphs with new trajectories, all possible trajectory options are considered. All possible routes are generated with the combination of the trajectory alternatives of the new connections. We next illustrate Trajectory Exchange Improvement on an example.

Suppose $f_c = \{1, (1,a)^{k_1}, a, (a,b)^{k_2}, b, (b,c)^{k_3}, c, (c,d)^{k_4}, d, (d,n)^{k_5}, n\}$ (Figure 5.3(a)) and 2-opt move breaks the trajectories $(a,b)^{k_1}$ and $(c,d)^{k_4}$ (Figure 5.3(b)). We then obtain $f_c' = \{1, (1,a)^{k_1}, a, (a,c)^{?}, c, (c,b)^{?}, b, (b,d)^{?}, d, (d,n)^{k_5}, n\}$, where the trajectory between (a,c), (c,b) and (b,d) need to be decided. We use the same trajectory used between (b,c) for the new trajectory between (c,b). However, there are multiple trajectory options for the new connections (a,c) and (b,d). All possible routes are generated with the combination of the trajectories between (a,c) and (b,d). This creates $(|K_{ac}| \cdot |K_{bd}|)$ many new routes.

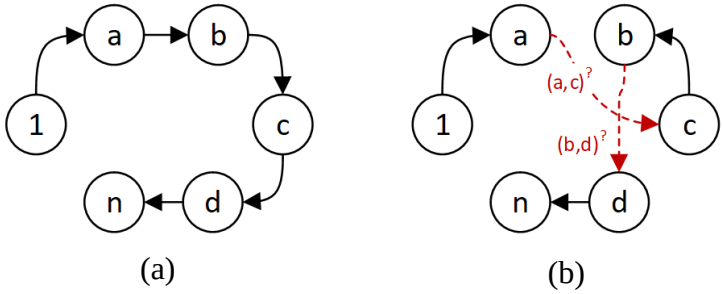


Figure 5.3: An example of the modified 2-opt improvement heuristic

This improvement algorithm runs in $O(|N|^2 \cdot (\max_{i,j \in N} |K_{ij}|)^2)$. In our problem, $|K_{ij}| \leq 4 \; \forall \; i,j \in N$ (see Section 5.2.3), and this reduces the run time of the trajectory exchange improvement to $O(|N|^2)$, which is same as the running time of 2-opt improvement used in TSP.

5.4.1.3 Node Insertion Improvement

Using the final route of the trajectory exchange improvement algorithm f_e, we next present the node insertion improvement algorithm and demonstrate it with an example.

Step 1. Initialization: Set $S_n = \{f_e\}, f_n = f_e$.

Step 2. Cadidate elimination: Set $R_r = R - RDT(f_n), U_n = i \in N: i \notin f_n, r_i \leq R_r$.

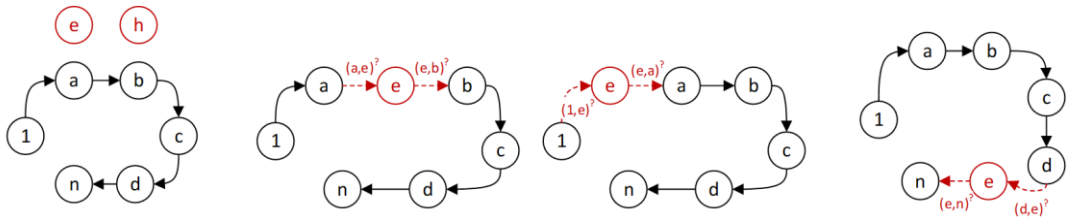
Step 3. Node insertion and new route generation: If $U_n = \emptyset$, go to Step 5. Otherwise, for each $u \in U_n$:

- **3.1.** Find node q_1 in f_n that can be reached in minimum flight duration from node u.
- **3.2.** Find node q_2 in f_n that can be reached in minimum radar detection threat from node u.
- **3.3.** For every $k \in \{1,2\}$ do following: If q_k is home base, insert node u after then q_k. If q_k is the final base, insert u before q_k. Otherwise, insert u both before and after node q_k, creating 2 different route orders.
- **3.4.** Generate all possible new routes using the combinations of the trajectory options between new connections, and add these routes to S_n.

Step 4. Feasibility check and updates: Set $S_n = \{f_m \in S_n: FD(f_m) \leq T, \; RDT(f_m) \leq R\}$ and find $f_n' = \underset{f_m \in S_n}{\operatorname{argmax}} Z(f_m)$. If $f_n' = f_n$, go to Step 5. Otherwise, set $S_n = \{f_n'\}$ and $f_n = f_n'$, and return to Step 2.

Step 5. Termination: Terminate the algorithm with the route f_n.

We next illustrate Node Insertion Improvement on an example in Figure 5.4. Let $f_n = \{1, (1,a)^{k_1}, a, (a,b)^{k_2}, b, (b,c)^{k_3}, c, (c,d)^{k_4}, d, (d,n)^{k_5}, n\}$, and the two nodes, e and h, are not visited (Figure 5.4(a)). Suppose that $r_h > R_r$, which makes $U_n = \{e\}$ in Step 2. Let q_1 be target a and q_2 be node n. In Step 3.3, target e is inserted after and before target a, which creates the routes $f_n^1 = \{1, (1,a)^{k_1}, a, (a,e)^?, e, (e,b)^?, b, (b,c)^{k_3}, c, (c,d)^{k_4}, d, (d,n)^{k_5}, n\}$ and $f_n^2 = \{1, (1,e)^?, (e,a)^?, a, (a,b)^{k_2}, b, (b,c)^{k_3}, c, (c,d)^{k_4}, d, (d,n)^{k_5}, n\}$ (Figure 5.4(b)). Then target e is inserted before n, which creates the route $f_n^3 = \{1, (1,a)^{k_1}, a, (a,b)^{k_2}, b, (b,c)^{k_3}, c, (c,d)^{k_4}, d, (d,e)^?, e, (e,n)^?, n\}$ (Figure 5.4(c)). Due to the multiple trajectories, $(|K_{ae}| \cdot |K_{eb}| + |K_{1e}| \cdot |K_{ea}| + |K_{de}| \cdot |K_{en}|)$ many new routes are generated.



(a) (b) (c)

Figure 5.4: An example for node insertion improvement

Node insertion improvement runs in $O\left((|N|-3)^2 \cdot (\max_{i,j \in N} |K_{ij}|)^2\right)$. $|N|-3$ is the maximum number of unvisited targets that can be considered for insertion in a solution. Since $|K_{ij}| \leq 4 \; \forall \; i,j \in N$, this node insertion improvement runs in $O(|N|^2)$.

5.4.2 Performance Analysis of PRSH

We demonstrate that we can navigate PRSH towards high quality solutions by selecting suitable parameter settings. We demonstrate its average performance empirically based on computational experiments. We will omit the augmentation part of Objective (5.21) as our aim is to demonstrate the performance on the *I* value.

Z_P and Z^* are the objective function values of PRSH's solution and optimal solution, respectively.

Suppose that PRSH is run on a problem instance with node set $N = \{1, \dots, n\}$. Let $h \in N$ and $(1,h)^{k'}$ be in the set of feasible trajectories leaving from the home base, i.e. $(1,h)^{k'} \in F_1$. Assume that PRSH finds the feasible solution that has the worst I value and the optimal solution is at the other extreme containing the sum of highest possible information gains of all targets. To construct the worst I value, suppose that the heuristic selects $(1,h)^{k'}$ to visit target h and $F_h = \emptyset$. Then no more targets can be visited, and PRSH terminates with a route that only visits target h and has $Z_P = I_h$, where $I_h = \min_{p \in P}(g_h^p)$ and P is the set of time periods. Assume that the optimal solution corresponds to a solution where all targets can be visited when the available information they contain is highest. In this case, the objective function value would be $Z^* = \sum_{j \in N \setminus \{1,n\}} I_j$, where $I_j = \max_{p \in P}(g_j^p)$ for all $j \in N \setminus \{1,n\}$.

Since PRSH generates M routes, and in order to find I_h as the final objective function value, the trajectory $(1,h)^{k'}$ must be selected in each of the M runs. This happens with probability $\left(P_{1h}^{k'}\right)^M$:

$$\left(P_{1h}^{k'}\right)^M = \left(\frac{D_{1h}^{k'}}{\sum_{(i,j)^k \in F_1} D_{ij}^k}\right)^M, \text{where } D_{1h}^{k'} = \left(\frac{I_h}{c_{1h}^{k'}}\right)^q \tag{5.31}$$

As $I_h \to 0$, $\left(P_{ij}^{k'}\right)^M \to 0$. In addition, the likelihood of ending up with I_h decreases exponentially with increasing M since $\left(P_{ij}^{k'}\right)^M \to 0$ as $M \to \infty$. In the following, we also experimentally show that increasing the value of parameter M improves the overall performance of PRSH. Both the average and worst-case empirical results of PRSH are very good for intermediate to large values of parameter M.

PRSH runs in $O(M \cdot |N|^2 \cdot (\max_{i,j \in N} |K_{ij}|)^2)$ where $|K_{ij}| \leq 4 \; \forall \; i,j \in N$ and M is a user defined scalar. Therefore, we know that PRSH runs in polynomial time. We

investigate the average performance of PRSH using different resource restrictions and M values on the 15 and 30-node problems in Table 5.3 and Table 5.4, respectively. The details of the problem instances are explained later in conjunction with computational experiments. I_j is integer-valued and varies in a scale from 1 to 5. Column 1 shows the value of M, and Columns 2-6, 7-11, 12-16 show the statistics for (T, R) pairs (10, 100), (15, 200), and (24, 500), respectively. Z^* represent the optimal objective function value. The objective function value of the worst feasible solution is 1 for all cases. For each setting, we run PRSH 200 times to obtain statistical significance. For each replication, we calculate the percent deviation of PRSH's approximation from the optimal objective function value. The maximum, average, and standard deviation values of these 200 deviations are reported in columns *% deviation* CPU columns report the average and standard deviation of the CPU times in seconds of the 200 replications.

On the average, PRSH is able to approximate the optimal objective function value well. The average approximation quality of PRSH is very good. As expected, average and standard deviation of the percent deviations from the optimal objective function value decreases with increasing M. The CPU times increase roughly linearly with M. There is a trade-off between solution time and approximation quality, and the selection of M is a design choice for the route planner. We recommend larger M values when the restrictions on resources are larger. We use $M = 1000$ in the rest of our computational experiments.

Table 5.3: Average performance of PRSH on the 15-node problem

	$T = 10, R = 100, Z^* = 13.29$					$T = 15, R = 200, Z^* = 28.20$					$T = 24, R = 500, Z^* = 49.2$				
	% deviation			**CPU**		**% deviation**			**CPU**		**% deviation**			**CPU**	
M	***max***	***avg***	***sd***	***avg***	***sd***	***max***	***avg***	***sd***	***avg***	***sd***	***max***	***avg***	***sd***	***avg***	***sd***
1	30.02	4.18	9.76	0.03	0.03	28.23	17.55	5.56	0.09	0.06	32.30	18.35	6.79	0.05	0.08
10	0	0	0	0.18	0.02	17.70	11.68	3.71	0.35	0.04	22.28	10.51	4.47	0.46	0.05
100	0	0	0	1.82	0.08	7.09	5.85	2.57	3.08	0.11	12.17	8.20	2.39	4.39	0.06
1000	0	0	0	17.45	0.60	6.95	2.14	3.22	29.58	0.86	9.98	4.89	2.51	43.82	0.23
2000	0	0	0	34.01	0.35	6.95	0.41	1.65	57.57	0.24	7.95	4.13	2.49	87.77	0.97
4000	0	0	0	68.11	1.21	0	0	0	114.60	0.35	7.91	3.51	2.25	175.14	1.70

Table 5.4: Average performance of PRSH on the 30-node problem

	$T = 10, R = 100, Z^* = 18.22$					$T = 15, R = 200, Z^* = 38.21$					$T = 24, R = 500, Z^* = 80.22$				
	% deviation			CPU		% deviation			CPU		% deviation			CPU	
M	***max***	***avg***	***sd***	***avg***	***sd***	***max***	***avg***	***sd***	***avg***	***sd***	***max***	***avg***	***sd***	***avg***	***sd***
1	38.42	17.79	8.8	0.10	0.10	38.94	20.90	7.03	0.32	0.19	42.23	25.09	7.92	1.00	0.51
10	16.25	9.62	3.21	0.52	0.05	20.83	12.34	3.93	0.93	0.14	29.91	17.80	5.56	2.06	0.40
100	10.7	1.25	2.95	5.06	0.21	13.11	6.81	2.56	8.32	0.19	21.22	13.55	3.52	14.35	0.33
1000	0	0	0	47.95	1.28	7.75	3.16	1.78	78.8	1.21	14.92	8.56	2.39	132.98	0.27
2000	0	0	0	94.61	0.90	5.18	2.30	1.71	156.88	1.71	13.70	7.36	2.49	265.47	0.35
4000	0	0	0	189.99	2.64	5.18	1.65	1.39	314.33	4.29	11.24	6.16	1.95	532.76	3.81

5.4.3 Finding an Alternative Upper Bound

For a maximization type MIP problem, the objective function value of the best-known feasible solution is a lower bound, and the best bound known on the optimal objective function value obtained through relaxations is the upper bound. The GUROBI solver finds initial lower and upper bounds, and updates them until the difference between the two bounds falls below a desired optimality gap. In our experiments, especially for large instances, we observe that the GUROBI Solver is slow in reducing the gap between these bounds. The improvement in the upper bound seems to be especially slow. As an alternative, we use the optimal solution of Model Q as an upper bound on the optimal objective function value of Model S. Since Model Q is a relaxation of Model S, its optimal solution (Z_Q^*) is an upper bound for any solution of Model S.

