INTERACTIVE APPROACHES FOR BI-OBJECTIVE UAV ROUTE PLANNING IN CONTINUOUS SPACE

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

BY

HANNAN TÜRECİ

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

FEBRUARY 2017

Approval of the thesis:

INTERACTIVE APPROACHES FOR BI-OBJECTIVE UAV ROUTE PLANNING IN CONTINUOUS SPACE

submitted by **HANNAN TÜRECİ** in partial fulfillment of the requirement for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Murat Köksalan
Head of Department, **Industrial Engineering**

Prof. Dr. Murat Köksalan
Supervisor, **Industrial Engineering Dept., METU**

Assist. Prof. Dr. Diclehan Tezcaner Öztürk
Co-Supervisor , **Industrial Engineering Dept., TEDU**

Examining Committee Members

Prof. Dr. Yasemin Serin
Industrial Engineering Dept., METU

Prof. Dr. Murat Köksalan
Industrial Engineering Dept., METU

Assist. Prof. Dr. Diclehan Tezcaner Öztürk
Industrial Engineering Dept., TEDU

Prof. Dr. Meral Azizoğlu
Industrial Engineering Dept., METU

Assist. Prof. Dr. Mustafa Kemal Tural
Industrial Engineering Dept., METU

Date: **09.02.2017**

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 : Hannan TÜRECİ

Signature :

ABSTRACT

INTERACTIVE APPROACHES FOR BI-OBJECTIVE UAV ROUTE PLANNING IN CONTINUOUS SPACE

TÜRECİ, Hannan

M.S., Department of Industrial Engineering
Supervisor : Prof. Dr. Murat Köksalan
Co-Supervisor : Assist. Prof. Dr. Diclehan Tezcaner Öztürk

February 2017, 69 pages

We study the route planning problem of unmanned air vehicles (UAVs). We consider two objectives; minimizing total distance traveled and minimizing total radar detection threat since these objectives cover most of the other related factors. We consider routing in a two-dimensional continuous terrain, in which we have infinitely many efficient trajectories between target pairs.

We develop interactive algorithms that find the most preferred solution of a route planner (RP), who has either of the underlying preference function structures: linear or quasiconvex. To implement the algorithms to route planning problems, we use approximated nondominated frontiers of the trajectories between targets. In the linear case, we search for supported efficient solutions in two stages. In the first stage, we find the best trajectory between each target pair. In the second stage, we find the tour visiting all targets (traveling salesperson problem, TSP). In the quasiconvex case, we search for both supported and unsupported efficient solutions. We first reduce the objective space to rectangular regions around at most three supported efficient solutions. We then search inside these rectangular regions to find supported/unsupported efficient solutions and narrow our search region. We proceed with pairwise comparisons from the RP and reduce our search space until the two solutions to be compared are close enough. To generate random problem instances, we develop a mathematical model that randomly locates radars in a terrain with known target locations. We then demonstrate the interactive algorithm developed for linear preference functions on two randomly generated problems.

Keywords: Bi-objective Routing, Interactive, Routing in Continuous Space, Unmanned Air Vehicles, UAV Route Planning

ÖZ

SÜREKLİ UZAYDA İKİ AMAÇLI İHA GÜZERGAH PLANLAMASINA İLİŞKİN ETKİLEŞİMLİ YAKLAŞIMLAR

TÜRECİ, Hannan

Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Murat Köksalan
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Diclehan Tezcaner Öztürk

Şubat 2017, 69 Sayfa

Bu tezde, İnsansız Hava Aracı (UAV) rota planlama problemlerini ele alıyoruz. İlgili faktörlerin çoğunu kapsamalarından dolayı toplam mesafeyi en aza indirmeyi ve toplam radara algılanma tehdidini en aza indirmeyi amaçlarımız olarak belirledik. Rotalamayı iki boyutlu sürekli hareket alanında yapıyoruz,dolayısıyla, hedefler arasında sonsuz sayıda etkin güzargah bulunduğunu biliyoruz.

İki temel tercih fonksiyonu (doğrusal ve konveks benzeri) için rota planlayıcısı'nın (RP) en çok tercih ettiği çözümü bulan etkileşimli algoritmalar geliştirdik. Rota planlama problemine algoritmaları uygulayabilmek için hedefler arasındaki etkin yolları tahmin eden yöntemleri kullandık. Doğrusal durumda desteklenen çözümler bulmak için, rota planlama problemini iki kısımda çözdük. İlk kısımda her hedef ikilisi arasında en iyi yolu bulduk. İkinci kısımda tüm hedefleri gezen turu bulduk (gezgin satıcı problemi). Konveks benzeri fonksiyonlar için geliştirilen algoritmada hem desteklenen hem desteklenmeyen etkin çözümleri arıyoruz. Öncelikle RP'nin en çok tercih ettiği çözümü içinde bulunduran dikdörtgen alanlar elde ediyoruz. Sonrasında bu bölgelerin içinde yeni çözüm (desteklenen/desteklenmeyen) arıyoruz. Algoritmalara RP'nin iki alternatif arasındaki tercihi ile ilerliyoruz ve karşılaştırılacak çözümler yeterince yakın olunca algoritmaları durduruyoruz. Rastgele problemler oluşturmak için, belli bir alana radarlar yerleştiren bir model geliştirdik. Çalışmanın sonunda, doğrusal fonksiyonlar için geliştirdiğimiz algoritmayı rastgele oluşturulmuş iki problem üzerinde gösteriyoruz.

Anahtar Kelimeler: İki amaçlı rotalama, İnteraktif, Sürekli Uzayda Rotalama,
İnsansız Hava Aracı, İHA Rota Planlama

ACKNOWLEDGEMENTS

I wish to thank my supervisors Prof. Dr. Murat Köksalan and Assist. Prof. Dr. Diclehan Tezcaner Öztürk for their valuable contributions and guidance during this study. I am also very grateful for their understanding and support. It is a great honor for me to work with such brilliant supervisors.

I would like to thank Air Force Office of Scientific Research for their funding. This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF under Award No. FA9550-16-1-0005. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Air Force Office of Scientific Research, Air Force Materiel Command, USAF.

To my father…

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Unmanned air vehicles (UAVs) are unpiloted aircrafts that were originally designed for military purposes. Currently, these vehicles also serve civilian purposes in environments such as surveillance against crimes and minimizing hazardous effects of natural disasters. Route planning or a mission for these vehicles involves finding the path that the UAV follows in a terrain visiting all the predetermined target points. Several objectives may be of interest in the route planning problem. Minimizing distance traveled, fuel consumption, flight duration, detection threat, and maximizing navigation performance are some of the meaningful objectives.

In this project, we consider multi-objective route planning of UAVs in a two-dimensional continuous terrain. We consider two objectives: minimizing total distance traveled and minimizing total radar detection threat since they capture the main concerns. In a two dimensional continuous terrain, UAV can travel through infinitely many path alternatives.

Tezcaner Öztürk (2013) considers multi-objective route planning problems as a combination of multi-objective Shortest Path Problem (MOSPP) and multi-objective Travelling Salesman Problem (MOTSP) for the discretized problems. She approximated efficient paths between targets for the continuous terrains in her study. In the continuous case, we do not have MOSPP between target pairs since we have infinitely many nodes between targets. Therefore, we do not have standard SPP between targets. However, we again find the best trajectory between targets by solving the approximated frontiers for a desired linear function. In this thesis, we use her findings and develop interactive algorithms for bi-objective route planning problem of UAVs. We consider two underlying preference function structures: linear and quasiconvex preference functions.

In the rest of this study, we first present the background on UAV route planning in Chapter 2. In Chapter 3, we explain the UAV route planning problems in terms of problem terrain, UAV movements and objectives. In Chapter 4 and 5, we explain our interactive algorithms for linear and quasiconvex preference functions, respectively. We develop a mathematical model to randomly generate a problem instance for demonstration purposes. In Chapter 6, we give this model and demonstrate our interactive algorithm developed for linear functions on two example problems. We give our conclusions in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

In the literature, there are many studies on the route planning problem of UAVs considering multiple objectives. The past work on UAV route planning typically addresses routing UAVs between an initial point and a destination. They consider a linear aggregation of objectives (hence convert the problem into a single-objective problem) and employ a heuristic approach to determine the route. Zheng et al. (2003) consider the routing problem of a UAV between two targets in three dimensional space. They develop an evolutionary algorithm (EA) that finds near-optimal routes by using the weighted combination of objectives. They consider three objectives; route length, average altitude of the route above the sea level, and closeness to threat zones. Another study, Foo et al. (2009) again consider three objectives (target reconnaissance, total distance traveled and safety) for the route planning of UAVs in three dimensional space. They use particle swarm optimization and b-splines to generate altenative paths by changing the bias of objectives in the weighted combination of objectives. Then, they present alternative paths to the Decision Maker (DM).

Zheng et al. (2005) develop EA for 3-D route planning of UAVs between two targets by considering some constraints related with UAVs separately. They consider single and multiple vehicles. Pohl and Lamont (2008) also consider multiple vehicles and develop an EA without aggregating the objectives.

Many of the past work have treated the terrain the UAV flies through as a discretized terrain, approximating it by a grid structure. The early examples are Olsan (1993) and Gudaitis (1994). They consider route planning problem for a single UAV visiting a single target. They also linearly combine two objectives which are to be minimized; route length and radar exposure, and optimize this single composite objective using heuristics. In Yavuz’s (2002) study, the UAV visits multiple targets optimizing the same two objectives. He develops an approach that is the synthesis of Particle Swarm Optimization and Ant System, to find a good solution.

In practice, the terrain the UAV travels through is a continuous terrain. However, the literature on UAV route planning problem in continuous terrain is scarce, since even two conflicting objectives in a continuous terrain lead to infinitely many efficient solutions, and finding these efficient solutions is computationally demanding. The studies of Pachter and Hebert (2002) and Kan et al. (2011) can be considered as relevant studies for routing in continuous terrain. Pachter and Hebert (2002) find the best trajectory between two locations that minimizes the total detection for a given distance that the vehicle can move. Kan et al. (2011) consider the same problem with the same objectives. They get threshold altitude as an input and generate safe paths by using B-splines.

As explained above, most of the previous studies dealt with a simplified linear aggregation of the two-objective version of the problem; converting it into a single-objective problem. Using a linear aggregation of the objectives limits the solutions that can be found to a subset of the efficient solutions. An exception to linearly aggregating the objectives is the study of Tezcaner and Köksalan (2011), where they consider two objectives separately in a problem environment with multiple target points. They develop an interactive algorithm that finds the most preferred solution of a decision maker (DM) with an underlying linear preference function. Tezcaner Öztürk and Köksalan (2016) also consider the same two objectives separately. They develop an interactive algorithm to find the most preferred solution of a DM with an underlying quasiconvex preference function. In both of these studies, they use a discretized terrain. Later, Tezcaner Öztürk (2013) considered the route planning problem in continuous terrain for the same two objectives and developed solution methods to generate the efficient solutions, both by exact and heuristic methods. In our study, we also work on the route planning problem of UAVs in continuous terrain optimizing the same two objectives. Under the presence of a DM, it is not meaningful to generate all efficient solutions, since the DM would be interested in solutions satisfying a certain tradeoff between the objectives. Therefore, instead of generating all efficient solutions, we focus our search on the most preferred solutions of the decision maker. We develop two interactive algorithms that approximate the most preferred solution of a decision maker. Our first algorithm assumes that the DM has an underlying linear preference structure. Our second algorithm addresses a more general case, where the DM has an underlying quasiconvex preference function.

CHAPTER 3

PROBLEM DEFINITION

In this chapter, we explain route planning problem of UAVs. We give an example problem terrain to visualize the problem that we consider in this thesis. We give information about radars, objectives of route planning problem, and UAV movements in 2-D continuous space. At the end of the chapter, we present the formulation of the bi-objective UAV route planning problem, developed by Tezcaner Öztürk (2013). Before these, we give some definitions and the problem definition in Section 3.2.

3.1 Definitions

We give some definitions that we commonly use in this thesis. We adapted them from Tezcaner Öztürk and Köksalan (2016).

Let x be the decision variable vector in X, where X is the feasible set. The image of feasible set in objective function space is denoted with Z. Assume that there are p objectives to minimize and $z(x) = \left(z_1(x), z_2(x), \dots, z_p(x)\right)$ is the objective function vector of x.

Definition 3.1 A solution $\mathrm{x} \in \mathrm{X}$ is efficient if there does not exist $x' \in \mathrm{X}$ such that $z_k(x') \leq z_k(x)$ for $\mathrm{k} = 1,2, \dots, \mathrm{p}$ and $z_k(x') < z_k(x)$ for at least one objective. If there is such an x', x is said to be inefficient. All efficient solutions constitute the efficient frontier (set).

Definition 3.2 If a solution $x \in X$ is efficient, then $z(x)$ is said to be nondominated, and if x is inefficient, and then $z(x)$ is said to be dominated. All nondominated points constitute the nondominated frontier (set).

Definition 3.3 A nondominated solution $z(x)$ is a supported nondominated solution iff there exists a positive linear combination of objectives minimized by x. If $z(x)$ is a supported nondominated solution then x is supported efficient solution. Otherwise, $z(x)$ is an unsupported nondominated solution and x is supported efficient solution.

Definition 3.4 An extreme nondominated point $z(x)$ is a supported nondominated point that has the minimum possible value in at least one of the objectives.

Definition 3.5 If $X \subset Z^n$ and Z_{ND} is the set of all nondominated points, let $T = \{t|z(x^t) \in Z_{ND}\}$ and $z(x^i) \in Z_{ND}$ be a supported nondominated point. x^i is said to be adjacent efficient to x^j iff there does not exist x^t such that $\sum_{t\neq j} \mu_t z(x^t) \leq \lambda z(x^j) + (1-\lambda)z(x^i)$ where $\sum_{t\neq j} \mu_t = 1, 0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$.

3.2 UAV Route Planning Problem

UAVs travel through a continuous terrain visiting a number of target points. We consider a two-dimensional terrain and assume that the vehicle travels with a constant altitude. An example terrain structure can be seen in Figure 3.1. The vehicle is required to visit all five targets (triangles) in the figure, and the objectives are to minimize distance traveled and radar detection threat at radar-covered territories (circular regions). The radar is less effective towards the circumferences of the circular regions and ineffective in the white regions. We show several of the infinitely many efficient tours with dashed lines in Figure 3.1.

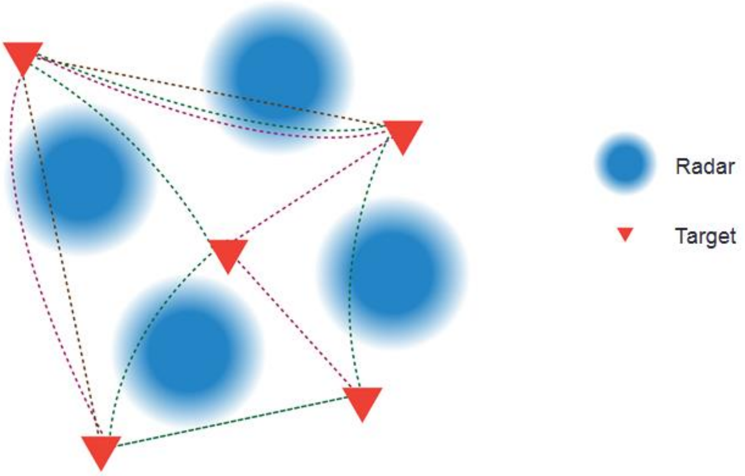


Figure 3.1 Terrain Representation and Efficient Solutions

In this study, we make some assumptions about the problem terrain. As it is illustrated in the figure, radar regions do not intersect with each other and targets are located outside the radar regions. Also, for the sake of simplicity, we assume that there is only one effective radar region between each target pair, and if there is more than one radar region between a pair of targets, we select the most threatening radar region. We use radar detection threat measures for this selection. We select the radar region that gives the higher radar detection threat value over the straight path between a target pair. Therefore, we somewhat underestimate the radar detection threat if traveled trajectory goes inside more than one radar region.

3.2.1 Objectives

There are many factors affecting the route selection of UAVs. Weather, terrain conditions, traveled distance, time limitations, min-max altitude, fuel consumption/ refueling points and risk of detection to enemy threats can be given as examples (Bahnij,1985). In the literature related with UAV route planning problems, generally two or three objectives are used. We consider two objectives that cover most of the related factors in our study: minimizing total distance traveled and minimizing radar detection threat.

The first objective is measured with the length of the path the UAV follows. For the second objective, we use the radar exposure measure that is developed by Gudaitis (1994). This measure sums up all radar detection probabilities on the path of the UAV. It is equivalent to approximating how long the UAV is exposed to each detection probability. More details on the calculation of these objectives can be found in Appendix A.

The radar is located at the center of the radar region, and it is ineffective in detecting the UAV outside its region. Inside the radar region, we have two parts: (i) inner region where the detection probability is 1, (ii) outer region where the detection probability ranges between 0 and 1. These two regions can be seen in Figure 3.2. The detection probability reduces from 1 to 0 as we move from the inner radar region towards the circumference of the outer radar region.

