

HEURISTIC AND EXACT APPROACHES FOR MULTI-OBJECTIVE ROUTING

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

BY

DİCLEHAN TEZCANER ÖZTÜRK

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

NOVEMBER 2013

Approval of the thesis:

**HEURISTIC AND EXACT APPROACHES FOR MULTI-OBJECTIVE
ROUTING**

submitted by **DİCLEHAN TEZCANER ÖZTÜRK** in partial fulfillment of the
requirement for the degree of **Doctor of Philosophy in Industrial Engineering
Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
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** _____

Examining Committee Members

Assoc. Prof. Dr. Esra Karasakal
Industrial Engineering Dept., METU _____

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

Assoc. Prof. Dr. Hande Yaman
Industrial Engineering Dept., Bilkent University _____

Assoc. Prof. Dr. Sinan Gürel
Industrial Engineering Dept., METU _____

Assist. Prof. Dr. Cem İyigün
Industrial Engineering Dept., METU _____

Date: _____ **27.11.2013** _____

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 : Diclehan TEZCANER ÖZTÜRK

Signature :

ABSTRACT

HEURISTIC AND EXACT APPROACHES FOR MULTI-OBJECTIVE ROUTING

TEZCANER ÖZTÜRK, Diclehan

Ph.D., Department of Industrial Engineering

Supervisor: Prof. Dr. Murat Köksalan

November 2013, 134 pages

In this thesis, we consider the bi-objective routing problem. This problem is a combination of the bi-objective shortest path problem (finding the efficient arcs between nodes) and the bi-objective traveling salesperson problem (finding the efficient tours composed of efficient arcs). We develop solution procedures to find efficient tours. We consider two different terrain structures; discretized terrain and continuous terrain. In the discretized terrain, the terrain is approximated with grids and we allow the vehicle to move between adjacent grid points. In the continuous terrain, we consider a two dimensional plane and there are no restrictions on the movement of the vehicle.

To find the most preferred solution, we first develop a general interactive algorithm for a decision maker whose preferences are consistent with an underlying quasiconvex function to be minimized for any bi-criteria integer program. We then apply our algorithm to the bi-objective routing problem. In each iteration of the algorithm, we find a number of efficient tours made up of the efficient arcs. For this, we initially find

all efficient arcs between all pairs of nodes and introduce these arcs to the interactive algorithm. We establish some rules that decrease the number of efficient arcs required for finding an efficient tour that satisfies some constraints. We demonstrate the approach on the routing problem of unmanned air vehicles (UAVs) which are assumed to travel on a discretized terrain. We also study the routing problem in continuous terrain, specifically for the UAV routing problem. We develop methods to find the approximate efficient frontier of the shortest path problem between each node pair. We then find the efficient tours that use a subset of the efficient arcs using a mixed integer nonlinear program. We also discuss the implementation of the interactive algorithm for the routing problem in the continuous terrain.

Keywords: Bi-objective Routing, Combinatorial, Interactive, Unmanned Air Vehicles

ÖZ

ÇOK AMAÇLI ROTALAMA İÇİN SEZGİSEL VE KESİN YAKLAŞIMLAR

TEZCANER ÖZTÜRK, Diclehan

Doktora, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Murat Köksalan

Kasım 2013, 134 Sayfa

Bu tezde, iki amaçlı rotalama problemini ele alıyoruz. Bu problem iki amaçlı en kısa yolu bulma problemi (hedefler arası etkin yolların bulunması) ve iki amaçlı gezgin satıcı probleminin (etkin yollardan oluşan etkin turların bulunması) birleşimidir. Etkin turların bulunması için çözüm yolları geliştirdik. Rotalama problemini iki farklı hareket alanında inceledik; karesel parçalara bölünmüş ayrık hareket alanı ve sürekli hareket alanı. Ayrık hareket alanında, hareket bölgesini eş büyüklükteki karelerle tanımladık. Aracın komşu noktalar arasında hareketine izin verdik. Sürekli hareket alanı durumunda, aracın iki boyutlu uzayda her türlü hareketi yapabileceği kabul edildi.

En çok tercih edilen çözümü bulmak için öncelikle minimize edilecek tercih fonksiyonu konveks benzeri olan bir karar verici için iki amaçlı tamsayı programlamada kullanılabilir genel bir interaktif algoritma geliştirdik. Sonrasında bu algoritmayı iki amaçlı rotalama probleminde uyguladık. Algoritma her iterasyonunda, birkaç etkin tur bulmaktadır. Bu etkin turları bulmak için, öncelikle hedef çiftleri arasındaki tüm etkin yolları bulduk ve bu yolları algoritmaya girdi olarak

verdik. Kullanılabilecek etkin yolların sayısını azaltmak için, amaç fonksiyonları üzerinde kısıtlar olduđu durumda bazı kurallar geliřtirdik. Yaklařımımızı, ayrıık hareket alanındaki insansız hava aracı (İHA) rotalama probleminde denedik. Ayrıca, sürekli hareket alanında rotalama problemini İHA rotalama problem özelinde inceledik. Her hedef çifti arasındaki etkin yolları bulmak için yöntemler geliřtirdik. Sonrasında, etkin yollardan oluřan etkin turları karıřık tamsayılı doğrusal olmayan programlama ile bulduk. Buna ek olarak, interaktif algoritmanın sürekli hareket alanındaki rotalama problemine nasıl uygulanabileceğini ele aldık.

Anahtar Kelimeler: İki amaçlı rotalama, Kombinatoryal, İnteraktif, İnsansız Hava Aracı

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor Prof. Dr. Murat Köksalan. I am grateful to him for his support and motivation throughout my Ph.D. study and for his guidance in this study and academic life.

I am indebted to my parents, Olcay Tezcaner and Tevfik Tezcaner, and my brother, Batuhan Tezcaner, for their endless patience, supportive point of views and motivation.

I would like to thank my friends Gülşah Karakaya and Ceren Tuncer Şakar for sharing all the great moments during my Ph.D. study.

I would like to express my gratitude to my examining committee members; Assoc. Prof. Dr. Hande Yaman, Assoc. Prof. Dr. Esra Karasakal, Assoc. Prof. Dr. Sinan Gürel and Assist. Prof. Dr. Cem İyigün for their valuable comments.

I would like to thank TÜBİTAK for the funding they have provided during my Ph.D. study.

My deepest appreciation goes to Yalın. Without him, neither this study nor my life would be complete.

To my family...

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ.....	vii
ACKNOWLEDGEMENTS	ix
TABLE OF CONTENTS	xi
LIST OF FIGURES.....	xiii
LIST OF TABLES	xv
CHAPTERS	
1 INTRODUCTION	1
2 LITERATURE SURVEY.....	3
3 AN INTERACTIVE ALGORITHM FOR QUASICONVEX PREFERENCE FUNCTIONS	7
3.1 An Interactive Algorithm	9
3.2 The Steps of the Interactive Algorithm.....	13
3.3 Finding Adjacent Efficient Solutions.....	16
4 APPLICATION OF THE INTERACTIVE ALGORITHM TO THE BI- OBJECTIVE ROUTING PROBLEM.....	23
4.1 Finding Efficient Arcs.....	26
4.2 Finding Efficient Solutions in the Original and Reduced Objective Spaces	27
4.3 Arc Reduction	28
4.4 Additional Constraints in Search for Unsupported Efficient Solutions	32
4.5 Improvements in the Interactive Algorithm and the Arc Reduction Approach.....	37
4.6 Alternatives for Solving the Constrained TSP and Finding Lower Bounds in Arc Reduction.....	41
4.7 Finding Efficient Arcs during the Interactive Algorithm.....	42

4.8	Application to the UAV Routing Problem.....	44
4.9	An Example.....	48
5	ROUTING IN CONTINUOUS SPACE.....	53
5.1	Terrain Structure and Objectives.....	54
5.2	Transforming the Terrain Structure.....	58
5.3	Movement between Two Targets.....	59
5.4	Enumeration of Movement between Two Targets.....	64
5.5	Generating the Efficient Frontier for the Bi-Objective Shortest Path Problem.....	74
5.6	Finding D_{tan}	80
5.7	Implementation.....	82
5.8	A Heuristic to Find the Efficient Solution Corresponding to a Distance D	84
5.9	Approximating the Efficient Frontier.....	96
5.10	Generating the Efficient Frontier for the Bi-Objective Routing Problem.....	98
5.11	Comparison of Solutions for Discretized and Continuous Terrain.....	103
5.12	Integrating the Bi-Objective Routing Problem in Continuous Terrain to the Interactive Algorithm.....	105
6	CONCLUSIONS.....	109
	REFERENCES.....	113
	APPENDICES	
A.	MARTINS ALGORITHM.....	119
B.	COMPUTATIONAL RESULTS FOR THE INTERACTIVE ALGORITHM ON BI-OBJECTIVE ROUTING PROBLEM.....	121
C.	REGRESSION ANALYSIS FOR x_m vs. y_m	125
D.	METHODS TO APPROXIMATE INTEGRALS.....	129
E.	FIBONACCI SEARCH METHOD.....	131
	CURRICULUM VITAE.....	133

LIST OF FIGURES

Figure 3.1 Admissable Regions for the Most Preferred Solution	12
Figure 3.2 Demonstration of the Interactive Algorithm.....	16
Figure 3.3 Finding Adjacent Efficient Solutions – Procedure 1	19
Figure 3.4 Finding Adjacent Efficient Solutions – Procedure 2	21
Figure 4.1 An Example MOTSP	24
Figure 4.2 Bounds on the Objectives in the Reduced Objective Space	29
Figure 4.3 The Two Phase Algorithm.....	33
Figure 4.4 Additional Constraint for Second Phase.....	34
Figure 4.5 Additional Constraints for the Second Phase	36
Figure 4.6 The Bounds on Objectives after Steps 5 and 6.....	38
Figure 4.7 Bounds on the Objectives in the Reduced Objective Space	40
Figure 4.8 Terrain Structure for UAV Routing Problem	45
Figure 4.9 Movement Representation in two Dimensional Terrain.....	45
Figure 4.10 Efficient Frontier of the UAV Routing Problem	48
Figure 5.1 UAV Routing Problem in Continuous Terrain.....	55
Figure 5.2 Continuous Terrain Representation	57
Figure 5.3 Movement Inside the Outer Radar Region	60
Figure 5.4 Movement Passing through the inner Radar Region	62
Figure 5.5 First Extreme Efficient Solution.....	66
Figure 5.6 The Second Extreme Efficient Solution	67
Figure 5.7 Coordinates of a Point on the Radar Circle	69
Figure 5.8 Efficient Frontier of the Example Shortest Path Problem	73
Figure 5.9 Graphs of $f(x)$ versus x for different (a,b,r) values	78
Figure 5.10 Line of Middle Points of Efficient Solutions.....	84
Figure 5.11 RDT vs. y_m Corresponding to Different D Values.....	86
Figure 5.12 Finding y_{max} in Step 4	88

Figure 5.13 $G(r)$ vs. r	90
Figure 5.14 An Example Efficient Frontier Approximated for Nodes (a,b) in E_{both}	99
Figure 5.15 An Example UAV Routing Problem	104
Figure 5.16 Efficient Frontier for the Arcs Between Targets 4 and 5	104
Figure 5.17 Efficient Frontier for the Tours	105
Figure 5.18 An Example Efficient Frontier for the MOTSP	107
Figure 5.19 Most preferred region of the DM.....	108

LIST OF TABLES

Table 4.1 Efficient solutions	49
Table 4.2 Arc reduction for $z_1 \leq 32.140-\varepsilon$ and $z_2 \leq 12.764$	50
Table 4.3 The upper bounds on the combined objective and the reduced arcs.....	51
Table 5.1 Approximation with different methods.....	79

CHAPTER 1

INTRODUCTION

We study the multi-objective route planning problem in which we search for efficient tours that start from an initial point, visit all the nodes and return to the initial point again. This problem has two parts: the first part constitutes the efficient arcs between node pairs. This is a multi-objective shortest path problem (MOSPP). The second part corresponds to the identification of efficient tours that use a subset of the efficient arcs. The overall problem can be considered as a multi-objective traveling salesperson problem (MOTSP).

We develop solution procedures to find efficient tours made up of the efficient arcs. We consider finding efficient solutions both in the original objective space (that is not constrained with bounds on objectives) and in reduced objective space (that is constrained with bounds on objectives). We analyze two types of the routing problem; routing in a discretized terrain and routing in a continuous terrain. In the discretized terrain, the terrain is approximated with grids and we allow for movement between adjacent grid points. In the continuous terrain, we consider a two dimensional plane and the vehicle can make any move on this plane.

To find the most preferred solution, we first develop a general interactive algorithm for bi-objective integer programming where the decision maker (DM) has an underlying quasiconvex value function to be minimized. In the literature, quasiconvex value functions are widely used since they represent a large set of preference functions. We apply the interactive algorithm to the bi-objective route planning problem. For the

routing problem in discretized terrain, we find all the efficient arcs utilizing Martin's algorithm (see Gandibleux et al., 2006), an algorithm developed for solving MOSPP. We introduce these efficient arcs to the MOTSP to find efficient tours. We develop rules that combine the solution process for the bi-objective shortest path problem (SPP) with that of the bi-objective traveling salesperson problem (TSP). We demonstrate the interactive algorithm and our rules on the bi-objective routing problem for unmanned air vehicles (UAVs). We first find the efficient arcs between nodes in the two dimensional terrain. We then search for the efficient tours composed of a subset of efficient arcs. Each time we search for a new tour, we reduce the efficient arcs with the rules we develop.

We analyze the routing problem in continuous terrain for the objectives specifically developed for the UAV routing problem. In this problem, we find the efficient routes for the UAVs in the two dimensional plane under two objectives: minimization of total distance traveled and minimization of total radar detection threat. We study the case where there is a single radar between node pairs. We develop a movement model for the UAV inside the radar region and derive properties for efficient arcs. Using the properties of the objectives, we establish both exact and heuristic methods to structure the efficient frontiers of the arcs between node pairs. Using the approximated efficient arcs, we reach efficient tours using a bi-objective nonlinear programming model. We also integrate the continuous terrain routing problem in the interactive algorithm we develop.

We review the literature and define the problem in Chapter 2. We explain the interactive algorithm in Chapter 3 and the application of the bi-objective routing problem in Chapter 4. We present the routing problem in continuous terrain and our solution approaches in Chapter 5. We present our conclusions and future work in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

The multi-objective routing problem can be considered as a MOTSP with multiple efficient arcs between nodes. There are a number of studies on MOTSP in the literature. However, most of these studies assume that the nodes in MOTSP are connected with a single efficient arc. This is not a realistic assumption, since there would typically be a number of efficient arcs each better than the others in at least one objective, under the presence of multiple objectives. For instance, consider a problem in which we search for the routes between an initial point and a termination point. We travel by railways and we want to minimize the total duration of travel, the total cost of travel and total number of train interchanges. One route may have the least duration with the least number of interchanges, while having a high cost. The least cost route may have a moderate number of train interchanges and the highest duration. Finding a single arc between the initial and the termination points that has the best value in all of these criteria is not possible for most of the cases. Therefore, we refer to the case where each node is connected by a number of efficient arcs as the generalized MOTSP. The MOTSP with a single efficient connection between nodes is a special case of the generalized MOTSP. In this study, we consider the route planning problems as a generalized MOTSP. The generalized MOTSP is also studied in Tezcaner and Köksalan (2011).

MOTSP with a single efficient connection between nodes is NP-hard (Ehr Gott, 2000). Mostly, heuristics have been developed for the solution of this problem. Lust and Teghem (2010) classify these solution approaches and develop a new method, two-

phase pareto local search, for MOTSP. Paquete and Stützle (2009) develop a two-phase local search method and Jaszkiwicz and Zielniewicz (2009) consider a Pareto memetic algorithm using path relinking and Pareto local search. Jozefowicz et al. (2008), Karademir (2008) and Berube et al. (2009) consider a special traveling salesperson problem (TSP), TSP with profits. Jozefowicz et al. construct the efficient frontier by an evolutionary algorithm (EA), Karademir proposes a genetic algorithm and Berube generates the efficient frontier using the ε -constraint method (see Chankong and Haimes, 1983). Hansen (2000) uses a scalarizing function to solve MOTSP. Special, polynomially solvable cases of MOTSP are studied in Özpeynirci and Köksalan (2009, 2010).

We demonstrate our approaches on the routing problem for UAVs. UAVs are unpiloted air vehicles. They were first designed for military purposes. Currently they are also used for civilian purposes like surveillance against crimes, crop spraying, fire prevention etc.

Most of the literature on UAV routing resort to heuristics. Zheng et al. (2003) consider the routing problem of a UAV with an initial point and a destination. They develop an evolutionary algorithm (EA) for this problem under a single nonlinear objective function that is a weighted combination of three objectives (total distance traveled, total altitude of the route, closeness to threat sites). Foo et al. (2009) consider the same problem with three objectives (arc length, deviation from reconnaissance and violation of threat zones) linearly combined. They generate a number of alternative arcs by changing the weights assigned to each objective in the weighted composite objective function using particle swarm optimization and b-splines. Pohl and Lamont (2008) and Zheng et al. (2005) consider routing multiple UAVs. Pohl and Lamont use three EAs in which they treat three objectives (total distance traveled, total waiting times of UAVs and the number of UAVs used) separately. Zheng et al. solve the problem with an EA that considers constraints on turn angle, tour length etc.

Gudaitis (1994) and Olsan (1993) consider the routing problem with a single destination. They both consider the same two objectives, minimization of distance traveled and minimization of radar detection threat. Combining these objectives linearly under a single objective, Gudaitis uses the A* algorithm and Olsan (1993) uses a genetic algorithm for the solution. Yavuz (2002) considers the problem of routing with multiple nodes. He uses the same two objectives and employs Particle Swarm Optimization and Ant System for the solution. Jia and Vagners (2004) consider the routing problem of a single UAV visiting multiple sites and finally reaching a goal position. They run an EA in parallel a number of times to escape local optima.

In this study, we solve the bi-objective routing problem using exact methods. Unlike the methods in the literature, we consider the two contained problems, generalized MOTSP and MOSPP together. We develop the necessary theory and solution methods that address the issues in these interrelated problems. We also establish problem specific solution approaches for the terrain types, discretized and continuous.

CHAPTER 3

AN INTERACTIVE ALGORITHM FOR QUASICONVEX PREFERENCE FUNCTIONS

We develop an interactive algorithm that finds the most preferred solution of a decision maker (DM) who is assumed to have an underlying quasiconcave (quasiconvex) value function to be maximized (minimized). In the literature, quasiconcave utility functions (with objectives to be maximized) are widely used since they cover a larger set of preference functions including linear preference functions. These functions are considered to represent the human behavior well. As one criterion gets better, to further improve that criterion, the amount of sacrifice from other criteria decreases. In other words, the marginal rate of substitution decreases for these preference functions. In the literature, there are a number of studies on multi-objective interactive algorithms. Lokman et al. (2011) develop an interactive algorithm that finds the most preferred solution of a DM whose underlying preference function is quasiconcave. This algorithm is applicable for multi-objective integer programs. Tezcaner and Köksalan (2011) developed an algorithm, *BestSol*, for bicriteria discrete optimization problems to find the most preferred solution of a DM whose preferences are consistent with a linear utility function. The algorithm we develop in this thesis addresses bi-objective integer programming problems with general quasiconcave preference functions.

Before we present the problem in detail, we give the necessary definitions. These definitions are directly taken from Tezcaner and Köksalan (2011), and we repeat them here for the sake of completeness.

Let x denote the decision variable vector, X denote the feasible space, Z denote the corresponding feasible objective function space, and $z(x) = (z_1(x), z_2(x), \dots, z_p(x))$ be the objective function vector. We assume, without loss of generality, that all objectives are to be minimized.

Definition 3.1 A solution $x \in X$ is said to be *efficient* iff there does not exist $x' \in X$ such that $z_j(x') \leq z_j(x)$ $j = 1, \dots, p$ and $z_j(x') < z_j(x)$ for at least one j . If there exists such an x' , x is said to be *inefficient*. The set of all efficient solutions constitute the efficient set.

Definition 3.2 If x is efficient, then $z(x)$ is said to be *nondominated*, and if x is inefficient, $z(x)$ is said to be *dominated*.

Definition 3.3 An efficient solution x is a *supported efficient solution* iff there exists a positive linear combination of objectives that is minimized by x . Otherwise, x is an *unsupported efficient solution*.

Definition 3.4 An *extreme efficient solution* is a supported efficient solution that has the minimum possible value in at least one of the objectives.

Definition 3.5 Two solutions are *adjacent efficient solutions* iff none of their convex combinations are dominated by any convex combinations of other solutions in the objective function space. That is, x_j is adjacent efficient to x_i iff there does not exist $x_t \in X \ni \sum_{t \neq j} \mu_t z(x_t) \leq \lambda z(x_j) + (1 - \lambda)z(x_i)$ where $\sum_{t \neq j} \mu_t = 1$, $0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$, and $z(x)$ is the objective function vector. In a bicriteria problem, there are at most two distinct adjacent efficient solutions of a solution (Ramesh et al., 1990).

Definition 3.6 f is a quasiconvex function if $f(\sum_{i=1}^p \mu_i x_i) \leq \max_i f(x_i)$ for $\sum_{i=1}^p \mu_i = 1$, $\mu_i \geq 0$.

3.1 An Interactive Algorithm

In multi-objective problems, there are typically many efficient solutions. The DM may not be interested in many of those solutions. Converging the preferred solutions quickly is important. Interactive approaches aim to do this by progressively obtaining preference information from a DM. With our interactive approach, we generate a small subset of the efficient frontier. We select and provide the DM a pair of solutions and ask for the preferred one. This procedure is repeated and the preferences of the DM lead our search to the most preferred solution of the DM.

We assume that the DM's preferences are consistent with a quasiconvex preference function to be minimized. In the literature, quasiconcave utility functions (with objectives to be maximized) are widely used. If all objectives are minimization type, the same theory directly applies. We use the following lemma (adapted from Lemma 1 of Korhonen et al., 1984) for the quasiconvex case to reduce the objective space eliminating the implied inferior regions based on the expressed preferences of the DM:

Lemma 1. 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 $f(z) \geq f(x_k)$.

Proof. The proof directly follows from Korhonen et al.'s (1984) proof.

We use the following lemma to find the regions in the objective space that do not contain any efficient solutions using the information that two solutions are adjacent efficient solutions.

Lemma 2. Let x_k and x_r be two adjacent efficient solutions, $k \neq r$. Then, there does not exist any x_i , $i \neq k$, $i \neq r$, such that x_i dominates some affine combination of x_k and x_r .

Proof. It is given in Definition 3.5 that x_r is adjacent efficient to x_k iff there does not exist $x_t \in X \ni \sum_{t \neq r} \mu_t z(x_t) \leq \lambda z(x_r) + (1 - \lambda)z(x_k)$ where $\sum_{t \neq r} \mu_t = 1$, $0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$, and $z(x)$ is the objective function vector.

Similarly, x_k is adjacent efficient to x_r iff there does not exist $x_t \in X \ni \sum_{t \neq k} \mu_t z(x_t) \leq \lambda z(x_k) + (1 - \lambda)z(x_r)$ where $\sum_{t \neq k} \mu_t = 1$, $0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$.

Suppose there exists a solution x_a dominating some affine combination of x_k and x_r .

$$z(x_a) \leq \lambda_r z(x_r) + \lambda_k z(x_k) \text{ for some } \lambda_k, \lambda_r \in R \text{ and } \lambda_k + \lambda_r = 1. \quad (3.1)$$

$$\text{This reduces to: } z(x_a) - z(x_k) \leq \lambda_r (z(x_r) - z(x_k)) \text{ for some } \lambda_r \in R. \quad (3.2)$$

$$\text{Similarly, } z(x_a) - z(x_r) \leq \lambda_k (z(x_k) - z(x_r)) \text{ for some } \lambda_k \in R. \quad (3.3)$$

Rearranging the terms in $\sum_{t \neq r} \mu_t z(x_t) \leq \lambda z(x_r) + (1 - \lambda)z(x_k)$, we obtain the following inequality:

$$\sum_{t \neq r} \mu_t z(x_t) - z(x_k) \leq \lambda (z(x_r) - z(x_k)) \quad 0 < \lambda \leq 1, \sum_{t \neq r} \mu_t = 1, \quad 0 \leq \mu_t \leq 1 \quad (3.4)$$

Similarly, rearranging the terms in $\sum_{t \neq k} \mu_t z(x_t) \leq \lambda z(x_k) + (1 - \lambda)z(x_r)$, we obtain the following inequality:

$$\sum_{t \neq k} \mu_t z(x_t) - z(x_r) \leq \lambda (z(x_k) - z(x_r)) \quad 0 < \lambda \leq 1, \sum_{t \neq k} \mu_t = 1, \quad 0 \leq \mu_t \leq 1 \quad (3.5)$$

If we set $\mu_k = 1 - \mu_a$ for $\mu_a > 0$ and $\mu_t = 0$ for $t \neq k, a$ in (3.4), we obtain the following inequality:

$$\begin{aligned} \mu_a z(x_a) + (1 - \mu_a)z(x_k) - z(x_k) &\leq \lambda (z(x_r) - z(x_k)) \quad 0 < \lambda, \mu_a \leq 1 \\ z(x_a) - z(x_k) &\leq \frac{\lambda}{\mu_a} (z(x_r) - z(x_k)) \quad 0 < \lambda, \mu_a \leq 1 \end{aligned} \quad (3.6)$$

Similarly, if we set $\mu_r = 1 - \mu_a$ for $\mu_a > 0$ and $\mu_t = 0$ for $t \neq r, a$ in (3.5), we obtain the following inequality:

$$\begin{aligned} \mu_a z(x_a) + (1 - \mu_a)z(x_r) - z(x_r) &\leq \lambda (z(x_k) - z(x_r)) \quad 0 < \lambda, \mu_a \leq 1 \\ z(x_a) - z(x_r) &\leq \frac{\lambda}{\mu_a} (z(x_k) - z(x_r)) \quad 0 < \lambda, \mu_a \leq 1 \end{aligned} \quad (3.7)$$

In (3.6) and (3.7), $\frac{\lambda}{\mu_a}$ can only be positive for $0 < \lambda, \mu_a \leq 1$ and get any value in \mathbb{R}^+ .

Either λ_k or λ_r should be positive to satisfy $\lambda_k + \lambda_r = 1$ in (3.1). Therefore, if $\lambda_r > 0$ in (3.2), we satisfy (3.6) for some $0 < \lambda, \mu_a \leq 1, \lambda = \lambda_r \mu_a$ that contradicts with x_r

being adjacent efficient to x_k . If $\lambda_k > 0$ in (3.3), we satisfy (3.7) for some $0 < \lambda, \mu_a \leq 1, \lambda = \lambda_k \mu_a$ that contradicts with x_k being adjacent efficient to x_r . #

In the following lemma, we find the representation of the reduced objective space when we know that a solution is preferred to its two adjacent efficient solutions.

Lemma 3. Let $z(x_i) = (z_1(x_i), z_2(x_i))$ be the objective function vector of solution x_i . Let x_a and x_c be the left and right adjacent efficient solutions of the supported efficient solution x_b . Let x_b be preferred to both x_a and x_c . Let $n(x_i, x_j) = (\max(z_1(x_i), z_1(x_j)), \max(z_2(x_i), z_2(x_j)))$. The most preferred solution of the DM, x^* should be inside the convex region $Z^* =$

$$\{z | z = \mu_a z(x_a) + \mu_b z(x_b) + \mu_{ab} z(n(x_a, x_b)), \mu_a + \mu_b + \mu_{ab} = 1, \mu_a, \mu_b, \mu_{ab} \geq 0\} \cup \{z | z = \mu_c z(x_c) + \mu_b z(x_b) + \mu_{cb} z(n(x_c, x_b)), \mu_c + \mu_b + \mu_{cb} = 1, \mu_c, \mu_b, \mu_{cb} \geq 0\}.$$

Proof. If solution x_b and solutions x_a and x_c are adjacent efficient solutions, there should not be any solution dominating any affine combinations of the adjacent efficient solutions due to Lemma 2. This eliminates the region $Z_1 = \{z | z < \lambda_a z(x_a) + \lambda_b z(x_b) \text{ for some } \lambda_a, \lambda_b \in R \text{ and } \lambda_a + \lambda_b = 1\} \cup \{z | z < \lambda_c z(x_c) + \lambda_b z(x_b) \text{ for some } \lambda_c, \lambda_b \in R \text{ and } \lambda_c + \lambda_b = 1\}$. We eliminate the cone dominated regions $Z_2 = \{z | z \geq x_a + \mu_b(x_a - x_b), \mu_b \geq 0\}$ and $Z_3 = \{z | z \geq x_c + \mu_b(x_c - x_b), \mu_b \geq 0\}$ since the most preferred solution of the DM cannot be in these regions due to Lemma 1. The DM would not prefer any solution in the region composed of solutions x_d that are dominated by x_a or x_b or x_c , $Z_4 = \{z | z_j \geq z_j(x_i), j = 1, 2, \text{ and } z_j > z_j(x_i) \text{ for at least one } j \text{ for } i = a, b, c\}$. This leaves only regions

$$Z^* = \{z | z = \mu_a z(x_a) + \mu_b z(x_b) + \mu_{ab} z(n(x_a, x_b)), \mu_a + \mu_b + \mu_{ab} = 1, \mu_a, \mu_b, \mu_{ab} \geq 0\} \cup \{z | z = \mu_c z(x_c) + \mu_b z(x_b) + \mu_{cb} z(n(x_c, x_b)), \mu_c + \mu_b + \mu_{cb} = 1, \mu_c, \mu_b, \mu_{cb} \geq 0\} \text{ for the most preferred solution of the DM.} \quad \#$$

We demonstrate Lemma 3 in Figure 3.1. We eliminate the regions in which the most preferred solution of the DM cannot lie. Given that solution x_b is preferred to its two adjacent efficient solutions, x_a and x_c , there cannot be any solutions in Region 1 due to Lemma 2. The best solution of the DM cannot lie in Regions 2 and 3 due to Lemma 1. Region 4 is composed of dominated solutions by x_a, x_b or x_c . Therefore, the only regions where the best solution of the DM can lie are the triangular regions between the current best solution, x_b , and its two adjacent efficient solutions, x_a and x_c .

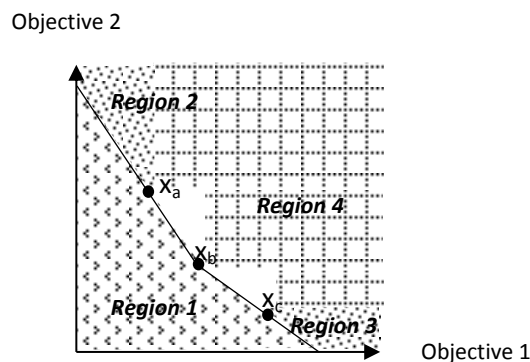


Figure 3.1 Admissible Regions for the Most Preferred Solution

Before finding the supported efficient solution that is preferred to all its adjacent efficient solutions, we do not impose any additional constraints on the objective space. We refer to this objective space as the *original objective space*.

The three additional constraints; an upper bound on the first objective ($z_1(x) \leq UB_1$), an upper bound on the second objective ($z_2(x) \leq UB_2$), and a lower bound on the weighted combination of the two objectives ($wz_1(x) + (1 - w)z_2(x) \geq LB$) define a triangle in which the most preferred solution of the DM can lie. The lower bound on the weighted combination of the two objectives is a redundant constraint, since there are no solutions that do not satisfy this constraint given the upper bounds on the objectives. Therefore, we will not consider this lower bound on the weighted combination of the two objectives in our study.

The upper bounds on the objectives also eliminate the adjacent solutions the DM found inferior. In Figure 3.1, the upper bound on the second objective value of the left triangle eliminates solution A and the upper bound on the first objective value in the right triangle eliminates solution C. Let $z_p(Q)$ denote the p^{th} objective value of solution Q . Then, the bounds on the left triangle are $z_1(x) \leq z_1(B)$ and $z_2(x) \leq z_2(A) - \varepsilon$, for a small positive constant, ε . Similarly, the bounds on the right triangle are $z_1(x) \leq z_1(C) - \varepsilon$ and $z_2(x) \leq z_2(B)$. We refer to this objective space as the *reduced objective space*.

