MULTIOBJECTIVE AERIAL SURVEILLANCE PROBLEM

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

BY

GÜLİZ MARAŞ

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

DECEMBER 2013

Approval of the thesis:

MULTIOBJECTIVE AERIAL SURVEILLANCE PROBLEM

submitted by **GÜLİZ MARAŞ** in partial fulfillment of the requirements for the degree of **Master of Science 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** __________________

Assoc. Prof. Dr. Esra KARASAKAL
Supervisor, **Industrial Engineering Dept., METU** __________________

Assoc. Prof. Dr. Orhan KARASAKAL
Co-Supervisor, **Turkish Navy** __________________

Examining Committee Members:

Assoc. Prof. Dr. Z. Pelin BAYINDIR
Industrial Engineering Dept., METU __________________

Assoc. Prof. Dr. Esra KARASAKAL
Industrial Engineering Dept., METU __________________

Assist. Prof. Dr. M. Kemal TURAL
Industrial Engineering Dept., METU __________________

Assist. Prof. Dr. Özgen KARAER
Industrial Engineering Dept., METU __________________

Assist. Prof. Dr. Altan ÖZKİL
School of Civil Aviation, Atılım University __________________

Date: 12.12.2013

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: Güliz MARAŞ

Signature:

ABSTRACT

MULTIOBJECTIVE AERIAL SURVEILLANCE PROBLEM

Maraş, Güliz
M.Sc., Department of Industrial Engineering
Supervisor: Assoc. Prof. Dr. Esra KARASAKAL
Co-Supervisor: Assoc. Prof. Dr. Orhan KARASAKAL

December 2013, 132 pages

In this study, we address mission planning for aerial reconnaissance and surveillance platforms. In Aerial Surveillance Problem (ASP), an air platform with surveillance sensors searches a specified number of rectangular areas once by covering inside of rectangles in strips and turns back to base where it starts. This study proposes methods to solve ASP with two conflicting objectives, minimizing distance travelled and maximizing minimum probability of target detection. Computational results show that the proposed methods produce high quality solutions. We propose an interactive procedure to help decision maker choose the most satisfying solution among all the pareto optimal solutions.

Keywords: Travelling Salesman Problem, Aerial Surveillance, Multiobjective Optimization, ε-Constraint Method

ÖZ

ÇOK AMAÇLI HAVADAN GÖZETLEME PROBLEMİ

Maraş, Güliz
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Esra KARASAKAL
Ortak Tez Yöneticisi: Doç. Dr. Orhan KARASAKAL

Aralık 2013, 132 sayfa

Bu çalışmada, havadan keşif ve gözetleme yapan platformlar için görev planlaması yapılması ele alınmıştır. Havadan Gözetleme Problemi'nde (HGP), gözetleme amaçlı kullanılan sensörlerle donatılmış bir hava aracı bir kalkış noktasından göreve başlar, belirli bir sayıdaki dikdörtgen şeklindeki alanı şeritler halinde kapsayacak şekilde tarama yapar ve kalkış noktasına geri döner. Bu çalışmada, birbiri ile çelişen iki amaçlı bir Havadan Gözetleme Problemi'ni çözmek için yöntemler önerilmiştir. Bu amaçlar, görev boyunca dolaşılan mesafeyi enazlamak ve en küçük hedef tespiti olasılığını ençoklamaktır. Performans metrikleri ile alınan sonuçlar, önerilen yöntemlerin yüksek kaliteli sonuçlar ürettiğini göstermiştir. Son olarak, önerilen yöntemlerle oluşturulan etkin sınırlardaki çözümlerden karar vericinin isteklerini en iyi karşılayan çözümü seçmesi için karar vericiyi yönlendirecek bir yöntem sunulmuştur.

Keywords: Gezgin Satıcı Problemi, Havadan Gözetleme, Çok Amaçlı Eniyileme, ε-Kısıt Methodu

To my precious family and my lovely husband

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Esra KARASAKAL and my co-advisor Assoc. Prof. Dr. Orhan KARASAKAL for their guidance, advice, criticism, patience and insight throughout the study.

I would also like to express my sincere appreciation for my lovely husband Cihan MARAŞ for his valuable love, friendship, support and help.

For their understanding, endless patience and encouragement throughout my study, my sincere thanks go to my parents, Sebahattin KAYA and Seyhan KAYA, and my sister, Nergiz TOPUZ.

I am also grateful to my company ASELSAN and my coworkers Esra EROĞLU, Eda GÖKSOY, Çağla ATEŞALP and Başak DİLBER for their encouragement and support during my thesis.

I would like to express my deepest gratitude to Meltem ERSOY who listened my complaints and motivated me during this study.

I would like to thank The Scientific and Technological Research Council of Turkey (TUBITAK) for the funding they have provided during my study.

My last appreciation goes to members of “PİKNİKÇİLER” group.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Aerial Surveillance is conducted by manned or unmanned air vehicles with the aim of reconnaissance and surveillance in many military and civilian applications such as land and coastal border security applications to detect and track illegal activities, search and rescue operations, mapping of surface topology, oil spill detection, traffic monitoring etc. Sensors like electro-optics and radars are mounted on air vehicles and used for surveillance missions.

Aerial surveillance can be performed in two ways. Either the search area is monitored entirely or only the most interested areas in the region are monitored. In this study, the focus is on the latter. In the latter case, the regions where the target is supposed to be are searched. Target can be a smuggler in border security applications, a lost person in search and rescue operations or a vessel discharging oil to the sea in oil spill detection. In all these operations, air vehicle takes off from a base station, visits the regions to be searched and turns back to the base station again.

While planning these operations, one of the aims is to minimize cost due to high operational costs of aerial surveillance. Moreover, time is another important concern in target detection. As operation cost and operation time of an aircraft are proportional to the distance travelled, the main objective is to minimize the distance travelled.

In border security applications, search and rescue operations and oil spill detection, goal of the operation is to detect the target. Therefore, another important objective in aerial surveillance is maximization of the target detection probability.

For planning aerial surveillance operations, Aerial Surveillance Problem (ASP) is defined by Ng & Sancho (2009) and Karasakal (2013). ASP is to find a path for an air platform that starts to travel from a base station, visits each rectangular area, conducts a search in strips in each rectangle and turns back to the base station. ASP can be considered as an extension of Travelling Salesman Problem (TSP). The difference between TSP and ASP is the visiting points. In TSP, only points are visited. On the other hand, in ASP rectangles are visited and each rectangle is searched in strips.

Two mixed integer linear programming models are developed for solving two single objective ASPs with the objectives of minimizing distance travelled and maximizing minimum probability of target detection (Karasakal, 2013).

In this study, we try to optimize these two objectives simultaneously. We formulate a Multiobjective ASP by combining the models developed by Karasakal (2013). An exact method and heuristic methods are proposed to solve Multiobjective ASP. Moreover, an interactive solution procedure is proposed to guide decision maker while selecting one solution from the pareto optimal set. To the best of our knowledge, our study is the first attempt to solve Multiobjective Aerial Surveillance Problem.

Organization of the thesis is as follows: In Chapter 2, a literature review on Aerial Surveillance and Multiobjective Travelling Salesman Problem (MOTSP) is given. MOTSP literature is reviewed as ASP and TSP are similar problems. Problem is defined and formulated in Chapter 3. An exact method and a heuristic method are proposed to solve Biobjective ASP in Chapter 4. Computational results are reported in Chapter 5. In Chapter 6, an interactive solution procedure is presented. The study is concluded in Chapter 7 with some remarks.

CHAPTER 2

LITERATURE SURVEY

This chapter aims to make a review of the previous works done on Aerial Surveillance and Multiobjective Travelling Salesman Problem. Some definitions about Multiobjective Optimization are given in the second section of the chapter.

2.1 Aerial Surveillance

The literature is sparse on aerial surveillance in the literature. The first study about aerial surveillance is conducted by Panton and Elbers (1999). They develop an optimization software for a military organization to plan flight of an aircraft with synthetic aperture radar with the aim of minimizing distance travelled. Aircraft takes off from a base, flies over up to 20 rectangular regions named as swaths and lands to a possibly different base. Swaths are of same width and variable length and orientation. Aircraft travels along the length of swath with four different possible patterns. An IP formulation is proposed for the problem. The method also incorporate with constrains about mandatory runs and no fly zones. The weakness of their method is that it is specific to be used for search zones that are like strips.

Panton et al. (2001) divides a region to swaths for regional surveillance. Regional surveillance means systematically monitoring of an entire region. The aim of the study is to minimize the distance travelled while searching over entire region. Two aspects make the problem harder to solve. The first one is that regions of interest generally are not rectangular areas. The second one is that the aircraft travels over rectangular areas as swaths. Therefore, the problem is to divide the region to swaths and plan the sequence of travelling the swaths. An IP formulation and a heuristic method with three variations of genetic algorithm are developed for the problem.

There are also some studies of aerial surveillance that focus on maritime surveillance. Maritime aerial surveillance is to monitor surface vehicles at sea with an air platform. Gihanmi (2002) develops a maritime surface surveillance system to detect and track illegal vessels. The system composes of three subsystems. The first one is used to generate and update the probability map for target position when a target is detected. The area to be searched is decided based on that probability map. Then, the target's position is assumed to be distributed over the selected area to be searched. The second subsystem decides the way of searching the area. Two different heuristics are used for rectangular regions and irregular regions. If the target is close to the boundary of search area and target cannot be detected by area search, with third subsystem, barrier search is applied on the boundary.

Maritime surface surveillance is also studied by Grob (2006). The aim of the study is to get as much information as possible about surface ships at sea. A mathematical formulation is described for planning operation of a single surveillance unit. Operation of the unit is changed continuously according to the current state of surface ships. As current state changes very rapidly, it is not possible to solve the problem optimally for each state. Therefore, a heuristic method is proposed to solve problem and it is tested on a simulation tool.

In some cases, using just a single surveillance unit may not be feasible and multiple surveillance units are needed. There are some studies which focus on aerial surveillance with multiple platforms. Jacobson et al. (2006) offer a method for routing of multiple aerial search platforms that are used with the aim of reconnaissance and surveillance. The difference in their method is that they plan multiple search platforms simultaneously. Moreover, search area is not divided into swaths or small regions as in previous works, because there is no need for full coverage of region. Instead, search platforms have to visit a number of points in the search area. Therefore, the problem is modeled as Multiple Travelling Salesman Problem. A simultaneous generalized hill climbing algorithm is used to solve the problem.

Simonin et al. (2008) consider planning of multi-sensors in multi-zones. The aim is to maximize the probability of detection for a moving target. A bi-level solution procedure is proposed. In the first level, sensors are assigned to search areas and in the second level, the best search strategy for each sensor is found. Extension of the proposed method to multi-period search is also given.

The first study on ASP is done by Ng and Sancho (2009). A dynamic programming formulation is proposed for ASP with the aim of minimizing distance travelled. The method can solve problems with up to 6-7 rectangles. The problems with more rectangles cannot be solved due to curse of dimensionality. The approach used in the model to define the travel in the rectangles has a weakness as the width of search strips in each rectangle is equal. This results in surveillance of unnecessary regions if the rectangle's width is not divisible by the width of search strips.

Karasakal (2013) proposes efficient solution models that contribute to the study of Ng and Sancho (2009). Two mixed integer linear programming models are introduced to solve ASP with two different objectives, minimizing distance and maximizing the minimum probability of target detection. Both of the models can solve problems up to 40 rectangles in reasonable times using a commercial solver. Moreover, the models allow search strips with different widths in each rectangle to eliminate surveillance of unnecessary regions. The details of the models are given in Chapter 3.

2.2 Multiobjective Travelling Salesman Problem (MOTSP)

In this chapter, we provide a literature review on MOTSP. There are many different approaches such as local search methods, evolutionary algorithms and memetic algorithms that are used to solve multiobjective travelling salesman problem.

We provide a short introduction to multiobjective optimization before the survey of literature on MOTSP below:

In multiobjective optimization, there is more than one objective to be optimized simultaneously. In general, multiple objectives are in conflict with each other and for only trivial cases, there is only one global optimum solution for multiobjective optimization problems. Generally, there is a set of solutions none of which are better than the other solutions in the set for all objectives. This results in trade-offs between objectives.

A multiobjective optimization problem with p objectives is formulated as

Minimize $(z_1(x), z_2(x), \ldots, z_p(x))$

subject to $x \in X$;

where $z_k(x)$ represents the k^{th} objective function and X is the set of feasible solutions or decision space. $Z = \{z(x) : x \in X\}$ is the image of *X* in the objective space.

Definition 1. A feasible solution $x \in X$ dominates $x' \in X$ if and only if, $z_k(x) \geq z_k(x')$ $\forall k \in \{1, \ldots, p\}$, with at least one index k for which the inequality is strict. The image of solution x in objective space Z is called a non-dominated solution or efficient solution.

Definition 2. A feasible solution $x \in X$ is a weakly efficient or weakly nondominated solution if and only if there is no $x' \in X$ such that $z_k(x) > z_k(x')$ $\forall k \in \{1, \ldots, p\}$.

The set of all nondominated solutions is defined as “Pareto Optimal Set”, “Efficient Frontier” and “Efficient Solutions Set”.

In general, MOTSP can be formulated as follows:

Minimize $(z_1(x), z_2(x), \ldots, z_p(x))$ where $z_k(\mathbf{x}) = \sum_{i=1}^{n-1} c_k^{\mathbf{x}(i),\mathbf{x}(i+1)} + c_k^{\mathbf{x}(n),\mathbf{x}(1)}$

subject to $x \in X$;

In the formulation, there are p objectives defined by cost matrix c_k and x is the cyclic permutation of n cities. Different objectives can be represented in the formulation by cost matrix. The cost matrix can be related with distance, cost, time, profit, safety etc.

One of the first studies about MOTSP is conducted by Hansen (2000). In local search based heuristic optimization problems, the effects of metrics that are used to guide the search are analyzed. Two substitute scalarizing functions that are based on Tchebycheff metric and weighted sum are tested by using tabu search method. It is shown that substitute scalarizing functions gives better results over the Tchebycheff metric. However, it is not possible to generalize that outcome for all MOTSP instances as just a few instances are studied.

Genetic Algorithm is a widely used method in MOTSP. Genetic algorithm is generally combined with a local search method. In that sense, Multi-Objective Genetic Local Search (MOGLS) is proposed by Jaszkiewicz (2002) to generate approximately efficient solutions for multiobjective combinatorial optimization problems. MOTSP is used to present the results of the algorithm. The method tries to optimize a randomly selected utility function at each iteration. Firstly, an initial population is generated. Then, a temporary population is constructed by the selection of best solutions among initial population according to randomly selected utility function. A pair of solutions is selected from this population as offsprings. A local search procedure is applied to these offsprings. If the solution found is better than the worst solution in temporary population, solution is added to the population. Local search used in the method for MOTSP is 2-opt neighborhood method. Weighted linear functions are used as utility functions instead of weighted Tchebycheff functions relying on Hansen's research (2000). The model is solvable even for 3 objective problems with 100 cities.

Samanlioglu et al. (2008) proposes a memetic random-key genetic algorithm that is inspired by MOGLS (Jaszkiewicz, 2002). The method combines Genetic Algorithm with 2-opt move as in MOGLS. In the method, as opposed to MOGLS, search is guided by weighted Tchebycheff metric. At the initialization process of the algorithm, 2-opt is applied to all chromosomes and also 2-opt is applied after implementation of each genetic operator. The algorithm aims to solve TSPs with five or more objectives. The results of the method are compared with the results that found by Hansen (2000) for 2, 3, and 5 objective problems. It is shown that their method gives results superior to or at least as good as those found by Hansen. Another advantage of the method is that random-key genetic algorithm eliminates the infeasible tours in crossovers which are encountered commonly in other GA approaches.

Another method that uses Genetic Algorithm and 2-opt local search method is proposed by Agrawal (2011). The method uses Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) with a modified 2-Opt and Jumping Gene adaptation to solve MOTSP. Firstly, an initial population is generated and the population is ranked based on non-domination. Then, first N solutions are selected and this population is called as Pi. Then, crossover and Jumping Gene mutation is applied to Pi. In Jumping Gene mutation, two points are chosen randomly on chromosome and the part between these two places is replaced with a random same length gene. Then, 2-opt is applied to the population. This modified population is called as Qi. Pi and Qi are combined. The combined population is sorted again according to non-domination to construct P_{i+1} and this continues until a stopping condition is achieved.

Instead of 2-opt local search, Kumar and Singh (2007) used 3-opt move with Genetic algorithm. They introduce a method by hybridizing Pareto Converging Genetic Algorithm (PCGA) with 3-opt local search to find pareto optimal set of biobjective TSP. By using steepest local search with 3-opt move, an initial population which is clustered at the extreme points of both objectives is generated. Then, PCGA is applied to initial population and all offsprings are improved by steepest local search. It is concluded that hybridization improves the quality of solution set compared to using just PCGA method.

A quite different method based on Genetic Algorithm is proposed by Yan et al. (2003). Multi-objective Evolutionary Algorithm (MOEA) starts searching with a randomly generated population and each solution in that population is modified by both crossover and mutation operators to generate the new individuals. New individuals are selected by two different selection operators, namely Family Competition and Population Competition. Family Competition compares new individual with only father of it and selects if it is better than its father. Population Competition compares new individual with whole population and selects if it is better than any solution in the population. “Better” property is used instead of fitness functions.

There are some other studies that are based on mainly Local Search Methods. Two-Phase Local Search (TPLS) method is an example for these methods. It is introduced by Paquete and Stützle (2003) for finding an approximate set of efficient solutions for biobjective combinatorial optimization problems. In the first phase of the procedure, an initial solution is found out by optimizing only one single objective. This initial solution is used in the first iteration of second phase. In second phase, Iterated Local Search method is applied on the initial solution. The search is guided by weighted sum of two objective functions. The outcome of local search is used as the initial solution of upcoming iteration of second phase. The objective function of second phase is modified at each iteration by changing the weights of two objectives. The solution found at each iteration is saved as an efficient solution. The second phase is repeated until all combinations of weights are searched. Computational experiment is done on symmetric biobjective TSP. Method is compared with Jaszkiewicz’s MOGLS method and it is shown that this method outperforms MOGLS in terms of solution quality for the test problems. However, TPLS is specific to biobjective problems while MOGLS can be used for problems with 3 objectives.

Another local search method based heuristic, Pareto Local Search (PLS), is proposed by Paquete et al. (2004) for biobjective TSPs. PLS starts with a random initial tour and searches the neighborhood of it. When a nondominated neighbor is found, it is saved as an efficient solution and added to the archive. The algorithm continues until all solutions' neighborhood in the archive is searched. 3-opt is used to define the neighborhood. The results indicate that PLS has a good performance comparable to MOGLS method proposed by Jaszkiewicz. However, the disadvantage of PLS is that the computation times are much higher.

As a more complex neighborhood search technique, dynasearch is used by Angel et al. (2004) to find pareto optimal set for biobjective TSP. Dynasearch allows a series of moves at each iteration. ds-2-opt is used as neighborhood structure in algorithm. The ds-2-opt neighborhood is defined by set of a series of independent 2-opt moves. Two 2-opt moves are independent if the vertices involved are distinct. Algorithm uses dynamic programming to search the neighborhood.

For biobjective TSPs, Li (2005) proposes a method using 2-opt local search method to find pareto optimal sets by using attractors of each single objective. Proposed method firstly attempts to find attractors of each single objective and then merges them to find pareto optimal set. To find the attractors, 2-opt local search method is applied on randomly generated tours. The edges involved in the local optimal solutions are recorded in a hit frequency matrix. Hit frequency matrix gives the arcs that will be most probably in the optimal solution of the single objective problem. Then, attractors of two objectives are merged and searched to find out a non-dominated solutions set.

Path-relinking concept is applied together with 2-opt local search and tabu search in Pareto Memetic Algorithm (PMA) for biobjective TSPs (Jaszkiewicz et. al., 2009). The algorithm firstly uses Lin–Kernighan algorithm to optimize each objective separately and found solutions are saved in solution set. Then, with same method, whole solution set is constructed based on optimizing weighted sum of objectives with randomized weights. After that, a sample set is chosen out of solution set and the best and the second best solutions are selected according to randomly selected weighted sum of objectives. These two solutions are combined by path relinking and improved by a local search method. Local search used in this step is composed of two stages, 2-opt local search method and a short run of tabu search. If the solution is better than the second best solution, solution set is updated. Finally, all solutions' 2-opt neighborhood is searched by Pareto Local Search. The performance of algorithm is compared with Paquete and Stützle's (2003) Two-Phase Local Search method. Presented solutions suggest that PMA generates better solutions, although further analysis is needed to prove that.

There are many heuristic methods introduced above for MOTSP. However, these heuristic methods are tested on only a set of sample problems. Thus, their performance is not guaranteed for all instances. To cope with this uncertainty, approximation algorithms are introduced. Approximation algorithms produce solutions that are guaranteed to be within a specific range of optimum solution. Manthey et al.(2009) proposes approximation algorithms to find approximate pareto curves for MOTSPs that have both maximization and minimization objectives. The algorithms can be used for any number of max and min objectives. As maximization and minimization objectives have different properties, maximization objectives are firstly dealt and the paths that have sufficient weight for maximization objectives are collected. Then, these paths are connected to get a Hamiltonian cycle while only minimization objectives are considered in this step. These algorithms have polynomial running-time.

Another approximation algorithm is proposed by Bazgan et al. (2012) to find a single tour instead of finding the efficient set for biobjective TSPs whose objectives are of maximization type. In algorithm, firstly the set of edges having maximum weights for both objectives are found and then they are connected in a systematic way.

Stochastic Local Search (SLS) algorithms are proposed to deal with the weaknesses of local search methods. As getting stuck in local optima is a typical problem of Local Search, in SLS, randomized initialization and search steps are used to overcome this problem. Paquete and Stützle (2009) study the effects of components used in the SLS algorithms for MOTSP on the algorithms' performance. The components analyzed are search strategy, component-wise step, neighborhood structure, number of scalarizations and search length. As search strategy, they analyze Two-Phase Local Search Strategy, where at each iteration of algorithm previous iteration's outcome is used as initial solution in the next iteration, and Restart Strategy, where at each iteration search starts with a random initial solution. They compare both strategies with some experiments and show that Two-Phase strategy gives better results. As component-wise step, they analyze the effect of including all nondominated solutions in the neighborhood of the solution returned at each iteration. This strategy increases number of nondominated solutions in the search, while causing a slight increase in computation time. For neighborhood structure, using an iterative improvement algorithm, like starting with 2-opt exchange and continuing search with 3-opt exchange afterwards, is recommended. As the number of scalarizations (using different weights of objectives) and the search length increases, algorithm gives better results. An SLS algorithm constructed by using these recommended components is compared with MOGLS (Jaszkiewicz, 2002). For the test problems, it is shown that SLS algorithm outperforms MOGLS.

Lust et al. (2010) make a survey to introduce existing works on MOTSP literature. Interested readers are referred to this study for additional information on MOTSP. They also propose a two phase method for finding efficient solutions of MOTSP. In the first phase, supported efficient solutions are found by solving single objective TSPs where objective is the weighted sum of the objectives of MOTSP. Single objective TSP is solved by Lin Kernighan heuristic with all set of weights. In the second phase, aim is to find the efficient solutions that are between supported efficient solutions. A Pareto Local Search similar to Paquete et al. (2004)'s work is used. The proposed method gives better solutions than Paquete et al.'s PLS method and gives comparable results with respect to Jaszkiewicz et al. (2007)'s PMA and Paquete and Stützle (2003)'s TPLS.

The literature on MOTSP reviewed up to now includes studies done on MOTSPs that include only static parameters. The classification of solution methods reviewed up to now is given in Table 1. The first classification is done on number of objectives as biobjective and multiobjective. “Multiobjective” is used to represent methods to solve problems with more than 2 objectives. It is observed that there is almost equal number of biobjective and multiobjective studies. Second classification is done on Search Strategy; local search, evolutionary, memetic and other. In literature, methods to find pareto optimal solutions of MOTSP generally relies on local search methods. If evolutionary methods are used, they are also combined with local search methods as in memetic methods. As local search methods are dominant in MOTSP literature, neighborhood structure is also investigated and given in Table 1. It is observed that 2-opt and/or 3-opt neighborhood structure is used commonly in MOTSP literature.

Moreover, there are some studies on MOTSPs which include dynamic features such as cost matrices, number of cities and number of objectives. Dynamic characters of the problem make it harder to solve. To tackle this complexity, Yang et al. (2008) introduce Multi-Algorithm Co-evolution Strategy (MACS). MACS uses multiple algorithms to get advantage of each algorithm’s powerful properties. The algorithms embedded in MACS separately are good at optimizing the population locally, optimizing the population globally, finding optimal for each objective and making the population close to the Pareto optimal front. MACS selects which algorithms to use as the problem parameters changes.

Another method is proposed for MOTSPs with dynamic features by Li et al. (2012). In the method, by using parallel processors, extreme solutions of k objectives are found at the same time and send to another processor to be merged into a solution matrix. Then, this solution matrix is analyzed to find non-dominated solutions. When decision maker asks for a solution, nondominated solutions are presented. Whenever a change in problem parameters occurs, changes are sent to processors and procedure starts again. The method is tested with randomly generated number of cities in the range of [10, 100] and randomly selected 2 or 3 objectives.

TSP with profits is a variant of multiobjective travelling salesman problem. It has two conflicting objectives; the minimization of the tour length and the maximization of the collected profits. Salesman gains profits when a city is visited. TSP with profits differs from multiobjective travelling salesman problem, as salesman does not have to visit all cities in the network. Instead, a trade-off between distance travelled and collected profits is done. A survey of the studies on TSP with profit is done by Feillet (2005). According to the survey, TSPs with profit is classified on the way the two objectives are addressed. Two objectives are either combined in one objective or one objective is used as constraint while other is optimized. These problems are referred in literature as Profitable Tour Problem, Orienteering Problem, Selective TSP and Prize Collecting TSP. After that survey, there are some other studies relying on multiobjective approaches that do not treat the problem as it is single-objective.

As an early work, Keller and Goodchild (1988) propose a method to solve TSP with profit. The heuristic proposed starts with an initial solution and tries to reduce the distance travelled by eliminating crossing paths and changing the place of each node in the tour. When no further improvement can be achieved on distance travelled, profit is tried to be maximized while ensuring that distance travelled does not exceed a predefined distance limit. In this step, node insertion methods are used. If an improvement is achieved, distance travelled is tried to be decreased again. If no improvement can be achieved, then eliminating every single node in the tour and eliminating the isolated node clusters are tried. The heuristic is tested on a 25 city problem.

ε-constrained method is used to convert multiobjective problems into single objective ones. One objective is selected as objective function and all other objectives are added to the problem as constraints. It is applied to TSP with profits in different ways. A multiobjective approach is developed (Şimsek, 2007) by combining ε-constrained method with a heuristic method in order to find the efficient frontier for the TSP with profit. Maximizing profit is chosen as objective and distance travelled is used as constraint. The method used to solve that single objective problem is CGW

(Chao, Golden, and Wasil) heuristic. CGW heuristic finds an initial solution with a greedy algorithm. Then, two-point exchange, one point movement and 2-opt procedures are applied to the initial solution in succession. The computational results show that proposed method gives good results for test problems. Even though, run times increase significantly for problems larger than 50 cities.

Berube et al. (2009) also use ε-constraint method to find pareto-optimal set of solutions for TSP with profits. As opposed to Şimşek's study (2007), minimization of distance travelled is used as objective function and profit is used as constraint. The problem is solved by branch and cut procedure. To speed up the method, some heuristic improvements are proposed on branch and cut algorithm. Improvements includes removing dominated points and improving the initial feasible solutions by using similarities of consecutive problems

Jozefowiez et al. (2008) proposes a new approach for TSP with profits that does not convert the problem into single-objective problems. They introduce a method that uses ejection chain local search combined with a multiobjective evolutionary algorithm that is a variant of NSGA II. The method is compared with a ε-constraint method and it is shown that their method has advantages when city size increases up to 150s.

Another study that also tries to optimize two objectives simultaneously is done by Karademir (2008). A modified Multiobjective Genetic Algorithm NSGA II and the Lin-Kernighan Heuristic is used to find pareto optimal solutions for TSP with profit. With the proposed method, pareto optimal set for problems including less than 150 cities are found and Pareto-optimal set with at most 2% deviation is found for larger problem sizes.

Table 1 : Classification of MOTSP with Static Parameters

	# of Objectives		Search Strategy				
	Biobjective	Multiobjective	Evolutionary	Local Search	Memetic	Other	Neighborhood Structure
Hansen (2000)		x		x			2-opt
Jaszkiewicz (2002)		x			x		2-opt
Paquete and Stützle (2003)	X			x			2-opt / 3-opt
Yan et al. (2003)		x	x				-
Paquete et al. (2004)	X			x			3-opt
Angel et al. (2004)	X			x			ds-2-opt
Li (2005)	X			x			2-opt
Kumar and Singh (2007)	X				x		3-opt
Samanlıoğlu (2008)		x			x		2-opt
Jaszkiewicz et al. (2009)	X				x		2-opt
Paquete and Stützle (2009)		x		x			2-opt/3-opt
Manthey (2009)		x				x	
Lust et al. (2010)	X			x			2-opt
Agrawal (2011)		x			x		2-opt
Bazgan et al. (2012)	X					x	

CHAPTER 3

PROBLEM DEFINITION AND MODEL

In Aerial Surveillance Problem (ASP), an air platform with surveillance sensors visits a specified number of rectangles and inside of each rectangle is covered in strips. Then, air platform turns back to the base where it starts the travel.

In this chapter, two existing models proposed to solve single objective ASPs are reviewed and a model to solve biobjective ASP is introduced. In Section 3.1, two models developed by Karasakal (2013) to solve single objective ASP are examined in details. In Section 3.2, a new model is proposed to solve biobjective ASP.

3.1 Single Objective Aerial Surveillance Problem Definition and Models

Karasakal (2013) develops two mixed integer linear programming models to solve single objective Aerial Surveillance Problems. The first problem is minisum ASP with the objective of minimizing total distance travelled. The second problem is maximin ASP with objective of maximizing the minimum probability of target detection of all rectangles.

In this section, firstly, some definitions and assumptions, that are common in both of the models, are given. Then, detailed explanation of the models is made in subsections 3.1.1 and 3.1.2.

The proposed models for minisum ASP and maximin ASP are based on Travelling Salesman Problem formulation. As it is already stated, the main difference between ASP and TSP is the visiting points. In TSP, points are visited and no time is spent on these points. In ASP, visiting points are rectangular areas and these areas are searched with specified patterns. Therefore, ASP can be considered as a more complicated version of TSP.

Minisum ASP and maximin ASP also rely on the assumption that the areas to be searched are in rectangular shape. However, areas to be searched may not be rectangular in real life applications. To represent such areas, the smallest circumscribed rectangle can be used.

In the models, there are some specified patterns for travelling in the rectangles. Each pattern has different target detection probability and length. All target detection probabilities and travel distances related with each pattern of each rectangle are calculated in advance and feed into the models as parameters.

In calculation of target detection probabilities in the rectangles, target's characteristic plays an important role. In ASP, it is assumed that target is non-evading and uniformly distributed over the area to be searched. By relying on this assumption, Koopman's area search equation is used to calculate target detection probabilities: (Wagner, 1999)

$P = 1 - e^{-W/S}$ where W is sweep width and S is track spacing.

P is the probability of detecting a target when sweep width is W and track spacing is S with the assumption that target is uniformly distributed over the area.

Sweep width is a metric that is used to measure detection ability. It is the width of the swath where the number of targets not detected inside the swath is equal to the number of targets that are detected outside the swath. (Koester et al., 2004). An example is given in Figure 1 to illustrate sweep width. Sweep width is the width on lateral range curve where sum of areas A and sum of areas B are equal.

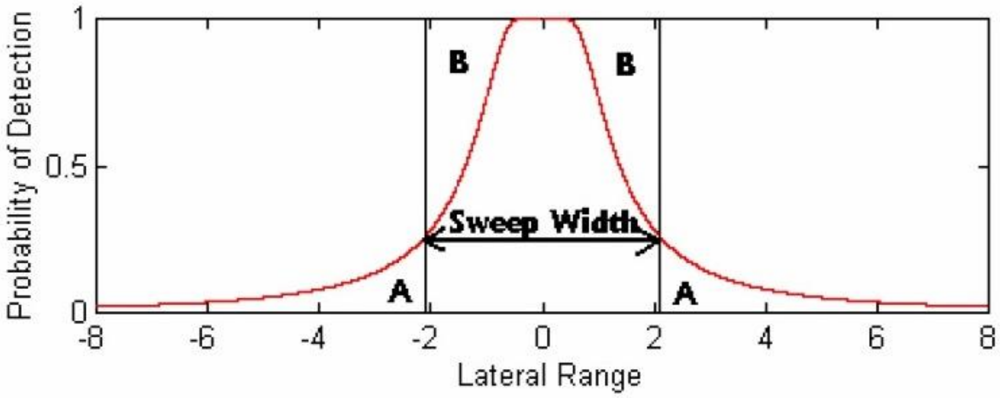

Figure 1 : A lateral range curve showing sweep width (Koester et al., 2004)

Sweep width is specific to the sensor used for surveillance missions. The sensor having larger sweep width is more capable in detection. Sweep width of a sensor can change in different environmental conditions such as rain, snow and fog. Therefore, it should be measured under a specific set of environmental conditions. In addition, as can be seen from Figure 1, when the lateral range between the target and sensor increases, probability of target detection decreases.

Track Spacing (S) is the width of each strip that the rectangular area to be searched is divided into. It directly affects target detection probability and distance travelled in the area. It should be chosen to ensure that the rectangular area is fully covered by the strips and no unnecessary area is covered. Moreover, if there is a lower limit on target detection probability in the area, track spacing should ensure that the probability of target detection is larger or equal to the lower limit.

Therefore, track spacing should be determined according to the following limitations:

- Track spacing is larger or equal to sweep width. ($S \geq W$)
- Length / width of the rectangle is divisible by track spacing (If aircraft travels parallel to the length of the rectangle, width should be divisible by S. If

aircraft travels parallel to the width of the rectangle, length should be divisible by S.)

- Track spacing ensures that minimum acceptable probability of target detection (Pmin) is satisfied.

There are usually more than one track spacing values that assure the above limitations for one entry point.

It is also assumed that air platform always enters to the rectangular area from one of the corners. Then, it travels in the middle of strips which are parallel to the width or length of the rectangle. While travelling in a strip, when air platform comes to the end of the rectangular area, it makes two turns with 90˚ angle and continues to search in opposite direction. Turning radius of the air platforms is ignored in calculation of the distance travelled in the rectangles.

In Figure 2, an example search over a rectangle is given. In the example, width of the rectangle is divided to 6 equal strips.