We relax the following constraints of Model S to develop Model Q:

1. **Arrival time constraints:** We replace the Constraints (5.4) and (5.5) that determine the exact arrival times to the visited targets with Constraints (5.32) and (5.33) that allow the arrival times to take values within intervals.

$$t_j \leq t_i + \max_{k \in K_{ij}} t_{ij}^k + MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.32)$$

$$t_j \geq t_i + \min_{k \in K_{ij}} t_{ij}^k - MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.33)$$

2. **Binary requirements on the trajectory variables (x_{ij}^k):** We replace Constraint (5.17), the binary requirement constraints on the trajectory selection variables, with Constraints (5.34) - (5.36) that force the model to use a convex combination of the efficient trajectories that connect two consecutively-visited targets.

$$x_{ij}^k \geq 0 \qquad \forall i \in N\backslash\{n\}, \forall j \in N\backslash\{1\}\ i \neq j, k \in K_{ij} \qquad (5.34)$$

$$\sum_{k \in K_{ij}} x_{ij}^k = z_{ij} \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.35)$$

$$z_{ij} \in \{0,1\} \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.36)$$

For the sake of completeness, we present Model Q below:

Model Q

$$\max \sum_{j \in N\backslash\{1,n\}} I_j - \alpha \cdot (\overline{FD} + \overline{RDT}) \qquad (5.37)$$

s.t.

$$FD \leq T \qquad (5.38)$$

$$RDT \leq R \qquad (5.39)$$

$$\sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(t_{ij}^k x_{ij}^k\right) - FD = 0 \qquad (5.40)$$

$$\sum_{i \in N\backslash\{n\}} \sum_{\substack{j \in N\backslash\{1\} \\ j \neq i}} \sum_{k \in K_{ij}} \left(r_{ij}^k x_{ij}^k\right) + \sum_{j \in N\backslash\{1,n\}} r_j \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k - RDT = 0 \qquad (5.41)$$

$$t_j \leq t_i + \max_{k \in K_{ij}} {t_{ij}}^k + MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.42)$$

$$t_j \geq t_i + \min_{k \in K_{ij}} {t_{ij}}^k - MT\left(1 - \sum_{k \in K_{ij}} x_{ij}^k\right) \qquad \forall i \in N\backslash\{n\},\ \forall j \in N\backslash\{1\}, i \neq j \qquad (5.43)$$

$$t_j \leq MT \sum_{\substack{i \in N\backslash\{n\} \\ i \neq j}} \sum_{k \in K_{ij}} x_{ij}^k \qquad \forall j \in N\backslash\{1,n\} \qquad (5.44)$$

$$t_1 = 0 \qquad (5.45)$$

$$s^p y_j^p \leq t_j < s^{p+1} y_j^p + MY\left(1 - y_j^p\right) \qquad \forall j \in N\backslash\{1,n\}, \forall p \in P\backslash\{l\} \qquad (5.46)$$

$$s^l y_j^l \le t_j < e^l y_j^l + MY\left(1 - y_j^l\right) \qquad \forall j \in N\backslash\{1,n\} \quad (5.47)$$

$$\sum_{p\in P} y_p^j = 1 \qquad \forall j \in N\backslash\{1,n\} \quad (5.48)$$

$$I_j \le MG_j \sum_{\substack{i\in N\backslash\{n\}\\ i\neq j}} \sum_{k\in K_{ij}} x_{ij}^k \qquad \forall j \in N\backslash\{1,n\} \quad (5.49)$$

$$I_j \le \sum_{p\in P} g_j^p y_j^p \qquad \forall j \in N\backslash\{1,n\} \quad (5.50)$$

$$I_j \ge \sum_{p\in P} g_j^p y_j^p - MG_j \left(1 - \sum_{\substack{i\in N\backslash\{n\}\\ i\neq j}} \sum_{k\in K_{ij}} x_{ij}^k\right) \qquad \forall j \in N\backslash\{1,n\} \quad (5.51)$$

$$\sum_{j\in N\backslash\{1\}} \sum_{k\in K_{1j}} x_{1j}^k = 1 \qquad (5.52)$$

$$\sum_{i\in N\backslash\{n\}} \sum_{k\in K_{in}} x_{in}^k = 1 \qquad (5.53)$$

$$\sum_{\substack{i\in N\backslash\{n\}\\ i\neq j}} \sum_{k\in K_{ij}} x_{ij}^k - \sum_{\substack{i\in N\backslash\{1\}\\ i\neq j}} \sum_{k\in K_{ij}} x_{ji}^k = 0 \qquad \forall j \in N\backslash\{1,n\} \quad (5.54)$$

$$\sum_{k\in K_{ij}} x_{ij}^k = z_{ij} \qquad \forall i \in N\backslash\{n\}, \forall j \in N\backslash\{1\}\ i \neq j \quad (5.55)$$

$$x_{ij}^k \ge 0 \qquad \forall i \in N\backslash\{n\}, \forall j \in N\backslash\{1\}\ i \neq j, k \in K_{ij} \quad (5.56)$$

$$z_i^j \in \{0,1\} \qquad \forall i \in N\backslash\{n\}, \forall j \in N\backslash\{1\}\ i \neq j \quad (5.57)$$

$$y_j^p \in \{0,1\} \qquad \forall j \in N\backslash\{1,n\}, \forall p \in P \quad (5.58)$$

$$0 \le I_j \le \max_{p\in P}\left(g_j^p\right) \qquad \forall j \in N\backslash\{1,n\} \quad (5.59)$$

$$0 \le t_j \le C \qquad \forall j \in N\backslash\{1\} \quad (5.60)$$

Objective function (5.37) maximizes the sum of collected information from the visited targets and is augmented with the other two objective functions, the scaled total flight duration $\overline{FD}$ and the scaled total radar detection threat $\overline{RDT}$, both multiplied by a sufficiently small positive constant, α. Constraints (5.38) and (5.39) are resource restrictions. Constraints (5.40) and (5.41) determine the values of FD and RDT, respectively. Constraints (5.42) - (5.44) determine an interval for the visited targets, and sets arrival time to an unvisited target to 0. Constraints (5.46) -

(5.48) define the time periods that the arrival times to the visited targets belong to. Constraints (5.49) - (5.51) set the information gain from the unvisited targets to 0, and from the visited targets to the information available at the arrival times. Constraints (5.52) - (5.55) are flow balance equations. Constraints (5.56) - (5.60) specify the decision variables. Constraints (5.55) and (5.57) ensure that if there is a flow from a node to another, then the summation of the flows on all trajectory options between that node pair must be 1. MT and MY can be set to T. MG_j can be set to $\underset{p \in P}{max}(g_j^p)$ for each target $j \in N \setminus \{1, n\}$.

Any feasible solution to Model S is feasible to Model Q, but a feasible solution of Model Q is not guaranteed to be feasible to Model S. Let P_S and P_Q represent the feasible regions of Model S and Model Q, respectively.

Proposition 5.4: The optimal objective function value of Model Q, Z_Q^*, is an upper bound for the objective function value of any feasible solution of Model S, Z_S.

Proof of Proposition 5.4: Since $P_S \subseteq P_Q$, $Z_Q^* \geq Z_S$.

Proposition 5.5: PRSH's solution is a feasible solution for Model Q.

Proof of Proposition 5.5: PRSH is guaranteed to find a feasible solution for Model S by its design and any feasible solution for Model S is also a feasible solution for Model Q as $P_S \subseteq P_Q$.

5.5 Computational Experiments

We programmed PRSH using R 3.5.3 programming language (R Core Team, 2021) and solved the models using GUROBI Optimizer 8.1.0 (Gurobi Optimization, 2021) through R 3.5.3 programming language on a Dell 32-core computer running Linux Centos 7.5.x with a processor Intel Xeon Gold 6130 CPU, @32 x 2.10GHz and 192 GB usable RAM (CCR, 2021).

5.5.1 Problem Instances and Parameter Settings

The UAVs are employed for limited time missions due to battery or fuel limitations. For example, MQ-1 Predator has approximately 24-hour flight endurance, and can fly up to approximately 135 km/h in a range of 1240 km (US Air Force, 2015). We assume that the UAV is able to fly at most 24 hours at a constant speed of 100 km/h in a square operation area with a side length of 500 km.

We generate problem instances of 15 and 30 nodes (see Figure 5.5). These problem instances represent practical reconnaissance missions well as the number of targets on a mission is often within this range as the UAVs are restricted by their battery lives or fuel tanks. The core sub-region (dark grey) and the outer sub-region (light grey) of the radars have 30 km and 70 km radii, respectively, in our application. There is also an additional radar detection threat at each target location $j \in N \setminus \{1, n\}$. We generate the detection threat values at the target locations from a discrete uniform distribution between 10 and 50. We randomly generate the available information at target j in time period p, g_j^p, from the discrete uniform distribution between between 1 and 5. We assume four time periods of six hours each for each target. Therefore, time period set is $P = \{1,2,3,4\}$, and start times of the periods are $s^1 = 0,\ s^2 = 6,\ s^3 = 12,\ s^4 = 18$, respectively, for each target. We set the maximum time that the UAV can operate, C, to 24 and hence the end time of the fourth period is $e^4 = 24$.

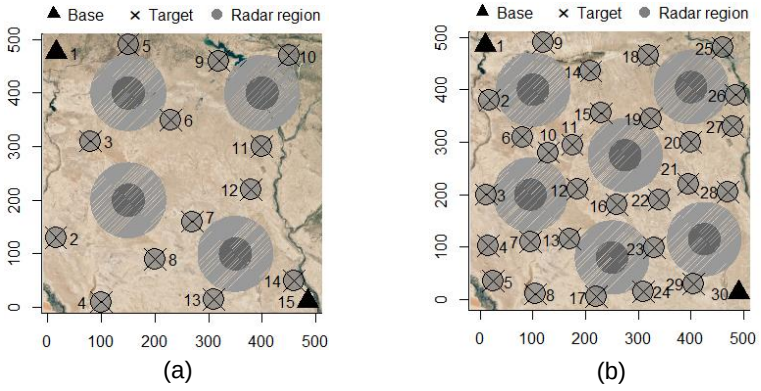


Figure 5.5: Problem instances: (a) 15-node problem, (b) 30-node problem

5.5.2 Limits of the Exact Model

To establish the computational limitations of the exact model, we solve Model S for 12 cases with different time T and radar detection threat R restrictions. We present CPU times in Table 5.5. Column # shows the order of run. Columns T and R show the resource restrictions. The last two columns show the CPU times (in seconds) spent to find the optimal solution.

As T and R values increase, so do the solution times. This is expected because as the solution space is relaxed there are more solutions to evaluate. We observe that the solution times are excessive in some of the cases. For example, Run 12 of the 30-node problem takes 1,154,544.395 CPU seconds (320.7 CPU hours), where the UAV visits 23 of the 30-nodes. This is consistent with Vansteenwegen et al. (2011)'s observation that the OP instances are difficult to solve when the visited nodes turn out to be a little more than half of the total nodes.

Table 5.5: Run times of the GUROBI solver for Model S on 15 and 30-node problems

#	T	R	15-node CPU	30-node CPU
1	15	200	58	276
2	15	300	333	1,927
3	15	400	145	2,399
4	15	500	131	463
5	20	200	168	5,284
6	20	300	1,329	17,497
7	20	400	1,356	562,090
8	20	500	768	1,154,544
9	24	200	349	11,319
10	24	300	3,221	25,595
11	24	400	8,960	157,336
12	24	500	3,103	462,080

5.5.3 Experimenting with HA

We compare the performances of Model S and HA, and study the results of HA in more detail. We define the maximum allowed deviation from the optimal solution as

5%, i.e. $U = 5\%$, and we use the optimality gap of 5% as the termination criteria for Model S as well in order to have a meaningful comparison.

Termination gaps of Model S, PRSH and HA are reported in columns gap^S, gap^P and gap^H, respectively. All gaps are calculated with the lower and upper bounds they use. Specifically, termination gaps of Model S, PRSH, and HA are calculated using Equations (5.61), (5.62) and (5.63), respectively. Here, Z_S, Z_P, Z_H and Z_Q^* are the objective function values of the best feasible solutions found by Model S, PRSH, HA and the optimal solution of Model Q, respectively. UB_S and UB_S^H are the best upper bounds on the optimal objective function value returned by the GUROBI solver upon termination when Model S is solved alone and when Model S is solved in HA, respectively.

$$gap^S = \frac{UB_S - Z_S}{Z_S} \tag{5.61}$$

$$gap^P = \frac{min\left(Z_Q^*, UB_S^H\right) - Z_P}{Z_P} \tag{5.62}$$

$$gap^H = \frac{min\left(Z_Q^*, UB_S^H\right) - Z_H}{Z_H} \tag{5.63}$$

5.5.3.1 Experiments on the 15-node Problem

We compare Model S and HA for 12 different T and R combinations in Table 5.6. Z^* represents the optimal objective function value, dev^S and dev^H represent the percentages deviations of the objective values obtained by the corresponding methods from Z^*. The remaining abbreviations in the table are as defined above.