We next summarize the steps of the interactive algorithm to find the most preferred solution of a DM with a quasiconvex preference function for a bi-objective integer program. In this algorithm, we generalize the concepts used in Tezcaner and Köksalan (2011)'s approach for the quasiconvex case. We use the ideas in Lemmas 1, 2 and 3 to narrow down the objective space and introduce the bounds explained above for the reduced objective space.

3.2 The Steps of the Interactive Algorithm

Let ε be a small positive constant, $w_{LE} = 1 - \varepsilon$, and $w_{RE} = \varepsilon$.

Step 1. Find the extreme efficient solutions of the problem by minimizing $U(x) = w z_1(x) + (1 - w) z_2(x)$ equating w to w_{LE} and w_{RE} . Let the solutions be x_{LE} and x_{RE} , respectively. If $x_{LE} = x_{RE}$, the problem has only one solution in the preferred region. $x' = x_{LE} = x_{RE}$. Go to Step 10. Otherwise, go to Step 2.

Step 2. Let $w' = \frac{z_2(x_{LE}) - z_2(x_{RE})}{(z_2(x_{LE}) - z_2(x_{RE})) + (z_1(x_{RE}) - z_1(x_{LE}))}$

Find the solution that minimizes the function, $U(x) = w z_1(x) + (1 - w) z_2(x)$ formed with weight w' . Let the solution be x' . If $x' = x_{LE}$ or $x' = x_{RE}$, set $x_L = x_{LE}$ and $x_R = x_{RE}$ and go to Step 3. Otherwise, go to Step 4.

Step 3. Ask the DM x_L versus x_R .

- If x_L is preferred to x_R , update the bounds on objectives as follows: $z_2(x) \leq z_2(x_L)$, $z_1(x) \leq z_1(x_R) - \varepsilon$. Set $x' = x_L$. Go to Step 8.
- If x_R is preferred to x_L , update the bounds on objectives as follows: $z_2(x) \leq z_2(x_L) - \varepsilon$, $z_1(x) \leq z_1(x_R)$. Set $x' = x_R$. Go to Step 6.

Step 4. *Creating the left triangle* Find the left adjacent solution (x_L) of x' . Ask the DM x' versus x_L .

- If x' is preferred to x_L , set the bounds for the left triangle as follows: $z_2(x) \leq z_2(x_L) - \varepsilon$, $z_1(x) \leq z_1(x')$. If the right triangle is not formed, go to Step 5. Otherwise, go to Step 6.
- Otherwise, set the bounds for the right triangle as follows: $z_2(x) \leq z_2(x_L)$, $z_1(x) \leq z_1(x') - \varepsilon$. Set $x' = x_L$. If $x' = x_{LE}$, go to Step 8. If $x' \neq x_{LE}$, go to Step 4.

Step 5. *Creating the right triangle* Find the right adjacent solution (x_R) of x' . Ask the DM x' versus x_R .

- If x' is preferred to x_R , set the bounds for the right triangle as $z_1(x) \leq z_1(x_R) - \varepsilon$, $z_2(x) \leq z_2(x')$. Go to Step 6.
- Otherwise, set the bounds for the left triangle as follows: $z_2(x) \leq z_2(x') - \varepsilon$, $z_1(x) \leq z_1(x_R)$. Set $x' = x_R$. If $x' = x_{RE}$, go to Step 6. If $x' \neq x_{RE}$, go to Step 5.

Step 6. Search for the left adjacent solution of x' in the left triangle.

- If no new solution is found and if right triangle is formed, go to Step 8. If no new solution is found and if right triangle is not formed, go to Step 10.
- If a new solution is found, let the solution be x_L . Go to Step 7.

Step 7. Ask the DM x' versus x_L .

- If x' is preferred to x_L , update the bound on the second objective of left triangle as $z_2(x) \leq z_2(x_L) - \varepsilon$. Go to Step 6.

- If x_L is preferred to x' , update the bound on the left triangle as $z_1(x) \leq z_1(x_L)$. Update the bounds on the right triangle as $z_2(x) \leq z_2(x_L)$, $z_1(x) \leq z_1(x') - \varepsilon$. Set $x' = x_L$. Go to Step 6.

Step 8. Search for the right adjacent solution of x' in the right triangle. If no new solution is found, go to Step 10. Otherwise, let the solution be x_R . Go to Step 9.

Step 9. Ask the DM x' versus x_R .

- If x' is preferred to x_R , update the bound on the first objective of right triangle as $z_1(x) \leq z_1(x_R) - \varepsilon$. Go to Step 8.
- If x_R is preferred to x' , update the bound on the left triangle as $z_2(x) \leq z_2(x') - \varepsilon$, $z_1(x) \leq z_1(x_R)$. Update the bound on the second objective of right triangle as $z_2(x) \leq z_2(x_R)$. Set $x' = x_R$. Go to Step 6.

Step 10. The best solution is x' .

Suppose we have a problem with the set of efficient solutions given in Figure 3.2. The best solution of the DM that minimizes the underlying preference function shown in Figure 3.2 (a) is solution C. In Steps 4 and 5, we reduce the objective space around solution C as shown in Figure 3.2 (b). In Step 6, we search in the left triangle where we cannot find a new solution. Then we pass to Step 8, where we search the right triangle. We find solution D and we ask the DM to compare these two solutions. Since C is preferred, we reduce the objective space further as in Figure 3.2 (c). Since we cannot find a new solution in these reduced objective spaces, we conclude that C is the best solution.

Linear preference functions are quasiconvex functions, too. Therefore, this algorithm can also be used for linear preference functions. If we know that the DM has a linear preference function, we can terminate the algorithm at the end of Step 5. The solution that is preferred to both its left and right adjacent efficient solutions in the original objective space would be the most preferred solution.

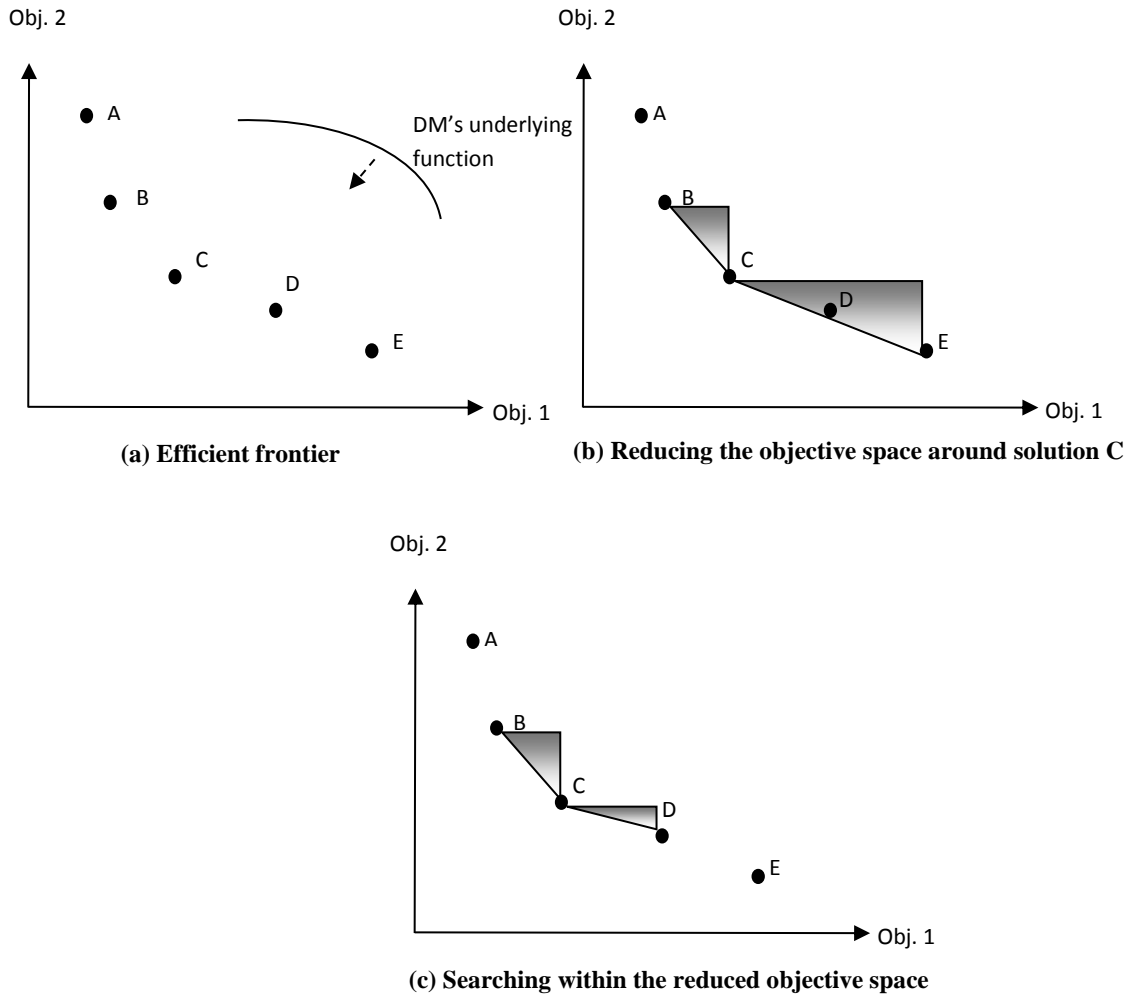


Figure 3.2 Demonstration of the Interactive Algorithm

3.3 Finding Adjacent Efficient Solutions

During the interactive algorithm, we find adjacent efficient solutions both in the original objective space (in Steps 4 and 5) and the reduced objective space (in Steps 6 and 8).

We search for adjacent efficient solutions in the original objective space using the method presented in Tezcaner and Köksalan (2011). In this method, firstly, the left extreme efficient solution, x_L , is found by minimizing $U(x) = (1 - \varepsilon)z_1(x) + \varepsilon z_2(x)$

where ε is a small positive constant used to guarantee that the resulting solution is efficient. We follow the steps A.1 and A.2., if $x_L \neq x$. Otherwise, there are no left adjacent efficient solutions of x in the search region.

We next present the steps of the algorithm.

Step A.1. Let $w' = \frac{z_2(x_L) - z_2(x)}{(z_2(x_L) - z_2(x)) + (z_1(x) - z_1(x_L))}$

Find the solution minimizing $U'(x) = w'z_1(x) + (1 - w')z_2(x)$ and denote it as x' . If $x' = x_L$ or $x' = x$, go to Step A.2. Otherwise, let $x_L = x'$ and go to Step A.1.

Step A.2. The left adjacent solution of x is x_L .

The right adjacent in the original objective space can be found similarly.

The only difference of finding adjacent efficient solutions in the reduced objective space is the addition of the constraints $z_1(x) \leq UB_1, z_2(x) \leq UB_2$ in the latter case. We develop two methods to find the adjacent efficient solutions in the reduced space.

3.3.1 Procedure 1 – Moving from the Outermost Solution to the Adjacent

In the first procedure, we also find the adjacent efficient solutions in the reduced objective space using the method presented in Tezcaner and Köksalan (2011).

Suppose we have a reduced objective space in which x is preferred to its current left adjacent solution x_L' and suppose we remove x_L' from the search space with the constraints $z_i(x) \leq UB_i, i = 1, 2$. We first find the left extreme efficient solution minimizing $U(x) = w_L z_1(x) + (1 - w_L) z_2(x)$ for $w_L = 1 - \varepsilon$, where ε is a very small positive constant inside the region $z_i(x) \leq UB_i, i = 1, 2$. Let the solution be x_L . We follow steps A.1 and A.2 given above, if $x_L \neq x$. Otherwise, there are no left adjacent efficient solutions of x in the search region.

While searching for the left adjacent solution, we first find the leftmost solution in the reduced objective space. Each time we enter Step A.1, we update w_L such that we finally obtain the w_L such that the corresponding $U(x)$ passes through solution x and its left adjacent x_L . The algorithm is demonstrated on an example problem shown in Figure 3.3. Here, we search for the left adjacent efficient solution of x in the reduced objective space.

The dashed lines in Figure 3.3(a) show the region where we search for the left adjacent of x . We first find the leftmost solution, x_L , in the search region as shown in Figure 3.3 (b). Then, we minimize the function that passes through x_L and x as shown in Figure 3.3(c). Since we do not find a new solution, we conclude that x_L is the left adjacent of x .

The right adjacent can be found similarly.

3.3.2 Procedure 2 – Moving from a Central Solution to the Adjacent

In the second procedure, we reduce the objective space with some constraints and gradually update w_L to come up with the adjacent solution. The steps to find the left adjacent efficient solution of an efficient solution x are given below.

Let x be the solution that is preferred to its left adjacent solution, x_L' . Let $w_L = \frac{z_2(x_L') - z_2(x)}{(z_2(x_L') - z_2(x)) + (z_1(x) - z_1(x_L'))}$, the weight of the function that passes through x and x_L' .

Suppose we search its left adjacent in the reduced objective space: $z_i(x) \leq UB_i, i = 1, 2$. In this method, both x and x_L' are excluded from the search region.

We next present the steps of the algorithm.

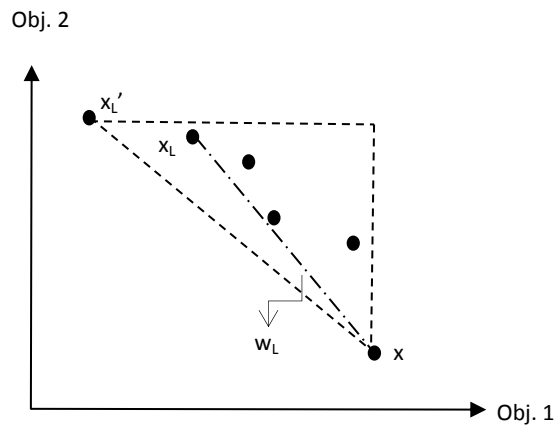
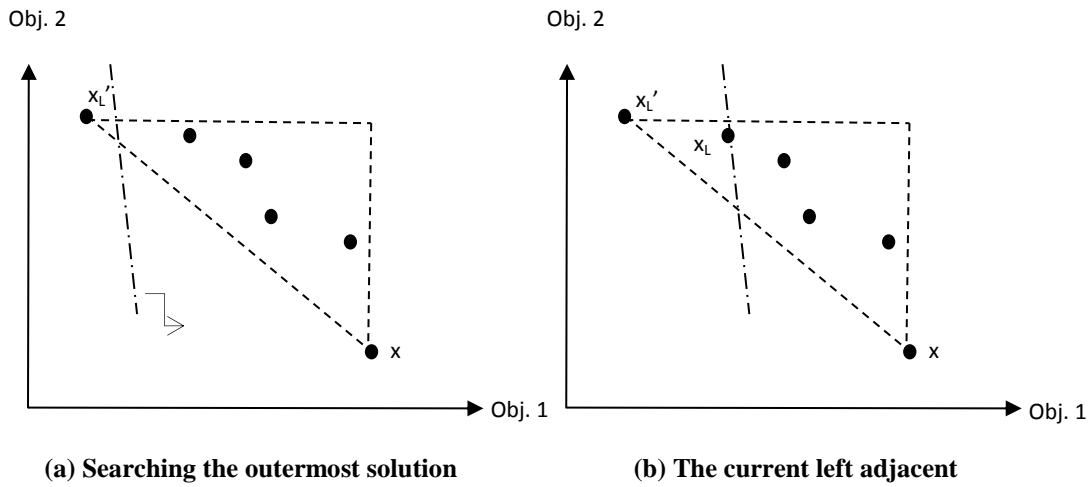


Figure 3.3 Finding Adjacent Efficient Solutions – Procedure 1

Step B.1. Find the solution that minimizes the function $U(x) = w_L z_1(x) + (1 - w_L) z_2(x)$ within the search region. If no solution is found, go to Step B.3. If a new solution is found, let that solution (x_L) be the current left adjacent of x . Find the point that the function $w_L z_1(x_L) + (1 - w_L) z_2(x_L)$ intersects with $z_2(x_L) - \varepsilon$. Let the objective values of that point be $z_1(x_D)$ and $z_2(x_D)$.

Step B.2. For $w' = \frac{z_2(x_L) - z_2(x)}{(z_2(x_L) - z_2(x)) + (z_1(x) - z_1(x_L))}$, add the constraints $w'z_1(x) + (1 - w')z_2(x) \leq w'z_1(x_L) + (1 - w')z_2(x_L)$ and $w_Lz_1(x) + (1 - w_L)z_2(x) \geq w_Lz_1(x_L) + (1 - w_L)z_2(x_L)$ in addition to the constraints defining the triangle. Let $w_L = \frac{z_2(x_D) - z_2(x)}{(z_2(x_D) - z_2(x)) + (z_1(x) - z_1(x_D))}$. Go to Step B.1.

Step B.3. If x_L is not updated during the algorithm, the triangle within which we are searching is empty; there are no solutions that can be the left adjacent of x' . Therefore, the search in the left triangle is finished. If x_L is updated at least once during the algorithm, x_L is the left adjacent of x' .

We demonstrate the algorithm on the same problem of Procedure 1 in Figure 3.4. We start with minimizing the function passing through x and x_L' (as shown in Figure 3.4(a)) and find solution x_L as shown in Figure 3.4(b). We update the bounds on the triangle and the w_L as shown in Figure 3.4(c). The shaded region shows where left adjacent of x' can be. We again search for the solution minimizing the function with w_L . We find a new solution, x_L as shown in Figure 3.4(d). After updating the bounds on the triangle and w_L as shown in Figure 3.4(e), we cannot find a new solution. We conclude that x_L is the left adjacent solution of x' in the reduced objective space.

The right adjacent of a solution is found similarly.

We made a number of tests on these procedures and concluded that they reach the adjacent efficient solutions by finding almost the same number of efficient solutions. The first procedure requires less computation, therefore we select procedure 1 in finding adjacent efficient solutions in the reduced objective space.

CHAPTER 4

APPLICATION OF THE INTERACTIVE ALGORITHM TO THE BI-OBJECTIVE ROUTING PROBLEM

In the routing problem, we find the tour that passes through all nodes and returns to the starting point. In the single objective routing problem, we first find the optimal arc between each node pair. This problem is the shortest path problem. Then we form the optimal tour that uses a subset of the optimal arcs between node pairs. This is a traveling salesperson problem. Overall, routing problem is a combination of these two combinatorial optimization problems. Under multiple objectives, the problem becomes harder. There may be multiple efficient arcs between nodes and different subsets of those efficient arcs form the efficient tours. Finding the tours composed of the efficient arcs can be considered as the generalized MOTSP.

If the DM has a linear preference function, we know that the most preferred tour is composed of the most preferred arcs between node pairs (Tezcaner and Köksalan, 2011). Therefore, we first find the most preferred arcs between node pairs and reduce the problem to a MOTSP where each node pair is connected with a single arc. Then we solve the MOTSP and find the most preferred tour. However, we cannot treat the two problems separately for the quasiconvex preference function case. The best arcs between node pairs with respect to a quasiconvex preference function are not necessarily used in the best tour of that quasiconvex function. Therefore, finding the best arcs is not sufficient in search for the best tour of the DM. We show this property in the following example.

Example 4.1 Suppose the DM has the quasiconvex function $U(z) = z_1^2 + z_2^2$ to be minimized while visiting the four nodes: A, B, C and D as shown in Figure 4.1. Between all node pairs except (A,C), there are two efficient arcs. The objective function values of these arcs, (z_1, z_2) are written in parentheses next to the arcs.

Between nodes A and B, the arc with objective values (2,8) has a lower quasiconvex function value ($U(z_{(2,8)}) = 68$) when compared to the arc with objective values (5,7) ($U(z_{(5,7)}) = 74$). Therefore, arc (2,8) is the most preferred arc between nodes A and B. When similar comparisons are made, arc (3,5), arc (5,7), arc (3,5) and arc (7,1) become the most preferred arcs between nodes A and D, B and C, B and D, and C and D, respectively. The most preferred solution of the tour A-C-B-D-A is composed of the arcs with objective values (4,3) for A-C, (5,7) for C-B, (3,5) for B-D and (6,2) for D-A. One can see that, the most preferred arc between A and D is not used in the most preferred solution of tour A-C-B-D-A.

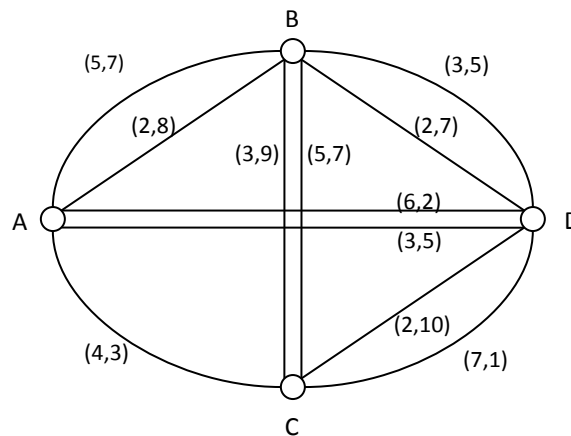


Figure 4.1 An Example MOTSP

As shown in the example, finding the most preferred arc between each node pair is not sufficient in finding the most preferred tour of the DM. Therefore, the problem cannot

be decomposed into two subproblems that complement each other as in the linear function case.

Next, we give the formulation for the generalized MOTSP. We assume that we have all the efficient arcs between node pairs and we input this information into the formulation.

Let N denote the node set, E denote the arc set, x_{ij}^k be the binary decision variable to indicate whether the k^{th} efficient arc of edge (i, j) is used or not. Let K_{ij} be the number of efficient arcs between nodes i and j . Assuming that the TSP is symmetric, the overall formulation for the problem is as follows:

$$\text{Min } z_1 = \sum_{x \in X} z_1(x) \quad (4.1)$$

$$\text{Min } z_2 = \sum_{x \in X} z_2(x) \quad (4.2)$$

$$\mathbf{x} \in X$$

where X is made up of the following constraints:

$$\sum_{j \in N} \sum_{k=1}^{K_{ij}} x_{ij}^k = 1 \quad \forall i \in N \quad (4.3)$$

$$\sum_{i \in N} \sum_{k=1}^{K_{ij}} x_{ij}^k = 1 \quad \forall j \in N \quad (4.4)$$

$$u_i - u_j + \sum_{k=1}^{K_{ij}} (|N| - 1) x_{ij}^k + \sum_{k=1}^{K_{ji}} (|N| - 3) x_{ji}^k \leq |N| - 2 \quad \forall i, j \in N - \{1\}, i \neq j \quad (4.5)$$

$$1 \leq u_i \leq |N| - 1 \quad \forall i \in N - \{1\} \quad (4.6)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in N, k = 1, \dots, K_{ij} \quad (4.7)$$

We minimize first and second objectives in (4.1) and (4.2), respectively. Equations (4.3) and (4.4) ensure departure from and arrival to each node, respectively. There are different possible formulations for subtour elimination constraints. Laporte (1992) reviews the algorithms developed for TSP. He covers some subtour elimination

constraints including the formulation introduced by Miller, Tucker and Zemlin (1960), which was strengthened later by Desrochers and Laporte (1991). Constraints (4.5) and (4.6) are the strengthened version of Miller et al.'s formulation. This formulation requires a smaller number of additional constraints compared to other subtour elimination constraints. However it requires including $|N| - 1$ positive decision variables in the formulation.

4.1 Finding Efficient Arcs

MOSPP is NP-complete and intractable (Ehrgott, 2000). In the literature, many solution approaches are developed for MOSPP. Label correcting algorithms (see Skriver and Andersen, 2000), label setting algorithms (see Gandibleux et al., 2006), ranking path based algorithms (see Raith and Ehrgott, 2009) are examples of methods developed for the solution of this problem.

We use a label setting algorithm, Martin's algorithm to find all efficient arcs between node pairs. This algorithm is explained in Appendix A. Briefly, we produce labels for each point in the network that contain information about the objective values from the starting point to that point. In each iteration a label is set permanent; which shows that an efficient arc from the initial point to that point is found. During the algorithm, we check for dominance between each newly produced label and the existing ones. If the arcs corresponding to the previous labels of a point have smaller objective values than the point's newly created arc, this new arc is dominated. Another possibility is that the arc corresponding to a new label dominates arcs of a number of previously created labels. If neither of these cases occur, the label is put in the label list of that point. The algorithm continues until all labels are set permanent.

In the routing problem, we divide the terrain into equal-sized grids in the two dimensional space. The problem becomes computationally harder as the number of grids increase. Also, the dominance check and the need to evaluate all labels further

complicate the solution process. We include one more step in the dominance check for all the arcs generated. As soon as an arc to the destination node is found, we compare its objective values with those of all the other arcs to all nodes. If any arc to any node has higher objective values than the arc(s) to the destination node, we remove those dominated arcs.

4.2 Finding Efficient Solutions in the Original and Reduced Objective Spaces

We consider finding efficient solutions in the original and reduced objective spaces separately.

Finding Efficient Solutions in the Original Objective Space

We first find all the efficient arcs between all node pairs before running the algorithm using Martin's algorithm. In Steps (1) through (5), we find tours for the combined objective $U(x) = wz_1(x) + (1 - w)z_2(x)$ without any additional constraints in the objective space. The only difference of this formulation from the single objective TSP is the existence of multiple efficient arcs between node pairs. Since we combine the two objectives linearly, we can reduce the efficient arcs between each node pair (i, j) to one. Let $z_p(i, j, k)$ denote the p^{th} objective value of edge node pair (i, j) , $(i, j) \in E, p = 1, 2$. We assign the combined objective value (e_{ij}) of node pair (i, j) to the minimum combined objective value over its arcs as follows:

$$e_{ij} = \begin{cases} 0 & i = j \\ \min_{k=1, \dots, K_{ij}} \{wz_1(i, j, k) + (1 - w)z_2(i, j, k)\} & i \neq j \end{cases}$$

We then solve the TSP using the following objective function value, where y_{ij} is a binary variable to indicate whether the node pair (i, j) is used or not.

$$\text{Min } z = \sum_{(i,j) \in E} e_{ij} y_{ij} \tag{4.8}$$

We solve the resulting TSP using the software Concorde (see <http://www.math.uwaterloo.ca/tsp/concorde.html>), for which we only need to input the combined objective value for each node pair (i, j) .

Finding Efficient Solutions in the Reduced Objective Space

In Steps (6) – (9) of the interactive algorithm, we have the following two constraints in addition to (4.1)-(4.8). The introduction of these constraints destroys the input structure required by Concorde, and we cannot use it in this case.

$$\sum_{(i,j) \in E} \sum_{k=1}^{K_{ij}} z_1(i, j, k) x_{ij}^k \leq UB_1 \quad (4.9)$$

$$\sum_{(i,j) \in E} \sum_{k=1}^{K_{ij}} z_2(i, j, k) x_{ij}^k \leq UB_2 \quad (4.10)$$

Considering formulation (4.1)-(4.8), (4.9) and (4.10), as the number of efficient arcs increase, the problem becomes harder to solve. In the next section we explain how we reduce the number of arcs considering constraints (4.9) and (4.10).

4.3 Arc Reduction

As mentioned above, we find all the efficient arcs between all node pairs before applying our interactive algorithm. During the algorithm, we evaluate all these arcs and determine which ones to use. However, in Steps (6)-(9), we consider a small region in the objective space that is specified with constraints (4.9) and (4.10). Figure 4.2 illustrates the two constraints on the objective space.

The next theorem shows how we can use the bounds on the objectives for the bi-objective TSP to find bounds on the objectives for the bi-objective SPP. This will be useful to find the efficient arcs that would not be used in any feasible tour.

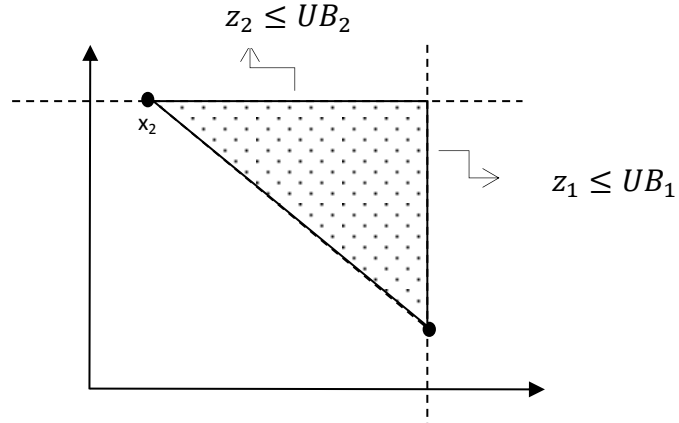


Figure 4.2 Bounds on the Objectives in the Reduced Objective Space

Theorem 1. Let UB_p be an upper bound on the p^{th} objective value. Let each node pair, except (a, b) , be assigned to its smallest p^{th} objective value as its arc cost.

$$e_{ij} = \begin{cases} 0 & i = j \text{ or } (i, j) = (a, b) \\ \min_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\} & i \neq j \end{cases}$$

Let T_{ab}^p be a solution that can be used as a lower bound for the objective value of a tour that minimizes the sum of arc costs, e_{ij} , and includes a connection between (a, b) . Let the objective function value corresponding to solution T_{ab}^p be B_{ab}^p ($= \sum_{(i,j) \in \{T_{ab}^p - (a,b)\}} \min_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\}$). Then, the efficient arcs connecting nodes (a, b) having p^{th} objective value larger than $UB_p - B_{ab}^p$ cannot be used in any feasible tour and $UB_{ab}^p = UB_p - B_{ab}^p$ is an upper bound on the p^{th} objective value of arc (a, b) .

Proof. The constraint on the upper bound on the p^{th} objective value can be written as follows:

$$\sum_{(i,j) \in E - \{(a,b)\}} \sum_{k=1}^{K_{ij}} z_p(i, j, k) x_{ij}^k + \sum_{k=1}^{K_{ab}} z_p(a, b, k) x_{ab}^k \leq UB_p \quad (4.11)$$

Constraints (4.3)-(4.7) ensure that $|N|$ binary variables are equal to 1 to form a tour and there should be an outgoing and an incoming arc to each node. Suppose we introduce constraint (4.11).

Let T_{ab}^p be a solution with objective value B_{ab}^p that can be used as a lower bound for the objective value of a tour that minimizes the sum of arc costs, e_{ij} , and includes a connection between (a, b) . Since B_{ab}^p is a lower bound for the p^{th} objective value of a tour containing (a, b) , $\sum_{(i,j) \in E - \{(a,b)\}} \sum_{k=1}^{K_{ij}} z_p(i, j, k) x_{ij}^k \geq B_{ab}^p$. This restricts the p^{th} objective value of the arcs of (a, b) as follows:

$$\sum_{k=1}^{K_{ab}} z_p(a, b, k) x_{ab}^k \leq UB_p - B_{ab}^p \quad (4.12)$$

Let $UB_{ab}^p = UB_p - B_{ab}^p$. The nodes (a, b) can only be connected with a single efficient arc. Therefore, (4.12) reduces to the following inequality:

$$z_p(a, b, k) x_{ab}^k \leq UB_{ab}^p \quad k = 1, \dots, K_{ab}$$

The efficient arcs between (a, b) with p^{th} objective value higher than UB_{ab}^p cannot satisfy (4.11). #

According to this theorem, we can reduce the efficient arcs between node pairs when we have the bounds UB_p for $p = 1, 2$. For this, we need to find a lower bound for the objective value of the tour that minimizes the sum of arc costs considering e_{ij} explained above. The lower bound for the tour can be found by a number of approximations. A TSP tour can be formed that minimizes the p^{th} objective when each arc is assigned the above e_{ij} values. Cook et al. (1998) describe a number of lower bounding methods for the TSP. They explain *1-tree bound* as another approximation method. *1-tree bound* is based on minimum spanning tree (MST). We first construct a MST considering all but one of its nodes. We then add the remaining node to the tree by the smallest two edges incident to that node. This lower bound depends on the node selected to be connected with two edges.

In any approximation we use, we need to ensure that the edge (a, b) is included in the solution. Otherwise, we cannot reduce the arcs that have their p^{th} objective value higher than UB_{ab}^p .

We follow the steps outlined below for arc reduction when we have the bounds UB_1 and UB_2 on the first and second objectives, respectively. We find the lower bounds by forming tours. We initially construct the set Y that includes all node pairs.

We next present the steps of the algorithm.

Step P.1. Set $p = 1$. Select a node pair (a, b) from set Y . Set $Y \leftarrow Y - \{(a, b)\}$. Go to Step P.2.