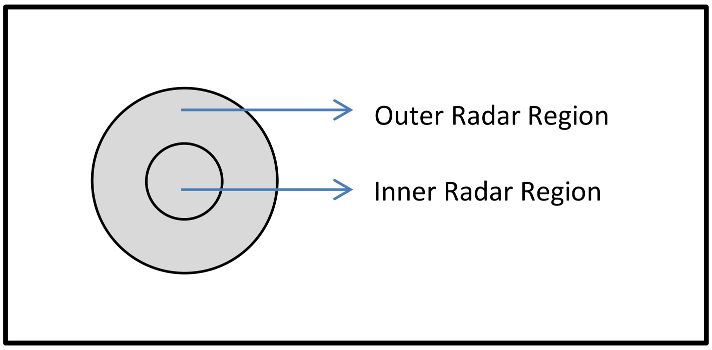


Figure 3.2 Radar Region

3.2.2 Movement of UAV in the Continuous Terrain

In the route planning problem for UAVs, there are infinitely many path options for visiting a number of targets. Given that we have T targets, the UAV can visit the targets in $(T-1)!/2$ different orders (assuming that the terrain is symmetrical; going from target i to target j has the same objective values as going from j to i). If there are E efficient trajectories between each target pair, each order can be visited in E^T different combinations. In total we have $E^T(T-1)!/2$ different path alternatives. In the route planning problem in continuous terrain, there can be infinitely many trajectories between any two targets, and thus we have infinitely many path alternatives to visit all targets.

In our solution approach, we decompose the overall route planning problem into two parts. In the first part, we determine the efficient trajectories between consecutive target pairs. In the second part, we determine the visiting order to the targets. First problem is a multi-objective shortest path problem between each target pair, and the second problem is a multi-objective traveling salesperson problem (MOTSP) with multiple efficient edges between node pairs. Tezcaner and Köksalan (2011) refer this problem as generalized MOTSP.

To find the efficient trajectories between target pairs, we use the findings of Tezcaner Öztürk (2013). We next explain the movement types between two targets, and the structure of their nondominated frontiers.

3.2.3 Movement of the UAV between Two Targets

The two objectives conflict with each other only inside the radar regions since the radar is ineffective outside the radar regions. In the ineffective regions, we only minimize the total distance traveled. The UAV should follow the shortest distance between two points in the ineffective regions, which is the Euclidean distance (the straight line connecting two points). Inside the radar regions, Tezcaner Öztürk (2013) assumes that the UAV makes a circular move. To classify the moves between two targets, we first find the extreme efficient solution with smallest distance traveled. This path is on the straight line that connects two target points. If this path does not pass through any radar region, we classify this path as Type I, and conclude that there is only one efficient solution between these two targets. If the path with the smallest distance passes through only the outer radar region, we classify it as Type II, and if the path passes through both the inner and the outer radar regions, we classify it as Type III. The moves of the UAV between two targets can follow one of the three types. For each movement type, the calculation of the objectives is given below. To use the formulas presented below, we need to adjust the coordinates of the targets and the effective radar between the two targets. First, all coordinates are rearranged so that the radar center is located at the origin. A straight line connecting the initial and the final target should intersect with the y-axis where the intersection point's y-coordinate is positive. Moreover, the angle between x-axis and the straight line should be in $[-\pi/4, \pi/4]$ range. If these conditions are not satisfied, targets are rotated around the origin for $\pi/2$ angle increments until all conditions are satisfied. For detailed information on terrain transformation, please see Tezcaner Öztürk's (2013) study.

Formulas and explanations for each movement type are taken directly from Tezcaner Öztürk (2013). Figures are also drawn similar to the ones present in her study. We give more details on the movement types below.

Type 1. No Intersection with the Radar Region (Figure 3.3)

In this type of move, the shortest distance between two targets does not pass through any radar region. We therefore have a single efficient solution with total distance, D, and total radar detection threat, $RDT = 0$. Assuming that the target with the smaller x-coordinate is located at coordinates (x_s, y_s), and the other target located at coordinates (x_f, y_f), we calculate the total distance (D) with equation (3.1).

$$D = \sqrt{\left(x_f - x_s\right)^2 + \left(y_f - y_s\right)^2} \qquad (3.1)$$

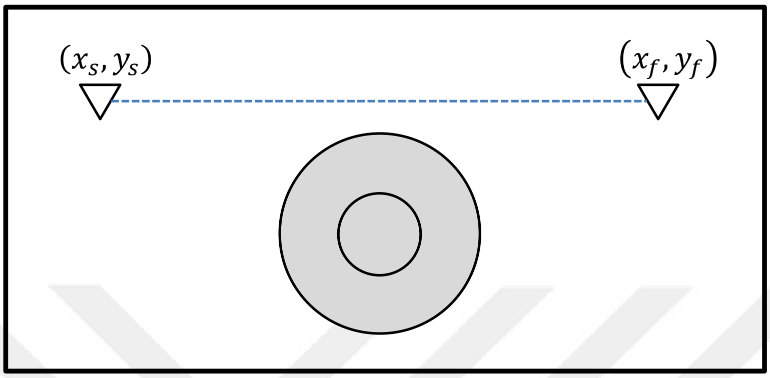


Figure 3.3 Movement between Radar Regions – Type 1

Type 2. Moves through only the Outer Radar Region (Figure 3.4)

For this case, all efficient trajectories of the UAV pass through only the outer radar region. The trajectories that pass through the inner radar region result in longer paths with higher detection threats, and are therefore inefficient. As in Tezcaner Öztürk (2013), we assume that the vehicle makes a move inside the outer radar region as if it is moving on a circle, $(x - a)^2 + (y - b)^2 = r^2$ centered at (a, b) with radius r. For further information, please see Appendix B. The distance and radar detection threat measure are calculated with equations (3.2) and (3.3), respectively.

$$D = \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2.arcsin\left(\frac{\sqrt{(x_{ex}-x_{en})^2+(y_{ex}-y_{en})^2}}{2r}\right).r \; + \sqrt{\left(x_f - x_{ex}\right)^2 + \left(y_f - y_{ex}\right)^2} \qquad (3.2)$$

Here, the first and the last terms are the Euclidean distances corresponding to the straight paths outside the radar regions. First term is the direct distance between initial point (x_s, y_s) and entrance point to the outer radar region (x_{en}, y_{en}). Likewise, the last term is the distance of the straight path between exit point from the outer radar region (x_{ex}, y_{ex}) and destination point (x_f, y_f). The middle term is the length of the arc traveled inside the outer radar region. It gives the circular distance between entrance and exit points.

$$RDT = \int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N}-LB_{S/N}} log_{10} \left(\frac{C}{\left(x^2+\left(\sqrt{r^2-(x-a)^2}+b\right)^2\right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N}-LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2-(x-a)^2}} dx \qquad (3.3)$$

Above equation gives the total radar detection threat value for the arc traveled inside the outer radar region. For the derivation of the equations, please see Tezcaner Öztürk’s (2013) study.

In Figure 3.4, we give the demonstration of type 2 move. Illustrated trajectory is one of the infinitely many efficient trajectories between origin and destination points.

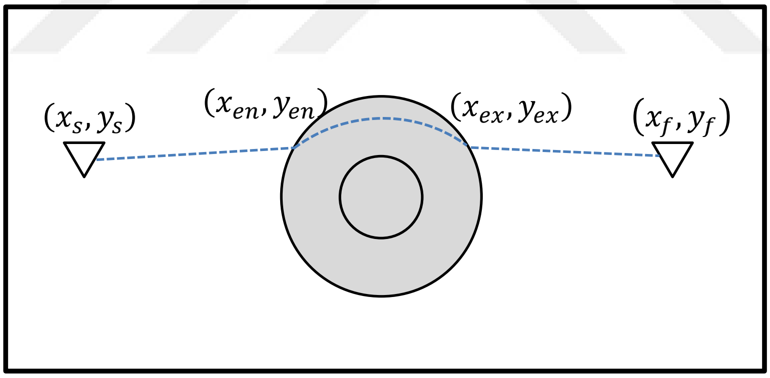


Figure 3.4 Example Movement between Radar Regions – Type 2

<u>Type 3</u>. Moves through Both the Inner and the Outer Radar Regions (Figure 3.5)

In this type, the efficient trajectories between two targets can pass through both the outer and the inner radar regions. Inside the outer radar region, it is assumed that UAV makes a circular movement as in Type 2 (centered at (a, b) with radius r). Inside the inner radar region, objectives do not conflict with each other since the probability of detection is 1 throughout the region. In other words, both objectives can be decreased simultaneously. Therefore, UAV follows the shortest path inside the inner radar region. Formulas for the total distance and total radar detection threat are given in (3.4) and (3.5), respectively. Here, (x_{ien}, y_{ien}) is the entrance point to the inner radar region and (x_{iex}, y_{iex}) is the exit point from the inner radar region.

$$D = \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2.arcsin\left(\frac{\sqrt{(x_{ien}-x_{en})^2+(y_{ien}-y_{en})^2}}{2r}\right).r +$$
$$\sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + 2.arcsin\left(\frac{\sqrt{(x_{ex}-x_{iex})^2+(y_{ex}-y_{iex})^2}}{2r}\right).r +$$
$$\sqrt{\left(x_f - x_{ex}\right)^2 + \left(y_f - y_{ex}\right)^2} \tag{3.4}$$

Here, the first and the last terms are the lengths of straight paths outside the outer radar region. The middle term is the direct distance between the entrance point to the inner radar region and the exit point from the inner radar region. The rests are the lengths of the arcs traveled inside the outer radar region.

$$RDT = \int_{x_{en}}^{x_{ien}}\left(\frac{10}{UB_{S/N}-LB_{S/N}}log_{10}\left(\frac{c}{\left(x^2+\left(\sqrt{r^2-(x-a)^2}+b\right)^2\right)^2}\right)-\frac{LB_{S/N}}{UB_{S/N}-LB_{S/N}}\right).\sqrt{\frac{r^2}{r^2-(x-a)^2}}dx \ +$$
$$\int_{x_{iex}}^{x_{ex}}\left(\frac{10}{UB_{S/N}-LB_{S/N}}log_{10}\left(\frac{c}{\left(x^2+\left(\sqrt{r^2-(x-a)^2}+b\right)^2\right)^2}\right)-\frac{LB_{S/N}}{UB_{S/N}-LB_{S/N}}\right).\sqrt{\frac{r^2}{r^2-(x-a)^2}}dx +$$
$$\sqrt{(x_{iex} - x_{ien})^2 + \left(y_{iex} - y_{ien}\right)^2} \tag{3.5}$$

In the above equation, the first and the second terms give the radar detection threat values corresponding to the circular paths inside the outer radar region. Last term is the radar detection threat value of the movement inside the inner radar region. Inside the inner radar region, radar detection probability is always 1 so the total radar detection threat measure is proportional to the distance of the straight path.

Due to conflicting objectives, there are infinitely many efficient trajectories between targets. In Figure 3.5, one of the efficient trajectories is illustrated. In the example, UAV goes inside the outer radar region first where it follows a curved path. Then, it goes inside the inner radar region and it follows a straight path.

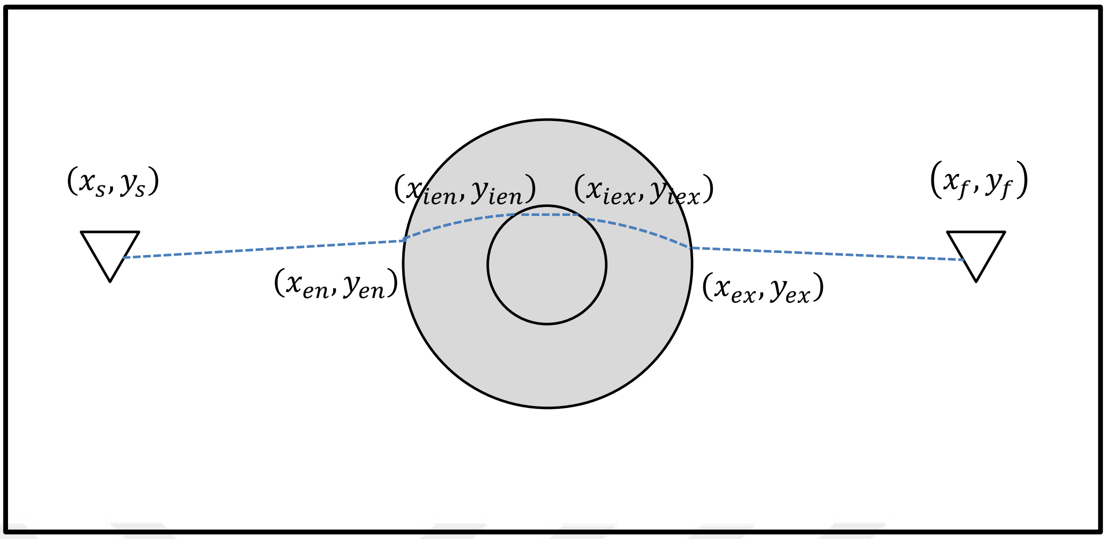
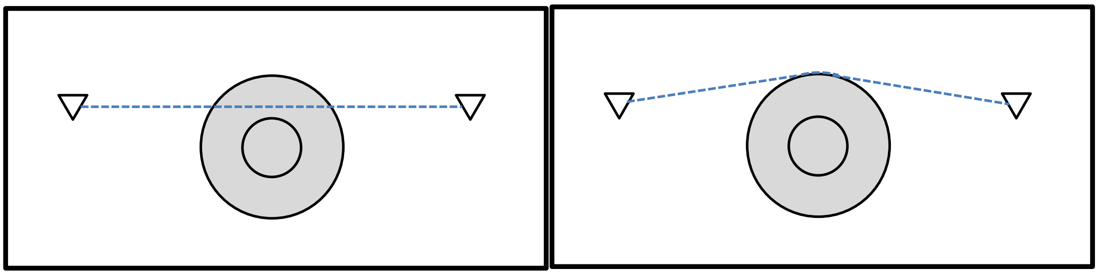


Figure 3.5 Example Movement between Radar Regions – Type 3

3.2.4 Nondominated Frontiers of Trajectories between two Targets

Tezcaner Öztürk (2013) developed the structure of the nondominated frontiers for each type of move. Specifically, for moves of Type 1 we have a single nondominated point (x^S), for moves of Type 2 we have a nondominated frontier that is curved, and for moves of Type 3 we have a two-piece nondominated frontier (one piece is a straight line and the other is curved). There are some important points that we use for determining the general structure of the nondominated frontiers. First we present extreme movements corresponding to these important points and then we show the structure of the nondominated frontiers.

For Type 2 moves, we have two extreme movements as illustrated in Figure 3.6. In Figure 3.6 (a), UAV follows a straight path between targets. Therefore, it is the minimum distance trajectory. In Figure 3.6 (b), UAV avoids the radar region and passes through the circumference of the outer radar region. Thus, it shows the minimum radar detection threat and maximum distance trajectory for Type 2 moves.



(a) Min D- Max RDT path

(b) Max D- Min RDT path

Figure 3.6 Type 2 Extreme Movements

Movements shown on the above figures form the two extreme efficient solutions of Type 2 nondominated frontier. Structure of the frontier is given in Figure 3.7. As it can be seen from the figure, the nondominated frontier consists of a curve. Left extreme point of the curve $\left(x^{LEC}_{(i,j)}\right)$ corresponds to the path with shortest distance between targets i and j (see Figure 3.6 (a)). Right extreme of the curve $\left(x^{REC}_{(i,j)}\right)$, on the other hand, corresponds to the objective values of the shortest path that poses no detection threat (Figure 3.6 (b)).

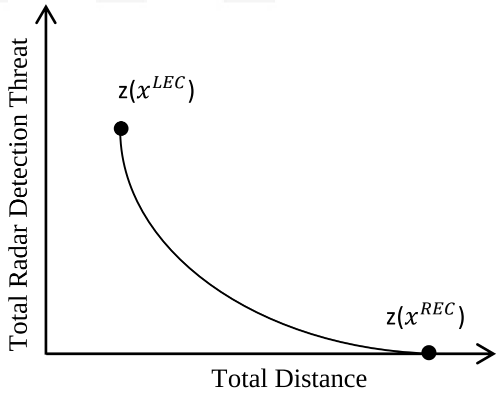
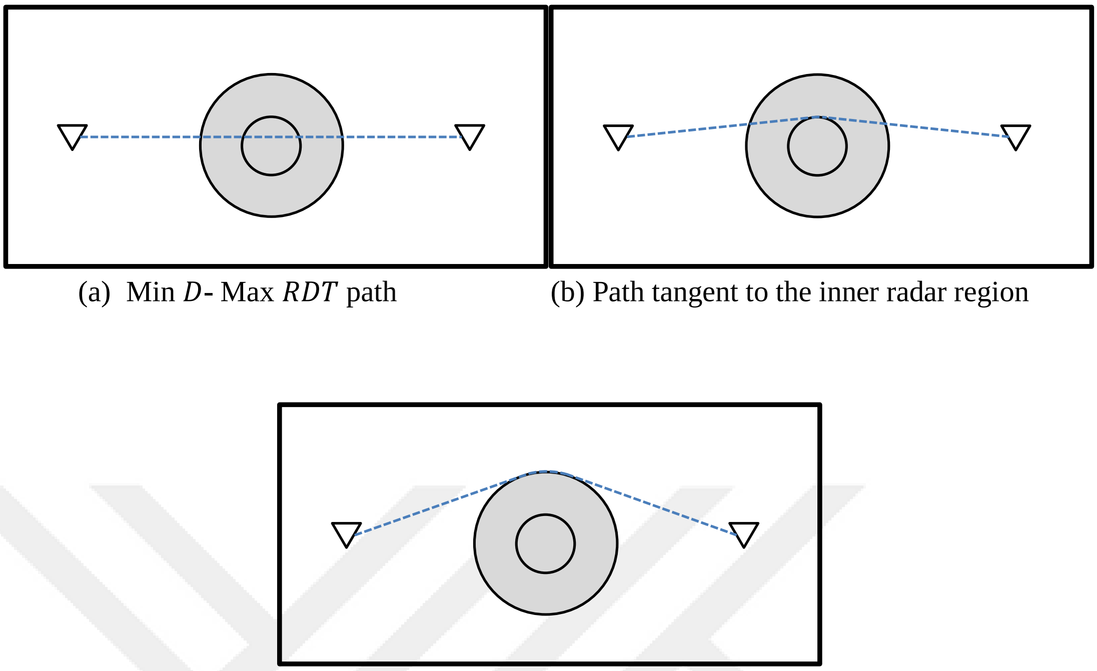


Figure 3.7 Nondominated Frontier of Type 2 Moves

For the third movement type, there are three special points used to define the structure of the nondominated frontier. As it is illustrated on Figure 3.8 (a), one of these points correspond to the straight path between targets. This is the minimum distance trajectory (with maximum radar detection threat). The maximum distance trajectory is given in Figure 3.8 (c). These two form the extreme solutions of the nondominated frontier. Beside these trajectories, UAV follows a curved path inside the outer radar region and it passes through the circumference of the inner radar region (see Figure 3.8 (b)).