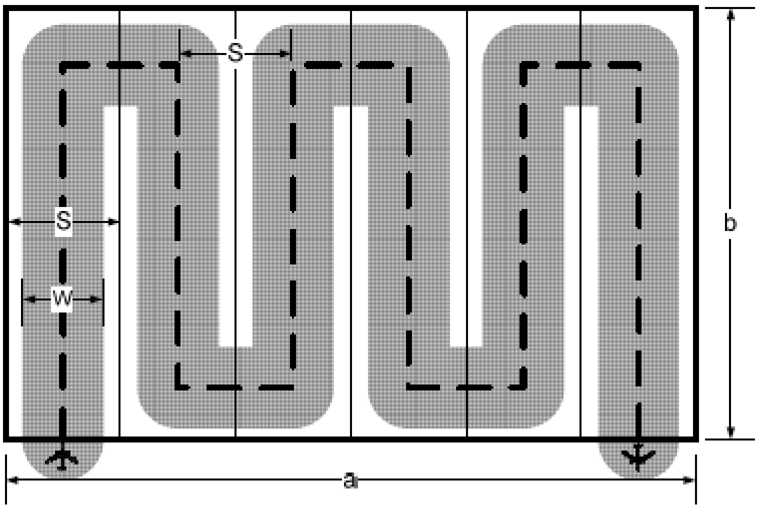

Figure 2 : An example search over a rectangle (Karasakal, 2013)

When aircraft enters the rectangular area from one edge of corner point and the selected track spacing value results in odd numbered strips, aircraft leaves the area from a place on the same edge in opposite direction. If the selected track spacing value results in even numbered strips, it leaves the area on the reciprocal edge in same direction. Two examples are given in Figure 3 to show it.

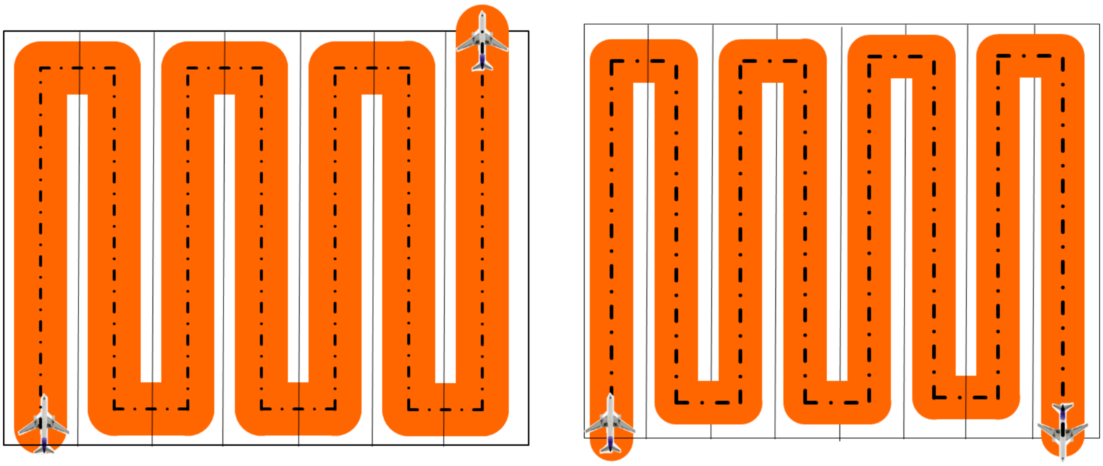

Figure 3 : Two different travelling examples starting from same entry point in the same rectangle and resulting in different exit points.

When air platform enters to the rectangular area from one edge of one corner, the entry and exit points are different for each track spacing value. This results in a number of entry/exit points in one edge of one corner. The distance between exit/entry points on one edge of one corner is small with respect to the distance travelled in the rectangle. So that, different exit/entry points on one edge of one corner are aggregated to decrease computational effort. Then, there is 8 entry and exit points for each rectangle. In Figure 4, an example showing entry and exit points is given.

Although aircraft uses same entry point and same exit point, it can travel in the rectangle in a different way due to different track spacing values. This different travels are referred as Search Pattern. In addition, entry point, exit point and search pattern for a rectangle constitutes a Search Scenario.

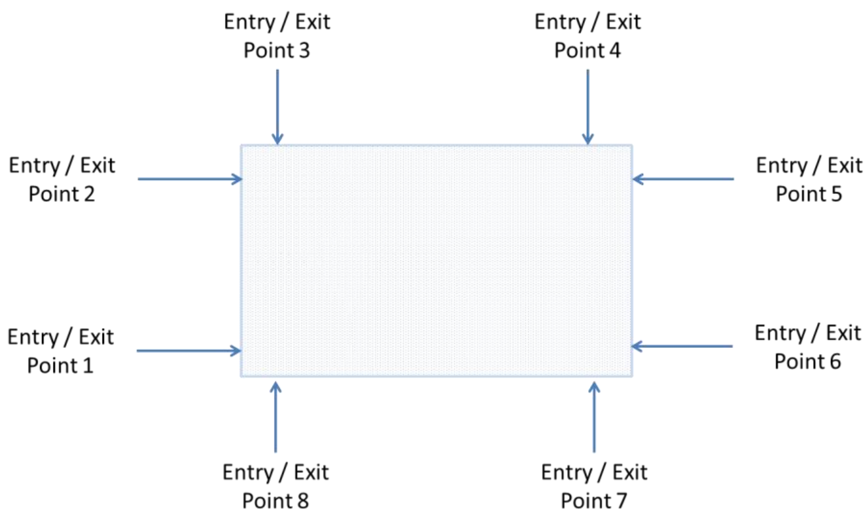

Figure 4 : An example representation of Entry and Exit points on a sample rectangle

3.1.1 Model 1: Minisum ASP

One of the major objectives in Aerial Surveillance is to minimize the operation cost of the flight. Flight operations are costly due to high fuel rates and high maintenance costs. Especially in border security operations, periodical surveillance of same areas is needed. Any decrease in the cost of these flight operations can result in remarkable decreases in surveillance costs as the number of operations increases.

Another important objective in Aerial Surveillance is to minimize the flight duration. In search and rescue operations, this objective becomes vital, because any time spent unnecessarily can cost someone’s life.

These two goals are in line with minimizing distance travelled. Hence, in Minisum ASP, aim is to find the shortest path of visiting all rectangles to be searched, travelling in all rectangles and turning back to the base.

In addition, the operations should guarantee a minimum probability of target detection to ensure that the operation is efficient. In Minisum ASP, feasible search patterns are determined according to minimum acceptable probability of target detection. Therefore, no extra constraint is needed to be added to the model for minimum probability of target detection. Minisum ASP model is given below: (Karasakal, 2013)

Indices:

i, j : index of disjoint rectangles, i=1 and j=1 represent the base station.

l, k : index of entry and exit points of rectangles, $l = 1,...,8$ and $k = 1,...,8$.

s : index of the search pattern within a rectangle for each feasible combination of entry and exit points of the rectangle.

Parameters:

N : total number of disjoint rectangles plus the base station.

D_{iljk} : distance from point l of rectangle i to point k of rectangle j.

D'_{ilks} : distance from point l to point k of rectangle i using search pattern s.

P_{ilks} : probability of detecting target by flying from point l to point k in the rectangle i using search pattern s.

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

Decision Variables:

$$y_{iljk} = \begin{cases} 1, & \text{if the air platform flies from point l of rectangle i to point k of rectangle j} \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ilks} = \begin{cases} 1, & \text{if the air platform flies from point } l \text{ to point } k \text{ of rectangle } i \text{ using search pattern } s \\ 0, & \text{otherwise} \end{cases}$$

(Minisum ASP)

$$\text{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \tag{1}$$

Subject to

$$\sum_{ilk} y_{iljk} = 1 \quad \forall j = 1, \ldots, N \tag{2}$$

$$\sum_{jlk} y_{iljk} = 1 \quad \forall i = 1, \ldots, N \tag{3}$$

$$\sum_{lks} z_{ilks} = 1 \quad \forall i = 2, \ldots, N \tag{4}$$

$$\sum_{ks} z_{ilks} \le \sum_{jk} y_{jkil} \quad \forall l = 1, \ldots, 8,\ i = 2, \ldots, N \tag{5}$$

$$\sum_{ls} z_{ilks} \le \sum_{jl} y_{ikjl} \quad \forall k = 1, \ldots, 8,\ i = 2, \ldots, N \tag{6}$$

$$u_i - u_j + N \sum_{lk} y_{iljk} \le N - 1 \quad \forall i = 2, \ldots, N \quad j = 2, \ldots, N \quad i \ne j \tag{7}$$

$$y_{iljk} \in \{0,1\} \quad \forall iljk \in \{(i,l,j,k) \mid i \ne j, i = 1 \wedge l = 1, j = 1 \wedge k = 1\} \tag{8}$$

$$z_{ilks} \in \{0,1\} \quad \forall ilks \in \{(i,l,k,s) \mid i \ne 1, l \ne k\} \tag{9}$$

$$u_i \ge 0 \quad \forall i = 1, \ldots, N \tag{10}$$

Equation (1) represents the objective function that is minimization of the total distance travelled. The first part of the objective function is the distance travelled between rectangles and the origin. The second part is the total distance travelled in rectangles. Constraint set (2) ensures that air platform leaves each rectangle from one exit point and it leaves the base station once. Constraint set (3) ensures that air platform enters each rectangle from one entry point and it enters to the base station once. Constraint set (4) ensures that one search pattern is selected for each rectangular area. Constraint set (5) ensures that if an exit point is chosen for a city, a search pattern compatible with that exit point is chosen for the city. Constraint set (6) ensures that if an entry point is chosen for a city, a search pattern compatible with that entry point is chosen for the city. Constraint set (7) represents a set of constraints that ensure subtours are not constructed. Constraint sets (8) and (9) enforce binary restriction on decision variables. Constraint set (10) represents nonnegativity restrictions for related variables.

3.1.2 Model 2: Maximin ASP

In aerial surveillance, money and time may not be the first concern every time. In such cases, maximizing efficiency of the search becomes more important. For that purpose, Minisum ASP is modified to maximize efficiency of the operation. To guarantee at least same efficiency in all rectangles, maximizing the minimum probability of target detection of all rectangles is used as the objective function of Maximin ASP. Nevertheless, as aircrafts have a limitation on flight time due to fuel restrictions, maximum flight distance of the aircraft is added to the problem as a constraint. Maximin ASP model is given below: (Karasakal, 2013)

M : maximum distance the air platform can fly.

(Maximin ASP)

$$\text{Max } \alpha \tag{11}$$

Subject to

Constraints (2) – (10)

$$\sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \le M \tag{12}$$

$$\sum_{lks} P_{ilks} z_{ilks} \ge \alpha \quad \forall i \tag{13}$$

$$\alpha \ge 0 \tag{14}$$

Equation (11) represents the objective function that is the maximization of the minimum probability of target detection of all rectangles (α). Constraint (12) ensures that total distance travelled by aircraft cannot exceed M. Constraint set (13) ensures that probability of target detection corresponding to the selected scenario for each rectangle is larger or equal to (α). Constraint (14) represents nonnegativity restriction for (α).

3.2 Multiobjective Aerial Surveillance Problem Definition and Model

Minisum ASP and Maximin ASP can be used effectively to solve single objective ASPs with objectives of minimizing total distance travelled and maximizing the minimum probability of target detection in all rectangles, respectively. Both of the models can be solved for problems up to 40 rectangles in reasonable times using commercial MIP solvers. Nonetheless, there is a need of simultaneously optimizing the conflicting objectives of Minisum ASP and Maximin ASP. To find solutions in this multiobjective case, Multiobjective Aerial Surveillance Problem (MASP) is defined.

Mathematical model for MASP is given below:

(MASP)

$$\text{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \tag{1}$$

$$\text{Max } \alpha \tag{11}$$

Subject to

Constraints (2) – (10)

$$\sum_{lks} P_{ilks} z_{ilks} \geq \alpha \quad \forall i \tag{13}$$

$$\alpha \geq 0 \tag{14}$$

Proposed methods to solve MASP are given in Chapter 4.

CHAPTER 4

PROPOSED BIOBJECTIVE APPROACH

In this chapter, we propose an exact method and heuristic methods to find the pareto optimal set for MASP. Firstly, an exact method is introduced in Section 4.1 and then, in Section 4.2 heuristic methods are presented.

4.1 ε-Constraint Method

ε-Constraint Method is used to find nondominated solutions in Multiobjective Problems. In this method, an objective is chosen to be optimized while other objectives are given as constraints. For the multiobjective problem with p objectives given below,

Minimize $(z_1(x), z_2(x), \ldots., z_p(x))$
Subject to
$x \in X$

ε-Constraint Method is applied as follows:

Minimize $z_k(x)$
Subject to
$z_i(x) \leq \varepsilon_i \quad i=1,2,\ldots.k-1,k+1,\ldots.,p$
$x \in X$

$z_i(x)$ represents the i^{th} objective function and X is the set of feasible solutions. The vector $(\varepsilon_1, \varepsilon_2, \ldots \varepsilon_{k-1}, \varepsilon_{k+1},.., \varepsilon_p)$ denotes the upper bounds of corresponding objectives.

In multiobjective optimization, objectives are generally in conflict with each other. When $z_k(x)$ is tried to be optimized, the objective functions other than $z_k(x)$ usually get worse. In other words, $z_i(x)$ $i \neq k$ values usually increases as the objectives are of minimization type. Therefore, an upper bound is set for each $z_i(x)$ $i \neq k$ to define the maximum value that they can have.

The weakness of ε-Constraint Method is that the optimal solution found can be a weakly efficient solution. If all of the constraints related with objective functions are not binding, there can be alternative optimal solutions that result in lower values of objective functions whose constraints are not binding. To deal with that weakness, Augmented ε-Constraint Method is proposed (Mavrotas, 2009).

Augmented ε-Constraint Method (AUGMECON) is applied as follows:

Minimize $\quad z_k(x) - \mu \sum_{i \neq k} d_i$

Subject to

$z_i(x) + d_i = \varepsilon_i \qquad\qquad i=1,2,\ldots.k-1,k+1,\ldots.,p$

$x \in X$

$d_i \geq 0$

where μ is a very small number and d_i is the slack variable of i^{th} objective function.

In AUGMECON, slack variables (di) are added to the constraints related with objective functions. In model's objective function, slack variables are maximized as they are extracted from $z_k(x)$. When slack variables are maximized, related objective functions are minimized. As slack variables are multiplied by a small number (μ), their effect on the model's objective function is small with respect to $z_k(x)$. Therefore, $z_k(x)$ is minimized while it is guaranteed that the other objective values are at their possible minimum value. μ should be selected with respect to objective function values' range and it should be a sufficiently small constant.

In biobjective problems, one objective is selected to be optimized and the other is transformed to a constraint. In MASP, "minimizing total distance travelled" is chosen as objective function and "maximizing minimum probability of target detection" is used as constraint.

ε-Constraint Method is applied to MASP as given below :

$$\text{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \tag{1}$$

Subject to

Constraints (2) – (10)

$$\sum_{lks} P_{ilks} z_{ilks} \geq \quad \varepsilon \qquad \forall i \tag{16}$$

As the objective of "maximizing minimum probability of target detection" is of maximization type, it is converted to minimization type by taking negative of it.

To eliminate weakly efficient solutions, Augmented ε-Constraint Method is applied to MASP as given below (it is referred as ε-MASP):

$$(\varepsilon\text{-MASP}) \qquad \text{Min} \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} + \mu\, d \tag{17}$$

Subject to

Constraints (2) – (10)

$$\sum_{lks} P_{ilks} z_{ilks} - d = \varepsilon \qquad \forall i \tag{18}$$

$$d \geq 0 \tag{19}$$

As "d" is minimized, minimum probability of target detection is forced to get its possible maximum value.

Solution of ε-MASP gives a single nondominated solution for an lower bound ε. To find complete set of pareto optimal solutions, ε-MASP should be solved iteratively with different lower bounds (ε). The algorithm for finding Efficient Solutions with ε-MASP is given below:

EfficientSolutions	: set of nondominated solutions
MinProbability	: minimum of the target detection probabilities of all rectangles
StepSize	: Constant value

Step 1 : Set *EfficientSolutions*= Ø

Step 2 : Set ε =Minimum Acceptable Probability of Target Detection

Step 3 : Solve ε-MASP with ε.

Step 4 : If a feasible solution is found, add this solution to *EfficientSolutions*. Otherwise, STOP.

Step 5 : Set *MinProbability* to the minimum of the target detection probabilities of the selected search scenarios for all rectangles in the current solution of ε-MASP.

Step 6 : Set ε = *MinProbability* + *StepSize* and Go to Step 3.

StepSize is used to increase the lower bound "ε". It ensures that the solution found in next iteration has a *MinProbability* value larger than the solution found in current iteration. *StepSize* is a value between 0 and 1. It should be a very small positive constant close to 0 to generate whole pareto optimal set. If whole pareto optimal set is not needed to be found, *StepSize* can be arranged accordingly.

4.2 Heuristic Methods

In real life applications, MASP is to be solved repeatedly in short times even up to 40 rectangles. However, MASP is NP-Hard and search space of ASP increases exponentially as the number of rectangles in the problem increases. An example calculation of number of solutions in search space is given below:

- Problem with 10 rectangles and 8 different scenarios for each rectangle
  Number of possible solutions= (10-1)! / 2 * 10 * 8 = 14.515.200
- Problem with 11 rectangles and 8 different scenarios for each rectangle
  Number of possible solutions= (11-1)! / 2 * 11 * 8 = 159.667.200

As the number of rectangles in ASP increases, problem cannot be solved by exact methods. Therefore, in this section, we propose heuristic methods to solve MASP. As the number of rectangles in ASP increases, problem cannot be solved by exact methods. Therefore, in this chapter, we propose heuristic methods to solve MASP. Three construction methods and an improvement method are defined to find efficient solution set for MASP. Construction methods find an initial solution and improvement method tries to find set of efficient solutions by searching neighborhood of the initial solution.

In Section 4.2.1, Construction Methods are explained and in Section 4.2.2, details about Improvement Method are given.

4.2.1 Construction Methods

In this study, three construction methods are defined to find the initial solution for MASP.

4.2.1.1 Construction Method 1

In Construction Method 1, firstly, the sequence of rectangles in the tour is found. Then, for each rectangle, a search scenario is selected. Thus, an initial feasible solution is found.

Construction Method 1 uses the similarity between ASP and TSP. Optimal TSP sequence of rectangles is used to set the sequence in the initial tour. Optimal TSP tour is found based on midpoints of the rectangles by using Concorde TSP Solver (Cook, 2010). Then, for all rectangles, Search Scenario (entry point, exit point and search pattern) resulting in largest probability of target detection is selected.

Figure 5 shows a flow chart for Construction Method 1.

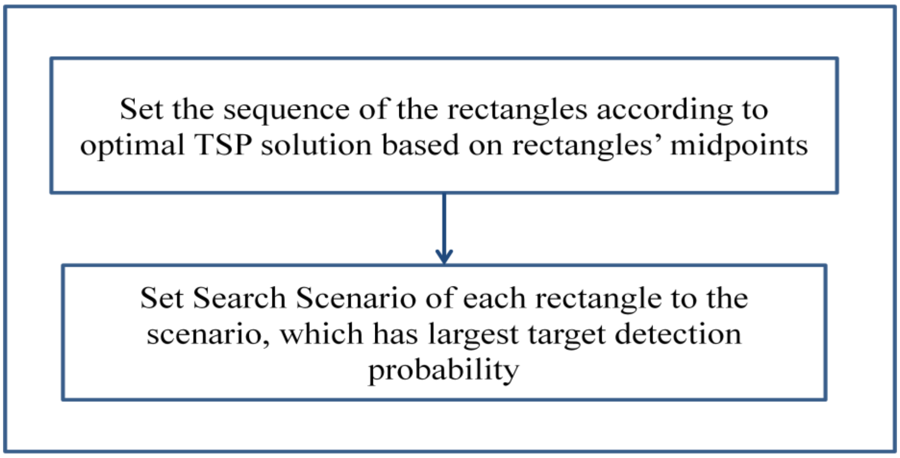

Figure 5 : Flow Chart for Construction Method 1

4.2.1.2 Construction Method 2

Construction Method 2 relies on an optimal solution. An optimal solution for Minisum ASP is used as initial tour. Minisum ASP is solved optimally in GAMS (General Algebraic Modeling Language) utilizing CPLEX MIP solver. To reduce the computation time, instead of solving the problem optimally for a number of times as in ε-MASP, it is solved optimally for only once and then Improvement Heuristic is applied.

Figure 6 shows a flow chart for Construction Method 2.

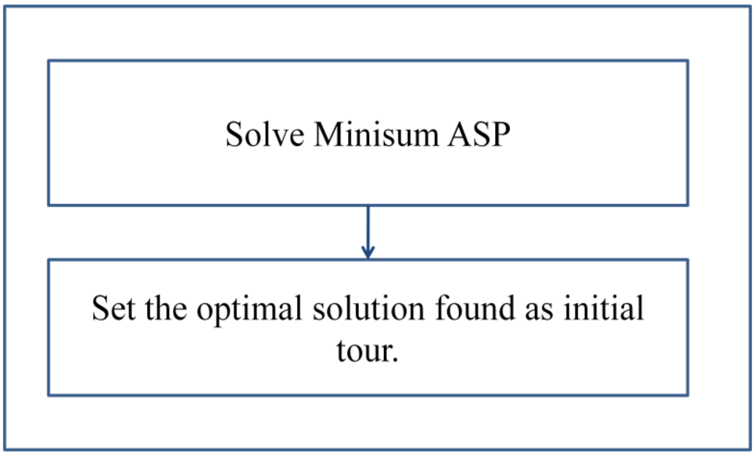

Figure 6 : Flow Chart for Construction Method 2

4.2.1.3 Construction Method 3

In Construction Method 3, a good initial solution is tried to be found randomly. In this method, a specified number of random sequences of rectangles are generated. Total distance of the sequences is calculated based on midpoints of rectangles. The sequence that results in minimum distance is selected and it is used as the sequence in the initial tour. Then, for all rectangles, Search Scenario (entry point, exit point and search pattern) resulting in largest probability of target detection is selected. In this method, the use of Concorde TSP Solver in Construction Method 1 is tried to be eliminated.

Figure 7 shows a flow chart for Construction Method 3.

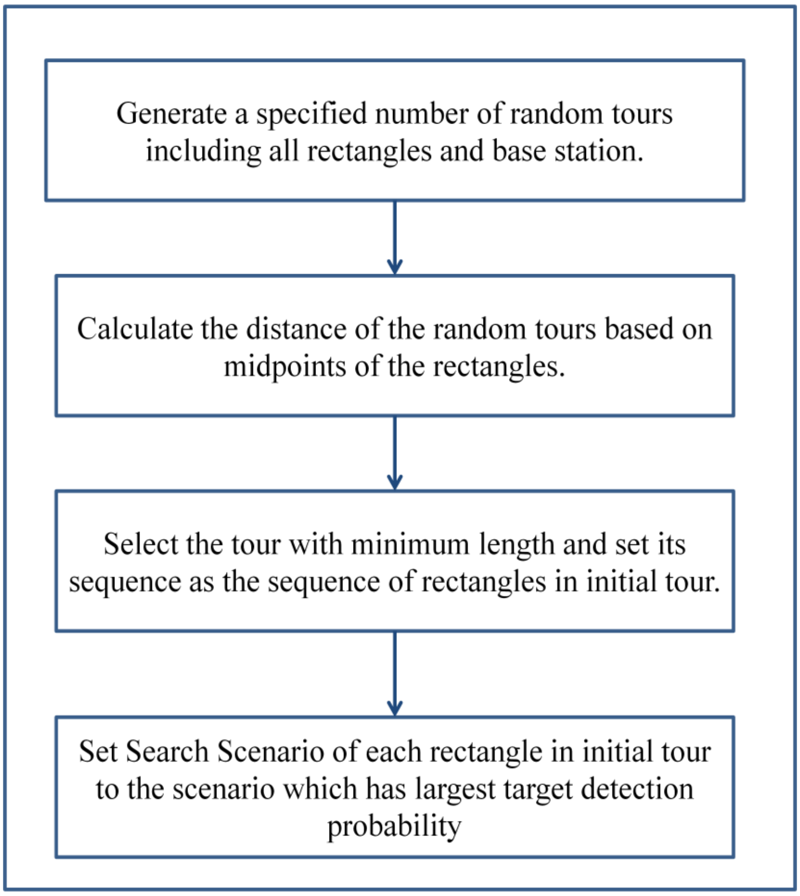

Figure 7 : Flow Chart for Construction Method 3

Algorithm of Construction Methods is given in APPPENDIX A**.**

4.2.2 Improvement Method

In improvement method, the aim is to search the neighborhood of the initial solution by directing the search to find nondominated solutions. We define an improvement method that uses an approach similar to Pareto Local Search that is introduced by Paquete (2004) to solve biobjective TSPs.

Firstly, neighborhood of the initial solution found by Construction Method is searched. Then, along the search for each nondominated solution found so far, neighborhood search is done again.

Neighborhood search relies on the rectangles' sequence in the tour and search scenarios of the rectangles. Two-Opt exchange move is used to change the sequence of rectangles. For each new solution found with Two-Opt exchange move, a set of solutions having different "minimum of target detection probability of all rectangles" values are generated by changing Search Scenarios of rectangles. Search Scenarios are selected in accordance with the aim of minimizing the distance travelled.

Figure 8 shows flow chart for the improvement method.

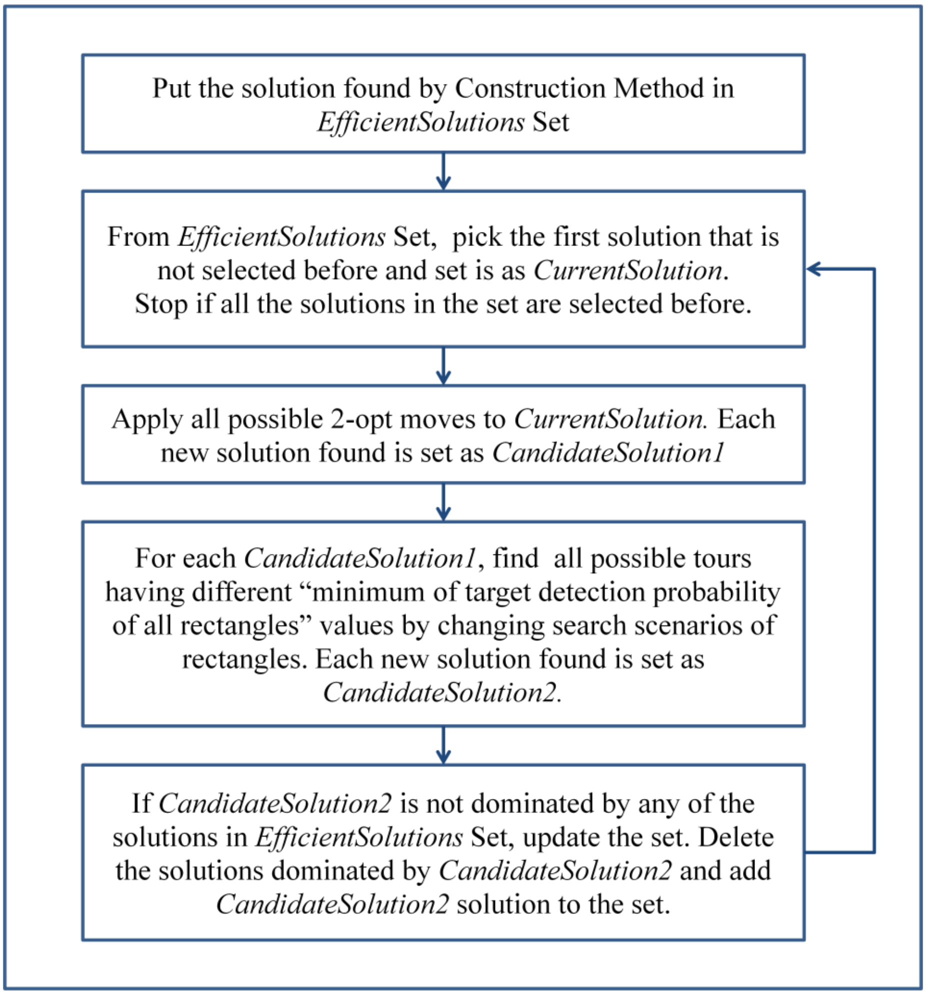

Figure 8 : Flow Chart for Improvement Method

Improvement algorithm is explained below:

Step 1 : Initial solution found by the Construction Method is added to *EfficientSolutions* Set.

Step 2 : The first solution that is not selected before in *EfficientSolutions* is selected and set as *CurrentSolution*. If the all solutions in the set are selected before, algorithm finishes.

Step 3 : Apply 2-opt move.

All possible 2-opt moves are applied on *CurrentSolution*. Each new solution found is named as *CandidateSolution1*.

In 2-opt move, two rectangles are selected as Rectangle 1 and Rectangle 3. In the tour, next rectangles of Rectangle 1 and Rectangle 3 are named as Rectangle 2 and Rectangle 4, respectively. The edges between Rectangle 1 - Rectangle 2 and Rectangle 3 - Rectangle 4 are broken and the edges between Rectangle 1 - Rectangle 3 and Rectangle 2 - Rectangle 4 are added. In this move, Search Scenarios (entry point, exit point and search pattern) of all rectangles stay same.

Figure 9 and Figure 10 are given to explain 2-opt move.

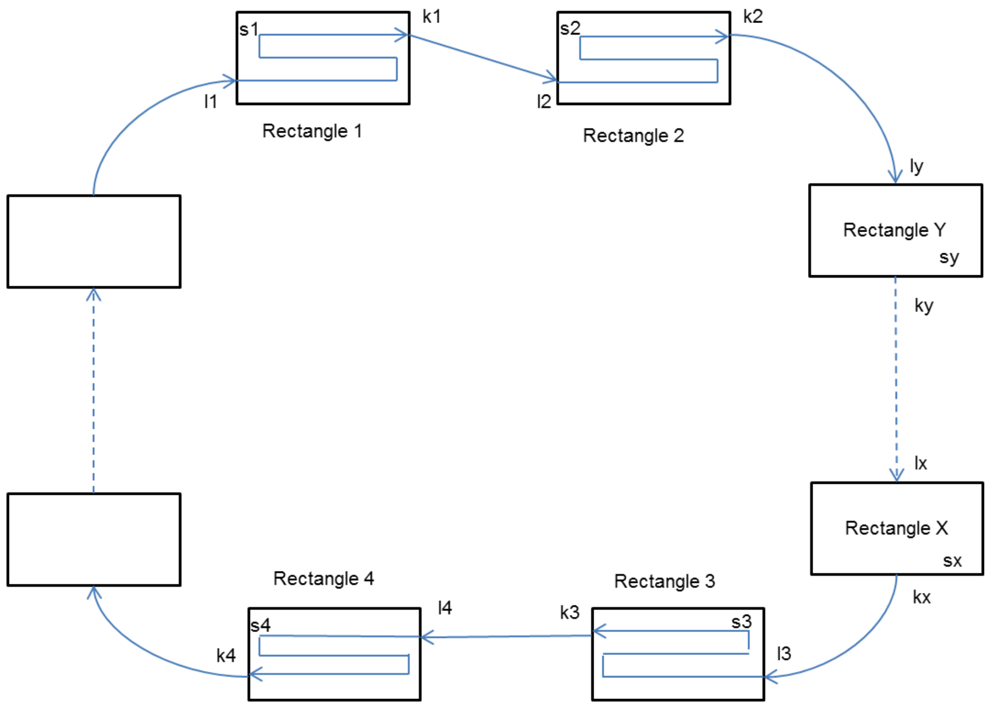

Figure 9 : Representation of a tour before 2-opt move

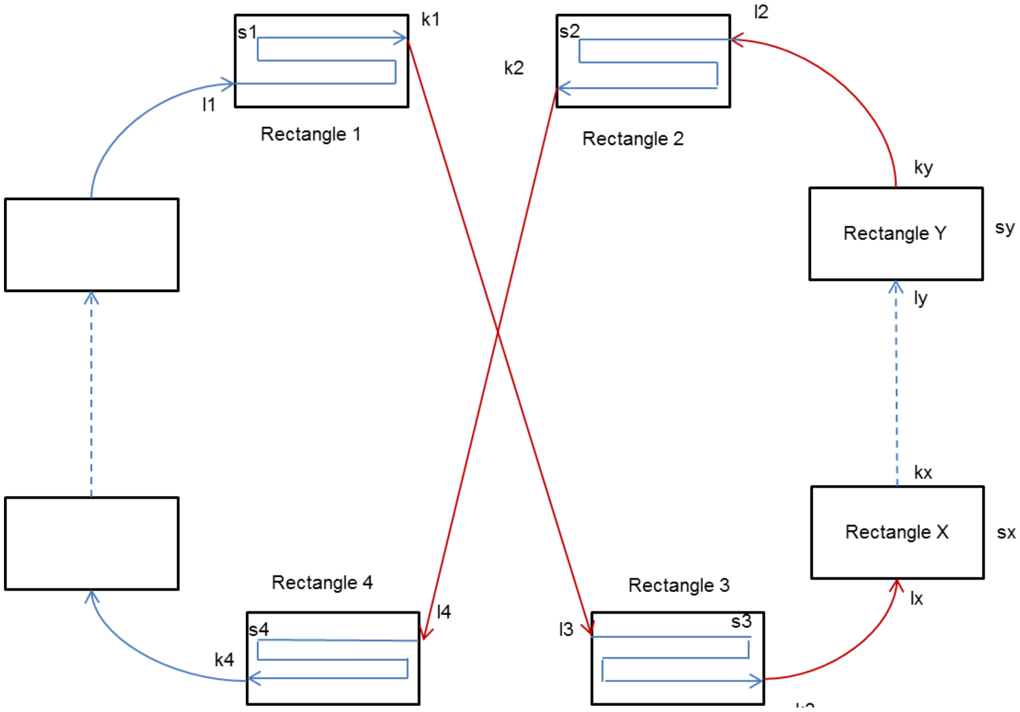

Figure 10 : Representation of a tour after 2-opt move

Step 4 : Change search scenarios in the rectangles.

This step is applied for each *CandidateSolution1* found in Step 3. The aim is to explore the neighborhood of *CandidateSolution1* by changing the Search Scenarios (Entry Point, Exit Point and Search Pattern) of rectangles. Each solution generated by changing the Search Scenarios of rectangles in *CandidateSolution1* is called as *CandidateSolution2.* In this step, the sequence of rectangles are not changed, in other words, it is kept same as in the *CandidateSolution1*.

For a fixed sequence of rectangles, a set of solutions having different "minimum of target detection probability of all rectangles" values is generated.

Probability is used to control the target detection probabilities of the selected Search Scenarios for rectangles. This step is repeated for each possible *Probability* value. Starting from "Minimum acceptable probability of target detection" value, *Probability* is increased iteratively until it reaches its maximum value.

For each rectangle, Search Scenario that minimizes the distance travelled in the path starting from previous rectangle's exit point and ending in next rectangle's entry point is selected. It is ensured that the selected Search Scenario's target detection probability is larger than *Probability* value. For each *Probability* value, a new *CandidateSolution2* is found.

Step 5 : Update *EfficientSolutions* set.

Each *CandidateSolution2* found in Step 4 is compared with the solutions in *EfficientSolutions* Set to check nondominance relation. If none of the solutions in *EfficientSolutions* Set dominates *CandidateSolution2*, *CandidateSolution2* is added to *EfficientSolutions* Set and the solutions dominated by *CandidateSolution2* are deleted from the set. If a solution that dominates *CandidateSolution2* is found, any update in *EfficientSolutions* is not made. Then, search continues with Step 2 where a new *CurrentSolution* is selected.