Step P.2. Set the arc costs, e_{ij} , as follows:

$$e_{ij} = \begin{cases} 0 & i = j \text{ or } (i, j) = (a, b) \\ \min_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\} & i \neq j \end{cases} \quad \text{Go to Step P.3.}$$

Step P.3. Construct a tour that minimizes the sum of arc costs, e_{ij} , using Concorde. Let the resulting solution be T_{ab}^p and the objective function of the tour be $B_{ab}^p = \sum_{(i,j) \in \{T_{ab}^p - (a,b)\}} \min_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\}$. If $(a, b) \in T_{ab}^p$, set the upper bound on the p^{th} objective value of arcs of (a, b) as $UB_{ab}^p = UB_p - B_{ab}^p$. Go to Step P.4. Otherwise, increment e_{ij} enough to make sure edge (a, b) is included in T_{ab}^p as follows:

$$e_{ij} = \begin{cases} 0 & i = j \text{ or } (i, j) = (a, b) \\ e_{ij} + M & i \neq j \end{cases} \quad \text{for a large } M. \text{ Go to Step P.3.}$$

Step P.4. Eliminate the efficient arcs between nodes (a, b) with p^{th} objective value larger than UB_{ab}^p . Set $p \leftarrow p + 1$. If $p \leq 2$, go to Step P.2. If $p > 2$ and $Y = \emptyset$, terminate the algorithm. If $p > 2$ and $Y \neq \emptyset$, set $p = 1$ and select a new node pair (a, b) from Y . Set $Y \leftarrow Y - \{(a, b)\}$. Go to Step P.2.

A similar algorithm would be followed if the 1-tree bound is used as a lower bound. In Step P.3., we form a 1-tree. We select either node a or b as the node to be connected with two edges. For the remaining nodes, we form a MST by connecting each node to the tree with the minimum arc cost. Let the node to be connected with two edges be node a . Then the MST solution for nodes $i \in N - \{a\}$ be T_{ab}^p . Then, B_{ab}^p should equal $\min_{j \in N - \{a,b\}, k=1, \dots, K_{aj}} \{z_p(a, j, k)\} + \sum_{(i,j) \in T_{ab}^p} \min_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\}$. Since arc (a, b) has zero arc cost, we do not include it in the calculations.

During arc reduction, an arc that is used in finding the lower bound for the tour to evaluate an arc's arcs can be eliminated. For instance, suppose m^{th} arc of arc (i, j) is reduced and suppose $(i, j) \in T_{ab}^p$ and $m = \text{arcm}_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\}$. If this occurs, we need to find a new lower bound, B_{ab}^p for all (a, b) and p that satisfies $(i, j) \in T_{ab}^p$ and $m = \text{arcm}_{k=1, \dots, K_{ij}} \{z_p(i, j, k)\}$. This way, we find tighter bounds on the arcs and eliminate more of the efficient arcs of (a, b) .

We next explain the additional constraints we impose on the objective space when we search for unsupported efficient solutions. We also use these constraints in arc reduction.

4.4 Additional Constraints in Search for Unsupported Efficient Solutions

Tuytens et al. (2000) study the so called *two phase algorithm* on the bi-objective assignment problem. In this algorithm, the efficient frontier is generated in two phases. In the first phase, the supported efficient solutions are generated. This phase is computationally relatively easy because each supported solution can be obtained by minimizing a suitable linear combination of the two objectives. In the second phase, unsupported solutions are found. To find the unsupported solutions, the region between each adjacent supported solution pair is searched. The two phases are demonstrated in Figure 4.3.

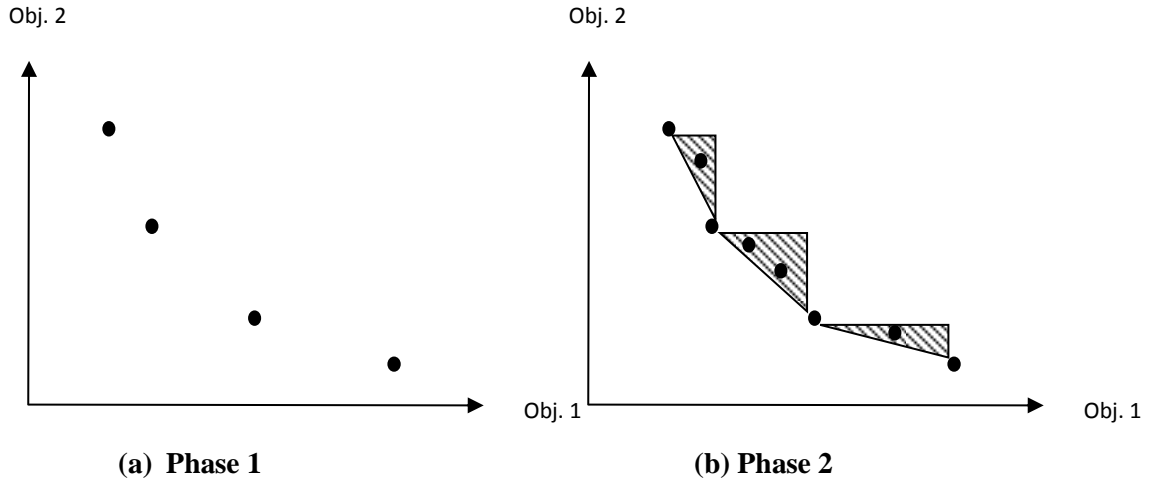


Figure 4.3 The Two Phase Algorithm

Tuytens et al. (2000) propose an additional constraint that can be used to narrow down the search space in the second phase. The constraint reduces the dominated part of the triangles between supported efficient solutions. This is illustrated in Figure 4.4. Suppose two new unsupported efficient solutions, x_3 and x_4 , are found between the two supported efficient solutions x_1 and x_2 . From the definition of efficiency, we know that the shaded region is dominated by either one or both of the unsupported efficient solutions. Tuytens et al. define a line parallel to the line connecting x_1 and x_2 such that all the points in the triangle that are dominated are also dominated by the solutions on that line. In the example in Figure 4.4, there are three possibilities for this line to pass through, the points with the worst objective values of the two consecutive supported efficient solution. Przybylski et al. (2008) refer to these points as local nadir points. In Figure 4.4 (b), the local nadir points are $(z_1(x_3), z_2(x_1))$, $(z_1(x_4), z_2(x_3))$ and $(z_1(x_2), z_2(x_4))$. The lines that pass through these local nadir points are shown in Figure 4.4 (b).

Among these three possibilities, the line through the second point reduces only dominated regions whereas the other two lines eliminate both dominated and nondominated regions. Therefore, the combined objective value passing through the second local nadir point is set as an upper bound for the combined objective of the

problem $(wz_1(x) + (1 - w)z_2(x) \leq (wz_1(x_4) + (1 - w)z_2(x_3) = UB_C)$ as shown in Figure 4.4 (c). Przybylski et al. (2008) modify this bound exploiting the fact that all variables are integer valued.

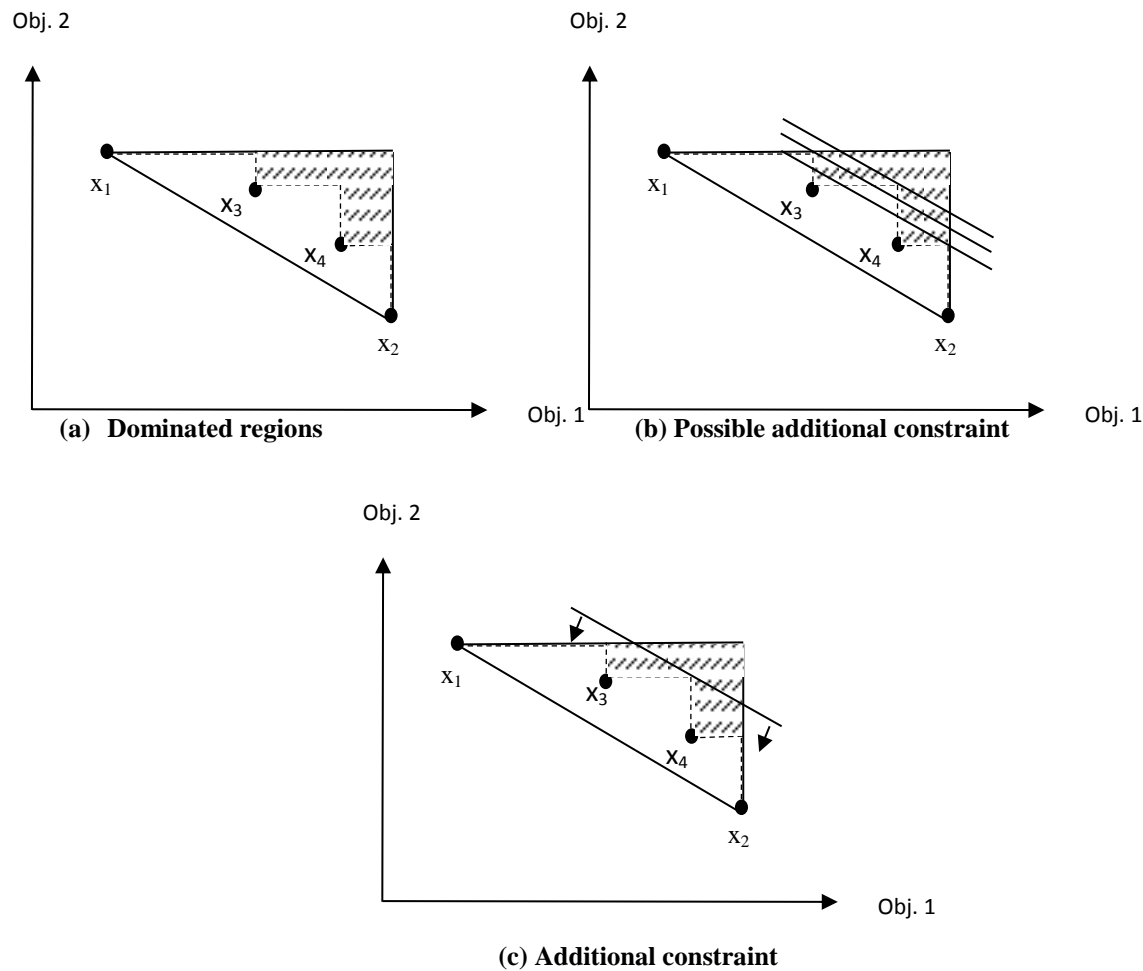


Figure 4.4 Additional Constraint for Second Phase

The constraint $wz_1(x) + (1 - w)z_2(x) \leq UB_C$ reduces a part of the dominated set, but still an unaccounted dominated region stays in the search region. We improved this bound in order to reduce more of the dominated region. We developed the following steps to determine a number of upper bounds in the triangles.

Let the upper bound on the first and second objectives be UB_1 and UB_2 , respectively. We search within the two efficient solutions, x_1 and x_2 , where $z_1(x_1) < z_1(x_2)$ and $z_2(x_1) > z_2(x_2)$, without loss of generality. Let $w = \frac{z_2(x_1) - z_2(x_2)}{(z_2(x_1) - z_2(x_2)) + (z_1(x_2) - z_1(x_1))}$ be the weight given to the first objective of the line that passes through x_1 and x_2 . Let C be the set of all consecutive efficient solutions present in the search space, $C = \{(x_1, x_{US1}), (x_{US1}, x_{US2}), \dots, (x_{USF}, x_2)\}$ where x_{USF} stands for the f^{th} unsupported efficient solution.

We next present the steps of the algorithm.

Step U.1. Find all $|C|$ local nadir points and place them in the set L where the objective values of the l^{th} local nadir point are $z_1(x_l)$ and $z_2(x_l)$.

Step U.2. Find the local nadir point that gives the highest combined objective value. $x_0^* = \arg \max_{x_l \in L} \{wz_1(x_l) + (1 - w)z_2(x_l)\}$. Set $x_0 = x_0^*$.

Step U.3. Find the left adjacent solution (x_{L0}) of x_0 among the local nadir points, treating the problem as a maximization problem. If there are no left adjacent solutions in the search space, set $x_0 = x_0^*$ and go to Step U.4. Otherwise, include the constraint $w'z_1(x) + (1 - w')z_2(x) \leq w'z_1(x_0) + (1 - w')z_2(x_0)$ for $w' = \frac{z_2(x_{L0}) - z_2(x_0)}{(z_2(x_{L0}) - z_2(x_0)) + (z_1(x_0) - z_1(x_{L0}))}$. If $z_2(x_{L0}) = UB_2$, set $x_0 = x_0^*$ and go to Step U.4. Otherwise, set $x_0 = x_{L0}$ and go to Step U.3.

Step U.4. Find the right adjacent solution (x_{R0}) of x_0 among the local nadir points, treating the problem as a maximization problem. If there are no right adjacent solutions in the search space, terminate the algorithm. Otherwise, include the constraint $w'z_1(x) + (1 - w')z_2(x) \leq w'z_1(x_0) + (1 - w')z_2(x_0)$ for $w' = \frac{z_2(x_0) - z_2(x_{R0})}{(z_2(x_0) - z_2(x_{R0})) + (z_1(x_{R0}) - z_1(x_0))}$. If $z_1(x_{R0}) = UB_1$, terminate the algorithm. Otherwise, set $x_0 = x_{R0}$ and go to Step U.4.

In Steps U.3 and U.4, we find left and right adjacent solutions of a solution for a maximization problem where all solutions are available. To find the left adjacent of a solution x_0 , we check the line that passes through x_0 and any solution at the left side of x_0 . The solution with the highest slope is the left adjacent of x_0 . Similarly, the solution which has the smallest slope for the line that connects it with x_0 is the right adjacent of x_0 .

$$x_{L0} = \operatorname{argmax}_{x_i \in Q} \left\{ \frac{z_2(x_i) - z_2(x_0)}{(z_2(x_i) - z_2(x_0)) + (z_1(x_0) - z_1(x_i))} \right\} \quad \text{where} \quad Q = \{x_i | x_i \in L, z_1(x_i) < z_1(x_0)\}.$$

$$x_{R0} = \operatorname{argmin}_{x_i \in Q} \left\{ \frac{z_2(x_0) - z_2(x_i)}{(z_2(x_0) - z_2(x_i)) + (z_1(x_i) - z_1(x_0))} \right\} \quad \text{where} \quad Q = \{x_i | x_i \in L, z_1(x_i) > z_1(x_0)\}.$$

At the end of these steps, we would end up with a number of upper bounds in the search region, each of which is a linear combination of a different w' value. The constraints that we put for the example problem in Figure 4.4 can be seen in Figure 4.5. These constraints eliminate more of the dominated region than the constraint proposed by Tuytens et al. (2000). We solved several problems and observed that the improvement did indeed eliminate additional arcs.

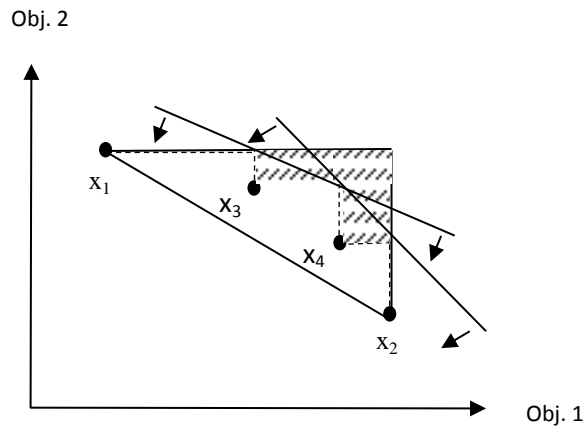


Figure 4.5 Additional Constraints for the Second Phase

We utilize these constraints in arc reduction. During the evaluation of arcs between nodes (a, b) , for the m^{th} constraint $w'_m z_1(x) + (1 - w'_m) z_2(x) \leq UB_{C_m}$, we set the arc costs of each node pair as follows:

$$e_{ij}^m = \begin{cases} 0 & i = j \text{ or } (i, j) = (a, b) \\ \min_{k=1, \dots, K_{ij}} \{w'_m z_1(i, j, k) + (1 - w'_m) z_2(i, j, k)\} & i \neq j \end{cases}$$

We find a lower bound for the tour minimizing the m^{th} weighted objective value using e_{ij}^m and set that lower bound to B_{C_m} . The efficient arcs between nodes (a, b) with larger weighted objective values (for $w = w'_m$) than $UB_{ab}^m = UB_{C_m} - B_{C_m}$ are reduced.

When a new unsupported efficient solution is found inside the search region, we update the local nadir points. Consequently, we also update the upper bounds and eliminate more of the search space. Since we continue the search inside the same region, we reduce arcs from the reduced efficient arc set.

From the computational aspect, the additional constraints did not provide a considerable reduction in the CPU times with the current lower bounding methods in our experiments. However, we believe that these methods can be further improved to obtain tighter bounds and sizeable reductions can be obtained in the CPU times.

4.5 Improvements in the Interactive Algorithm and the Arc Reduction Approach

In arc reduction, for each arc, we find lower bounds for tours considering the two upper bounds on the objectives and a number of upper bounds on the combined objectives. If there are M upper bounds on the combined objectives, we need to find $(M + 2) \frac{(N)(N-1)}{2}$ lower bounds. Also, if an arc used in finding the lower bound of the tour for an arc is reduced in the succeeding steps, new lower bounds that do not include these

arcs should be found. In this section, we explain how we decrease the number of arc reduction stages and the number of lower bound computations.

Permanent Arc Reduction

In Steps 5 and 6 of the interactive algorithm, we reduce the search space around the most preferred solution (x') and its left (x_L) and right (x_R) adjacent solutions, if they exist. We are sure that the most preferred solution (x^*)'s objective values should satisfy the following constraints demonstrated in Figure 4.6.

$$z_1(x_L) + \varepsilon \leq z_1(x^*) \leq z_1(x_R) - \varepsilon$$

$$z_2(x_R) + \varepsilon \leq z_2(x^*) \leq z_2(x_L) - \varepsilon$$

Considering the two upper bounds $z_1(x_R)$ and $z_2(x_L)$, we can reduce the arcs that would not be present in any of the solutions in the search space. These reduced arcs would not be considered again in the search within the left and right triangles and therefore can be eliminated permanently from the efficient arc sets.

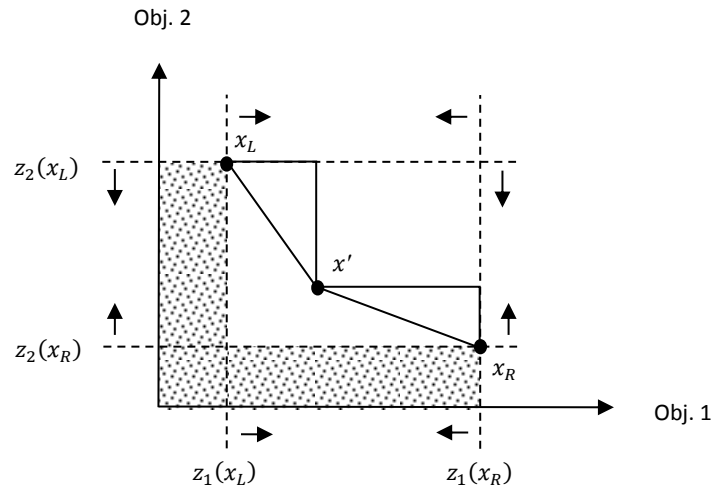


Figure 4.6 The Bounds on Objectives after Steps 5 and 6

The lower bounds on the two objectives $z_1(x_L)$ and $z_2(x_R)$ do not lead to any additional arc reduction. In the presence of the upper bounds, the lower bounds only eliminate the regions that dominate x_L and x_R . Since x_L and x_R are efficient, there should not be any solutions in the regions prohibited by the two lower bounds. These regions are shaded in Figure 4.6. Therefore, we do not consider the two lower bounds in arc reduction.

Arc Reduction when Search Space is Reduced within the Previous Search Space

For each search space with different bounds on objectives, we eliminate different efficient arcs. Therefore, for each search space, we need to start arc reduction considering all efficient arcs. We may use the previous bounds (UB_{ab}^p) and continue with the previous efficient arc set only if we reduce the search space within the previous search space. An example is shown in Figure 4.7. Suppose the DM prefers x_1 to x_2 in Figure 4.7 (a). The two bounds on the objectives, $UB_{1,first}(= z_1(x_1))$ and $UB_{2,first}(= z_2(x_2) - \varepsilon)$, eliminate x_2 from the search region. Considering these constraints, we reduce the efficient arcs between node pairs. In the succeeding iterations, we search for the left adjacent of x_1 in the reduced objective space. Suppose we find x_3 and the DM prefers x_1 to x_3 . We now have the bounds $UB_{1,second}(= z_1(x_1))$ and $UB_{2,second}(= z_2(x_3) - \varepsilon)$, on the objectives as shown in Figure 4.7 (b). Here, the bound on the first objective has not changed ($UB_{1,first} = UB_{1,second}$) and the bound on the second objective has decreased ($UB_{2,first} > UB_{2,second}$). We can continue arc reduction from the reduced arc set of the bounds $UB_{1,first}$ and $UB_{2,first}$ by only changing the upper bound on the second objective value, UB_{ab}^2 , of each arc (a, b) as $UB_{ab}^2(new) = UB_{ab}^2 - UB_{2,first} + UB_{2,second}$. This way, we avoid $\frac{(N)(N-1)}{2}$ lower bound computations for the upper bound of $p = 2$ and reduce new arcs without finding new lower bounds.

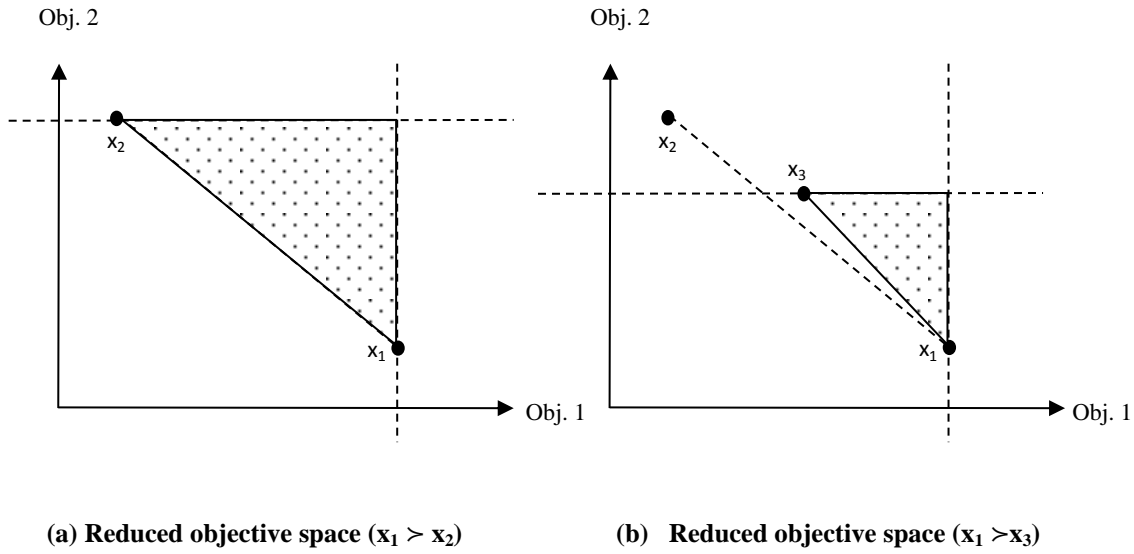


Figure 4.7 Bounds on the Objectives in the Reduced Objective Space

Finding Adjacent Efficient Solutions in the Partly Searched Region

We always store the efficient solutions we find during the algorithm in set S . Whenever we need to find a new adjacent efficient solution inside a triangle which is partly already searched, we use the previously found solutions in determining the weight, w' , in Step A.1. If we cannot find a solution in the left region, we set w' to $1 - \varepsilon$ where ε is a small positive constant. We set w' for finding the left adjacent of x inside the region defined by bounds UB_1 and UB_2 as follows:

$$w' = \min \left\{ 1 - \varepsilon, \min_{x_i \in A} \left\{ \frac{z_2(x_i) - z_2(x)}{(z_2(x_i) - z_2(x)) + (z_1(x) - z_1(x_i))} \right\} \right\}$$

where $A = \{x_i | x_i \in S, x_i \neq x, z_1(x_i) \leq UB_1, z_2(x_i) \leq UB_2\}$ and ε is a small positive constant.

Similarly, we set the weight, w' , for finding the right adjacent of x inside the region defined by bounds UB_1 and UB_2 as follows:

$$w' = \max \left\{ \varepsilon, \max_{x_i \in A} \left\{ \frac{z_2(x) - z_2(x_i)}{(z_2(x) - z_2(x_i)) + (z_1(x_i) - z_1(x))} \right\} \right\}$$

where $A = \{x_i | x_i \in S, x_i \neq x, z_1(x_i) \leq UB_1, z_2(x_i) \leq UB_2\}$ and ε is a small positive constant.

4.6 Alternatives for Solving the Constrained TSP and Finding Lower Bounds in Arc Reduction

In Steps 6 and 8 of the interactive algorithm, we search for solutions in the reduced objective space. This turns into solving the single objective constrained TSP for the routing problem. We use two methods. Our first approach is a branch and bound algorithm that starts with an assignment model and continues with breaking subtours until a tour is formed. The general structure of the branch and bound algorithm is given below. To differentiate between the nodes in the branch and bound tree and the nodes in the TSP, we refer to the nodes in the TSP as targets.

- *Initialization*: For a weight w , minimize the objective $U(x) = wz_1(x) + (1 - w)z_2(x)$ subject to (4.3), (4.4), (4.7), (4.9) and (4.10). This model may result in a number of subtours.
- *Branching*: Branch from a node if that solution results in subtours. To break the subtours, force a single edge to take the value 0 in each branch. Check all the subtours that are formed and break the subtour containing the minimum number of targets. This creates the least number of new nodes for the next level in the branch and bound tree.
 - o *Branching from the initial node*: The problem is a symmetrical TSP with $z_p(i, j, k) = z_p(j, i, k)$ for all $i, j \in N, k = 1, \dots, K_{ij}, p = 1, 2$. From the initial node, create a single branch, breaking only a single edge in the subtour having the smallest number of targets. This prevents forming a branch and bound tree with all solutions replicated twice.
 - o *Branching from the remaining nodes*: Each node may contain subtours. If the node does not contain any subtours, it becomes a candidate for the optimal

solution. If there exist subtours, one of the subtours (the subtour containing the smallest number of targets) is broken. A new node is created by setting one edge to zero each time.

- *Node selection*: While progressing in the branch and bound tree, continue with the first generated node.
- *Pruning a node*: Prune a node if it is infeasible or if it has a worse objective value than the upper bound found so far. If the node turns out to be feasible and does not contain any subtour, the objective value of this node (z_c^*) is compared with the current upper bound (\bar{B}). If $z_c^* < \bar{B}$, the upper bound is updated as $\bar{B} = z_c^*$.

In the second approach, we solve the TSP directly considering constraints (4.3)-(4.7), (4.9) and (4.10). The problems we solve have a small number of nodes and large number of efficient arcs between node pairs. For these problems, both the branch and bound method and solving the TSP directly are computationally efficient.

During arc elimination, we find lower bounds (B) for the tours that include an arc between a specific node pair (a, b). We then find the difference between the upper and lower bounds and disregard the arcs between (a, b) that have a value exceeding the difference. A lower bound with the largest value would be preferred since it would provide a tighter upper bound (UB_{ab}) for the efficient arcs. Since the solution of the routing problem is a tour, the best lower bound would again be a tour. For this, we follow steps P.1 to P.4. Each time we need to find a tour, we use the software Concorde. We also use 1-tree bound as a lower bound. This method finds bounds more efficiently than the Concorde solution at the expense of providing less tight bounds.

4.7 Finding Efficient Arcs during the Interactive Algorithm

In our current approach, we find all the efficient arcs at the beginning of the algorithm. Then, we choose which arcs to use in finding the efficient tours and reduce the arcs

permanently or temporarily from the efficient arc set. Another approach could be finding the efficient arcs during the algorithm.

Finding Efficient Arcs in the Original Objective Space

To find efficient tours in the original objective space, we reduce the problem to a single objective problem by combining the two objectives linearly as $U(x) = wz_1(x) + (1 - w)z_2(x)$. In our current approach, we select the efficient arc that can be used between nodes (i, j) as follows:

$$e_{ij} = \begin{cases} 0 & i = j \\ \min_{k=1, \dots, K_{ij}} \{wz_1(i, j, k) + (1 - w)z_2(i, j, k)\} & i \neq j \end{cases}$$

Instead of selecting the arc to be used, we can find the efficient shortest arc that minimizes the composite objective value and assign this value to e_{ij} .

Finding Efficient Arcs in the Reduced Objective Space

In the reduced objective space, to find a tour, we again minimize the composite objective $U(x) = wz_1(x) + (1 - w)z_2(x)$. However, in this part, the constraints on the objective values of the tours indicate some constraints on the objective values of the efficient arcs. Therefore, for this part we do not need to find all efficient arcs.

We first reduce the objective space around the triangles between the most preferred solution and its adjacent efficient solutions. Using these bounds on objectives, we find upper bounds on the objective values of the efficient arcs using some lower bounds for the tour approximations.

To calculate these lower bounds for each node pair (a, b) , we need to give the efficient arcs connecting other node pairs as input. For the two upper bounds on the two objectives, we first need to find the two efficient arcs between each node pair (i, j) , one with the lowest first objective value ($z_1(i, j)$) and the other with the lowest second

objective value ($z_2(i, j)$). Then we find a lower bound for objective p of node pair (a, b) using the following arc costs between each node pair (i, j) :

$$e_{ij} = \begin{cases} 0 & i = j \text{ or } (i, j) = (a, b) \\ z_p(i, j) & i \neq j \end{cases}$$

As a result, we have upper bounds (UB_{ab}^p) on the objectives $p = 1, 2$ for the efficient arcs of each node pair (a, b) . Then we need to find all efficient arcs inside the bounds of the first and second objectives to solve the constrained TSP using constraints (4.3)-(4.7), (4.9) and (4.10). Generation of all efficient arcs inside these bounds can be done using the ε -constraint method. In this method, we optimize one objective (p_1) and treat the remaining objective (p_2) as a constraint to satisfy an aspiration level, ε . While doing this, we can start with $\varepsilon = UB_{ab}^{p_2}$ and decrease ε until we find a solution with p_1^{th} objective value higher than $UB_{ab}^{p_1}$. We continue the remaining steps with these efficient arcs between each node pair.

4.8 Application to the UAV Routing Problem

We consider the route planning problem of UAVs through a defended area with a number of nodes (or targets) to visit. We divide the two dimensional terrain into equidistant grids as in Olsan (1993). We represent each grid point by two dimensions (x, y) that refer to latitude and longitude, respectively. The grid structure is shown in Figure 4.8 (Tezcaner and Köksalan, 2011). In the figure, the terrain is divided into 10 by 10 equidistant grids. We place 5 targets and 3 radar sites in the terrain. For the radar sites, we assume that the radar is located at the center of the circular area that is within reach of the radar. While going from the center to the circumference, the effectiveness of the radar decreases, leading to lower radar detection threat. Beyond the circular area, the radar is assumed to be ineffective.

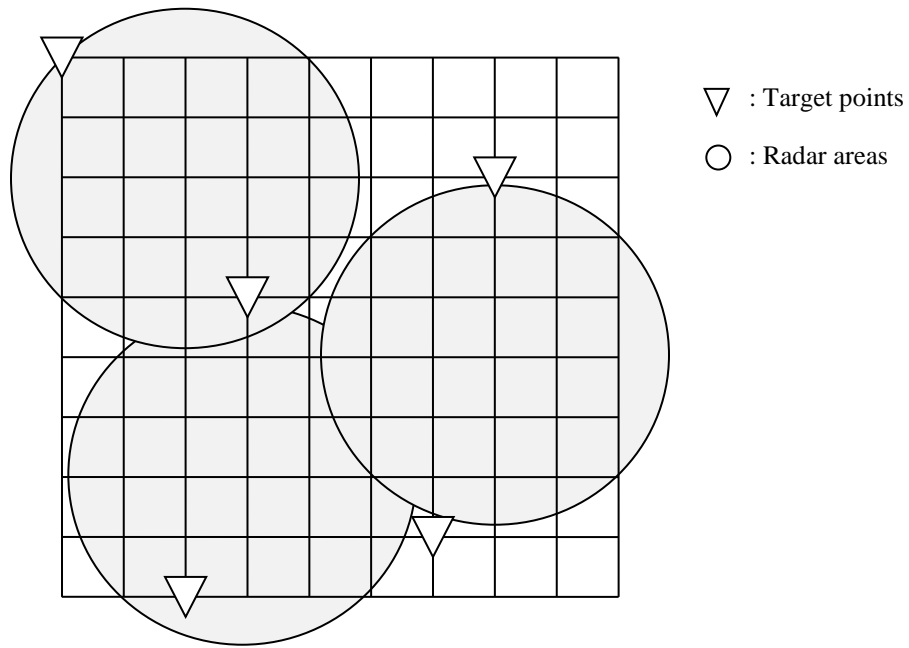


Figure 4.8 Terrain Structure for UAV Routing Problem

The UAV is assumed to be capable of moving to only its adjacent grids. The adjacent grid points of grid point A are shown in Figure 4.9.