Table 5.6: Results of Model S and HA for the 15-node problem

				Model S					HA					
#	T	R	Z^*	Z_S	UB_S	gap^S	dev^S	CPU	Z_H	Z_Q^*	UB_S^H	gap^H	dev^H	CPU
1	15	200	28.204	28.204	28.204	0.0%	0.0%	63	28.204	28.205	-	0.0%	0.0%	52
2	15	300	29.336	29.336	30.668	4.5%	0.0%	316	29.333	30.309	-	3.3%	0.0%	75
3	15	400	30.325	30.325	30.325	0.0%	0.0%	162	30.325	30.398	-	0.2%	0.0%	93
4	15	500	30.395	30.395	31.359	3.2%	0.0%	131	30.395	30.451	-	0.2%	0.0%	123
5	20	200	33.235	33.235	34.864	4.9%	0.0%	64	33.222	33.235	-	0.0%	0.0%	118
6	20	300	39.204	39.204	41.156	5.0%	0.0%	809	38.214	39.204	-	2.6%	2.5%	332
7	20	400	41.280	41.280	43.344	5.0%	0.0%	1,364	41.280	42.222	-	2.3%	0.0%	275
8	20	500	42.269	42.269	44.382	5.0%	0.0%	703	41.325	42.297	-	2.4%	2.2%	271
9	24	200	35.233	35.233	36.985	5.0%	0.0%	200	34.255	35.233	38.208	2.9%	2.8%	159
10	24	300	42.213	42.213	44.323	5.0%	0.0%	1,964	42.213	42.223	-	0.0%	0.0%	1,022
11	24	400	47.233	47.233	49.593	5.0%	0.0%	6,403	46.218	47.233	-	2.2%	2.1%	1,460
12	24	500	49.213	49.213	51.672	5.0%	0.0%	3,021	48.237	49.237	60.655	2.1%	2.0%	1,668

HA runs in a small fraction of the time required by Model S and terminates with solutions that are either the same or only slightly worse than the solutions of Model S. Specifically, the total CPU time of 12 cases is 15,199 seconds for Model S and 5,647 seconds for HA. HA takes about a third of the total CPU time of Model S. There are cases where improvements are more. For example, in Case 7, HA finds the same (optimal) solution as Model S in a fifth of the duration of Model S. In Case 11, HA deviates only 2.1% from the optimal solution in a run time of about a fifth of Model S.

In general, the termination gaps of HA (gap^H) are smaller than Model S (gap^S), due to the fact that HA obtains better upper bounds by solving Model Q. This shows that HA terminates with stronger evidence for the quality of its solution compared to that of Model S when the optimal solution is unknown. In 10 of the 12 cases, HA terminates with the solution of PRSH, as gap^H is smaller than 5%. In all cases, HA uses Z_Q^* as the upper bound when terminating.

We further investigate the performances of the components of HA in Table 5.7. Columns 4-5 show the termination gaps of PRSH and HA, respectively. Columns 6-9 show the individual CPU times of HA's elements and their total. In general, Model Q dominates the total execution time of HA, and it grows with T and R values. The

termination gaps of PRSH are very small in most cases and slightly exceed 5% in two cases. This shows that PRSH often approximates the optimal solutions well and Model Q provides good enough bounds to establish solution quality.

Table 5.7: Further investigation of HA for the 15-node problem

			GAP		CPU			
#	**T**	**R**	**gap^P**	**gap^H**	**PRSH**	**Model Q**	**Model S**	**Total**
1	15	200	0.0%	0.0%	30.95	20.79	-	51.74
2	15	300	3.3%	3.3%	33.26	41.63	-	74.89
3	15	400	0.2%	0.2%	36.06	56.54	-	92.60
4	15	500	0.2%	0.2%	34.54	88.69	-	123.22
5	20	200	0.0%	0.0%	36.26	82.03	-	118.29
6	20	300	2.6%	2.6%	41.27	290.43	-	331.70
7	20	400	2.3%	2.3%	46.41	228.49	-	274.89
8	20	500	2.4%	2.4%	43.25	228.11	-	271.36
9	24	200	5.8%	2.9%	41.45	99.99	17.10	158.55
10	24	300	0.0%	0.0%	43.84	977.78	-	1,021.62
11	24	400	2.2%	2.2%	51.11	1,409.26	-	1,460.37
12	24	500	6.4%	2.1%	49.47	1,610.83	7.74	1,668.05

5.5.3.2 Experiments on the 30-node Problem

We compare Model S and HA in Table 5.8, which has the same structure as Table 5.6. Similar to the results of the 15-node problem, we sacrifice slightly from the final solution's performance in HA in return for substantial gain in computation times. The total CPU time required to solve 12 cases is 659,844 seconds for Model S and 277,462 seconds for HA. HA runs in about 40% of the total CPU time required by Model S. The solution time of HA is as small as a fourth of that of Model S in Case 7. Case 8 has the longest duration with 157,747 CPU seconds. We are able to obtain this solution in approximately 1.4 hour real time with the 32-core computer we experiment on.

Table 5.8: Results of Model S and HA on the 30-node problem

				Model S					HA					
#	T	R	Z^*	Z_S	UB_S	gap^S	dev^S	CPU	Z_H	Z_Q^*	UB_S^H	gap^H	dev^H	CPU
1	15	200	38.211	38.209	40.048	4.8%	0.0%	413	37.228	38.221	-	2.7%	2.6%	566
2	15	300	47.226	47.226	49.571	5.0%	0.0%	1,720	45.243	47.226	-	4.4%	4.2%	985
3	15	400	55.224	55.224	57.954	4.9%	0.0%	2,014	53.208	55.224	62.063	3.8%	3.7%	1,329
4	15	500	59.210	59.210	61.546	3.9%	0.0%	482	57.228	59.210	80.844	3.5%	3.3%	691
5	20	200	41.259	41.259	43.312	5.0%	0.0%	2,327	40.213	41.259	-	2.6%	2.5%	3,441
6	20	300	54.208	54.203	56.913	5.0%	0.0%	6,060	53.205	54.213	63.905	1.9%	1.9%	8,105
7	20	400	63.209	63.209	66.369	5.0%	0.0%	150,286	61.204	64.200	80.890	4.9%	3.2%	38,470
8	20	500	70.220	70.208	73.719	5.0%	0.0%	418,352	69.213	71.207	81.748	2.9%	1.4%	157,747
9	24	200	44.238	44.226	46.436	5.0%	0.0%	3,935	44.201	44.250	50.354	0.1%	0.1%	3,832
10	24	300	59.208	58.221	61.129	5.0%	1.7%	4,308	57.222	59.208	64.639	3.5%	3.4%	3,536
11	24	400	70.218	70.218	73.729	5.0%	0.0%	12,464	67.210	70.218	80.957	4.5%	4.3%	9,749
12	24	500	80.216	80.216	84.226	5.0%	0.0%	57,484	78.217	80.216	91.166	2.6%	2.5%	49,012

HA provides slightly stronger evidence about the optimality of its solution, when optimal solution is not known, as its termination gaps are smaller than the termination gaps of Model S. The columns of UB_S^H and Z_Q^* show that HA uses Z_Q^* as the upper bound for the termination in all cases. We provide detailed analysis of HA for the 30-node problem in Table 5.9, using the same structure of Table 5.7. Model Q dominates the total execution time of HA. Compared to the 15-node problem, HA resorts to Model S more frequently in order to improve the solutions produced by PRSH.

Table 5.9: Further investigation of HA for the 30-node problem

			GAP		CPU			
#	T	R	gap^P	gap^H	PRSH	Model Q	Model S	Total
1	15	200	2.7%	2.7%	81.49	484.25	-	565.74
2	15	300	4.4%	4.4%	90.34	894.84	-	985.18
3	15	400	7.8%	3.8%	93.59	987.25	247.73	1,328.57
4	15	500	9.2%	3.5%	97.90	591.80	1.09	690.8
5	20	200	2.6%	2.6%	94.51	3,346.66	-	3,441.17
6	20	300	5.8%	1.9%	105.85	7,963.75	35.78	8,105.38
7	20	400	6.6%	4.9%	113.00	38,355.84	0.92	38,469.76
8	20	500	5.9%	2.9%	115.30	157,521.00	110.27	157,746.57
9	24	200	7.3%	0.1%	94.91	3,736.69	0.84	3,832.44
10	24	300	9.2%	3.5%	116.71	3,123.96	294.96	3,535.63
11	24	400	9.3%	4.5%	128.61	9,619.72	0.95	9,749.27
12	24	500	9.5%	2.6%	132.67	48,875.53	3.37	49,011.57

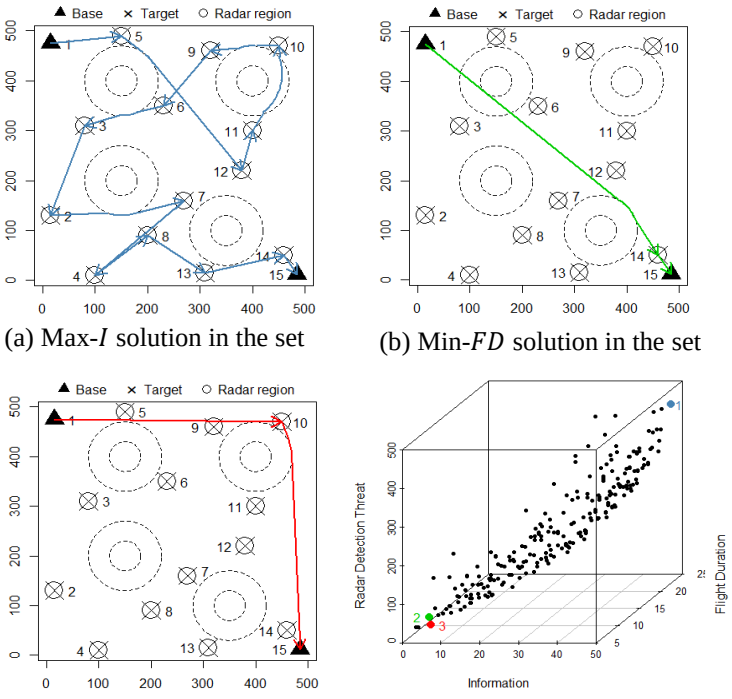
5.5.4 Generating the Pareto-optimal Frontiers of the Problem Instances

We approximate the Pareto-optimal sets of the 15-node and 30-node problems solving HA progressively while decreasing T and R restrictions in each iteration. We first set $T = 24$ and $R = 2000$. When a solution is returned, we update R to a value slightly smaller then the RDT of that solution. When the model becomes infeasible, we decrease T by 30 min, set $R = 2000$ and start the same process again. The procedure terminates when the new T value does not produce a feasible solution.

The solution set we find approximates the Pareto-optimal set for two reasons. First, we have the possibility of missing some solutions since we decrease T by discrete intervals (30 mins). Second, the objective function value of HA's solution can deviate from the objective function value of the optimal solution up to 5% as $U = 5\%$. The aim here is to present an approximation of the Pareto-optimal set to the route planner. Based on his/her preferences, a further detailed search can be conducted in the preferred region by decreasing the value of parameter U, if necessary.

5.5.4.1 The 15-node Problem

We present an approximation of the Pareto-optimal set of the 15-node problem (Figure 5.6(d)) and three efficient solutions having max I (Figure 5.6(a)), min FD (Figure 5.6(b)), and min RDT (Figure 5.6(c)). Max-I, min-FD and min-RDT solutions and are the solutions with the largest I, smallest FD, and smallest RDT values in the approximate set, respectively. The objective function values of max-I solution are $I = 49$, $FD = 23.35$, $RDT = 453.87$; min-FD solution are $I = 5$, $FD = 6.83$, $RDT = 50.00$; and min-RDT solution are $I = 3$, $FD = 9.02$, $RDT = 10.00$.