Detailed algorithm of Improvement Method is given in APPPENDIX B.

Combination of Construction Method 1 and Improvement Method is called as Heuristic Method 1 (HM1). Combination of Construction Method 2 and Improvement Method is called as Heuristic Method 2 (HM2). Combination of Construction Method 3 and Improvement Method is called as Heuristic Method 3 (HM3).

CHAPTER 5

COMPUTATIONAL RESULTS

In previous chapter, an exact method and a heuristic method with 3 different construction steps are introduced. As MASP is a new problem, there are no previous results reported that can be used to compare the results and the performance of the proposed methods. Heuristic solutions are compared with optimal solutions found by ε-MASP. Moreover, performance of the heuristic methods is compared with each other.

In Section 5.1, performance measures that are used to evaluate the performance of the methods are introduced. Then, generation of the test problems are explained in Section 5.2. In Section 3, the results are given and discussed.

5.1 Performance Measures

To measure the performance of the proposed heuristic methods, some performance metrics are used. Simple performance measures that are used for each feasible solution are given below:

DT: Acronym for DistanceTravelled, which is total distance travelled in the tour (including the distance travelled between rectangles and in rectangles).

MDP: Acronym for MinDetectionProbabillity, which is minimum of the target detection probabilities for all rectangles based on selected search scenarios.

%GAP: Percentage deviation of the DT found by HM1 or HM2 from optimal DT of corresponding MDP.

In multiobjective optimization, there are two different goals for generating the pareto optimal set:

1. Minimizing the distance of the generated solutions from pareto optimal solutions. (Convergence)
2. Maximizing the diversity of the generated solutions on pareto front. (Diversity)

There are many performance metrics to quantify the performance of multiobjective evolutionary algorithms. Some of them are specific to measure one of the goals. However, some of them are used to measure both convergence and diversity in one measure.

From the literature, Generational Distance (GD) is selected to measure convergence. Additionally, Hypervolume (HV) is used to measure both diversity and convergence simultaneously. Total run time of the methods is also presented to quantify the performance of the methods.

Generational Distance (GD)

GD is used to calculate the average distance of solutions generated by heuristic methods from their corresponding optimal solutions (Deb, 2001). Generally, Euclidian distance is used to calculate the distances between solutions. As the ranges of the objectives are different, aggregating the objectives in one distance measure can cause loss of information. Therefore, Percent GD metric is used to measure average distance in each objective separately (Karademir, 2008). The metrics are given below:

$$GD_{dist}^{\%} = \frac{\sum_{t=1}^{|Q|} \frac{|z_t^{dist} - z_t^{dist*}|}{z_t^{dist*}}}{|Q|} * 100$$

$$GD_{prob}^{\%} = \frac{\sum_{t=1}^{|Q|} \frac{|z_t^{prob} - z_t^{prob*}|}{z_t^{prob*}}}{|Q|} * 100$$

$GD_{dist}^{\%}$: average percentage deviation in "minimizing the distance travelled" objective.

$GD_{prob}^{\%}$: average percentage deviation in "maximizing the minimum of target detection probabilities of all rectangles" objective.

z_t^{dist*} : total distance travelled in optimal solution t.

z_t^{dist} : total distance travelled in heuristic solution t corresponding to z_t^{dist*}.

z_t^{prob*} : minimum of target detection probabilities of all rectangles in optimal solution t.

z_t^{prob} : minimum of target detection probabilities of all rectangles in heuristic solution t corresponding to z_t^{prob*}.

Hypervolume

Hypervolume is used to evaluate convergence and diversity together. It calculates the volume covered by the solutions generated in objective space. Volume constituted by each solution is calculated with respect to a reference point. This reference point is nadir point. Nadir point is the point in objective space, which is a vector of the worst feasible objective values. HV is calculated as follows: (Deb, 2001)

$$HV = \sum_{t=1}^{|Q|} \text{Volume}(t)$$

Figure 11 shows the hypervolume enclosed by solutions generated for a biobjective problem in which objective 1 is of minimization type and objective 2 is of maximization type as in MASP.

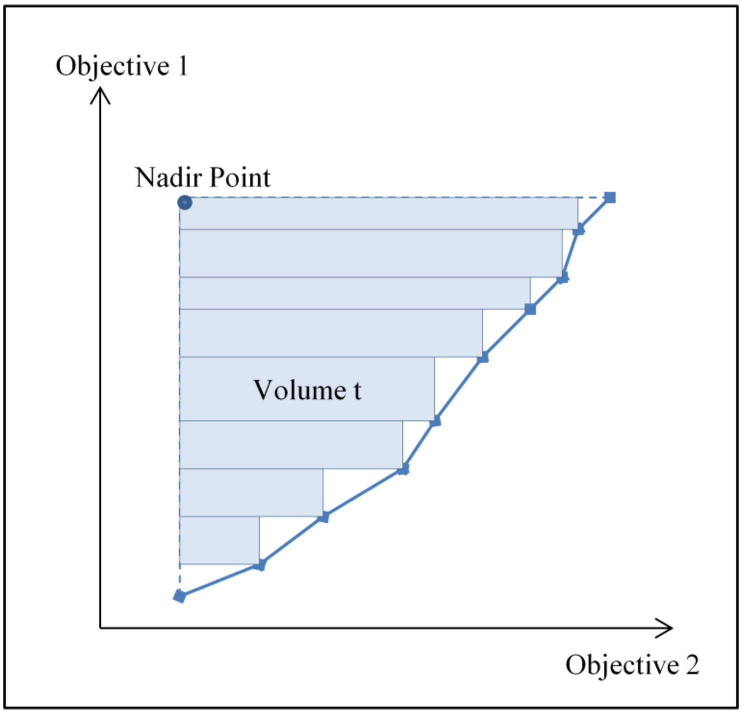

Figure 11 : An example representation of Hypervolume

To eliminate the effect of the scales of the objectives on the measure, measure is normalized and Hypervolume Ratio (HVR) metric is used. HVR is calculated as follows: (Deb, 2001)

$$HVR = \frac{HV(Q)}{HV(P)}$$

P: Set of optimal solutions.

Q: Set of solutions found by heuristic methods.

HV(Q) : Hypervolume enclosed by Q.

HV(P) : Hypervolume enclosed by P.

If HVR is closer to 1, q is closer to the pareto optimal set.

5.2 Problem Generation

A set of test problems are needed to evaluate the results of the proposed methods. For that purpose, test set generated by Karasakal (2013) is used.

60 disjoint rectangles are drawn randomly on a plane. 5, 10, 20, 30 and 40 rectangles are selected randomly from these 60 rectangles. Two values, 0.25 and 0.35, are used for sweep width. Three values, 0.30, 0.40 and 0.50, are used for minimum acceptable target detection probability. Each combination of these settings constitutes a problem instance. Therefore, 5*2*3=30 different problems are solved and their solutions are evaluated. The parameter setting for the problems are given in Table 2.

5.3 Results

Firstly, we solved the test problems optimally by using ε-MASP. ε-MASP is written and solved optimally in GAMS (Brook et al., 1996) utilizing CPLEX MIP solver with zero absolute and relative gap. Heuristic Methods are written in C Programming Language. In construction method of HM1, Concorde TSP Solver is used. In construction method of HM2, GAMS is used. Runs are performed using a computer with Intel Core i5 2.30 GHz processor and 6 GB of RAM.

As an example, the results of a test problem found by HM1 and HM2 are given in Table 3 and Figure 12. This test problem is Ins2_10 with 10 rectangles, 0.4 minimum acceptable target detection probability and 0.25 sweep width.

Table 2: Parameter Settings for Test Problems

Problem Name	Number of Rectangles	Minimum Acceptable Target Detection Probability	Sweep Width
Ins1_5	5	0.50	0.25
Ins1_10	10	0.50	0.25
Ins1_20	20	0.50	0.25
Ins1_30	30	0.50	0.25
Ins1_40	40	0.50	0.25
Ins2_5	5	0.40	0.25
Ins2_10	10	0.40	0.25
Ins2_20	20	0.40	0.25
Ins2_30	30	0.40	0.25
Ins2_40	40	0.40	0.25
Ins3_5	5	0.30	0.25
Ins3_10	10	0.30	0.25
Ins3_20	20	0.30	0.25
Ins3_30	30	0.30	0.25
Ins3_40	40	0.30	0.25
Ins4_5	5	0.50	0.35
Ins4_10	10	0.50	0.35
Ins4_20	20	0.50	0.35
Ins4_30	30	0.50	0.35
Ins4_40	40	0.50	0.35
Ins5_5	5	0.40	0.35
Ins5_10	10	0.40	0.35
Ins5_20	20	0.40	0.35
Ins5_30	30	0.40	0.35
Ins5_40	40	0.40	0.35
Ins6_5	5	0.30	0.35
Ins6_10	10	0.30	0.35
Ins6_20	20	0.30	0.35
Ins6_30	30	0.30	0.35
Ins6_40	40	0.30	0.35

Table 3 : Results for problem Ins2_10

ε -MASP		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
149,25	0,4052	152,28	0,4052	2,03	149,25	0,4052	0,00
151,30	0,4092	152,44	0,4171	0,57	151,30	0,4092	0,00
151,58	0,4171	156,23	0,4262	1,09	151,58	0,4171	0,00
154,55	0,4262	159,55	0,4447	0,36	154,55	0,4262	0,00
158,98	0,4447	161,46	0,4450	1,05	159,19	0,4447	0,13
159,78	0,4450	161,60	0,4512	0,58	159,78	0,4450	0,00
160,66	0,4512	165,47	0,4541	0,00	160,66	0,4512	0,00
165,47	0,4541	165,98	0,4545	0,00	167,61	0,4700	0,70
165,98	0,4545	166,45	0,4700	0,00	170,46	0,4715	0,76
166,45	0,4700	169,17	0,4715	0,00	171,08	0,4776	0,61
169,17	0,4715	170,04	0,4776	0,00	173,43	0,4866	0,27
170,04	0,4776	172,97	0,4866	0,00	173,77	0,4969	0,00
172,97	0,4866	173,92	0,4969	0,09	174,65	0,5034	0,39
173,77	0,4969	173,97	0,5034	0,00	176,37	0,5105	0,00
173,97	0,5034	177,47	0,5105	0,62	177,73	0,5209	0,00
176,37	0,5105	179,22	0,5209	0,83	178,37	0,5276	0,00
177,73	0,5209	181,77	0,5276	1,90	180,59	0,5312	0,00
178,37	0,5276	184,66	0,5312	2,26	182,89	0,5406	0,00
180,59	0,5312	186,79	0,5406	2,13	183,81	0,5412	0,00
182,89	0,5406	187,55	0,5412	2,03	186,34	0,5439	0,00
183,81	0,5412	189,01	0,5439	1,43	187,23	0,5459	0,00
186,34	0,5439	189,06	0,5459	0,98	187,56	0,5478	0,00
187,23	0,5459	189,47	0,5478	1,02	190,29	0,5507	0,00
187,56	0,5478	192,20	0,5507	1,00	196,17	0,5654	0,38
190,29	0,5507	195,43	0,5654	0,00	197,40	0,5657	0,42
195,43	0,5654	196,58	0,5657	0,00	200,36	0,5823	0,21
196,58	0,5657	199,94	0,5823	0,00	204,08	0,5862	*
199,94	0,5823	202,91	0,5889	0,00	204,51	0,5865	*
202,91	0,5889	203,89	0,5934	0,00	204,64	0,5889	0,85
203,89	0,5934	211,03	0,5966	0,95	204,98	0,5934	0,54
209,04	0,5966				209,04	0,5966	0,00

* For that solution, there is no corresponding optimal solution that has same MPD value. Therefore, %GAP cannot be calculated.

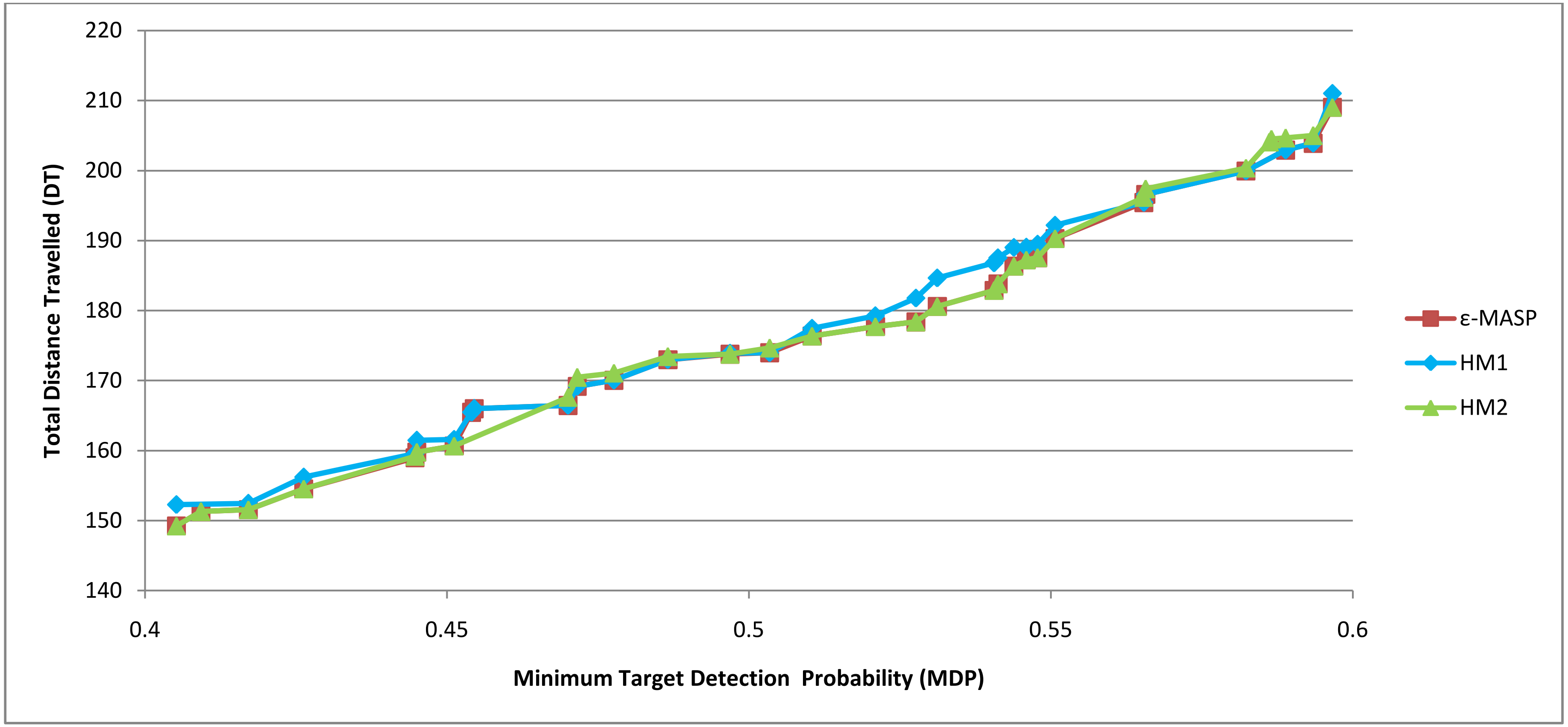

Figure 12: The objective space of Ins2_10 showing solution sets found

As can be seen from Table 3 and Figure 12, %GAP is below 2.5 % for all solutions. To see the differences of the solutions found by the methods HM1 and HM2 from the optimal solution found by ε-MASP, one of the solutions from efficient set is selected and represented graphically. The selected efficient solution of Ins2_10 is given in Table 4. We call this problem is as “sample problem”. The solutions in sample problem have same MDP value.

Table 4: Solutions of the Sample problem of Ins2_10

ε –MASP		**HM1**			**HM2**		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
158.98	0.4447	159.55	0.4447	0.36	159.19	0.4447	0.13

The solution found by ε-MASP for the sample problem is given in Figure 13.

The initial solution of Ins2_10 found by Construction Method 1 is given in Figure 14. This initial solution is improved by Improvement Method and the solution given in Figure 15 is found for sample problem.

The initial solution of Ins2_10 found by Construction Method 2 is given in Figure 16. This initial solution is improved by Improvement Method and the solution given in Figure 17 is found for sample problem.

The results for 30 test problems are given in APPENDIX C in figures. For each test problem, the figure includes the pareto optimal set found by ε-MASP and solution sets found by HM1 and HM2.

The results on test problems show that HM1 and HM2 produce high quality solutions. However, HM3 fails to produce results that enable us to produce pareto optimal solutions. The reason is that Construction Method 3 does not produce promising initial solutions. Construction Method 3 is proposed to eliminate the use of Concorde TSP Solver in HM1. It tries to find the sequence of rectangles that has minimum length by generating a specified number of random sequences of rectangles. However, the solutions generated randomly are not close to the solutions generated by Construction Method 1.

As an example, for the problems of Instance 1, the initial solution found by Construction Method 1 and the initial solutions found by Construction Method 3 for different number of random sequences are given in Table 5. As can be seen from the results given, there is large gap between the initial solutions found by Construction Method 1 and Construction Method 3. Moreover, by relying on the runs performed on test problems, it is observed that HM3 does not give results that can be used to solve MASP. Therefore, the results of HM3 are not included in the rest of the computational results.

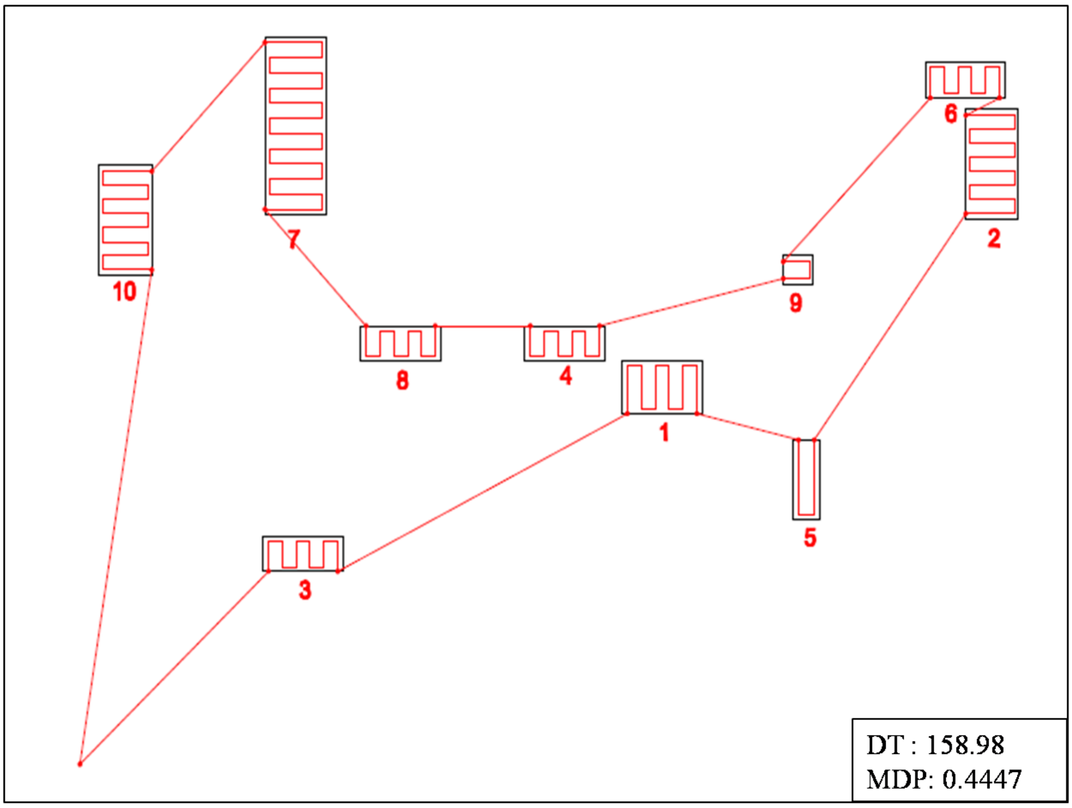

Figure 13: The optimal solution of sample problem found by ε-MASP.

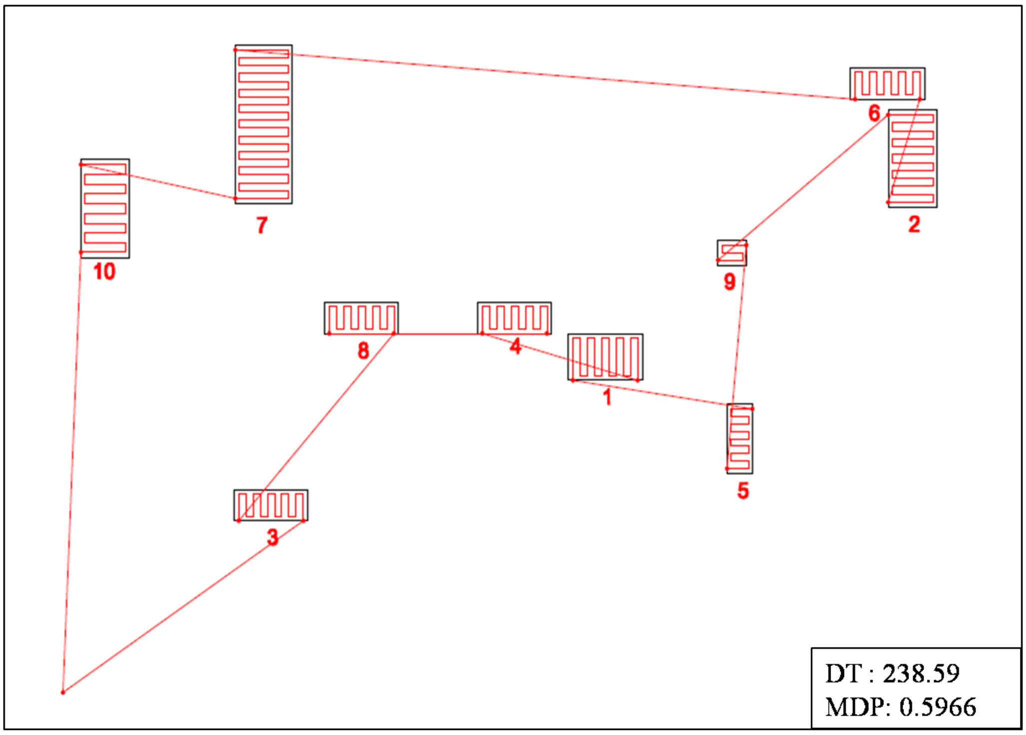

Figure 14: The initial solution of sample problem found by Construction Method 1

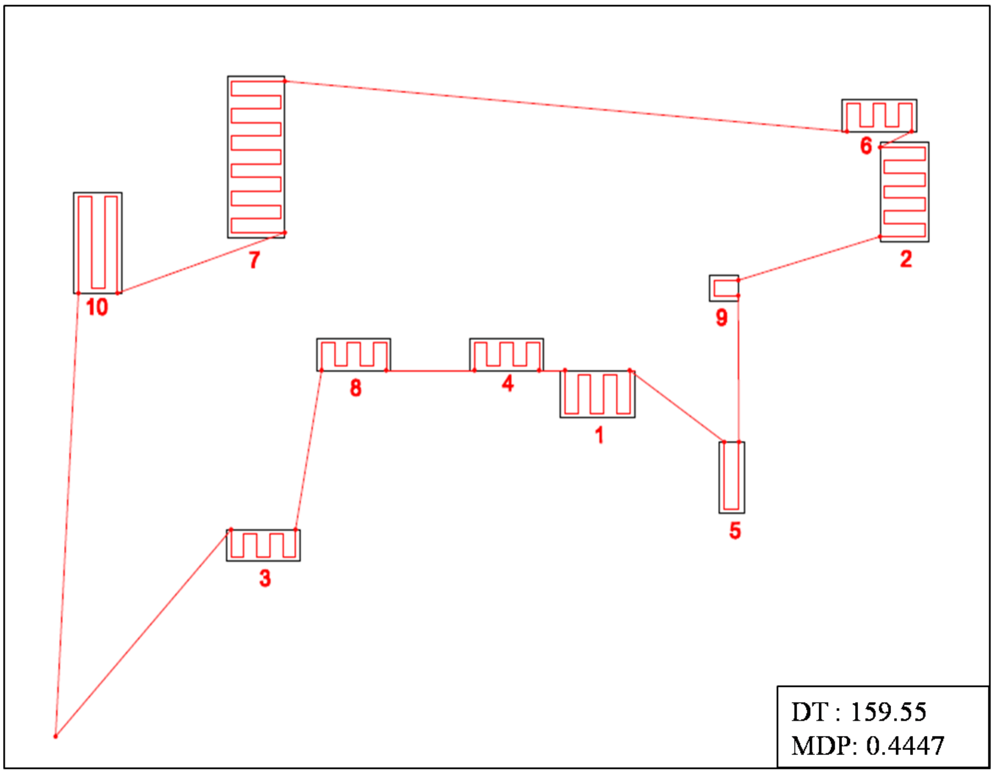

Figure 15: The solution found for sample problem by HM1.

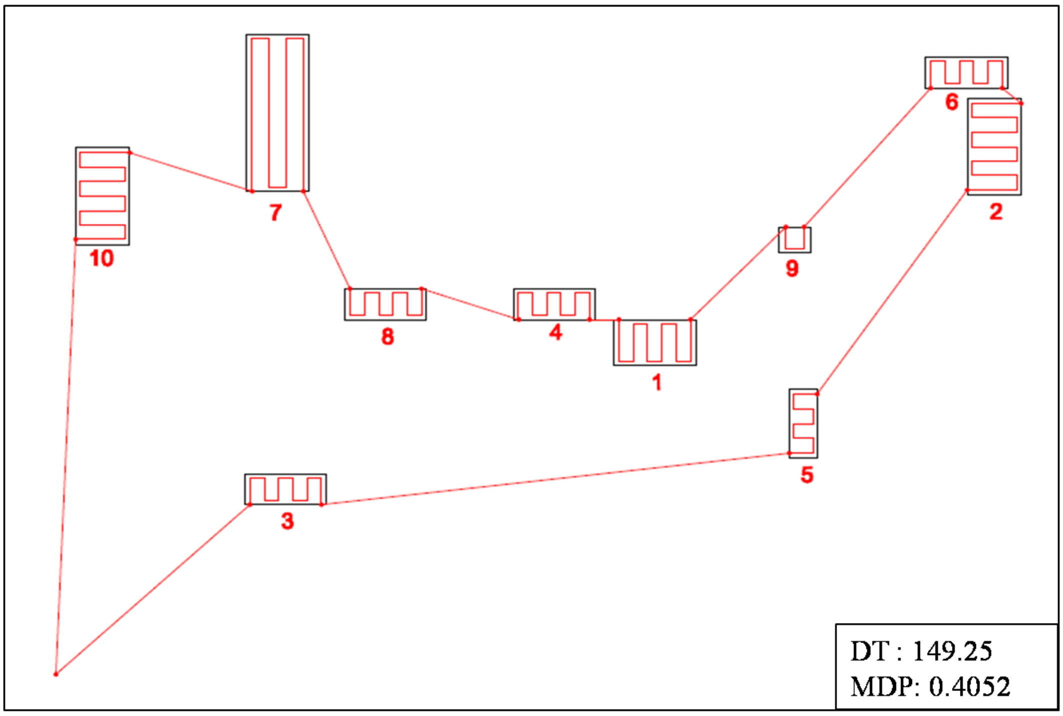

Figure 16 : The initial solution of sample problem found by Construction Method 2.

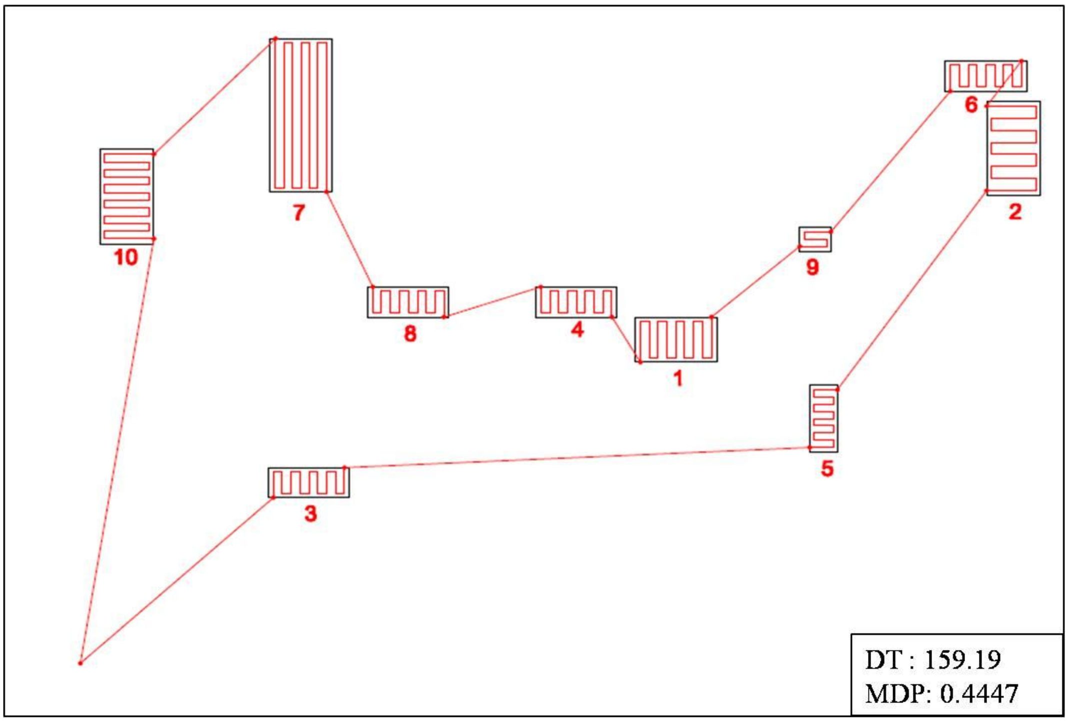

Figure 17: The solution found for sample problem by HM2.

Table 5: DT values of initial solutions of problems in Instance 1 found by Construction Method 1 and 3.

Rectangle No		Construction Method 1*	Construction Method 3*					
			Number of Random Solutions					
			100	500	1,000	5,000	10,000	100,000
	5	128.415	128.415	128.415	128.415	128.415	128.415	128.415
	10	238.769	254.424	245.285	245.285	244.309	244.309	244.309
	20	394.216	530.691	527.355	513.527	493.828	493.828	480.945
	30	577.685	781.328	781.328	781.328	781.328	764.991	733.031
	40	795.095	1131.679	1115.704	1106.150	1090.123	1075.968	1050.315

* MPD value of all initial solutions given in the table is 0.5966

To measure the performance of the proposed approaches, the performance measures defined in Section 5.1 are used. Only 3 of the test problems with 40 rectangles cannot be solved optimally due to resource limitation of the computer used. The results of performance measures do not include the problems that cannot be solved optimally, which are Ins3_40, Ins5_40 and Ins6_40. These problems are examined separately. For each number of rectangles (5, 10, 20, 30 and 40), average of all performance measures is calculated in Table 6, Table 7 and Table 9. Average values in performance measures are calculated by averaging the performance measures of 6 instances with same number of rectangles and the problems that are not solved optimally are not included.

In Table 6, for the test problems with different number of rectangles, average computation times are given. Computation times for all test problems are given in Appendix D. According to the results, it can be concluded that ε-MASP is solved in higher computation times compared to HM1 and HM2. For real life applications, ε-MASP can be used for the problems with 10 or less rectangles. For the cases with more rectangles, computation times of ε-MASP are not suitable to be used periodically. The computation times for HM2 include the time spent for finding the initial solution with GAMS. Finding initial solution by Construction Method 2 consumes approximately 0.70% of the total computation time of HM2. In HM1, applying Construction Method 1 with Concorde TSP Solver takes only a few seconds. Therefore, HM2 has higher computation times than HM1.

Table 6: Average computation times for test problems with different number of rectangles

Rectangle No	ε –MASP Computation Time (sec)	HM1 Computation Time (sec)	HM2 Computation Time (sec)
5	304.49	2.70	5.40
10	3,590.79	4.10	72.32
20	2,000.35	23.29	105.99
30	44,195.83	104.94	2,541.39
40	74,586.63	355.47	2,029.02

HM1 and HM2 are able to find approximately all MDP values on pareto optimal set. Deviation occurs in DT values for corresponding MDPs. Therefore, for Percent GD metric, only average percentage deviation in “minimizing the distance travelled” objective is examined, $GD^{\%}_{dist}$. In Table 7, firstly average Percent GD of the problems with different number of rectangles is given. It is observed that both HM1 and HM2 results in average Percentage GD’s lower than 2.5%. For the results of average percentage GD, it is not possible to conclude that one heuristic is superior to the other. Moreover, in Table 7 total number of solutions found and total number of optimal solutions found is given. According to the reported results on test problems, HM2 finds more optimal solutions than HM1. The results for all problems are given in Appendix E.

Table 7 : Average Percent GD, total number of solutions found and total number of optimal solutions found for test problems with different number of rectangles

Rectangle No	HM1			HM2		
	$GD^{\%}_{dist}$	Total # of solutions found	Total # of optimal solutions found	$GD^{\%}_{dist}$	Total # of solutions found	Total # of optimal solutions found
5	0.00	87	83	0.00	87	83
10	0.72	134	40	0.35	134	74
20	0.03	206	177	0.02	205	179
30	1.65	232	13	0.60	230	55
40	0.98	132	0	1.30	136	14

To further analyze the performance of HM1 and HM2 in terms of deviation from optimal solution, we examine how close the solutions found by heuristic methods to the optimal solutions. Therefore, in Table 8, total number of solutions, found by HM1 and HM2 for all instances, whose %GAP is in specified intervals is given. As can be seen from the table, there is only one solution whose %GAP is higher than 4%. Moreover, 99% of the solutions found have %GAP values that are lower than %3. In terms of performance, although HM2 does generally better than HM1, it is not possible conclude that one heuristic dominates the other. The results for all problems are given in Appendix F.

In Table 9, Average Hypervolume and Hypervolume Ratio for test problems with different number of rectangles are given. HM1 and HM2 have HV and HVR values that are very close to each other. As all HVR values of the heuristic methods are very close to or equal to 1, we can conclude that both heuristics find solutions that are very close to Pareto Optimal Set. The results for all problems are given in Appendix G.

Table 8: Total number of solutions found whose %GAP is in a specified interval

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
5	HM1	87	83	4	0	0	0	0
	HM2	87	83	4	0	0	0	0
10	HM1	134	40	52	32	10	0	0
	HM2	134	74	49	8	3	0	0
20	HM1	206	177	29	0	0	0	0
	HM2	205	179	26	0	0	0	0
30	HM1	232	13	35	114	48	21	1
	HM2	230	55	119	52	4	0	0
40	HM1	132	0	87	45	0	0	0
	HM2	136	14	61	36	21	4	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)
B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)
C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)
D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)
E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 9 : Average Hypervolume and Hypervolume Ratio for test problems with different number of rectangles

Rectangle No	Average HV		Average HVR	
	HM1	HM2	HM1	HM2
5	2.39	2.39	1.00	1.00
10	5.03	5.10	0.95	0.97
20	10.96	10.96	1.00	1.00
30	16.12	16.79	0.90	0.96
40	11.34	11.27	0.91	0.88

Three of the test problems with 40 rectangles (Ins3_40, Ins5_40 and Ins6_40) cannot be solved optimally. These problems are solved in GAMS with 5% relative gap and compared with the results of heuristic methods. It is observed that both of the heuristic methods find solutions that are equal to and better than the solutions found with 5% relative gap. Therefore, for those problems, HM1 and HM2 produce results which have at most 5% percent deviation from optimal. The results for those problems are given in Appendix H in Table 26, Table 27 and Table 28.