(a) Min D- Max RDT path

(b) Path tangent to the inner radar region

(c) Max D- Min RDT path

Figure 3.8 Type 3 Extreme Movements

In Figure 3.9, Type 3 nondominated frontier structure is given. Left extreme solution of the nondominated frontier $\left(x_{(i,j)}^{LE}\right)$ corresponds to the straight path between targets (Figure 3.8 (a)). Left extreme of the curved part $\left(x_{(i,j)}^{LEC}\right)$ is the path that does not get inside the inner radar region but passes through the boundary (Figure 3.8 (b)). Right extreme solution$\left(x_{(i,j)}^{REC}\right)$ is the path with zero radar detection threat (Figure 3.8 (c)).

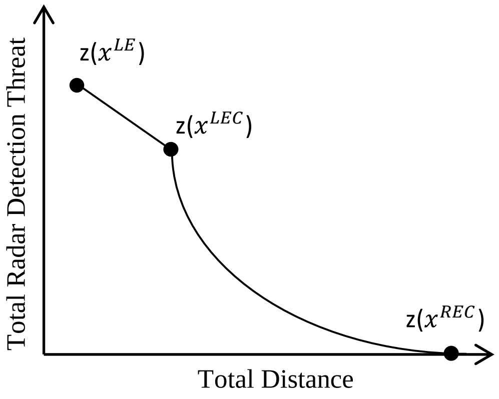


Figure 3.9 Nondominated Frontier of Type 3 Moves

Although there are infinitely many efficient trajectories in Type 2 and 3 moves, we do not need to generate all efficient solutions. Approximating the nondominated frontiers with equations that relate the two objectives is sufficient to characterize the nondominated solutions. Fitting an equation for the straight part in Type 3 move is straightforward. For the curved, convex, and continuous parts of the nondominated frontiers, we use Lq distance functions. These functions were first developed by Köksalan (1999) in a scheduling context. Later, Köksalan and Lokman (2009) showed on many combinatorial problems that an Lq distance function fitted using only a few nondominated points is able to approximate the nondominated set well. To fit an Lq distance function, we need three points on the nondominated frontiers; two points on the extremes of the curved-line and one central point on it, as shown in Figure 3.10. These three nondominated points can be found using exact or heuristic methods developed in Tezcaner Öztürk (2013).

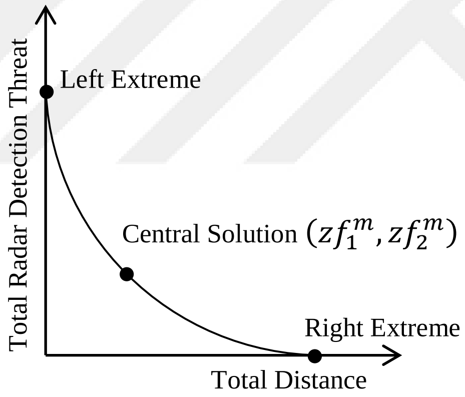

Total Radar Detection Threat
Left Extreme
Central Solution (zf_1^m, zf_2^m)
Right Extreme
Total Distance

Figure 3.10 Lq Distance Function

Let the first and second objective values of the left and right extreme points of the curved-line be (c_L^1, c_L^2) and (c_R^1, c_R^2), respectively. Let the first and second objective values of a central point be $(c_{central}^1, c_{central}^2)$. Then, we find the Lq distance function using (3.6) below.

$$(1 - zf_1^m)^q + (1 - zf_2^m)^q = 1 \qquad (3.6)$$

where, $zf^m = (zf_1^m, zf_2^m) = \left(\frac{c_{central}^1 - c_L^1}{c_R^1 - c_L^1}, \frac{c_{central}^2 - c_R^2}{c_L^2 - c_R^2}\right)$

The only unknown in equation (3.6) is the q value, that can be found using a nonlinear programming problem with a pseudo objective function satisfying (3.6) as the only constraint.

3.3 Formulation for Bi-objective UAV Route Planning Problem

Tezcaner Öztürk (2013) developed the mathematical model for the bi-objective route planning problem with infinitely many efficient trajectories between target pairs. In the model, approximated nondominated frontiers of each target pair are used. Therefore, to use the formulation, nondominated frontiers of each target pair should be approximated first.

Let $G = (N, E)$ be an undirected graph with target (node) set $N = \{1,2,..,n\}$. E is set of target pairs i and j. Then, target pairs (E) are classified into three sets as follows:

- E_{safe} : Set of target pairs having a single efficient solution (Type 1 move).
- E_{outer}: Set of taget pairs whose nondominated frontier consists of a single curved part (Type 2 move).
- E_{both} : Set of target pairs whose nondominated frontier consists of two parts: curved and a straight line (Type 3 move).

Some important points on the nondominated frontiers are illustrated in Section 3.1.4. These points are summarized below for the sake of completeness:

- *Type 1*: First and second objective values of the single efficient solution are $z_1(x^S_{(i,j)})$ and $z_2(x^S_{(i,j)})$.
- *Type 2:* First and second objective values of the left and right extreme points of the nondominated frontier are $\left(z_1\left(x^{LEC}_{(i,j)}\right), z_2(x^{LEC}_{(i,j)})\right)$ and $\left(z_1\left(x^{REC}_{(i,j)}\right), z_2(x^{REC}_{(i,j)})\right)$, respectively. Approximated *Lq* function's *q* value for target pair (*i,j*) is $q(i,j)$.
- *Type 3:* $\left(z_1\left(x^{LE}_{(i,j)}\right), z_2(x^{LE}_{(i,j)})\right)$ are the first and second objective values of the nondominated frontier's left extreme point. $\left(z_1\left(x^{LEC}_{(i,j)}\right), z_2(x^{LEC}_{(i,j)})\right)$ and $\left(z_1\left(x^{REC}_{(i,j)}\right), z_2(x^{REC}_{(i,j)})\right)$ are the objective values of the extreme points of the nondominated frontier's curved part respectively. Moreover, q value of the fitted *Lq* curve is $q(i,j)$.

For $(i,j) \in E, i < j$, d_{ij} and r_{ij} state the distance and radar detection threat values of the chosen trajectory between targets i and j. $y_{1(i,j)}$ determines whether a trajectory between targets i and j is used or not; it gets the value 0 if a trajectory between targets i and j is chosen and it gets the value 1 otherwise. For type 3 moves, nondominated frontier consists of two parts: a straight line and a curved part. Therefore, $y_{2(i,j)}$ and $y_{3(i,j)}$ are used in addition to variables $y_{1(i,j)}$. If no arc is used between targets, $y_{1(i,j)}$ again takes value 1. If the arc that passes through the outer radar region is used, $y_{2(i,j)}$ takes value 1. If the arc that passes through the inner radar region is used, $y_{3(i,j)}$ takes value 1. Furthermore, we have variables t in the model for Type 3 moves. These variables are used to define the region of the solution on the nondominated frontier.

Formulation of the Bi-objective Route Planning Problem with infinitely many efficient trajectories between target pairs:

$$\text{Min } D = \textstyle\sum_{(i,j)\in E} d_{ij} \qquad (3.7)$$

$$\text{Min } RDT = \textstyle\sum_{(i,j)\in E} r_{ij} \qquad (3.8)$$

$$d_{ij} = z_1(x^{S}_{(i,j)})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{safe} \qquad (3.9)$$

$$r_{ij} = z_2(x^{S}_{(i,j)})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{safe} \qquad (3.10)$$

$$d_{ij} \le z_1(x^{REC}_{(i,j)})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{outer} \qquad (3.11)$$

$$d_{ij} \ge z_1(x^{LEC}_{(i,j)})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{outer} \qquad (3.12)$$

$$r_{ij} \le z_2(x^{LEC}_{(i,j)})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{outer} \qquad (3.13)$$

$$r_{ij} \ge z_2(x^{LEC}_{(i,j)}) - z_2(x^{LEC}_{(i,j)})\left(1 - \left(1 - \frac{d_{ij} - z_1(x^{LEC}_{(i,j)})}{z_1(x^{REC}_{(i,j)}) - z_1(x^{LEC}_{(i,j)})}\right)^{q(i,j)}\right)^{1/q(i,j)} - My_{1(i,j)}$$

$$\forall (i,j) \in E_{outer} \qquad (3.14)$$

$$d_{ij} = 0 * t_{1(i,j)} + z_1(x^{LE}_{(i,j)}) * t_{2(i,j)} + z_1(x^{LEC}_{(i,j)}) * t_{3(i,j)} + z_1(x^{REC}_{(i,j)}) * t_{4(i,j)}$$

$$\forall (i,j) \in E_{both} \qquad (3.15)$$

$$t_{1(i,j)} + t_{2(i,j)} + t_{3(i,j)} + t_{4(i,j)} = 1 \qquad \forall (i,j) \in E_{both} \qquad (3.16)$$

$$t_{1(i,j)} \le y_{1(i,j)} \qquad \forall (i,j) \in E_{both} \qquad (3.17)$$

$$t_{2(i,j)} \leq y_{2(i,j)} \qquad \forall (i,j) \in E_{both} \qquad (3.18)$$

$$t_{3(i,j)} \leq y_{2(i,j)} + y_{3(i,j)} \qquad \forall (i,j) \in E_{both} \qquad (3.19)$$

$$t_{4(i,j)} \leq y_{3(i,j)} \qquad \forall (i,j) \in E_{both} \qquad (3.20)$$

$$y_{1(i,j)} + y_{2(i,j)} + y_{3(i,j)} = 1 \qquad \forall (i,j) \in E_{both} \qquad (3.21)$$

$$r_{ij} \leq z_2(x_{(i,j)}^{LE})(1 - y_{1(i,j)}) \qquad \forall (i,j) \in E_{both} \qquad (3.22)$$

$$r_{ij} \geq m_{ij}d_{ij} + n_{ij} - M(1 - y_{2(i,j)}) \qquad \forall (i,j) \in E_{both} \qquad (3.23)$$

$$r_{ij} \geq z_2(x_{(i,j)}^{LEC}) - z_2(x_{(i,j)}^{LEC})\left(1 - \left(1 - \frac{d_{ij} - z_1(x_{(i,j)}^{LEC})}{z_1(x_{(i,j)}^{REC}) - z_1(x_{(i,j)}^{LEC})}\right)^{q(i,j)}\right)^{1/q(i,j)} - M(1 - y_{3(i,j)})$$

$$\forall (i,j) \in E_{both} \qquad (3.24)$$

$$\textstyle\sum_{j \in N, j < i} y_{1(j,i)} + \sum_{j \in N, j > i} y_{1(i,j)} = |N| - 3 \qquad \forall i \in N \qquad (3.25)$$

$$\textstyle\sum_{i \in S} \sum_{j \in N-S} y_{1(i,j)} \leq |S||N - S| - 2 \qquad |S| = 3, \dots, \lfloor N/2 \rfloor \qquad (3.26)$$

$$0 \leq t_{t(i,j)} \leq 1 \qquad \forall (i,j) \in E_{both}, t = 1, \dots, 4 \qquad (3.27)$$

$$d_{ij}, r_{ij} \geq 0 \qquad \forall (i,j) \in E \qquad (3.28)$$

$$y_{1(i,j)} \in \{0,1\} \qquad \forall (i,j) \in E_{outer}, E_{safe} \qquad (3.29)$$

$$y_{t(i,j)} \in \{0,1\} \qquad \forall (i,j) \in E_{both}, t = 1, \dots, 4 \qquad (3.30)$$

In constraints (3.7) and (3.8), total distance and radar detection threat values of the route is minimized, respectively. If an edge of a type 1 target pair is used, D and RDT values of the chosen trajectory are set to the single solution's values, (3.9) and (3.10), respectively. If type 2 target pair is selected, D value of the trajectory should be between the extreme distance values of the nondominated frontier ((3.11) and (3.12)). RDT value of the trajectory should also be between the extreme RDT values of the frontier ((3.13) and (3.14)). D and RDT values should satisfy Lq function together (3.14). Furthermore, if a type 3 target pair is selected ($y_{1(i,j)} = 0$), distance value of the chosen trajectory is written as a convex combination of special points of nondominated frontier in (3.15) and (3.16). If the selected movement goes inside inner radar region $y_{2(i,j)}$ takes value 1, if outer radar region is passed $y_{3(i,j)}$ takes value 1. In (3.16) and (3.17), $t_{1(i,j)}$ takes value 1 if a target pair is not used in the tour

$(y_{1(i,j)} = 1)$ and other t variables take value 0. (3.18), (3.19), (3.20) and (3.21) are used to define the area in which the trajectory lies on the nondominated frontier. In (3.22), (3.23) and (3.24), constraints on RDT value of the trajectory is stated. Finally, (3.25) ensures that for each node there is only one incoming and one outgoing trajectory. Equations (3.26) are the subtour elimination constraints. The rest states the variable types and bounds. For more detailed explanation on the constraints, please see Tezcaner Öztürk (2013).

CHAPTER 4

INTERACTIVE ALGORITHM FOR UNDERLYING LINEAR PREFERENCE FUNCTIONS

We develop an interactive algorithm to find the most preferred solution of a route planner (RP) where the RP has an underlying linear preference function. A linear function implies that the marginal rates of substitution between the objective function values are constant. In our approach, we consider two objectives and we treat them separately. The problem terrain is two dimensional and it is continuous. In the literature, there are studies working on the same problem but most of the related studies discretize the problem terrain. Therefore, the main contribution of our study is that we develop interactive algorithms for the route planning problem of UAVs in continuous terrain. In this section, we assume that RP has an underlying linear preference function. We first give the idea of our interactive algorithm and its steps which is followed by the implementation details.

4.1 Solution Approach for Underlying Linear Preference Functions

The general form of a linear preference function $U(x)$, is as follows:

$$U(x) = w_1\, z_1(x) + w_2\, z_2(x) + \cdots + w_p\, z_p(x)$$

Here, there are p objectives, $z_k(x)$ represents the value of the k^{th} objective corresponding to solution x, and w_k represents the weight (importance) given to objective k.

In our case, we have two objective functions, and without loss of generality, we normalize the weights given to the two objectives such that they sum to 1. Then the preference function reduces to (4.1).

$$U(x) = w\, z_1(x) + (1 - w)\, z_2(x) \text{ where } 0 < w < 1 \qquad (4.1)$$

The most preferred solution of the RP is the nondominated point that gives the *smallest* $U(x)$ value.

Before starting the algorithm, the only information we have is that w takes a value between 0 and 1; $w \in (0{,}1)$. During the algorithm, we ask the RP to compare a pair of points, and we reduce the search region and find a shorter interval around the exact value of w as $w \in [l, u]$. We next explain how we interpret the preferences of the RP, and demonstrate the algorithm on a figure.

4.1.1 Interpreting the Pairwise Comparisons of the RP

Each time the RP compares two solutions, s/he can either state a preference, or be indifferent between the two alternatives. For the first case, assume that alternative A is the preferred one among alternatives A and B. We assume that the RP can only make a preference if the preference function values of the two alternatives differ more than a threshold, δ. The motivation behind this threshold is that, we select nondominated points on a continuous frontier, where the objective function values of the chosen solutions A and B can be really close. In reality, the RP may not be that sensitive to such small preference function differences and may not be able to differentiate between very similar alternatives. We write inequality (4.2) when A is preferred to B.

$$U\big(z(x^B)\big) - U\big(z(x^A)\big) \geq \delta \qquad (4.2)$$

This provides a lower or upper bound on the value of w as given in (4.3) and (4.4).

$$w \geq \frac{\delta - [\, z_2(x^B) - z_2(x^A)]}{[\, z_1(x^B) - z_1(x^A) - z_2(x^B) + z_2(x^A)]} \text{ if } [\, z_1(x^B) - z_1(x^A) - z_2(x^B) + z_2(x^A)] > 0 \qquad (4.3)$$

$$w \leq \frac{\delta - [\, z_2(x^B) - z_2(x^A)]}{[\, z_1(x^B) - z_1(x^A) - z_2(x^B) + z_2(x^A)]} \text{ if } [\, z_1(x^B) - z_1(x^A) - z_2(x^B) + z_2(x^A)] < 0 \qquad (4.4)$$

If the RP is indifferent between alternatives A and B, then their preference function value difference is at most the threshold, δ. We write inequality (4.5) for this case.

$$-\delta \leq U\big(z(x^B)\big) \; - U(z(x^A)) \leq \delta \tag{4.5}$$

Inequality (4.5) results in estimates on both the lower and upper bounds on w. Let the current lower bound on the weight be w_{LB} and the new lower bound estimated after a preference from the RP be $w_{LB,new}$. The w_{LB} is updated only if $w_{LB,new} > w_{LB}$. Otherwise, we do not make any updates on the current bound. That is,

$$w_{LB} \leftarrow max\{w_{LB}, w_{LB,new}\}$$

Similarly, the upper bound estimate is updated only if the new estimate is smaller.

$$w_{UB} \leftarrow min\{w_{UB}, w_{UB,new}\}$$

Throughout the algorithm, we keep a range (a lower and an upper bound) for the weight, w. These new bounds help us reduce the search area around a smaller region near the most preferred solution of the RP.