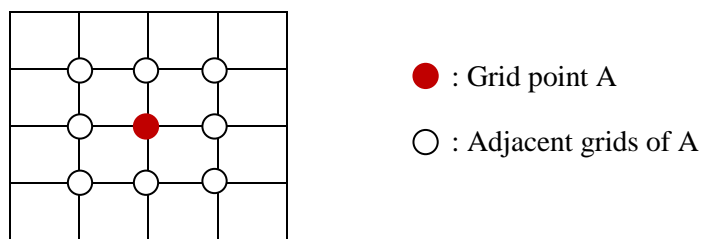


Figure 4.9 Movement Representation in two Dimensional Terrain

The objectives in this problem are the minimization of the distance traveled and the minimization of the radar detection threat. The first objective is also representative of the total flight duration if we assume constant speed. It can also approximately represent the total fuel consumption.

Let grid point g be represented by its x and y coordinates as (x_g, y_g) for $g \in G$, the distance and the radar detection probability between grid points g and f be d_{gf} and pd_{gf} , respectively. Let the binary decision variable y_{gf} denote whether the link between grid points g and f is used or not. Let the objectives, minimization of distance traveled and minimization of radar detection threat similar to the formulation in Tezcaner and Köksalan (2011) be as follows:

$$\text{Min } z_1 = \sum_{g \in G} \sum_{f \in G} d_{gf} y_{gf} \quad (4.13)$$

$$\text{Min } z_2 = \sum_{g \in G} \sum_{f \in G} d_{gf} pd_{gf} y_{gf} \quad (4.14)$$

We sum all the distances between the used grid points to find the total traveled distance in (4.13). We use the measure in Gudaitis (1994) to calculate the radar detection probability. In (4.14), we multiply the distance traveled with the detection probability of the used grid points. Here, we assume that if the vehicle moves at constant speed, the distance traveled can represent the duration of the flight. Multiplying the duration with the detection probability, we obtain a meaningful measure of how long the vehicle has been exposed to a certain radar detection probability. Summing all the radar detection probabilities, we approximate the radar detection threat. We calculate the detection probability pd_{gf} using signal-to-noise ratios (S/N). We first find the S/N_g value of any grid point g using equation (4.15). Then we take the average of the values S/N_g and S/N_f to approximate the S/N value of the edge (g, f) .

$$S/N_g = 10 \log_{10} \left(\frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_t^2 R^4} \right) \quad (4.15)$$

where,

P_t : power transmitted by radar (watts)
 G_t : power gain of transmitting antenna
 L_t : Transmitting system loss
 λ : wave length 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 grid point g (meters)

In this formulation all the symbols correspond to constant values, except for R . Given $P_t, G_t, L_t, \lambda, T_s, B_n, K$ and σ are constants, we can combine them under a new constant $C = \frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_t^2}$. We use the values for the constants as in Tezcaner (2009). We take 0.5 square meters as the σ (Skolnik, 1980, p. 44). We assume C equals $2.27 \times 10^{15} \text{ m}^4$ ($=2270 \text{ km}^4$).

Knowing the approximate S/N_{gf} value of the edge (g, f) , we find its detection probability using equation (4.16).

$$p d_{gf} = \begin{cases} 1 & \text{if } S/N_{gf} > UB_{S/N} \\ \frac{(S/N_{gf}) - LB_{S/N}}{UB_{S/N} - LB_{S/N}} & \text{if } LB_{S/N} < S/N_{gf} \leq UB_{S/N} \\ 0 & \text{if } S/N_{gf} \leq LB_{S/N} \end{cases} \quad (4.16)$$

The values $LB_{S/N}$ and $UB_{S/N}$ are used as 5 and 15, respectively, in Gudaitis (1994). In our study, to obtain a larger number of efficient solutions, we use the values 15 and 30 for the parameters, $LB_{S/N}$ and $UB_{S/N}$, respectively, as in Tezcaner and Köksalan (2011).

4.9 An Example

We solve the UAV routing problem with 5 nodes and 3 radars that was first introduced by Tezcaner and Köksalan (2011).

We assume that the DM has an underlying Tchebycheff preference function $U(z) = \max[0.33(z_1 - z_1^*), 0.67(z_2 - z_2^*)]$ to be minimized, which we pretend is unknown to us at the beginning of the algorithm. The ideal points z_1^* and z_2^* are taken as 0. The efficient frontier of the problem can be seen in Figure 4.10 and the corresponding efficient solutions are given in Table 4.1. The best solution of the DM is the first efficient solution according to the underlying preference function. Before running the algorithm, we find all the efficient arcs between the node pairs. These efficient arcs can be seen in Table 4.2. In our implementation, we create tours for lower bound approximations to reduce the arcs, as discussed previously. To find the unsupported efficient solutions, we solve the constrained TSP using the constraints (4.3)-(4.7), (4.9) and (4.10) directly.

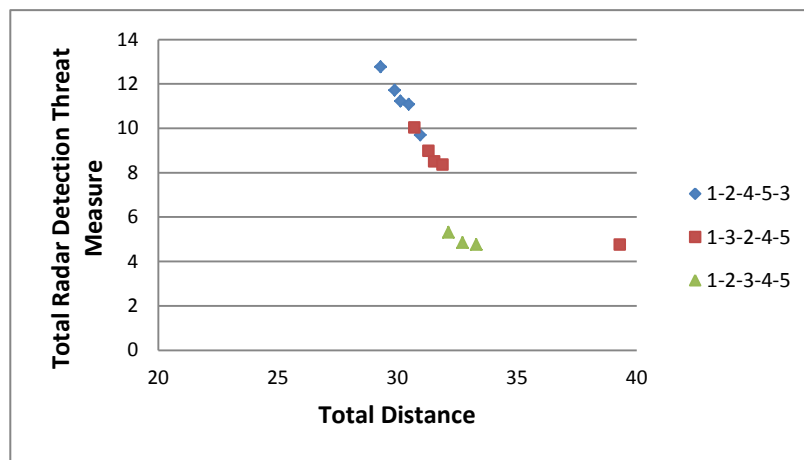


Figure 4.10 Efficient Frontier of the UAV Routing Problem

In executing the algorithm, we first reduce the search region between the two adjacent solutions (29.312, 12.764) and (32.140, 5.305) where the former is preferred according to the underlying preference function of the DM. This results in the arc reductions given in Table 4.2 where the reduced arcs are indicated with a checkmark. These reduced arcs are permanently deleted from the efficient arc sets and we continue the algorithm with the remaining arcs.

Table 4.1 Efficient solutions

Efficient Solution	Distance	Radar Detection Threat		Efficient Solution	Distance	Radar Detection Threat
1	29.312	12.764		8	31.554	8.491
2	29.898	11.704		9	31.898	8.344
3	30.140	11.226		10	32.140	5.305
4	30.484	11.079		11	32.726	4.851
5	30.726	10.029		12	33.312	4.757
6	30.968	9.688		13	39.312	4.744
7	31.312	8.969				

During the search inside the right triangle, we find unsupported efficient solutions and update the upper bounds on the combined objective values. Considering the upper bound on the combined objective, we reduce several arcs. In Table 4.3, we summarize a step where we find the three unsupported efficient solutions. We update the bounds on the composite objective and accordingly, we reduce several arcs between node pairs.

If P is the number of efficient solutions that constitute the efficient set, we need to find P solutions and ask for $P-1$ pairwise comparisons in the worst case. If P is small, the constraints in the objective space may not result in a favorable reduction both in the objective space and in the number of comparisons to reach the most preferred solution of the DM. For larger P values, the comparisons of the DM lead to larger areas of reduction in the objective space and, consequently, decreases the additional number of questions required to reach the most preferred solution. For our example problem, we

end up generating only 9 of the 13 efficient solutions. To reach the most preferred solution of the DM, we ask the DM to compare 4 solution pairs only. We reduce 37 efficient arcs considering the upper bound on the first objective and 10 efficient arcs considering the upper bounds on the combined objective values throughout the algorithm.

Table 4.2 Arc reduction for $z_1 \leq 32.140 - \epsilon$ and $z_2 \leq 12.764$

Node pair (i,j)	Distance	Radar Detection Threat	Reduced	Node pair (i,j)	Distance	Radar Detection Threat	Reduced
(1,2)	7.828	1.078		(2,5)	9.070	4.959	
(1,3)	5.242	3.765			9.656	3.847	√
	5.828	2.705			10.242	3.275	√
	6.414	2.080			10.484	2.976	√
	7.000	1.987			10.828	2.821	√
	7.828	1.726			11.07	2.252	√
	8.414	1.679	√		11.656	2.158	√
	9.828	1.294	√		11.898	1.894	√
(1,4)	10.484	4.985	√		12.484	1.44	√
	11.070	4.106	√		13.070	0.986	√
	11.656	3.482	√	(3,4)	5.242	1.401	
	12.242	2.978	√		5.828	1.307	√
	12.828	2.524	√		14.656	1.295	√
	13.656	2.263	√	(3,5)	5.414	3.574	
	14.828	2.156	√		6.242	2.036	
(1,5)	9.828	1.701			6.828	1.942	
	10.414	1.247			7.070	1.678	
(2,3)	4.828	0.216			7.656	1.224	
(2,4)	6.414	3.438			8.242	0.770	√
	7.000	3.352		(4,5)	4.414	0.909	
	7.242	1.900					
	7.828	1.446					
	8.656	1.185					
	9.828	1.078	√				

Table 4.3 The upper bounds on the combined objective and the reduced arcs

Unsupported Efficient Solution	Upper bounds on the combined objective	Reduced arcs (Node pair – Efficient arc)	
(31.898, 8.344)	$0.948z_1(x) + 0.052z_2(x) \leq 30.905$	-	
(31.554, 8.491)	$0.883z_1(x) + 0.117z_2(x) \leq 29.354$	(1,3)	(7.828, 1.726)
		(2,5)	(9.070, 4.959)
(30.726, 10.029)	$0.758z_1(x) + 0.242z_2(x) \leq 26.372$	(2,4)	(8.656, 1.185)
		(3,5)	(7.656, 1.224)

We also solve the interactive algorithm for randomly generated six-target problems for different efficient arc sizes between targets. We explain the results in Appendix B.

CHAPTER 5

ROUTING IN CONTINUOUS SPACE

The routing problem can be considered in many domains; routing of air vehicles, trucks on highways, trains on railways and vessels on oceans. For some of these domains, the possible arcs the vehicle can follow are specified before the routes are determined. For example, for the routing on railways, the path should be composed of several connections. The terrain of this problem can be considered as a network. For the routing problem of air vehicles, we have infinite number of arc alternatives between any two points (targets). There can be regions that are forbidden or that cannot be passed through (like mountains). Beyond the unreachable areas, the vehicle can move to any place in the terrain in practice. Therefore, considering a continuous terrain is a more realistic representation, although it brings computational difficulties.

In this part of our study, we consider the bi-objective routing problem of air vehicles. For these problems, we may have many objectives; minimization of distance traveled, radar detection threat, fuel consumption, flight duration, risks from weather conditions etc. In our study, we specifically consider the routing problem of UAVs for the two objectives; minimization of distance traveled and minimization of radar detection threat. Previously in Chapter 4, we discretized the two dimensional terrain into equidistant grid points and allowed for movement only between adjacent grid points, for this problem. The routes were defined by a number of grid points that the vehicle should pass in succession. The efficient routes may then be restructured letting the vehicle make smoother direction changes. With the discretization, we only allow certain moves. Since some possible moves are restricted, the obtained moves may not be efficient in reality. Routing in continuous space is more realistic as it overcomes this

deficiency of the discretized space at the expense of higher computation times and complexity of finding the efficient frontier.

There are many studies on routing problems for UAVs in the literature. Olsan (1993), Gudaitis (1994) and Yavuz (2002) consider a discretized terrain. They all approximate three dimensional terrain with equidistant grid points. Some studies consider the routing problem in a continuous terrain. The studies of Pachter and Hebert (2002) and Kan et al. (2011) are examples of this field. Pachter and Hebert (2002) develop the path trajectory model for UAV in continuous terrain. They find the optimal trajectory that minimizes the power reflected from the radar given that the vehicle can move a constant distance. Kan et al. (2011) propose a heuristic search algorithm in which the DM can provide some preferences over the altitude of the vehicle. They use the same objective as Pachter and Hebert (2002).

We first explain how we find the efficient frontier of the bi-objective shortest path problem between any two points. Then, we solve the bi-objective traveling salesperson problem in conjunction with the bi-objective shortest path problem.

5.1 Terrain Structure and Objectives

We first explain the bi-objective shortest path problem between two targets.

The continuous space representation of the discretized terrain in Figure 4.8 of 5-target UAV routing problem can be seen in Figure 5.1. In the continuous space, we allow movement to any point in the terrain. We indicate the placement of each object on the terrain by their x and y coordinates. In the figure, the targets are placed at $(0,9)$, $(2,0)$, $(3,5)$, $(6,1)$ and $(7,7)$. The radars are located at coordinates $(2,7)$, $(3,2)$, and $(7,4)$.

Here, $pd_{(x,y)}$ is the detection probability of a point (x, y) and ds is an infinitesimal part in the movement of the vehicle inside the radar region. This equation can be used to approximate the radar detection threat for any function of detection probability, pd . As the expression for pd becomes more complex, so does the solution of this measure. We use the following function to approximate the detection probability of any point (x, y) inside a radar region with center at (k, l) .

$$pd_{(x,y)} = \begin{cases} 1 & S/N_{(x,y)} > UB_{S/N} \\ \frac{S/N_{(x,y)} - LB_{S/N}}{UB_{S/N} - LB_{S/N}} & \text{if } LB_{S/N} < S/N_{(x,y)} \leq UB_{S/N} \end{cases} \quad (5.3)$$

where $S/N_{(x,y)} = 10 \log_{10} \left(\frac{C}{((x-k)^2 + (y-l)^2)^2} \right)$.

Here, we use 15 and 30 for $LB_{S/N}$ and $UB_{S/N}$, respectively, as in Tezcaner and Köksalan (2011). We assume C equals 2270 km⁴.

The radii of the radar are determined based on formula (5.3). Beyond points where the signal-to-noise ratio of a point is less than $LB_{S/N}$, the radar is ineffective. The effective region of radar is where the S/N ratio is larger than $LB_{S/N}$; i.e.

$$10 \log_{10} \left(\frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_t^2 R^4} \right) \geq LB_{S/N}.$$

To make $10 \log_{10} \left(\frac{C}{R^4} \right) \geq LB_{S/N}$, R should be less than or equal to $R_{outer} =$

$$\left(\frac{C}{10^{(LB_{S/N}/10)}} \right)^{1/4}. R_{outer} = 2.9108 \text{ km for } LB_{S/N} = 15. \text{ The detection probability is 1}$$

for $10 \log_{10} \left(\frac{C}{R^4} \right) \geq UB_{S/N}$. $R_{inner} = \left(\frac{C}{10^3} \right)^{1/4} = 1.2274 \text{ km for } UB_{S/N} = 30$. Inside a

circle with radius $R_{outer} = 2.9108$ kms, the radar is effective with detection probability increasing from the circumference to the center between the values 0 and 1. Inside a

secondary circle with radius $R_{inner} = 1.2274$ km, the detection probability is 1. An

example radar region and two targets are shown in Figure 5.2. The starting target is

placed at coordinates (x_s, y_s) and the final target is placed at coordinates (x_f, y_f) . The

radar is located at (k, l) . In the inner area with radius R_{inner} , the probability of detection is 1. In the outer area with radius R_{outer} , the detection probability decreases from 1 to 0 (at the circumference). We refer the region where the detection probability is 1 as the inner radar region, and where the detection probability ranges between 0 and 1 as the outer radar region.

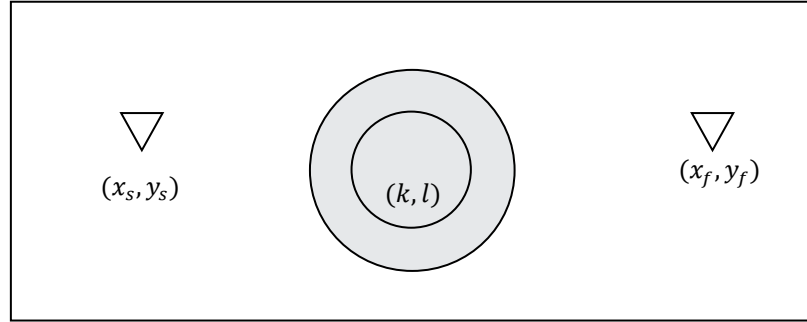


Figure 5.2 Continuous Terrain Representation

In the continuous space, the radar detection probability depends on the points that the vehicle moves inside the radar region. Let (x_{en}, y_{en}) and (x_{ex}, y_{ex}) denote the coordinates of the entrance and exit points of the vehicle to the radar region, respectively. For all the moves that are in the radar's ineffective region, the detection probability is zero. Then, the total radar detection threat measure (RDT) can be calculated as follows:

$$RDT = \int_{(x_{en}, y_{en})}^{(x_{ex}, y_{ex})} pd_{(x,y)} ds \quad (5.4)$$

Let the indicator function, $I_{[u,v)}(t) = 1$ if $t \in [u, v)$ and 0, otherwise. The overall radar detection threat can be found as follows:

$$RDT = \int_{(x_{en}, y_{en})}^{(x_{ex}, y_{ex})} \left[\left(\frac{S/N_{(x,y)} - LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot I_{[LB_{S/N}, UB_{S/N})} (S/N_{(x,y)}) + 1 \cdot I_{[UB_{S/N}, \infty)} (S/N_{(x,y)}) \right] ds$$

5.2 Transforming the Terrain Structure

The locations of the starting and final targets and the radar in between them can vary a lot. These variations result in different formulations for the RDT measure or the movement of the UAV inside the radar region. However, the two objectives D and RDT are calculated based on the relative placements of targets and radars with respect to each other. Therefore, we transform the locations of the targets and radars such that the radar center is located at the origin $(0,0)$. Suppose originally that the radar is located at (k, l) . To transform target t at (x_t, y_t) accordingly, we set the coordinates as $(x_t - k, y_t - l)$. The left and right extreme points of the radar circle become $(-R_{outer}, 0)$ and $(R_{outer}, 0)$, respectively. The top and bottom extreme points become $(0, R_{outer})$ and $(0, -R_{outer})$, respectively. We then rotate the locations of the targets such that the straight line connecting two targets make an angle θ with the x -axis such that $\theta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. Finally, we also make sure that the straight line passing through the two targets go above the radar's center.

The new coordinates (a', b') of a vector (a, b) when it is rotated around the origin by θ is found as follows:

$$\begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad (5.5)$$

Here, we make transformations for $\theta = \frac{\pi}{2}, \pi, \frac{3\pi}{2}$. With these changes, equation (5.3) reduces to equation (5.6).

$$pd_{(x,y)} = \begin{cases} 1 & S/N_{(x,y)} > UB_{S/N} \\ \frac{S/N_{(x,y)} - LB_{S/N}}{UB_{S/N} - LB_{S/N}} & \text{if } LB_{S/N} < S/N_{(x,y)} \leq UB_{S/N} \end{cases} \quad (5.6)$$

where $S/N_{(x,y)} = 10 \log_{10} \left(\frac{c}{(x^2 + y^2)^2} \right)$.

5.3 Movement between Two Targets

The two objectives (total distance traveled and total radar detection threat) conflict with each other only inside the radar regions. Since the radar is ineffective beyond the radar regions, any move at those regions would have zero radar detection probability. Therefore, in those regions, we minimize only the total distance traveled. Between any two points in the regions where the radar is ineffective, the vehicle should follow the straight line that connects the two points. This is the Euclidean distance; which is the shortest distance between any two points.

Inside the radar regions, the vehicle moves between an entrance point and an exit point. We assume that the vehicle follows a circular path inside those regions. The movement can be seen in Figure 5.3. The vehicle starts from (x_s, y_s) and moves along the straight line to reach the entrance point (x_{en}, y_{en}) to the radar region. It follows a circular path whose center is at (a, b) and radius is r , and exits the radar region at (x_{ex}, y_{ex}) . It then reaches the final target at (x_f, y_f) following the straight line that connects these two points.

The locations of the targets and the radar make the entrance and exit points follow the condition $x_{en} < x_{ex}$.

5.3.1 Movement Inside the Outer Radar Region

The vehicle follows a circular path defined as $(x - a)^2 + (y - b)^2 = r^2$ inside the radar area. It moves along straight lines beyond the radar region. The total distance traveled (D) is calculated as follows:

$D =$

$$\sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex} - x_{en})^2 + (y_{ex} - y_{en})^2}}{2r}\right) \cdot r + \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} \quad (5.7)$$

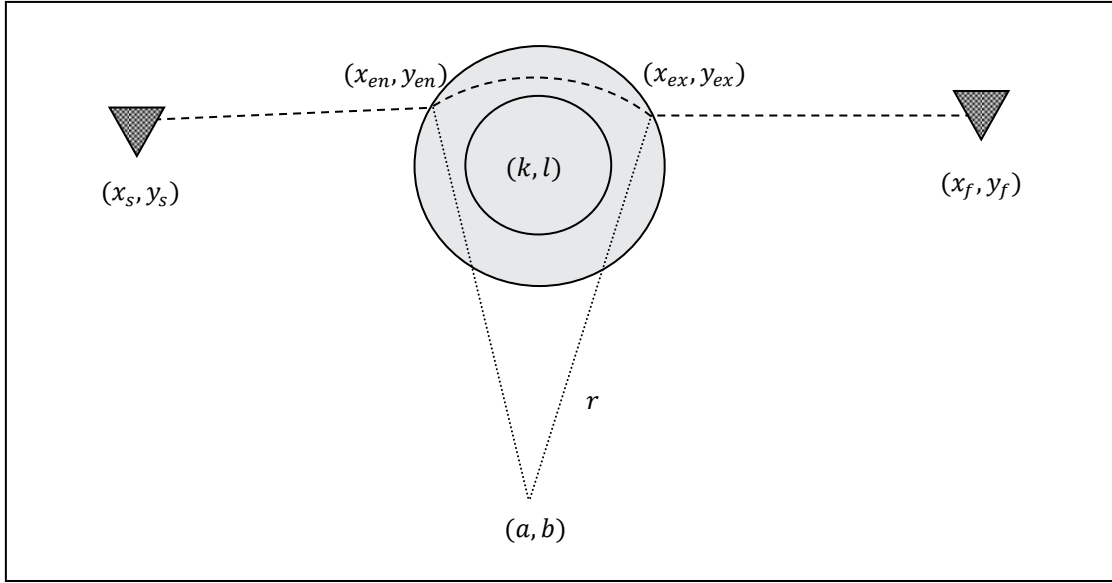


Figure 5.3 Movement Inside the Outer Radar Region

The first and the third terms are the Euclidean distances between (x_s, y_s) and (x_{en}, y_{en}) and (x_f, y_f) and (x_{ex}, y_{ex}) , respectively. The second term is the distance of the arc of $(x - a)^2 + (y - b)^2 = r^2$ between points (x_{en}, y_{en}) and (x_{ex}, y_{ex}) . Let A be the distance of the arc $(x - a)^2 + (y - b)^2 = r^2$ between points (x_{en}, y_{en}) and (x_{ex}, y_{ex}) and 2α be the angle between the two radii that connects (k, l) to (x_{en}, y_{en}) and (x_{ex}, y_{ex}) . We have the following two equations:

- The ratio of 2α to 2π equals the ratio of A to the circumference of the circle:

$$\frac{2\alpha}{2\pi} = \frac{A}{2\pi r}$$

- The sine of α is the ratio of half the Euclidean distance between (x_{en}, y_{en}) and (x_{ex}, y_{ex}) and the radius of the circle:

$$\sin(\alpha) = \frac{\sqrt{(x_{ex}-x_{en})^2+(y_{ex}-y_{en})^2}}{2r}$$

From these two equations, A is found as $2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex}-x_{en})^2+(y_{ex}-y_{en})^2}}{2r}\right) \cdot r$.

We find the RDT using (5.4).

$$RDT = \int_{(x_{en}, y_{en})}^{(x_{ex}, y_{ex})} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{(x^2 + y^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot ds \quad (5.8)$$

From Pythagoras' theorem, $ds^2 = dx^2 + dy^2$.

$$\frac{ds^2}{dx^2} = 1 + \frac{dy^2}{dx^2}$$

$$ds = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$$

Then, we can write the RDT in terms of the x coordinate as follows:

$$RDT = \int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{(x^2 + y^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \quad (5.9)$$

Since we transform the locations of the targets and the radar such that the movement is on the upper region of the radar's center, we consider the movement from the upper side of the circle $(x - a)^2 + (y - b)^2 = r^2$. Therefore, we write the y coordinate of the motion in terms of the x coordinate as $y = \sqrt{r^2 - (x - a)^2} + b$. This makes $\frac{dy}{dx} = \frac{1}{2\sqrt{r^2 - (x - a)^2}} \cdot (-2) \cdot (x - a) = \frac{a - x}{\sqrt{r^2 - (x - a)^2}}$. RDT reduces to the following equation:

$RDT =$

$$\int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + (\sqrt{r^2 - (x - a)^2} + b)^2\right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x - a)^2}} dx \quad (5.10)$$

5.3.2 Movement that Passes Through the Inner Radar Region

The vehicle can only be assumed to follow a circular path when the radar detection probability takes values between 0 and 1. In the regions where the detection probability is 1, it is optimal to minimize distance since each point in that region has the same detection probability. Therefore, for the regions where the detection probability is 1, we assume that the vehicle moves along a straight line and leaves the region by following the shortest trajectory. The move of the vehicle that passes through the inner radar region can be seen in Figure 5.4.

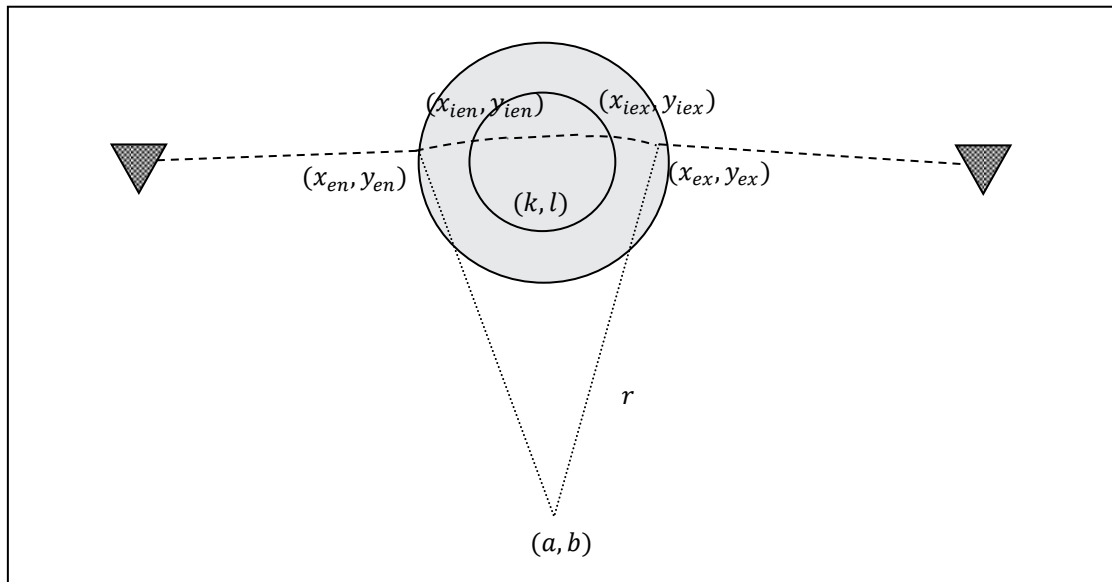


Figure 5.4 Movement Passing through the inner Radar Region

For this kind of movement, we define two additional points; the entrance point to the inner radar region (x_{ien}, y_{ien}) and the exit point from the inner radar region (x_{iex}, y_{iex}) . Between these two points, the vehicle should spend the shortest possible distance since inside the inner region the detection probability is 1. Therefore, the *RDT* inside the inner region is found as $\int_{x_{ien}}^{x_{iex}} (1) \cdot \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$. If the vehicle follows a

straight line, i.e. $y = mx + n$, $\frac{dy}{dx}$ equals m . Therefore, total radar detection threat in the inner region (RDT_i) equals $(x_{iex} - x_{ien})\sqrt{1 + m^2}$. This also equals the Euclidean distance between (x_{ien}, y_{ien}) and (x_{iex}, y_{iex}) .

$$RDT_i = \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2}$$

Knowing that (x_{ien}, y_{ien}) and (x_{iex}, y_{iex}) satisfy $y = mx + n$,

$$RDT_i = \sqrt{(x_{iex} - x_{ien})^2 + (mx_{iex} + n - (mx_{ien} + n))^2} = \sqrt{(x_{iex} - x_{ien})^2(1 + m^2)}$$

Similar to the inequality $x_{en} < x_{ex}$, the entrance and exit points to the inner radar region should satisfy the inequality, $x_{ien} < x_{iex}$. Therefore, $RDT_i = (x_{iex} - x_{ien})\sqrt{(1 + m^2)}$.

The overall radar detection threat is the sum of radar detection measures in the inner (RDT_i) and the outer (RDT_0) radar regions.

$$RDT = RDT_0 + RDT_i \tag{5.11}$$

where,

$$RDT_0 = \int_{x_{en}}^{x_{iex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b))^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx +$$

$$\int_{x_{iex}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b))^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx$$

$$RDT_i = \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2}$$

The total distance traveled is the sum of distances between the consecutive points; (x_s, y_s) , (x_{en}, y_{en}) , (x_{ien}, y_{ien}) , (x_{iex}, y_{iex}) , (x_{ex}, y_{ex}) and (x_f, y_f) .

$D =$

$$\sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin \left(\frac{\sqrt{(x_{ien} - x_{en})^2 + (y_{ien} - y_{en})^2}}{2r} \right) \cdot r +$$

$$\sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex} - x_{iex})^2 + (y_{ex} - y_{iex})^2}}{2r}\right) \cdot r + \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} \quad (5.12)$$

5.4 Enumeration of Movement between Two Targets

The objectives (D, RDT) are nonlinear in terms of the decision variables. RDT cannot be simplified and taken out of the integral. We also tried to simplify the RDT objective by expressing the entrance and exit points in polar coordinates. Let the entrance and exit points be $(r \cdot \cos t_{en} + a, r \cdot \sin t_{en} + b)$ and $(r \cdot \cos t_{ex} + a, r \cdot \sin t_{ex} + b)$, respectively. We define RDT that passes through the outer radar region by integrating through angle t as follows:

$$RDT = r \int_{t_{en}}^{t_{ex}} \left(\log_{10} \left(\frac{((r \cdot \cos t + a)^2 + (r \cdot \sin t + b)^2)^{(4/3)}}{c^{(2/3)}/10} \right) \right) dt$$

This equation cannot be simplified further and taken out of the integral. Consequently, we could not find a closed form for efficient solutions. Therefore, we made an enumeration to see the structure of the efficient frontier. We first explain how we find the extreme efficient solutions. Then we continue with our enumeration method.