According to the results reported, HM2 does generally better than HM1 in terms of solutions. However, the results do not warrant us to claim dominance of HM2 over HM1. The only significant difference between them is computation times. As HM2 has higher computation times, HM1 can be preferred over HM2.

CHAPTER 6

INTERACTIVE SOLUTION PROCEDURE

In multiobjective optimization problems, finding and presenting the pareto optimal set to decision maker (DM) is not the final step. After that, DM has to make a selection from the pareto optimal set. To aid DM in selection, an interactive solution procedure is proposed in this chapter.

The proposed procedure relies on pairwise comparisons of the solutions in the pareto optimal set. Firstly, whole pareto optimal set is divided into groups. This divison induces partitioning of decision space into subspaces. These subspaces are called *cell*s (Köksalan et al., 1995). The number of cells changes between 2 and 9 for the solution sets with different number of solutions. The reason for limiting the number of cells with 9 is that the experiments have shown that individuals can compare at most nine pieces of information simultaneously (Miller, 1956). For each cell, ideal point of the *cell* (*cell ideal*) is presented to DM. After that, DM makes a selection form the *cell ideal*s. Then, the *cell* corresponding to the selected *cell ideal* is scrutinized. The solutions in the selected *cell* are shown on two different graphs to present the tradeoff between the objectives to DM. The first graph is drawn by showing one of the objectives in right axis and the other in left axis. The second graph is same as the first graph, except in the second graph normalized objective values are used to show the DM marginal gain and loss more clearly. Normalization is done by setting maximum objective value to 1 and minimum objective value to 0 and scaling the other objective values between 0 and 1. After examining the graphs, DM is expected to make a selection from the solutions shown. The selected solution and its adjacent solutions are presented to DM and she/he makes another preference between these 3 solutions. If DM is satisfied with the solution, algorithm stops. If not, DM can choose to search the adjacent solutions of the last selected solution or to change the *cell* that is selected at the beginning of the algorithm. Algorithm continues in this way, until DM is satisfied with the solution.

The proposed algorithm is given below:

Step 0: Initialization.

a. Rank the solution set according to one of the objectives in ascending order.

b. Divide the solution set into cells. In the following table, for solution sets with different number of solutions the number of cells to be divided is given.

Table 10: Number of Cells to be divided

Number of solutions in the solution set (|Q|)	Number of Cells
$0 < \|Q\| \leq 10$	2
$10 < \|Q\| \leq 20$	3
$20 < \|Q\| \leq 30$	4
$30 < \|Q\| \leq 40$	5
$40 < \|Q\| \leq 50$	6
$50 < \|Q\| \leq 60$	7
$60 < \|Q\| \leq 70$	8
$70 < \|Q\|$	9

c. Find the ideal points of the solutions in all of the *cell*s. Set these solutions as *cell ideal*s.

Step 1: Ask Decision Maker to select one of the *cell ideal*s. Remove the selected *cell ideal* from the *cell ideals list*.

Step 2: Present the solutions in the *cell* corresponding to the selected *cell ideal* in Step 1 with two different graphs. The first graph is drawn by showing one of the objectives (DT) in right axis and the other (MDP) in left axis. The second graph uses normalized values of the objectives and drawn by showing one of the normalized objectives in right axis and the other in left axis.

Step 3: Ask Decision Maker to select a solution shown on the graphs. Set the selected solution as *SelectedSolution1*.

Step 4: Show the objective function values of *SelectedSolution1* and its adjacent solutions to Decision Maker. Set these 3 solutions as *CandidateSolutions*.

Step 5: Ask Decision Maker to make a preference between *CandidateSolutions*. Set the selected solution as *SelectedSolution2.*

If Decision Maker is satisfied with *SelectedSolution2*, STOP. Otherwise, go to Step 6.

Step 6: If *SelectedSolution1* and *SelectedSolution2* are the same solution, go to Step 1.

Else, ask Decision Maker to select searching a new cell or the adjacent solutions of *SelectedSolution2.*

If Decision Maker selects searching a new *cell* go to Step1,

Otherwise set *SelectedSolution1=SelectedSolution2* and go to Step 4.

The proposed algorithm is applied on two examples. The results of HM1on the test problems Ins2_5 and Ins2_40 are used as examples.

Example 1: Application of the algorithm on the results of HM1on Ins2_5

Step 0: Initialization.

The solution set is ranked according to DT values in ascending order and divided into 3 *cell*s as |Q|=18.

*Cell ideal*s are presented to Decision Maker as given in Table 11.

Table 11: *Cell ideal*s for *cell*s of the results of HM1on Ins2_5

Cell	*Cell Ideal*	
	DT	MDP
1	107.06	0.4866
2	107.06	0.5507
3	115.85	0.6142

Step 1: Decision Maker selects the 2^{nd} *cell ideal*.

Step 2: The solutions in the 2^{nd} *cell* are presented to Decision Maker by Figure 18 and Figure 19.

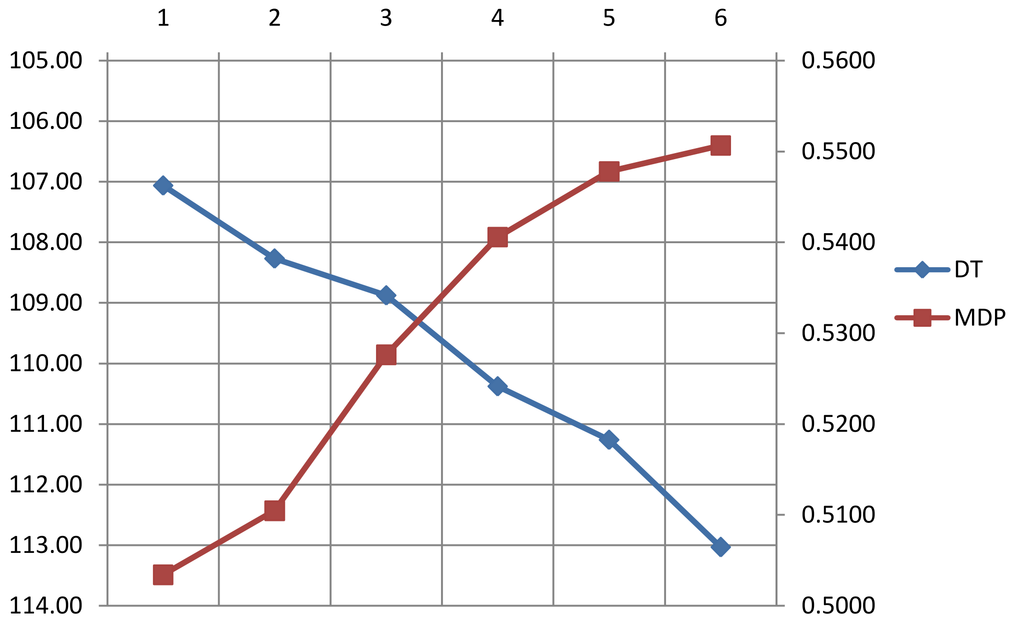

Figure 18 : Tradeoff between the objectives of solutions in 2^{nd} *cell* of the results of HM1on Ins2_5. (Numbers at the top of the graph shows each distinct solution within the selected cell.)

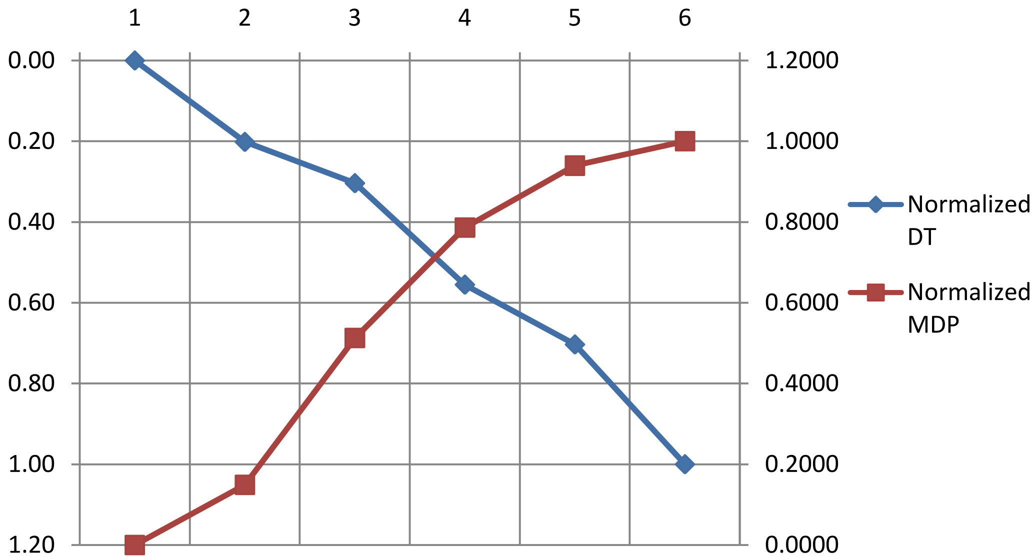

Figure 19 : Tradeoff between the normalized objectives of solutions in 2^{nd} *cell* of the results of HM1on Ins2_5

Step 3: Decision Maker selects the 5th solution. 5th solution is set as *SelectedSolution1*.

Step 4: *CandidateSolutions* are presented to Decision Maker as in Table 12.

Table 12 : *CandidateSolutions* selected from the results of HM1on Ins2_40

No	DT	MDP
1	110.38	0.5406
2	111.26	0.5478
3	113.03	0.5507

Step 5: Decision Maker selects the 3rd solution. The 3rd solution is set as *SelectedSolution2.*

Decision Maker is not satisfied with the solution.

Step 6: Decision Maker prefers to search a new *cell.*

Step 1: Decision Maker selects the 1st *cell ideal*.

Step 2: The solutions in the 1st *cell* are presented to Decision Maker by Figure 20 and Figure 21.

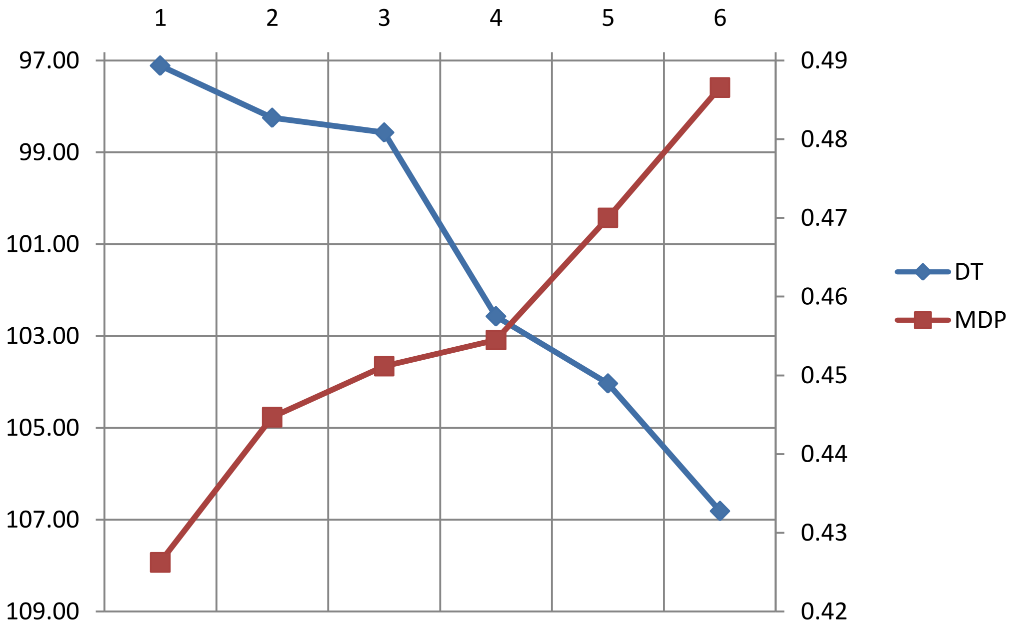

Figure 20 : Tradeoff between the objectives of solutions in 1st *cell* of the results of HM1on Ins2_5

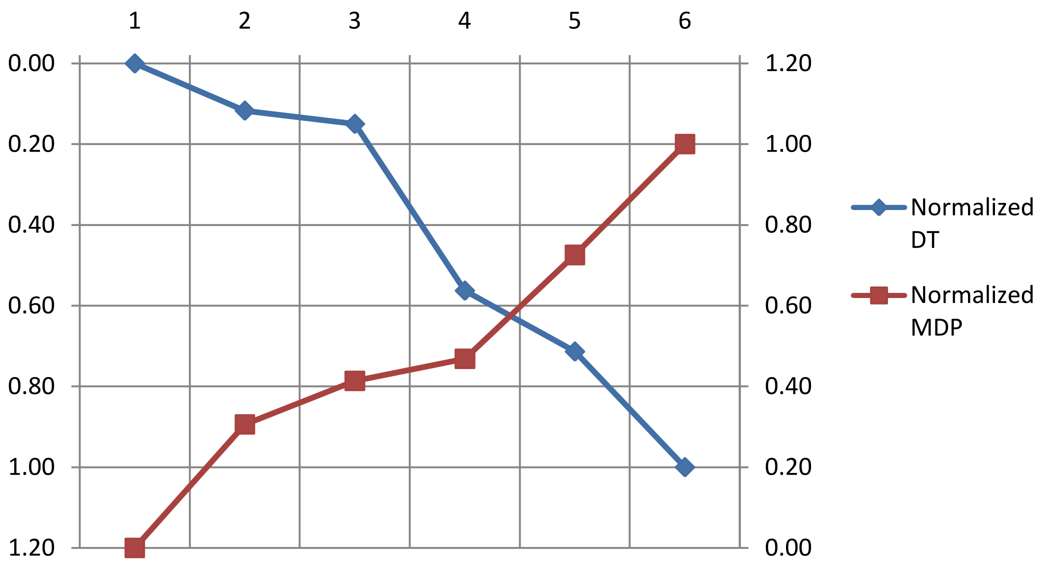

Figure 21 : Tradeoff between the normalized objectives of solutions in 1st *cell* of the results of HM1on Ins2_5

Step 3: Decision Maker selects the 3rd solution. 3rd solution is set as *SelectedSolution1*.

Step 4: *CandidateSolutions* are presented to Decision Maker as in Table 13.

Table 13 : *CandidateSolutions* selected from the results of HM1on Ins2_5

No	DT	MDP
1	98.25	0.4447
2	98.57	0.4512
3	102.57	0.4545

Step 5: Decision Maker selects the 2nd solution. The 2nd solution is set as *SelectedSolution2.*

Decision Maker is satisfied with the solution. Therefore, *SelectedSolution2* is the solution that is selected to be used.

Example 2: Application of the algorithm on the results of HM1on Ins2_40

Step 0: Initialization.

The solution set is ranked according to DT values in ascending order and divided into 9 *cell*s as |Q|=72.

*Cell ideal*s are presented to Decision Maker as given in Table 14.

Table 14: *Cell ideal*s for *cell*s of the results of HM1on Ins2_40

Cell	*Cell Ideal*	
	DT	MDP
1	448.76	0.4262
2	469.82	0.4475
3	481.76	0.4715
4	523.02	0.4996
5	548.74	0.5276
6	589.05	0.5439
7	604.37	0.5569
8	662.69	0.5823
9	689.43	0.5966

Step 1: Decision Maker selects the 3rd *cell ideal*.

Step 2: The solutions in the 3rd *cell* are presented to Decision Maker by Figure 22 and Figure 23.

Step 3: Decision Maker selects the 7th solution. 7th solution is set as *SelectedSolution1*.

Step 4: *CandidateSolutions* are presented to Decision Maker as in Table 15.

Table 15 : *CandidateSolutions* selected from the results of HM1on Ins2_40

No	DT	MDP
1	510.42	0.4596
2	510.90	0.4700
3	522.09	0.4715

Step 5: Decision Maker selects the 1st solution. The 1st solution is set as *SelectedSolution2.*

Decision Maker is not satisfied with the solution.

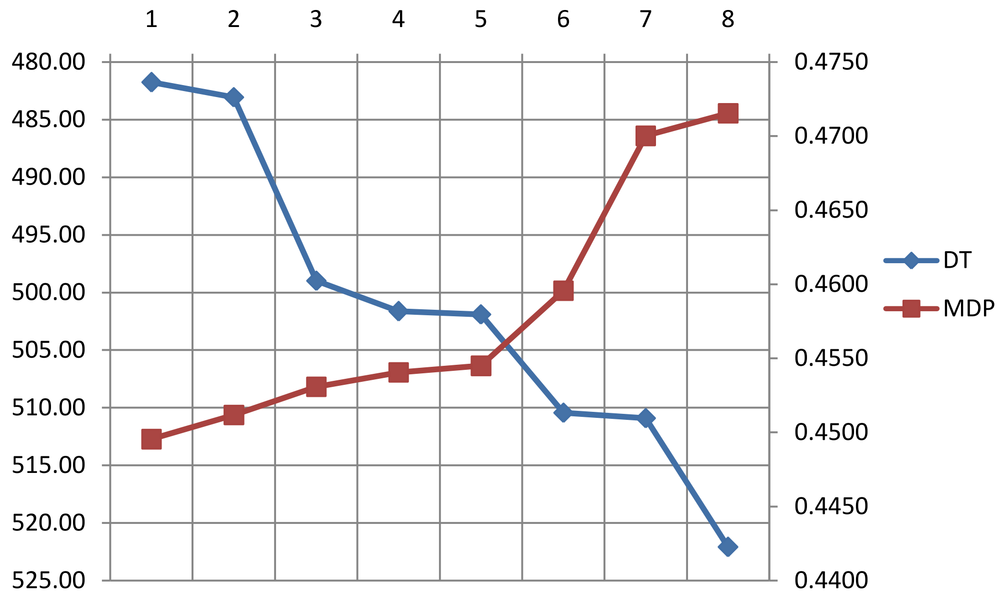

Figure 22 : Tradeoff between the objectives of solutions in 3rd *cell* of the results of HM1on Ins2_40

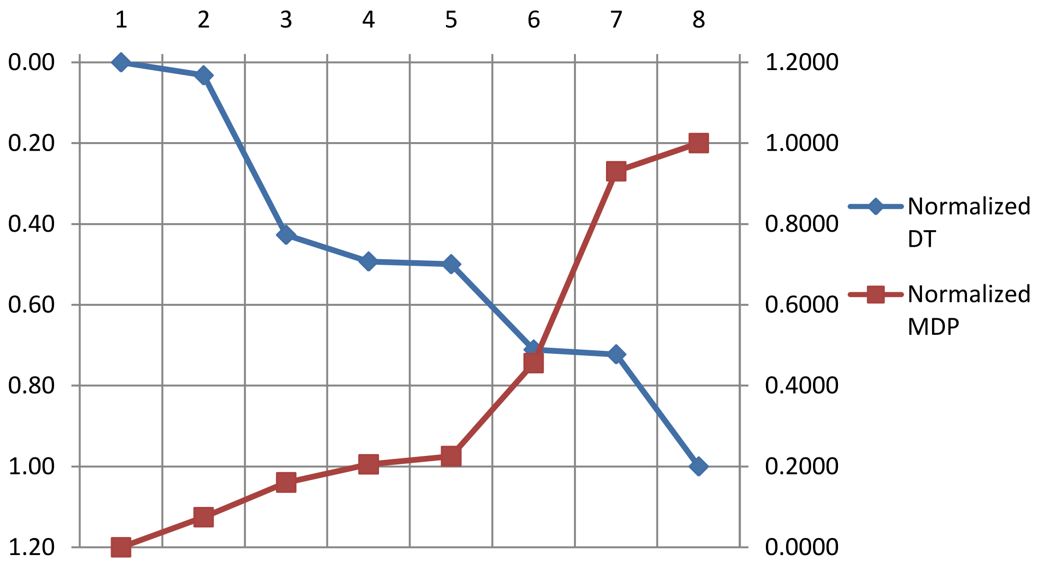

Figure 23 : Tradeoff between the normalized objectives of solutions in 3rd *cell* of the results of HM1on Ins2_40

Step 6: Decision Maker prefers to select searching the adjacent solutions of *SelectedSolution2.*

Step 4: *CandidateSolutions* are presented to Decision Maker as in Table 16.

Table 16 : *CandidateSolutions* selected from the results of HM1on Ins2_40

No	DT	MDP
1	501.90	0.4545
2	510.42	0.4596
3	510.90	0.4700

Step 5: Decision Maker selects the 1st solution. The 1st solution is set as *SelectedSolution2.*

Decision Maker is satisfied with the solution. Therefore, *SelectedSolution2* is the solution that is selected to be used.

CHAPTER 7

CONCLUSION

In this study, we consider a new problem in literature as Multiobjective Aerial Surveillance Problem. Although ASP is not well studied in literature, there are many civilian and military operations that ASP can be applied.

We study Multiobjective Aerial Surveillance Problem (MASP) with two objectives that are minimizing distance travelled and maximizing the minimum target detection probability of all rectangles. These two objectives are the most important two goals in the applications of ASP. We generate Pareto Optimal Set by using several methods. We use ε-MASP to solve MASP optimally. This method produced Pareto Optimal Set for all of the 30 test problems except 3 problems with 40 rectangles. However, ε-constraint method requires high computation times for problems with more than 10 rectangles. Therefore, heuristic methods are developed to cope with this disadvantage of ε-MASP. Three heuristic methods are introduced for solving MASP. The difference between the heuristic methods is in the construction of initial solution. After initial solution is found, all proposed heuristic methods use same improvement step. In construction step, HM1 uses optimal TSP sequence of the rectangles. HM2 uses an optimal solution of single objective ASP. HM3 produces random solutions and uses the best of them.

Our experimental results show that HM1 and HM2 find a set of solutions that are very close to the Pareto Optimal Set found by ε-MASP. On the other hand, it is observed that HM3 does not give good results and it is disregarded in the rest of the study. In terms of the performance measures calculated, HM1 and HM2 produce similar results. Both HM1 and HM2 find solutions with at most 2.5% deviation from optimal solution. The only significant difference between them is computation times. As HM2 has higher computation times, HM1 can be preferred over HM2.

Finally, we propose an interactive solution procedure based on pairwise comparisons for decision maker to give support in making a selection among the generated pareto optimal solutions.

In conclusion, in this study an exact method, two effective heuristic methods and an interactive solution procedure are proposed for MASP. To the best of our knowledge, our study is the first attempt to solve Multiobjective Aerial Surveillance Problem.

REFERENCES

Angel, E., Bampis, E., Gourvès, L. 2004. A Dynasearch Neighborhood for the Bicriteria Traveling Salesman Problem. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.): Metaheuristics for Multiobjective Optimization, Lecture Notes in Economics and Mathematical Systems 535. Springer-Verlag Berlin, 153-176.

Agrawal, R. 2011. Application of the Modified 2-opt and Jumping Gene Operators in Multi-Objective Genetic Algorithm to Solve MOTSP. Ph.D. Thesis, Cornell University of India, India, Noida.

Bazgan, C., Gourvès, L., Monnot, J., Pascual, F. 2012. Single approximation for biobjective max TSP. Lecture Notes in Computer Science, v 7164 LNCS, 49-62.

Bérubé, J.F., Gendreau, M., Potvin, J.Y. 2009. An exact e-constraint method for biobjective combinatorial optimization problems: Application to the traveling salesman problem with profits. European Journal of Operational Research 194, 39–50.

Brook A., Kendrik D. and Meeraus A. GAMS: A User's Guide. Massachusetts: Body & Fraser Publishing Company, 1996.

Cook, W. (2010). Concorde TSP solver. See: http://www. tsp. gatech. edu/concorde. html.

Deb K., 2001. Multiobjective Optimization using Evolutionary Algorithms. John Wiley & Sons LTD, 306-324.

Feillet,D., Dejax P., Gendreau, P. 2005. Traveling salesman problems with profits.Transportation Science 39.188–205.

Grob, M.J.H.B., 2006. Routing of platforms in a maritime surface surveillance operation. European Journal of Operational Research 170:613-628.
Hansen, M.P. 2000. Use of subsitute scalarizing functions to guide a local search base heuristics: the case of moTSP. Journal of Heuristics, 6:419–431.

Jacobson,S.H., McLay, L.A., Hall,S.N., Henderson, D.,Vaughan,D.E., 2006. Optimal search strategies using simultaneous generalized hill climbing algorithms. Mathematical and Computer Modelling 43:1061-1073.

Jaszkiewicz, A. 2002. Genetic local search for multiple objective combinatorial optimization. European Journal of Operational Research, 1(137):50–71.

Jaszkiewicz, A., Zielniewicz, P. 2009. Pareto memetic algorithm with path-relinking for biobjective traveling salesman problem. European Journal of Operational Research 193(3), 885–890.

Jozefowiez, N., Glover, F., Laguna, M. 2008. Multi-objective meta-heuristics for the traveling salesman problem with profits. Journal of Mathematical Modelling and Algorithms, 7(2), 177–195.

John,M.,Panton, D., White, K. ,2001. Mission planning for regional surveillance. Annals of Operations Research 108: 157-173.

Karasakal, O., 2013. Minisum and maximin aerial surveillance over disjoint rectangles. Submitted to Journal of the Operational Research Society.

Keller, C.P., Goodchild, M. 1988. The multiobjective vending problem: A generalization of the traveling salesman problem. Environment and Planning B: Planning and Design 15, 447–460.

Koester RJ, Cooper DC, Frost JR, et al: Sweep width estimation for ground search and rescue, Prepared for the National Search and Rescue Committee, USCG Contract No. DTCG32-02-D-R00010, Alexandria, Va, 2004, Potomac Management Group, Inc.

Köksalan, M., and. Sagala, P.N.S Interactive approaches for discrete alternative multiple criteria decision making with monotone utility functions. Management Science, 41(7):1158-1171, 1995

Kumar, R., Singh, P.K. 2007. Pareto evolutionary algorithm hybridized with local search for biobjective TSP. In: Grosan, C., Abraham, A., Ishibuchi, H. (eds.) Hybrid Evolutionary Algorithms, ch. 14. Springer, Heidelberg.

Li,W. 2005. Finding Pareto-optimal set by merging attractors for a bi-objective traveling salesmen problem. In: Coello Coello, C.A., Hern´andez Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 797–810. Springer, Heidelberg.

Li, W., Feng, M. 2012. A Parallel Procedure for Dynamic Multi-objective TSP. 10th IEEE International Symposium on Parallel and Distributed Processing with Applications.

Lust, T., Teghem, J. 2010. The multiobjective traveling salesman problem: a survey and a new approach, in: Advances in Multi-Objective Nature Inspired Computing, in: SCI, vol. 272, 119–141.

Manthey, B., 2009. Multi-criteria TSP: Min and max combined. In Proceedings of the 7th International Workshop on Approximation and Online Algorithms (WAOA 2009), LNCS 5893, pages 205–216.

Mavrotas, G., 2009. Effective implementation of the e-constraint method in multi-objective mathematical programming problems. Applied Mathematics and Computation 213 (2), 455–465.

Miller, G.A. (1956). The Magical Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review, 63: 81-97.

Ng, KYK, Ghanmi, A., 2002. An automated surface surveillance system, Journal of the Operational Research Society 53, 697-708.

Ng KYK, Sancho,NGF, 2009. Regional surveillance of disjoint rectangles: a travelling salesman formulation. Journal of the Operational Research Society 60: 215-220.

Panton, D.M., Elbers,A.W., 1999. Mission planning for synthetic aperture radar surveillance. Interfaces 29: 73–88.

Paquete, L., Stützle, T. 2003. A two-phase local search for the biobjective traveling salesman problem. In C. Fonseca, P. Fleming, E. Zitzler, K. Deb & L. Thiele (Eds.), *Lecture notes in computer science : Vol.* 2632. *Evolutionary multi-criterion optimization (EMO 2003)* (pp. 479–493). New York: Springer.

Paquete, L., Chiarandini, M, Stützle, T. 2004. Pareto local optimum sets in the bi-objective traveling salesman problem: An experimental study, in: X. Gandibleux, M. Sevaux, K. Sörensen, V. T'Kindt (Eds.), Metaheuristics for Multiobjective

Optimization, Lecture Notes in Economics and Mathematical Systems, vol. 535, Springer, 177–199.

Paquete, L., Stützle, T. 2009. Design and analysis of stochastic local search for the multiobjective traveling salesman problem. Computers and Operations Research 36(9), 2619–2631.

Samanlioglu F Jr, Ferrell WG, Kurz ME .2008. A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem. Comput Ind Eng 55(2):439–449.

Simonin,C., Le Cadre,J.P., Dambreville,F., 2009. A hierarchical approach for planning a multisensor multizone search for a moving target. Computers & Operations Research 36:2179-2192.

Şimşek. Ö., 2007. The Biobjective Travelling Salesman Problem With Profit. Ph.D. Thesis, Middle East Technical University, Turkey, Ankara.

Wagner DH, Mylander WC and Sanders TJ. Naval Operations Analysis. 3rd ed. Maryland: Naval Institute Press; 1999.

Yang, M., Kang, L., and Guan, J. 2008. Multi-algorithm co-evolution strategy for dynamic multi-objective TSP, IEEE Congress on Evolutionary Computation (CEC2008), IEEE, 466-471.

Yan, Z., Zhang, L., Kang, L., Lin, G. 2003. A New MOEA for Multi-objective TSP and Its Convergence Property Analysis. In: Fonseca, C. M., Fleming, P. J., Zitzler, E., Deb, K., Thiele, L. (eds.): Evolutionary Multi-Criterion Optimization, Proceedings of Second International Conference, EMO2003. Springer Berlin, 342-354.

APPENDIX A

ALGORITHM OF CONSTRUCTION METHODS

Indices:

i,j : index of disjoint rectangles, i=0 is base station.

l,k : index of entry and exit points of rectangles, l=1,2,..,8 and k=1,2,..,8.

s : index of search pattern within a rectangle, s=1,…S.

Parameters:

D_{iljk} : distance from point l of rectangle i to point k of rectangle j.

DP_{ilks} : distance from point l to point k of rectangle i using search pattern s.

P_{ilks} : probability of detecting target by flying from point l to point k in the rectangle i using search pattern s.

N : number of rectangles.

Variables:

CurrentNode : The rectangle whose search scenario is being decided.

CurrentNodeEntry : Selected entry point of *CurrentNode.*

CurrentNodeExit : Selected exit point of *CurrentNode.*

CurrentNodeSearchPattern : Selected search pattern of *CurrentNode.*

MinofMaxProbability : Minimum of maximum target detection probability for all rectangles.

MaximumProbabillity : Maximum target detection probability that is found for *CurrentNode.*

DistanceTravelled : Total distance travelled in the tour (including the distance travelled between rectangles and in rectangles).

MinDetectionProbabillity : Minimum of the target detection probabilities for all rectangles based on selected search scenarios.

Construction Method 1:

Step 1: Find optimal TSP tour based on middle points of the rectangles by using Concorde TSP Solver. Set sequence of rectangles in the initial tour according to optimal TSP sequence.

Step 2: Select the entry point, exit point and search pattern resulting in largest probability of target detection for all rectangles (*CurrentNode).*

Set *MaximumProbabillity*=0.

Repeat for *CurrentNode*=1 to *CurrentNode*=N

Repeat for all valid sets of l, k and s for *CurrentNode*

If $P_{CurrentNode,l,k,s}$>*MaximumProbabillity*

Set *CurrentNodeEntry*=l

Set *CurrentNodeExit*=k

Set *CurrentNodeSearchPattern*=s

Set *MaximumProbabillity*= $P_{CurrentNode,l,k,s}$

<u>Construction Method 2:</u>

Set *MinofMaxProbability* to minimum acceptable target detection probability and solve Minisum ASP optimally with the constraint below by using GAMS.

$P_{ilks} \geq$ *MinofMaxProbability*

Set sequence and search scenarios of rectangles in the initial tour according to optimal sequence found.

<u>Construction Method 3 :</u>

Step 1: Generate a specified number of random sequences of all rectangles and base station. Connect the first and the last node in the sequence to generate a tour.

Step 2: For each tour, calculate the total length of the tour by using the midpoints of rectangles.

Step 3: Select the tour having minimum length. Set the sequences in the initial tour according to the tour having minimum length.

Step 4: Select the entry point, exit point and search pattern resulting in largest probability of target detection for all rectangles (*CurrentNode)*.

Set *MaximumProbabillity*=0.

Repeat for *CurrentNode*=1 to *CurrentNode*=N

Repeat for all valid sets of l, k and s for *CurrentNode*

If $P_{CurrentNode,l,k,s}$>*MaximumProbabillity*

Set *CurrentNodeEntry*=l

Set *CurrentNodeExit*=k

Set *CurrentNodeSearchPattern*=s

Set *MaximumProbabillity*= $P_{CurrentNode,l,k,s}$

APPENDIX B

ALGORITHM OF IMPROVEMENT METHOD

Indices:

i,j : index of disjoint rectangles, i=0 is base station.

l,k : index of entry and exit points of rectangles, l=1,2,..,8 and k=1,2,..,8.

s : index of search pattern within a rectangle, s=1,…S.

Parameters:

D_{iljk} : distance from point l of rectangle i to point k of rectangle j.

DP_{ilks} : distance from point l to point k of rectangle i using search pattern s.

P_{ilks} : probability of detecting target by flying from point l to point k in the rectangle i using search pattern s.

N : number of rectangles.

Set:

EfficientSolutions : Set of nondominated solutions.

Variables:

CurrentSolution : A solution chosen from *EfficientSolutions* to be improved.

CandidateSolution1 : A solution found when 2-opt move is applied to *CurrentSolution.*

CandidateSolution2 : A solution found when search scenarios of the rectangles are changed on *CandidateSolution1*.

Probability : Target detection probability that is used to restrict the rectangles' target detection probability in *CandidateSolution2.*

MinofMaxProbability : Minimum of maximum target detection probability for all rectangles.

DistanceTravelled : Total distance travelled in the tour (including the distance travelled between rectangles and in rectangles).

MinDetectionProbabillity : Minimum of the target detection probabilities for all rectangles based on selected search scenarios.

Rectangle1-4 : Rectangles chosen to apply 2-opt move.

Step 1: Start with an initial feasible solution found by one of the Construction Methods and add it to *EfficientSolutions.*

Step 2: Pick the first solution from *EfficientSolutions* which is not flagged as *visited* and set it as *CurrentSolution.* If all solutions in *EfficientSolutions* set is flagged as *visited*, STOP.

Step 3: Flag *CurrentSolution* as *visited.*

Step 4: Repeat for *Rectangle1*=0 to *Rectangle1*=N

Step 4.1: Repeat for *Rectangle3*=N to *Rectangle3*=0

Step 4.1.1:

If it is feasible, apply 2-opt move on *CurrentSolution* by using the edges connecting *Rectangle1 and Rectangle3* to their next nodes in the tour, which are *Rectangle2* and *Rectangle4.*

Else, Go to Step 4.1.