4.1.2 The Steps of the Interactive Algorithm

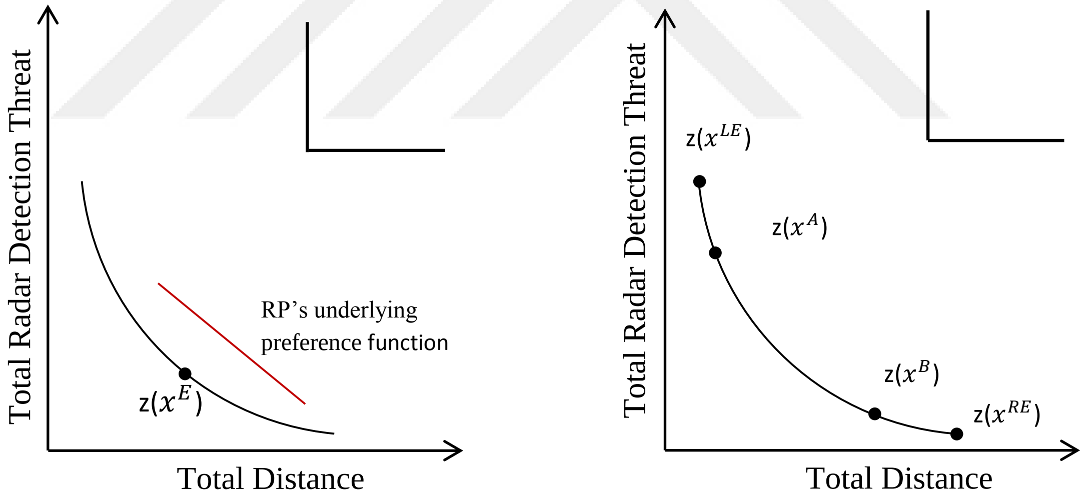
Our approach is inspired from Tezcaner and Köksalan's (2011) study, in which they develop algorithm *BestSol* to find the most preferred solution of a decision maker for bi-objective integer programs. The main idea of our algorithm is to ask for comparison between point pairs in order to make a sizeable update on the bounds of w. Each time the bounds on w are updated, we narrow the region on the nondominated frontier that covers the most preferred point of the DM. Since we have a continuous nondominated frontier with infinitely many points, it is not meaningful to expect the algorithm to finally result in a single most preferred point of the RP, since the RP could be indifferent between many solutions that have close preference values with the most preferred solution. Rather, we expect to the algorithm to give a point that is close to the most preferred point of the RP.

We illustrate the algorithm on Figure 4.1. Let the nondominated frontier and the underlying linear preference function be as shown in Figure 4.1 (a), and the most preferred solution of the RP be solution E. Before starting the algorithm, we do not have any information on the structure of the nondominated frontier and the preference function. The initial bounds on w are represented by the two linear lines on the top right corner, $w \in (0,1)$, and will be updated during the algorithm as the RP states preferences among alternatives. We can ask for comparison between any two solutions, but we try to select a meaningful pair that would result in useful information on the weight range.

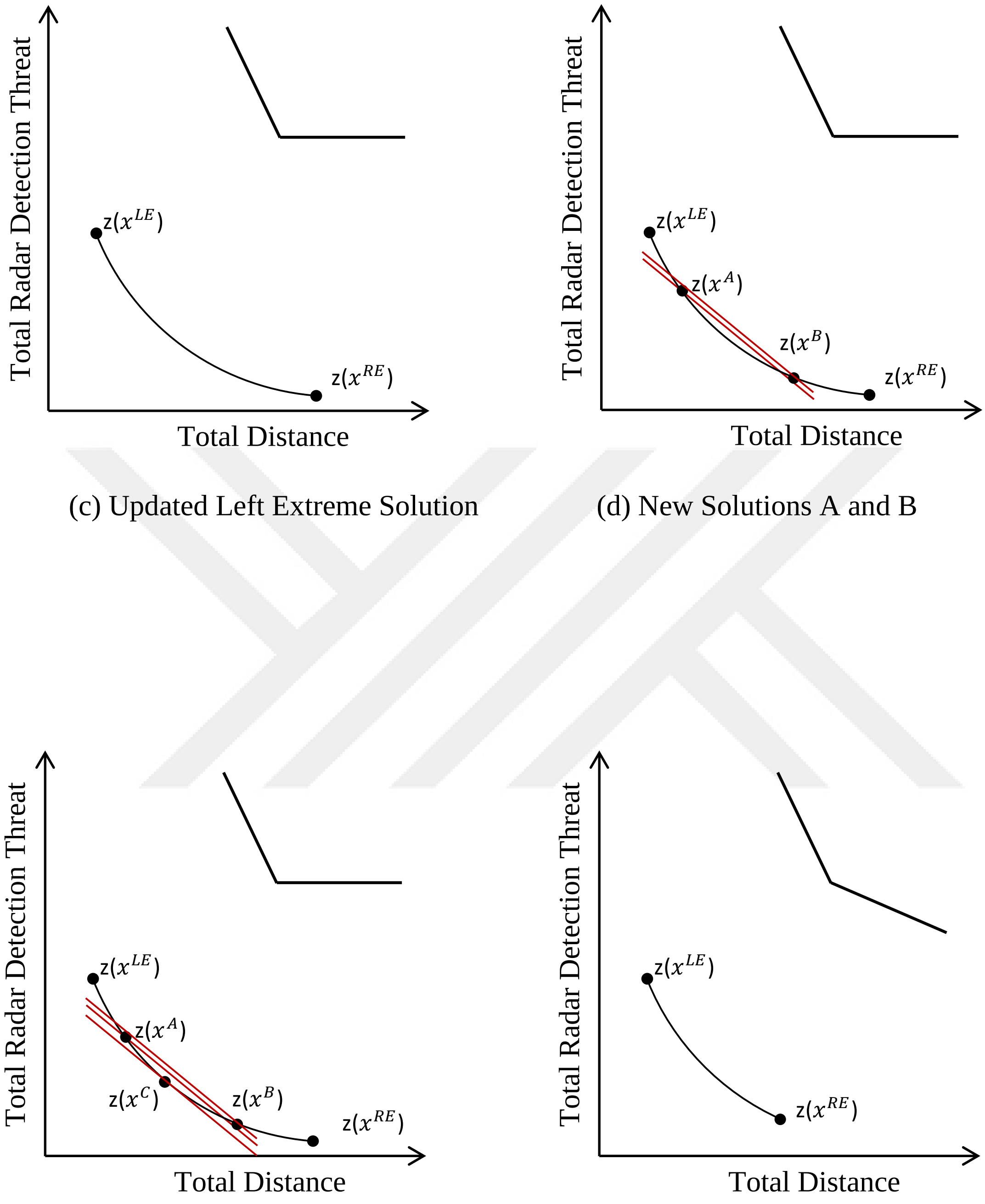
We start the algorithm by finding the left and right extreme efficient tours, $x^{LE}$and x^{RE}, as shown in Figure 4.1 (b). We then divide the weight range into three equal-length intervals, and obtain two weights (w_A, w_B) that correspond to the end points of the first and second intervals. Let the corresponding solutions for $w = w_A$ and w_B be A and B, respectively. The objective function values of A and B are compared by the RP, and since the underlying preference function results in a lower value for solution B, the RP prefers solution B. This preference results in an update on the upper bound on weight (as shown on the top right corner) and the left extreme nondominated point is updated accordingly (Figure 4.1 (c)). We divide the new range on w to three equal-length intervals, and find new solutions, A and B, to compare (Figure 4.1 (d)). Here, assume that the RP is indifferent between the two alternatives since their preference function values are very close to each other. Let us assume that this indifference does not lead to an update in the weight range. In this case, we find a new solution C in between A and B, and ask the RP to compare C with one of the solutions, A or B (Figure 4.1 (e)). Assume that C is compared with B, and the RP prefers C. We update the upper bound on the weight and find a new right extreme nondominated point (Figure 4.1 (f)). We find new points A, B and C and this time ask for comparison between A and C (Figure 4.1 (g)). Assume that the RP is indifferent between these two solutions, and we cannot update the bounds on the weight. After the RP is indifferent twice (Figure 4.1 (d) and (g)), we terminate the algorithm. We estimate that the most preferred solution corresponds to the weight dividing the weight region to two equal intervals. In this example, solution F is estimated as the most preferred solution (Figure 4.1 (h)), which is a solution that is close to the true most preferred solution of the RP (solution E).

The algorithm starts with the comparison of two solutions that divide the feasible region equally, and proceeds in the same manner until the RP is indifferent between two alternatives. After this first indifference answer, we find three points, and ask for comparison between the solution in the middle and one solution from either side. We continue until we obtain another indifference answer, in which we terminate the algorithm. Alternatively, we may terminate the algorithm if the normalized Euclidean distance in the objective space between two solutions that are to be compared is less than a threshold, Δ. We then estimate the most preferred solution using the same method.

When we terminate the algorithm, we know that the true best solution lies between the extreme solutions and we present the solution dividing the extreme region to two in terms of the preference weight, *w*, to the RP. However, more detailed search can be performed between the extreme solutions. Weight range can be discretized into small intervals and solutions corresponding to these weights can be found and presented to the RP. Then, the RP can select among these solutions.


Total Radar Detection Threat
RP's underlying preference function
$z(x^E)$
Total Distance
Total Radar Detection Threat
$z(x^{LE})$
$z(x^A)$
$z(x^B)$
$z(x^{RE})$
Total Distance

(a) Nondominated Frontier (b) Extreme Solutions and Solutions A and B

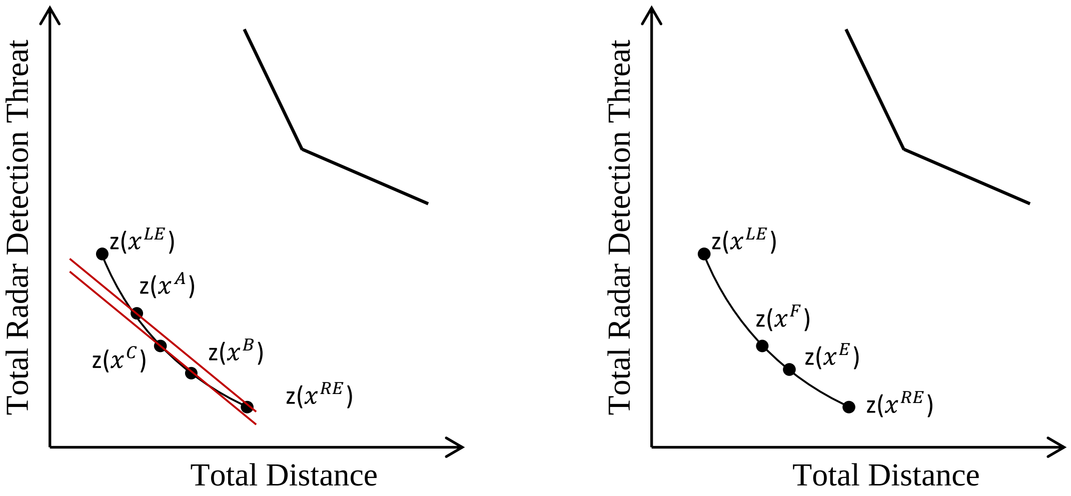


(c) Updated Left Extreme Solution

(d) New Solutions A and B

(e) Solution C

(f) Updated Right Extreme Solution



(g) New Solutions A, B and C (h) Estimated Solution F and Best Solution E

Figure 4.1 Demonstration of the Interactive Algorithm for Linear Preference Functions

Until now, the steps of the algorithm is explained and demonstrated. The interactive algorithm uses another algorithm; OptTSP. OptTSP is used to find a solution (a tour) that minimizes the preference function for a w value. We first explain Algorithm OptTSP below and then present the steps of the interactive algorithm.

Algorithm OptTSP

To find the most preferred tour corresponding to a w value, we do not need to consider all efficient trajectories between each target pair. Rather, there is a single best trajectory between each target pair that is certainly used for that w value if the vehicle moves between these two targets consecutively. Because of this simplification, to find the most preferred tour corresponding to a w value, we find the single best trajectory to be used between each target pair. We first find the move types between target pairs as explained in 3.1.3, and then construct their nondominated frontiers as explained in 3.1.4. Then we use Algorithm OptTSP to find the best trajectory to be used between each target pair, for a w value and construct the best tour for that w value. Some points used in the OptTSP Algorithm are shown in Figure 4.2. We refer these points in the algorithm OptTSP.

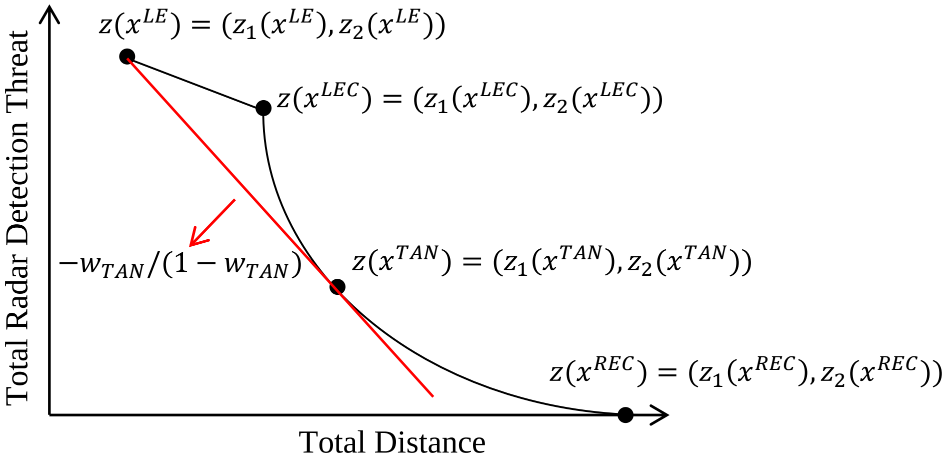

$z(x^{LE}) = (z_1(x^{LE}), z_2(x^{LE}))$
$z(x^{LEC}) = (z_1(x^{LEC}), z_2(x^{LEC}))$
$-w_{TAN}/(1 - w_{TAN})$
$z(x^{TAN}) = (z_1(x^{TAN}), z_2(x^{TAN}))$
$z(x^{REC}) = (z_1(x^{REC}), z_2(x^{REC}))$
Total Radar Detection Threat
Total Distance

Figure 4.2 Tangent line and points representation on target pair's nondominated frontiers – Movement Type 3

In Figure 4.2, it is shown that for a specific linear weight (w_{TAN}) the preference function values of the left extreme point, x^{LE}, and the tangent point, x^{TAN}, are the same. For higher values than w_{TAN}, slope of the preference function gets steeper. If the preference line is steeper than the tangent line, there is a single preferred solution which is the left extreme point. On the other hand, for lower values than w_{TAN}, the most preferred solution lies on the curve between x^{TAN} and x^{REC}.

Steps of Algorithm OptTSP

Step 1: For a given weight (w), find a single best solution ($x_{(i,j)}^{Best}$) for each target pair $(i,j) \in E$ depending on the pair's nondominated frontier structure as follows:

- If $(i,j) \in E_{safe}$ (*Type* 1)*:* Set $x_{(i,j)}^{Best}$ to the single efficient solution.
- If $(i,j) \in E_{outer}$ (*Type* 2): Solve the following model and set $x_{(i,j)}^{Best}$ to the resulting solution.

$$(MT2_{(i,j)}) \qquad \text{Min} \quad w\, z_1(x) + (1\text{-}\, w)\, z_2(x)$$

$$\left[1 - \frac{z_1(x) - z_1(x^{LEC})}{z_1(x^{REC}) - z_1(x^{LEC})}\right]^q + \left[1 - \frac{z_2(x) - z_2(x^{REC})}{z_2(x^{LEC}) - z_2(x^{REC})}\right]^q = 1$$

$$z_1(x^{LEC}) \le z_1(x) \le z_1(x^{REC})$$

$$z_2(x^{REC}) \le z_2(x) \le z_2(x^{LEC})$$

- If $(i,j) \in E_{both}$ (*Type 3*): Solve the following model and find w_{TAN} :

  $(MT3_{(i,j)})$ Max w_{TAN}

  $$\left[1-\frac{z_1(x^{TAN})-z_1(x^{LEC})}{z_1(x^{REC})-z_1(x^{LEC})}\right]^q+\left[1-\frac{z_2(x^{TAN})-z_2(x^{REC})}{z_2(x^{LEC})-z_2(x^{REC})}\right]^q=1$$

  $$w_{TAN}z_1(x^{LE})+(1-w_{TAN})z_2(x^{LE}) = w_{TAN}z_1(x^{TAN})+(1-w_{TAN})z_2(x^{TAN})$$

  $$z_1(x^{LEC}) \le z_1(x^{TAN}) \le z_1(x^{REC})$$

  $$z_2(x^{REC}) \le z_2(x^{TAN}) \le z_2(x^{LEC})$$

  - If $w \ge w_{TAN}$, set $x_{(i,j)}^{Best} = x^{LE}$.
  - If $w < w_{TAN}$, solve model $(MT2_{(i,j)})$ and set $x_{(i,j)}^{Best}$ to the resulting solution.