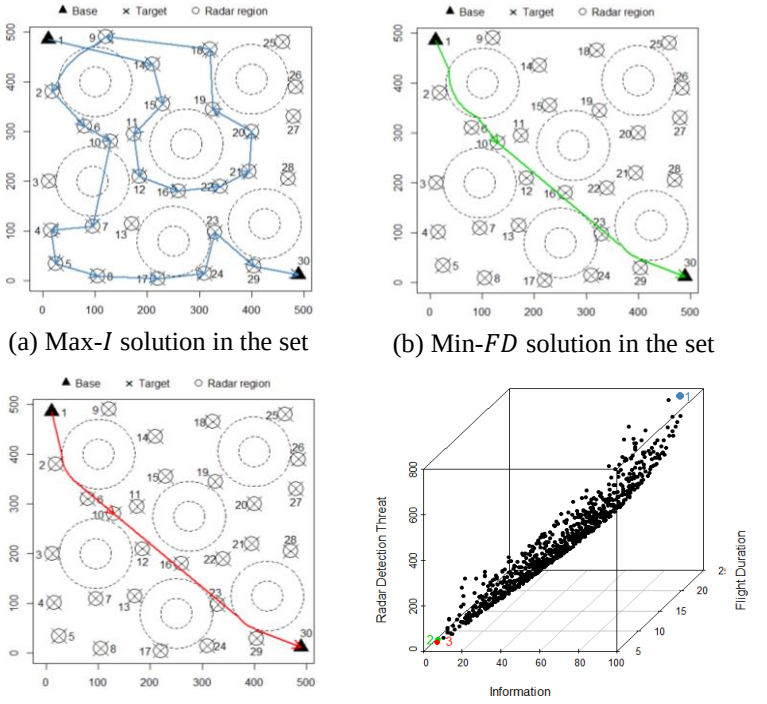
(a) Max-I solution in the set (b) Min-FD solution in the set

(c) Min-RDT solution in the set (d) Approximate Pareto-optimal set

Figure 5.6: Approximation of the Pareto-optimal set of the 15-node problem

5.5.4.2 The 30-node Problem

We present the results of the 30-node problem in Figure 5.7, using the same structure as the 15-node problem. Max-I solution (Figure 5.7(a)), min-FD solution (Figure 5.7(b)), and min-RDT solution (Figure 5.7(c)) are the solutions with the largest I, smallest FD, and smallest RDT values in the approximate set, respectively. The objective function values of the solutions are $I = 91$, $FD = 23.68$, $RDT = 791.61$; $I = 3$, $FD = 6.97$, $RDT = 18.77$; $I = 3$, $FD = 7.04$, $RDT = 10.00$, respectively.



(a) Max-I solution in the set (b) Min-FD solution in the set

(c) Min-RDT solution in the set (d) Approximate Pareto-optimal set

Figure 5.7: Approximation of the Pareto-optimal set of the 30-node problem

5.6 Demonstration on the Colorado Case Study

There does not exist readily available application data for our problem. In order to demonstrate our approach and create useful application data, we develop an approach that uses the topographical features of a region to generate problem instances. We implement our approach in the State of Colorado.

We need: (1) locations of targets, radars and bases, (2) available information at targets, (3) radar detection risks at targets.

(1) Colorado has 58 fourteeners (a mountain peak with an elevation in excess of 14,000 feet) reported by (Colorado Geological Survey, 2011). We use the peak points of the fourteeners as the locations of the radars. Colorado is rich in terms of water reservoirs and we use the locations of 41 water reservoirs from the region as the locations of targets. We set Blanding, Utah and Goodland, Kansas as the home

and final bases of the UAV, respectively. They are close enough to Colorado and both have small airports where a UAV can take off and land.

(2) We base the information content of a water reservoir on its water capacity assuming higher levels correspond to more information. We randomly generate information for each period from the discrete uniform distribution between 1-5, ensuring that the average of the available information of the time periods of a water reservoir is at least as much as the average of the time periods of the ones with less capacity.

(3) We determine the radar detection threat at a water reservoir based on its capacity as well. We randomly generate threat values from a discrete uniform distribution in the range 10 to 30. We then assign these threat values to reservoirs in the order of their capacities, highest threat being assigned to the reservoir with largest capacity.

We use ArcGIS Pro 2.4.2® software throughout this section to generate maps, and the location and elevation data (Esri, 2021).We present the Colorado map in Figure 5.8. New problem instances can be generated using this data set, by defining minimum elevation levels for fourteeners and water reservoirs. For our purposes in this paper, we generate two problem instances. For the first problem, we select fourteeners that have elevations of at least 14,270 feet as our radar locations and select water reservoirs that have elevations of at least 9,000 feet as our target locations. For the second problem, we decrease the elevation limits to 14,200 feet and to 8,000 feet for fourteeners and water reservoirs, respectively, in order to increase the problem size. We assume that core and outer radar regions of radars have 20 km and 50 km inner and outer radii, respectively. Some radar regions may intersect with each other. If they intersect in the inner radar regions, we arbitrarily select and use one of them as the effective radar. If the intersection is in the outer radar region, we manually adjust the locations of the radar regions on the map until they are fully separated. Some targets may fall inside radar regions. In such cases, we move the target to the closest point that is outside the radar region, since we assume that each target has its own radar protection.

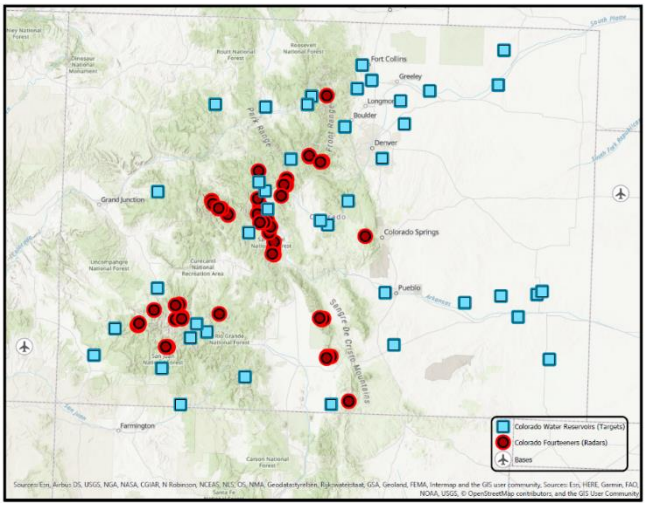


Figure 5.8: The map of Colorado's 58 fourteeners and 41 water reservoirs

5.6.1 Problem Instance 1

This problem instance has 5 radar regions, 10 targets, and 2 bases (Figure 5.9(a)). Model S and HA for 12 different T and RDT combinations are compared in Table 5.10. Column 1 indicates run order, Columns 2-3 give T and R restrictions, and Column 4 shows the optimal solution (Z^*). Columns 5-9 and 10-15 show the results of Model S and HA, respectively. Z_S and Z_H are the objective function values of the best feasible solutions of Model S and HA, respectively. gap^S and gap^H report the termination gaps of Model S and HA, respectively. UB_S and UB_S^H are the best upper bounds on the optimal objective function value returned by the GUROBI solver upon termination when Model S is solved alone and when Model S is solved in HA, respectively. CPU columns report the CPU times spent in seconds until termination. dev^S and dev^H represent the percentages deviations of the objective values obtained by the corresponding methods from Z^*.

Table 5.10: Results of Model S and HA on the Colorado Problem Instance 1

				Model S					HA					
#	T	R	Z^*	Z_S	UB_S	gap^S	dev^S	CPU	Z_H	Z_Q^*	UB_S^H	gap^H	dev^H	CPU
1	15	200	35.292	35.292	36.278	2.8%	0.0%	21.37	35.292	37.226	36.265	2.8%	0.0%	39.88
2	15	300	37.329	37.326	37.469	0.4%	0.0%	10.00	37.329	37.351	-	0.1%	0.0%	38.40
3	15	400	37.397	37.397	37.438	0.1%	0.0%	8.19	37.397	37.413	37.438	0.0%	0.0%	51.29
4	15	500	37.438	37.438	37.476	0.1%	0.0%	7.90	37.438	37.451	-	0.0%	0.0%	45.39
5	20	200	43.247	42.264	43.729	3.5%	2.3%	7.44	42.273	43.247	-	2.3%	2.3%	47.80
6	20	300	43.367	41.386	43.395	4.9%	4.8%	10.51	42.383	43.367	-	2.3%	2.3%	45.43
7	20	400	43.427	42.360	43.471	2.6%	2.5%	8.93	42.453	43.427	-	2.3%	2.2%	50.58
8	20	500	43.463	42.489	44.374	4.4%	2.3%	11.82	42.489	43.463	-	2.3%	2.2%	51.32
9	24	200	43.313	43.284	45.242	4.5%	0.1%	6.51	43.290	43.313	-	0.1%	0.1%	75.38
10	24	300	43.433	43.375	45.397	4.7%	0.1%	0.59	43.404	43.433	-	0.1%	0.1%	64.20
11	24	400	43.493	43.478	45.447	4.5%	0.0%	4.83	43.476	43.493	-	0.0%	0.0%	56.66
12	24	500	43.529	43.497	45.498	4.6%	0.1%	0.35	43.508	43.529	-	0.0%	0.0%	67.25

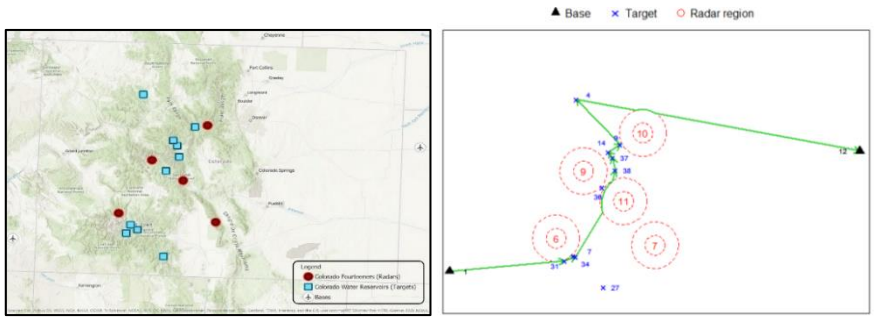
HA finds solutions that are either the same or only slightly better than the solutions of Model S in terms of deviation of the results from the optimal objective function value. Moreover, HA is better in terms of providing stronger evidence on the quality of its solution compared to that of Model S since the termination gaps of HA (gap^H) are smaller than those of Model S (gap^S). However, the solution times of Model S are shorter than HA. This is expected because this problem is a small-sized problem with only 12 nodes and exact models can be solved fast. We expect HA to prove its advantage in terms of CPU requirements on the Colorado Problem Instance 2 since it is relatively a larger problem.

We investigate the performances of the components of HA in Table 5.11. Columns 4-5 show the termination gaps of PRSH (gap^P) and HA (gap^H), respectively. Columns 6-9 show the individual CPU times of HA's elements and their total. In general, PRSH dominates the total execution time of HA. The termination gaps of PRSH are very small in most cases and slightly exceed 5% in two cases. Hence, PRSH often approximates the optimal solutions well and Model Q often provides good enough bounds to terminate the HA without solving Model S.

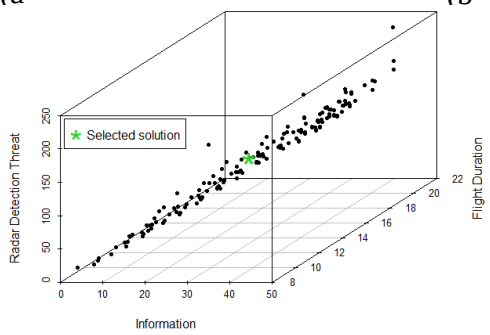
Table 5.11: Further investigation of HA for the Colorado Problem Instance 1

			GAP		CPU			
#	***T***	***R***	gap^P	gap^H	**PRSH**	**Model *Q***	**Model *S***	**Total**
1	15	200	5.5%	2.8%	29.66	10.21	6.49	46.37
2	15	300	0.1%	0.1%	30.50	7.90	-	38.40
3	15	400	5.5%	0.0%	32.42	10.59	8.28	51.29
4	15	500	0.0%	0.0%	31.60	13.79	-	45.39
5	20	200	2.3%	2.3%	34.98	12.83	-	47.80
6	20	300	2.3%	2.3%	35.95	9.48	-	45.43
7	20	400	2.3%	2.3%	36.45	14.13	-	50.58
8	20	500	2.3%	2.3%	36.50	14.81	-	51.32
9	24	200	0.1%	0.1%	37.15	38.23	-	75.38
10	24	300	0.1%	0.1%	38.04	26.16	-	64.20
11	24	400	0.0%	0.0%	38.26	18.40	-	56.66
12	24	500	0.0%	0.0%	38.24	29.01	-	67.25

The approximate Pareto-optimal set generated by HA is given in Figure 5.9(c). We show the route of an arbitrarily-selected Pareto-optimal solution in Figure 5.9(b), whose objective function values are $I = 31$, $FD = 12.87$, and $rdt = 132.58$.



(a (b



(c)

Figure 5.9: Colorado Problem Instance 1: (a) Map, (b) Route of the selected solution, (c) Pareto-optimal set.

5.6.2 Problem Instance 2

This problem instance has 7 radars, 17 targets and 2 bases (Figure 5.10(a)). This problem instance has 19 nodes and is larger than the first problem instance. We compare Model S and HA on 12 cases in Table 5.12 that has the same structure as Table 5.10.