Step 4.1.2: Save new solution found after 2-opt move as *CandidateSolution1*

Step 4.1.3: Set *MinofMaxProbability* as the minimum of maximum target detection probability for all rectangles.

Step 4.1.4: Set *Probability* to the minimum acceptable target detection.

Step 4.1.5: If *Probability<=MinofMaxProbability*, go to Step 4.1.6.
Else, Go to Step 4.1.

Step 4.1.6: Search alternative ways of travelling in the rectangles.
For all rectangles i=1 to i=N in *CandidateSolution1*, select the way of travelling in the rectangle for which probability P_{ilks} is larger than *Probability* and which results in minimum distance for travelling from previous rectangle to rectangle i, travelling in rectangle i and travelling from rectangle i to the next rectangle. Save new solution as *CandidateSolution2.*

Set *MinDetectionProbabillity* to the minimum of $P_{CurrentNode,CurrentNoEntry,CurrentNodeExit,CurrentNodeSearchPattern}$ for all rectangles in *CandidateSolution2*.

Find total distance travelled (*DistanceTravelled)* by summing D_{iljk} and DP_{ilks} values of the tour in *CandidateSolution2*.

If *CandidateSolution2* is not dominated by any solution in *EfficientSolutions*

Add *CandidateSolution2 to EfficientSolutions.*

Update *EfficientSolutions* by deleting solutions from *EfficientSolutions* set if they are dominated by *CandidateSolution2*

Step 4.1.7: Set *Probabillity* to the minimum of selected P_{ilks} values in *CandidateSolution2.* Go to Step 4.1.5.

Step 5: Go to Step 2.