    $(MT2_{(i,j)})$ Min $w\, z_1(x)$ + (1- w) $z_2(x)$

    $$\left[1-\frac{z_1(x)-z_1(x^{LEC})}{z_1(x^{REC})-z_1(x^{LEC})}\right]^q+\left[1-\frac{z_2(x)-z_2(x^{REC})}{z_2(x^{LEC})-z_2(x^{REC})}\right]^q=1$$

    $$z_1(x^{LEC}) \le z_1(x) \le z_1(x^{REC})$$

    $$z_2(x^{REC}) \le z_2(x) \le z_2(x^{LEC})$$

Step 2: For each target pair (i,j), calculate the edge cost as follows:

$$Edgecost_{(i,j)} = w * z_1(x_{(i,j)}^{Best}) + (1-w) * z_2(x_{(i,j)}^{Best})$$

Step 3: Find the tour visiting all targets that minimizes the sum of edge costs. In Step 2, we find the cost of each trajectory to be followed between the target pairs. After multiple trajectories reduce to a single trajectory, the problem turns into a single objective Traveling Salesperson Problem (TSP) whose mathematical model is given in Appendix C. We find the best TSP solution $z(x^{TSPBest})$ solving this problem exactly in Step 3. In UAV route planning problems, the vehicle generally visits a few number of targets. Therefore, we solve this model using CPLEX, since it is not computationally demanding. Alternatively, CONCORDE (single objective symmetric TSP solver) can be used.

Steps of the Interactive Algorithm:

Step 0: (Initialization) Set the parameters ε and Δ. Let $w^* \in [w_{UB}, w_{LB}]$ for $w_{UB} = 1 - \rho$ and $w_{LB} = \rho$ where ρ is a very small positive constant. Find $w_{TAN}(i,j)$, $(i,j) \in E_{both}$ using model $(MT3_{(i,j)})$.

Step 1: Find $z(x^{TSPLE})$ and $z(x^{TSPRE})$ using Algorithm OptTSP for weights w_{UB} and w_{LB}, respectively.

Step 2: Divide the feasible weight range into three equal intervals. Let the end points of first and second intervals be w_A and w_B, respectively.

$$w_A = \frac{2}{3}(w_{UB} - w_{LB}) + w_{LB}$$

$$w_B = \frac{1}{3}(w_{UB} - w_{LB}) + w_{LB}$$

Find $z(x^A)$ and $z(x^B)$ using Algorithm OptTSP for weights w_A and w_B, respectively. Let the alternative with the lower first objective value be x^L and the other one be x^R without loss of generality.

Step 3: Calculate the relative distance, d_{rel}, between x^L and x^R as follows:

$$d_{rel} = \sqrt{(z_1(x^L) - z_1(x^R))^2 + (z_2(x^L) - z_2(x^R))^2}$$

- If $d_{rel} \leq \Delta$, go to Step 6.
- If $d_{rel} > \Delta$, ask the DM to compare $z(x^L)$and $z(x^R)$.
    - If $z(x^L)$ is preferred to $z(x^R)$, find w_{LB} as follows:

    $$w_{LB} = \frac{z_2(x^L) - z_2(x^R) + \mathcal{E}}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

    Find $z(x^{TSPRE})$ using Algorithm OptTSP for weight w_{LB} and go to Step 2.
    - If $z(x^R)$ is preferred to $z(x^L)$, find w_{UB} as follows:

    $$w_{UB} = \frac{z_2(x^L) - z_2(x^R) - \mathcal{E}}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

    Find $z(x^{TSPLE})$ using Algorithm OptTSP for weight w_{UB} and go to Step 2.

- If the DM is indifferent, find $w_{LB'}$ and $w_{UB'}$ as follows:

$$w_{LB'} = \frac{z_2(x^L) - z_2(x^R) + \mathcal{E}}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

$$w_{UB'} = \frac{z_2(x^L) - z_2(x^R) - \mathcal{E}}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

  Set $w_{LB} \leftarrow max\{w_{LB}, w_{LB'}\}$ and $w_{UB} \leftarrow min\{w_{UB}, w_{UB'}\}$ and find $z(x^{TSPRE})$ and $z(x^{TSPLE})$ using Algorithm OptTSP for weights w_{LB} and w_{UB}, respectively. Go to Step 4.

Step 4: Divide the feasible weight range into six equal intervals. Let the end points of second and fourth intervals be w_A and w_B, respectively. If there is no Point C available, let the end point of third interval be w_C.

$$w_A = \frac{2}{3}(w_{UB} - w_{LB}) + w_{LB}$$

$$w_C = \frac{1}{2}(w_{UB} - w_{LB}) + w_{LB}$$

$$w_B = \frac{1}{3}(w_{UB} - w_{LB}) + w_{LB}$$

Find $z(x^A)$ and $z(x^B)$ using Algorithm OptTSP for weights w_A and w_B, respectively. If there is no Point C available, find $z(x^C)$ using Algorithm OptTSP for weight w_C.

Each time in this step, choose one of the points A or B sequentially to compare with Point C. Rename Point C and the selected point (A or B) such that the one with the lower first objective value be x^L and the other one be x^R without loss of generality.

Calculate the relative distance, d_{rel}, between x^L and x^R as follows:

$$d_{rel} = \sqrt{(z_1(x^L) - z_1(x^R))^2 + (z_2(x^L) - z_2(x^R))^2}$$

- If $d_{rel} \leq \Delta$, change the selected point. Rename them such that the one with the lower first objective value be x^L and the other one be x^R without loss of generality. Calculate d_{rel} with new x^L and x^R points.
  - If $d_{rel} \leq \Delta$, go to Step 6.
  - If $d_{rel} > \Delta$, go to Step 5.
- If $d_{rel} > \Delta$, go to Step 5.

Step 5: Ask the DM to compare $z(x^L)$ and $z(x^R)$.

- If $z(x^L)$ is preferred to $z(x^R)$, find w_{LB} as follows:

$$w_{LB} = \frac{z_2(x^L) - z_2(x^R) + \varepsilon}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

Let x^C be x^L. Find $z(x^{TSPRE})$ using Algorithm OptTSP for weight w_{LB} and go to Step 4.

- If $z(x^R)$ is preferred to $z(x^L)$, find w_{UB} as follows:

$$w_{UB} = \frac{z_2(x^L) - z_2(x^R) - \varepsilon}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

Let x^C be x^R. Find $z(x^{TSPLE})$ using Algorithm OptTSP for weight w_{UB} and go to Step 4.

- If the DM is indifferent, find $w_{LB'}$ and $w_{UB'}$ as follows:

$$w_{LB'} = \frac{z_2(x^L) - z_2(x^R) + \varepsilon}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

$$w_{UB'} = \frac{z_2(x^L) - z_2(x^R) - \varepsilon}{z_1(x^R) - z_1(x^L) - z_2(x^R) + z_2(x^L)}$$

Set $w_{LB} \leftarrow max\{w_{LB}, w_{LB'}\}$ and $w_{UB} \leftarrow min\{w_{UB}, w_{UB'}\}$ and find $z(x^{TSPRE})$ and $z(x^{TSPLE})$ using Algorithm OptTSP for weights w_{LB} and w_{UB}, respectively. Go to Step 6.

Step 6: Find the estimate for the most preferred point $z(x^*)$ using Algorithm OptTSP for weight w^*, where $w^* = \frac{1}{2}(w_{UB} - w_{LB}) + w_{LB}$.

4.2 Scaling the Objectives

During the interactive algorithm, we find a narrow weight range around the true weight of the RP. Since we are combining the two objectives linearly, the magnitudes of the objectives should be comparable. For this, we scale both of the objectives between 0-1 scale using their extreme values. Then, the weight estimate gives an insight of the relative importance of the objectives. We next show how we transfer the true weight, w, of the RP for the route planning problem to the weight we use in $MT2_{(i,j)}$ and $MT3_{(i,j)}$.

Let x^{TSPLE} and x^{TSPRE} be the left and right extreme points for the routing problem. We obtain the following preference function (to be minimized) when the two objectives are scaled between 0 and 1 using these extreme points:

$$\text{Min} \quad w\frac{\mathrm{z}_1(x^{TSP}) - \mathrm{z}_1(x^{TSPLE})}{\mathrm{z}_1(x^{TSPRE}) - \mathrm{z}_1(x^{TSPLE})} + (1-w)\frac{\mathrm{z}_2(x^{TSP}) - \mathrm{z}_2(x^{TSPRE})}{\mathrm{z}_2(x^{TSPLE}) - \mathrm{z}_2(x^{TSPRE})} \qquad (4.6)$$

Here, $\mathrm{z}_1(x^{TSP}) = \sum_{(i,j)\in E} \mathrm{z}_1\left(x_{(i,j)}^{Best}\right) y_{ij}$ and $\mathrm{z}_2(x^{TSP}) = \sum_{(i,j)\in E} \mathrm{z}_2\left(x_{(i,j)}^{Best}\right) y_{ij}$ where y_{ij} states whether the trajectory between targets i and j is used or not.

When we rewrite (4.6), we obtain the following:

$$\text{Min} \quad w\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_1\left(x_{(i,j)}^{Best}\right) y_{ij}\right] - \mathrm{z}_1(x^{TSPLE})}{\mathrm{z}_1(x^{TSPRE}) - \mathrm{z}_1(x^{TSPLE})} + (1-w)\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_2\left(x_{(i,j)}^{Best}\right) y_{ij}\right] - z_2(x^{TSPRE})}{z_2(x^{TSPLE}) - z_2(x^{TSPRE})}$$

$$= w\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_1\left(x_{(i,j)}^{Best}\right) y_{ij}\right]}{\mathrm{z}_1(x^{TSPRE}) - \mathrm{z}_1(x^{TSPLE})} + (1-w)\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_2\left(x_{(i,j)}^{Best}\right) y_{ij}\right]}{z_2(x^{TSPLE}) - z_2(x^{TSPRE})} -$$

$$\left[\mathrm{w}\frac{\mathrm{z}_1(x^{TSPLE})}{\mathrm{z}_1(x^{TSPRE}) - \mathrm{z}_1(x^{TSPLE})} + (1-\mathrm{w})\frac{\mathrm{z}_2(x^{TSPRE})}{\mathrm{z}_2(x^{TSPLE}) - \mathrm{z}_2(x^{TSPRE})}\right]$$

We can omit the last term in square brackets which is constant and does not affect the objective function. Moreover, if we set $\mathrm{z}_1(x^{TSPRE}) - \mathrm{z}_1(x^{TSPLE}) = \text{TSPrange1}$ and $\mathrm{z}_2(x^{TSPLE}) - \mathrm{z}_2(x^{TSPRE}) = \text{TSPrange2}$, then we can write the objective function as follows:

$$\text{Min}\ w\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_1\left(x_{(i,j)}^{Best}\right) y_{ij}\right]}{\text{TSPrange1}} + (1-w)\frac{\left[\sum_{(i,j)\in E} \mathrm{z}_2\left(x_{(i,j)}^{Best}\right) y_{ij}\right]}{\text{TSPrange2}} \qquad (4.7)$$

Equation (4.7) is made of independent terms for each target pair, so that for each pair (i,j) we can find the edge minimizing the following objective function and set the resulting solution to $x_{(i,j)}^{Best}$:

$$\text{Min} \quad w\frac{z_1(x)}{\text{TSPrange1}} + (1-w)\frac{z_2(x)}{\text{TSPrange2}}$$

Thus, while we are applying the interactive algorithm to the scaled-TSP, we need to modify models $MT2_{(i,j)}$ and $MT3_{(i,j)}$. In this case, to find the preferred solution for one-region frontier pairs, the model below should be solved instead of $MT2_{(i,j)}$:

$$(MT2'_{(i,j)}) \qquad \text{Min} \quad w\,\frac{\mathrm{z}_1(\mathrm{x})}{\text{TSPrange1}} + (1\text{- w})\,\frac{\mathrm{z}_2(\mathrm{x})}{\text{TSPrange2}}$$

$$\left[1-\frac{\mathrm{z}_1(\mathrm{x})-\mathrm{z}_1(x^{LEC})}{\mathrm{z}_1(x^{REC})-\mathrm{z}_1(x^{LEC})}\right]^{\mathrm{q}}+\left[1-\frac{\mathrm{z}_2(\mathrm{x})-\mathrm{z}_2(x^{REC})}{\mathrm{z}_2(x^{LEC})-\mathrm{z}_2(x^{REC})}\right]^{\mathrm{q}}=1$$ (When all points are scaled in Lq function, they cancel out each other and the resulting equation is the same equation for the unscaled Lq function)

$$z_1(x^{LEC}) \le z_1(x) \le z_1(x^{REC})$$

$$z_2(x^{REC}) \le z_2(x) \le z_2(x^{LEC})$$

Furthermore, the updated model for $MT3_{(i,j)}$ is as follows:

$$(MT3'_{(i,j)}) \qquad \text{Max} \quad w_{TAN}$$

$$\left[1-\frac{\mathrm{z}_1(x^{TAN})-\mathrm{z}_1(x^{LEC})}{\mathrm{z}_1(x^{REC})-\mathrm{z}_1(x^{LEC})}\right]^{\mathrm{q}}+\left[1-\frac{\mathrm{z}_2(x^{TAN})-\mathrm{z}_2(x^{REC})}{\mathrm{z}_2(x^{LEC})-\mathrm{z}_2(x^{REC})}\right]^{\mathrm{q}}=1$$

$$\mathrm{w_{TAN}}\,\frac{\mathrm{z}_1(x^{LE})}{\text{TSPrange1}}+(1-\mathrm{w_{TAN}})\,\frac{\mathrm{z}_2(x^{LE})}{\text{TSPrange2}} = \mathrm{w_{TAN}}\,\frac{\mathrm{z}_1(x^{TAN})}{\text{TSPrange1}}+(1-\mathrm{w_{TAN}})\,\frac{\mathrm{z}_2(x^{TAN})}{\text{TSPrange2}}$$

$$\mathrm{z}_1(x^{LEC}) \le \mathrm{z}_1(x^{TAN}) \le z_1(x^{REC})$$

$$\mathrm{z}_2(x^{REC}) \le \mathrm{z}_2(x^{TAN}) \le z_2(x^{LEC})$$

CHAPTER 5

INTERACTIVE ALGORITHM FOR UNDERLYING QUASICONVEX PREFERENCE FUNCTIONS

We develop an interactive algorithm for two underlying preference functions. In the previous chapter, we assume that RP has an underlying linear preference function and develop interactive algorithm. In this chapter, we consider a similar problem with the exception of the underlying preference function. We have two objectives to be minimized: total distance and total radar detection threat. We consider routing of UAVs in continuous space. However, this time we assume that RP has a more general quasiconvex underlying preference function. Quasiconvex preference functions are a family of preference functions, including linear preference functions. In the literature, these functions are widely used since they are considered to represent human behavior well. The marginal rate of substitution is decreasing for these functions; as one criterion gets better, to further improve that criterion, the amount of sacrifice from the other criterion decreases. We give the structure of quasiconvex functions in the following definition.

Definition 5.1 f is a quasiconvex function if $f(\sum_{i=1}^{p}\mu_i x_i) \leq \underset{i}{max}\, f(x_i)$ for $\sum_{i=1}^{p}\mu_i = 1\,, \mu_i \geq 0$.

We do not have a general closed form for quasiconvex preference functions as we had for the linear preference functions, but due to their special structures, we can eliminate some regions in the search area that we are sure the RP is not interested in. The following lemma shows this idea for minimization type objectives. Korhonen et al. (1984) developed cone dominance idea for maximization type objectives and Tezcaner Öztürk (2013) adapted it to minimization problems.

Lemma 1 (Tezcaner Öztürk, 2013) Consider a quasiconvex function f defined in a p-dimensional Euclidean space R^p. Consider distinct points $x_i \in R^p$, $i = 1,2,\dots,m$ and let $f(x_k) > f(x_i), i \neq k$. If $z \in Z$ and $z \neq x_k$, where $Z = \{z | z = x_k + \sum_{i=1;i\neq k}^{m} \mu_i(x_k - x_i), \mu_i \geq 0\}$ it follows that $f(z) \geq f(x_k)$.

To illustrate this idea, assume that we have two objectives to be minimized, and two solutions A and B, in the objective function space as demonstrated in Figure 5.1. If the RP prefers solution A to solution B, we infer that all solutions in the shaded region are at most as preferred as solution B (please see Korhonen et al., 1984 for more details). These solutions are referred as *cone-dominated solutions*, which are dominated by the cone (the dashed line) that initiates at solution B and moves in the south-east direction, with the same slope of the line that connects solutions A and B. The RP is not interested in any of the solutions on the right side of solution B, and we should only consider efficient solutions on the left side of B. Solution B is now the extreme efficient solution of the region at which the most preferred solution of the RP lies.

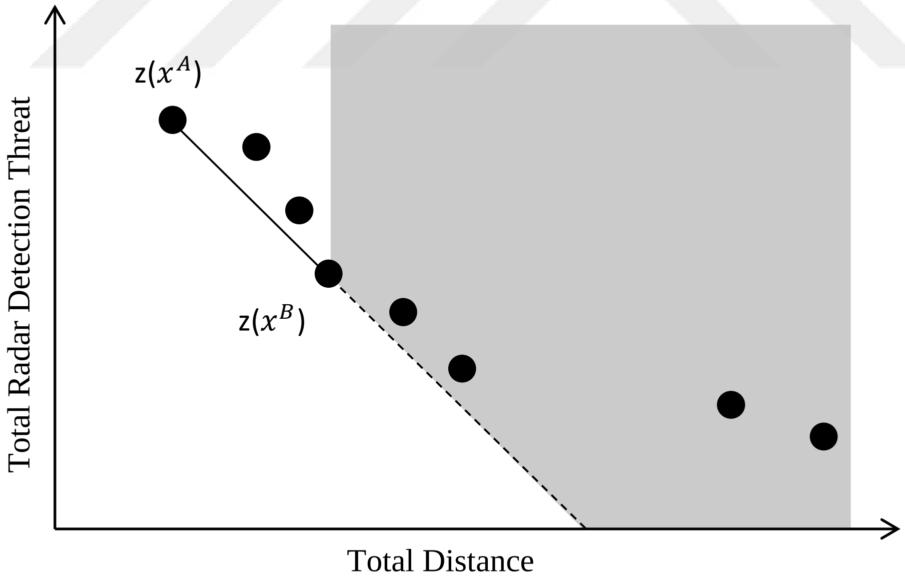


Figure 5.1 Elimination of Inferior Regions – Quasiconvex Preference Functions

Unlike linear preference functions, the most preferred solution for a RP with an underlying quasiconvex preference function, can be either an unsupported efficient solution or a supported efficient solution. Finding supported efficient solutions in biobjective problems is easier, since each supported efficient solution minimizes a weighted combination of the two objectives. This, in turn, reduces to a single objective variant of the biobjective problem, which is easier to solve. Finding unsupported efficient solutions is harder since we need to introduce more constraints in the objective space to restrict the search region. Our approach focuses on these two types of solutions separately, and uses the above properties for quasiconvex preference functions, to narrow the search region around the true preferred solution of the RP.