Table 5.12: Results of Model S and HA on the Colorado Problem Instance 2

				Model S					HA					
#	T	R	Z^*	Z_S	UB_S	gap^S	dev^S	CPU	Z_H	Z_Q^*	UB_S^H	gap^H	dev^H	CPU
1	15	200	40.223	40.223	42.233	5.0%	0.0%	7,034	39.205	40.228	-	2.6%	2.6%	7,024
2	15	300	44.214	44.214	46.424	5.0%	0.0%	15,761	43.309	44.234	58.558	2.1%	2.1%	14,543
3	15	400	44.311	44.311	46.526	5.0%	0.0%	17,913	42.336	44.326	-	4.7%	4.7%	11,248
4	15	500	44.370	44.370	46.588	5.0%	0.0%	19,272	42.386	44.381	-	4.7%	4.7%	12,256
5	20	200	49.200	48.203	50.597	5.0%	2.1%	2,412	47.206	49.200	49.671	4.2%	4.2%	1,956
6	20	300	61.210	61.210	64.270	5.0%	0.0%	24,907	59.241	61.221	69.859	3.3%	3.3%	31,268
7	20	400	61.307	61.307	64.373	5.0%	0.0%	56,066	59.336	61.316	74.936	3.3%	3.3%	37,415
8	20	500	61.366	61.366	64.434	5.0%	0.0%	68,264	59.395	61.373	75.384	3.3%	3.3%	38,175
9	24	200	50.233	50.226	52.716	5.0%	0.0%	1,337	49.220	51.220	-	4.1%	2.1%	3,462
10	24	300	69.202	68.215	71.626	5.0%	1.4%	11,910	66.228	69.202	-	4.5%	4.5%	14,880
11	24	400	70.289	69.300	72.765	5.0%	1.4%	31,928	67.301	70.289	77.423	4.4%	4.4%	18,122
12	24	500	70.351	69.362	72.829	5.0%	1.4%	39,508	68.371	70.351	-	2.9%	2.9%	14,783

Deviation and CPU columns show that we sacrifice slightly from the final solution's performance in HA in return for a 30% gain in total computation time. The total CPU time required to solve 12 cases is 296,313 seconds and 205,132 seconds for Model S and HA, respectively. Termination gaps, gap^S and gap^H, show that HA provides slightly stronger evidence about the optimality of its solutions compared to that of Model S, when optimal solution is not known. This is mainly because Model Q solved in HA provides better upper bounds than the upper bounds Model S finds.

Table 5.13, which has the same structure as Table 5.11, presents performances of the components of HA for the Colorado Problem Instance 2. Unlike the results of the Colorado Problem Instance 1, the dominating component in terms of CPU requirements is now Model Q. This is expected since exact models take more time for large problem instances. PRSH's termination gaps are below 5% or slightly larger

than 5%. For the cases where termination gaps are over 5%, HA reduces the termination gaps to below 5% solving Model S quickly.

Table 5.13: Further investigation of HA for the Colorado Problem Instance 2

			GAP		CPU			
#	**T**	**R**	**gap^P**	**gap^H**	**PRSH**	**Model Q**	**Model S**	**Total**
1	15	200	2.6%	2.6%	56.24	6,968.14	-	7,024.38
2	15	300	7.2%	2.1%	55.96	14,317.99	168.78	14,542.73
3	15	400	4.7%	4.7%	56.97	11,190.81	-	11,247.78
4	15	500	4.7%	4.7%	57.20	12,198.80	-	12,256.00
5	20	200	6.4%	4.2%	61.03	1,604.26	290.63	1,955.92
6	20	300	5.1%	3.3%	68.55	30,872.12	327.37	31,268.04
7	20	400	5.1%	3.3%	67.90	37,149.57	197.68	37,415.15
8	20	500	5.1%	3.3%	70.23	38,085.08	20.10	38,175.41
9	24	200	4.1%	4.1%	69.36	3,392.51	-	3,461.87
10	24	300	4.5%	4.5%	80.73	14,799.03	-	14,879.77
11	24	400	6.0%	4.4%	79.08	18,042.69	0.20	18,121.98
12	24	500	2.9%	2.9%	83.09	14,700.36	-	14,783.45

We approximate the Pareto-optimal set using HA (Figure 5.10(c)). We also show the route of an arbitrarily-selected Pareto-optimal solution in Figure 5.10(b), whose objective function values are $I = 52$, $FD = 16.50$, and $RDT = 319.62$.

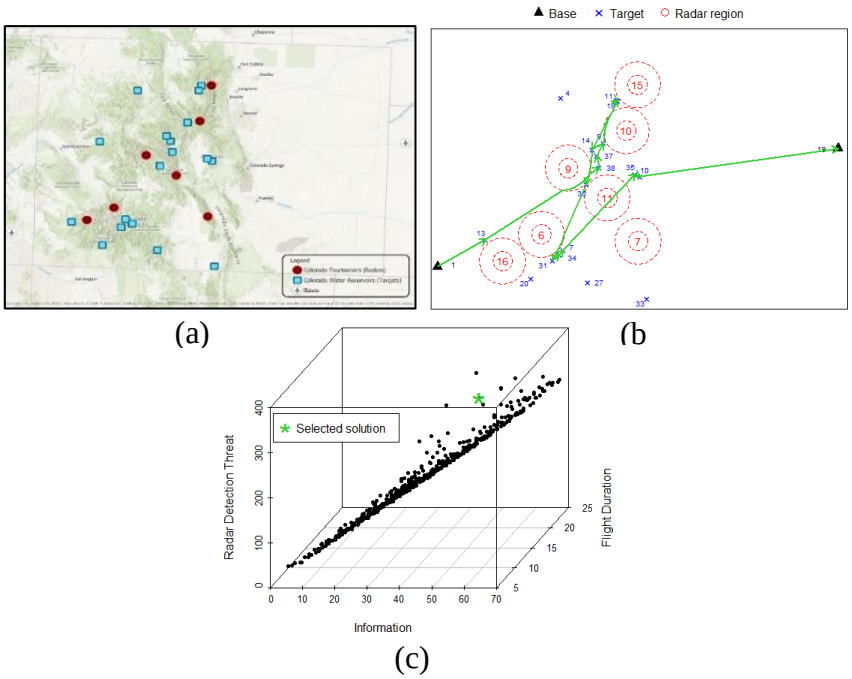


Figure 5.10: Colorado Problem Instance 2: (a) Map, (b) Route of the selected solution, (c) Pareto-optimal set.

5.7 Discussions for Chapter 5

In this chapter, we study the UAV route planning problem in a radar-monitored environment as a multi-objective orienteering problem with time dependent prizes and multiple connections. We approximate the continuous terrain that the UAV moves by allowing a finite number of representative trajectories between target pairs. We aim to generate the Pareto-optimal frontier of the problem in order to present the trade-offs between three objectives to the route planner. This study introduces a new orienteering problem to the literature and contributes to the UAV routing literature.

We formulate an MIP model for the problem in order to generate a Pareto-optimal solution based on some aspiration levels on flight duration and radar detection threat, and maximizing the collected information. Our model can be adapted to similar practical problems that include time-dependent prizes/costs. We generate the Pareto-optimal frontier using the $\varepsilon-$constraint method and systematically changing the aspiration levels. Our computational experiments demonstrate that only small-sized problems can be solved exactly in reasonable computational times. We develop a hybrid algorithm (HA) to reduce the computational effort without sacrificing much from the quality of the solutions. HA consists of two elements. The first element is a heuristic, PRSH, that approximates the optimal solution. The second element is a new MIP model which is a relaxation of the original MIP model. Its optimal solution is used as an alternative upper bound to that found by the GUROBI solver when solving the original MIP model. HA, starting with the heuristic PRSH, quickly finds a solution to the problem guaranteeing a desired solution quality.

We compare HA and the MIP model on 15 and 30-node problem instances and empirically show that HA substantially improves the solution time in exchange of a slight decrease in solution quality. We then use HA to approximate the Pareto-optimal frontiers. Lastly, we demonstrate the solution approaches on a case study by extracting information using the terrain characteristics of the State of Colorado.

This study can we extended in several ways. From a multi-objective optimization perspective, preferences of the route planner can be incorporated to the solution process to focus on the desired regions of the Pareto-optimal frontier. Since our

problem is NP-hard, focusing on a specific region of the Pareto-optimal frontier may reduce the computational burden and increase the approximation quality. Our computational experiments on four problem instances (15-node, 30-node, and two Colorado instances) show that HA significantly improves the computational requirements of the original MIP model. This finding can be further tested by conducting experiments on more problem instances in future.

We focused on the optimization of the UAV routing problem from an operational perspective in this study. We discuss some practical implementation issues below and provide some directions to enhance our study.

(I) Loitering: Currently, no loitering is allowed in the mission terrain. The UAV may be allowed to loiter in the mission terrain in order to take advantage of higher information collection by arriving at a target later.

(II) 3D Environment, Ground Obstacles, Flight Dynamics and Physical Limitations: Addressing these issues when developing trajectory options between target pairs may improve the representation of the terrain. Our model and solution approaches are directly applicable in such environments, since the trajectories are inputs to our model. There are few studies in the literature that briefly address these issues (see, for example, Mittal and Deb, 2007; Myers et al., 2016).

(III) Uncertainty: Incorporating uncertainties such as fuel consumption, information, pop-up targets could be a complex but useful future work (see, for example, Evers, Barros, et al., 2014; Evers, Dollevoet, et al., 2014; Xia et al., 2017; Cho and Batta, 2020).

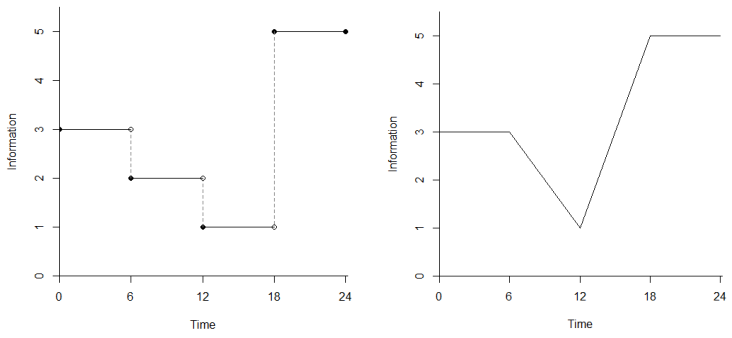
(IV) Sensors used for information collection: In practice, a UAV can carry different types of sensors that can be used interchangeably to collect different types of information. Incorporating sensor effectiveness may also prove to be a useful future research (see, for example, Mufalli et al., 2012; Moskal and Batta, 2019).

(V) Must-visit and can-visit destinations: In our consideration, it is optional to visit any of the targets. In practice, there could be two types of targets: must-visit targets need to be visited definitely, can-visit targets are visited if desirable. Incorporation of different visit types to the constraints of the problem may be a useful future work.

(VI) Alternative solution approaches: We developed a hybrid algorithm consisting of MIP models and a heuristic algorithm. Our heuristic PRSH is a route search heuristic that constructs a route by adding trajectories one at a time. There are similar alternative search algorithms used in the route planning literature. For example, A* algorithm (Hart et al., 1968) has been used in the path planning literature. Gudaitis (1994) consider A* for a UAV routing problem where the movement is between two targets and considerations are flight duration and radar detection threat. In future, A* algorithm can be adapted to our problem to cover multiple targets, target selection and three considerations (information, flight duration and radar detection threat).

(VII) We considered approximating the Pareto-optimal frontier of the OP. The study can be extended to choosing a preferred solution of a DM from approximated sets. There are many preference-based approaches in the MCDM literature. A suitable approach can be adapted to our problem, either from the start or as an approach to choose from the generated Pareto-optimal solutions, for selecting a final solution

(VIII) Continuous information case: We study this problem considering that the available information at a target changes discretely from one period to another (Figure 5.11(a)). An alternative approach could be considering continuously-changing information at targets. Continuous information case (Figure 5.11(b)) can also easily be incorporated to our model.



(a) Discrete (b) Continuous

Figure 5.11: Discrete vs continuous information change

In the continuous case, the value of information can be represented with a piece-wise linear function. Let for target j, $B_j = \{1, .. p, \dots, b\}$ be the set of break points, b_j^p be

the starting time of the break point p, and g_j^p be the available information at the break point p. Then our problem with continuous information change can be modeled by introducing the binary variables w_j^p, and replacing Constraints (5.8) - (5.13) and (5.18) of Model S with Constraints (5.64) - (5.72). Constraints (5.65) - (5.72) determine the time interval each target j is visited and Constraint (5.64) determines the value of information collected from target j.

$$I_j = \sum_{p \in B_j} g_j^p y_j^p \qquad \forall j \in N \setminus \{n\} \tag{5.64}$$

$$t_j = \sum_{p \in B_j} b_j^p y_j^p \qquad \forall j \in N \setminus \{n\} \tag{5.65}$$

$$y_j^1 \leq w_j^1 \qquad \forall j \in N \setminus \{n\} \tag{5.66}$$

$$y_j^p \leq w_j^{p-1} + w_j^p \qquad \forall p \in B_j \setminus \{1, b\}, \quad j \in N \setminus \{n\} \tag{5.67}$$

$$y_j^b \leq w_j^{b-1} \qquad \forall j \in N \setminus \{n\} \tag{5.68}$$

$$\sum_{p \in B_j} y_j^p = 1 \qquad \forall j \in N \setminus \{n\} \tag{5.69}$$

$$\sum_{p \in B_j \setminus \{b\}} w_j^p = 1 \qquad \forall j \in N \setminus \{n\} \tag{5.70}$$

$$y_j^p \geq 0 \qquad \forall p \in B_j, \quad \forall j \in N \setminus \{n\} \tag{5.71}$$

$$w_j^p \in \{0,1\} \qquad \forall p \in B_j \setminus \{b\}, \quad \forall j \in N \setminus \{n\} \tag{5.72}$$

CHAPTER 6

CONCLUSIONS

UAVs are special types of aircraft operating without a human pilot on board. In the military context, they are often employed for aerial reconnaissance operations performed in enemy terrains to explore the adversarial activities and collect information. They have become very popular in the last decade as they eliminate the risk of losing troops and are advantageous in terms of operational costs compared to the manned aircraft. In general, UAVs follow the routes stored in their memories and remote interventions on their routes during flights are limited. Hence, preparing efficient route plans that increase the success and decrease the costs of UAV missions has become crucial. In this thesis, we address the optimization of the UAV route planning problem. We have studied the problem from three different perspectives.