5.4.1 Finding Extreme Efficient Solutions

First Extreme Efficient Solution: Finding the Efficient Arc with the Shortest Distance

We start finding the efficient arcs with the arc that has the shortest (Euclidean) distance. For this solution, the vehicle moves along the line that connects the two targets with a straight line. The efficient arc that has the smallest distance is on the line, $y = \left(\frac{y_f - y_s}{x_f - x_s}\right) \cdot (x - x_s) + y_s$. The total distance of the first extreme solution (D_{E1}) is found as;

$$D_{E1} = \sqrt{(x_f - x_s)^2 + (y_f - y_s)^2} \quad (5.13)$$

If this efficient arc does not pass through any radar region, it also has the smallest radar detection threat value; $RDT_{E1} = 0$. Since we cannot find any other solution that is not dominated by this extreme efficient solution, we conclude that this is the only efficient solution connecting the two targets. To find whether this line passes through a radar region, we solve the following two equations simultaneously. If these equations do not have a common solution $(x, y) \in \mathbb{R}^2$, we conclude that this line does not pass through the radar region.

$$\text{Equation of the line: } y = \left(\frac{y_f - y_s}{x_f - x_s} \right) \cdot (x - x_s) + y_s \quad (5.14)$$

$$\text{Equation of the outer radar circle: } x^2 + y^2 = R_{outer}^2 \quad (5.15)$$

If, on the other hand, we find common solutions $(x, y) \in \mathbb{R}^2$ for equations (5.14) and (5.15), it means that this extreme efficient solution passes through the radar region. These common solutions are the entrance and exit points to the outer radar region, (x_{en1}, y_{en1}) and (x_{ex1}, y_{ex1}) . We next check whether this solution passes through the inner radar region as well, by solving the two equations (5.14) and (5.16) together.

$$\text{Equation of the inner radar circle: } x^2 + y^2 = R_{inner}^2 \quad (5.16)$$

If we cannot find a common solution $(x, y) \in \mathbb{R}^2$, the vehicle does not pass through the inner radar region. The total radar detection threat of this solution is, then, found as follows:

$$RDT_{E1} = \int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + \left(\frac{y_f - y_s}{x_f - x_s} \right) \cdot (x - x_s) + y_s \right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{1 + \left(\frac{y_f - y_s}{x_f - x_s} \right)^2} dx \quad (5.17)$$

The common solutions $(x, y) \in \mathbb{R}^2$ for equations (5.14) and (5.16) are the entrance and exit points to the inner radar region, (x_{ien1}, y_{ien1}) and (x_{iex1}, y_{iex1}) . The total radar detection threat of this solution is found as follows:

$$\begin{aligned}
 RDT_{E1} = & \int_{x_{en}}^{x_{ien}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + \left(\frac{y_f - y_s}{x_f - x_s} \right) (x - x_s) + y_s \right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\left(1 + \left(\frac{y_f - y_s}{x_f - x_s} \right)^2 \right)} dx + \\
 & \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + \\
 & \int_{x_{iex}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + \left(\frac{y_f - y_s}{x_f - x_s} \right) (x - x_s) + y_s \right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\left(1 + \left(\frac{y_f - y_s}{x_f - x_s} \right)^2 \right)} dx
 \end{aligned} \tag{5.18}$$

An example for the first extreme efficient solution that passes through both the inner and outer radar regions can be seen in Figure 5.5.

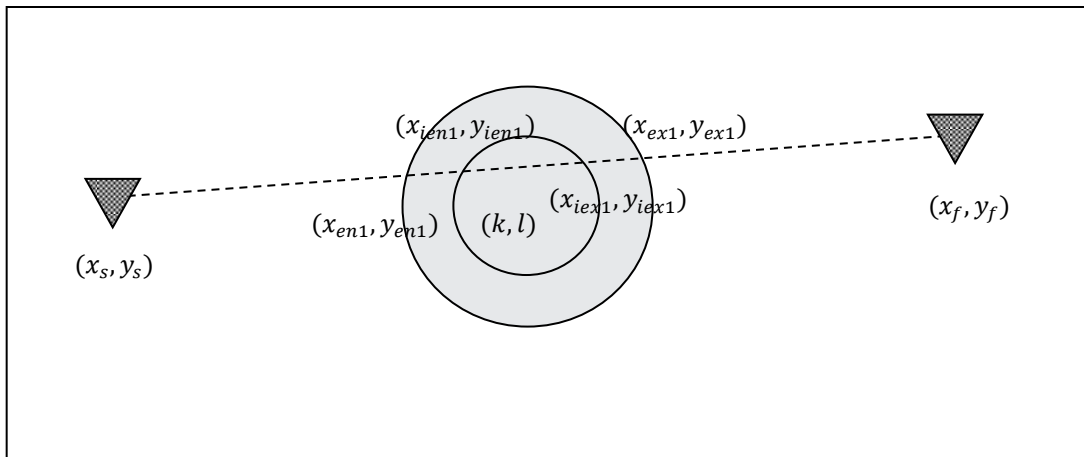


Figure 5.5 First Extreme Efficient Solution

Second Extreme Efficient Solution: Finding the Efficient Arc that has the Smallest Radar Detection Threat

Before explaining the solution, we first state our assumption that both targets are located outside the effective radar area. This allows the vehicle to move through points outside the effective region of the radar and have efficient arcs with zero radar detection threat. If these assumptions are not satisfied, then we need to make slight changes in the procedure.

Any move through the regions where the radar is ineffective has zero radar detection probability; so there are many solutions with zero detection probability. However, we also have to minimize the first objective; total distance traveled. This solution should be composed of two tangent lines from the targets to the circle, and an arc on the circle connecting the two tangent points. There are 2 tangent lines to a circle from a point. An example can be seen in Figure 5.6. The tangent lines from (x_s, y_s) intersect the circle at the points of tangency; (x'_{en2}, y'_{en2}) and (x''_{en2}, y''_{en2}) . Similarly, the tangent lines from (x_f, y_f) intersect the circle at (x'_{ex2}, y'_{ex2}) and (x''_{ex2}, y''_{ex2}) . We illustrate the solution for the connection between (x_s, y_s) and (x_{en}, y_{en}) . The connection between (x_{ex}, y_{ex}) and (x_f, y_f) is found in a similar way.

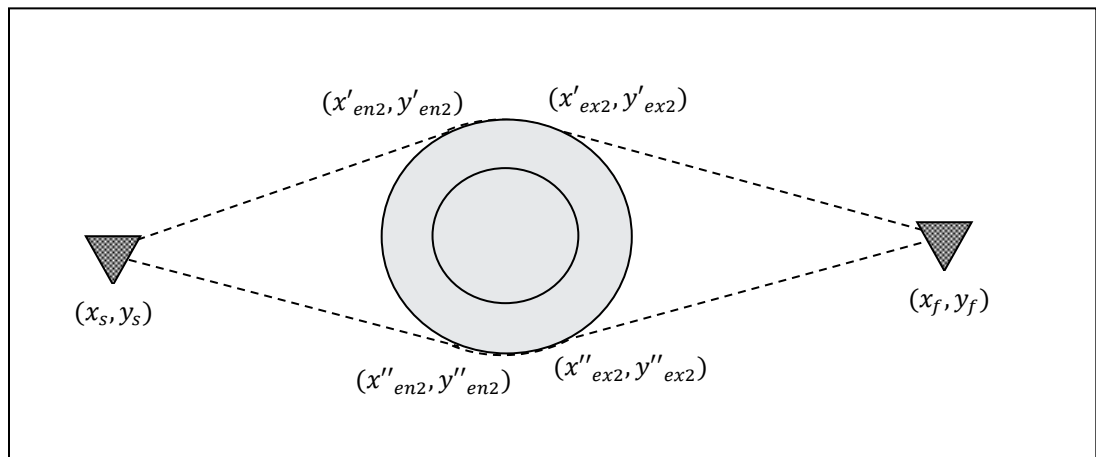


Figure 5.6 The Second Extreme Efficient Solution

We first find the slope of the line (which is tangent to the radar area) that passes through (x_s, y_s) and (x_{en}, y_{en}) . The slope, m , can be found using the formula that finds the shortest distance of a point to a line. Here, we know that the shortest distance from the radar center (k, l) to the line that passes through (x_s, y_s) and (x_{en}, y_{en}) equals R_{outer} ; since the tangent line is perpendicular to the line connecting the tangent point (x_{en}, y_{en}) and (k, l) at the tangent point (x_{en}, y_{en}) . The following is the formula that finds the shortest distance of a point to a line.

$$\frac{|m(k-x_s)+y_s-l|}{\sqrt{m^2+1}} = R_{outer}$$

For radar center at $(k, l) = (0,0)$, the equality reduces to $\frac{|y_s-mx_s|}{\sqrt{m^2+1}} = R_{outer}$.

Here, we find two different values of m , since from a point, we can draw two tangent lines to a circle. Solving the following two equations together for both m values, we find the point of tangencies (x'_{en2}, y'_{en2}) and (x''_{en2}, y''_{en2}) .

$$y = m(x - x_s) + y_s$$

$$x^2 + y^2 = R_{outer}^2.$$

For the example in Figure 5.6, we have two options for the arc with zero radar detection threat; we either pass from (x'_{en2}, y'_{en2}) and (x'_{ex2}, y'_{ex2}) (*alternative 1*) or (x''_{en2}, y''_{en2}) and (x''_{ex2}, y''_{ex2}) (*alternative 2*). The total distances of alternative 1 (D_1) and alternative 2 (D_2) are calculated below:

$$D_1 = \sqrt{(x_s - x'_{en})^2 + (y_s - y'_{en})^2} + 2 R_{outer} \arcsin\left(\frac{\sqrt{(x'_{ex}-x'_{en})^2+(y'_{ex}-y'_{en})^2}}{2 R_{outer}}\right) + \sqrt{(x_f - x'_{ex})^2 + (y_f - y'_{ex})^2}$$

$$D_2 = \sqrt{(x_s - x''_{en})^2 + (y_s - y''_{en})^2} + 2 R_{outer} \arcsin\left(\frac{\sqrt{(x''_{ex}-x''_{en})^2+(y''_{ex}-y''_{en})^2}}{2 R_{outer}}\right) + \sqrt{(x_f - x''_{ex})^2 + (y_f - y''_{ex})^2}$$

The solution with the smaller distance would dominate the other. Due to the structure of tangent lines, the distances from the starting and final targets to the tangent points

are equal; i.e. $\sqrt{(x_s - x'_{en})^2 + (y_s - y'_{en})^2} = \sqrt{(x_s - x''_{en})^2 + (y_s - y''_{en})^2}$. Therefore, the alternative with the shortest arc length would have a shorter arc length. The total distance of the second extreme solution equals $D_{E2} = \min(D_1, D_2)$.

Since the straight line connecting (x_s, y_s) and (x_f, y_f) goes over the center of the radar, the first arc with distance D_1 is shorter than the second arc. Therefore, the distance of this extreme solution is $D_{E2} = D_1$.

We know for the second extreme solution that $RDT_{E2} = 0$.

5.4.2 Finding Remaining Efficient Solutions

In the enumeration, we use the polar coordinates of entrance and exit points. The Cartesian coordinates of any point (x, y) on the circle $(x - a)^2 + (y - b)^2 = r^2$ can be written as $x = r \cdot \cos \theta + a$ and $y = r \cdot \sin \theta + b$ for $\theta \in (0, 2\pi)$. For the points on the radar circle $x^2 + y^2 = R_{outer}^2$, we have $x = R_{outer} \cos \theta$ and $y = R_{outer} \sin \theta$ for $\theta \in (0, 2\pi)$. Figure 5.7 shows the correspondence of parameters of polar coordinates.

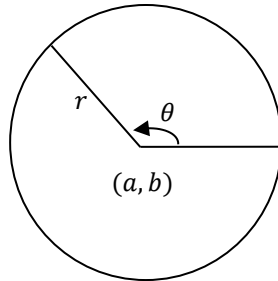


Figure 5.7 Coordinates of a Point on the Radar Circle

Enumeration of a subset of possible arcs between (x_s, y_s) and (x_f, y_f) can be done using the following steps. For this, we use the property that the entrance (x_{en}, y_{en}) and

exit (x_{ex}, y_{ex}) points of the intermediate efficient solutions should satisfy the following inequalities:

$$y_{en1} \leq y_{en} \leq y_{en2}$$

$$y_{ex1} \leq y_{ex} \leq y_{ex2}$$

We next present the steps of the algorithm.

Step N.1. Find the first extreme efficient solution that has the least distance. If it does not pass through the radar region, there is a single efficient solution. Terminate the algorithm. Otherwise, set the first entrance and exit points of this solution to (x_{en1}, y_{en1}) and (x_{ex1}, y_{ex1}) , respectively and go to Step N.2.

Step N.2. Find the second extreme efficient solution that has zero radar detection threat. Set the first entrance and exit points of this solution to (x_{en2}, y_{en2}) and (x_{ex2}, y_{ex2}) , respectively.

Step N.3. For $j = 1, 2$, find the parametric coordinates $\theta_{en,j}$ and $\theta_{ex,j}$ of the entrance and exit points $(x_{en,j}, y_{en,j})$ and $(x_{ex,j}, y_{ex,j})$, respectively. Set $k = 1$.

Step N.4.

For $\theta_{en} = \theta_{en,1} : \theta_{en,2}$ by α decrements;

for $\theta_{ex} = \theta_{ex,2} : \theta_{ex,1}$ by α decrements;

for $r = R_{outer} : 100$ by β increments;

Set the entrance point to the radar region (x_{en}^k, y_{en}^k) as $(R_{outer} \cos \theta_{en}, R_{outer} \sin \theta_{en})$, the exit point from the radar region (x_{ex}^k, y_{ex}^k) as $(R_{outer} \cos \theta_{ex}, R_{outer} \sin \theta_{ex})$.

Find the center of the movement circle (a, b) .

Find whether the arc enters the inner radar region. If it does, set the entrance and exit points to the inner radar region to (x_{eni}^k, y_{eni}^k) and (x_{exi}^k, y_{exi}^k) respectively.

Calculate the distance (D_k) and radar detection threat (RDT_k) of solution k .

Set $k \leftarrow k + 1$.

In the first three steps, we find the extreme efficient solutions. In the third step, we find the parametric coordinates using the formula $x_{en,j} = R_{outer} \cos \theta_{en,j}$, $x_{ex,j} = R_{outer} \cos \theta_{ex,j}$, $y_{en,j} = R_{outer} \sin \theta_{en,j}$, $y_{ex,j} = R_{outer} \sin \theta_{ex,j}$. In the fourth step, we enumerate a number of solutions by varying the entrance, exit points to the radar region and the radius of the circle that the vehicle follows inside the radar region. We change the entrance and exit angles by decrements of $\alpha=0.0034$, which corresponds to a decrement of 0.01 in Euclidean distances between consecutive entrance/exit points. We alter the radius of the movement from R_{outer} to 100 by $\beta=0.5$. The radius, $r = 100$ refers to almost a straight line rather than a circle. The least radius value is set to R_{outer} (that is also the radius of the outer radar region) to make each solution k pass through the radar region. Radius values smaller than R_{outer} would result in infeasible moves. We find the center of movement solving the following two equations together:

$$\begin{aligned} (R_{outer} \cos \theta_{en} - a)^2 + (R_{outer} \sin \theta_{en} - b)^2 &= r^2 \\ (R_{outer} \cos \theta_{ex} - a)^2 + (R_{outer} \sin \theta_{ex} - b)^2 &= r^2 \end{aligned}$$

To find whether the move enters the inner circle, we solve the two equations simultaneously:

$$\begin{aligned} x^2 + y^2 &= R_{inner}^2 \\ (x - a)^2 + (y - b)^2 &= r^2 \end{aligned}$$

We find the distance and radar detection threat of an arc k as follows:

$$\begin{aligned} D_k = & \sqrt{(x_{en}^k - x_s)^2 + (y_{en}^k - y_s)^2} + \sqrt{(x_f - x_{ex}^k)^2 + (y_f - y_{ex}^k)^2} + \\ & \left[2 \cdot \arcsin \left(\frac{\sqrt{(x_{ex}^k - x_{en}^k)^2 + (y_{ex}^k - y_{en}^k)^2}}{2r} \right) \cdot r \right] I_{\mathbb{R}^c}(x_{eni}^k) + \\ & \left[2 \cdot \arcsin \left(\frac{\sqrt{(x_{eni}^k - x_{en}^k)^2 + (y_{eni}^k - y_{en}^k)^2}}{2r} \right) \cdot r + \sqrt{(x_{exi}^k - x_{eni}^k)^2 + (y_{exi}^k - y_{eni}^k)^2} + \right. \\ & \left. 2 \cdot \arcsin \left(\frac{\sqrt{(x_{ex}^k - x_{exi}^k)^2 + (y_{ex}^k - y_{exi}^k)^2}}{2r} \right) \cdot r \right] I_{\mathbb{R}}(x_{eni}^k) \end{aligned}$$

$$\begin{aligned}
RDT_k = & \\
& \left[\int_{x_{en}^k}^{x_{ex}^k} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \right. \right. \\
& \left. \left. \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \right] I_{\mathbb{R}^c}(x_{eni}^k) + \\
& \left[\int_{x_{en}^k}^{x_{ex}^k} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \right. \right. \\
& \left. \left. \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx + \sqrt{(x_{exi}^k - x_{eni}^k)^2 + (y_{exi}^k - x_{eni}^k)^2} + \right. \\
& \left. \int_{x_{exi}^k}^{x_{ex}^k} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \right. \right. \\
& \left. \left. \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \right] I_{\mathbb{R}}(x_{eni}^k)
\end{aligned}$$

We enumerate the efficient frontier of an example problem using Matlab R2012a. The starting and final targets are placed at $(-10, -2)$ and $(9, 2)$, respectively. The radar is at $(0, 0)$. We find the first extreme efficient solution with entrance and exit points to the radar regions as $(-2.8678, -0.4985)$ and $(2.8253, 0.7001)$, respectively. The second extreme efficient solution enters the radar region at $(-1.3636, 2.5716)$ and exits at $(0.3056, 2.8947)$. We find the parameters for parametric representation as $\theta_{en,1} = 3.3132$, $\theta_{en,2} = 2.0584$, $\theta_{ex,1} = 0.2429$, $\theta_{ex,2} = 1.4656$. Varying the entrance and exit points, as well as the radius of the movement, we generate close to 26 million solutions. Among those solutions, 816 turn out to be efficient. The efficient frontier of the efficient arcs between $(-10, -2)$ and $(9, 2)$ can be seen in Figure 5.8.

We represent each solution (k) with its objective values as (D_k, RDT_k) . The right end of the efficient frontier corresponds to the second extreme solution $(20.2373, 0)$. The arcs of solutions between the second extreme solution and efficient solution $(19.559, 2.976)$ go through the inner radar region. For solutions with higher distances than

19.559, their efficient arcs go only through the outer radar region. We can also differentiate the solution that the vehicle starts moving only at the outer region from the efficient frontier. The frontier has a curved shape until a point with nearly 19.55 distance and 3 total radar detection threat. This point is very close to the efficient solution (19.559, 2.976). If the detection probability calculations stayed the same throughout the radar region, the curved shape would continue until the first extreme solution. Since we change both the formula of radar detection measure and the movement inside the inner radar region, we obtain lower detection values for the same distance value than we would have with the presumed movement and radar detection measure.

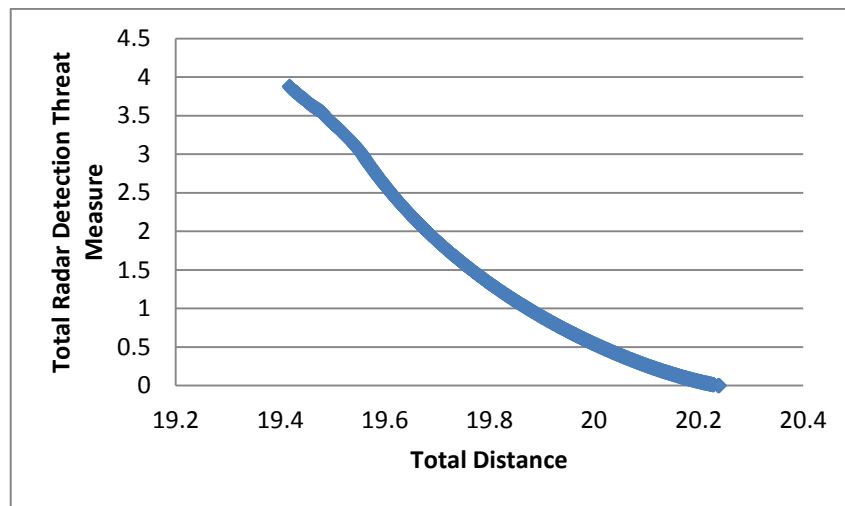


Figure 5.8 Efficient Frontier of the Example Shortest Path Problem

The efficient frontier in Figure 5.8 can be considered as an approximation of its original efficient frontier. We might have missed some efficient solutions since we enumerate only a subset of the solutions and since we assume a circular motion inside the radar region. Despite these, we observe that the middle points of the entrance (x_{en}^k, y_{en}^k) and exit (x_{ex}^k, y_{ex}^k) points of efficient solution k lie almost on a straight line, $y = \hat{u}x + \hat{v}$. We first find the middle point (x_m^k, y_m^k) of solution k where $x_m^k =$

$\frac{x_{en}^k + x_{ex}^k}{2}, y_m^k = \frac{y_{en}^k + y_{ex}^k}{2}$. We then perform a regression analysis to estimate the relationship of x and y -coordinates of the middle point of solution k . The results of the regression analysis can be seen in Appendix C. We perform the regression analysis just to fit the best line that shows the relationship of x_m^k and y_m^k for $\forall k$. Therefore, the assumptions on the error terms are not so important. Despite this, we evaluate the results in Appendix C. We explain 98.3% of the y -coordinate with the x -coordinate. The p -value of ANOVA ($p=0.000$) indicates that the model explains relation between x_m and y_m well. Also, p -values for both \hat{u} and \hat{v} are 0. Considering these results, we make a second assumption on the movement inside the radar region. We assume that the middle points of the entrance and exit points to the radar region lie on a line.

We consider three possibilities for the estimation of this line:

- a. The line that passes through the center of radar and the middle point of the entrance and exit points of the first extreme solution

$$y = -4.7435x ,$$
- b. The line that passes through the center of radar and the middle point of the entrance and exit points of the second extreme solution

$$y = -5.1666x ,$$
- c. The line that passes through the middle points of the entrance and exit points of the first and second extreme solutions

$$y = -0.0094 - 5.1843x .$$

The third option is closer to the regression equation; therefore it can be used as an estimate $y = \hat{v} + \hat{u}x$ for the line of middle points.

5.5 Generating the Efficient Frontier for the Bi-Objective Shortest Path Problem

The model that we solve to find an efficient solution depends on the position of that solution on the efficient frontier. If we search for an efficient solution that passes

through only the outer radar region, we only need to find the entrance (x_{en}, y_{en}) and exit (x_{ex}, y_{ex}) points to the radar region and the parameters of the circular motion (a, b, r) . If the efficient solution also passes through the inner radar region, we need to find additionally the entrance (x_{ien}, y_{ien}) and exit (x_{iex}, y_{iex}) points to the inner region.

We first give the nonlinear model for the solution that moves only in the outer radar region.

$$\begin{aligned} \text{Min } D = & \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex} - x_{en})^2 + (y_{ex} - y_{en})^2}}{2r}\right) \cdot r + \\ & \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} \end{aligned} \quad (5.19)$$

$$\text{Min } RDT = \int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2\right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \quad (5.20)$$

$$x_{en}^2 + y_{en}^2 = R_{outer}^2 \quad (5.21)$$

$$x_{ex}^2 + y_{ex}^2 = R_{outer}^2 \quad (5.22)$$

$$(x_{en} - a)^2 + (y_{en} - b)^2 = r^2 \quad (5.23)$$

$$(x_{ex} - a)^2 + (y_{ex} - b)^2 = r^2 \quad (5.24)$$

We minimize the first and second objectives in (5.19) and (5.20), respectively. The entrance and exit points' coordinates should satisfy the radar circle's equation ((5.21), (5.22)), and the circle that the vehicle moves on ((5.23), (5.24)). The decision variables for this problem are x_{en}, y_{en}, a, b and r .

For efficient solutions that pass through the inner radar region, we modify the objectives by introducing the entrance and exit points to the inner radar region as follows:

$$\begin{aligned}
\text{Min } D = & \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin \left(\frac{\sqrt{(x_{ien} - x_{en})^2 + (y_{ien} - y_{en})^2}}{2r} \right) \cdot r + \\
& \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + 2 \cdot \arcsin \left(\frac{\sqrt{(x_{iex} - x_{ex})^2 + (y_{iex} - y_{ex})^2}}{2r} \right) \cdot r + \\
& \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} \tag{5.25}
\end{aligned}$$

$$\begin{aligned}
\text{Min } RDT = & \int_{x_{en}}^{x_{ien}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx + \\
& \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + \\
& \int_{x_{iex}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \tag{5.26}
\end{aligned}$$

In addition to constraints (5.20)-(5.23), we add the following four constraints:

$$x_{ien}^2 + y_{ien}^2 = R_{inner}^2 \tag{5.27}$$

$$x_{iex}^2 + y_{iex}^2 = R_{inner}^2 \tag{5.28}$$

$$(x_{ien} - a)^2 + (y_{ien} - b)^2 = r^2 \tag{5.29}$$

$$(x_{iex} - a)^2 + (y_{iex} - b)^2 = r^2 \tag{5.30}$$

The entrance and exit points to the inner radar region should satisfy the inner radar circle's equation (5.27, 5.28) and they should be on the circle of movement (5.29, 5.30). We additionally have the decision variables x_{ien} and y_{ien} for this problem.

For the routing problem in the continuous space, we observe that the efficient frontier is continuous. Therefore, we can find an efficient solution corresponding to each D value for $D_{E1} \leq D \leq D_{E2}$. Let D_{tan} be the distance value of the efficient solution that moves tangent to the inner region. To find the k^{th} solution with distance value D_k , if $D_k \leq D_{tan}$, we minimize (5.26) subject to constraints (5.21)-(5.24), (5.27)-(5.30) and (5.31).

$$\begin{aligned}
& \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ien} - x_{en})^2 + (y_{ien} - y_{en})^2}}{2r}\right) \cdot r + \\
& \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{iex} - x_{ex})^2 + (y_{iex} - y_{ex})^2}}{2r}\right) \cdot r + \\
& \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} = D_k
\end{aligned} \tag{5.31}$$

On the other hand, if $D_k > D_{tan}$, we minimize (5.20) subject to constraints (5.21)-(5.24) and (5.32).

$$\begin{aligned}
& \sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex} - x_{en})^2 + (y_{ex} - y_{en})^2}}{2r}\right) \cdot r + \\
& \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} = D_k
\end{aligned} \tag{5.32}$$

We use LINDOGlobal solver in the GAMS Optimization Package. LINDOGlobal finds global optimal solutions for nonlinear programs. It supports many nonlinear functions including *log*, *arcsin* and *power*. For more details on this solver, please see <http://www.gams.com/dd/docs/solvers/lindo.pdf>.

Integral equations could not be expressed in Gams/LINDOGlobal. Therefore, we approximated the integral equation. We optimize the approximated function (that is a combination of the nonlinear functions that can be expressed in Gams/LINDOGlobal) subject to original constraints. We then find the second objective value by placing the optimal decision variable values to either (5.20) or (5.26).

There are a number of methods developed for approximating integral equations. Trapezoid Rule, Midpoint Rule and Simpson's Rule are three approximation methods that can be applied to integral equations (Adams, 1999, p.382). We briefly explain these methods for the approximation of the integral equation $\int_a^b f(x)dx$ in Appendix D.

Recall

that

$$f(x) = \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + \left(\sqrt{r^2 - (x-a)^2} + b \right)^2 \right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \cdot \sqrt{\frac{r^2}{r^2 - (x-a)^2}}.$$

We find the structure of the integral equation $f(x)$ in terms of x for different values of entrance and exit points and circular motion's characteristics (a, b, r) . Two of these graphics are given in Figure 5.9. For the graph in Figure 5.9(a), the path inside the radar region is $(x - 0.2008)^2 + (y + 1.0471)^2 = 3.9108^2$. We only check the structure of $f(x)$ within the radar region. The corresponding entrance and exit point's x -coordinates are $x_{en} = -1.65$ and $x_{ex} = 0.645$. For the graph in Figure 5.9(b), the path inside the radar region is $(x - 2.9878)^2 + (y + 14.9560)^2 = 16.511^2$ with $x_{en} = -2.8708$ and $x_{ex} = 2.4656$.

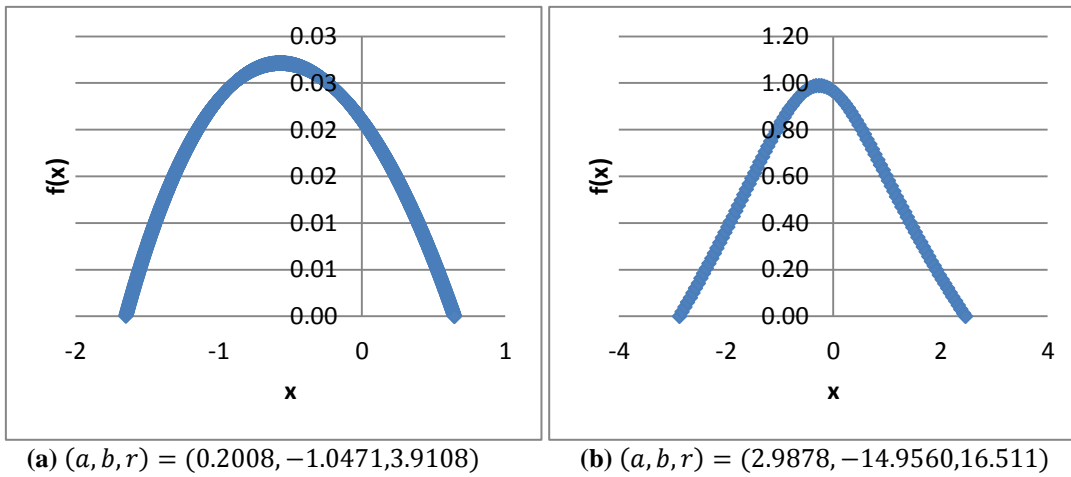


Figure 5.9 Graphs of $f(x)$ versus x for different (a, b, r) values

We also approximate five different radar values with different entrance and exit points and different parameters for the motion inside the radar region using the three approximation methods. The results can be seen in Table 5.1. As the number of intervals, n , increases, the approximation value gets closer to the original integral value $\int_a^b f(x) dx$. Despite this fact, we formed 2 or 4 intervals to keep the computation times

of the nonlinear program reasonable. Therefore, we check how these rules perform when we have $n = 2$ and $n = 4$. The approximation method that gives the closest value to the original value is bold. Although we will not use the approximate values instead of the original integral values, we use the approximation that fits to the original integral function more. Simpson's rule gives the closest values and $f(x)$ has more or less a curved shape. Therefore, we use Simpson's method for approximation.

Table 5.1 Approximation with different methods

(a, b, r)	$\int_a^b f(x)dx$	$n = 2$			$n = 4$		
		Trapezoidal Rule	Midpoint Rule	Simpson's Rule	Trapezoidal Rule	Midpoint Rule	Simpson's Rule
(0.2008,-1.0471,3.9108)	0.0418	0.0310	0.0472	0.0414	0.0391	0.0432	0.0418
(2.9878,-14.9560,16.511)	2.9156	2.6367	2.9847	3.5154	2.8107	2.9671	2.8687
(0.094,-0.5193,3.4108)	0.0151	0.0111	0.0170	0.0148	0.014	0.0155	0.0150
(0.5308,-2.6289,5.4108)	0.1417	0.1063	0.1594	0.1418	0.1329	0.1462	0.1417
(0.6658,-3.2404,5.9108)	0.2941	0.2226	0.3308	0.2968	0.2767	0.3038	0.2947

We approximate the integral equation with 2 and 4 intervals. The following is the approximation of the radar detection threat measure $RDT =$

$$\int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{\left(x^2 + (\sqrt{r^2 - (x-a)^2} + b) \right)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \quad \text{with}$$

2 intervals.

$$\text{Let } l = \frac{x_{ex} - x_{en}}{2}, x_0 = x_{en}, x_1 = \frac{x_{en} + x_{ex}}{2}, x_2 = x_{ex}.$$

$RDT \approx$

$$\begin{aligned}
& \left(\frac{x_{ex} - x_{en}}{6} \right) \left[\left[\left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x_0^2 + (\sqrt{r^2 - (x_0 - a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x_0 - a)^2}} \right] + \right. \\
& 4 \left[\left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x_1^2 + (\sqrt{r^2 - (x_1 - a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x_1 - a)^2}} \right] + \\
& \left. \left[\left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{c}{(x_2^2 + (\sqrt{r^2 - (x_2 - a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x_2 - a)^2}} \right] \right] \\
& \hspace{15em} (5.33)
\end{aligned}$$

5.6 Finding D_{tan}

We solve the inner region problem if we search for an efficient solution with $D \leq D_{tan}$ and solve the outer region problem if we search for an efficient solution with $D > D_{tan}$. To decide which model to solve, we need to know the value of D_{tan} .