We next explain our interactive algorithm for underlying quasiconvex preference functions in details.

5.1 Solution Approach for Underlying Quasiconvex Preference Functions

For underlying quasiconvex preference functions, we develop an interactive algorithm that consists of two parts. In the first part, similar to our approach for the linear case, we ask for comparison between supported nondominated solutions. However, we do not limit our search with only supported nondominated solutions since the most preferred solution of an RP can also be an unsupported nondominated solution as explained above. Therefore, to find solutions close to the true most preferred solution, we search all type of nondominated solutions, supported and unsupported in the second part of the algorithm.

First Part of the Interactive Algorithm for Quasiconvex Preference Functions

In the first part, our aim is to find a region between supported efficient solutions that the true most preferred solution of the RP lies. Suppose we want to decide on the region that the true most preferred solution lies between n supported nondominated solutions. Since these are supported nondominated solutions, each solution minimizes a weighted combination of the objectives for a different w value in equation (4.1). Instead of finding all n solutions, we only find the corresponding linear weights, and when we need a solution, we optimize objective (4.1) for that weight. For this purpose, initial linear weight range of (0,1) is divided into $n-1$ equal-length intervals. The end point of each interval i is set to w_{i+1} where $1 \leq i \leq$

$n-2$ and w_1 and w_n are set to ρ and $1-\rho$, respectively. These weights will be used to find the supported nondominated solutions as required. Each time a preference information is obtained, we also update the linear weight range (as we did for the linear preference functions), but this range is not used to define the structure of the preference function. Rather, it is used to find two points to be compared. A weight dividing the weight range into two equal-length intervals is found and the closest weight out of predefined n weights is selected. Assume that this weight is w_X. Then the solution X corresponding to w_X is found with the same method used in linear case. To find the other solution for pairwise comparison, We select one of the closest predefined weights to w_X . Let this weight be w_Y and its corresponding solution be Y. We refer the two solutions, X and Y, as neighboring solutions. We present X and Y to the RP and depending on her/his answer, we update linear weight range using equations (4.3) and (4.4). Range is updated so that the weight corresponding to the inferior solution is assigned to one of the linear weight bounds by using the cone dominance idea. Algorithm continues in the same manner. When we come up with a solution which is preferred to both of its neighboring solutions, we stop the first part of the algorithm.

This approach is inspired from Tezcaner and Köksalan’s (2011) approach, where they find the most preferred solution of a DM for bicriteria integer programs. They ask for comparison between adjacent efficient solutions and find the solution that is preferred to both its adjacent solutions. In this algorithm, we try to implement this idea to the routing problem with a continuous nondominated frontier. Since we have infinitely many efficient solutions, finding adjacent solutions is not meaningful. Therefore, we ask for comparison between neighboring solutions; whose linear weights are the closest among the predefined weights. For example, among three consecutive supported solutions out of n solutions demonstrated in Figure 5.2, suppose L is the most preferred one. It is preferred to its neighboring solutions M and K. Therefore, we know that the true best solution lies in one of the two shaded rectangular regions in Figure 5.2.

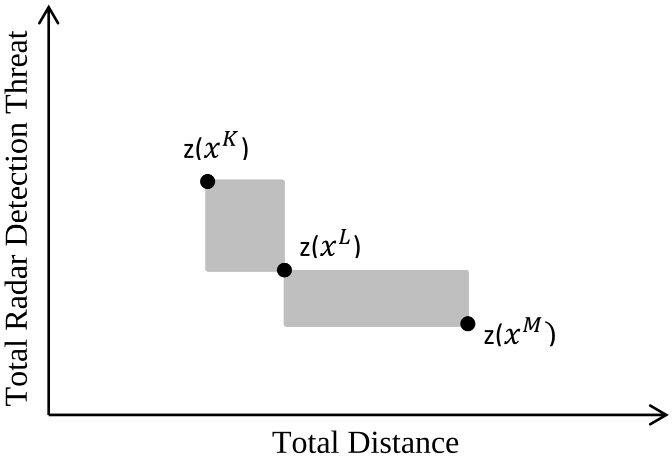


Figure 5.2 First stage stopping case of the algorithm

If the most preferred supported solution out of n solutions is one of the extreme nondominated solutions, we conclude this stage with one rectangular region. For example, in Figure 5.3 right extreme solution is the most preferred solution among n supported nondominated solutions. Therefore, we ended up with one rectangular region for further examination.

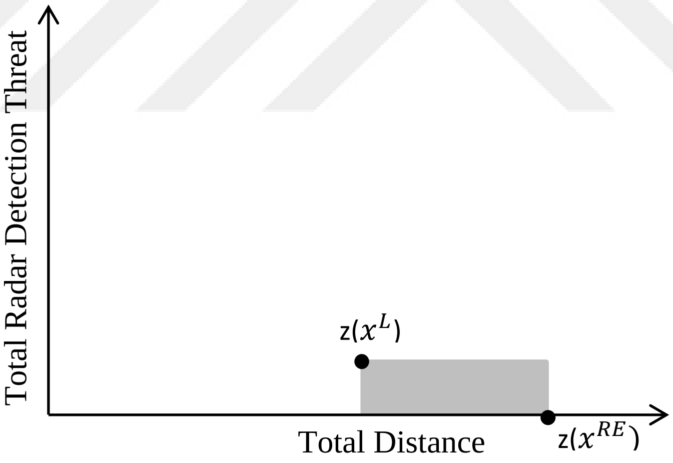


Figure 5.3 First stage stopping case when extreme solution is the most preferred solution out of n solutions

Second Part of the Interactive Algorithm for Quasiconvex Preference Functions

In the second part of the algorithm, we search inside the rectangular regions obtained in the first part. We again ask pairwise comparisons to the RP. However, we do not limit the search with supported solutions. We use the mathematical model developed by Lokman et al. (2016) to find the true best solution for multi-objective integer programs for underlying quasiconcave preference functions. After getting a preference information, they write constraints to reduce the inferior regions by using the cone dominance idea and solve a model to find a new nondominated solution. We find new solutions in the same way. However, since we are working on a problem with a continuous nondominated frontier, finding the exact true best solution would require asking infinitely many questions to the RP. Instead, our aim is to get as close to the true best solution as possible and make a good estimate. Therefore, when the incumbent solution and newly obtained solution are close to each other in terms of a threshold distance, Δ, the algorithm terminates.

First, we present the theorem in Lokman et al. (2016) which leads us to inferior region elimination constraints for a maximization problem. Then, we explain how we implement their findings on a minimization problem.

In the paper, they assume that the DM has an underlying nondecreasing quasiconcave preference function. They partition the criteria indices into two sets $S_{\leq}^{M,K}$ and $S_{>}^{M,K}$ for any two points x^M and x^K such that:

$$S_{\leq}^{M,K} = \{i: z_i(x^K) - z_i(x^M) \leq 0\}$$

$$S_{>}^{M,K} = \{j: z_j(x^K) - z_j(x^M) > 0\}$$

Theorem (Lokman et al. 2016) Let U be a nondecreasing quasiconvex function defined in a p-dimensional Euclidean space $\mathfrak{R}^p$. Consider two distinct nondominated points x^M and x^K such that $U(x^K) < U(x^M)$. Then, a point x is cone dominated by cone $C(x^M\,; x^K)$ if and only if the following conditions hold:

(i) $z_i(x) \leq z_i(x^K), \quad \forall i \in S_{\leq}^{M,K}$

(ii) $z_i(x)\left(z_j(x^K) - z_j(x^M)\right) + z_j(x)\left(z_i(x^M) - z_i(x^K)\right) \leq z_j(x^K)z_i(x^M) - z_i(x^K)z_j(x^M), \quad \forall i \in S_{\leq}^{M,K}, \forall j \in S_{>}^{M,K}$

From the above theorem, they make an inference about points which are not cone dominated. A solution x is not cone dominated by $C(x^M\,; x^K)$ if at least one of the following conditions holds:

(i') There exists $i \in S_{\leq}^{M,K}$ satisfying $z_i(x^K) < z_i(x)$.

(ii') There exists $i \in S_{\leq}^{M,K}$ and $j \in S_{>}^{M,K}$ satisfying $z_j(x^K)z_i(x^M) - z_i(x^K)z_j(x^M) < z_i(x)\left(z_j(x^K) - z_j(x^M)\right) + z_j(x)\left(z_i(x^M) - z_i(x^K)\right)$

Similarly, for a minimization problem, a point x is not cone dominated by $C(x^M\,; x^K)$ if at least one of the following conditions holds where the criteria index partition stays the same:

(i'') There exists $j \in S_{>}^{M,K}$ satisfying $z_j(x^K) > z_j(x)$.

(ii'') There exists $i \in S_{\leq}^{M,K}$ and $j \in S_{>}^{M,K}$ satisfying $z_j(x^K)z_i(x^M) - z_i(x^K)z_j(x^M) > z_i(x)\left(z_j(x^K) - z_j(x^M)\right) + z_j(x)\left(z_i(x^M) - z_i(x^K)\right)$

These conditions are represented on a bi-criteria example in Figure 5.4. Conditions *(i'')* and *(ii'')* are represented by *C₁* and *C₂*, respectively. If either *C₁* or *C₂* is satisfied, a solution is not cone dominated by $C(x^M\,; x^K)$ where $S_{>}^{M,K} = \{1\}$ and $S_{\leq}^{M,K} = \{2\}$.

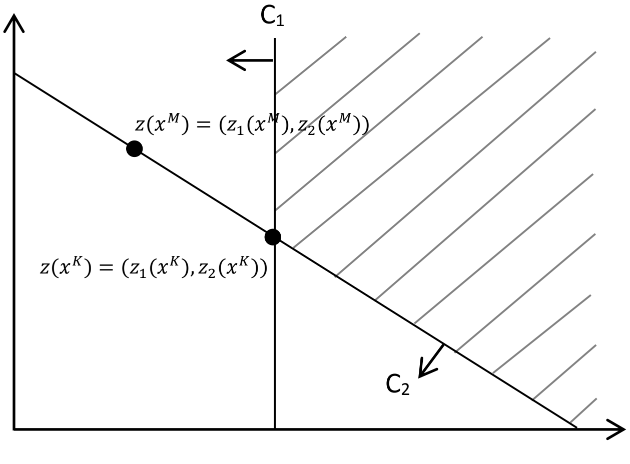

C₁
$z(x^M) = (z_1(x^M), z_2(x^M))$
$z(x^K) = (z_1(x^K), z_2(x^K))$
C₂

Figure 5.4 Conditions for a solution for not being cone dominated in a minimization problem

In Lokman et al. (2016)'s paper, they develop a mathematical model to find a new nondominated solution satisfying conditions *(i')* and *(ii')*. We modified their model which makes use of conditions *(i'')* and *(ii'')* for a bi-objective routing problem where both objectives are to be minimized.

We also want the new point to be on the nondominated frontier of routing problem. Therefore, we added additional constraints specific to our bi-objective routing problem. These constraints are developed by Tezcaner Öztürk and Köksalan (2013) and given in 3.2. The model is as follows:

$T = \{(x^M, x^K): U(x^K) > U(x^M)\}$ is the set of preferences

$S_{\leq}^{M,K} = \{i: z_i(x^K) - z_i(x^M) \leq 0\}$

$S_{>}^{M,K} = \{j: z_j(x^K) - z_j(x^M) > 0\}$

$x^{inc} = (z_1(x^{inc}), z_2(x^{inc}))$ is the incumbent solution whose preference function value is the smallest among all considered solutions. Then, the formulation is as follows:

Min $\quad \alpha$

$$\alpha \geq \lambda(z_1(x) - z_1(x^{TSPLE}))/(z_1(x^{TSPRE}) - z_1(x^{LTSPE})) \tag{5.1}$$

$$\alpha \geq (1 - \lambda)(z_2(x) - z_2(x^{TSPRE}))/(z_2(x^{TSPLE}) - z_1(x^{TSPRE})) \tag{5.2}$$

$$z_j(x^K) - \varepsilon + M(1 - r_j^{M,K}) \geq z_j(x)\ , \qquad \forall(x^M, x^K) \in T, \quad \forall i \in S_{\leq}^{M,K} \tag{5.3}$$

$$z_j(x^K)z_i(x^M) - z_i(x^K)z_j(x^M) - \varepsilon + M(1 - t_{ij}^{M,K}) \geq z_i(x)\left(z_j(x^K) - z_j(x^M)\right) + z_j(x)\left(z_i(x^M) - z_i(x^K)\right), \qquad \forall(x^M, x^K) \in T, \quad \forall i \in S_{\leq}^{M,K}, \quad \forall j \in S_{>}^{M,K} \tag{5.4}$$

$$\sum_{j \in S_{>}^{M,K}} r_j^{M,K} + \sum_{\substack{i \in S_{\leq}^{M,K} \\ j \in S_{>}^{M,K}}} t_{ij}^{M,K} = 1, \quad \forall(x^M, x^K) \in T \tag{5.5}$$

$$z_i(x^{inc}) - \varepsilon + M(1 - y_i) \geq z_i(x) \qquad\qquad i = 1,2 \tag{5.6}$$

$$\sum_{i=1}^{2} y_i = 1 \tag{5.7}$$

$$r_j^{M,K}, t_{ij}^{M,K} \in \{0,1\}, \quad \forall(x^M, x^K) \in T, \quad \forall i \in S_{\leq}^{M,K}, \quad \forall j \in S_{>}^{M,K} \tag{5.8}$$

$$y_i \in \{0,1\} \qquad i = 1,2 \tag{5.9}$$

$$x \in Efficient\ Frontier\ of\ Route\ Planning\ Problem \tag{5.10}$$

The model tries to find a new point on the predefined Tchebycheff direction from the ideal point. (5.1) and (5.2) are written for this purpose and the maximum weighted distance is minimized in the objective. (5.3) and (5.4) are the cone constraints explained previously. (5.5) ensures that at least one the constraints (5.3) or (5.4) is satisfied. We search for a new nondominated solution different than the incumbent solution in constraint (5.6). In one criterion, the new solution should be better than the incumbent solution ((5.6) and (5.7)). (5.8) and (5.9) state variable types. (5.10) are the constraints given in Section 3.2, and they guarantee that the new point is on the nondominated frontier of the route planning problem.

In the second part of our interactive algorithm, each time we search for a new nondominated solution, we solve the model given above. We specify the direction from the ideal point for finding the new solution by setting the weight, λ. For this purpose, after each RP's answer, we update the Tchebycheff weight range which we use in finding new nondominated solutions in the search region. We select a weight (λ) dividing the weight range into two equal length intervals to find the new point in that direction. Each time we find a new nondominated point, we ask for comparison with the incumbent solution. The algorithm terminates when the relative distance between the two solutions to be compared is less than a threshold distance value.

Tchebycheff weight range update

We update Tchebycheff weight range with the method suggested by Bozkurt et al. (2010). In their paper, for bi-criteria case, boundary weight giving the same weighted Tchebycheff function value for two points is found. Depending on the preference, one of the Tchebycheff weight range bounds can be updated to the boundary weight. However, since the RP's underlying preference function may be different than Tchebycheff, we may get infeasibility after some questions. If we encounter such situations, we remove the past preferences until we obtain feasibility.

To summarize, our interactive algorithm for underlying quasiconvex preference functions is developed in two stages. In the first stage, we search the objective space with supported nondominated solutionsand reduce the search region to rectangular regions around the most preferred supported nondominated solution. In the second stage, we search inside the rectangular regions using the mathematical model developed by Lokman et al. (2016). We terminate the algorithm when the incumbent and the newly found nondominated solutions are close enough in terms of a predefined threshold distance.

CHAPTER 6

DEMONSTRATION OF THE ALGORITHM

6.1 Problem Generation

Tezcaner Öztürk (2013) randomly generated an example five-target UAV route planning problem. The locations of the radars and targets are given in Appendix D. We implement the interactive algorithms on her problem. Additionally, we implement the algorithms on a larger problem that we generate randomly. For this, we develop a mathematical model to locate a given number of radars and targets to a predefined terrain size.

First, we randomly locate the targets. Depending on the targets' locations, we place the radars. Targets are located randomly one by one into the terrain. While doing this, we make sure that the direct distances between the target to-be-located and all pre-located targets are greater than the diameter of the radar region. By doing so, we leave enough space for radars to be located between any two targets. After target locations are set, we locate the radars use the mathematical model below. Radars should be located such that they are responsible from the surveillance of a group of targets. Therefore, we assign each target to one radar and locate each radar close to the targets assigned to it. For this, we minimize the maximum Tchebycheff distance between each radar and the targets assigned to it.