In the first two studies, we have studied the bi-objective and three-objective versions of a UAV route planning problem where a UAV is tasked to visit all given targets in an enemy region. In the broader sense, the problem can be classified as TSP. Route plans for these problems require determining both the visiting order of the targets and the specific trajectories to use between consecutive target pairs. In the bi-objective version, two objectives, minimization of the total flight duration and risk of being detected, are considered simultaneously. In the three-objective case, we include maximization of the collected information as the third objective. In order to make maximization of the collected information a meaningful objective for a TSP where all targets are visited in any feasible solution, time-dependent prize concept is considered. The terrain the UAV moves is considered as continuous terrain allowing infinitely many trajectory options between targets. As there are infinitely many efficient solutions of the problems, we addressed them by developing a preference-based MOEA. This algorithm elicits the preferences of route planners through

reference points and approximates preferred Pareto-optimal solutions. In addition to the preferred solutions, the algorithm is able to find an approximation of the whole Pareto-optimal frontier, which provides a general idea to route planners on the structure and positioning of the Pareto-optimal frontier. Route planners are also provided with the flexibility to change their reference points throughout the algorithm. The research contributes to the literatures of MOEA and UAV route planning. The developed algorithm is quite generic and can be adapted for the solution of any multi-objective optimization problem.

In our third study, we consider the routing of a UAV on a reconnaissance mission performed in an enemy terrain, defining the problem as an orienteering problem. In addition to determining both the visiting order of the targets and the specific trajectories to use between consecutive target pairs, now the targets to be visited need to be determined as well. Unlike the existing literature, we consider multiple routing objectives, time-dependent prizes, and multiple connections between nodes. We develop mathematical models as well as heuristics to explore the efficient solutions of the problem. The results confirm that the hybrid algorithm consisting of exact and heuristic solution approaches is able to reduce the computational requirements of the exact methods and approximates the efficient solutions of the problem well. This research contributes to the literatures of orienteering problem and UAV route planning problem as a prize collection problem in a radar-monitored terrain with time-dependent prizes and multiple connections.

We provide the future extension directions for each study at the end of the corresponding chapter. We believe this thesis has multiple important contributions that may trigger important developments in related areas. Although the UAV route planning problem has been studied widely in different domains so far, the multi-objective route planning of UAVs with multiple targets in continuous terrain stays as an important research area and will attract the attention of many researchers in the near future. This study is one of the few early studies in this area and we expect more approaches to be developed since routing in the continuous terrain represents the practice better. There are many other UAV route planning problems in different

environments. Although military-type UAV route planning problems are considered in this thesis, the approaches we developed are quite general and can be adapted to the civilian UAV problems such as inspection of wildlife, search and rescue operations, crowd monitoring, border patrolling, video recording in sport games, as well as to the operations of underwater and ground vehicles. Application of aspects such as using multiple objectives, allowing continuous terrain, and capturing time-dependent information we explored in this thesis are applicable to those problems and await future research.

REFERENCES

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks, 19*(6), 621-636.

Bérubé, J.-F., Gendreau, M., and Potvin, J.-Y. (2009). An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research, 194*(1), 39-50.

Blum, A., Chawla, S., Karger, D. R., Lane, T., Meyerson, A., and Minkoff, M. (2007). Approximation algorithms for orienteering and discounted-reward TSP. *SIAM Journal on Computing, 37*(2), 653-670.

Braysy, I., and Gendreau, M. (2005). Vehicle routing problem with time windows, part 1: Route construction and local search algorithms. *Transportation Science, 39*(1), 104-118.

Calvete, H. I., Gale, C., Oliveros, M. J., and Sanchez-Valverde, B. (2007). A goal programming approach to vehicle routing problems with soft time windows. *European Journal of Operational Research, 177*(3), 1720-1733.

CCR. (2021). The center for computational research. In *University at Buffalo*. http://hdl.handle.net/10477/79221

Chao, I. M., Golden, B. L., and Wasil, E. A. (1996). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research, 88*(3), 475-489.

Chen, N., Chen, W. N., Gong, Y. J., Zhan, Z. H., Zhang, J., Li, Y., and Tan, Y. S. (2015). An evolutionary algorithm with double-level archives for multiobjective optimization. *Ieee Transactions on Cybernetics, 45*(9), 1851-1863.

Cho, P. C., and Batta, R. (2020). UAV search path optimization for recording emerging targets. *Military Operations Research, in press*.

Colorado Geological Survey. (2011). *Official list of colorado fourteeners*. Retrieved November 11 from https://www.colorado.gov/pacific/sites/default/files/atoms/files/Fourteeners%20Official%20List%20A_0.pdf

Coutinho, W. P., Battarra, M., and Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering, 120*, 116-128.

Dasdemir, E., Köksalan, M., and Tezcaner Öztürk, D. (2020). A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. *Computers & Operations Research, 114*, 104811.

Deb, K., and Koksalan, M. (2010). Guest editorial special issue on preference-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation, 14*(5), 669-670.

Deb, K., and Kumar, A. (2007a). Interactive evolutionary multi-objective optimization and decision-making using reference direction method *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. London, England.

Deb, K., and Kumar, A. (2007b). Light beam search based multi-objective optimization using evolutionary algorithms. *2007 IEEE Congress on Evolutionary Computation*, Singapore, Singapore.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182-197.

Deb, K., and Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. Seattle, Washington, USA.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg (Eds.), *Evolutionary multiobjective optimization: Theoretical advances and applications* (pp. 105-145). Springer London.

Dell'Amico, M., Maffioli, F., and Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research, 2*(3), 297-308.

Ekici, A., Keskinocak, P., and Koenig, S. (2009). Multi-robot routing with linear decreasing rewards over time. *2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan.

Ekici, A., and Retharekar, A. (2013). Multiple agents maximum collection problem with time dependent rewards. *Computers & Industrial Engineering, 64*(4), 1009-1018.

Ergezer, H., and Leblebicioglu, K. (2013). Path planning for UAVs for maximum information collection. *IEEE Transactions on Aerospace and Electronic Systems, 49*(1), 502-520.

Ergezer, H., and Leblebicioglu, K. (2014). 3d path planning for multiple UAVs for maximum information collection. *Journal of Intelligent & Robotic Systems, 73*(1-4), 737-762.

Erkut, E., and Zhang, J. (1996). The maximum collection problem with time-dependent rewards. *Naval Research Logistics (NRL), 43*(5), 749-763.

Esri. (2021). Arcgis pro 2.4.2. In *Environmental Systems Research Institute*. http://www.esri.com

Evers, L., Barros, A. I., Monsuur, H., and Wagelmans, A. (2014). Online stochastic UAV mission planning with time windows and time-sensitive targets. *European Journal of Operational Research, 238*(1), 348-362.

Evers, L., Dollevoet, T., Barros, A. I., and Monsuur, H. (2014). Robust UAV mission planning. *Annals of Operations Research, 222*(1), 293-315.

Feillet, D., Dejax, P., and Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science, 39*(2), 188-205.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C-Emerging Technologies, 18*(5), 668-679.

Fonseca, C. M., and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation, 3*(1), 1-16.

Foo, J. L., Knutzon, J., Kalivarapu, V., Oliver, J., and Winer, E. (2009). Path planning of unmanned aerial vehicles using b-splines and particle swarm optimization. *Journal of Aerospace Computing, Information, and Communication, 6*(4), 271-290.

Fu, Z., Eglese, R., and Li, L. Y. O. (2008). A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society, 59*(5), 663-673.

Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL), 34*(3), 307-318.

Gudaitis, M. S. (1994). Multicriteria mission route planning using a parallel A* search. *Air Force Inst of Tech Wright-Patterson AFB OH School of Engineering*.

Guerriero, F., Surace, R., Loscri, V., and Natalizio, E. (2014). A multi-objective approach for unmanned aerial vehicle routing problem with soft time-windows constraints. *Applied Mathematical Modelling, 38*(3), 839-852.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research, 255*(2), 315-332.

Gurobi Optimization, L. (2021). Gurobi optimizer reference manual. In http://www.gurobi.com

Haimes, Y. Y., Lasdon, L., and Wismer, D. A. (1971). On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 296-297.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100-107.

Jaszkiewicz, A., and Słowiński, R. (1999). The 'light beam search' approach – an overview of methodology applications. *European Journal of Operational Research, 113*(2), 300-314.

Kataoka, S., and Morito, S. (1988). An algorithm for single constraint maximum collection problem. *Journal of the Operations Research Society of Japan, 31*(4), 515-531.

Keane, J. F., and Carr, S. S. (2013). A brief history of early unmanned aircraft. *Johns Hopkins APL Technical Digest, 32*(3), 558-571.

Korhonen, P., and Wallenius, J. (1988). A pareto race. *Naval Research Logistics (NRL), 35*(6), 615-623.

Köksalan, M., and Karahan, I. (2010). An interactive territory defining evolutionary algorithm: Itdea. *IEEE Transactions on Evolutionary Computation, 14*(5), 702-722.

Köksalan, M., and Lokman, B. (2009). Approximating the nondominated frontiers of multi-objective combinatorial optimization problems. *Naval Research Logistics, 56*(2), 191-198.

Köksalan, M., and Soylu, B. (2010). Bicriteria p-hub location problems and evolutionary algorithms. *Informs Journal on Computing, 22*(4), 528-542.

Köksalan, M., and Tezcaner Öztürk, D. (2017). An evolutionary approach to generalized biobjective traveling salesperson problem. *Computers & Operations Research, 79*, 304-313.

Köksalan, M. M. (1999). A heuristic approach to bicriteria scheduling. *Naval Research Logistics (NRL), 46*(7), 777-789.

Laporte, G., and Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics, 26*(2), 193-207.

Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., and Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review, 13*(2), 129-170.

Leifer, A. C., and Rosenwein, M. B. (1994). Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research, 73*(3), 517-523.

Malandraki, C., and Daskin, M. S. (1992). Time-dependent vehicle-routing problems - formulations, properties and heuristic algorithms. *Transportation Science, 26*(3), 185-200.

Mei, Y., Salim, F. D., and Li, X. D. (2016). Efficient meta-heuristics for the multi-objective time-dependent orienteering problem. *European Journal of Operational Research, 254*(2), 443-457.

Mittal, S., and Deb, K. (2007). Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms. *2007 IEEE Congress on Evolutionary Computation*,

Moskal, M. D., and Batta, R. (2017). A macrogrid approach for routing UAVs in support of information gathering. *Military Operations Research, 22*(4), 35-54.

Moskal, M. D., and Batta, R. (2019). Adaptive unmanned aerial vehicle surveillance using a prize-collecting vertex routing model. *Military Operations Research, 24*(4), 5-22.

Mufalli, F., Batta, R., and Nagi, R. (2012). Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Computers & Operations Research, 39*(11), 2787-2799.

Myers, D., Batta, R., and Karwan, M. (2016). A real-time network approach for including obstacles and flight dynamics in UAV route planning. *Journal of Defense Modeling and Simulation-Applications Methodology Technology-Jdms, 13*(3), 291-306.

Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., and Kostaras, A. N. (2003). Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 33*(6), 898-912.

Olsan, J. B. (1993). Genetic algorithms applied to a mission routing problem. *Air Force Inst of Tech Wright-Patterson AFB OH School of Engineering*

Peng, G. S., Dewil, R., Verbeeck, C., Gunawan, A., Xing, L. N., and Vansteenwegen, P. (2019). Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times. *Computers & Operations Research, 111*, 84-98.

Peng, G. S., Song, G. P., Xing, L. N., Gunawan, A., and Vansteenwegen, P. (2020). An exact algorithm for agile earth observation satellite scheduling with time-dependent profits. *Computers & Operations Research, 120*.

Pfeiffer, B., Batta, R., Klamroth, K., and Nagi, R. (2009). Path planning for UAVs in the presence of threat zones using probabilistic modelling. In A. B. Badiru and M. U. Thomas (Eds.), *Handbook of military industrial engineering*. CRC Press.

Phelps, S., and Köksalan, M. (2003). An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science, 49*(12), 1726-1738.

R Core Team. (2021). R: A language and environment for statistical computing. In *R Foundation for Statistical Computing*. http://www.R-project.org/

Sahinidis, N. V. (2017). Baron 17.8.9: Global optimization of mixed-integer nonlinear programs, user's manual. In *The Optimization Firm, LLC*. http://www.minlp.com/downloads/docs/baron%20manual.pdf.

Schilde, M., Doerner, K. F., Hartl, R. F., and Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence, 3*(3), 179-201.

Sindhya, K., Ruiz, A. B., and Miettinen, K. (2011). A preference based interactive evolutionary algorithm for multi-objective optimization: Pie. In R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, *Evolutionary Multi-Criterion Optimization* Berlin, Heidelberg.

Sylva, J., and Crema, A. (2007). A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research, 180*(3), 1011-1027.

Tawarmalani, M., and Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming, 103*(2), 225-249.

Tezcaner, D., and Köksalan, M. (2011). An interactive algorithm for multi-objective route planning. *Journal of Optimization Theory and Applications, 150*(2), 379-394.