To approximate the solution that first enters the inner radar region and has total distance D_{tan} , we follow a procedure similar to that we use for finding the second extreme efficient solution. We first find the upper tangents from the starting and final targets to the inner radar circle. We explain the steps for finding the tangent point of the starting target to the inner radar circle. We use the equality $\frac{|y_s - mx_s|}{\sqrt{m^2 + 1}} = R_{inner}$ to estimate the slope of the tangent line. This equality results in two values for m . We take the larger m for the tangent line's slope. We then solve the following two equations simultaneously to find the tangent point $(x_{tan,si}, y_{tan,si})$ to the inner radar circle from the starting target.

$$y = m(x - x_s) + y_s$$

$$x^2 + y^2 = R_{inner}^2.$$

We perform similar steps to find the tangent point $(x_{tan,fi}, y_{tan,fi})$ to the inner radar circle from the final target. We combine these tangent points with the arc on the inner circle. The total distance of this solution is found as follows:

$$\begin{aligned}
 D_{tan} = & \sqrt{(x_{tan,si} - x_s)^2 + (y_{tan,si} - y_s)^2} \\
 & + 2 R_{inner} \arcsin \left(\frac{\sqrt{(x_{tan,fi} - x_{tan,si})^2 + (y_{tan,fi} - y_{tan,si})^2}}{2 R_{inner}} \right) \\
 & + \sqrt{(x_f - x_{tan,fi})^2 + (y_f - y_{tan,fi})^2}
 \end{aligned}$$

To find the radar detection threat of the solution with $D = D_{tan}$, we need the entrance and exit points to the radar region. Solving the following two equations, we find the entrance point $(x_{tan,en}, y_{tan,en})$. The exit point $(x_{tan,ex}, y_{tan,ex})$ can be found similarly.

$$\begin{aligned}
 y &= \left(\frac{y_{tan,si} - y_s}{x_{tan,si} - x_s} \right) (x - x_s) + y_s \\
 x^2 + y^2 &= R_{outer}^2.
 \end{aligned}$$

The radar detection threat can be found as follows:

$$\begin{aligned}
& RDT_{tan} \\
&= \int_{x_{tan,en}}^{x_{tan,si}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + \left(\frac{y_{tan,si} - y_s}{x_{tan,si} - x_s} (x - x_s) + y_s \right)^2 \right)^2} - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \right. \\
&\quad \left. \sqrt{1 + \left(\frac{y_{tan,si} - y_s}{x_{tan,si} - x_s} \right)^2} dx \right. \\
&+ \int_{x_{tan,si}}^{x_{tan,fi}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + \left(\sqrt{R_{inner}^2 - x^2} \right)^2 \right)^2} - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{R_{inner}^2}{R_{inner}^2 - x^2}} dx \right. \\
&+ \int_{x_{tan,fi}}^{x_{tan,ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + \left(\frac{y_f - y_{tan,fi}}{x_f - x_{tan,fi}} (x - x_f) + y_f \right)^2 \right)^2} - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \right. \\
&\quad \left. \sqrt{1 + \left(\frac{y_f - y_{tan,fi}}{x_f - x_{tan,fi}} \right)^2} dx \right)
\end{aligned}$$

For the example problem, the parameters take the following values: $(x_{tan,en}, y_{tan,en}) = (-2.8908, 0.3404)$, $(x_{tan,ex}, y_{tan,ex}) = (2.5254, 1.4474)$, $(x_{tan,si}, y_{tan,si}) = (-0.3838, 1.1658)$ and $(x_{tan,fi}, y_{tan,fi}) = (-0.1043, 1.2230)$. The distance and radar detection threat values are 19.547 and 3.171, respectively. In the enumeration, we lastly have the solution (19.559, 2.976) that moves in the outer radar region which has objective values close to our approximate solution.

5.7 Implementation

We write the code to find efficient solutions corresponding to a distance D both for $D \leq D_{tan}$ and for $D > D_{tan}$ in GAMS/LINDOGlobal.

Finding RDT for $D > D_{tan}$

We need to determine the lower and upper bounds for the decision variables. The minimum x-coordinate of the entrance point is $-R_{outer}$ and the maximum x-coordinate

for the exit point is R_{outer} . The entrance point's x-coordinate can take at most the value of second extreme solution's entrance point's x-coordinate and the exit point's x-coordinate can at least take the value of the second extreme solution's exit point's x-coordinate. Since the movement inside the radar region is at the upper side of the movement's circle, we eliminate constraints (5.23) and (5.24) and write y_{en} and y_{ex} in constraints (5.21), (5.22), (5.32), (5.36) as: $y_{en} = \sqrt{r^2 - (x_{en} - a)^2} + b$ and $y_{ex} = \sqrt{r^2 - (x_{ex} - a)^2} + b$. We set the lower bound for the radius of motion to R_{outer} . We set its upper bound to 100. Since the move is at the upper side of the circular motion, the upper bound for the y-coordinate of the center of circular motion is set to the y-coordinate of the center of the radar region. We set its lower bound to -100. Similarly, we set the lower bound for the x-coordinate of the center of circular motion to the x-coordinate of the center of the radar region. We set its upper bound to 100.

Finding RDT for $D \leq D_{tan}$

For this case, we have additionally the four variables $x_{ien}, y_{ien}, x_{iex}, y_{iex}$. We again write y_{ien} and y_{iex} as $y_{ien} = \sqrt{r^2 - (x_{ien} - a)^2} + b$ and $y_{iex} = \sqrt{r^2 - (x_{iex} - a)^2} + b$. The lower and upper bounds on the variable x_{ien} is set to $-R_{inner}$ and the tangent from the final point to the inner circle, $x_{tan,fi}$, respectively. Similarly, x_{iex} is set between the tangent from the starting point to the inner circle, $x_{tan,si}$ and R_{inner} .

The solution durations are almost 7.5 hours for the routing problem in outer radar region in GAMS/LINDOGlobal. The model has difficulty when solving for routes in the inner radar region. To reduce the computational times, we make a simplification by introducing a property for the arc to be used. We assume that the middle points of the entrance and exit points to the outer and inner radar regions lie on a line. We use the line that passes through the middle points of the entrance and exit points of the first and second extreme solutions to estimate the equation of this line. This line can be seen in Figure 5.10. We also add the following constraint to make sure that the middle point of the entrance and exit points lie on a line:

$$\frac{y_{en}+y_{ex}}{2} = \hat{u} \left(\frac{x_{en}+x_{ex}}{2} \right) + \hat{v} \tag{5.34}$$

The solution durations decrease to 0.5 hours for routing in the outer radar region, however the model still continues to have difficulty for routing in the inner radar region. Since we have many efficient solutions, generating the whole efficient frontier would be computationally expensive. In the next part, we explain the heuristic we develop for finding the efficient solution that corresponds to a distance D .

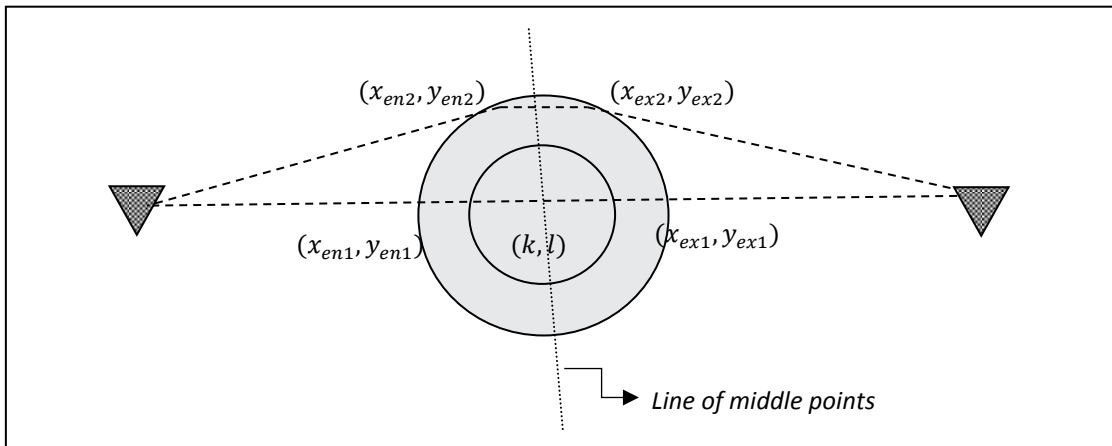


Figure 5.10 Line of Middle Points of Efficient Solutions

5.8 A Heuristic to Find the Efficient Solution Corresponding to a Distance D

The exact solution procedure requires a long computation time and it is not computationally efficient to generate the whole efficient frontier by solving the nonlinear program. Instead, we develop a heuristic that uses the axis of symmetry for the entrance and exit points.

If we assume that the middle points of the entrance and exit points to the radar region lie on a line $y = ux + v$, we can express x_m in terms of y_m . This reduces to finding the

radar detection threat of the solution that corresponds to a distance, D , by only knowing the variable y_m . We tried to write the radar detection threat in terms of only the unknown variable, y_m , however the equation became very complicated as we express all variables a, b, r, x_{en}, x_{ex} in terms of y_m . Therefore, we observe how RDT behaves for different y_m values when distance D is set to a value on the example problem. The variable y_m takes values in a range that makes the problem feasible for distance D . In Figure 5.11, we set D to take values between 19.7 and 20.2. The RDT seems to be a convex function in terms of y_m .

There are many solution methods to find the solution that minimizes unimodal functions. We first define these functions and go over some line search methods.

Definition 5.1. A function f over $S = [a, b]$ is unimodal if it has a unique minimum x^* in S if f is nonincreasing on the interval $\{x \in S: x \leq x^*\}$ and nondecreasing on the interval $\{x \in S: x \geq x^*\}$ (Bazaraa et al. 2006, p. 156)

For unconstrained optimization, Bazaara et al. (2006) explain a number of methods. Line search methods that do not use derivatives include dichotomous search method, golden section method and Fibonacci search method. The authors compare these methods and conclude that Fibonacci search method is the most efficient algorithm, followed by golden section method. We use Fibonacci Search Method in our approach. We explain the details of this method in Appendix E.

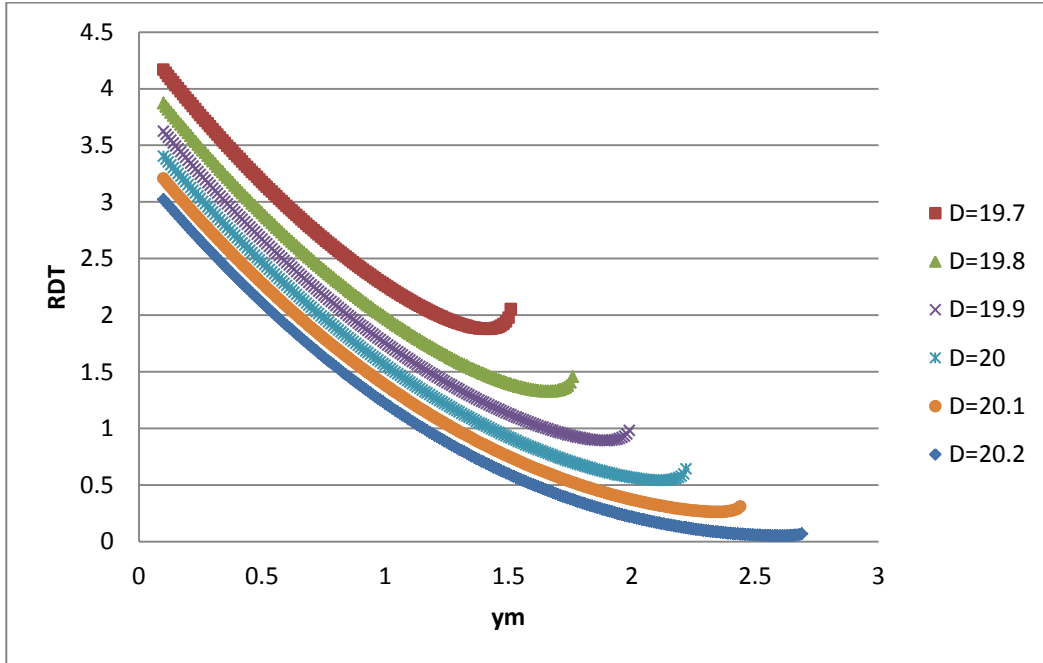


Figure 5.11 *RDT* vs. y_m Corresponding to Different D Values

We next explain the heuristic we develop for finding the *RDT* of the efficient solution that has total distance D .

Step H.1. Find the entrance and exit points of the first and second extreme solutions. Set $(x_{en,t}, y_{en,t})$ to t^{th} extreme efficient solution's entrance point and $(x_{ex,t}, y_{ex,t})$ to t^{th} extreme efficient solution's exit point for $t = 1, 2$.

Step H.2. Estimate the equation of the line, $y = ux + v$ that the middle points lie.

Step H.3. Set the minimum value (y_{min}) that y_m can take to the middle point of the first extreme solution. $y_{min} = \frac{y_{en1} + y_{ex1}}{2}$.

Step H.4. Find the maximum value (y_{max}) that y_m can take for a move with distance D .

Step H.5. Perform Fibonacci Search to find minimum *RDT* varying y_m between $[y_{min}, y_{max}]$.

The first 4 steps can be considered as initialization steps for the Fibonacci Search. We find the lower and upper bounds that make D feasible. In Step H.1, we search for the first and second extreme solutions. In Step H.2, we set the slope and intercept estimate using the line that passes through the middle points of the two extreme solutions' entrance $(x_{en,t}, y_{en,t})$ and exit points $(x_{ex,t}, y_{ex,t})$, for $t = 1, 2$. If the coordinates of t^{th} middle point is $x_m^t = \frac{x_{en,t} + x_{ex,t}}{2}$, $y_m^t = \frac{y_{en,t} + y_{ex,t}}{2}$, we set the slope estimate (\hat{u}) to $\frac{y_m^2 - y_m^1}{x_m^2 - x_m^1}$ and the intercept estimate (\hat{v}) to $y_m^1 - x_m^1 \left(\frac{y_m^2 - y_m^1}{x_m^2 - x_m^1} \right)$.

For any D , the vehicle can enter the radar region at the entrance and exit points of the first extreme solution. It can also enter from below this point but this would result in a higher radar detection threat. Therefore, we set the minimum middle point to the middle point of the first extreme solution's entrance and exit points. The upper bound solution we search in Step H.4 can be seen in Figure 5.12. This arc is composed of straight lines with a total distance of D . This is the highest point for the middle point of entrance and exit points and can be found by solving the following equations together.

$$x_{en,max}^2 + y_{en,max}^2 = R_{outer}^2 \quad (5.35)$$

$$x_{ex,max}^2 + y_{ex,max}^2 = R_{outer}^2 \quad (5.36)$$

$$x_{max} = \frac{x_{en,max} + x_{ex,max}}{2} \quad (5.37)$$

$$y_{max} = \frac{y_{en,max} + y_{ex,max}}{2} \quad (5.38)$$

$$y_{max} = \hat{u}x_{max} + \hat{v} \quad (5.39)$$

$$D = \sqrt{(x_{en,max} - x_s)^2 + (y_{en,max} - y_s)^2} + \sqrt{(x_{en,max} - x_{ex,max})^2 + (y_{en,max} - y_{ex,max})^2} + \sqrt{(x_f - x_{ex,max})^2 + (y_f - y_{ex,max})^2} \quad (5.40)$$

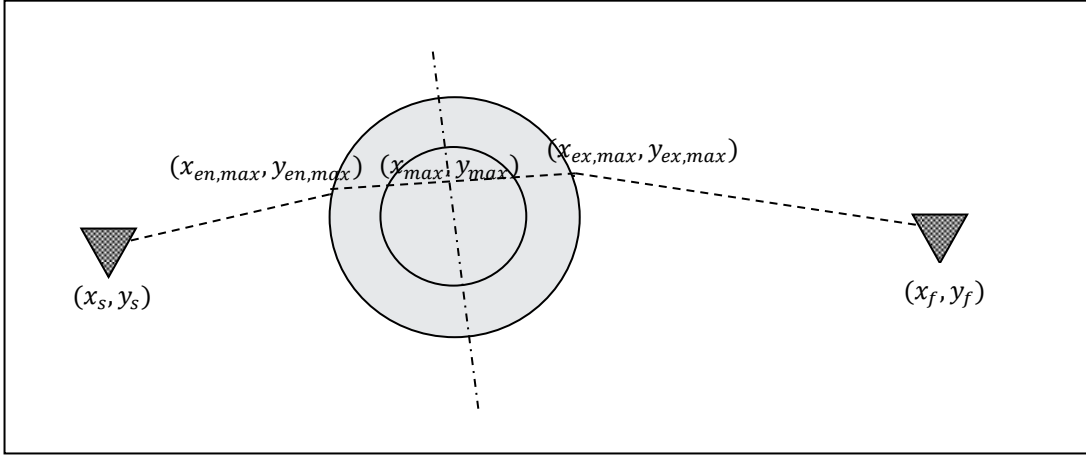


Figure 5.12 Finding y_{max} in Step 4

In (5.35) and (5.36), we make sure that the entrance and exit points are on the radar circle. We set the middle points' coordinates in (5.37) and (5.38). We set the total distance D to the summation of straight lines connecting consecutive points on the arc between (x_s, y_s) and (x_f, y_f) .

We perform the search for the y_m value that minimizes RDT in Step H.5. We follow the steps given below for Fibonacci Search for each y_m value.

H.5.1. Find x_m by:

$$x_m = \frac{y_m - \hat{v}}{\hat{u}} \quad (5.41)$$

H.5.2. Find the coordinates of the entrance and exit points that correspond to (x_m, y_m) by solving the following four equations together:

$$x_{en}^2 + y_{en}^2 = R_{outer}^2 \quad (5.42)$$

$$x_{ex}^2 + y_{ex}^2 = R_{outer}^2 \quad (5.43)$$

$$\frac{x_{en} + x_{ex}}{2} = x_m \quad (5.44)$$

$$\frac{y_{en} + y_{ex}}{2} = y_m \quad (5.45)$$

H.5.3. Find the distance left for the motion inside the radar region (D_R) by

$$D_R = D - \left(\sqrt{(x_{en} - x_s)^2 + (y_{en} - y_s)^2} + \sqrt{(x_f - x_{ex})^2 + (y_f - y_{ex})^2} \right) \quad (5.46)$$

H.5.4. Find the radius (r) of the circle of motion by

$$D_R = 2 \cdot \arcsin \left(\frac{\sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2}}{2r} \right) r \quad (5.47)$$

H.5.5. Find the center (a, b) of the circle of motion solving the following equations together:

$$(x_{en} - a)^2 + (y_{en} - b)^2 = r^2 \quad (5.48)$$

$$(x_{ex} - a)^2 + (y_{ex} - b)^2 = r^2 \quad (5.49)$$

H.5.6. Find RDT by

$$RDT = \int_{x_{en}}^{x_{ex}} \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{(x^2 + (\sqrt{r^2 - (x-a)^2} + b)^2)^2} \right) - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \quad (5.50)$$

Each time we need to find the corresponding RDT value for a y_m , we use equations (5.41)-(5.50). Among those equations, equation (5.47) is not directly solved. However, the function $G(r)$ (given in equation 5.51) is a unimodal function. Therefore, we use Fibonacci search to find the solution that minimizes $G(r)$. An example $G(r)$ function can be seen in Figure 5.13 for the constants $x_{en} = -2.3757, x_{ex} = 1.4459, y_{en} = 1.6819, y_{ex} = 2.5262$ and $D_R = 3.9608$.

$$G(r) = \left| 2 \cdot \arcsin \left(\frac{\sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2}}{2r} \right) r - D_R \right| \quad (5.51)$$

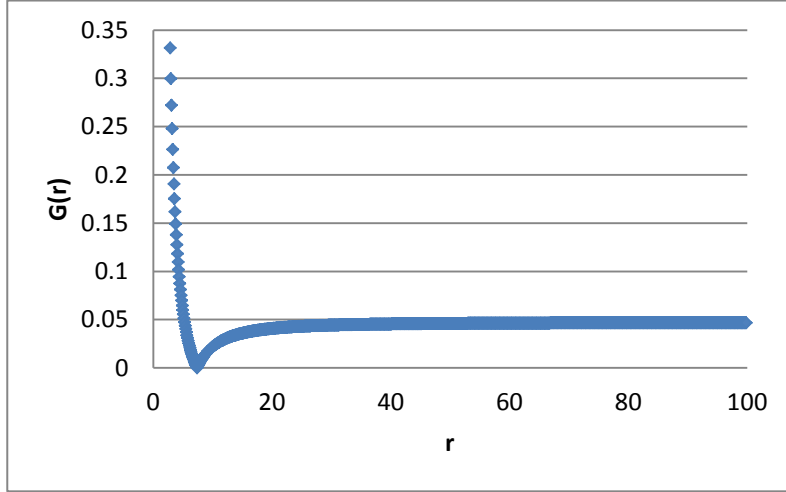


Figure 5.13 $G(r)$ vs. r

Corollary 1. $G(r) = \left| 2 \cdot \arcsin \left(\frac{\sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2}}{2r} \right) r - D_R \right|$ is a unimodal function.

Proof. We may write the numerator $\sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2}$ as a constant (Y) since it does not depend on r . $G(r)$ can be written as follows:

$$G(r) = \begin{cases} 2 \cdot \arcsin \left(\frac{Y}{2r} \right) r - D_R & 2 \cdot \arcsin \left(\frac{Y}{2r} \right) r \geq D_R \\ D_R - 2 \cdot \arcsin \left(\frac{Y}{2r} \right) r & 2 \cdot \arcsin \left(\frac{Y}{2r} \right) r < D_R \end{cases}$$

We know that $\frac{Y}{2} < r$, since r is the length of the hypotenuse of a triangle with one side's length $\frac{Y}{2}$. Let α be the angle that corresponds to side $\frac{Y}{2}$ for $0 < \alpha < \frac{\pi}{2}$.

In the first region, $\left(2 \cdot \arcsin \left(\frac{Y}{2r} \right) r \geq D_R \right)$, $\frac{dG(r)}{dr} = 2 \cdot \arcsin \left(\frac{Y}{2r} \right) - \frac{Y}{\sqrt{r^2 - \frac{Y^2}{4}}}$. This reduces to $2\alpha - 2 \tan(\alpha)$. α is always less than $\tan(\alpha)$ for $0 < \alpha < \frac{\pi}{2}$ since $\frac{d \tan(\alpha)}{d\alpha} =$

$\tan(\alpha)^2 + 1 \geq 1$. Therefore, the first derivative is negative for $0 < \alpha < \frac{\pi}{2}$. Its second derivative is, $\frac{d^2G(r)}{dr^2} = \frac{Y^3}{4r^4 \sqrt{\left(1 - \frac{Y^2}{4r^2}\right)^3}}$, which is always positive.

The second region, $\left(2 \cdot \arcsin\left(\frac{Y}{2r}\right) r < D_R\right)$, has exactly the reverse conditions for the first and second derivatives of $G(r)$.

The end points of this function are positive: $\lim_{r \rightarrow 0} G(r) \rightarrow D_R > 0$ and $\lim_{r \rightarrow \infty} G(r) \rightarrow D_R - \sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2} > 0$. In the first region, the function decreases from D_R with an increasing slope until it finds the minimum. In the second region, it increases from the minimum with a decreasing slope until it reaches $D_R - \sqrt{(x_{en} - x_{ex})^2 + (y_{en} - y_{ex})^2}$. Therefore, the function is unimodal. #

For the function $G(r)$, we also perform Fibonacci search to find the r that satisfies the equation (5.47).

In Step H.2, we estimate the slope and intercept of the line for middle points using the third option explained in Section 5.3. This may not be the exact equation for the line of middle points. Therefore we also search for the best slope of the middle points by altering the slope within the current value of the slope estimate. We change the slope estimate by small increments and decrements and move in the direction that performs better in terms of RDT . This way, we find a local minimum for the overall problem.

Suppose we finish one iteration of the heuristic and we are at the end of Step H.5 with a radar detection threat value of RDT_0 . After that step, we follow the steps outlined below to update the slope to the best value. We need to determine a step size (inc) that is used to update the slope estimate. Set iteration counter $k = 0$.

We next present the steps of the algorithm.

Step I.1. Set $k \leftarrow k + 1$. Increase the slope estimate (\hat{u}) by inc to find the right neighbor of the current slope estimate as $\hat{u}_R = \hat{u} + inc$. Go over the Steps H.4-H.5 of the heuristic and find a new RDT_R . If the entrance and exit points to the radar region are not within the first and second extreme solutions, set RDT_R a high number.

Step I.2. If $RDT_R < RDT_0$, update $\hat{u} = \hat{u}_R$ and $RDT_0 = RDT_R$. Go to Step I.1.

If $RDT_R > RDT_0$ and $k = 1$, go to Step I.3. If $RDT_R > RDT_0$ and $k > 1$, the local best slope estimate \hat{u} is found. Terminate the algorithm.

Step I.3. Set $k \leftarrow k + 1$. Decrease the slope estimate by inc to find the left neighbor of the current slope estimate as $\hat{u}_L = \hat{u} - inc$. Go over the Steps H.4-H.5 of the heuristic and find a new RDT_L . If the entrance and exit points to the radar region are not within the first and second extreme solutions, set RDT_L a high number. Go to Step I.4.

Step I.4. If $RDT_L < RDT_0$, update $\hat{u} = \hat{u}_L$ and $RDT_0 = RDT_L$. Go to Step I.3.

Else, the local best slope estimate \hat{u} is found. Terminate the algorithm.

By increasing (decreasing) the slope, we find the right (left) neighbor of the current slope. We finalize the search when the slope estimate results in the lowest RDT when compared to its two neighbor solutions. If the current slope estimate does not lead to a feasible solution; i.e., its entrance and exit points to the radar region are not within the extreme efficient solutions' entrance and exit points to the radar region, we do not consider that slope estimate.

5.8.1 An Example

We solve the problem we introduce in Section 5.3 for $D = 20$. We set the step size (inc) to 0.01. For the Fibonacci search, we set the parameters as $l = d = 0.000001$.

We report the iteration number, and the corresponding $\hat{u}, y_m, (x_{en}, x_{ex}), (y_{en}, y_{ex}), (a, b), r, RDT$ in columns 2-7, respectively. We reach the

best solution at iteration 9 with slope estimate -5.104 with total radar detection threat as 0.5420321.

Table 5.2 The Results of the Heuristic Iteratively for $D=20$

Iteration	\hat{u}	y_m	(x_{en}, y_{en})	(x_{ex}, y_{ex})	(a, b)	r	RDT
1	-5.184	2.117	(-2.330,1.745)	(1.510,2.489)	(0.943,-4.866)	7.376	0.5420715
2	-5.174	2.117	(-2.330,1.744)	(1.509,2.489)	(0.944,-4.863)	7.374	0.5420621
3	-5.164	2.116	(-2.331,1.743)	(1.508,2.490)	(0.945,-4.861)	7.372	0.5420539
4	-5.154	2.116	(-2.332,1.742)	(1.507,2.490)	(0.947,-4.862)	7.373	0.5420471
5	-5.144	2.116	(-2.332,1.741)	(1.506,2.491)	(0.949,-4.863)	7.375	0.5420414
6	-5.134	2.116	(-2.333,1.742)	(1.505,2.491)	(0.951,-4.861)	7.374	0.5420371
7	-5.124	2.116	(-2.334,1.740)	(1.504,2.492)	(0.953,-4.861)	7.374	0.5420341
8	-5.114	2.116	(-2.334,1.739)	(1.503,2.492)	(0.955,-4.860)	7.373	0.5420324
9	-5.104	2.116	(-2.335,1.738)	(1.502,2.493)	(0.957,-4.863)	7.376	0.5420321
10	-5.094	2.115	(-2.336,1.737)	(1.501,2.494)	(0.958,-4.861)	7.374	0.5420331

5.8.2 Finding RDT for $D \leq D_{tan}$

Searching for the RDT value of an efficient solution with $D \leq D_{tan}$ is not straightforward, since the UAV passes from both inner and outer radar regions. In Step 5.3, we find the distance left (D_R) for the movement in the radar region. In Step 5.4, we relate this D_R with the movement's radius, r . When the vehicle passes both from the inner and outer radar regions, the D_R is composed of three parts as follows:

$$D_R = 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ien} - x_{en})^2 + (y_{ien} - y_{en})^2}}{2r}\right) r + \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex} - x_{iex})^2 + (y_{ex} - y_{iex})^2}}{2r}\right) r$$

In this equation, we have five unknowns: $r, x_{ien}, y_{ien}, x_{iex}, y_{iex}$. We also know the relationship between the variables x_{ien}, y_{ien} and x_{iex}, y_{iex} by equations (5.27) and (5.28), respectively. Finding these unknowns from the three equations is not possible. Therefore, we use the following steps instead of Steps H.5.4-H.5.6. We search for the r with a trial and error method.

H.5.4’. Find the radius (r) of the circle of motion.

H.5.5’. Find the center (a, b) of the circle of motion.

H.5.6’. Find the entrance and exit points of the movement in the inner radar region, if they exist.

If $x_{ien}, y_{ien}, x_{iex}, y_{iex}$ cannot be found, go to Step H.5.6 in the heuristic for $D \geq D_{tan}$. Otherwise, go to Step H.5.7’.

H.5.7’. Find the actual distance traveled in the radar region using the following formula:

$$D_{Actual} = 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ien}-x_{en})^2+(y_{ien}-y_{en})^2}}{2r}\right)r + \sqrt{(x_{iex}-x_{ien})^2+(y_{iex}-y_{ien})^2} + 2 \cdot \arcsin\left(\frac{\sqrt{(x_{ex}-x_{iex})^2+(y_{ex}-y_{iex})^2}}{2r}\right)r \quad (5.52)$$

H.5.8’. Find difference ($Diff$) between the actual distance traveled (D_{Actual}) and

$$D_{Proposed} = D - \left(\sqrt{(x_{en}-x_s)^2+(y_{en}-y_s)^2} + \sqrt{(x_f-x_{ex})^2+(y_f-y_{ex})^2}\right) \text{ as follows:}$$

$$Diff = D_{Proposed} - D_{Actual} \quad (5.53)$$

If $Diff \neq 0$, increment D_R by $Diff$ as $D_R \leftarrow D_R + Diff$ and go to Step H.5.4’ .Otherwise, go to Step H.5.9’.

H.5.9’. Find RDT .

In Step H.5.4’ , we find r as if the UAV does not pass from the inner radar region with the equation.

$$D_R = 2 \cdot \arcsin\left(\frac{\sqrt{(x_{en}-x_{ex})^2+(y_{en}-y_{ex})^2}}{2r}\right)r \quad (5.54)$$

In Step H.5.5’ , we find the center of movement (a, b) using r by solving the following equations together:

$$(x_{en} - a)^2 + (y_{en} - b)^2 = r^2 \quad (5.55)$$

$$(x_{ex} - a)^2 + (y_{ex} - b)^2 = r^2 \quad (5.56)$$

We find the entrance and exit points of the movement to the inner radar region using H.5.6'. We solve the following equations together:

$$(x_{ien} - a)^2 + (y_{ien} - b)^2 = r^2 \quad (5.57)$$

$$x_{ien}^2 + y_{ien}^2 = R_{inner}^2 \quad (5.58)$$

$$(x_{iex} - a)^2 + (y_{iex} - b)^2 = r^2 \quad (5.59)$$

$$x_{iex}^2 + y_{iex}^2 = R_{inner}^2 \quad (5.60)$$

If the movement does not pass through the inner region, we go to Step H.5.6 in the original heuristic. Otherwise, we go to Step H.5.7' to find the actual distance traveled with the variables $r, x_{ien}, y_{ien}, x_{iex}, y_{iex}$. The difference with the actual and proposed distance is the additional distance that we can move additionally inside the radar region. Therefore, we increment the distance in the radar region D_R by the difference amount and go to Step H.5.4' again. We perform the search until the difference between the actual and proposed distances is zero. In the final step, we find the radar detection threat as follows:

$$\begin{aligned} RDT = \int_{x_{en}}^{x_{ien}} & \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + \left(\sqrt{r^2 - (x-a)^2} + b \right)^2 \right)^2} \right) \right. \\ & \left. - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx + \\ & \sqrt{(x_{iex} - x_{ien})^2 + (y_{iex} - y_{ien})^2} + \\ \int_{x_{iex}}^{x_{ex}} & \left(\frac{10}{UB_{S/N} - LB_{S/N}} \log_{10} \left(\frac{C}{\left(x^2 + \left(\sqrt{r^2 - (x-a)^2} + b \right)^2 \right)^2} \right) \right. \\ & \left. - \frac{LB_{S/N}}{UB_{S/N} - LB_{S/N}} \right) \sqrt{\frac{r^2}{r^2 - (x-a)^2}} dx \end{aligned} \quad (5.61)$$

5.9 Approximating the Efficient Frontier

The efficient solutions of any node pair (i, j) can be one of the three types.