Model takes the target locations as inputs and finds the radar locations. The details are given as follows:

Sets:

I : Set of targets

J : Set of radars

Decision Variables:

$maxdist_j$: Maximum of Tchebycheff distances of targets assigned to radar j

α_{ij} : Tchebycheff distance between target i and radar j

b_{ij} : 1 if target i is assigned to radar j, 0 otherwise

rx_j : x- coordinate of radar j

ry_j : y- coordinate of radar j

$u1_{ij}, u2_{ij}, u3_{ij}, u4_{ij}$.... $u12_{ij}$: Binary variables for the corresponding constraints

Parameters:

tx_i : x- coordinate of target i

ty_i : y- coordinate of target i

n : Number of targets to be assigned to each radar j

t : Number of targets

M : Very large positive constant

ε : Very small positive constant

R: Radar radius

Model:

Min $\sum_j maxdist_j$

S.to: $\alpha_{ij} \geq tx_i - rx_j \qquad \forall\, i \in I, j \in J$ (6.1)

$\alpha_{ij} \geq -tx_i + rx_j \qquad \forall\, i \in I, j \in J$ (6.2)

$$\alpha_{ij} \geq ty_i - ry_j \qquad \forall\, i\ \in I, j \in J \quad (6.3)$$

$$\alpha_{ij} \geq -ty_i + ry_j \qquad \forall\, i\ \in I, j \in J \quad (6.4)$$

$$maxdist_j \geq \alpha_{ij} - M\left(1 - b_{ij}\right) \qquad \forall\, i\ \in I, j \in J \quad (6.5)$$

$$\sum_i b_{ij}\ = n \qquad \forall\, j \in J \quad (6.6)$$

$$\sum_j b_{ij}\ = 1 \qquad \forall\, i \in I \quad (6.7)$$

$$tx_i - rx_j + Mu1_{ij}\ \geq R + \varepsilon \qquad \forall\, i\ \in I, j \in J \quad (6.8)$$

$$-tx_i + rx_j + Mu2_{ij}\ \geq R + \varepsilon \qquad \forall\, i\ \in I, j \in J \quad (6.9)$$

$$ty_i - ry_j + Mu3_{ij} \geq R + \varepsilon \qquad \forall\, i\ \in I, j \in J \quad (6.10)$$

$$-ty_i + ry_j + Mu4_{ij} \geq R + \varepsilon \qquad \forall\, i\ \in I, j \in J \quad (6.11)$$

$$u1_{ij} + u2_{ij} + u3_{ij} + u4_{ij} \leq 3 \qquad \forall\, i\ \in I, j \in J \quad (6.12)$$

$$rx_j - rx_k + Mu5_{jk} \geq 2R + \varepsilon \qquad \forall\, (j,k) \in J \quad (6.13)$$

$$-rx_j + rx_k + Mu6_{jk} \geq 2R + \varepsilon \qquad \forall\, (j,k) \in J \quad (6.14)$$

$$ry_j - ry_k + Mu7_{jk} \geq 2R + \varepsilon \qquad \forall\, (j,k) \in J \quad (6.15)$$

$$-ry_j + ry_k + Mu8_{jk} \geq 2R + \varepsilon \qquad \forall\, (j,k) \in J \quad (6.16)$$

$$u5_{jk} + u6_{jk} + u7_{jk} + u8_{jk} \leq 3 \qquad \forall\, (j,k) \in J \quad (6.17)$$

$$rx_j\ \leq tx_i - M\left(1 - b_{ij}\right) + 2Mu9_{ij} \qquad \forall\, (j,k) \in J \quad (6.18)$$

$$\sum_i u9_{ij}\ \leq t - 1 \qquad \forall\, j \in J \quad (6.19)$$

$$rx_j\ \geq tx_i - M\left(1 - b_{ij}\right) - Mu10_{ij} \qquad \forall\, i\ \in I, j \in J \quad (6.20)$$

$$\sum_i u10_{ij}\ \leq t - 1 \qquad \forall\, j \in J \quad (6.21)$$

$$ry_j\ \leq ty_i - M\left(1 - b_{ij}\right) + 2Mu11_{ij} \qquad \forall\, i\ \in I, j \in J \quad (6.22)$$

$$\sum_i u11_{ij}\ \leq t - 1 \qquad \forall\, j \in J \quad (6.23)$$

$$ry_j \geq ty_i - M(1 - b_{ij}) - Mu12_{ij} \qquad \forall\, i \in I, j \in J \quad (6.24)$$

$$\sum_i u12_{ij} \leq t - 1 \qquad \forall\, j \in J \quad (6.25)$$

Maximum Tchebycheff distance between each radar and the targets assigned to it is calculated. These maximum distances are summed for all radars and minimized in the objective function.

Constraints (6.1) and (6.2) find the x-coordinate differences for each radar-target pair. (6.3) and (6.4), on the other hand, calculates the y- coordinate differences. In all of these constraints, α_{ij} is set to be greater than or equal to these differences. Since we are minimizing the sum of distances in the objective function, α_{ij} gets the maximum difference value. In other words, α_{ij} stands for the Tchebycheff distance between pair (i,j). For each radar, the maximum Tchebycheff distance to itself from all targets assigned to it is found in constraint (6.5). In constraint (6.6), the number of targets that can be assigned to each radar is set to *n*. It is assumed that the number of targets is a multiple of the number of radars. Furthermore, in (6.7) we assign all targets to one radar.

Constraints (6.8), (6.9), (6.10), (6.11) and (6.12) ensure that distances between targets and radars are greater than radar radius in at least one coordinate. This guarantees that targets are located outside of radar regions, which is one of our assumptions stated in Chapter 3. Likewise, it is also desired to have non-overlapping radar regions as stated in constraints (6.13), (6.14), (6.15), (6.16) and (6.17). Tchebycheff distances between radar centers are forced to be greater than the effective radar diameter.

Constraints (6.18), (6.19), (6.20) and (6.21) allow radars to be placed between the x-coordinates of the targets assigned to it. Similarly, constraints (6.22), (6.23), (6.24) and (6.25) prevent radar center from being located outside of the y-coordinates of the targets assigned to it.

We use MATLAB for solving the terrain generation model. For locating the radars, we call CPLEX from MATLAB. This method can be used to generate different sized

UAV route planning problems. By using the above method, we generate a nine-target problem with three radars which is given Appendix E.

6.2 Results of Interactive Algorithm for Linear Preference Functions

We assume that the RP has the following underlying linear preference function, whose parameter, $w,$ is unknown to us.

$\mathrm{U}(z) = w\, z_1(x) + (1 - w)\, z_2(x)$ where $0 < w < 1.$

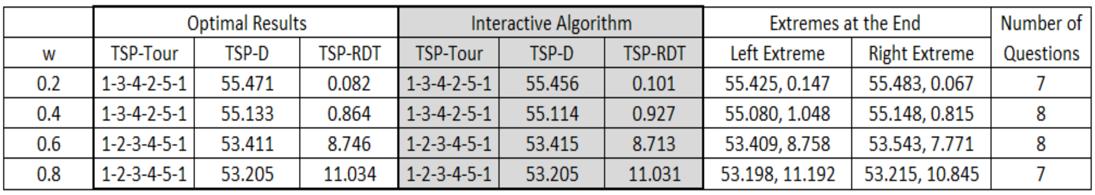
We set four values to w; 0.2, 0.4, 0.6, and 0.8; and solved the interactive algorithm for each case simulating the preferences of the RP for the given w. We assume that the RP cannot make a preference between two solutions if the difference between their preference function values is less than 0.001 ($= \delta$). A termination condition for the algorithm is when the Euclidean distance between two solutions to be compared is less than 0.0001 ($= \Delta$).

For the implementation of the interactive algorithms, we use two UAV Routing Problems; five-target and nine-target. The results of the interactive algorithm developed for linear underlying preference functions is presented separately for five-target and nine-target problems in the following subsections.

6.2.1 Five-Target UAV Route Planning Problem Results

The results of the interactive algorithm on five-target problem can be seen in Table 6.1. We report the most preferred solution of the RP in the first column for comparison purposes. We obtain the most preferred solution by solving bi-objective UAV route planning problem (see section 2.3) for the underlying linear preference function. In the second column, we present the interactive algorithm’s results. Furthermore, extreme solutions obtained at the end of the interactive algorithm are given in the third column. The last column gives the total number of comparisons the RP makes.

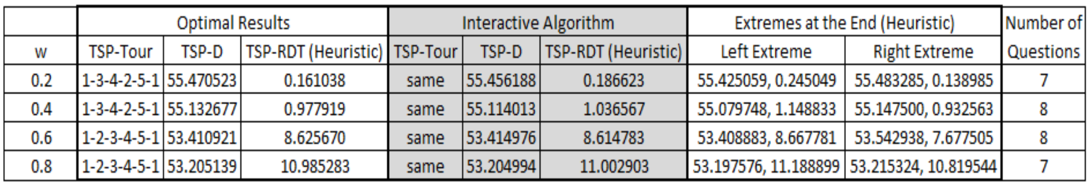
Table 6.1 Results of the Interactive Algorithm for Linear Preference Functions – 5 Target Problem



	Optimal Results			Interactive Algorithm			Extremes at the End		Number of
w	TSP-Tour	TSP-D	TSP-RDT	TSP-Tour	TSP-D	TSP-RDT	Left Extreme	Right Extreme	Questions
0.2	1-3-4-2-5-1	55.471	0.082	1-3-4-2-5-1	55.456	0.101	55.425, 0.147	55.483, 0.067	7
0.4	1-3-4-2-5-1	55.133	0.864	1-3-4-2-5-1	55.114	0.927	55.080, 1.048	55.148, 0.815	8
0.6	1-2-3-4-5-1	53.411	8.746	1-2-3-4-5-1	53.415	8.713	53.409, 8.758	53.543, 7.771	8
0.8	1-2-3-4-5-1	53.205	11.034	1-2-3-4-5-1	53.205	11.031	53.198, 11.192	53.215, 10.845	7

As it is stated above, we solve bi-objective UAV route planning formulation to find the most preferred solution of RP. Formulation uses the approximated nondominated frontiers between target pairs. Therefore, the obtained solution is actually an approximate solution. We want to find the true best solution. In Table 6.2, we take the distance measure of the tour obtained from the bi-objective UAV route planning formulation and find its corresponding radar detection threat value. We use a heuristic developed by Tezcaner Öztürk (2013) to find the real radar detection threat value. Heuristic finds the radar detection threat value of a path for a given distance between a target pair. Then, total distance and total radar detection threat values for the true best solution are obtained and given in the 'Optimal Results' column. We also find the real radar detection threat values of the tours obtained from the interactive algorithm. In other words, all radar detection threat values are updated in Table 6.2 to make a proper comparison.

Table 6.2 Results of the Interactive Algorithm for Linear Preference Functions with updated RDT values – 5 Target Problem



	Optimal Results			Interactive Algorithm			Extremes at the End (Heuristic)		Number of
w	TSP-Tour	TSP-D	TSP-RDT (Heuristic)	TSP-Tour	TSP-D	TSP-RDT (Heuristic)	Left Extreme	Right Extreme	Questions
0.2	1-3-4-2-5-1	55.470523	0.161038	same	55.456188	0.186623	55.425059, 0.245049	55.483285, 0.138985	7
0.4	1-3-4-2-5-1	55.132677	0.977919	same	55.114013	1.036567	55.079748, 1.148833	55.147500, 0.932563	8
0.6	1-2-3-4-5-1	53.410921	8.625670	same	53.414976	8.614783	53.408883, 8.667781	53.542938, 7.677505	8
0.8	1-2-3-4-5-1	53.205139	10.985283	same	53.204994	11.002903	53.197576, 11.188899	53.215324, 10.819544	7

The true most preferred solutions are (by construction) always between the extreme solutions at the final iteration. Preference value differences between the true best solution and the solution obtained by our algorithm are 0.000519, 0.000164, 0.000650, and 0.000247, for w=0.2, 0.4, 0.6, and 0.8, respectively. The respective preference value ranges between the true best and the worst nondominated solutions are 0.001841, 0.000192, 0.001249 and 0.000890, to put the performance of our algorithm in perspective. Left and right extreme solutions of the five target problem used for the scaling of the objectives are (53.181000, 11.929956) and (55.578398, 0.000000), respectively.

We demonstrate the routes that the algorithm finds for w =0.2 and 0.8 in Figure 6.1, with dashed and continuous lines, respectively.

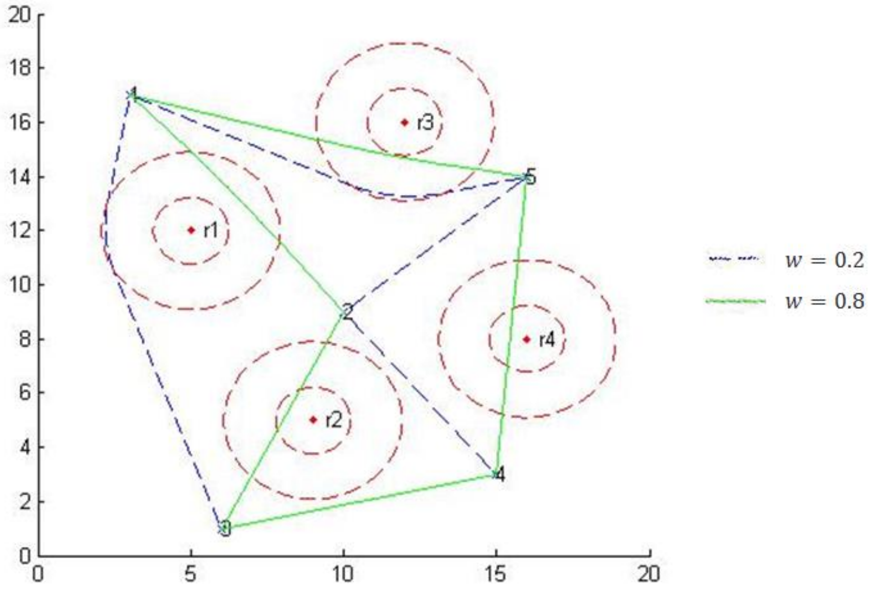


Figure 6.1 Resulting Routes for w =0.2 and 0.8, five-target problem

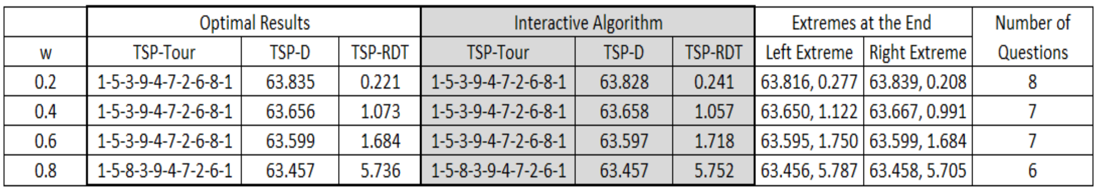
We also try different threshold values (distance and/or preference difference) and it is observed that when the threshold values decrease, the extremes at the end of the algorithm gets closer and the number of comparisons that the RP makes increases.

6.2.2 Nine-Target UAV Route Planning Problem Results

We solve the nine-target problem with the same parameter setting. We report the results for w =0.2, 0.4, 0.6, and 0.8. We scale both of the objectives between 0-1 scale using their extreme values of the route planning problem. Left and right extreme solutions of the nine target problem are (63.455953, 5.827748) and (63.963704, 0.000000), respectively.

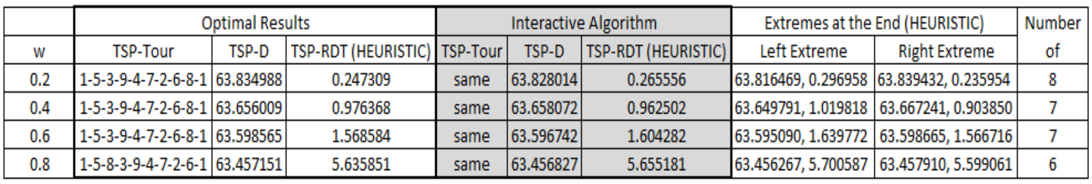
The results of the interactive algorithm can be seen in Table 6.3. In Table 6.4, we present the results with updated *RDT* values as in the 5-target case. We finalize the algorithm in a few questions for all values of w. The resulting solutions of our algorithm are very close to the true best solutions of the RP.

Table 6.3 Results of the Interactive Algorithm for Linear Preference Functions – 9 Target Problem



	Optimal Results			Interactive Algorithm			Extremes at the End		Number of
w	TSP-Tour	TSP-D	TSP-RDT	TSP-Tour	TSP-D	TSP-RDT	Left Extreme	Right Extreme	Questions
0.2	1-5-3-9-4-7-2-6-8-1	63.835	0.221	1-5-3-9-4-7-2-6-8-1	63.828	0.241	63.816, 0.277	63.839, 0.208	8
0.4	1-5-3-9-4-7-2-6-8-1	63.656	1.073	1-5-3-9-4-7-2-6-8-1	63.658	1.057	63.650, 1.122	63.667, 0.991	7
0.6	1-5-3-9-4-7-2-6-8-1	63.599	1.684	1-5-3-9-4-7-2-6-8-1	63.597	1.718	63.595, 1.750	63.599, 1.684	7
0.8	1-5-8-3-9-4-7-2-6-1	63.457	5.736	1-5-8-3-9-4-7-2-6-1	63.457	5.752	63.456, 5.787	63.458, 5.705	6

Table 6.4 Results of the Interactive Algorithm for Linear Preference Functions with updated RDT values – 9 Target Problem



	Optimal Results			Interactive Algorithm			Extremes at the End (HEURISTIC)		Number
w	TSP-Tour	TSP-D	TSP-RDT (HEURISTIC)	TSP-Tour	TSP-D	TSP-RDT (HEURISTIC)	Left Extreme	Right Extreme	of
0.2	1-5-3-9-4-7-2-6-8-1	63.834988	0.247309	same	63.828014	0.265556	63.816469, 0.296958	63.839432, 0.235954	8
0.4	1-5-3-9-4-7-2-6-8-1	63.656009	0.976368	same	63.658072	0.962502	63.649791, 1.019818	63.667241, 0.903850	7
0.6	1-5-3-9-4-7-2-6-8-1	63.598565	1.568584	same	63.596742	1.604282	63.595090, 1.639772	63.598665, 1.566716	7
0.8	1-5-8-3-9-4-7-2-6-1	63.457151	5.635851	same	63.456827	5.655181	63.456267, 5.700587	63.457910, 5.599061	6

Preference value differences between the true best solution and the solution found by our algorithm are 0.000242, 0.000198, 0.000296, and 0.000153, for w=0.2, 0.4, 0.6, and 0.8, respectively. The respective preference value ranges between the true best and the worst nondominated solutions are 0.000192, 0.001382, 0.000780, and 0.000829.