Tezcaner Öztürk, D., and Köksalan, M. (2016). An interactive approach for biobjective integer programs under quasiconvex preference functions. *Annals of Operations Research, 244*(2), 677-696.

Tezcaner Öztürk, D., and Köksalan, M. (2018). Bi-objective UAV route planning in continuous terrain (technical report). *Middle East Technical University Ankara, Turkey*.

Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation, 17*(3), 411-436.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society, 35*(9), 797-809.

Türeci, H. (2017). Interactive approaches for bi-objective UAV route planning in continuous space. *Middle East Technical University*. *Ankara, Turkey*.

US Air Force. (2015). *Mq-1b predator*. Retrieved January 24, 2021 from http://www.af.mil/About-Us/Fact-Sheets/Display/Article/104469/mq-1b-predator/

Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research, 209*(1), 1-10.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., and Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research, 36*(12), 3281-3290.

Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, *Multiple Criteria Decision Making Theory and Application* Berlin, Heidelberg.

Xia, Y., Batta, R., and Nagi, R. (2017). Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment. *Operations Research, 65*(3), 674-692.

Yu, V. F., Jewpanya, P., Lin, S. W., and Redi, A. A. N. P. (2019). Team orienteering problem with time windows and time-dependent scores. *Computers & Industrial Engineering, 127*, 213-224.

Zheng, C., Li, L., Xu, F., Sun, F., and Ding, M. (2005). Evolutionary route planner for unmanned air vehicles. *IEEE Transactions on Robotics, 21*(4), 609-620.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation, 1*(1), 32-49.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 8*(2), 173-195.

APPENDICES

A. Radar Detection Threat Measure

Similar to Tezcaner and Köksalan (2011) we use the radar detection threat measure developed by Gudaitis (1994) to calculate the radar detection threat that the UAV is exposed to in the radar-effective regions. Basically, all radar detection probabilities the UAV is exposed to during the route are summed up. Following definitions and formulations are adapted from Gudaitis (1994) and Tezcaner Öztürk and Köksalan (2018). Let (x_i, y_i) and (x_j, y_j) be the coordinates of targets i and j, respectively. Suppose that there is a radar located between the targets at (x_r, y_r). The radar detection threat of the move between targets i and j, rdt_{ij}, is calculated using Equation (A.1):

$$rdt_{ij} = \int_{(x_i,y_i)}^{(x_j,y_j)} pd_{(x,y)} ds \tag{A.1}$$

Here, ds represents the infinitesimal movement of the UAV between two targets. The detection probability of a point (x, y), $pd_{(x,y)}$ is computed using Equation (A.2):

$$pd_{(x,y)} = \begin{cases} 1, & 10log_{10}(S/N_{(x,y)}) > UB_{(S/N)} \\ \dfrac{10log_{10}(S/N_{(x,y)}) - LB_{(S/N)}}{UB_{(S/N)} - LB_{(S/N)}}, & LB_{(S/N)} < 10log_{10}(S/N_{(x,y)}) \leq UB_{(S/N)} \\ 0, & 10log_{10}(S/N_{(x,y)}) \leq LB_{(S/N)} \end{cases} \tag{A.2}$$

We compute the signal-to-noise ratio (S/N) of any point (x, y) using Equation (A.3). $LB_{(S/N)}$ and $UB_{(S/N)}$ are radar parameters indicating the lower and upper bound in terms of (S/N) required for the detection. Due to the structure of $pd_{(x,y)}$, radar-effective regions have two layers in our problem. The core radar region is composed of points with $10log_{10}(S/N_{(x,y)})$ larger than $UB_{(S/N)}$, and hence their detection probability is 1. In the outer radar region, all points have $10log_{10}(S/N_{(x,y)})$ between $LB_{(S/N)}$ and $UB_{(S/N)}$, and their detection probabilities ranges

between 0 and 1. Outside the radar region, the detection probability is 0, since all points have smaller $10log_{10}(S/N_{(x,y)})$ than $LB_{(S/N)}$. We use the values 15 and 30 for $LB_{(S/N)}$ and $UB_{(S/N)}$, respectively, as Tezcaner Öztürk and Köksalan (2018).

$$S/N_{(x,y)} = \left(\frac{C}{((x-x_r)^2 + (y-y_r)^2)^2}\right), C = \frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_t^2} \quad \text{(A.3)}$$

P_t is the power transmitted by radar (watts), G_t is the power gain of transmitting antenna, λ is the wave length of signal frequency (meters), σ is the aircraft radar cross-section (RCS) (square meters), K is Boltzmann constant (joules/kelvin), T_s is the receiving system noise temperature (kelvin), B_n is the noise bandwidth of receiver (Hertz), L_t is the transmitting system loss. We use the values for these constants as in Tezcaner Öztürk and Köksalan (2018).

B. Nonlinear Mathematical Model to Find the Trajectories Used Between Consecutively Visited Targets

The following nonlinear mathematical models are modified version of the nonlinear mathematical model developed by Tezcaner Öztürk and Köksalan (2018).

B.1 Finding Trajectories Used Between Consecutively Visited Targets for a Given Tour for the Bi-Objective UAV Route Planning Problem

Tezcaner Öztürk and Köksalan (2018)'s nonlinear mathematical model finds efficient tours by determining the visiting order to the nodes and selecting trajectories from the efficient frontiers of the consecutively visited targets. The following nonlinear mathematical model selects approximately efficient trajectories to be used between consecutive target pairs for a given tour π, making sure that the summation of the distances (d_{ij}, between $(i,j) \in \pi$) of the selected trajectories equals at most the tour's given total distance value, D_π.

As the objective function, we minimize the summation of the radar detection threat values of the selected trajectories (rdt_{ij}, between $(i,j) \in \pi$). The model selects one point on the approximated Pareto-optimal frontier of each consecutive target pair. These constraints make the model nonlinear, as Type 2 and 3 frontiers are approximated by L_q functions.

Let E_π be the set of target pairs of tour π. The nonlinear model is run to find the trajectories used between each target pair (i,j) in E_π. Let E_π^k be the set of target pairs in tour π with movement type k, $k = 1,2,3$, and $E_\pi = E_\pi^1 \cup E_\pi^2 \cup E_\pi^3$. Then E_π^1 is the set of target pairs (i,j) in tour π that have Type 1 movement. In Type I movement, there is no radar region between targets i and j, and hence MDP (minimum distance point) that is the trajectory with the shortest distance between targets i and j is the only efficient trajectory. E_π^2 is the set of target pairs (i,j) in tour

π that have Type 2 movement. In Type 2 movement, the movement has a have curved Pareto-optimal frontier, where MDP (minimum distance point) and MRP (minimum radar point) are the two extreme trajectories. E_π^3 is the set of target pairs (i,j) in tour π that have Type 3 movement. In Type 3 movement, the movement has a two-piece Pareto-optimal frontier that consists of a linear and a curved part. MDP and TP (tangent point) are the extreme trajectories of the linear part, and TP and MRP are the extreme trajectories of the curved part. The scalars d_{ijt} and r_{ijt} denote the distance and radar detection threat objective values of trajectory XP of target pair (i,j), where $t \in \{MDP, TP, MRP\}$. The variables d_{ij} and r_{ij} denote the distance and radar detection threat objective values of the trajectory chosen by the model for target pair (i,j). The variables $z_{ijt}, t \in \{MDP, TP, MRP\}$ and $y_{ij1}, y_{ij2} \in \{0,1\}$ are dummy variables helping the model to make its selection from the linear or curved parts of the efficient frontiers of target pairs having type 3 movements.

$$Min \sum_{ij \in E_\pi} r_{ij} + \alpha \sum_{ij \in E_\pi} d_{ij} \quad \text{(B.1)}$$

$$\sum_{ij \in E_\pi} d_{ij} \leq D_\pi \quad \text{(B.2)}$$

$$d_{ij} = d_{ijMDP} \quad \forall (i,j) \in E_\pi^1 \quad \text{(B.3)}$$

$$r_{ij} = 0 \quad \forall (i,j) \in E_\pi^1 \quad \text{(B.4)}$$

$$d_{ij} \leq d_{ijMRP} \quad \forall (i,j) \in E_\pi^2 \quad \text{(B.5)}$$

$$d_{ij} \geq d_{ijMDP} \quad \forall (i,j) \in E_\pi^2 \quad \text{(B.6)}$$

$$r_{ij} \leq rdt_{ijMDP} \quad \forall (i,j) \in E_\pi^2 \quad \text{(B.7)}$$

$$r_{ij} \geq rdt_{ijMDP} - rdt_{ijMDP}\left(1 - \left(1 - \frac{d_{ij} - d_{ijMDP}}{d_{ijMRP} - d_{ijMDP}}\right)^{q_{ij}}\right)^{\frac{1}{q_{ij}}} \quad \forall i(i,j)j \in E_\pi^2 \quad \text{(B.8)}$$

$$d_{ij} = d_{ijMDP} * z_{ijMDP} + d_{ijTP} * z_{ijTP} + d_{ijMRP} * z_{ijMRP} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.9)}$$

$$z_{ijMDP} + z_{ijTP} + z_{ijMRP} = 1 \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.10)}$$

$$z_{ijMDP} \leq y_{ij1} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.11)}$$

$$z_{ijTP} \leq y_{ij1} + y_{ij2} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.12)}$$

$$z_{ijMRP} \leq y_{ij2} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.13)}$$

$$y_{ij1} + y_{ij2} = 1 \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.14)}$$

$$r_{ij} \leq rdt_{ijMDP} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.15)}$$

$$r_{ij} \geq \left(\frac{rdt_{ijMDP} - rdt_{ijTP}}{d_{ijMDP} - d_{ijTP}}\right) d_{ij} + c_{ij} - M(1 - y_{ij1}) \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.16)}$$

$$r_{ij} \geq rdt_{ijTP} - rdt_{ijTP}\left(1 - \left(1 - \frac{d_{ij} - d_{ijTP}}{d_{ijMRP} - d_{ijTP}}\right)^{q_{ij}}\right)^{\frac{1}{q_{ij}}} - M(1 - y_{ij2}) \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.17)}$$

$0 \leq z_{ijXP} \leq 1$ $\forall (i,j) \in E_{\pi}^{3}, \quad XP \in \{MDP, TP, MRP\}$ (B.18)

$y_{ij1}, y_{ij2} \in \{0,1\}$ $\forall (i,j) \in E_{\pi}^{3}$ (B.19)

$d_{ij}, r_{ij} \geq 0$ $\forall (i,j) \in E_{\pi}$ (B.20)

The objective function (B.1) minimize the summation of the radar detection threat values of the used trajectories (rdt_{ij}, between $(i,j) \in \pi$). In the second part of the objective function, summation of the distance values of the used trajectories (d_{ij}, between $(i,j) \in \pi$), which is multiplied by a small enough positive constant, is added in order to break ties if there are alternative optimal solutions with the same total radar detection threat. Constraint (B.2) makes sure that the summation of the first objective values (d_{ij}, between $(i,j) \in \pi$) of the trajectories equals at most the tour's total distance value, D_{π}. The remaining constraints select a trajectory between each target pair of tour π according to the movement type of the target pair. Constraints (B.3) and (B.4) select the efficient trajectory with the shortest distance for each target pair having type1 movements. Constraints (B.5) - (B.8) select a trajectory on the L_q curve of each target pairs having type 2 movement. Similarly, Constraints (B.9) - (B.19) select a trajectory for each target pair having type 3 movement. As the efficient frontiers of type 3 movements have two parts (a linear and curved parts), the constraints make sure that the trajectory is selected either on the linear or curved part.

B.2 Finding Trajectories Used Between Consecutively Visited Targets for a Given Tour for the Three-Objective UAV Route Planning Problem

The following nonlinear mathematical model is an extended version of the model presented in Appendix B.1 for the three-objective version of the UAV route planning problem. The model selects approximately efficient trajectories to be used between consecutive target pairs for a given tour π with known distance (D_{π}), radar detection threat (RDT_{π}), and information collection (I_{π}) objective values, making sure that the summation of the distance, radar detection threat and information collection

objectives of the selected trajectories are as close as possible to the objective values of tour π.