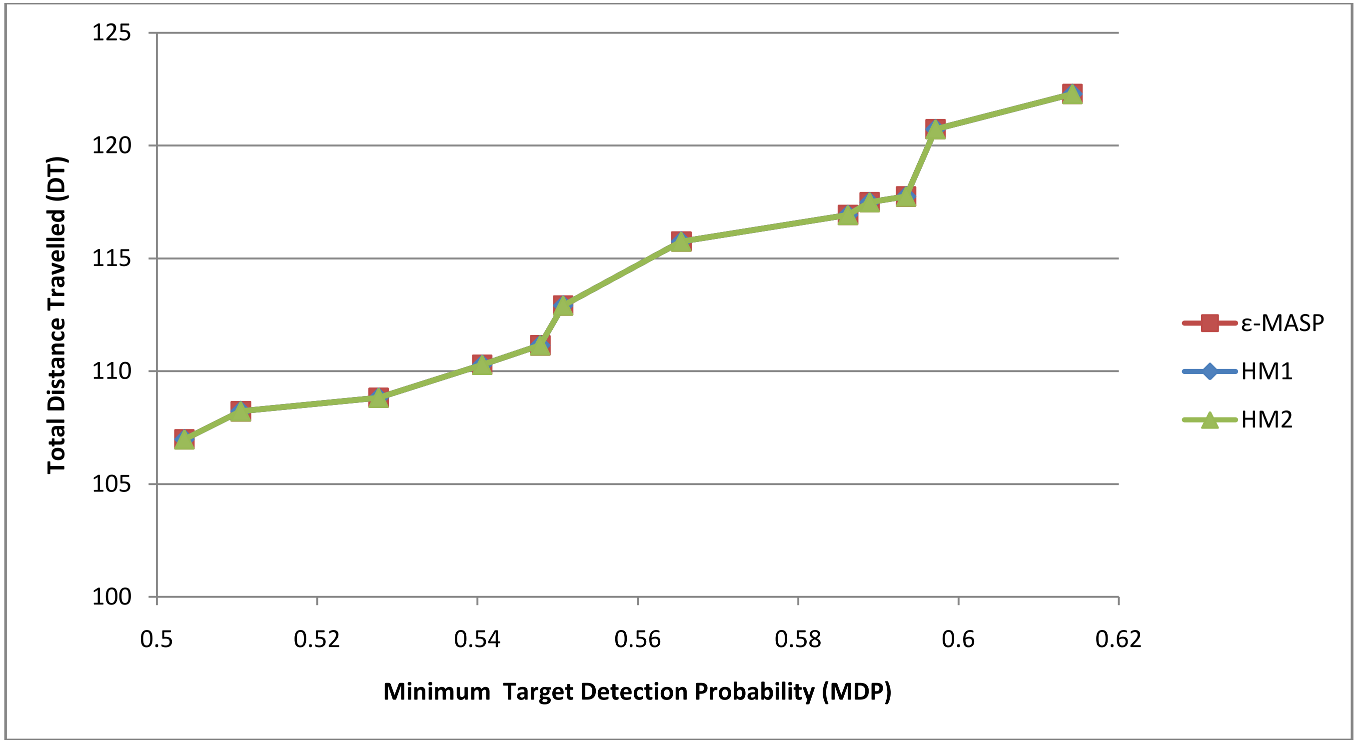

Figure 24 : The objective space of Ins1_5 showing solution sets found

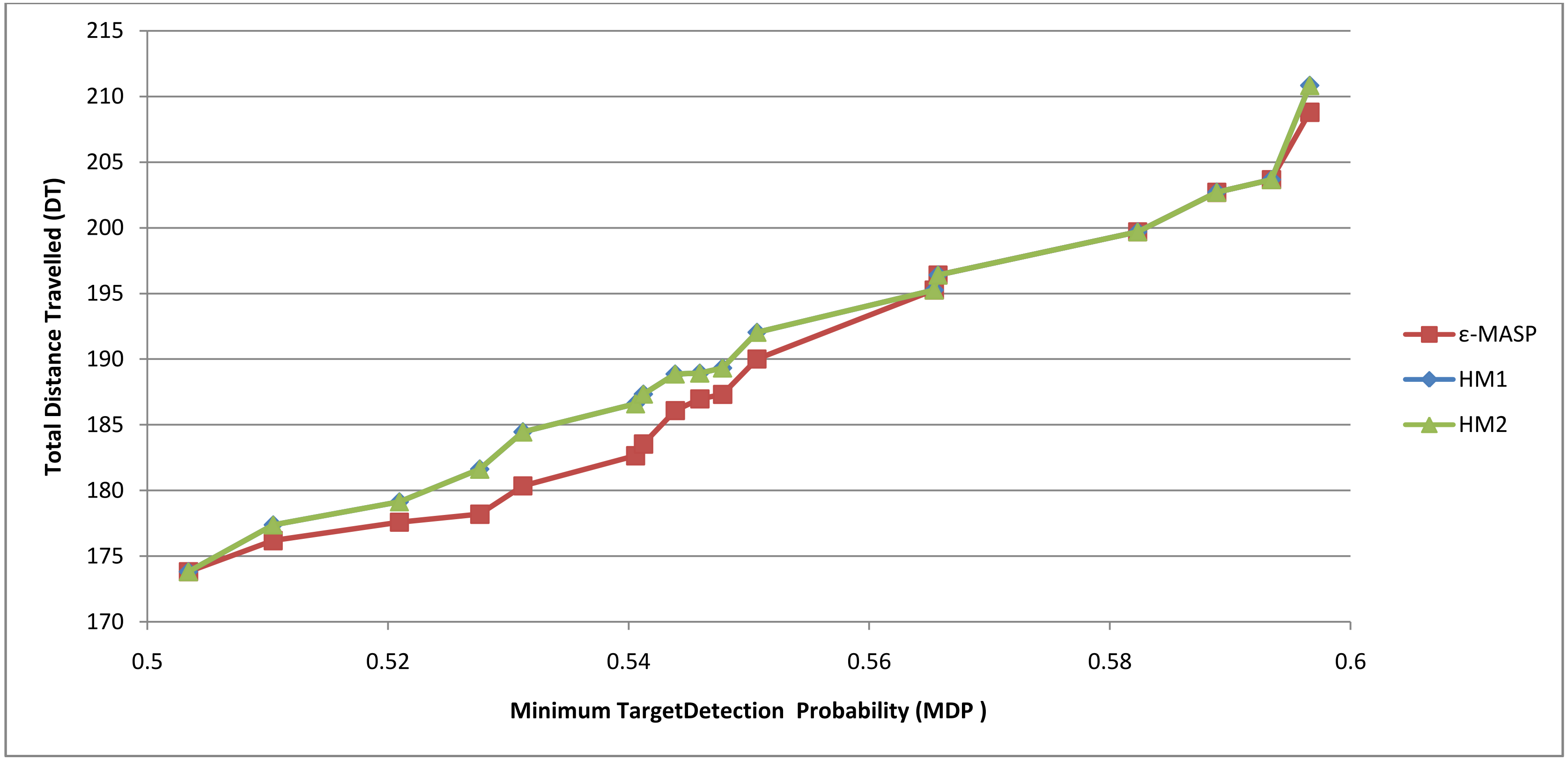

Figure 25: The objective space of Ins1_10 showing solution sets found

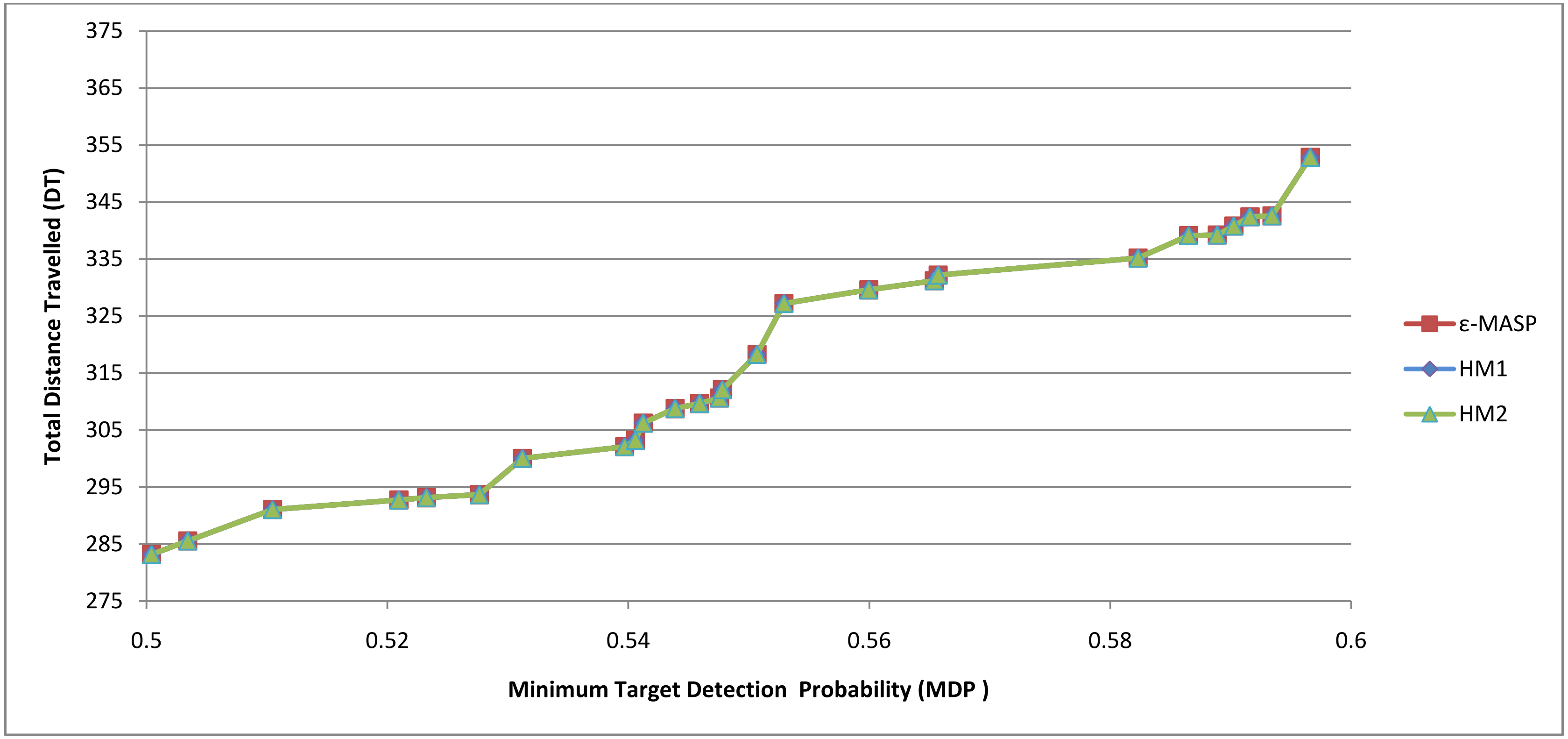

Figure 26 : The objective space of Ins1_20 showing solution sets found

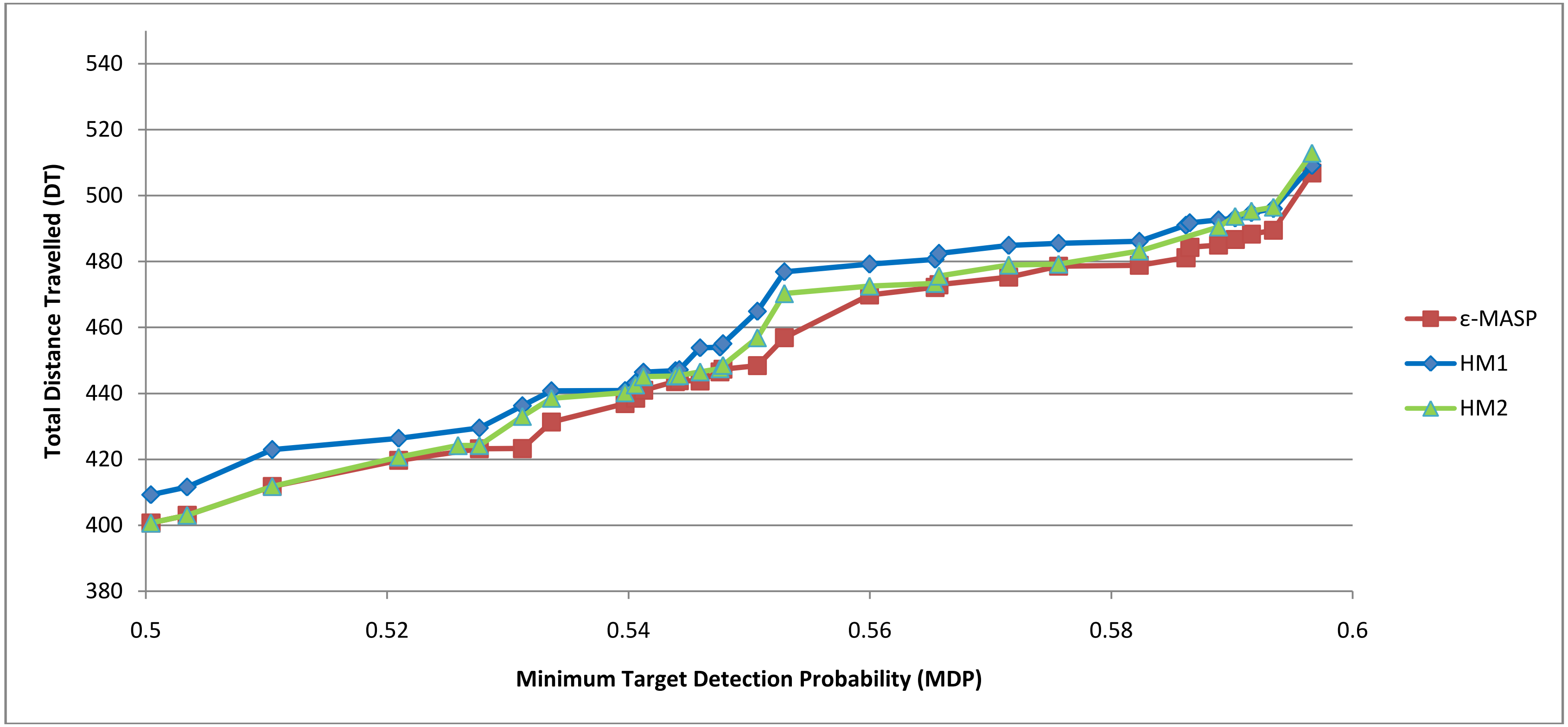

Figure 27 : The objective space of Ins1_30 showing solution sets found

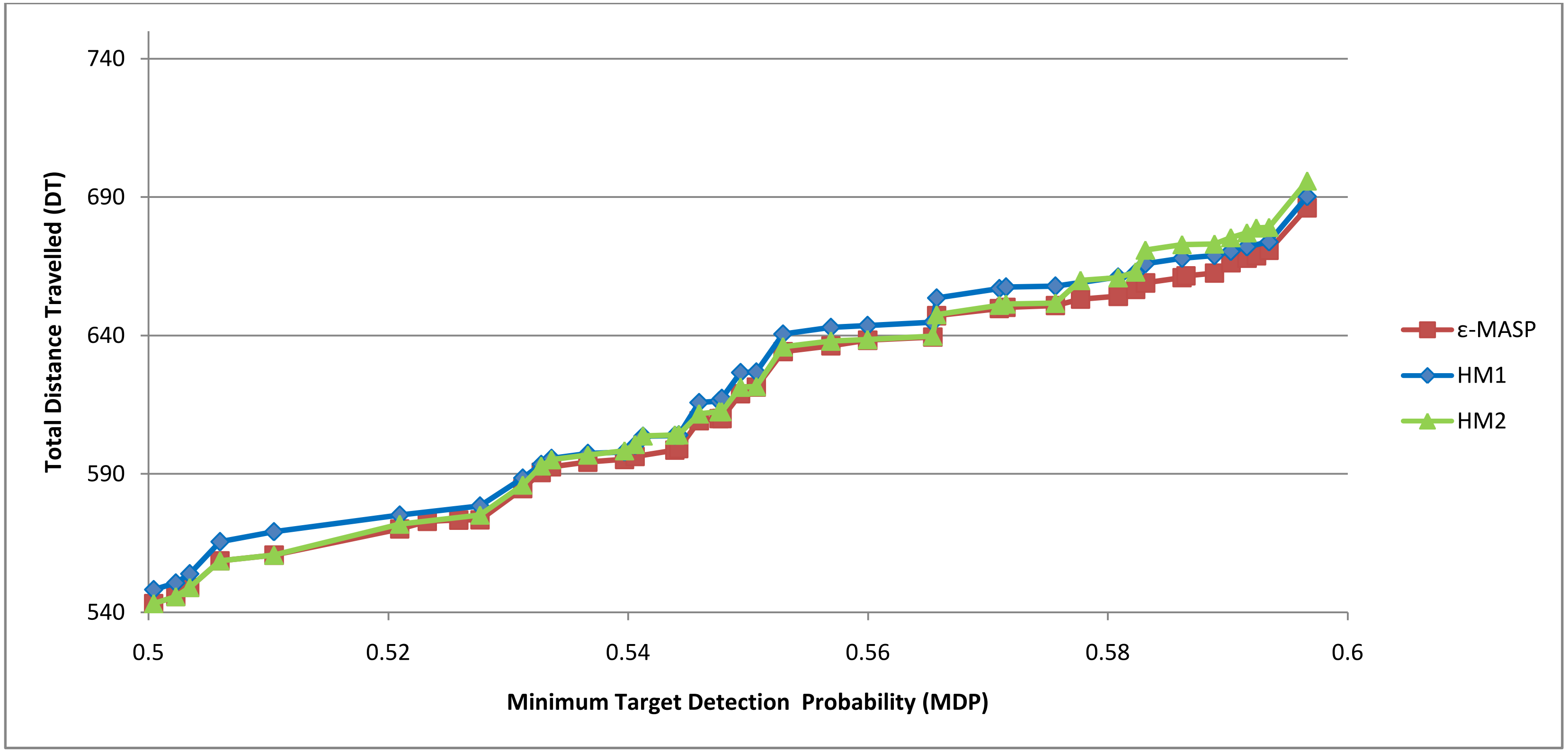

Figure 28 : The objective space of Ins1_40 showing solution sets found

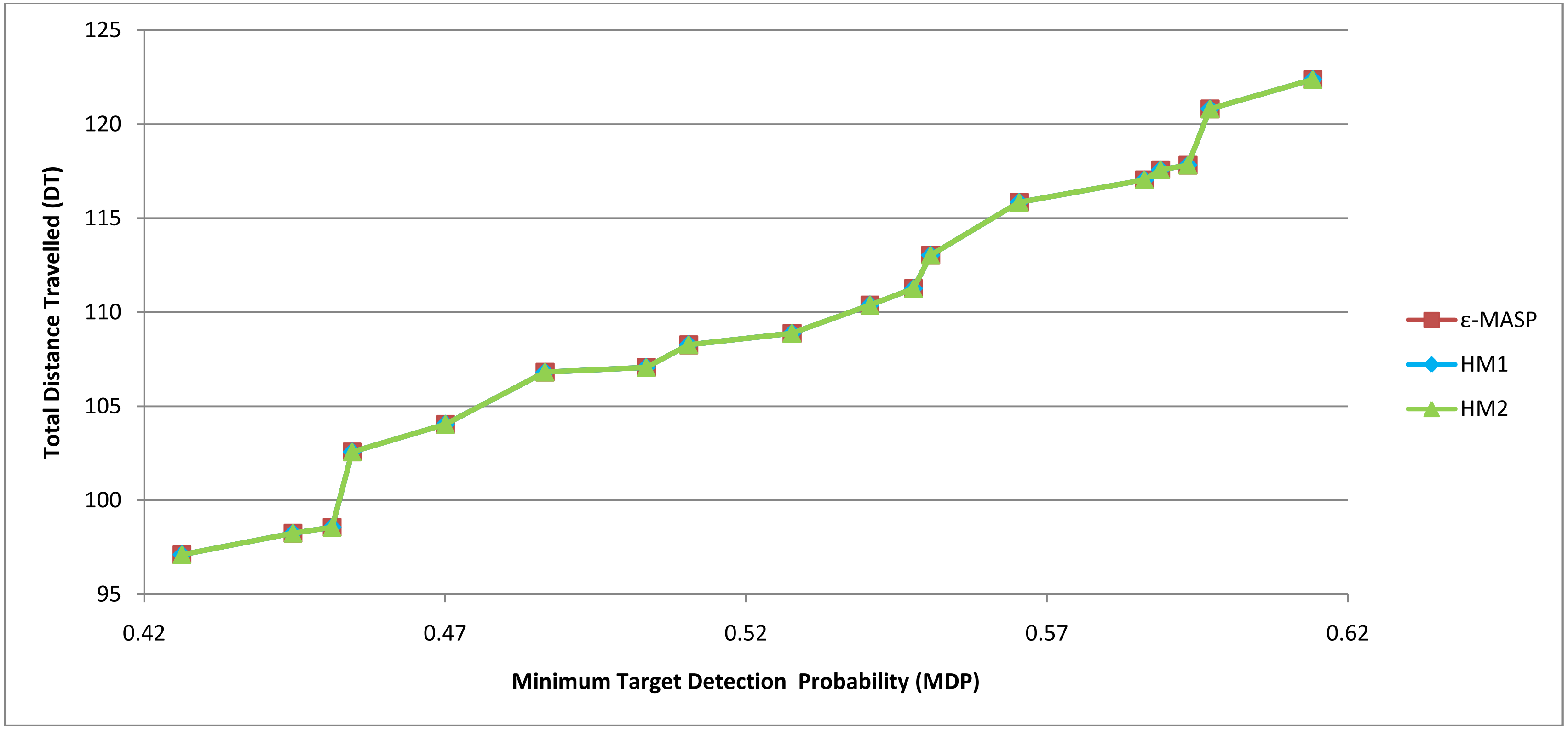

Figure 29 : The objective space of Ins2_5 showing solution sets found

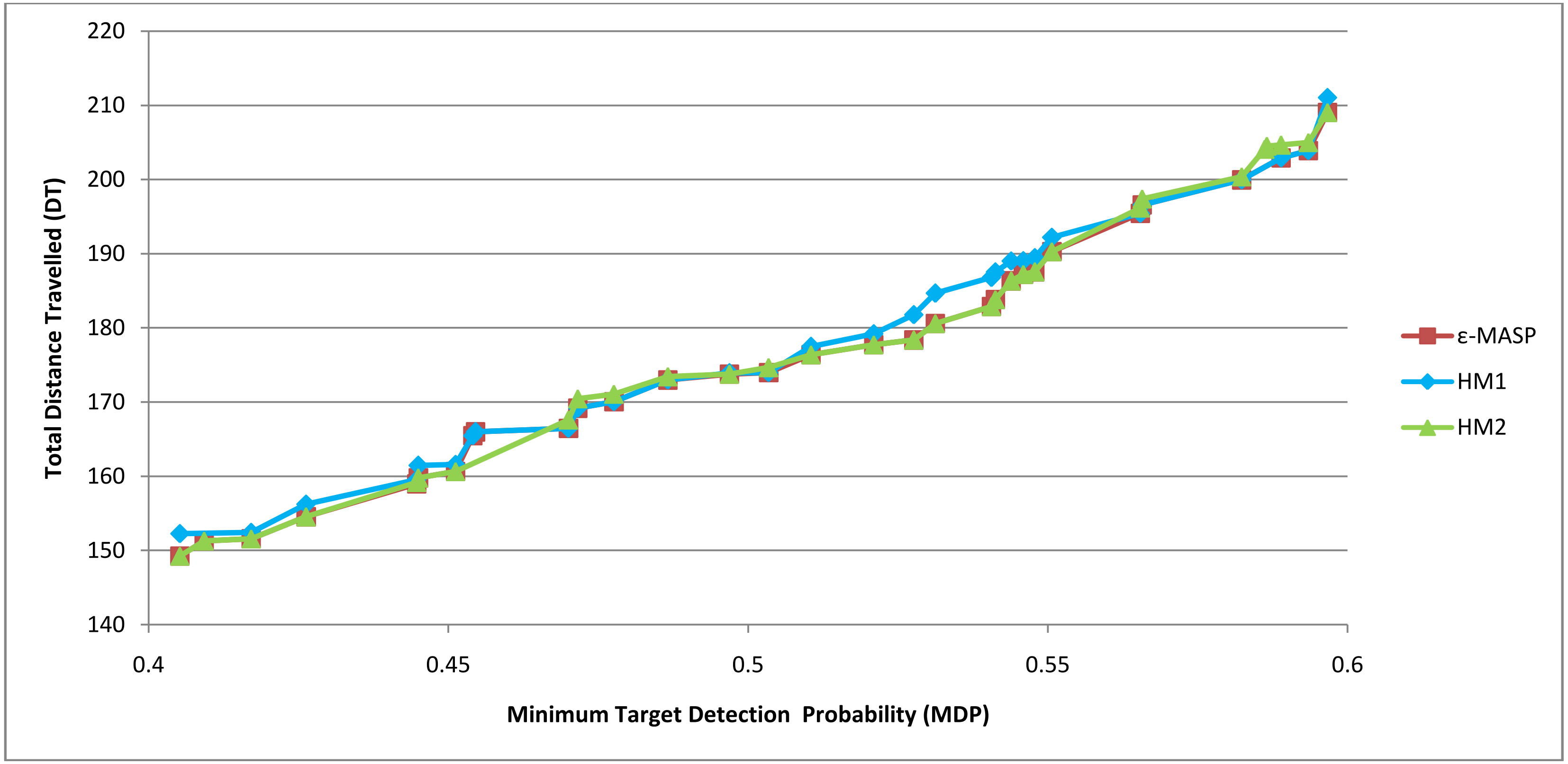

Figure 30 : The objective space of Ins2_10 showing solution sets found

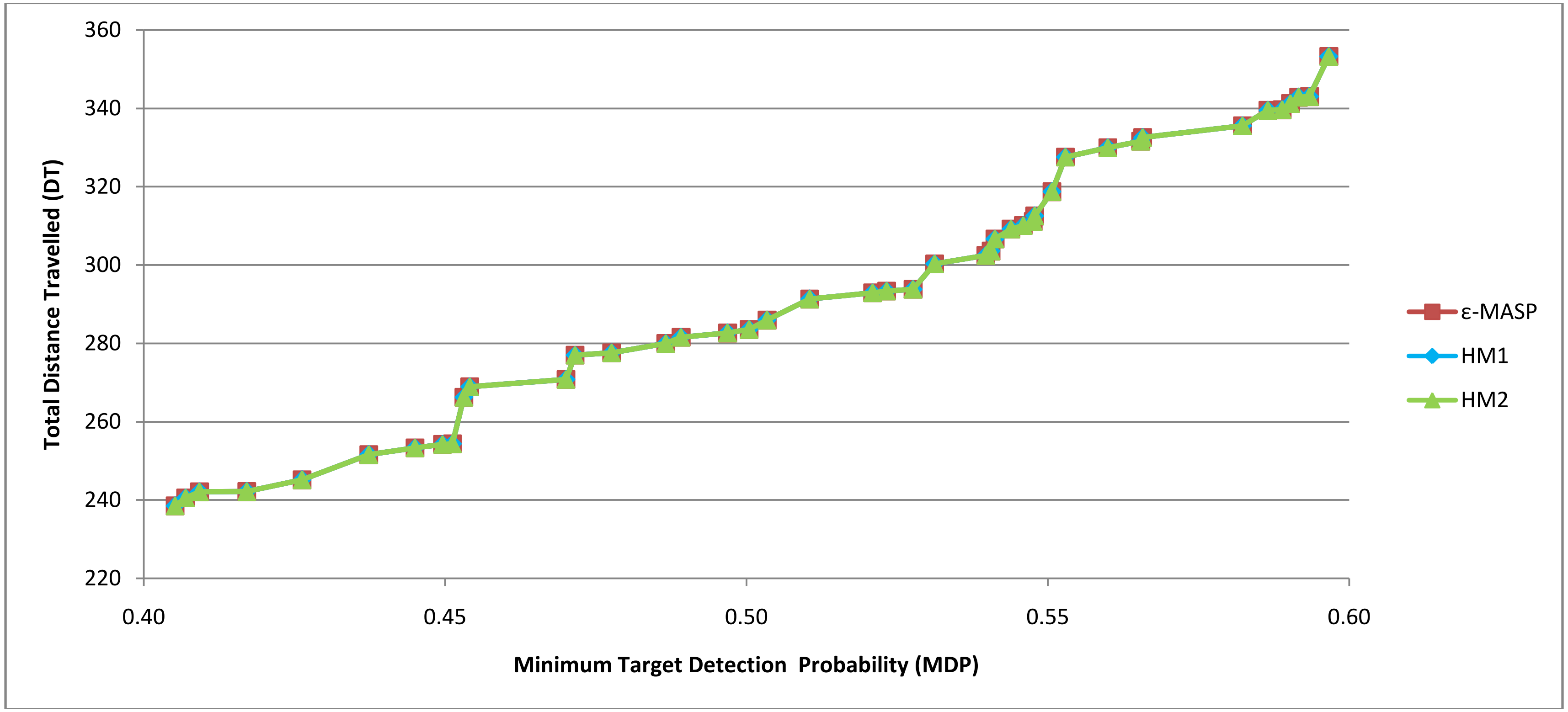

Figure 31 : The objective space of Ins2_20 showing solution sets found

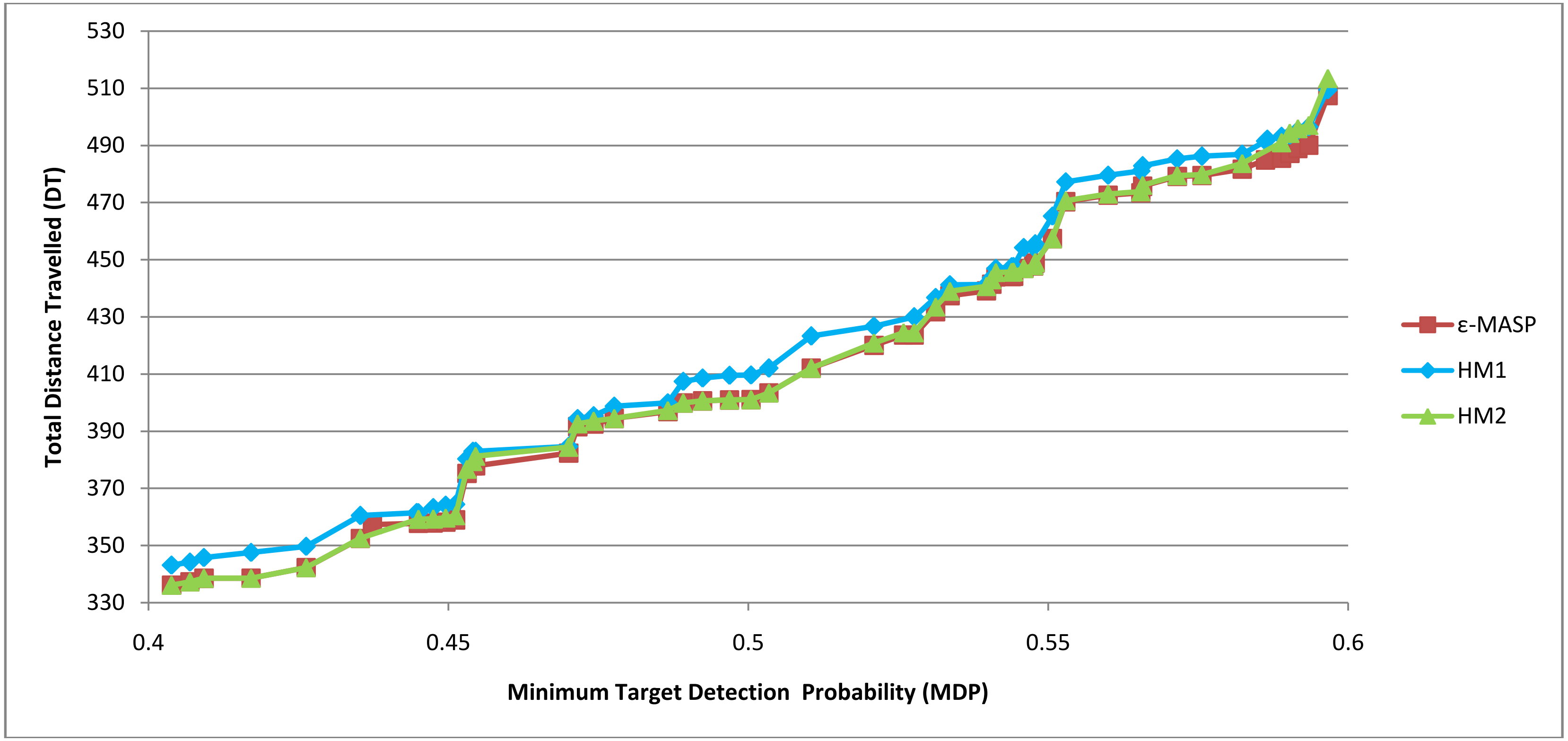

Figure 32 : The objective space of Ins2_30 showing solution sets found

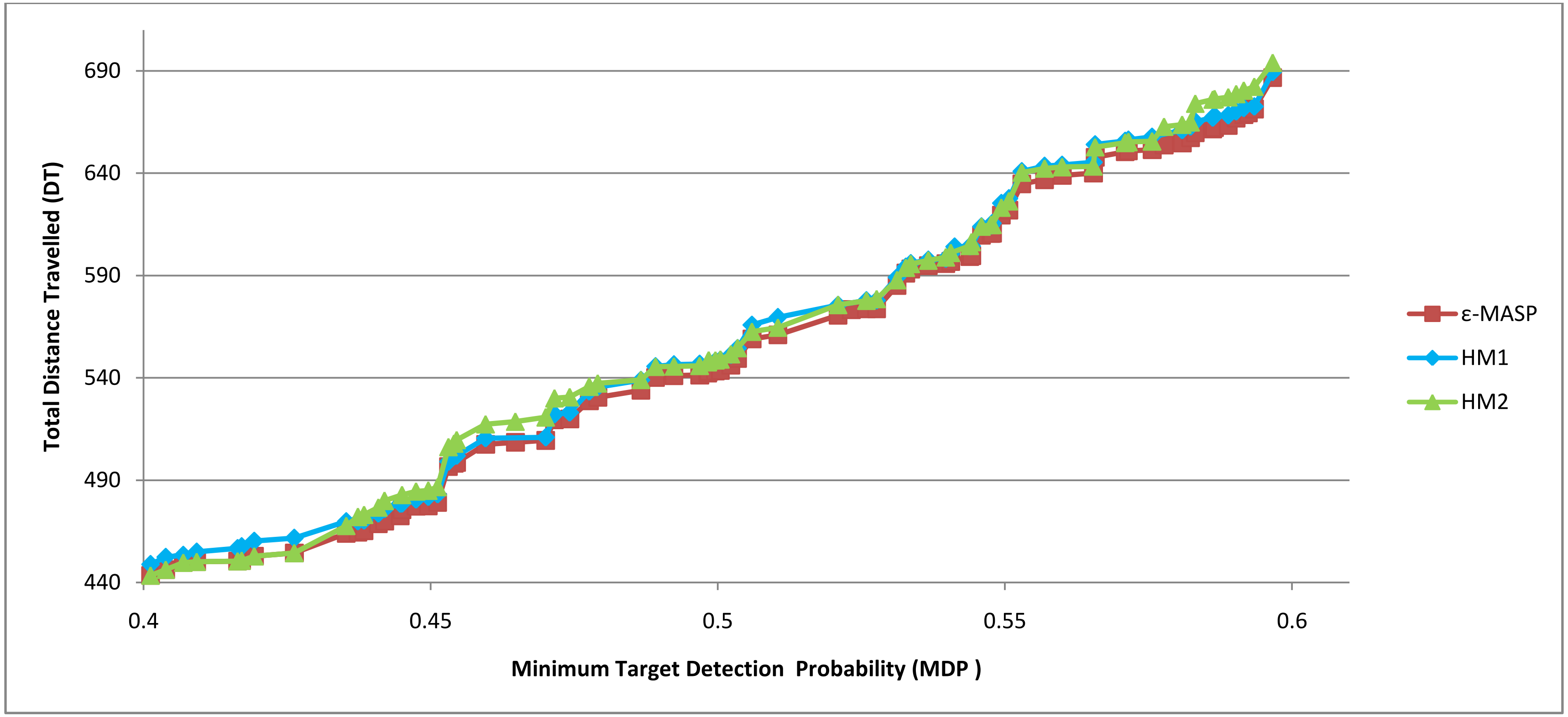

Figure 33 : The objective space of Ins2_40 showing solution sets found

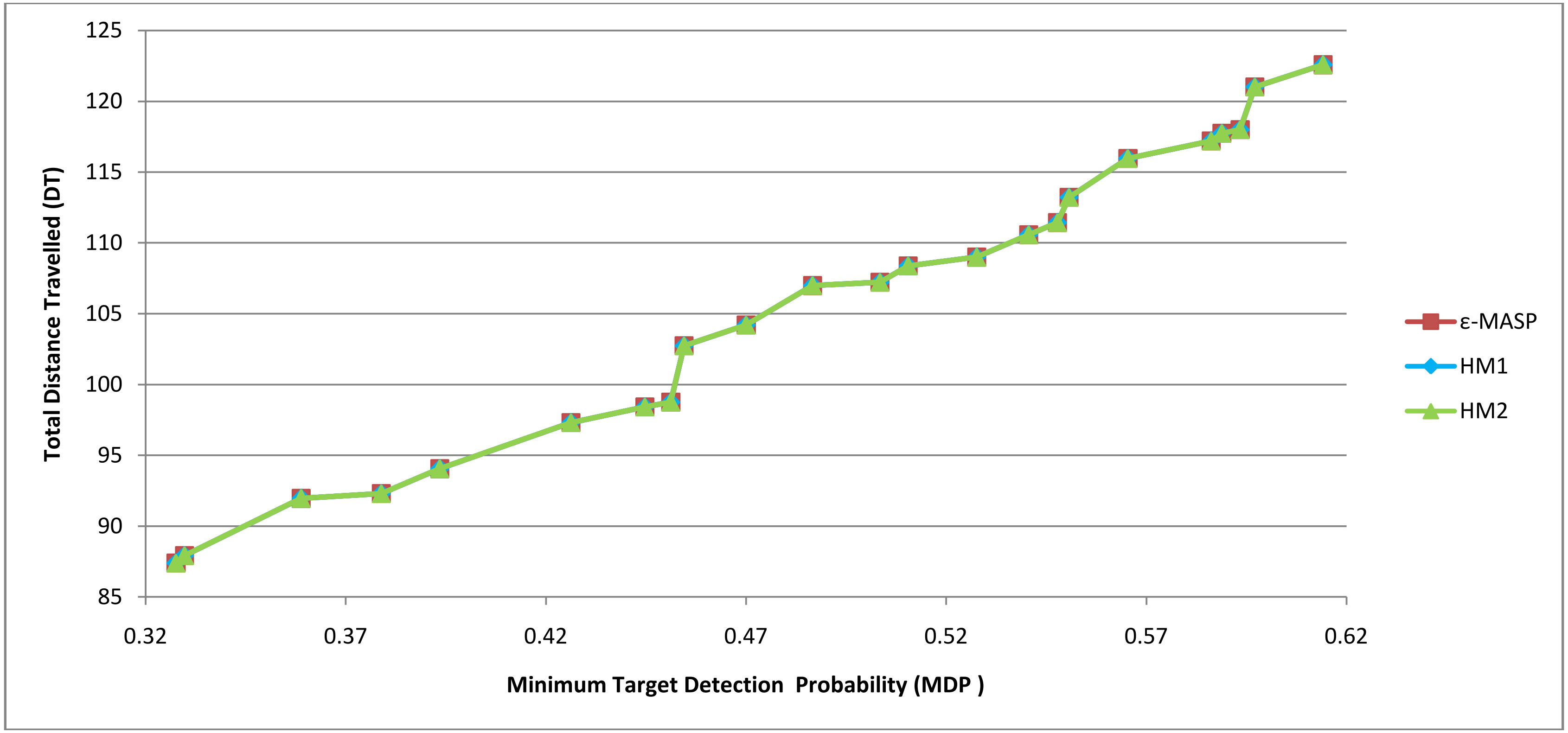

Figure 34 : The objective space of Ins3_5 showing solution sets found

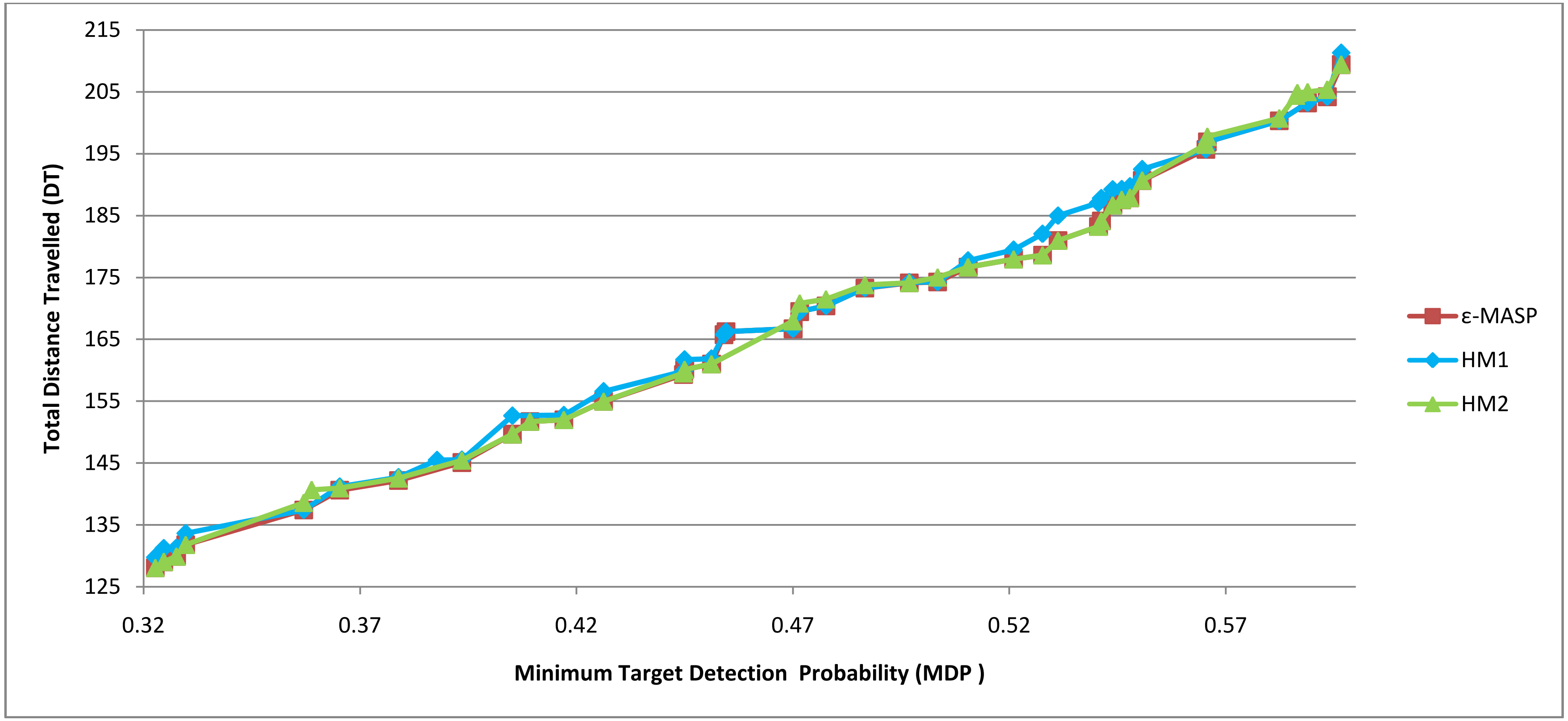

Figure 35 : The objective space of Ins3_10 showing solution sets found

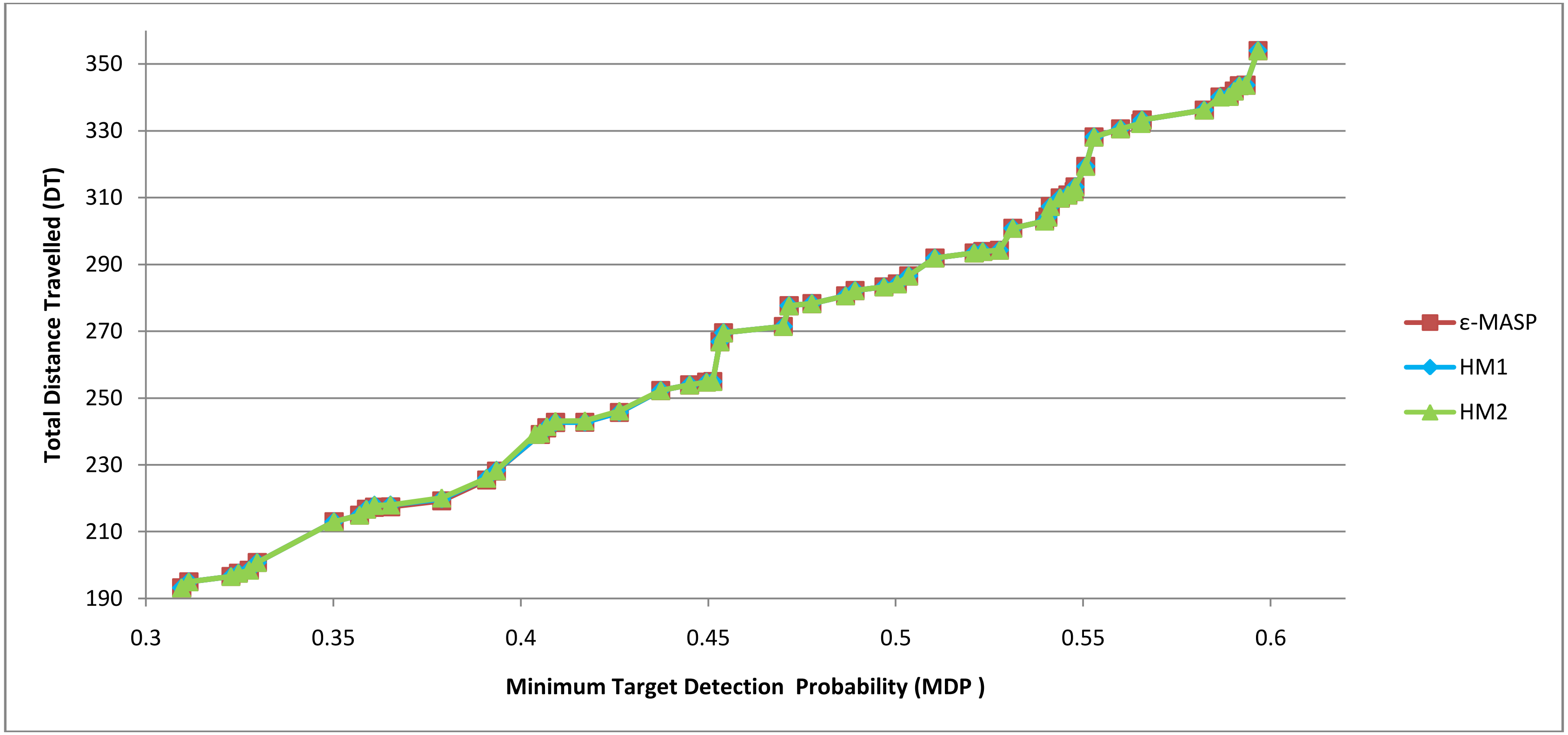

Figure 36 : The objective space of Ins3_20 showing solution sets found

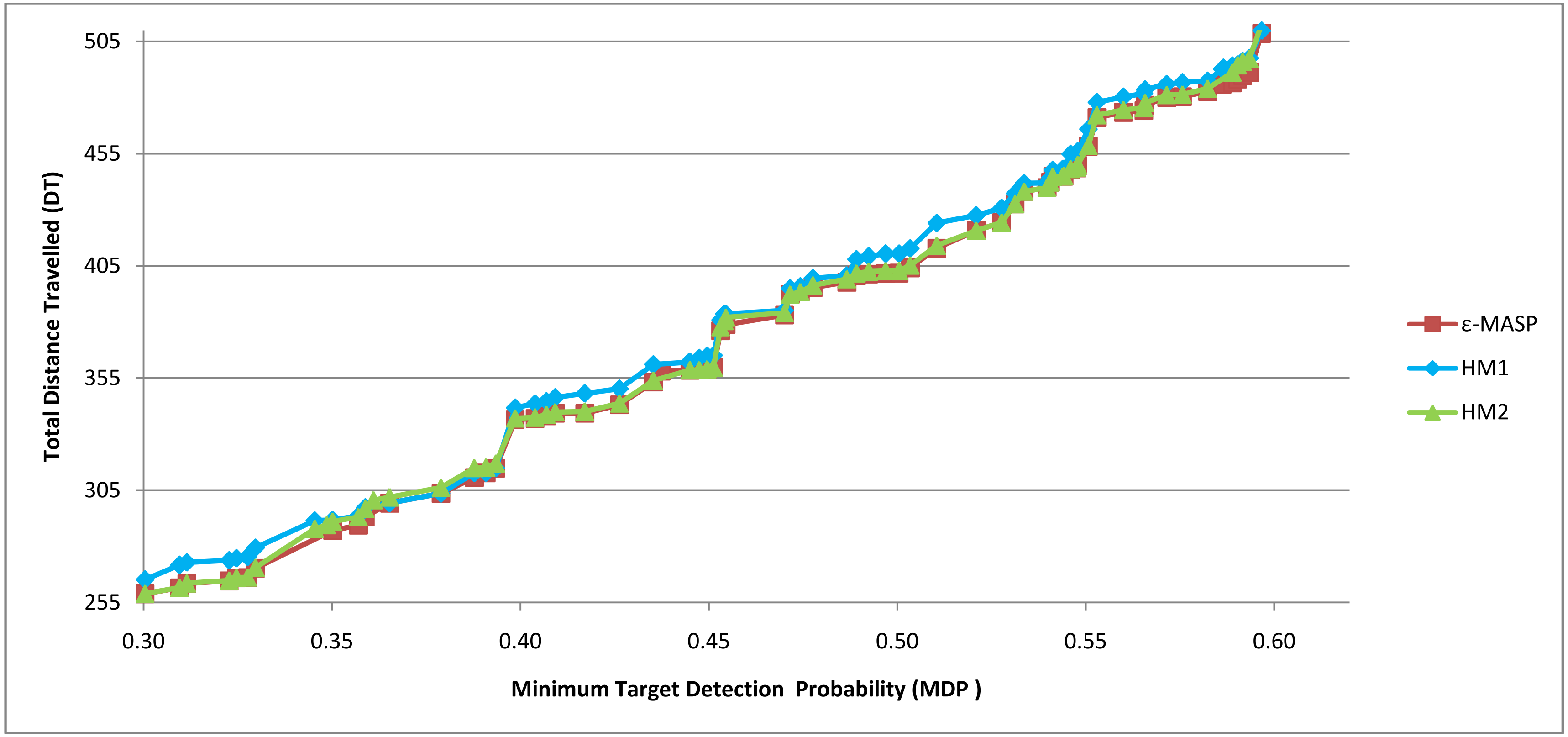

Figure 37 : The objective space of Ins3_30 showing solution sets found

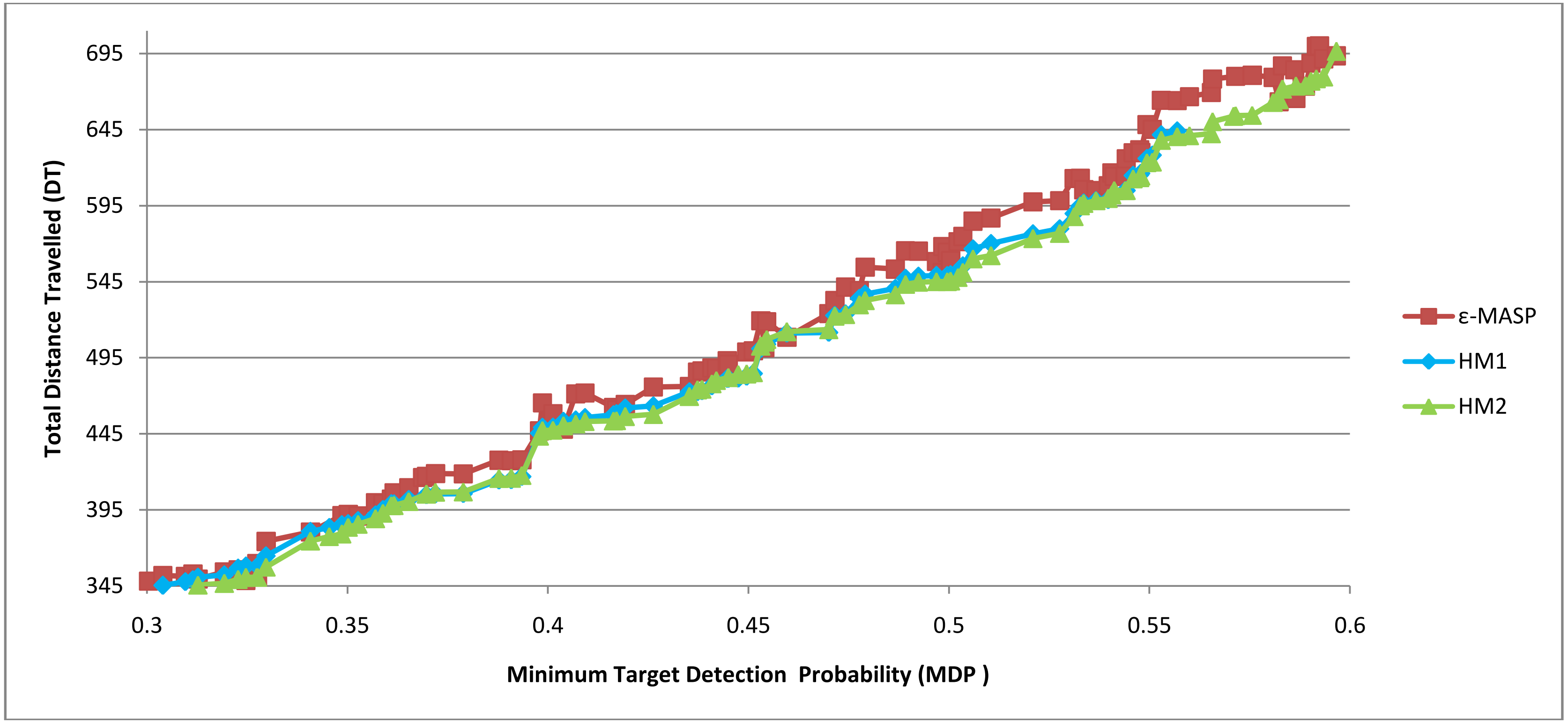

Figure 38 : The objective space of Ins3_40 showing solution sets found

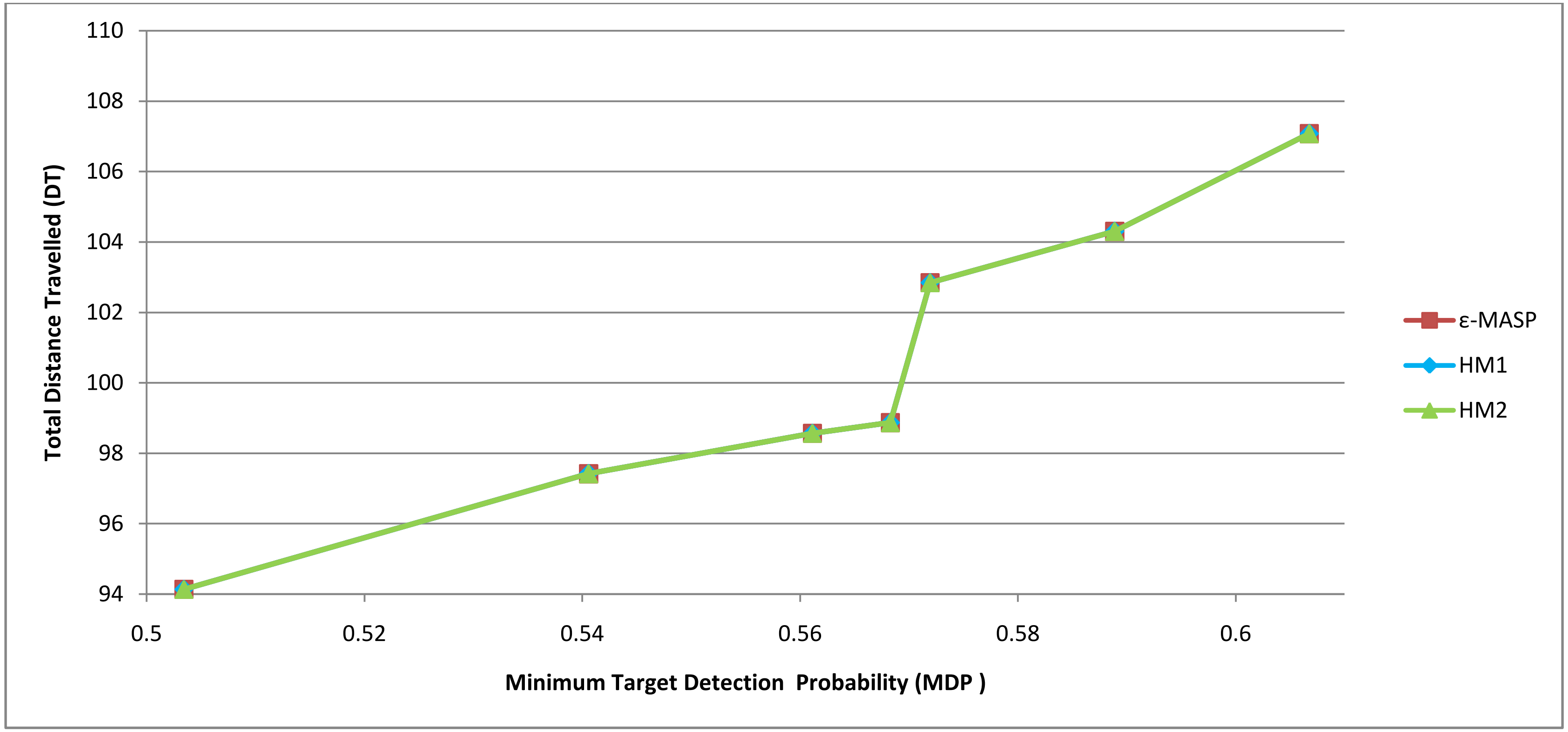

Figure 39 : The objective space of Ins4_5 showing solution sets found

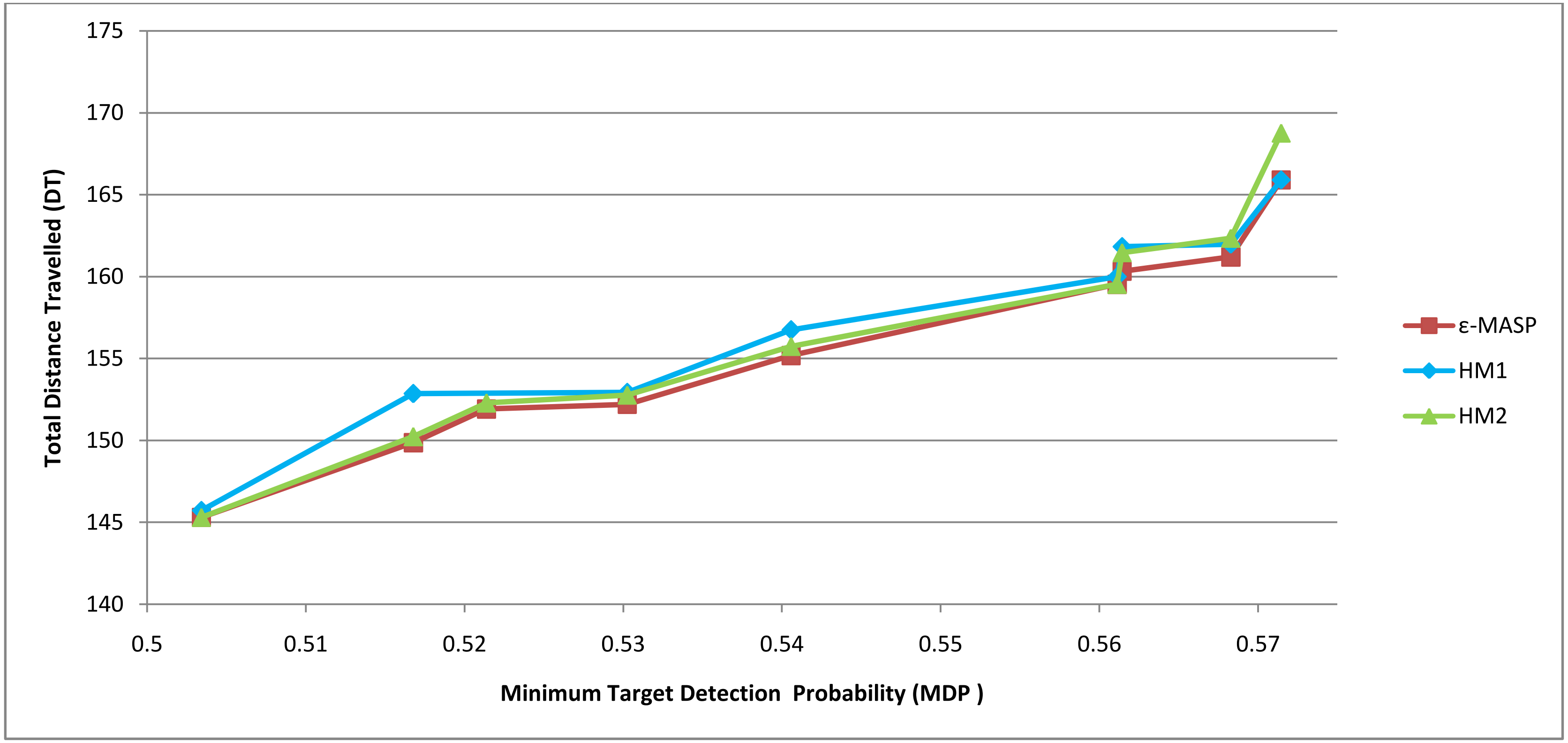

Figure 40 : The objective space of Ins4_10 showing solution sets found

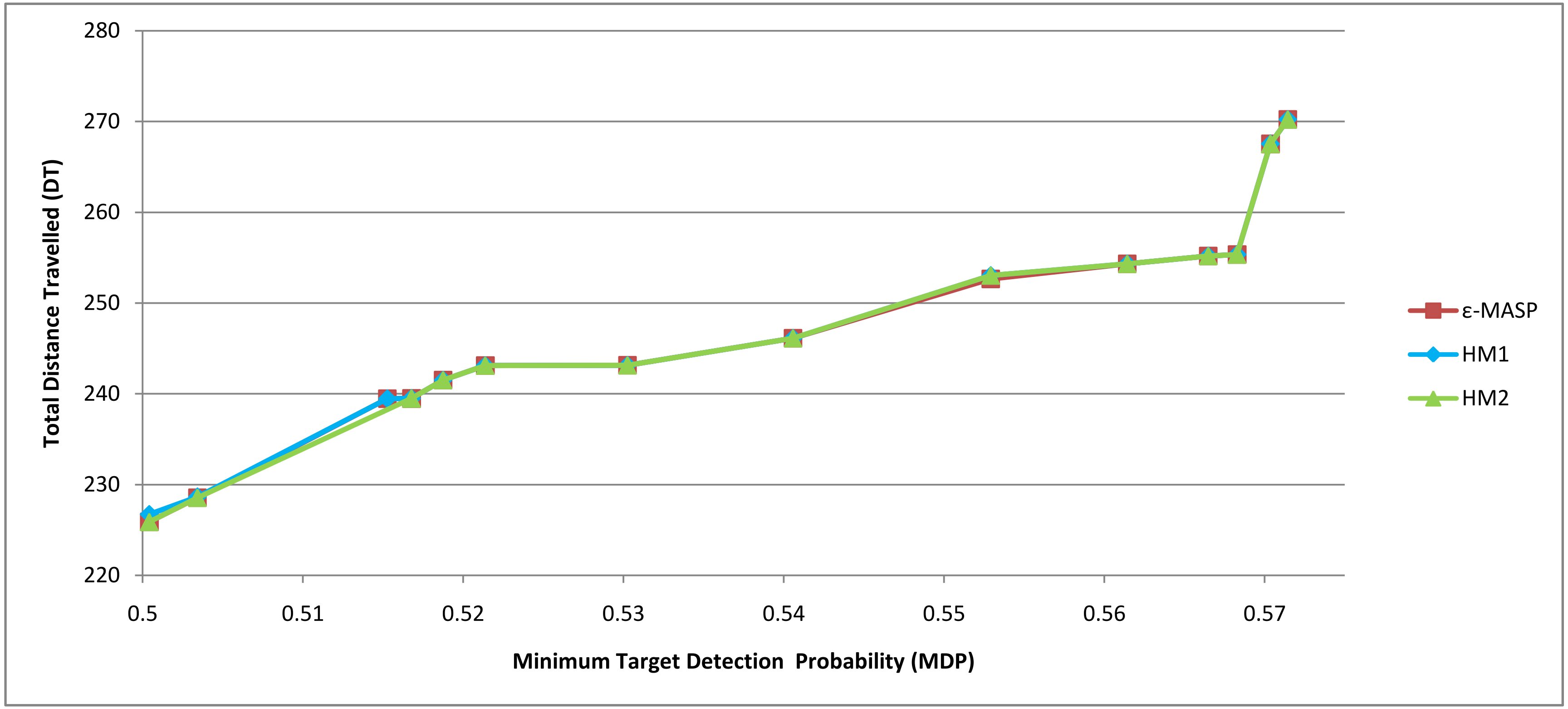

Figure 41 : The objective space of Ins4_20 showing solution sets found

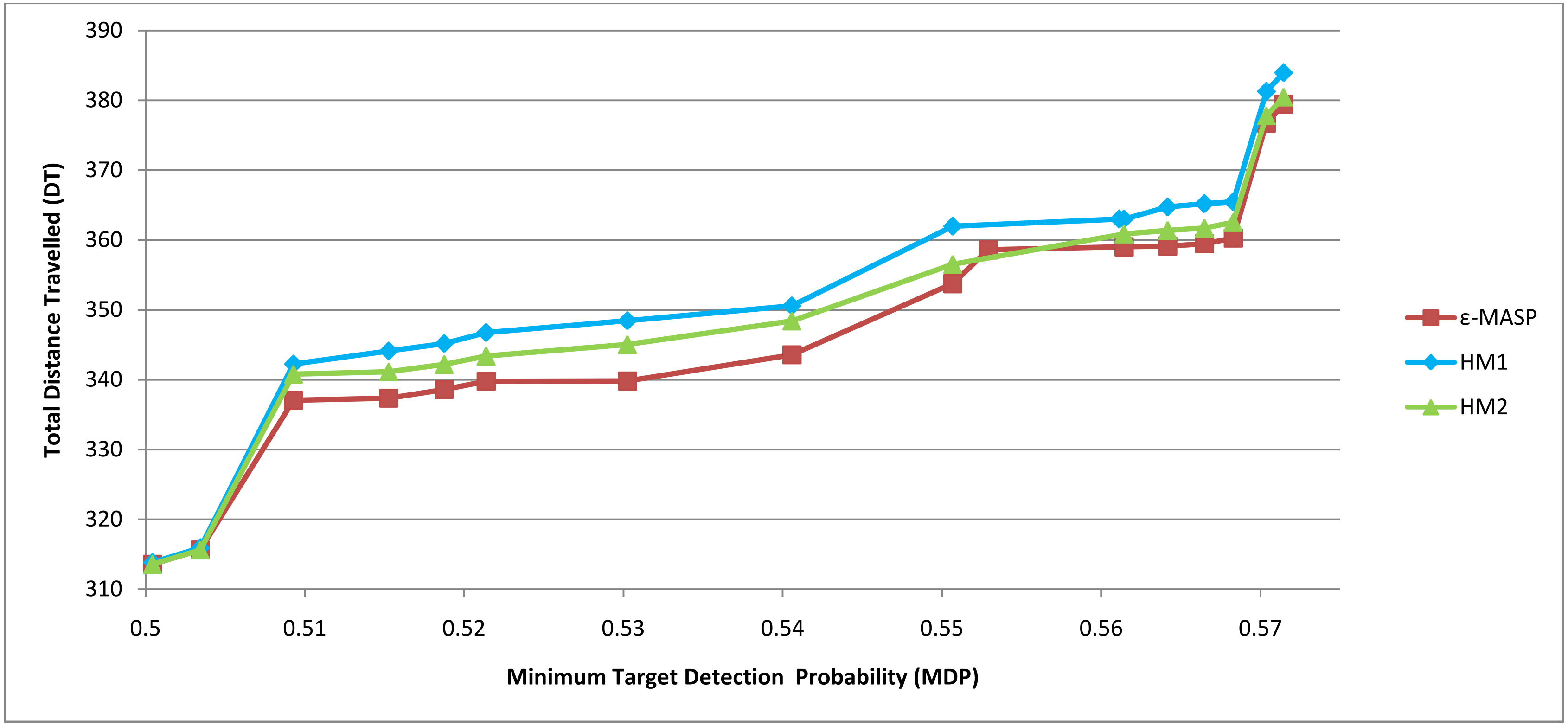

Figure 42 : The objective space of Ins4_30 showing solution sets found

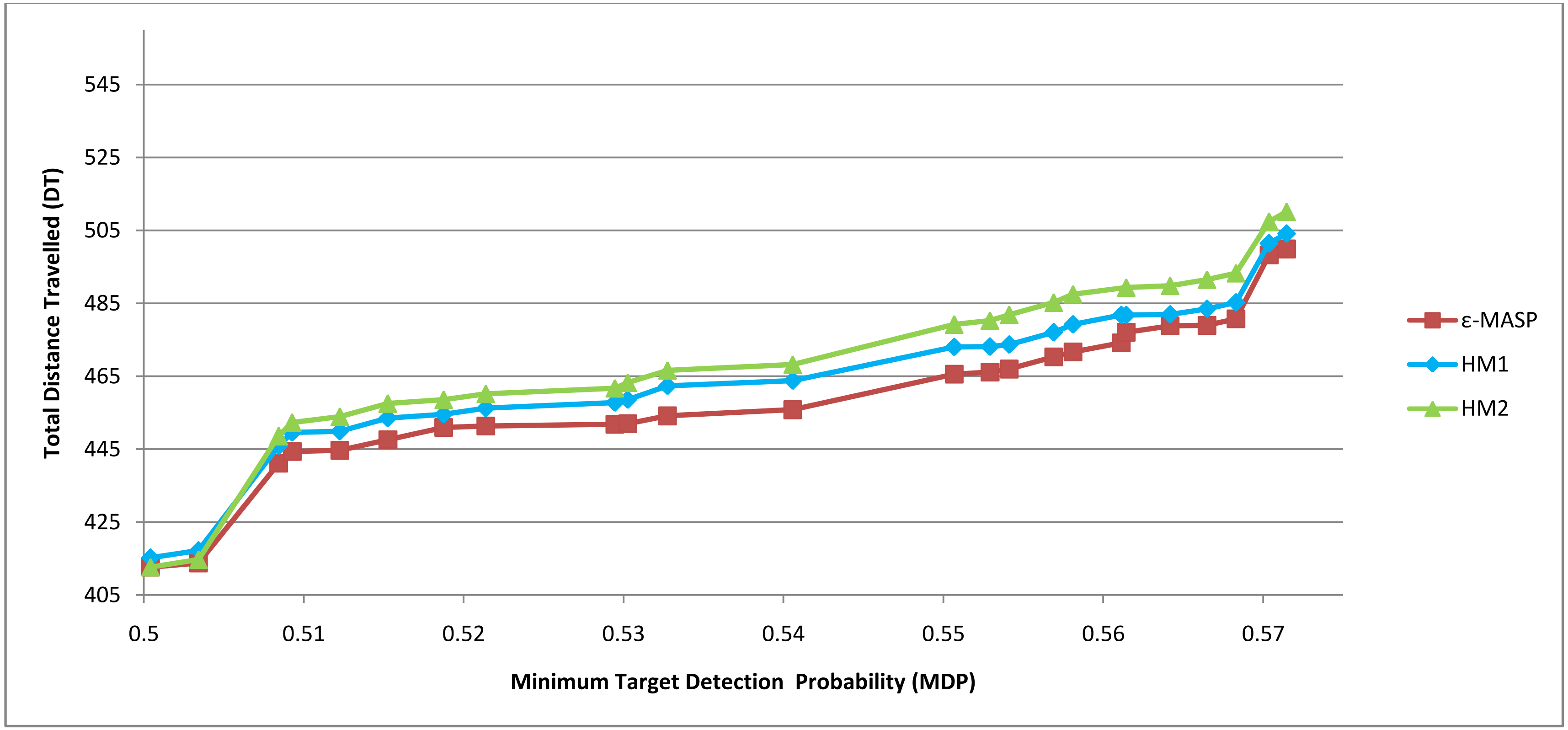

Figure 43 : The objective space of Ins4_40 showing solution sets found

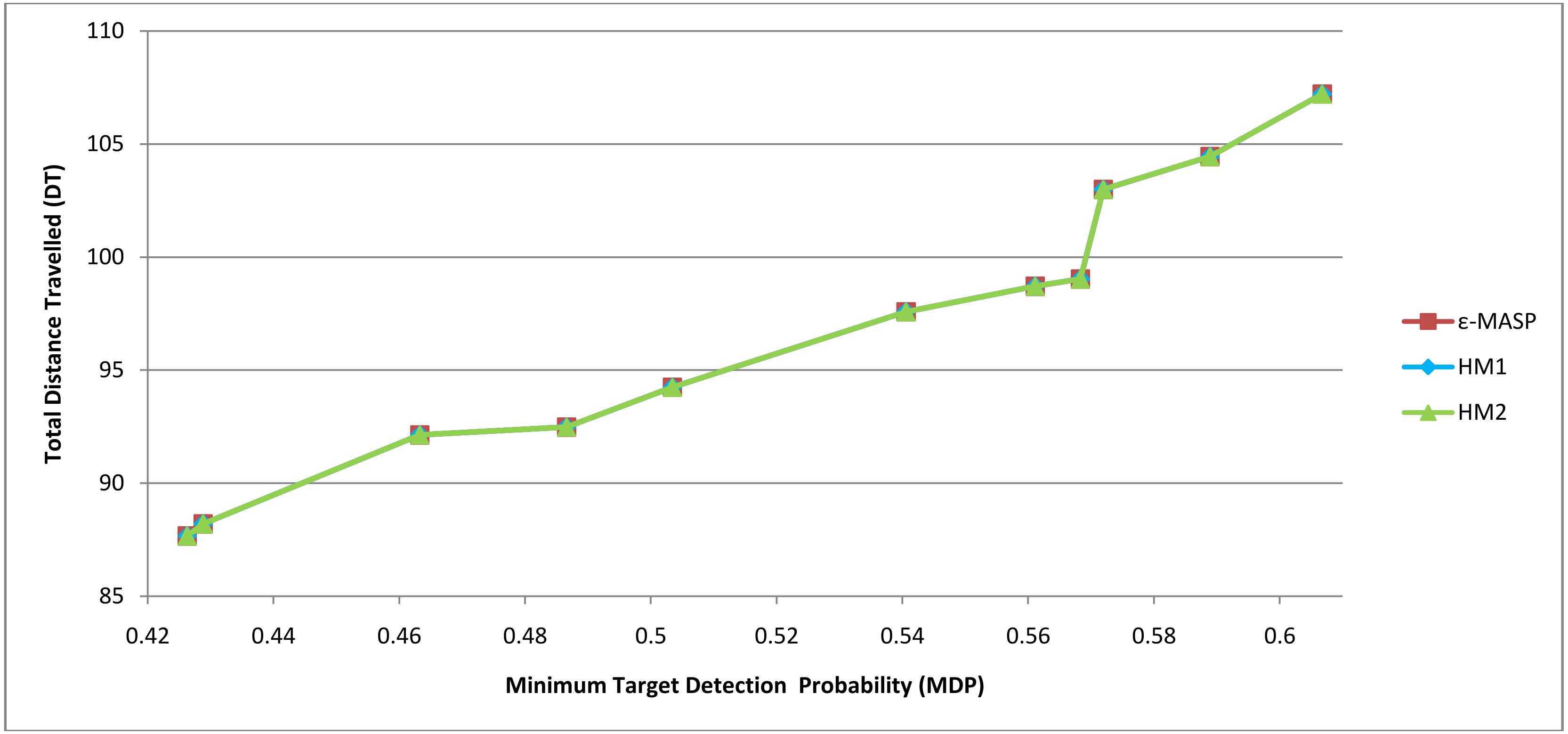

Figure 44 : The objective space of Ins5_5 showing solution sets found

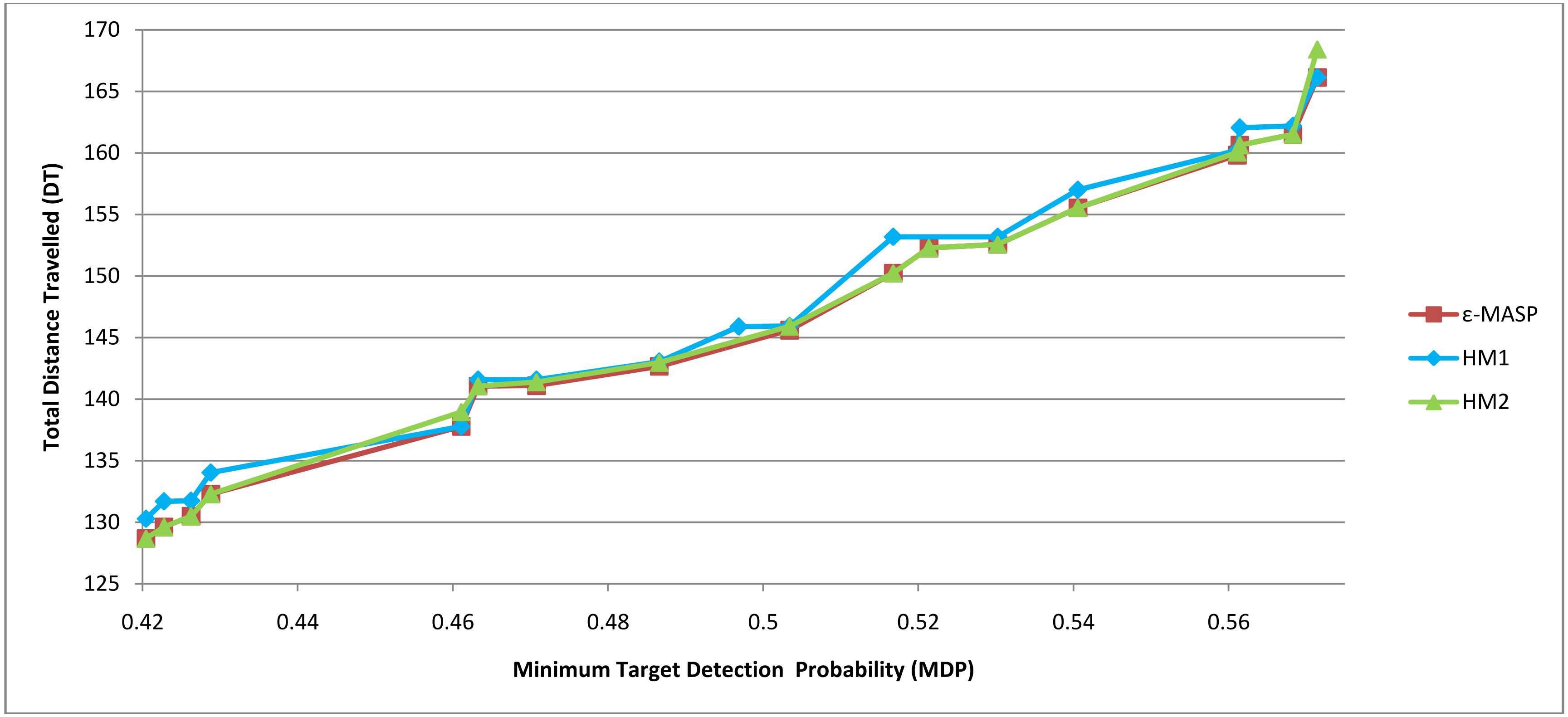

Figure 45 : The objective space of Ins5_10 showing solution sets found

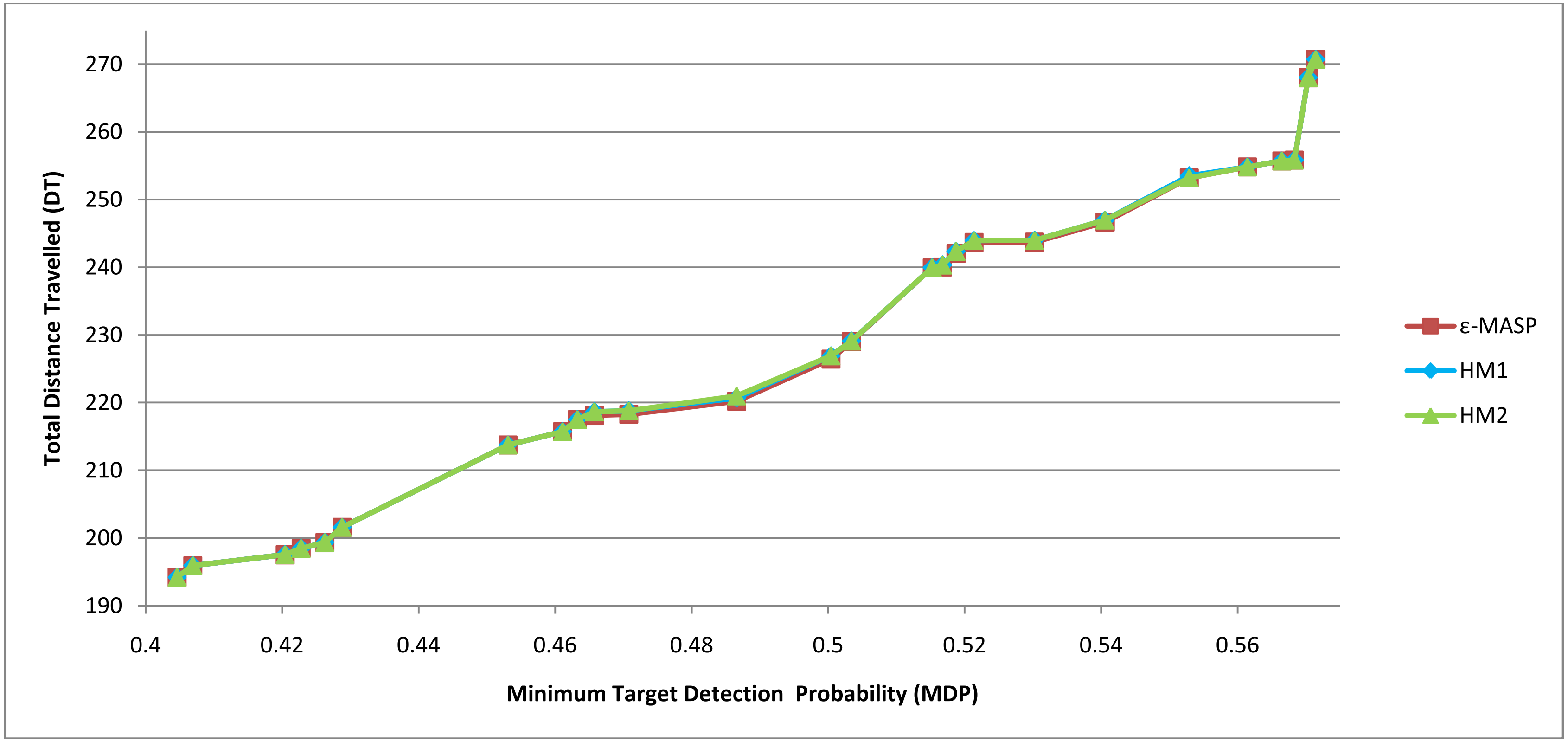

Figure 46 : The objective space of Ins5_20 showing solution sets found

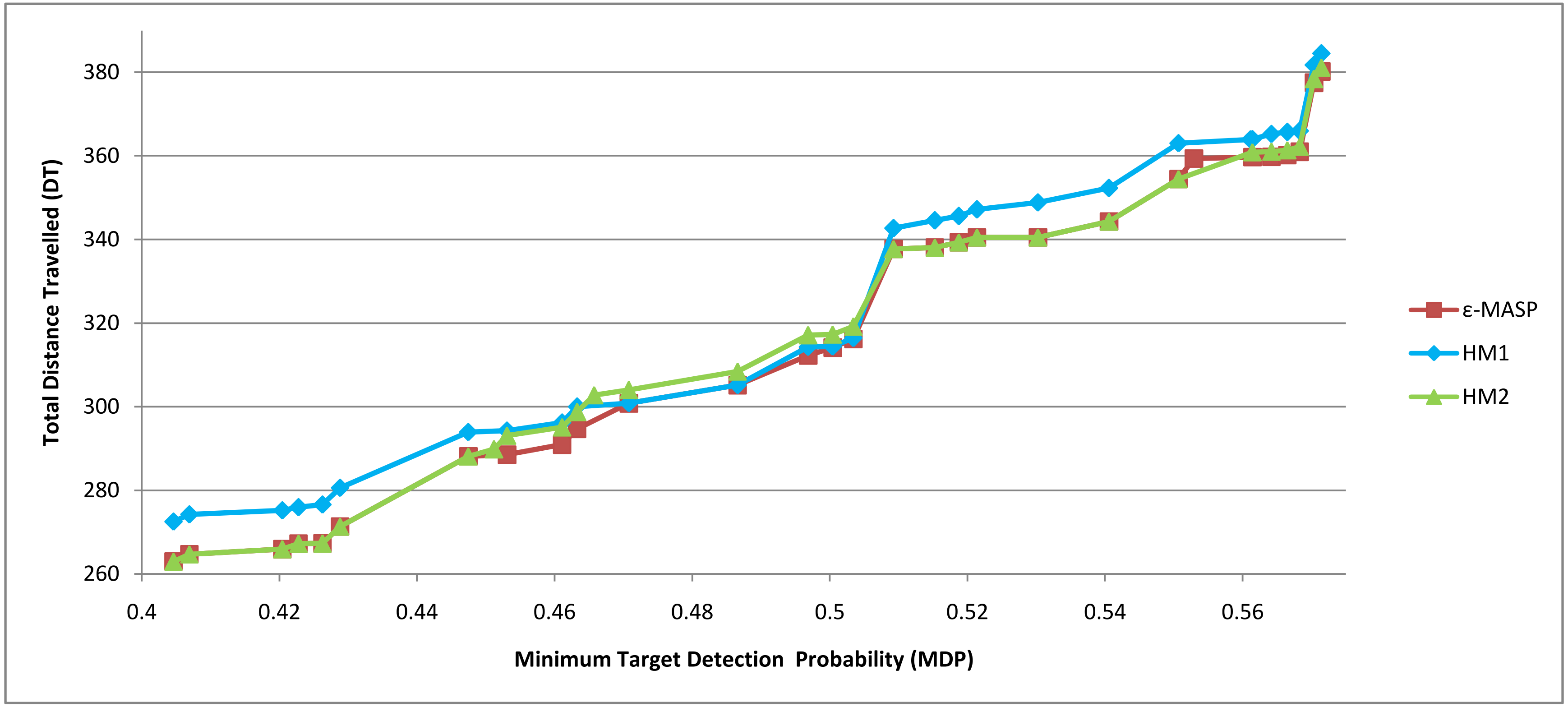

Figure 47 : The objective space of Ins5_30 showing solution sets found

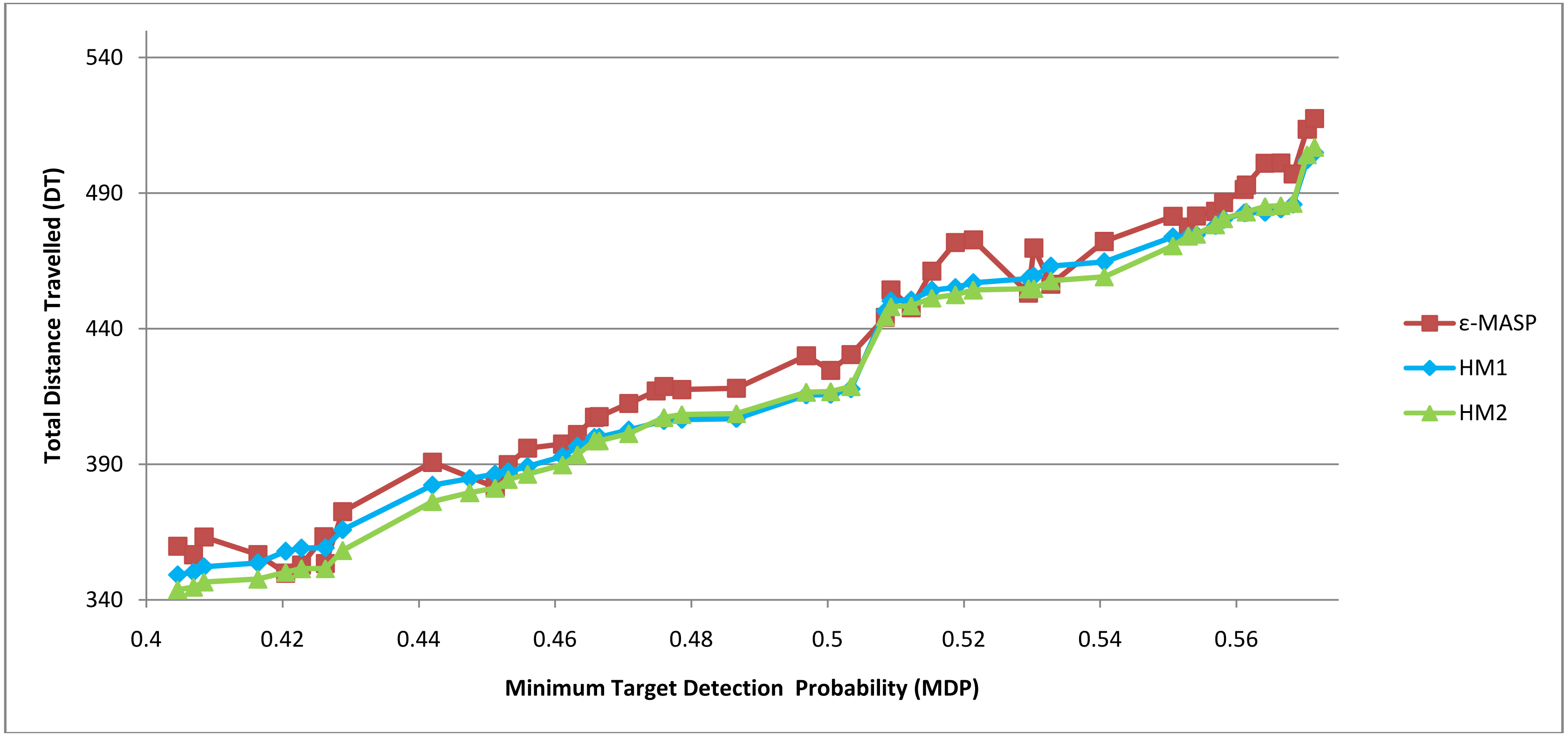

Figure 48 : The objective space of Ins5_40 showing solution sets found

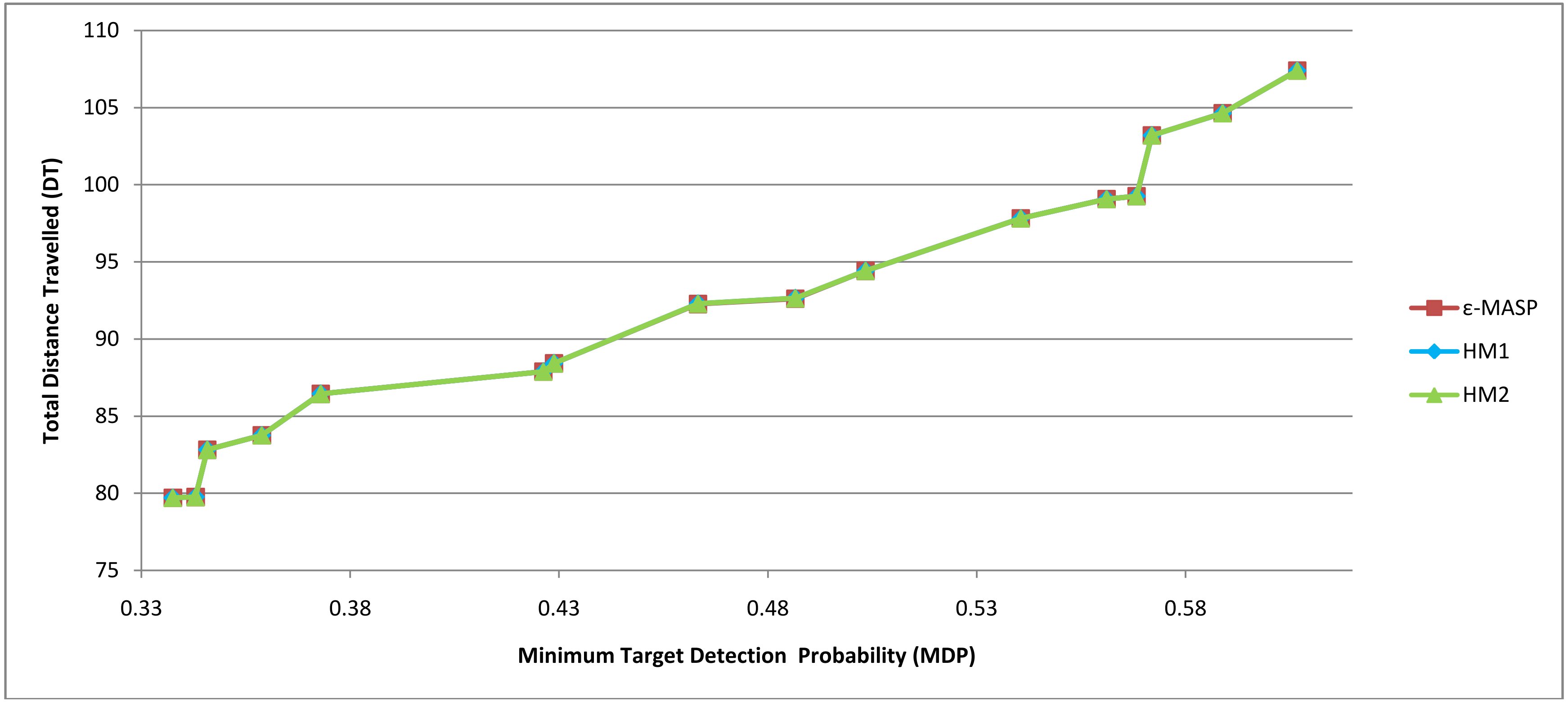

Figure 49 : The objective space of Ins6_5 showing solution sets found

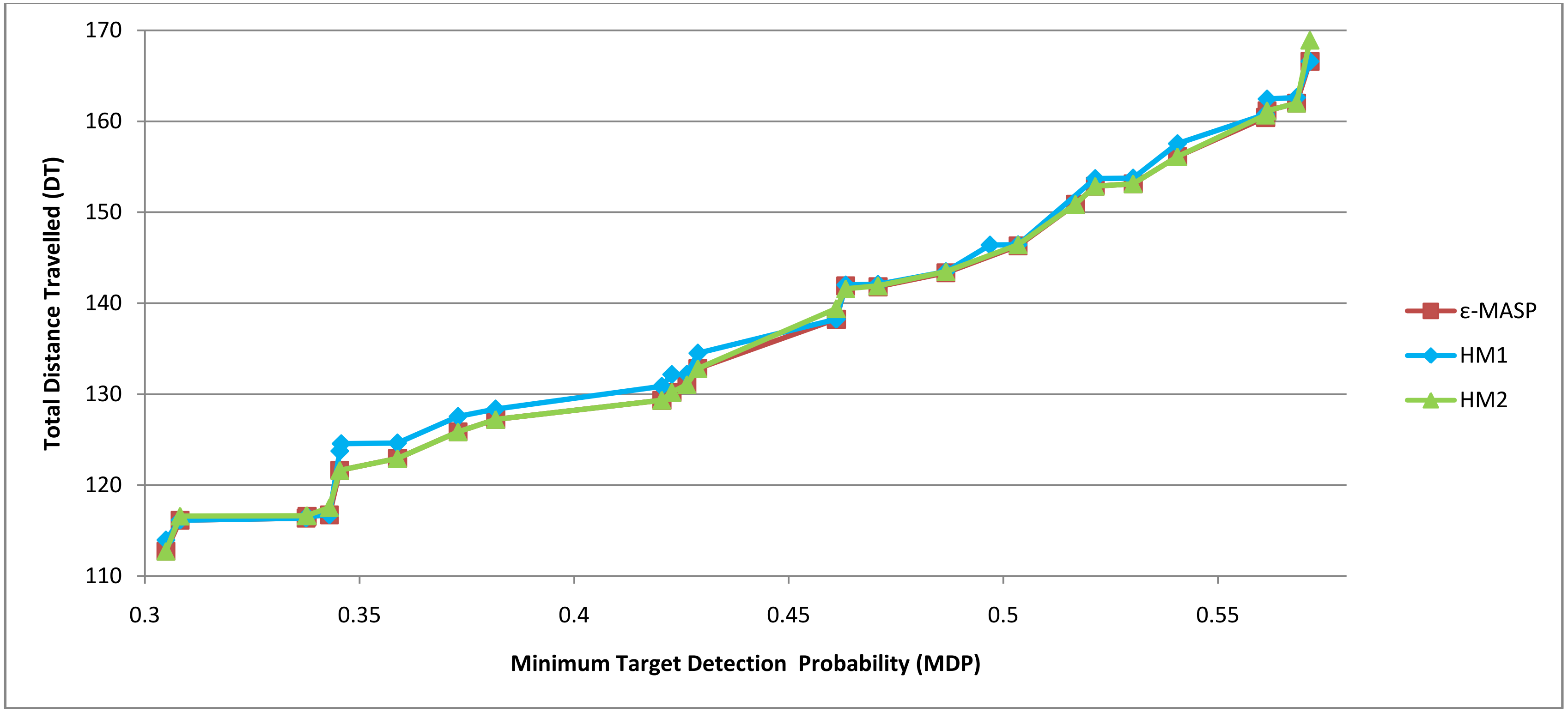

Figure 50 : The objective space of Ins6_10 showing solution sets found

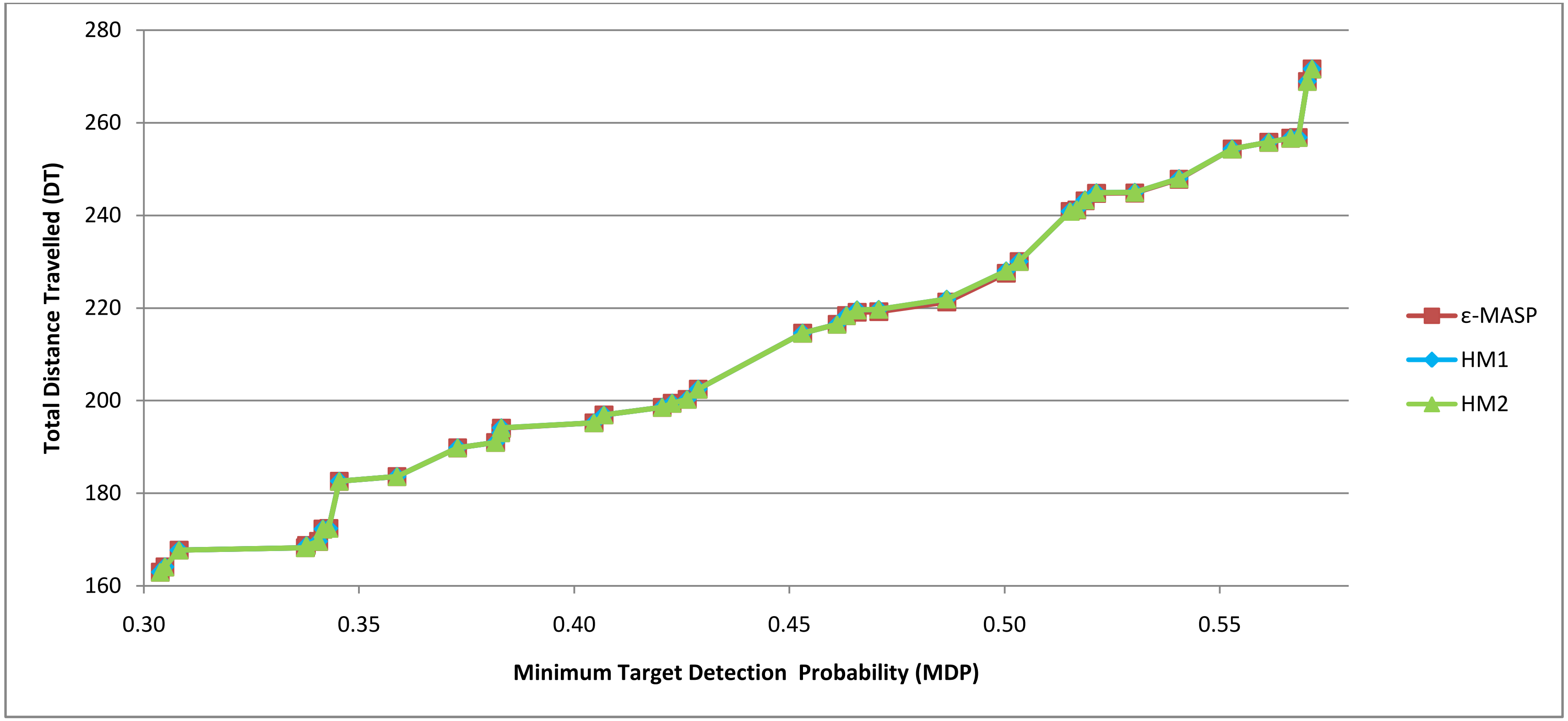

Figure 51 : The objective space of Ins6_20 showing solution sets found

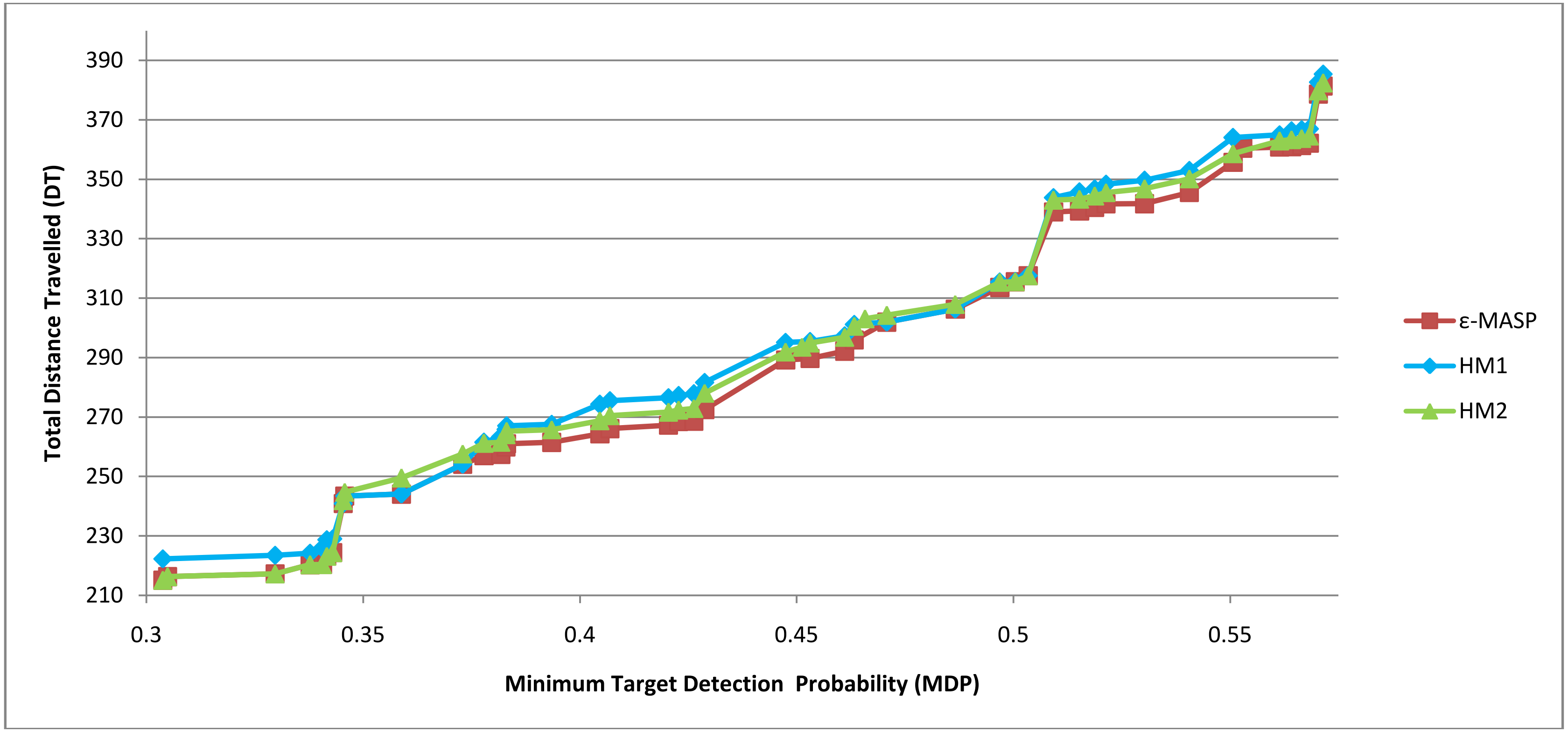

Figure 52 : The objective space of Ins6_30 showing solution sets found

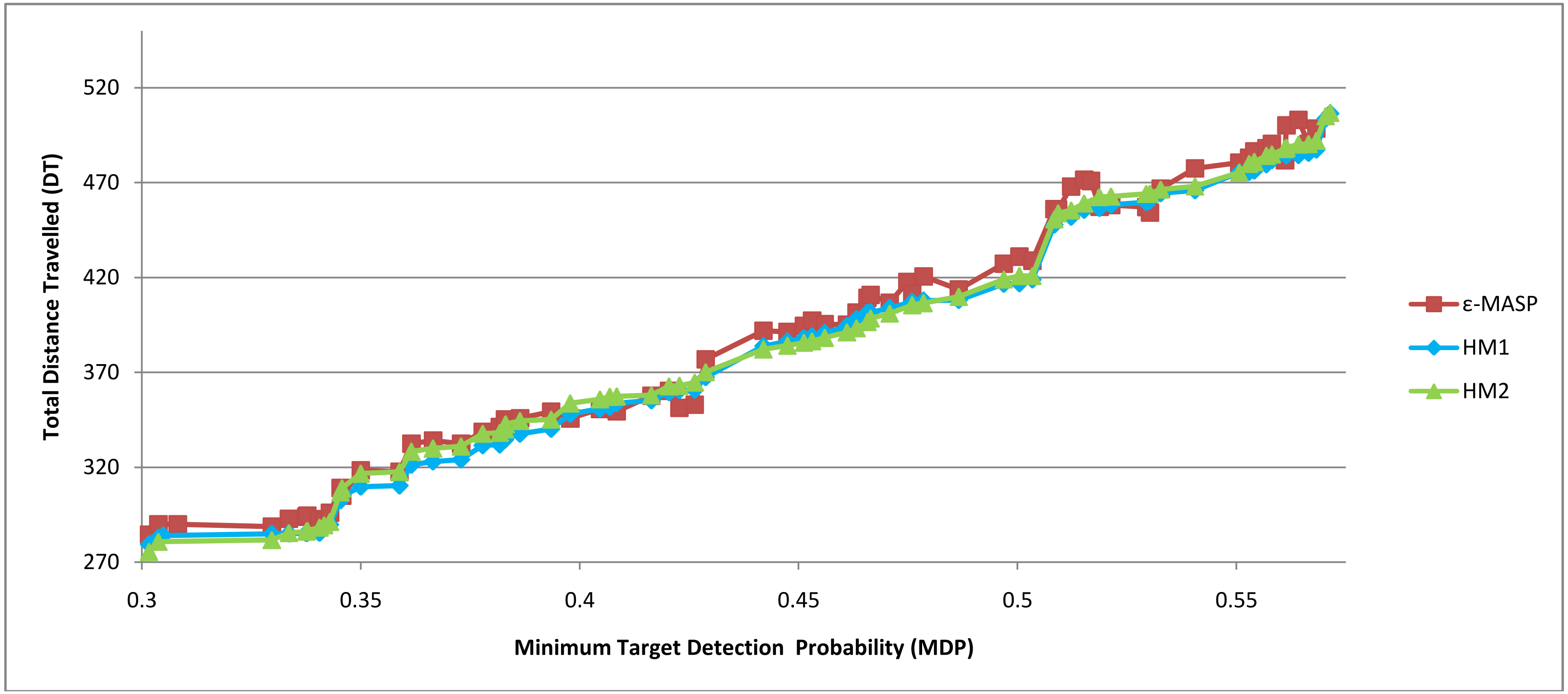

Figure 53 : The objective space of Ins6_40 showing solution sets found

APPENDIX D

COMPUTATION TIMES OF THE METHODS

Table 17 : Computation Times of The Methods

Problem	ε-MASP	HM1	HM2
Ins1_5	259.92	2.45	6.09
Ins1_10	824.21	3.11	139.73
Ins1_20	881.89	8.65	173.40
Ins1_30	10,382.30	40.79	704.43
Ins1_40	58,247.71	149.35	2,730.16
Ins2_5	326.30	2.25	4.95
Ins2_10	2,189.11	5.09	50.40
Ins2_20	1,108.99	26.74	116.88
Ins2_30	144,385.02	113.90	12,198.44
Ins2_40	154,008.49	867.90	3,266.60
Ins3_5	623.61	2.84	7.05
Ins3_10	5,259.27	6.34	111.57
Ins3_20	3,371.86	61.40	287.83
Ins3_30	34,612.14	319.89	1,333.05
Ins3_40*	17,918.37	1,846.81	1,846.26
Ins4_5	88.25	3.57	4.86
Ins4_10	2,418.27	2.67	34.17
Ins4_20	450.56	5.34	10.41
Ins4_30	18,812.07	12.13	408.09
Ins4_40	11,503.69	49.15	90.28
Ins5_5	199.21	2.56	4.44
Ins5_10	4,516.99	2.73	28.78
Ins5_20	2,524.56	12.11	16.07
Ins5_30	23,867.45	40.72	135.81
Ins5_40*	13,206.29	268.09	261.23
Ins6_5	329.62	2.51	4.98
Ins6_10	6,336.87	4.68	69.26
Ins6_20	3,664.23	25.49	31.32
Ins6_30	33,116.01	102.21	468.53
Ins6_40*	23,595.16	667.40	924.67

* In these problems, ε-MASP is solved with %5 relative gap.

APPENDIX E

PERCENT GD MEASURE AND NUMBER OF OPTIMAL SOLUTIONS FOUND OF THE METHODS

Table 18 : Percent GD Measure and Number of Optimal Solutions Found of the Methods

Problem	HM1			HM2		
	$GD_{dist}^{\%}$	Total # of solutions found	Total # of optimal solutions found	$GD_{dist}^{\%}$	Total # of solutions found	Total # of optimal solutions found
Ins1_5	0.00	12	12	0.00	12	12
Ins1_10	0.92	17	6	0.92	17	6
Ins1_20	0.00	26	26	0.00	26	26
Ins1_30	1.80	30	0	0.84	28	2
Ins1_40	0.89	37	0	0.59	39	5
Ins2_5	0.00	18	18	0.00	18	18
Ins2_10	0.70	30	12	0.18	29	18
Ins2_20	0.00	43	43	0.00	43	43
Ins2_30	1.43	50	0	0.30	50	14
Ins2_40	0.86	71	0	1.06	74	8
Ins3_5	0.00	23	23	0.00	23	23
Ins3_10	0.70	38	13	0.19	37	22
Ins3_20	0.02	57	52	0.04	57	49
Ins3_30	1.51	66	4	0.34	65	16
Ins4_5	0.00	7	7	0.00	7	7
Ins4_10	0.69	8	1	0.49	9	2
Ins4_20	0.04	14	11	0.01	13	12
Ins4_30	1.52	15	0	0.74	15	2
Ins4_40	1.19	24	0	2.24	23	1
Ins5_5	0.00	11	9	0.00	11	9
Ins5_10	0.70	16	2	0.18	17	11
Ins5_20	0.06	26	15	0.07	26	19
Ins5_30	1.84	28	1	0.42	28	14
Ins6_5	0.00	16	14	0.00	16	14
Ins6_10	0.60	25	6	0.16	25	15
Ins6_20	0.03	40	30	0.03	40	30
Ins6_30	1.78	43	8	0.95	44	7

APPENDIX F

NUMBER OF SOLUTIONS FOUND WHOSE %GAP IS IN A SPECIFIED INTERVAL

Table 19: Number of solutions found whose %GAP is in a specified interval in Instance 1

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
5	HM1	12	12	0	0	0	0	0
	HM2	12	12	0	0	0	0	0
10	HM1	17	6	3	5	3	0	0
	HM2	17	6	3	5	3	0	0
20	HM1	26	26	0	0	0	0	0
	HM2	26	26	0	0	0	0	0
30	HM1	30	0	4	15	8	2	1
	HM2	28	2	17	7	2	0	0
40	HM1	37	0	24	13	0	0	0
	HM2	39	5	24	10	0	0	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)
B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)
C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)
D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)
E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 20: Number of solutions found whose %GAP is in a specified interval in Instance 2

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
	HM1	18	18	0	0	0	0	0
5	HM2	18	18	0	0	0	0	0
	HM1	30	12	8	6	4	0	0
10	HM2	29	18	11	0	0	0	0
	HM1	43	43	0	0	0	0	0
20	HM2	43	43	0	0	0	0	0
	HM1	50	0	11	29	10	0	0
30	HM2	50	14	31	5	0	0	0
	HM1	71	0	54	17	0	0	0
40	HM2	74	8	36	21	9	0	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)
B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)
C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)
D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)
E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 21: Number of solutions found whose %GAP is in a specified interval in Instance 3

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
5	HM1	23	23	0	0	0	0	0
	HM2	23	23	0	0	0	0	0
10	HM1	38	13	14	8	3	0	0
	HM2	37	22	15	0	0	0	0
20	HM1	57	52	5	0	0	0	0
	HM2	57	49	8	0	0	0	0
30	HM1	66	4	13	34	9	6	0
	HM2	65	16	42	7	0	0	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)
B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)
C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)
D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)
E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 22: Number of solutions found whose %GAP is in a specified interval in Instance 4

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
5	HM1	7	7	0	0	0	0	0
	HM2	7	7	0	0	0	0	0
10	HM1	8	1	5	2	0	0	0
	HM2	9	2	6	1	0	0	0
20	HM1	14	11	3	0	0	0	0