- A single efficient solution that does not pass through the radar region
- A number of efficient solutions that only pass from the outer radar region
- A number of efficient solutions that pass from both the inner and outer radar regions

The first type of efficient solution corresponds to a problem where the shortest connection between nodes (i, j) does not pass through any radar region. We refer this set of nodes as E_{safe} . We refer any efficient arc between node pair (i, j) with their corresponding objective values. For $(i, j) \in E_{safe}$, we denote the single efficient solution as $(D_{E1(i,j)}, RDT_{E1(i,j)})$ for $RDT_{E1(i,j)} = 0$.

The second type of efficient solutions constitutes a curve shaped efficient frontier. We refer this set of nodes as E_{outer} . There is a continuous efficient frontier for this problem. Constructing the whole efficient frontier by finding discrete solutions is not possible. Therefore, we approximated the efficient frontier using a number of efficient solutions.

For the efficient arcs between node pair $(i, j) \in E_{outer}$, we denote the end points of the curve with the efficient arcs they correspond to. The rightmost efficient solution is the second extreme efficient solution $(D_{E2(i,j)}, RDT_{E2(i,j)})$ for $RDT_{E2(i,j)} = 0$ and the leftmost efficient solution is the first extreme efficient solution $(D_{E1(i,j)}, RDT_{E1(i,j)})$. To approximate the efficient frontier of nodes $(i, j) \in E_{outer}$, we fit a line using an L_q function that was developed by Köksalan (1999) and generalized by Köksalan and Lokman (2009). The function was utilized within an EA context in a bicriteria hub location problem by Köksalan and Soylu, (2010) and in a bicriteria routing problem by Tezcaner (2009). To fit this function, we need to know 3 points on the line; the end points of the curve and a central solution. Let the p^{th} objective value of solution f be

denoted as $z_p(f)$. Let $f = 1$ denote the left end solution of the curve (the first extreme solution), $f = 2$ denote the right end solution of the curve (the second extreme efficient solution) and $f = C$ denote a central solution on the curve. The L_q function is as follows:

$$(1 - zf_1^C)^q + (1 - zf_2^C)^q = 1 \quad (5.62)$$

where;

$zf^C = (zf_1^C, zf_2^C) = \left(\frac{z_1(C) - z_1(1)}{z_1(2) - z_1(1)}, \frac{z_2(C) - z_2(2)}{z_2(1) - z_2(2)} \right)$ are the normalized objective function values for the central solution.

Firstly, the objective function values of the central solution are normalized using the two extreme efficient solutions. Using these normalized objective values, we fit an L_q function using (5.62). For the computation of q , we solve a nonlinear program GAMS/LINDOGlobal with a single constraint (5.62) and a pseudo objective function.

For the efficient arcs between node pair $(i, j) \in E_{outer}$, the L_q function is as follows:

$$\left(1 - \frac{d_{ij} - D_{E1(i,j)}}{D_{E2(i,j)} - D_{E1(i,j)}} \right)^{q_{ij}} + \left(1 - \frac{r_{ij} - RDT_{E2(i,j)}}{RDT_{E1(i,j)} - RDT_{E2(i,j)}} \right)^{q_{ij}} = 1 \quad (5.63)$$

Using (5.63), we find the radar detection threat value (r_{ij}) of any solution between nodes $(i, j) \in E_{outer}$ given its distance value (d_{ij}) using the following formula:

$$r_{ij} = RDT_{E1(i,j)} - RDT_{E1(i,j)} \left(1 - \left(1 - \frac{d_{ij} - D_{E1(i,j)}}{D_{E2(i,j)} - D_{E1(i,j)}} \right)^{q(i,j)} \right)^{1/q(i,j)} \quad (5.64)$$

The third type of efficient solutions passes from both the inner and outer radar regions. We denote this set of nodes with E_{both} . The efficient frontier of this problem is composed of two parts: the almost linear part that represents the movement in the inner radar region and the curved part that represents the movement in the outer radar region. To approximate the motion in the linear part, it is sufficient to know two points on that line. The first point would be the extreme efficient solution that has the lowest distance and highest radar detection threat. We choose the second point to be the last point on

the line that has distance D_{tan} . The equation of the line that relates the two objectives for the movement in the inner radar region is $r_{ij} = m_{ij}d_{ij} + n_{ij}$. The curved part that represents the motion inside the outer radar region has the following formula:

$$\left(1 - \frac{d_{ij} - D_{tan(i,j)}}{D_{E2(i,j)} - D_{tan(i,j)}}\right)^{q_{ij}} + \left(1 - \frac{r_{ij} - RDT_{E2(i,j)}}{RDT_{tan(i,j)} - RDT_{E2(i,j)}}\right)^{q_{ij}} = 1 \quad (5.65)$$

An example efficient frontier of a node pair $(a, b) \in E_{both}$ can be seen in Figure 5.14. For this example, the first extreme solution corresponds to $(D_{E1(a,b)}, RDT_{E1(a,b)}) = (5, 10)$ and the second extreme solution corresponds to $(D_{E2(a,b)}, RDT_{E2(a,b)}) = (20, 0)$. The solution that corresponds to $D_{tan(a,b)}$ is $(D_{tan(a,b)}, RDT_{tan(a,b)}) = (10, 7)$. The equation for the efficient solutions that move in the inner radar region and outer region are $RDT = 13 - 0.6D$ and $\left(1 - \frac{D-10}{10}\right)^{1.407} + \left(1 - \frac{RDT}{7}\right)^{1.407} = 1$, respectively.

5.10 Generating the Efficient Frontier for the Bi-Objective Routing Problem

The bi-objective routing problem is a bi-objective TSP with multiple efficient arcs between nodes. The aim for this problem is to find tours that are composed of a subset of efficient arcs. For each pair of nodes, we firstly construct their efficient frontier knowing a number of efficient arcs.

Recall that E_{both} is the set of node pairs $(i, j), i < j$ whose efficient frontiers are composed of two regions, E_{outer} is the set of node pairs $(i, j), i < j$ whose efficient frontiers are composed of one region and E_{safe} is the set of node pairs $(i, j), i < j$ that have one efficient solution. The set of all node pairs constitute the set $E = E_{safe} \cup E_{outer} \cup E_{both}$. Let the variables d_{ij} and r_{ij} for $(i, j) \in E, i < j$ denote the distance and radar detection threat values chosen between nodes i and j .

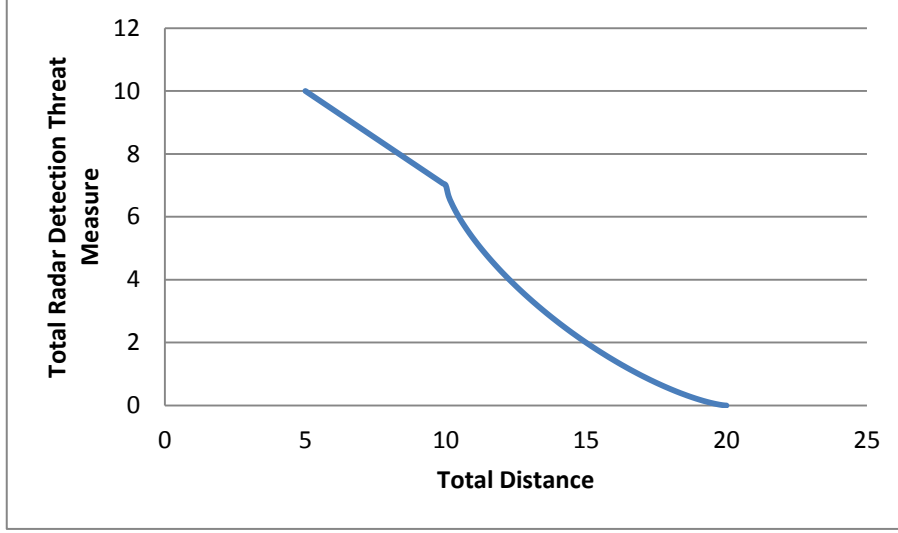


Figure 5.14 An Example Efficient Frontier Approximated for Nodes (a,b) in E_{both}

The formulation for the bi-objective TSP with node pairs connected with infinitely many efficient solutions is given below:

$$\text{Min } D = \sum_{(i,j) \in E} d_{ij} \quad (5.66)$$

$$\text{Min } RDT = \sum_{(i,j) \in E} r_{ij} \quad (5.67)$$

$$d_{ij} = D_{E1(i,j)}(1 - y_1(i,j)) \quad \forall (i,j) \in E_{safe} \quad (5.68)$$

$$r_{ij} = RDT_{E1(i,j)}(1 - y_1(i,j)) \quad \forall (i,j) \in E_{safe} \quad (5.69)$$

$$d_{ij} \leq D_{E2(i,j)}(1 - y_1(i,j)) \quad \forall (i,j) \in E_{outer} \quad (5.70)$$

$$d_{ij} \geq D_{E1(i,j)}(1 - y_1(i,j)) \quad \forall (i,j) \in E_{outer} \quad (5.71)$$

$$r_{ij} \leq RDT_{E1(i,j)}(1 - y_1(i,j)) \quad \forall (i,j) \in E_{outer} \quad (5.72)$$

$$r_{ij} \geq RDT_{E1(i,j)} - RDT_{E1(i,j)} \left(1 - \left(1 - \frac{d_{ij} - D_{E1(i,j)}}{D_{E2(i,j)} - D_{E1(i,j)}} \right)^{q(i,j)} \right)^{1/q(i,j)} - My_1(i,j) \quad \forall (i,j) \in E_{outer} \quad (5.73)$$

$$d_{ij} = 0 * z_1(i,j) + D_{E1(i,j)} * z_2(i,j) + D_{tan(i,j)} * z_3(i,j) + D_{E2(i,j)} * z_4(i,j) \quad \forall (i,j) \in E_{both} \quad (5.74)$$

$$z_{1(i,j)} + z_{2(i,j)} + z_{3(i,j)} + z_{4(i,j)} = 1 \quad \forall (i,j) \in E_{both} \quad (5.75)$$

$$z_{1(i,j)} \leq y_{1(i,j)} \quad \forall (i,j) \in E_{both} \quad (5.76)$$

$$z_{2(i,j)} \leq y_{2(i,j)} \quad \forall (i,j) \in E_{both} \quad (5.77)$$

$$z_{3(i,j)} \leq y_{2(i,j)} + y_{3(i,j)} \quad \forall (i,j) \in E_{both} \quad (5.78)$$

$$z_{4(i,j)} \leq y_{3(i,j)} \quad \forall (i,j) \in E_{both} \quad (5.79)$$

$$y_{1(i,j)} + y_{2(i,j)} + y_{3(i,j)} = 1 \quad \forall (i,j) \in E_{both} \quad (5.80)$$

$$r_{ij} \leq RDT_{E1(i,j)}(1 - y_{1(i,j)}) \quad \forall (i,j) \in E_{both} \quad (5.81)$$

$$r_{ij} \geq m_{ij}d_{ij} + n_{ij} - M(1 - y_{2(i,j)}) \quad \forall (i,j) \in E_{both} \quad (5.82)$$

$$r_{ij} \geq RDT_{tan(i,j)} - RDT_{tan(i,j)} \left(1 - \left(1 - \frac{d_{ij} - D_{tan(i,j)}}{D_{E2(i,j)} - D_{tan(i,j)}} \right)^{q(i,j)} \right)^{1/q(i,j)} - M(1 - y_{3(i,j)}) \quad \forall (i,j) \in E_{both} \quad (5.83)$$

$$\sum_{j \in N, j < i} y_{1(j,i)} + \sum_{j \in N, j > i} y_{1(i,j)} = |N| - 3 \quad \forall i \in N \quad (5.84)$$

$$\sum_{i \in S} \sum_{j \in N-S} y_{1(i,j)} \leq |S||N-S| - 2 \quad |S| = 3, \dots, \lfloor N/2 \rfloor \quad (5.85)$$

$$0 \leq z_{t(i,j)} \leq 1 \quad (i,j) \in E_{both}, t = 1, \dots, 4 \quad (5.86)$$

$$d_{ij}, r_{ij} \geq 0 \quad \forall (i,j) \in E \quad (5.87)$$

$$y_{1(i,j)} \in \{0,1\} \quad \forall (i,j) \in E_{outer}, E_{safe} \quad (5.88)$$

$$y_{t(i,j)} \in \{0,1\} \quad \forall (i,j) \in E_{both}, t = 1, \dots, 4 \quad (5.89)$$

We minimize total distance traveled and total radar detection threat measure in (5.66) and (5.67), respectively. The variables $y_{1(i,j)}$ determine whether we use any connection between nodes $(i,j) \in E$ or not. If $y_{1(i,j)}$ gets the value 1, we do not use any connection between nodes (i,j) ; if $y_{1(i,j)}$ gets the value 0, we use a connection between nodes (i,j) .

For the node pairs that are connected with a single efficient arc $((i,j) \in E_{safe})$, we either use the efficient arc between them (if $y_{1(i,j)} = 0$) or set that arc's objective

values to 0 (if $y_{1(i,j)} = 1$). The r_{ij} for $(i,j) \in E_{safe}$ will always be zero, however we wrote (5.69) for the sake of completeness. If $y_{1(i,j)} = 1$, the d_{ij} and r_{ij} get the value 0. If $y_{1(i,j)} = 0$, the d_{ij} and r_{ij} get the values $D_{E1(i,j)}$ and $RDT_{E1(i,j)}$ with (5.68) and (5.69), respectively.

For the node pairs whose efficient arcs only move in the outer radar region, we have two parts in their efficient frontier: either none of its connections are used ($y_{1(i,j)} = 1$) or a connection between $(i,j) \in E_{outer}$ is used ($y_{1(i,j)} = 0$). If an arc is used, its distance value should be between $[D_{E1(i,j)}, D_{E2(i,j)}]$ and its radar detection threat value should be less than $RDT_{E1(i,j)}$. We assure these by constraints (5.70), (5.71) and (5.72). The arc used should satisfy the L_q function for (i,j) . This relationship between d_{ij} and r_{ij} is given in (5.73).

For the node pairs whose efficient arcs can move in the inner and outer radar regions, their efficient frontiers are composed of three parts: either no arc is used ($y_{1(i,j)} = 1$), an arc moving in the inner radar region is used ($y_{2(i,j)} = 1$) or an arc moving in the outer radar region is used ($y_{3(i,j)} = 1$). With (5.74) and (5.75), we use the variables $z_{t(i,j)}$ to write the d_{ij} value as a convex combination of the points $(D_{E1(i,j)}, D_{tan(i,j)}, D_{E2(i,j)})$. If no connection between nodes (i,j) is used ($y_{1(i,j)} = 1$), the r_{ij} takes the value 0 by (5.81). The variable $z_{1(i,j)}$ equals 1 and all other $z_{t(i,j)}$ $t = 2,3,4$ equal 0 with equations (5.75) and (5.76). This makes d_{ij} be 0 by (5.74). If an arc that moves in the inner radar region is used ($y_{2(i,j)} = 1$), r_{ij} is bounded by $RDT_{E1(i,j)}$ in (5.81). The variables $z_{1(i,j)}$ and $z_{4(i,j)}$ equal 0 and $z_{2(i,j)}, z_{3(i,j)} \geq 0$ with equations (5.75) and (5.76). Therefore, we write d_{ij} as a convex combination of $D_{E1(i,j)}$ and $D_{tan(i,j)}$. The corresponding r_{ij} is found by (5.82). If an arc that moves in the outer radar region is used ($y_{3(i,j)} = 1$), the variables $z_{1(i,j)}$ and $z_{2(i,j)}$ equal 0 and $z_{3(i,j)}, z_{4(i,j)} \geq 0$ with equations (5.75) and (5.76). Therefore, we write d_{ij} as a convex

combination of $D_{tan(i,j)}$ and $D_{E2(i,j)}$. The corresponding r_{ij} that satisfies the L_q function is found by (5.83).

With (5.84), we ensure that there is exactly one incoming arc and one outgoing arc to each node. The left hand side of that equation is the total number of unconnected nodes from node i . There should be exactly two nodes that are connected to each node. Therefore, the total number of unconnected nodes from any node i should equal the total number of nodes that node i can be connected minus 2 (the node number that each node should be connected to). With (5.85), we eliminate subtours. The left hand side of the constraint calculates the number of unused node connections from a subset S . The total number of unused node connections from a subset S is $|S||N - S|$. The two subsets S and $N - S$ should be connected by at least two connections. Therefore, the unused node connections between subsets S and $N - S$ should be at most $|S||N - S| - 2$.

Similar to finding an efficient solution for the bi-objective shortest path problem with a continuous frontier, we consider the first objective as a constraint and minimize the second objective. We set an upper bound on the distance value as follows:

$$\sum_{(i,j) \in E} d_{ij} \leq D_{UB} \quad (5.90)$$

We also increment the second objective by multiplying the total distance with a very small positive constant (ρ) to avoid weakly efficient but inefficient solutions.

$$\text{Min } RDT = \sum_{(i,j) \in E} r_{ij} + \rho \sum_{(i,j) \in E} d_{ij} \quad (5.91)$$

We use the BARON solver in the GAMS Optimization Package. It is developed to solve nonlinear and mixed-integer nonlinear programs. For more details on this solver, please see <http://www.gams.com/dd/docs/solvers/baron.pdf>.

The efficient frontier of the bi-objective routing problem is made up of infinitely many efficient solutions. To find an approximate representation of the efficient frontier, we

may discretize it and find a representative set of efficient solutions. However, these representative efficient tours are approximate since they are formed from approximated efficient frontiers of arcs, and there may not exist true arcs with exactly the assumed objective values. For each consecutive target on an efficient tour, we use an arc with objective values that satisfy the equation approximating that target pair's efficient frontier. To find a set of real objective values for each consecutive target pair's arc that are close to the assumed objective values, we use the heuristic we develop for finding the minimum *RDT* value corresponding to a *D*. This way, we generate a new set of true objective values where each representative approximate efficient tour is replaced with a real tour.

5.11 Comparison of Solutions for Discretized and Continuous Terrain

We generate a UAV routing problem with five targets and four radars on a 400 km² terrain. The placement of the targets and radars can be seen in Figure 5.15. The targets are located at (3, 17), (6,1), (10, 5), (15, 3) and (16, 14). The radars are located at (5, 12), (9, 5), (12, 16) and (16, 8). Our developments for the continuous terrain are based on nonoverlapping radar regions and a single radar region affecting each target pair. Therefore, we created a problem situation that conforms with these assumptions.

We solve the routing problem for both discretized and continuous terrain representations. We divide the terrain into 20 by 20 kms for the discretized representation. As the UAV has more flexibility of movement in the continuous terrain case, we expect to obtain better results in both objectives.

The efficient set of the arcs between target pairs in the discretized terrain are generally dominated by those in the continuous terrain, as expected. An example efficient frontier can be seen for the arcs between targets 4 and 5 in Figure 5.16.

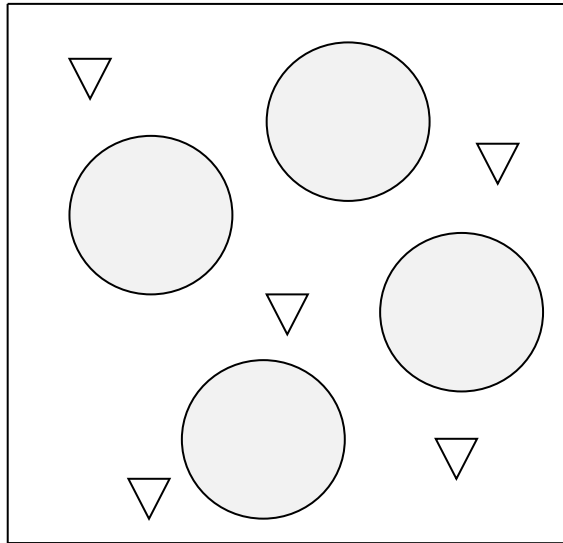


Figure 5.15 An Example UAV Routing Problem

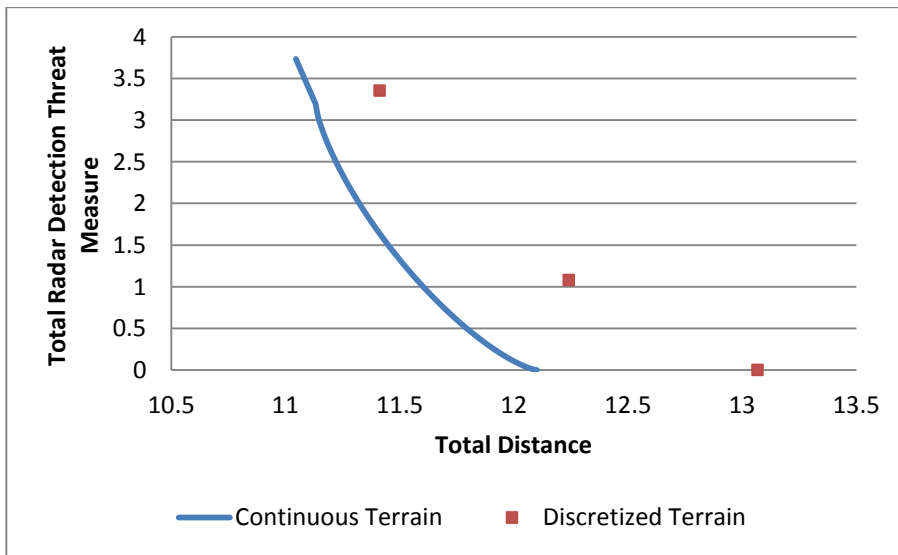


Figure 5.16 Efficient Frontier for the Arcs Between Targets 4 and 5

We also generate the efficient frontier of the bi-objective TSP for the two terrain types. For the continuous terrain, we approximate the frontier with 10 representative efficient

points. Then we find the real tours corresponding to the representative tours. They can be seen in Figure 5.17. The efficient frontier of the continuous terrain representation dominates the efficient set of discretized terrain representation. The order of visit to the targets of the efficient tours turned out to be the same for both terrains.

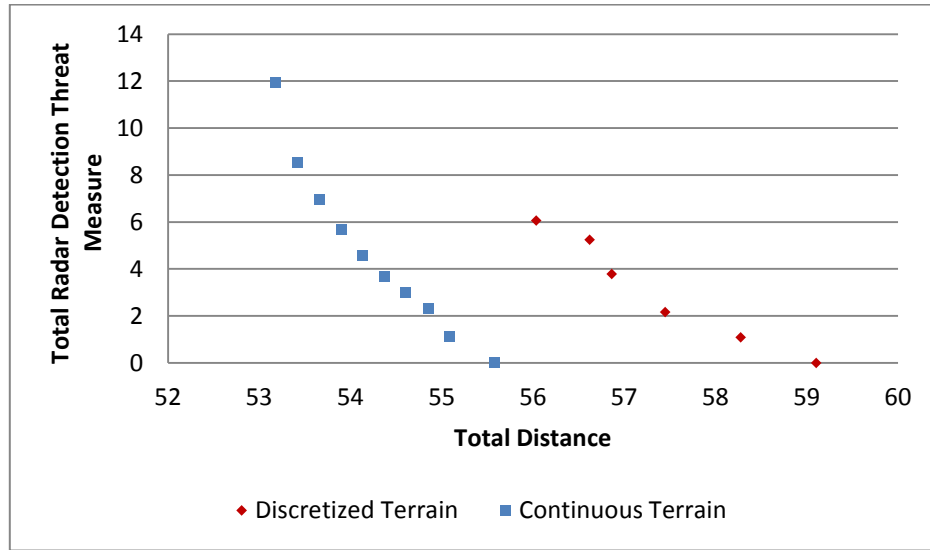


Figure 5.17 Efficient Frontier for the Tours

5.12 Integrating the Bi-Objective Routing Problem in Continuous Terrain to the Interactive Algorithm

The bi-objective routing problem in the continuous terrain can be solved with the interactive algorithm we develop in Chapter 3. However, there would be infinitely many efficient solutions and we may need to find a large number of them to arrive at the best solution of the DM. A computationally efficient method would be to discretize the efficient frontier and find a representative set of efficient solutions to be presented to the DM. For this, the efficient frontier can be divided into k intervals between two extreme efficient solutions. Let the left extreme solution of the MOTSP have distance

and radar values D_{LE} and RDT_{LE} , respectively. Similarly, let the right extreme solution be (D_{RE}, RDT_{RE}) . The interval $[D_{LE}, D_{RE}]$ can be divided into k intervals, each having length, $\Delta = \frac{D_{RE} - D_{LE}}{k}$. In total, we would have $k+1$ efficient solutions that represent the efficient frontier. The first representative solution has distance D_{LE} , the second representative solution would have distance $D_{LE} + \Delta$, and in general terms, the a^{th} representative solution would have distance $D_{LE} + (a - 1)\Delta$.

Suppose we have the efficient frontier shown in Figure 5.15. Initially, we need to find the extreme efficient solutions. To find the left extreme solution (smallest distance and largest RDT), we solve the single objective TSP using the Concorde package. We prepare the edge costs (e_{ij}) as input to Concorde by setting the efficient solution with the smallest distance as their edge costs as:

$$e_{ij} = \begin{cases} (w_{LE})D_{E1(i,j)} + (1 - w_{LE})RDT_{E1(i,j)} & (i, j) \in E_{safe} \\ (w_{LE})D_{E1(i,j)} + (1 - w_{LE})RDT_{E1(i,j)} & (i, j) \in (E_{both} \cup E_{outer}) \end{cases}$$

We then minimize the sum of arc costs using Concorde.

Similarly, to find the right extreme solution, we prepare the edge costs as follows:

$$e_{ij} = \begin{cases} (w_{RE})D_{E1(i,j)} + (1 - w_{RE})RDT_{E1(i,j)} & (i, j) \in E_{safe} \\ (w_{RE})D_{E2(i,j)} + (1 - w_{RE})RDT_{E2(i,j)} & (i, j) \in (E_{both} \cup E_{outer}) \end{cases}$$

Then, we decide on the interval length, Δ . This length could be kept large initially and decreased when more precision is required. In the example in Figure 5.18 we divide the frontier into 10 equally spaced intervals. Here, the objective values of each representative solution should be found. Since the distance value of the a^{th} representative solution is approximately $D_{LE} + (a - 1)\Delta$, the radar detection threat value of this solution can be found by minimizing (5.91) subject to constraints (5.68)-(5.90). When we solve this model, we may come up with a solution that has a distance value less than $D_{LE} + (a - 1)\Delta$ if the efficient frontier is not continuous at some points. However, the resulting solution could still be used as a representative solution since it is also an efficient solution for the MOTSP.

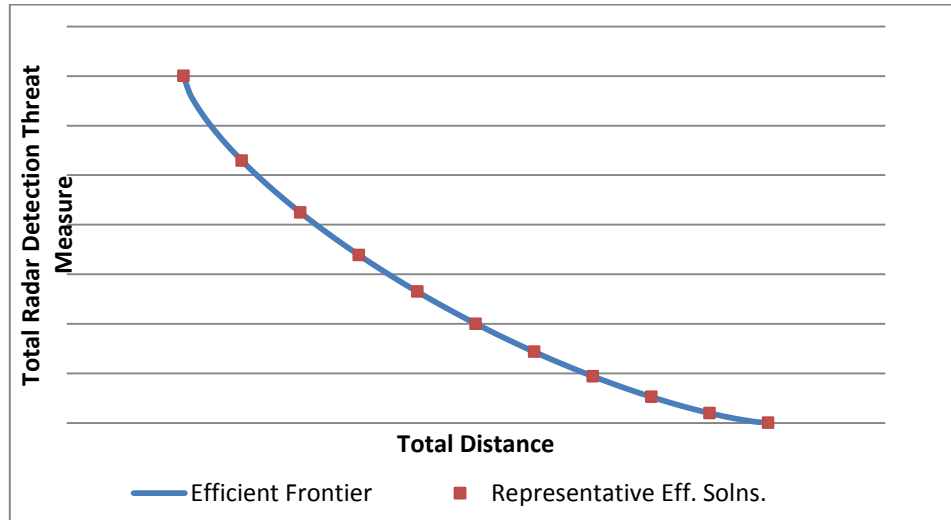


Figure 5.18 An Example Efficient Frontier for the MOTSP

Knowing the objective function values of a number of solutions, we can reduce the most preferred region around the representative solution that is preferred to all its adjacent efficient solutions. Here, actually we are not sure whether the two adjacent representative solutions are also adjacent efficient solutions. Therefore, we reduce the most preferred areas to rectangles (not triangles) between the most preferred representative point and its adjacent representative points. This part is shown in Figure 5.19.

We may want to reduce the most preferred region of the DM further. This time, the most preferred region can be divided into smaller intervals and representative points inside that region can be found. Using the new representative points, a new region around the most preferred representative and its adjacent representative points can be defined. This process can continue until a desired precision on the most preferred region is obtained.

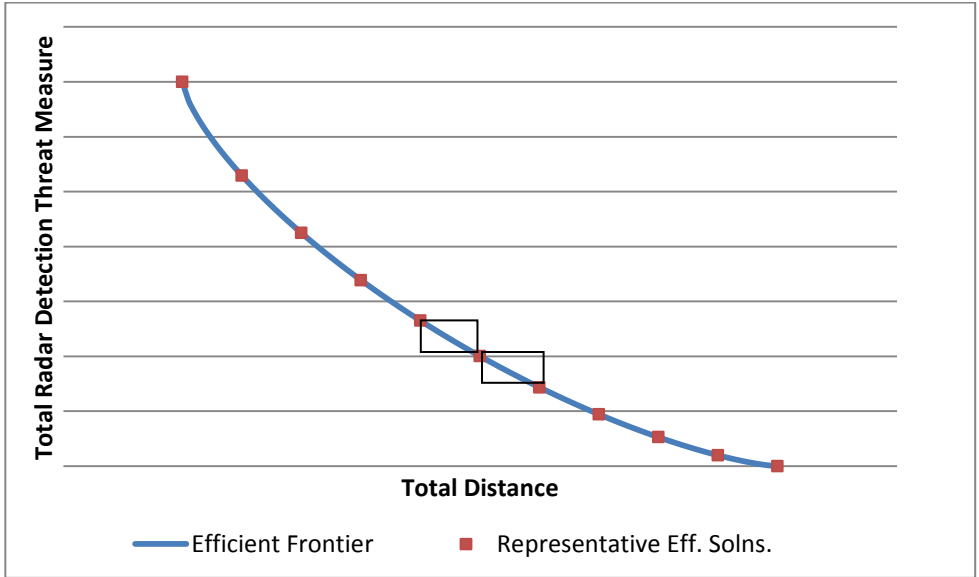


Figure 5.19 Most preferred region of the DM

CHAPTER 6

CONCLUSIONS

In this thesis, we study the bi-objective routing problem. We consider this problem as a MOTSP with multiple efficient arcs between nodes. We develop solution procedures to generate efficient solutions. We define rules that connect the two combinatorial problems that constitute the bi-objective routing problem: MOTSP and MOSPP. We consider the routing problem on two terrain structures: discretized terrain and continuous terrain.

We first develop a general interactive algorithm that finds the most preferred solution of a DM whose preferences are consistent with a quasiconvex preference function. This algorithm is applicable for any bi-objective integer program. We apply the algorithm to the bi-objective routing problem. To find supported efficient tours, we use the Concorde software. To find unsupported efficient tours inside a region defined by constraints, we need to provide all efficient arcs as input. Using the constraints on the objective space, we develop rules that connect the two subproblems TSP and SPP. These rules helped reduce the required number of efficient arcs to be used in the efficient tours. Using the efficient tours found so far in a region, we improved the upper bound on the combined objectives defining that region which was initially suggested by Tuytens et al. (2000). We demonstrate our solution approach on the bi-objective routing problem for UAVs. In this problem, we consider a two dimensional terrain that is discretized with equidistant grid points. We also solved randomly generated instances with varying number of efficient arcs.