Let N_π and E_π be the sets of targets and target pairs of tour π, respectively. Let E_π^k be the set of target pairs in tour π with movement type k, $k = 1,2,3$, and $E_\pi = E_\pi^1 \cup E_\pi^2 \cup E_\pi^3$. The properties of these sets are discussed above in Appendix B.1. The scalars d_{ijXP} and r_{ijXP} denote the distance and radar detection threat objective values of trajectory XP of target pair (i,j), where $XP \in \{MDP, TP, MRP\}$. For each target j, $B_j = \{1, .. p, \dots, b\}$ represent the set of break points in the information collection structure, b_j^p be the time of the break point p, and g_j^p be the available information at the break point p. V is the constant flight speed of the UAV, and t_{ij} is the time required to traverse trajectory (i,j). The preceding target of target j in tour π is denoted by j^-. d_{ij}, rdt_{ij} are the decision variables defining the selected trajectory by the model for target pair (i,j), respectively. According to the selected trajectories, first t_j that is the arrival time at target j, and then I_j that is the amount of information collected at target j is calculated. All the remaining variables in the model are dummy variables allowing model to perform necessary calculations. The model is presented below:

$$Min\ \varepsilon + \alpha(\bar{D}_{dif} + \overline{RDT}_{dif} + \bar{I}_{dif}) \tag{B.21}$$

$$D_{dif} = \sum_{ij \in E_\pi} d_{ij} - D_\pi \tag{B.22}$$

$$RDT_{dif} = \sum_{ij \in E_\pi} r_{ij} - RDT_\pi \tag{B.23}$$

$$I_{dif} = I_\pi - \sum_{j \in N_\pi} I_j \tag{B.24}$$

$$\bar{D}_{dif} \leq \varepsilon \tag{B.25}$$

$$\overline{RDT}_{dif} \leq \varepsilon \tag{B.26}$$

$$\bar{I}_{dif} \leq \varepsilon \tag{B.27}$$

$$d_{ij} = d_{ijMDP} \qquad \forall (i,j) \in E_\pi^1 \tag{B.28}$$

$$r_{ij} = 0 \qquad \forall (i,j) \in E_\pi^1 \tag{B.29}$$

$$d_{ij} \leq d_{ijMRP} \qquad \forall (i,j) \in E_\pi^2 \tag{B.30}$$

$$d_{ij} \geq d_{ijMDP} \qquad \forall (i,j) \in E_\pi^2 \tag{B.31}$$

$$r_{ij} \leq rdt_{ijMDP} \qquad \forall (i,j) \in E_\pi^2 \tag{B.32}$$

$$r_{ij} \geq rdt_{ijMDP} - rdt_{ijMDP}\left(1-\left(1-\frac{d_{ij}-d_{ijMDP}}{d_{ijMRP}-d_{ijMDP}}\right)^{q_{ij}}\right)^{\frac{1}{q_{ij}}} \quad \forall (i,j) \in E_\pi^2 \quad \text{(B.33)}$$

$$d_{ij} = d_{ijMDP} * z_{ijMDP} + d_{ijTP} * z_{ijTP} + d_{ijMRP} * z_{ijMRP} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.34)}$$

$$z_{ijMDP} + z_{ijTP} + z_{ijMRP} = 1 \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.35)}$$

$$z_{ijMDP} \leq y_{ij1} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.36)}$$

$$z_{ijTP} \leq y_{ij1} + y_{ij2} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.37)}$$

$$z_{ijMRP} \leq y_{ij2} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.38)}$$

$$y_{ij1} + y_{ij2} = 1 \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.39)}$$

$$r_{ij} \leq rdt_{ijMDP} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.40)}$$

$$r_{ij} \geq \left(\frac{rdt_{ijMDP} - rdt_{ijTP}}{d_{ijMDP} - d_{ijTP}}\right) d_{ij} + c_{ij} - M(1-y_{ij1}) \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.41)}$$

$$r_{ij} \geq rdt_{ijTP} - rdt_{ijTP}\left(1-\left(1-\frac{d_{ij}-d_{ijTP}}{d_{ijMRP}-d_{ijTP}}\right)^{q_{ij}}\right)^{\frac{1}{q_{ij}}} - M(1-y_{ij2}) \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.42)}$$

$$t_{j^-j} = \frac{d_{j^-j}}{V} \quad \forall (i,j) \in E_\pi \quad \text{(B.43)}$$

$$t_j = t_{j^-} + t_{j^-j} \quad \forall j \in N_\pi \quad \text{(B.44)}$$

$$t_1 = 0 \quad \text{(B.45)}$$

$$t_j = \sum_{p \in B_j} b_j^p w_j^p \quad \forall j \in N_\pi \quad \text{(B.46)}$$

$$I_j = \sum_{p \in B_j} g_j^p w_j^p \quad \forall j \in N_\pi \quad \text{(B.47)}$$

$$w_j^1 \leq q_j^1 \quad \forall j \in N_\pi \quad \text{(B.48)}$$

$$w_j^p \leq q_j^{p-1} + q_j^p \quad \forall p \in B_j \setminus \{1,b\}, \forall j \in N_\pi \quad \text{(B.49)}$$

$$w_j^b \leq q_j^{b-1} \quad \forall j \in N_\pi \quad \text{(B.50)}$$

$$\sum_{p \in B_j} w_j^p = 1 \quad \forall j \in N_\pi \quad \text{(B.51)}$$

$$\sum_{p \in B_j \setminus \{b\}} q_j^p = 1 \quad \forall j \in N_\pi \quad \text{(B.52)}$$

$$w_j^p \geq 0 \quad \forall p \in B_j, \quad \forall j \in N_\pi \quad \text{(B.53)}$$

$$q_j^p \in \{0,1\} \quad \forall p \in B_j \setminus \{b\}, \quad \forall j \in N_\pi \quad \text{(B.54)}$$

$$0 \leq z_{ijXP} \leq 1 \quad \forall (i,j) \in E_\pi^3, \quad XP \in \{MDP, TP, MRP\} \quad \text{(B.55)}$$

$$y_{ij1}, y_{ij2} \in \{0,1\} \quad \forall (i,j) \in E_\pi^3 \quad \text{(B.56)}$$

$$d_{ij}, r_{ij} \geq 0 \quad \forall (i,j) \in E_\pi \quad \text{(B.57)}$$

A pseudo variable, ε, is minimized mainly in objective function (B.21). This minimizes the maximum difference among the normalized differences between the summation of the objective function values of the selected trajectories and the objective function values of the given tour π (see constraints (B.22) - (B.27)). The purpose is selecting trajectories on the approximated Pareto-optimal frontiers of the target pairs of tour π in a way that the the summations of the objectives of the selected

trajectories are as close as possible to the objective function values of the given tour. In the objective function, the second part that is multiplied with a very small positive constant is to make sure that Pareto-optimal solutions are selected when there are alternative optimal solutions with the same ε value.

Constraints (B.28) and (B.29) select the efficient trajectory with the shortest distance for each target pair having type1 movements. Constraints (B.30) - (B.33) select a trajectory on the L_q curve of each target pairs having type 2 movement. Similarly, Constraints (B.34) - (B.42) together with (B.55) and (B.56) select a trajectory for each target pair having type 3 movement. As the efficient frontiers of type 3 movements have two parts (a linear and curved parts), the constraints make sure that the trajectory is selected either on the linear or curved part.

Constraints (B.43) - (B.54) first determines the arrival times to targets using the flight duration information of the selected trajectories, then use arrival times to define the information gain from the targets. The remaining constraints define the remaining decision variables.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	: Daşdemir, Erdi
Nationality	: Turkish (TC)
Date and Place of Birth	: 21 May 1990, Kars
Marital Status	: Married
Phone	: +905068197911
E-mail	: edasdemir@hacettepe.edu.tr, erdidasdemir@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
PhD	METU Industrial Engineering	2021
MS	Hacettepe U. Industrial Engineering	2015
BS	Hacettepe U. Industrial Engineering	2013
High School	Bartin Davut Fırıncıoğlu High School	2008

WORK EXPERIENCE

Year	Place	Enrollment
2013-Present	Hacettepe U. Industrial Engineering	Research Assistant
2019-2020	State University of New York at Buffalo	Visiting Scholar
2012-2013	BNB Consulting, Cyber Park, Ankara	Project Dev. Eng.

JOURNAL PUBLICATIONS

1. Dasdemir, E., Köksalan M., Tezcaner Öztürk, D. (2020). A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. *Computers & Operations Research, 114,* 104811.
2. Dasdemir, S., Dasdemir, E. (2019) Nesnelerin interneti ve RFID teknolojisi ile yeni nesil tedarik zinciri ve stok denetimi. *Vergi Raporu (ULAKBIM), 236*, 233-253.
3. Aydın, O., Shaygan, A. Dasdemir, E., Soydan G. (2017). Selecting health care improvement projects: A methodology integrating cause-and-effect diagram and analytical hierarchy process. *Quality Management in Health Care, 26*(1), 40-48.
4. Dasdemir, E., Weiss, C., Testik, M. and Knoth, S. (2016). Evaluation of Phase I analysis scenarios on Phase II performance of control charts for autocorrelated observations. *Quality Engineering, 28*(3), 293-304.

papers in process

5. Dasdemir, E. Testik, M.C., Tezcaner Öztürk, D., Tuncer Sakar, C., Guleryuz G., Aydin O. M. (under review). A multi-objective open vehicle routing problem with overbooking: Exact and heuristic solution approaches for an employee transportation problem. *OMEGA*
6. Moskal M.D., Dasdemir E., Batta R. (under review). Unmanned aerial vehicle information collection missions with uncertain characteristics. *EJOR*.
7. Dasdemir E., Batta R., Koksalan M., Tezcaner Öztürk, D. (under review). UAV routing for reconnaissance mission: a multi-objective orienteering problem with time-dependent prizes and multiple connections. *Computers & OR.*
8. Akduran Y., Dasdemir E., Testik M.C. (in submission). An approach to search efficient settings for the operators and parameters of genetic algorithms.

RESEARCH PROJECTS

1. Bi-objective UAV route planning in continuous space (1 October 2015 – 31 March 2017), Air Force Office of Scientific Research, Air Force Material Command, USAF, Award No: FA9550-16-1-0005, Principal Investigator: Prof. Dr. Murat KÖKSALAN, Senior Researcher: Dr. Diclehan Tezcaner Öztürk, Researchers: Erdi Dasdemir, Hannan Tureci.
2. Minimizing the financial losses of Hacettepe University hospitals: Improving hospital billing process and reducing billing errors, (2012 – 2013) TÜBİTAK, 2209-A Sanayi Entegre Proje Destek Programı, Award No: 1919B011202200, Principal Investigator: Erdi Daşdemir, Researchers: Murat Atalay, Volkan Bilgin, Macit Mete Oğuz.

CONFERENCES

1. Dasdemir, E., Tezcaner Öztürk D., Koksalan, M. Sürekli arazide hareket modellemesi ile insansız hava araçları için çok amaçlı rotalama, *YAEM 2021*, Online Conference, 4-7 July 2021.
2. Dasdemir, E., Batta, R., Köksalan M., Tezcaner Öztürk D. UAV routing to maximize information gain in a monitored environment with time and radar restrictions. *INFORMS Annual Meeting 2019*, Seattle, Washington, USA, 13-16 November, 2019.
3. Dasdemir, E., Tezcaner Öztürk D., Köksalan, M. A preference-based multi-objective evolutionary algorithm-based solution approach to bi and three-objective UAV route planning problems in continuous space. *MCDM 2019*, Istanbul, Turkey, 16-21 June 2019.

4. Dasdemir, E., Batta, R., Tezcaner Öztürk D., Köksalan, M. İnsansız hava araçları için çok amaçlı rotalama. *YAEM 2019*, Ankara, Turkey, 12-14 June 2018.
5. Dasdemir, E., Tezcaner Öztürk D., Köksalan, M. Esnek zaman aralıklı çok kriterli insansız hava aracı rota planlama problemi. *YAEM 2018*, Eskişehir, Turkey, 26-29 June 2018.
6. Dasdemir, E., Testik, M.C., Testik, O.M., Tuncer Sakar, C. A multi-objective evolutionary algorithm approach to employee bus transportation problem. *ICATCES'18*, Safranbolu, Turkey, 11-13 May 2018.
7. Dasdemir, E., Tezcaner Öztürk D., Köksalan, M. Çok kriterli insansız hava aracı rota planlama problemi için bir evrimsel algoritma. *YAEM 2017*, İstanbul, Turkey, 5-7 July 2017.
8. Dasdemir, E., Tezcaner Öztürk D., Köksalan, M. A preference-based evolutionary algorithm and implementation on UAV route planning in continuous space. *MCDM 2017*, Ottawa, Canada, 9-14 July 2017.
9. Dasdemir, E., Tezcaner Öztürk D., Köksalan M. A preference-based evolutionary algorithm for bi-objective UAV route planning in continuous space. *INFORMS Annual Meeting 2016*, Nashville, Tennessee, USA.
10. Dasdemir, E., Tezcaner Öztürk D., Köksalan M. A preference-based multi-objective evolutionary algorithm for bi-objective UAV route planning in continuous space. *Poster Presentation, 12th MCDA/M Summer School 2016*, Recife, Pernambuco, Brazil, 18-29 July 2016.
11. Dasdemir, E., Weiss, C., Testik, M., and Knoth, S. Phase I analysis of autocorrelated time series data. *European Network for Business and Industrial Statistics (ENBIS) 14 Conference*, Linz/Austria, 21-25 September 2014.
12. Aydın, O., Shaygan, A. and Dasdemir, E. A cause and effect diagram and AHP based methodology for selection of quality improvement projects. *European Network for Business and Industrial Statistics (ENBIS) 14 Conference*, Linz/Austria, 21-25 September 2014.
13. Sonmez, V., Dasdemir, E., Testik, M. and Aydin, O. A simulation-based decision support tool for hospital bed capacity planning. *Proceedings of Global Conference on Healthcare Systems Engineering (GCHSE) 2014*, Istanbul/Turkey, 5-8 August, 2014.
14. Dasdemir, E., Oguz, M., Atalay, M., Bilgin, V., Testik, M. and Soydan, G. Improving hospital billing processes for reducing costs of billing errors. *European Network for Business and Industrial Statistics (ENBIS) 13 Conference*, Ankara/Turkey, 15-19 September 2013.

HOBBIES

Camping, Trekking and Hiking, Ski and Snowboard.