	HM2	13	12	1	0	0	0	0
30	HM1	15	0	2	8	5	0	0
	HM2	15	2	7	6	0	0	0
40	HM1	24	0	9	15	0	0	0
	HM2	23	1	1	5	12	4	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)

B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)

C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)

D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)

E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 23: Number of solutions found whose %GAP is in a specified interval in Instance 5

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
	HM1	11	9	2	0	0	0	0
5	HM2	11	9	2	0	0	0	0
	HM1	16	2	10	4	0	0	0
10	HM2	17	11	5	1	0	0	0
	HM1	26	15	11	0	0	0	0
20	HM2	26	19	7	0	0	0	0
	HM1	28	1	4	13	4	6	0
30	HM2	28	14	7	7	0	0	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)

B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)

C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)

D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)

E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

Table 24: Number of solutions found whose %GAP is in a specified interval in Instance 6

Rectangle No	Method	# of solutions found	# of optimal solutions found	A	B	C	D	E
5	HM1	16	14	2	0	0	0	0
	HM2	16	14	2	0	0	0	0
10	HM1	25	6	12	7	0	0	0
	HM2	25	15	9	1	0	0	0
20	HM1	40	30	10	0	0	0	0
	HM2	40	30	10	0	0	0	0
30	HM1	43	8	1	15	12	7	0
	HM2	44	7	15	20	2	0	0

A : # of solutions found whose %GAP is between 0% and 1% (0 %< %GAP <=1%)
B : # of solutions found whose %GAP is between 1% and 2% (1 %< %GAP <=2%)
C : # of solutions found whose %GAP is between 2% and 3% (2 %< %GAP <=3%)
D : # of solutions found whose %GAP is between 3% and 4% (3 %< %GAP <=4%)
E : # of solutions found whose %GAP is more than 4% (4 %< %GAP)

APPENDIX G

HV AND HVR MEASURE OF THE METHODS

Table 25 : HV and HVR Measure of The Methods

	ε –MASP	HM1	HM2	HM1	HM2
Problem	HV(P)	HV(Q)	HV(Q)	HVR	HVR
Ins1_5	0.84	0.84	0.84	1.00	1.00
Ins1_10	1.74	1.62	1.62	0.93	0.93
Ins1_20	3.52	3.52	3.52	1.00	1.00
Ins1_30	5.35	4.56	5.02	0.85	0.94
Ins1_40	6.83	6.30	6.55	0.92	0.96
Ins2_5	2.31	2.31	2.31	1.00	1.00
Ins2_10	6.19	6.02	6.11	0.97	0.99
Ins2_20	12.13	12.13	12.13	1.00	1.00
Ins2_30	18.42	17.28	18.21	0.94	0.99
Ins2_40	26.17	25.23	25.11	0.96	0.96
Ins3_5	5.13	5.13	5.13	1.00	1.00
Ins3_10	11.84	11.61	11.71	0.98	0.99
Ins3_20	25.31	25.29	25.28	1.00	1.00
Ins3_30	40.86	39.27	40.51	0.96	0.99
Ins4_5	0.65	0.65	0.65	1.00	1.00
Ins4_10	0.68	0.59	0.65	0.88	0.96
Ins4_20	1.61	1.61	1.61	1.00	1.00
Ins4_30	2.28	1.85	2.05	0.81	0.90
Ins4_40	2.91	2.49	2.14	0.86	0.74
Ins5_5	1.78	1.78	1.78	1.00	1.00
Ins5_10	2.84	2.73	2.79	0.96	0.98
Ins5_20	7.11	7.08	7.08	1.00	1.00
Ins5_30	10.89	10.00	10.66	0.92	0.98
Ins6_5	3.65	3.65	3.65	1.00	1.00
Ins6_10	7.79	7.58	7.73	0.97	0.99
Ins6_20	16.15	16.13	16.13	1.00	1.00
Ins6_30	25.05	23.74	24.27	0.95	0.97

APPENDIX H

RESULTS FOR THE PROBLEMS NOT SOLVED OPTIMALLY

Table 26: Results for Ins3_40

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
348.17	0.3003	337.74	0.3003	-2.99	334.27	0.3003	-3.99
351.73	0.3039	345.24	0.3039	-1.85	340.09	0.3039	-3.31
351.19	0.3095	347.76	0.3095	-0.97	342.56	0.3095	-2.46
352.81	0.3114	348.80	0.3114	-1.14	343.59	0.3114	-2.61
349.51	0.3127	350.72	0.3127	0.35	345.51	0.3127	-1.14
354.11	0.3193	352.23	0.3193	-0.53	346.65	0.3193	-2.11
355.55	0.3227	356.50	0.3227	0.27	349.20	0.3227	-1.79
348.47	0.3246	357.87	0.3246	2.70	350.60	0.3246	0.61
359.77	0.3274	358.00	0.3275	2.09	350.70	0.3274	-2.52
350.68	0.3275	364.54	0.3297	-2.58	350.70	0.3275	0.00
374.19	0.3297	380.86	0.3408	0.11	357.35	0.3297	-4.50
380.43	0.3408	383.36	0.3454	*	374.50	0.3408	-1.56
391.26	0.3486	385.03	0.3486	-1.59	377.46	0.3454	*
392.04	0.3502	385.93	0.3502	-1.56	379.14	0.3486	
390.94	0.3526	387.85	0.3526	-0.79	383.58	0.3502	-2.16
399.76	0.3570	391.59	0.3570	-2.04	385.51	0.3526	-1.39
399.08	0.3588	395.25	0.3588	-0.96	389.18	0.3570	-2.65
402.08	0.3609	398.85	0.3609	-0.81	392.84	0.3588	-1.56
406.26	0.3616	398.94	0.3616	-1.80	397.75	0.3609	-1.08
409.54	0.3653	401.58	0.3653	-1.94	397.84	0.3616	-2.07
416.22	0.3687	404.82	0.3697	-2.93	400.48	0.3653	-2.21
417.06	0.3697	405.40	0.3719	-3.19	405.31	0.3697	-2.82
418.77	0.3719	405.63	0.3789	-3.12	406.64	0.3719	-2.90
418.69	0.3789	414.55	0.3878	-3.05	406.87	0.3789	-2.82
427.61	0.3878	414.75	0.3909	-2.93	415.61	0.3878	-2.81
427.28	0.3909	416.80	0.3935	-2.59	415.80	0.3909	-2.69
427.91	0.3935	445.45	0.3978	-0.32	417.70	0.3935	-2.39

Table 26: Results for Ins3_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
446.86	0.3978	449.09	0.3986	-3.48	443.43	0.3978	-0.77
465.29	0.3986	449.41	0.4012	-1.92	447.07	0.3986	-3.92
458.21	0.4012	453.01	0.4038	1.11	447.40	0.4012	-2.36
448.03	0.4038	454.07	0.4069	-3.65	450.27	0.4038	0.50
471.25	0.4069	455.68	0.4092	-3.41	451.49	0.4069	-4.19
471.78	0.4092	457.32	0.4164	-1.09	453.11	0.4092	-3.96
462.35	0.4164	458.18	0.4171	-0.56	453.45	0.4164	-1.92
460.76	0.4171	461.94	0.4193	-0.52	453.66	0.4171	-1.54
464.33	0.4193	463.35	0.4262	-2.61	456.34	0.4193	-1.72
475.75	0.4262	472.53	0.4353	-0.77	457.75	0.4262	-3.78
476.20	0.4353	472.61	0.4373	-2.69	469.51	0.4353	-1.40
485.66	0.4373	473.12	0.4384	-2.73	473.84	0.4373	-2.43
486.42	0.4384	476.45	0.4409	-2.42	474.36	0.4384	-2.48
488.25	0.4409	478.65	0.4420	-1.93	477.86	0.4409	-2.13
488.06	0.4420	481.22	0.4447	-2.39	480.12	0.4420	-1.63
493.02	0.4447	481.29	0.4450	-1.92	481.94	0.4450	-1.78
490.70	0.4450	481.43	0.4475	*	483.99	0.4475	*
498.74	0.4495	482.71	0.4495	-3.21	484.34	0.4495	-2.89
499.61	0.4512	484.48	0.4512	-3.03	485.15	0.4512	-2.89
519.23	0.4531	500.99	0.4531	-3.51	502.64	0.4531	-3.19
501.62	0.4541	503.68	0.4541	0.41	505.33	0.4541	0.74
518.75	0.4545	503.96	0.4545	-2.85	507.07	0.4545	-2.25
508.46	0.4596	511.13	0.4596	0.53	512.11	0.4596	0.72
524.00	0.4700	511.62	0.4700	-2.36	513.45	0.4700	-2.01
532.58	0.4715	522.76	0.4715	-1.84	522.35	0.4715	-1.92
541.44	0.4742	523.75	0.4742	-3.27	523.35	0.4742	-3.34
539.16	0.4776	534.07	0.4776	-0.94	529.79	0.4776	-1.74
554.51	0.4791	536.84	0.4791	-3.19	532.56	0.4791	-3.96
553.27	0.4866	540.56	0.4866	-2.30	536.43	0.4866	-3.04
565.28	0.4891	547.25	0.4891	-3.19	543.32	0.4891	-3.88
565.07	0.4924	548.62	0.4924	-2.91	544.71	0.4924	-3.60
558.23	0.4969	548.95	0.4969	-1.66	545.03	0.4969	-2.36
550.97	0.4983	549.04	0.4996	-2.70	545.13	0.4996	-3.39

Table 26: Results for Ins3_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	MDP	DT	MDP	%GAP	DT	MDP	%GAP
568.07	0.4984	549.65	0.5004	-1.63	545.54	0.5004	-2.37
564.26	0.4996	552.01	0.5023	-3.37	547.89	0.5023	*
558.78	0.5004	555.22	0.5034	-3.39	551.10	0.5034	-4.11
571.24	0.5023	566.52	0.5060	-3.12	560.04	0.5060	-4.23
574.70	0.5034	570.12	0.5105	-2.85	562.09	0.5105	-4.22
584.80	0.5060	576.28	0.5209	-3.54	573.32	0.5209	-4.03
586.84	0.5105	579.72	0.5276	-3.10	576.75	0.5276	-3.59
597.42	0.5209	589.69	0.5312	-3.79	587.77	0.5312	-4.10
598.26	0.5276	594.92	0.5327	-2.95	594.86	0.5327	-2.95
612.89	0.5312	596.59	0.5336	-1.47	596.53	0.5336	-1.48
612.97	0.5327	598.26	0.5366	-1.09	598.20	0.5366	-1.10
605.47	0.5336	598.82	0.5397	-1.52	599.76	0.5397	-1.37
604.83	0.5366	601.16	0.5406	-2.49	602.10	0.5406	-2.33
608.08	0.5397	604.79	0.5439	-1.55	604.83	0.5412	-1.52
616.48	0.5406	604.98	0.5442	-3.33	604.90	0.5439	-1.54
614.15	0.5412	614.71	0.5459	-2.40	604.99	0.5442	-3.32
614.33	0.5439	615.07	0.5476	-2.63	612.62	0.5459	-2.73
625.80	0.5442	616.18	0.5478	-2.18	613.37	0.5476	-2.90
629.80	0.5459	626.08	0.5494	-3.42	614.48	0.5478	-2.45
631.70	0.5476	628.18	0.5507	-2.62	623.32	0.5494	-3.85
629.90	0.5478	641.58	0.5529	-3.42	623.71	0.5507	-3.31
648.28	0.5494	644.17	0.5569	-3.00	637.93	0.5529	-3.97
645.08	0.5507	644.80	0.5600	-3.28	640.37	0.5569	-3.57
664.31	0.5529	645.93	0.5654	-3.50	641.00	0.5600	-3.85
664.07	0.5569	655.62	0.5657	-3.34	642.31	0.5654	-4.04
666.64	0.5600	659.10	0.5709	*	650.24	0.5657	-4.13
669.38	0.5654	659.64	0.5715	-2.99	653.73	0.5709	*