To the best of our knowledge, this is the first study that solves the bi-objective TSP considering all multiple efficient arcs between node pairs and selecting the ones that suit the DM's best tour. We use the constraints in the objective space of the bi-objective TSP to reduce the efficient arcs between node pairs. We develop properties to improve the efficiency of the algorithm. We find a number of lower bound approximations to reduce the arcs of each node pair, which becomes computationally hard as the number of nodes and the number of constraints increase. We develop some rules to decrease the required lower bound approximations. Also, we avoid searching a solution that has already been found by keeping track of all efficient solutions and modifying the procedure that finds adjacent efficient solutions.

We also study the routing problem in continuous terrain for the UAV routing problem. The objectives of this problem; minimization of distance traveled and minimization of radar detection threat; are nonlinear in terms of the decision variables. This nonlinearity makes the solution process computationally harder. We develop some properties for the efficient arcs between target pairs when there exists a single radar between them. We use these properties in the heuristic and exact approaches we develop to find efficient arcs. Knowing a number of efficient arcs, we generate the efficient frontier of the bi-objective SPP. We then solve the bi-objective TSP using a mixed integer nonlinear program. We also discuss the implementation of the interactive algorithm to the bi-objective routing problem in continuous terrain. To the best of our knowledge, this is the first study that considers the measure of radar detection threat developed by Gudaitis (1994) in continuous terrain.

In this thesis, we considered a static setting for the routing problem. However, some of the environmental parameters may be uncertain. For example, the weather conditions and hence the parameter values of different arcs may change during the flights of air vehicles, the travel times may vary due to traffic congestions for land transport, some parts of a tour may get blocked and unconnected in road networks, etc. One way to deal with uncertainties is to implement solutions that are less sensitive to changes in the environment. These solutions can be considered as robust solutions. Although these

solutions may not be “best” for the problem where the environment is known with certainty, they may be preferable because they are robust to changes in the environment. For example, a dynamic structure could be implemented in the routing problem where the positions of the nodes could vary dynamically, without certainty. Another example could be given for the UAV routing problem where the radar sites/targets change their locations. For this case, it would be useful to develop approaches that account for this dynamic nature.

Our approaches can also be adapted for different situations. For example, the routing problem can be generalized to multiple vehicles, each vehicle visiting a subset of nodes. For the UAV routing problem, in continuous terrain case, we currently assume that at most a single radar is located between two targets. If there are more radars, the problem becomes harder with more decision variables. Additionally we need to find the entrance and exit points to all radar regions and the movement characteristics inside those regions. Another issue would be considering regions where the radar sites overlap. For those regions, the detection probability should be larger than the summation of all the detection probabilities from each radar region. We can use a different formulation to amplify the cumulative effect of the radars. This problem would be more realistic though, and therefore is meaningful as a future study. Also, different real life properties can be incorporated into the UAV routing problem. The terrain can be generalized to three dimensions and obstacles on the terrain can be incorporated into the model. For the discretized terrain, incorporating the obstacles would be easier than it would be for the continuous terrain. For the discretized terrain, we may simply forbid movement to the grid points at which the obstacle is located. Although this might lead to missing some moves that are feasible, approximate efficient arcs could be found. For the continuous terrain, the obstacles may disrupt the movement model inside the radar regions. The radar regions may become nonconvex. Therefore, we may not use the property that the middle points lie on a line and consequently we may not be able to use the heuristic we develop. A new approach that covers nonconvex radar regions would be needed. In our approach, we assume that all targets have the same priority. A different variation of this problem could be to allow

assigning priorities to targets for the order of visit or to introduce time windows for visiting each target.

In this study, we develop an interactive algorithm for bi-objective integer programs. In the presence of two objectives, we put constraints on the objective space to eliminate the regions where the DM's most preferred solution cannot lie. If we consider more objectives, the constraints we put in the objective space would be different. We may need additional binary variables for the solution. These would result in problems that are computationally harder to solve. As a future work, the interactive algorithm can be generalized for more than two objectives.

REFERENCES

- Adams R. A., (1999) *Calculus, A Complete Course*, Addison-Wesley.
- Bazaara M.S., H.D. Sherali and C.M. Shetty, (2006) *Nonlinear Programming: Theory and Algorithms*, Wiley, 3rd edition.
- Berube J.F., M. Gendreau and J.Y. Potvin, (2009) “An Exact ε -constraint Method for Bi-objective Combinatorial Optimization Problems – Application to the Traveling Salesman Problem with Profits,” *European Journal of Operational Research* 194 (1), 39-50.
- Chankong V. and Y.Y. Haimes, (1983) *Multiobjective Decision Making: Theory and Methodology*, North-Holland, New York.
- Cook W.J., W. H. Cunningham, W. R. Pulleyblank and A. Schrijver, (1998) *Combinatorial Optimization*, John Wiley & Sons, Inc., New York.
- Desrochers M. and G. Laporte, (1991) “Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints,” *Operations Research Letters* 10, 27-36
- Ehrgott M., (2000) Multi Objective Optimization, Chapter 7. *Combinatorial Problems with Multiple Objectives*, The Travelling Salesman Problem, 211-220, Wiley.
- 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.
- Gandibleux X., F. Beugnies and S. Randriamasy, (2006) “Martins’ Algorithm Revisited for Multi-objective Shortest Path Problems with a MaxMin Cost Function,” *A Quarterly Journal of Operations Research* 4 (1), 47-59.

Gudaitis M.S., (1994) "Multi criteria Mission Route Planning Using a Parallel A* Search," M.S. thesis, Air Force Institute of Technology.

Hansen M.P., (2000) "Use of Substitute Scalarizing Functions to Guide a Local Search Based Heuristic: The Case of moTSP," *Journal of Heuristics* 6, 419-413.

Jaszkiewicz A. and P. Zielniewicz, (2009) "Pareto Memetic Algorithm with Path Relinking for Bi-objective Traveling Salesperson Problem," *European Journal of Operational Research* 193 (3), 885-890.

Jia D. and J. Vagners, (2004) "Parallel Evolutionary Algorithm for UAV Path Planning," AIAA 1st Intelligent Systems Technical Conference, Chicago Illinois.

Jozefowicz N., F. Glover, and M. Laguna, (2008) "Multi-objective Meta-heuristics for the Traveling Salesman Problem with Profits," *Journal of Mathematical Modelling and Algorithms* 7 (2), 177-195.

Karademir S., (2008) "A Genetic Algorithm for the Biobjective Traveling Salesman Problem with Profits," M.S. thesis, Department of Industrial Engineering, Middle East Technical University.

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., J. Wallenius and S. Zionts, (1984) "Solving the Discrete Multiple Criteria Problem Using Convex Cones," *Management Science* 30 (1), 1336-1345.

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

Köksalan, M. and Lokman, B., (2009) "Approximating the Nondominated Frontiers of Multi-objective Combinatorial Optimization Problems," *Naval Research Logistics* 56, 191-198.

Köksalan, M. and Soylu, B., (2010) “Bicriteria p-hub Location Problem and Evolutionary Algorithms”, *INFORMS Journal on Computing* 22 (4), 1-15.

Laporte G., (1992) “The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms,” *European Journal of Operational Research* 59 (2), 231-247.

Lokman B., M. Köksalan, P. J. Korhonen and J. Wallenius, (2011) “Solving Multi-objective Integer Programs Using Convex Preference Cones,” Technical Report, Department of Industrial Engineering, Middle East Technical University, 2011:02.

Lust T. and J. Teghem, (2010) “The Multiobjective Traveling Salesman Problem: A Survey and a New Approach,” *Advances in Multi-Objective Nature Inspired Computing Studies in Computational Intelligence* 272, 119-141.

Miller C.E., A.W. Tucker and R.A. Zemlin, (1960) “Integer Programming Formulation of Traveling Salesman Problems,” *Journal of the Association for Computing Machinery* 7 (4), 326-329.

Olsan J.B., (1993) “Genetic Algorithms Applied to a Mission Routing Problem,” M.S. thesis, Air Force Institute of Technology.

Özpeynirci, Ö. and M. Köksalan, (2009) “Multiobjective Traveling Salesperson Problem on Halin Graphs,” *European Journal of Operational Research* 196, 155-161.

Özpeynirci, Ö. and M. Köksalan, (2010) “Pyramidal Tours and Multiple Objectives,” *Journal of Global Optimization* 48 (4), 569-582.

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

Paquete L. and T. Stützle, (2003) “A Two-Phase Local Search for the Biobjective Traveling Salesman Problem,” Proc. of the 2nd International Conference on Evolutionary Multi-criterion Optimization (EMO 2003), LNCS 2632, 479-493, Springer.

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

Przybylski A., X. Gandibleux and M. Ehrgott, (2008) "Two Phase Algorithms for the Bi-Objective Assignment Problem," *European Journal of Operations Research* 185, 509-533.

Raith A. and M. Ehrgott, (2009) "A Comparison of Solution Strategies for Biobjective Shortest Path Problems," *Computers and Operations Research* 36 (4), 1299-1331.

Ramesh, R., M.H. Karwan and S. Zionts, (1990) "An Interactive Method for Bicriteria Integer Programming." *IEEE Transactions on Systems, Man, and Cybernetics* 20, 395-403.

Skolnik M., (1980) Introduction to Radar Systems. Chapter 2. The Radar Equation, 15-67, McGraw-Hill.

Skriver A.J.V and K.A. Andersen, (2000) "A Label Correcting Approach for Solving Bicriterion Shortest Path Problems" *Computers and Operations Research* 27, 507-524.

Tezcaner D., (2009) "Multi-Objective Route Selection," M.S. thesis, Department of Industrial Engineering, Middle East Technical University.

Tezcaner D. and M. Köksalan, (2011) "An Interactive Algorithm for Multi-Objective Route Planning," *Journal of Optimization Theory and Applications* 150 (2), 379-394.

Tuytens D., J. Teghem, P. Fortemps and K. Van Nieuwenhuyse, (2000) "Performance of the MOSA Method for the Bicriteria Assignment Problem," *Journal of Heuristics* 6, 295-310.

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

Zheng, C., L. Li, F. Xu and F. Sun, (2005) "Evolutionary Route Planner for Unmanned Air Vehicles." *IEEE Transactions on Robotics*, 21 (4), 609-620.

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.

APPENDIX A

MARTINS ALGORITHM

Martin's algorithm is an extension of Dijkstra's algorithm (Cook et al., 1998 p. 31) to the multi-objective case. Dijkstra's algorithm is an exact algorithm that is developed for the single objective SPP. In this algorithm, all the nodes have either permanent or temporary labels. If a node's label is set as permanent, it means that the shortest path from the initial node to that node is found. In each iteration, the label with the minimum value is set as permanent. The labels of all successors of that node are recalculated. The algorithm terminates when all the node labels are set as permanent. In Martin's algorithm, there may be multiple labels to the nodes, since there may be many efficient arcs from the initial node to any intermediate node. In each iteration, one label (the label that has the smallest value in the first objective) is set as permanent. From this node, all its successors' labels are recalculated. There is a dominance check for each newly created arc to a node: if the arcs corresponding to the previous labels are better than the newly created arc, in terms of all objectives, this new arc is dominated. Another possibility is that the arc corresponding to a new label dominates arcs of a number of previously created labels. If neither of these cases occur, the label is put in the label list of that node. The algorithm terminates when all of the labels are set as permanent. This algorithm is developed for the sum-type objectives (where the objective value is the sum of the values of the components that are in the solution). Gandibleux et al. (2006) extend Martin's algorithm to include one max-min function (where the corresponding objective's is equal to the value of the minimum of its components), additionally.

APPENDIX B

COMPUTATIONAL RESULTS FOR THE INTERACTIVE ALGORITHM ON BI-OBJECTIVE ROUTING PROBLEM

We consider UAV routing problem with five and six targets. For the five target problem, we solve the routing problem in the terrain of Figure 4.8. For the six target problem, we randomly generate the two objective function values (total distance and total radar detection threat) for the efficient arcs between target pairs. We consider three types of problems:

- 6-target MOTSP with 3 efficient arcs between target pairs
- 6-target MOTSP with at most 10 efficient arcs between target pairs
- 6-target MOTSP with 15 efficient arcs between target pairs

We solve all the problems assuming that the DM has an underlying Tchebycheff preference function $U(z) = \max[\lambda_1(z_1 - z_1^*), (1 - \lambda_1)(z_2 - z_2^*)]$ to be minimized, which we pretend is unknown to us at the beginning of the algorithm. The ideal points z_1^* and z_2^* are taken as 0. We solve the constrained TSP directly and use tours for lower bounds.

The efficient frontier of the first problem (5-target MOTSP) is given in Figure 4.10. The results of the algorithm for three different λ_1 values are summarized in Table B.1. For the first problem with six targets (6-target MOTSP with 3 efficient arcs between target pairs), we generate 3 efficient arcs between all node pairs. We generate the

efficient frontier of this problem by enumeration. It is composed of 60 efficient solutions and 13 efficient tours. The results of the algorithm for three different λ_1 values are summarized in Table B.2. For the second problem with six targets (6-target MOTSP with at most 10 efficient arcs between target pairs), we generate 10 efficient arcs between three node pairs, 5 efficient arcs between three target pairs, one efficient arc between two target pairs and 3 efficient arcs between the remaining target pairs. Its efficient frontier is composed of 75 efficient solutions and 11 efficient tours. The results of the algorithm for three different λ_1 values are summarized in Table B.3. The results of the last problem (6-target MOTSP with 15 efficient arcs between target pairs) for three different λ_1 values are summarized in Table B.4. We could not generate the efficient frontier of this problem due to its size.

Table B.1 Results of the algorithm for 5-target MOTSP

Weight of first objective (λ_1)	Number of efficient tours found	Number of comparisons	Number of arcs reduced by bounds on 1 st obj. / 2 nd obj. / UB on combined obj.
0.15	13	9	39 / 8 / 2
0.25	13	9	37 / 1 / 3
0.45	9	4	37 / 0 / 10

Table B.2 Results of the algorithm for 6-target MOTSP with 3 efficient arcs

Weight of first objective (λ_1)	Number of efficient tours found	Number of comparisons	Number of arcs reduced by bounds on 1 st obj. / 2 nd obj. / UB on combined obj.
0.25	23	13	0 / 17 / 38
0.50	23	15	0 / 0 / 43
0.80	10	9	39 / 0 / 15

Table B.3 Results of the algorithm for 6-target MOTSP with at most 10 efficient arcs

Weight of first objective (λ_1)	Number of efficient tours found	Number of comparisons	Number of arcs reduced by bounds on 1 st obj. / 2 nd obj. / UB on combined obj.
0.25	18	14	0 / 14 / 66
0.45	17	11	0 / 0 / 51
0.75	12	8	16 / 0 / 43

Table B.4 Results of the algorithm for 6-target MOTSP with 15 efficient arcs

Weight of first objective (λ_1)	Number of efficient tours found	Number of comparisons	Number of arcs reduced by bounds on 1 st obj. / 2 nd obj. / UB on combined obj.
0.25	29	15	0 / 0 / 80
0.45	45	31	0 / 0 / 262
0.75	26	19	0 / 0 / 98

In each problem instance, for some λ_1 values, some bounds became redundant and did not result in arc reductions. However, the arc reductions are generally effective. The bounds, except for the upper bounds on the combined objective, became ineffective in the largest problem instance (Table B.4). Although we could not generate the whole efficient set for this problem, we believe that this is due to the efficient set approaching a continuous frontier.

APPENDIX C

REGRESSION ANALYSIS FOR x_m vs. y_m

We evaluate the regression analysis results for x_m versus y_m . The normal probability plot and histogram in Figure C.1, show that the error terms are normally distributed. We test homoscedasticity with “Versus Fits” graph. The variance of the residuals seems to increase with increasing fitted values. We can say that the constant variance assumption may be violated for the residuals. The error terms do not follow a pattern in the “Versus Order” graph. They are almost randomly distributed around 0. We can conclude that the error terms are independent.

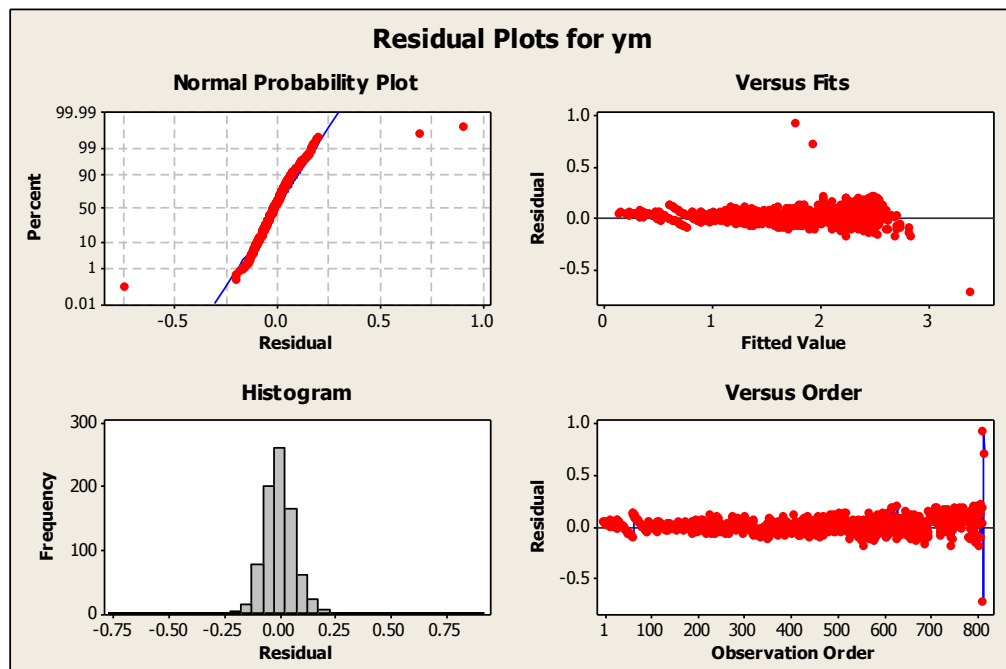


Figure C.1 Residual Plots for y_m

The results of the regression can be seen in Figure C.2.

The regression equation is
 $ym = - 0.0490 - 5.19 xm$

Predictor	Coef	SE Coef	T	P
Constant	-0.048980	0.008399	-5.83	0.000
xm	-5.19320	0.02389	-217.41	0.000

S = 0.0807021 R-Sq = 98.3% R-Sq(adj) = 98.3%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	1	307.83	307.83	47265.45	0.000
Residual Error	814	5.30	0.01		
Total	815	313.13			

Unusual Observations

Obs	xm	ym	Fit	SE Fit	Residual	St Resid
1	-0.036	0.16907	0.13849	0.00759	0.03058	0.38 X
2	-0.041	0.20500	0.16446	0.00748	0.04054	0.50 X
3	-0.051	0.23953	0.21639	0.00726	0.02314	0.29 X
4	-0.056	0.25569	0.24236	0.00715	0.01333	0.17 X
5	-0.056	0.25569	0.24236	0.00715	0.01333	0.17 X
6	-0.056	0.28473	0.24236	0.00715	0.04237	0.53 X
7	-0.066	0.31642	0.29429	0.00693	0.02213	0.28 X
8	-0.066	0.31642	0.29429	0.00693	0.02213	0.28 X
557	-0.442	2.05560	2.24745	0.00388	-0.19185	-2.38R
619	-0.402	2.20420	2.03972	0.00330	0.16448	2.04R
622	-0.402	2.21295	2.03972	0.00330	0.17323	2.15R
628	-0.402	2.23015	2.03972	0.00330	0.19043	2.36R
689	-0.497	2.35410	2.53308	0.00487	-0.17898	-2.22R
714	-0.442	2.42175	2.24745	0.00388	0.17430	2.16R
716	-0.442	2.42835	2.24745	0.00388	0.18090	2.24R
747	-0.527	2.49515	2.68880	0.00547	-0.19365	-2.41R
753	-0.462	2.53170	2.35124	0.00422	0.18046	2.24R
755	-0.467	2.53875	2.37720	0.00431	0.16155	2.00R
766	-0.467	2.54950	2.37720	0.00431	0.17230	2.14R
771	-0.472	2.57180	2.40317	0.00440	0.16863	2.09R
793	-0.482	2.64040	2.45510	0.00458	0.18530	2.30R
794	-0.487	2.64585	2.48107	0.00468	0.16478	2.05R
803	-0.492	2.70225	2.50703	0.00477	0.19522	2.42R
806	-0.557	2.65080	2.84459	0.00609	-0.19379	-2.41R
808	-0.487	2.68645	2.48107	0.00468	0.20538	2.55R
810	-0.497	2.70665	2.53300	0.00487	0.17365	2.16R
812	-0.502	2.72795	2.55897	0.00497	0.16898	2.10R
814	-0.662	2.64920	3.38988	0.00840	-0.74068	-9.23RX
815	-0.352	2.69315	1.77999	0.00287	0.91316	11.32R
816	-0.382	2.63775	1.93578	0.00308	0.70197	8.70R

R denotes an observation with a large standardized residual.
X denotes an observation whose X value gives it large leverage.

Figure C.2 Results of Regression Analysis for Middle Points of Entrance and Exit Points

We have 30 unusual observations among 816 data points which is almost 3.68%. 22 observations have a large standardized residual value which makes 2.70% of the data. Normally, we expect at most 4.55% of the data to have standardized values less than -2 or greater than 2. Our data satisfies this condition.

We also check how total distance traveled (D) changes with respect to other variables (y_{en}, y_{ex}, r, y_m). The scatter plots can be seen in Figure C.3. As the entrance and exit points increase (leading to increase in their middle points), the distance of the arc increases. This is due to the fact that the arc of the second extreme solution has higher entrance and exit points to the radar region compared to the first extreme solution. The increase in the entrance and exit points makes the solutions move closer to the second extreme solution; for which the distance of the arcs increase and their radar detection threats decrease. As we move to the second extreme solution from the first extreme solution, the curvature of the arc inside the radar region increases. The first extreme solution is straight, while the second extreme solution has the highest curvature with $r = 2.9108$. The first extreme solution's arc can also be considered as a part of a circle that has $r \rightarrow \infty$.

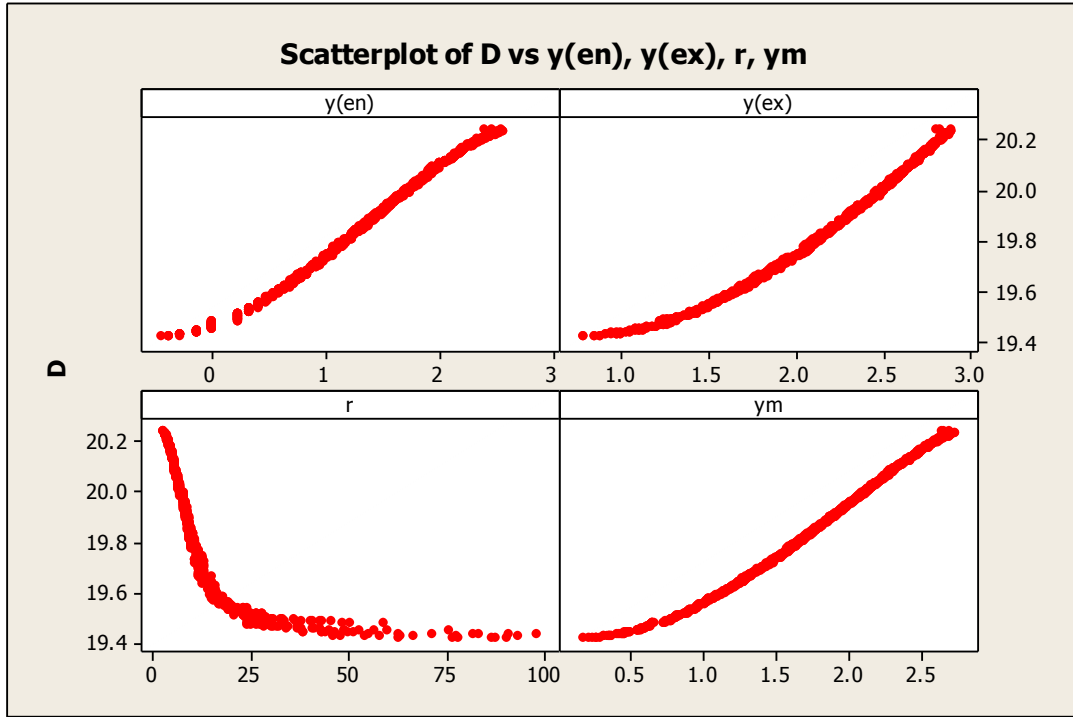


Figure C.3 Scatter Plot of (D) vs. (y_{en}, y_{ex}, r, y_m)

APPENDIX D

METHODS TO APPROXIMATE INTEGRALS

For all methods, the range $[a, b]$ is divided into n equal length intervals for the length $l = \frac{b-a}{n}$ and we assume that $f(x)$ is continuous on $[a, b]$ and the interval $[a, b]$ is finite.

Trapezoid Rule

For n equal length intervals, we have $n + 1$ points in the range $[a, b]$. Let $x_0 = a, x_1 = a + \frac{b-a}{n}, \dots, x_k = a + k \frac{b-a}{n}, \dots, x_n = b$ and $y_k = f(x_k)$, we have the following approximation.

$$\int_a^b f(x)dx \approx \frac{l}{2}(y_0 + 2 \sum_{j=1}^{n-1} y_j + y_n) \quad (\text{D.1})$$

This rule combines the consecutive points $y_k, k = 0, \dots, n$ with straight lines and makes n trapezoids. It then calculates the sum of the areas of these trapezoids.

Midpoint Rule

Let $m_1 = a + \left(\frac{1}{2}\right)\frac{b-a}{n}, m_2 = a + \left(\frac{3}{2}\right)\frac{b-a}{n} \dots, m_k = a + \left(k - \frac{1}{2}\right)\frac{b-a}{n}, \dots, m_n = a + \left(n - \frac{1}{2}\right)\frac{b-a}{n}$ and $y_k = f(x_k)$, we have the following approximation.

$$\int_a^b f(x)dx \approx l \sum_{j=1}^n f(m_j) \quad (\text{D.2})$$

This rule again combines the consecutive points $y_k, k = 0, \dots, n$ with straight lines and makes n trapezoids. The sides of the trapezoids are drawn from the middle points m_k of the consecutive points x_k . The integral is approximated as the sum of these trapezoid areas.

Simpson's Rule

For this rule, we should have an even number of subintervals, n . Let $x_0 = a, x_1 = a + \frac{b-a}{n}, \dots, x_k = a + k \frac{b-a}{n}, \dots, x_n = b$ and $y_k = f(x_k)$, we have the following approximation.

$$\int_a^b f(x)dx \approx \frac{l}{3} (y_0 + y_n + 4y_{\text{"odds"}} + 2y_{\text{"evens"}}) \quad (\text{D.3})$$

This rule combines the consecutive points $y_k, k = 0, \dots, n$ with parabolas. The integral is approximated as the sum of these parabolic regions' areas.

APPENDIX E

FIBONACCI SEARCH METHOD

Fibonacci Search Method is used for finding the global minimum (x^*) of a strictly quasiconvex function f over $S = [a, b]$. We summarize the steps of this method (Bazaraa et al., 2006). For this method, $F_0 = F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.

Let l be the length of uncertainty and d be the distinguishability constant. Number of observations is set to the minimum n such that $F_n > \frac{b-a}{l}$. Also, let $\lambda_1 = a_1 + \left(\frac{F_{n-2}}{F_n}\right)(b_1 - a_1)$ and $\mu_1 = a_1 + \left(\frac{F_{n-1}}{F_n}\right)(b_1 - a_1)$. Start the iteration counter $k = 1$.

Step F.1 If $f(\lambda_k) > f(\mu_k)$, go to Step F.2. Otherwise, go to Step F.3.

Step F.2 Set $a_{k+1} = \lambda_k$, $b_{k+1} = b_k$, $\lambda_{k+1} = \mu_k$, $\mu_{k+1} = a_{k+1} + \left(\frac{F_{n-k-1}}{F_{n-k}}\right)(b_{k+1} - a_{k+1})$. Increment $k \leftarrow k + 1$. If $k = n - 1$, go to Step F.4. Otherwise, go to Step F.1.

Step F.3 Set $a_{k+1} = a_k$, $b_{k+1} = \mu_k$, $\lambda_{k+1} = a_{k+1} + \left(\frac{F_{n-k-2}}{F_{n-k}}\right)(b_{k+1} - a_{k+1})$, $\mu_{k+1} = \lambda_k$. Increment $k \leftarrow k + 1$. If $k = n - 1$, go to Step F.4. Otherwise, go to Step F.1.

Step F.4 Set $\lambda_n = \lambda_{n-1}$, $\mu_n = \lambda_{n-1} + d$. If $f(\lambda_n) > f(\mu_n)$, set $a_n = \lambda_n$, $b_n = b_{n-1}$. Otherwise, set $a_n = a_{n-1}$, $b_n = \lambda_n$. Set $x^* = \frac{a_n + b_n}{2}$.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Tezcaner Öztürk, Diclehan
Nationality: Turkish (TC)
Date and Place of Birth: 9 May 1986, Ankara
Marital Status: Married
e-mail: diclehantezcaner@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU, Industrial Engineering	2009
BS	METU, Industrial Engineering	2007

WORK EXPERIENCE

Year	Place	Enrollment
2007-Present	METU, Industrial Engineering	Research Assistant
2006-2007	METU, Industrial Engineering	Student Assistant
2006	BOSCH	Intern Engineering Student
2005	ROKETSAN	Intern Engineering Student

PRESENTATIONS AT CONFERENCES

Tezcaner D. and M. Köksalan, "Multi Objective Route Planning for Unmanned Air Vehicles," 20th International Conference on MCDM, Chengdu, China, June 2009.

Tezcaner D. and M. Köksalan, "An Interactive Approach for Multi-Objective Route Selection Problem," 23rd European Conference on Operational Research, Bonn, Germany, July 2009.

Tezcaner D. and M. Köksalan, "An Interactive Algorithm for a Bi-Criteria Routing Problem," 21st International Conference on MCDM, Jyväskylä, Finland, June 2011.

Tezcaner D. and M. Köksalan, "Bi-Criteria Route Planning for Unmanned Air Vehicles," INFORMS Annual Meeting, Charlotte, North Carolina, USA, November 2011.

Tezcaner D. and M. Köksalan, "An Interactive Approach for Multi-Objective Route Selection Problem," 25th European Conference on Operational Research, Vilnius, Lithuania, July 2012.

Dehnokhalaji, A., P. Korhonen, M. Köksalan, D. Tezcaner, J. Wallenius and N. Nasrabadi, “Constructing a Strict Total Order for Alternatives Characterized by Multiple Criteria,” INFORMS Annual Meeting, Phoenix, Arizona, USA, October 2012.
Tezcaner D. and M. Köksalan, “An Interactive Algorithm for Bi-Objective Routing Problems,” 22nd International Conference on MCDM, Malaga, Spain, June 2013.
Tezcaner D. and M. Köksalan, “An Interactive Algorithm for Bi-Objective Routing Problems,” International IIE Conference and 33rd YA/EM National Conference, İstanbul, Turkey, June 2013.

WORKSHOPS ATTENDED

COST Action IC0602 Doctoral School on Decision Theoretic Artificial Intelligence, Cork, Ireland, April 2009.
MCDM Summer School, Ecole Centrale Paris, France, June 27–July 9, 2010.

SEMINARS

Multi-Objective Route Planning for Unmanned Air Vehicles, Middle East Technical University, Ankara, Turkey, March 2010.
An Interactive Algorithm for Multi-Objective Route Planning, Aalto University, School of Economics, Helsinki, Finland, March 2011.
An Interactive Algorithm for a Bi-Criteria Route Planning Problem, Preconference of the 21st International Conference on MCDM, Helsinki, Finland, June 2011.
Calling CPLEX from C++, TOL (Laboratory for Design and Optimization) Seminar, Middle East Technical University, Ankara, Turkey, March 2013.
An Interactive Algorithm for Multi-Objective Routing Problem, Middle East Technical University, Ankara, Turkey, April 2013.

PUBLICATIONS

Tezcaner D. and M. Köksalan, (2011) “An Interactive Algorithm for Multi-Objective Route Planning,” *Journal of Optimization Theory and Applications* 150 (2), 379–394.
Dehnokhalaji, A., P. Korhonen, M. Köksalan, N. Nasrabadi, D. Tezcaner and J. Wallenius, “Constructing a Strict Total Order for Alternatives Characterized by Multiple Criteria: An Extension,” *Naval Research Logistics*, *under review*.

FOREIGN LANGUAGES

English (fluent), German (intermediate)

HOBBIES

Playing the piano, Swimming