The resulting routes of the interactive algorithm for w =0.2 and 0.8 are shown together in Figure 6.2. Continuous lines correspond to w=0.2 and dashed lines represent the route found for w=0.8.

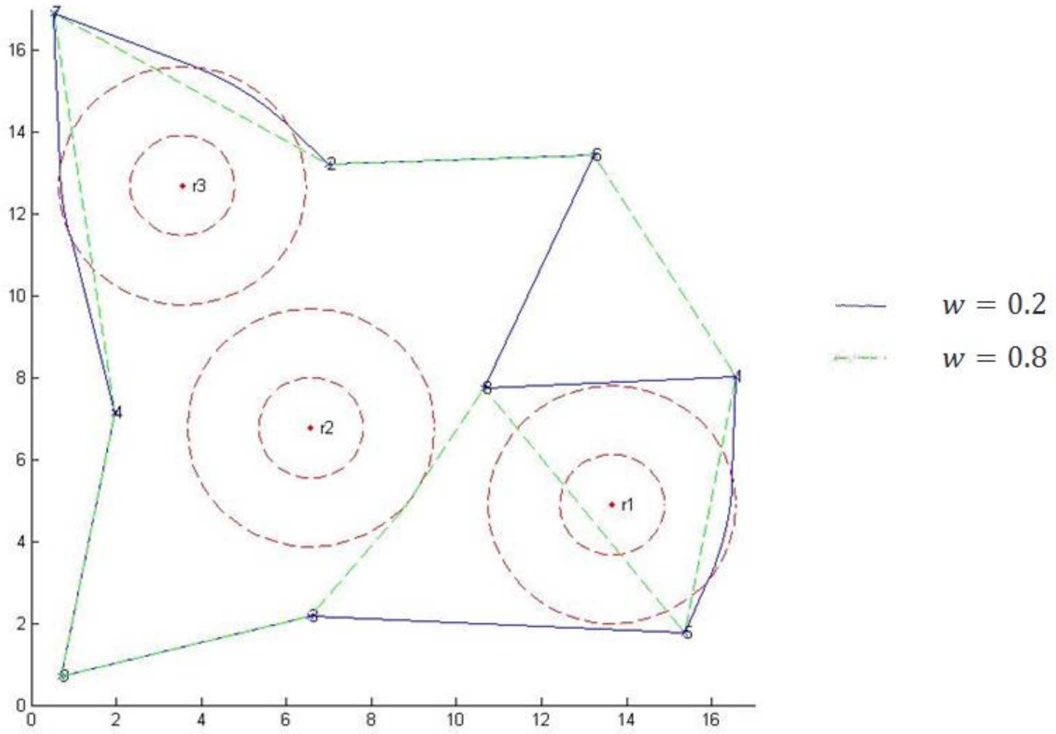


Figure 6.2 Resulting Routes for w =0.2 and 0.8, nine-target problem

Routing problem of UAVs is an NP-Hard problem. Therefore, obtaining a solution approach with reasonable computational time is important. Before each flight of UAV, we need to use the approach and find the route of the UAV. Our proposed interactive algorithm lasts less than a minute for all of these presented implementations. In UAV route planning problems, UAV visits few targets. Therefore, we do not expect to get high computational times for the algorithm.

CHAPTER 7

CONCLUSIONS

In this study, we consider the route planning problem of UAVs. The vehicle travels through a continuous terrain visiting a set of targets. We develop routes based on two objectives: minimization of distance traveled and minimization of radar detection threat. Although this problem is composed of infinitely many efficient solutions that have a different tradeoff between the two objectives, generating all these solutions is computationally demanding, and not meaningful under the presence of a RP.

We develop two interactive algorithms that find the most preferred solution of a RP. The algorithms are developed for two underlying preference function structures; linear and quasiconvex. For the linear case, we make use of the special structure of linear preference functions and make further reduction in the objective space. We always search for supported efficient solutions. In the quasiconvex case, we divide the algorithm into two stages. First stage of the algorithm is similar to the linear case. At the end of the first stage, we define rectangular regions in the objective space for the true most preferred solution. Then, we continue our search in these regions using the cone dominance idea in the second stage of the algorithm.

In Chapter 6, we demonstrate the linear algorithm on two randomly generated problems with 5 and 9 targets distributed in 400 km^2 and 289 km^2 terrains, respectively. The results show fast convergence to the most preferred regions of the RP by quickly eliminating the inferior regions. The solutions suggested by the algorithms are sufficiently close to the true most preferred solutions. As a future study, we are planning to demonstrate the interactive algorithm developed for underlying quasiconvex preference functions on 5 and 9 target problems.

There are a number of future research areas related with UAV route planning problems. In this thesis, we considered 2-D environment; we do not consider the altitude of the UAVs. When the altitude of the UAV is taken into consideration as the third dimension, we need to modify the calculations of total distance and total radar detection threat measures. Furthermore, we assume in this study, that the locations of threat areas and targets are known in advance and they are static. However, a more realistic version of this problem is that the targets and the threat areas change their locations dynamically. For this, “pop-up” threat areas can be considered and dynamic solution approach could be developed.

REFERENCES

Bahnij, R. (1985). A Fighter Pilot's Intelligent Aide for Tactical Mission Planning. *Air Force Institute of Technology*.

Bozkurt B., Fowler J.W., Gel E.S., Kim B., Köksalan M., Wallenius J. (2010). Quantitative Comparison of Approximate Solution for Multicriteria Optimization Problems with Weighted Tchebycheff Preference Function. *OPERATIONS RESEARCH*, 650-659.

Foo J.L., J. Knutzon, V. Kalivarapu, J. Olver and E. Winer. (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.

Gudaitis, M. (1994). Multicriteria Mission Route Planning Using a Parallel A* Search. *Air Force Institute of Technology*.

Kan E., M. Lim, S. Yeo, J. Ho and Z. Shao. (2011). Contour Based Path Planning with B-Spline Trajectory Generation for UAVs over Hostile Terrain. *Journal of Intelligent Learning Systems and Applications 3*, 122-130.

Korhonen P., Wallenius J., Zionts S. (1984). Solving the Discrete Multiple Criteria Problem using. *Management Science 30 (1)*, 1336-1345.

Köksalan, M. (1999). A Heuristic Approach to Bicriteria Scheduling. *Naval Research Logistics 46*, 777-789.

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

Lokman B., Köksalan M., Korhonen P. (2016). An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Annals of Operations Research*, 67-95.

Olsan, J. (1993). Genetic Algorithms Applied to a Mission Routing Problem. *Air Force Institute of Technology*.

Öztürk, D. T. (2013). HEURISTIC AND EXACT APPROACHES FOR MULTI-OBJECTIVE ROUTING.

Pachter M. and J. M. Hebert. (2002). Minimizing Radar Exposure in Air Vehicle Path Planning. *Proc. of the 15th Triennial World Congress, Barcelona, Spain*.

Pohl A.J. and G.B. Lamont. (2008). Multi-Objective UAV Mission Planning Using Evolutionary Computation. *Proceedings of the 2008 Winter Simulation Conference*.

Tezcaner D., Köksalan M. (2011). An Interactive Algorithm for Multi-objective Route. *Journal of Optimization Theory and Applications (150)*, 379–394.

Tezcaner Öztürk D., Köksalan M. (2016). An interactive approach for biobjective integer programs under quasiconvex preference functions. *Ann Oper Res (2016) 244*, 677–696.

Yavuz, K. (2002). Mission Route Planning Using Particle Swarm Optimization. *M.S. thesis, Air Force Institute of Technology*.

Zheng C., M. Ding and C. Zhou. (2003). Real-Time Route Planning for Unmanned Air Vehicle with an Evolutionary Algorithm. *International Journal of Pattern Recognition and Artificial Intelligence, 17 (1)*, 68-81.

Zheng, C., L. Li, F. Xu and F. Sun. (2005). Evolutionary Route Planner for Unmanned Air Vehicles. *IEEE Transactions on Robotics*, 609-620.

APPENDICES

APPENDIX A

COMPUTATION OF THE OBJECTIVES

Formulas given in this part are directly taken from Tezcaner Öztürk (2013)'s study.

A.1. Distance Calculation

Total distance from an initial point (x_s, y_s) to destination point (x_f, y_f) is calculated as follows:

$$D = \int_{(x_s,y_s)}^{(x_f,y_f)} ds \qquad \text{(A.1)}$$

In this formula, ds corresponds to the infinitesimal part of a path traveled. In other words, total distance is the summation of infinitely small parts of the movement.

A.2. Radar Detection Threat Calculation

First of all, signal to noise ratio (*S/N*) of a point (x, y) is calculated by using the formula below (A.2). All parameters, except R, are constant throughout the terrain. Therefore, we combine all constants under C, and reduce the formula as follows:

$$S/N_{(x,y)} = 10log\left(\frac{P_t {G_t}^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n {L_t}^2 R^4}\right) = 10log\left(\frac{C}{R^4}\right) \qquad \text{(A.2)}$$

P_t: Power transmitted by radar (Watts)

G_t: Power gain of transmitting antenna

L_t: Transmitting system loss

λ: Wave lenght of signal frequency (Meters)

T_s: Receive system noise temperature (Kelvin)

B_n:Noise bandwidth of receiver (Hertz)

K: Boltzman's constant (Joules/Kelvin)

σ: Aircraft radar cross section (RCS) (Square Meters)

R: Distance from the transmitter to aircraft's location (x, y) (Meters)

Then, depending on the value of $S/N_{(x,y)}$, probability of detection of a point $(pd_{(x,y)})$ is found as in A.3.

$$pd_{(x,y)} = \begin{cases} 1 & if\ S/N_{(x,y)} > UB_{S/N} \\ \frac{S/N_{(x,y)} - LB_{S/N}}{UB_{S/N} - LB_{S/N}} & if\ LB_{S/N} < S/N_{(x,y)} \leq UB_{S/N} \\ 0 & if\ S/N_{(x,y)} \leq LB_{S/N} \end{cases} \qquad \text{(A.3)}$$

Total radar detection threat, RDT, between points (x_s, y_s) and (x_f, y_f) is the summation of all the detection probabilities over the trajectory.

$$RDT = \int_{(x_s,y_s)}^{(x_f,y_f)} pd_{(x,y)} ds \qquad \text{(A.4)}$$

APPENDIX B

CURVED MOVEMENT INSIDE THE OUTER RADAR REGION

Tezcaner Öztürk (2013) assumes that UAV follows a circular path inside the outer radar region due to tradeoff between objectives. As it is illustrated on the below figure, center of the circular move is at (a, b) and the radius of the circle is r. The equation of the circle is then $(x - a)^2 + (y - b)^2 = r^2$.

For each efficient trajectory between a target pair, entrance point to radar region changes so does the circular movement of UAV. In other words, circle equation of movement is different for each efficient trajectory. For further information on calculations of entrance-exit points and equation of circular moves, please see Tezcaner Öztürk's study.

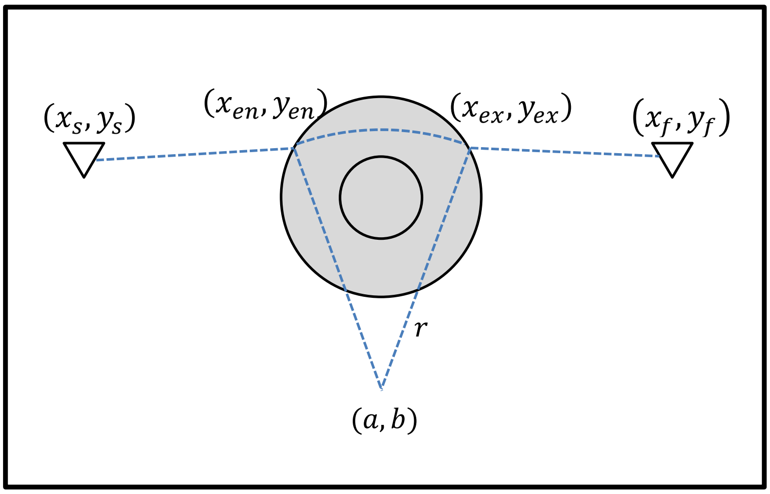


Figure B.1 Circular Move Inside Outer Radar Region

APPENDIX C

TRAVELING SALESPERSON PROBLEM

Formulation of the single objective TSP with n nodes is given below. c_{ij} denotes the distance between nodes i and j and x_{ij} states whether a connection between node i and j is used or not.

Min $\quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,..,n$

$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,..,n$

$u_i - u_j + n x_{ij} \leq n - 1 \qquad i = 2,..,n, j = 2,..,n$ and $i \neq j$

$x_{ij} \in \{0,1\} \qquad i = 1,..,n, j = 1,..,n$

$0 \leq u_i \qquad i = 2,..,n$

APPENDIX D

FIVE-TARGET PROBLEM REGION

Five-target problem is developed by Tezcaner Öztürk (2013). She randomly placed five targets and four radars in a 400 km^2 terrain as shown in the below figure.

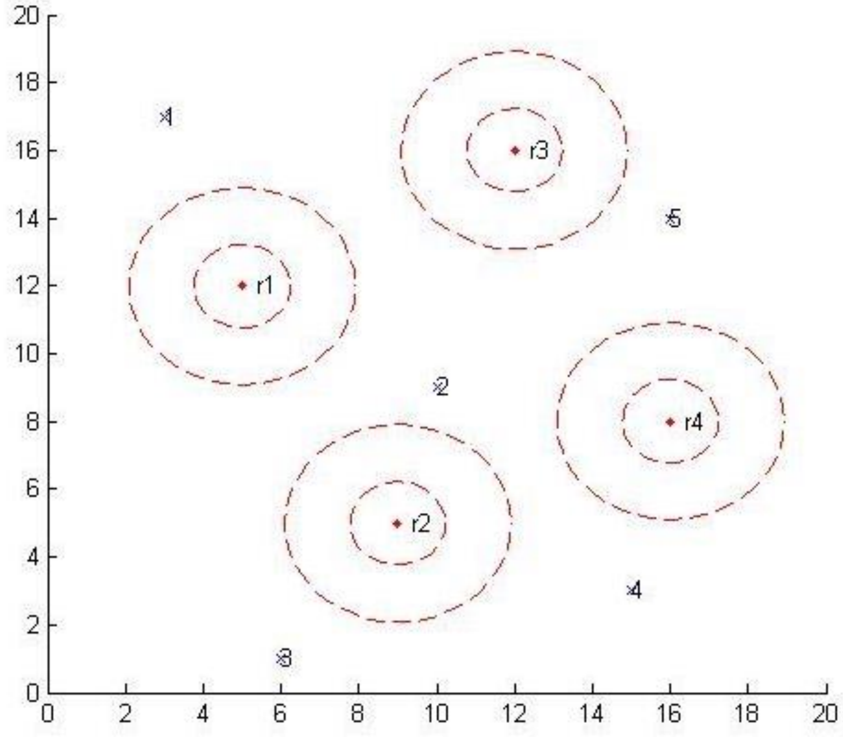


Figure D.1 Five-target problem region

APPENDIX E

NINE-TARGET PROBLEM REGION

Nine-target problem region generated by the method explained in Section 6.1 is given in the figure below. There are 9 targets and 3 radar areas in 289 km^2 terrain. We locate the radars and targets such that each radar is responsible from surveillance of three targets, and is therefore located close to those targets.

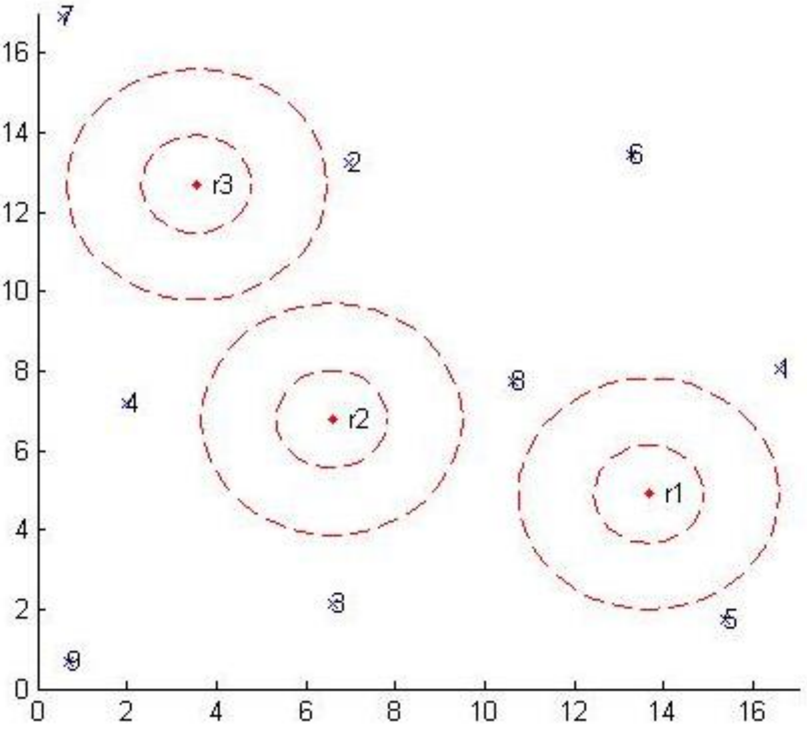


Figure E.1 Nine-target problem region