678.28	0.5657	659.98	0.5756	-3.04	654.13	0.5715	-3.80
679.96	0.5715	662.28	0.5809	-2.51	654.46	0.5756	-3.85
680.67	0.5756	663.90	0.5823	0.09	662.81	0.5809	-2.43
679.31	0.5809	667.14	0.5831	-2.87	664.60	0.5823	0.20
663.28	0.5823	669.11	0.5862	-2.22	671.63	0.5831	-2.22
686.85	0.5831	670.10	0.5889	-0.50	673.60	0.5865	1.24

Table 26: Results for Ins3_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
684.33	0.5862	671.97	0.5903	-2.43	673.87	0.5889	0.06
665.36	0.5865	673.57	0.5916	-3.73	676.86	0.5903	-1.72
673.45	0.5889	675.34	0.5934	-2.34	678.46	0.5916	-3.03
688.72	0.5903	691.91	0.5966	-0.23	679.65	0.5934	-1.72
699.68	0.5916				696.44	0.5966	*
700.04	0.5924						
691.52	0.5934						
693.51	0.5966						

* For that solution, there is no corresponding optimal solution that has same MPD value. Therefore, %GAP cannot be calculated.

Table 27: Results for Ins5_40

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
359.83	0.4046	349.26	0.4046	-2.94	343.69	0.4046	-4.49
356.66	0.4069	350.28	0.4069	-1.79	344.71	0.4069	-3.35
363.25	0.4084	352.19	0.4084	-3.04	346.62	0.4084	-4.58
356.74	0.4164	353.60	0.4164	-0.88	347.66	0.4164	-2.55
349.86	0.4204	357.88	0.4204	2.29	350.20	0.4204	0.10
352.86	0.4228	359.14	0.4228	1.78	351.52	0.4228	-0.38
363.33	0.4260	359.27	0.4262	1.65	351.60	0.4262	-0.52
353.45	0.4262	365.87	0.4288	-1.80	358.20	0.4288	-3.86
372.57	0.4288	382.34	0.4420	-2.15	376.31	0.4420	-3.70
390.76	0.4420	384.76	0.4475	*	379.54	0.4475	*
381.57	0.4512	386.41	0.4512	1.27	381.20	0.4512	-0.10
389.88	0.4531	387.32	0.4531	-0.66	384.41	0.4531	-1.40
395.97	0.4559	389.26	0.4559	-1.70	386.37	0.4559	-2.43
397.47	0.4611	392.92	0.4611	-1.14	389.84	0.4611	-1.92
400.96	0.4633	396.58	0.4633	-1.09	393.50	0.4633	-1.86
407.34	0.4657	399.94	0.4657	-1.82	398.64	0.4657	-2.14
407.55	0.4665	400.05	0.4665	-1.84	398.75	0.4665	-2.16

Table 27: Results for Ins5_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
412.43	0.4708	402.65	0.4708	-2.37	401.35	0.4708	-2.69
417.05	0.4748	406.06	0.4759	-3.03	407.24	0.4759	-2.74
418.73	0.4759	406.56	0.4786	-2.63	408.40	0.4786	-2.19
417.56	0.4786	406.85	0.4866	-2.68	408.70	0.4866	-2.24
418.06	0.4866	415.54	0.4969	-3.37	416.59	0.4969	-3.13
430.04	0.4969	415.74	0.5004	-2.10	416.79	0.5004	-1.85
424.63	0.5004	417.78	0.5034	-2.95	418.67	0.5034	-2.74
430.48	0.5034	446.52	0.5084	0.52	444.46	0.5084	0.05
444.23	0.5084	450.25	0.5093	-0.91	448.19	0.5093	-1.36
454.37	0.5093	450.59	0.5122	0.64	448.54	0.5122	0.18
447.71	0.5122	454.19	0.5153	-1.53	451.39	0.5153	-2.14
461.24	0.5153	455.23	0.5187	-3.51	452.59	0.5187	-4.08
471.82	0.5187	456.92	0.5214	-3.36	454.28	0.5214	-3.92
472.83	0.5214	458.48	0.5294	1.18	454.72	0.5294	0.35
453.13	0.5294	459.38	0.5303	-2.21	454.91	0.5303	-3.16
469.77	0.5303	463.15	0.5327	1.47	457.74	0.5327	0.29
456.43	0.5327	464.61	0.5406	-1.61	459.20	0.5406	-2.76
472.21	0.5406	473.91	0.5507	-1.57	470.59	0.5507	-2.26
481.46	0.5507	474.01	0.5529	-0.74	474.18	0.5529	-0.71
477.56	0.5529	474.70	0.5541	-1.44	474.88	0.5541	-1.40
481.62	0.5541	477.98	0.5569	-1.10	478.35	0.5569	-1.02
483.29	0.5569	480.22	0.5581	-1.31	480.59	0.5581	-1.23
486.58	0.5581	482.71	0.5611	-1.77	483.08	0.5614	-2.02
491.40	0.5611	482.75	0.5614	-2.09	485.09	0.5642	-3.19
493.06	0.5614	482.90	0.5642	-3.62	485.46	0.5665	-3.13
501.05	0.5642	484.13	0.5665	-3.40	486.27	0.5683	-2.18
501.14	0.5665	485.87	0.5683	-2.26	504.21	0.5704	-1.82
497.09	0.5683	502.28	0.5704	-2.19	506.93	0.5715	-2.05
513.55	0.5704	504.99	0.5715				
517.55	0.5715						

* For that solution, there is no corresponding optimal solution that has same MPD value. Therefore, %GAP cannot be calculated.

Table 28 : Results for Ins6_40

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
284.47	0.3016	279.20	0.3016	-1.85	275.38	0.3016	-3.19
289.89	0.3038	282.80	0.3038	-2.45	280.75	0.3038	-3.15
289.95	0.3082	283.98	0.3049	*	281.74	0.3297	-2.38
288.60	0.3297	284.87	0.3297	-1.29	285.33	0.3336	-2.55
292.79	0.3336	284.99	0.3336	-2.66	286.08	0.3375	-2.64
293.84	0.3375	285.25	0.3375	-2.93	286.57	0.3378	-2.67
294.42	0.3378	285.30	0.3378	-3.10	288.29	0.3406	-1.48
292.61	0.3406	285.37	0.3406	-2.48	289.45	0.3416	-0.64
291.32	0.3416	288.18	0.3416	-1.08	291.20	0.3430	-1.64
296.06	0.3430	289.81	0.3430	-2.11	306.79	0.3454	-0.73
309.05	0.3454	302.48	0.3454	-2.12	308.90	0.3457	1.26
305.06	0.3457	304.66	0.3457	-0.13	316.65	0.3500	-0.54
318.37	0.3500	309.69	0.3500	-2.72	317.65	0.3588	0.03
317.56	0.3588	310.26	0.3588	-2.30	328.19	0.3616	-1.26
332.37	0.3616	321.23	0.3616	-3.35	329.89	0.3665	-1.24
334.03	0.3665	322.93	0.3665	-3.32	330.97	0.3729	-0.44
332.44	0.3729	324.03	0.3729	-2.53	337.74	0.3779	-0.24
338.56	0.3779	331.52	0.3779	-2.08	338.14	0.3817	-0.89
341.19	0.3817	331.88	0.3817	-2.73	340.08	0.3829	-1.47
345.16	0.3829	334.46	0.3829	-3.10	342.78	0.3831	-0.19
343.45	0.3831	335.62	0.3831	-2.28	344.46	0.3864	-0.39
345.82	0.3864	337.61	0.3864	-2.37	345.17	0.3935	-1.16
349.22	0.3935	340.18	0.3935	-2.59	353.63	0.3978	2.29
345.72	0.3978	348.09	0.3978	0.68	355.81	0.4046	1.46
350.70	0.4046	350.73	0.4046	0.01	357.16	0.4069	1.41
352.20	0.4069	351.76	0.4069	-0.12	357.28	0.4084	2.25
349.43	0.4084	353.67	0.4084	1.21	358.00	0.4164	0.10
357.65	0.4164	355.13	0.4164	-0.70	362.73	0.4204	0.68
360.29	0.4204	359.42	0.4204	-0.24	362.92	0.4228	3.31
351.29	0.4228	360.56	0.4228	2.64	364.64	0.4262	3.32
352.91	0.4262	360.69	0.4262	2.20	370.20	0.4288	-1.76
376.81	0.4288	367.38	0.4288	-2.50	382.12	0.4420	-2.51
391.95	0.4420	383.85	0.4420	-2.07	384.10	0.4475	-1.84

Table 28 : Results for Ins6_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
391.29	0.4475	386.26	0.4475	-1.29	385.71	0.4512	-2.23
394.52	0.4512	387.90	0.4512	-1.68	386.43	0.4531	-2.70
397.15	0.4531	388.80	0.4531	-2.10	388.18	0.4559	-1.81
395.34	0.4559	390.72	0.4559	-1.17	391.21	0.4611	-0.97
395.05	0.4611	394.40	0.4611	-0.16	393.34	0.4633	-2.03
401.48	0.4633	398.05	0.4633	-0.86	396.59	0.4657	-3.10
409.26	0.4657	401.31	0.4657	-1.94	398.52	0.4665	-2.99
410.79	0.4665	401.43	0.4665	-2.28	401.02	0.4708	-1.40
406.73	0.4708	403.93	0.4708	-0.69	405.41	0.4759	-1.47
417.68	0.4748	407.15	0.4759	-1.04	406.47	0.4786	-3.35
411.44	0.4759	407.66	0.4786	-3.07	409.79	0.4866	-0.95
420.56	0.4786	408.03	0.4866	-1.37	419.03	0.4969	-1.92
413.71	0.4866	416.62	0.4969	-2.49	420.75	0.5004	-2.37
427.25	0.4969	416.82	0.5004	-3.28	420.83	0.5034	-1.87
430.97	0.5004	418.87	0.5034	-2.32	450.71	0.5084	-1.18
428.83	0.5034	447.76	0.5084	-1.82	453.88	0.5093	-0.43
456.07	0.5084	451.57	0.5093	-0.94	455.43	0.5122	-2.66
455.86	0.5093	451.93	0.5122	-3.41	459.02	0.5153	-2.64
467.90	0.5122	455.52	0.5153	-3.38	462.38	0.5187	1.12
471.47	0.5153	456.57	0.5187	-0.15	462.76	0.5214	1.01
470.89	0.5168	458.35	0.5214	0.05	464.07	0.5294	1.53
457.27	0.5187	459.61	0.5294	0.55	464.27	0.5303	2.23
458.13	0.5214	460.53	0.5303	1.41	466.33	0.5327	-0.11
457.08	0.5294	464.32	0.5327	-0.54	468.03	0.5406	-1.98
454.14	0.5303	465.82	0.5406	-2.45	475.34	0.5507	-1.07
466.85	0.5327	475.40	0.5507	-1.05	479.88	0.5529	-0.67
477.51	0.5406	475.52	0.5529	-1.57	480.90	0.5541	-1.14
480.46	0.5507	476.31	0.5541	-2.08	484.12	0.5569	-0.83
483.12	0.5529	479.53	0.5569	-1.77	484.92	0.5581	-1.11
486.43	0.5541	481.79	0.5581	-1.75	487.75	0.5611	1.26
488.17	0.5569	484.25	0.5614	-3.17	488.36	0.5614	-2.35
490.36	0.5581	484.40	0.5642	-3.70	490.14	0.5642	-2.56
481.70	0.5611	485.60	0.5665	-0.99	490.34	0.5665	-0.02
500.12	0.5614	487.31	0.5683	-2.25	492.09	0.5683	-1.29

Table 28 : Results for Ins6_40 (continued)

ε-MASP (5% Relative Gap)		HM1			HM2		
DT	**MDP**	**DT**	**MDP**	**%GAP**	**DT**	**MDP**	**%GAP**
503.03	0.5642	503.65	0.5704	-1.06	505.08	0.5704	-0.77
490.45	0.5665	506.34	0.5715	-1.49	506.77	0.5715	-1.41
498.51	0.5683						
509.02	0.5704						
514.01	0.5715						

* For that solution, there is no corresponding optimal solution that has same MPD value. Therefore, %GAP cannot be calculated.