GENERATING REPRESENTATIVE NONDOMINATED POINT SUBSETS IN MULTI-OBJECTIVE INTEGER PROGRAMS

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

BY

GÖKHAN CEYHAN

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

AUGUST 2014

Approval of the thesis:

GENERATING REPRESENTATIVE NONDOMINATED POINT SUBSETS IN MULTI-OBJECTIVE INTEGER PROGRAMS

submitted by **GÖKHAN CEYHAN** 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** _______________

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

Assist. Prof. Dr. Banu Lokman
Co-supervisor, **Industrial Engineering Dept., TED University** _______________

Examining Committee Members:

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

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

Assist. Prof. Dr. Banu Lokman
Industrial Engineering Dept., TED University _______________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering Dept., METU _______________

Assist. Prof. Dr. Ceren Tuncer Şakar
Industrial Engineering Dept., Hacettepe University _______________

Date: _______________

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: GÖKHAN CEYHAN

Signature :

ABSTRACT

GENERATING REPRESENTATIVE NONDOMINATED POINT SUBSETS IN MULTI-OBJECTIVE INTEGER PROGRAMS

Ceyhan, Gökhan
M.S., Department of Industrial Engineering
Supervisor : Prof. Dr. Murat Köksalan
Co-Supervisor : Assist. Prof. Dr. Banu Lokman

August 2014, 73 pages

In this thesis, we study generating a subset of all nondominated points of multi-objective integer programs in order to represent the nondominated frontier. Our motivation is based on the fact that generating all nondominated points of a multi-objective integer program is neither practical nor useful. The computational burden could be prohibitive and the resulting set could be huge. Instead of finding all nondominated points, we develop algorithms to generate a small representative subset of nondominated points. In order to assess the quality of representative subsets, we conduct computational experiments on randomly generated instances of combinatorial optimization problems and show that the algorithms work well.

Keywords: Multi-objective Integer Program, Nondominated Point, Representative Set

ÖZ

ÇOK AMAÇLI TAMSAYI PROGRAMLARINDA TEMSİLİ BASKIN NOKTA
ALT KÜMELERİNİN ÜRETİLMESİ

Ceyhan, Gökhan
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Murat Köksalan
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Banu Lokman

Ağustos 2014 , 73 sayfa

Bu tezde, çok amaçlı tamsayı programlarında baskın yüzeyi temsil etmek üzere tüm baskın noktaların bir alt kümesinin üretilmesi üzerine çalıştık. Motivasyonumuz çok amaçlı tamsayı programları için tüm baskın noktaları üretmenin pratik ve yararlı olmaması gerçeğine dayanmaktadır. Hesaplama zorluğu ve elde edilen kümenin büyüklüğü çok fazla olabilmektedir. Tüm baskın noktaları bulmak yerine, baskın noktaların küçük bir temsili alt kümesini üretmek için algoritmalar geliştiriyoruz. Temsili alt kümelerin kalitesini değerlendirmek için rassal olarak üretilmiş çok amaçlı bileşi problemleri üzerinde deneyler yaptık ve algoritmalarımızın iyi çalıştığını gösterdik.

Anahtar Kelimeler: Çok Amaçlı Tamsayı Programı, Baskın Nokta, Temsili Küme

To my newborn niece, Ece

ACKNOWLEDGMENTS

I would like to thank my supervisors Murat Köksalan and Banu Lokman for their brilliant guidance and comments throughout this study. I also thank Sinem Mutlu who has always believed in me and encouraged me to make things better.

I hope that anyone reading these pages finds useful parts of it for his or her research.

This work is supported by TÜBITAK-BİDEB National Graduate Scholarship Programme 2228.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DM	Decision Maker
MCDM	Multi-criteria Decision Making
MOLP	Multi-objective Linear Programming
MOIP	Multi-objective Integer Programming
MOMIP	Multi-objective Mixed-integer Programming
MILP	Mixed Integer Linear Programming
MOEA	Multi-objective Evolutionary Algorithm
MOCO	Multi-objective Combinatorial Optimization
MOKP	Multi-objective Knapsack Problem
MOAP	Multi-objective Assignment Problem

CHAPTER 1

INTRODUCTION

Decision makers (DMs) are usually faced with multiple objectives in their problem solving processes. If those objectives are conflicting with each other, then there is no single optimal solution for the problem at hand. Then, the general procedure is to generate "good" solutions for the problem and then let the DMs choose the most preferred solution. The "good" solutions are the ones which cannot be improved at one objective without sacrificing from another objective.

In multi-criteria decision making literature, these "good" solutions are defined as efficient solutions in the decision space. The image of an efficient solution in the objective space is called nondominated point and the set of nondominated points is referred as the nondominated frontier. More than one efficient solution may correspond to the same nondominated point. Therefore, the cardinality of the set of efficient solutions is always greater than or equal to the cardinality of the set of nondominated points. It is one of the main research areas in multi-criteria decision making to find these sets.

Solution approaches for multi-objective problems are classified into three main categories based on the articulation of the DMs' preferences. The first one is the "a priori mode" in which preference information is received from the DM before the phase of solution generation. The preference information can be in form of objective weights or aspiration levels for each cobjective.

In "a posteriori mode", nondominated frontier is generated first and presented to the DM. Then, among the nondominated points, the DM chooses the best solution based on his/her preferences. The last category is "interactive mode" which processes the preference information of the DMs during the solution phase.

We study multi-objective integer programming (MOIP) problems with any number of objectives. Generating the nondominated frontiers of MOIP problems is a difficult task since the number of nondominated points may be exponential in problem size (see [6]) and the computational complexity increases substantially.

Some nondominated points can be found by giving weights to the objectives and converting the problem to a single-objective problem. These points are called supported nondominated points and all of them can be generated by changing the weights assigned to the objectives although how to change the weights for more than two objectives is not straightforward. However, there exist nondominated points which cannot be generated by solving a weighted sum objective and those points are called unsupported nondominated points. Unsupported nondominated points compose a large portion of the nondominated frontier and it is a difficult task to generate them especially for more than two objectives.

Generating all nondominated points is theoretically appealing but may not be practical in real life decision making. In addition to the computational burden of finding the nondominated frontier, the DM needs to find a way to filter the highly preferred solutions. Therefore, it can be considered as an impractical procedure since finding even a single nondominated point may not be easy.

Due to above considerations, heuristic methods are widely used in the literature to approximate the nondominated frontier. Although the heuristics may approximate the nondominated frontier well in a reasonable amount of time, they also present a large number of solutions to the DM which are not necessarily nondominated.

In our study, we develop approaches to generate a small subset of nondominated points which can be easily analyzed by the DMs. In order to assess the quality of the representative subset, we also generate all nondominated points using Lokman and Köksalan [19] and use the quality measure defined by Masin and Bukchin [20]. Masin and Bukchin [20] name their measure as the *diversity* measure which is to be maximized for a minimization type problem. We consider a maximization type problem throughout this study where the smaller diversity values are the better based on the definition of this measure. However, due to the underlying positive meaning of the diversity word, minimizing the diversity of a subset is counterintuitive. Therefore, we call the diverstiy measure of Masin and Bukchin [20] as the *coverage gap* of a subset and minimize it to obtain higher quality subsets.

We only generate a subset of nondominated points to represent the nondominated frontier. We call the nondominated points in the subset as the representative points and the set as the representative subset. Each nondominated point is represented by its closest representative point. The nondominated point represented the worst defines the quality of the subset. The maximum difference in the objectives where this point is better than its representative point gives us the coverage gap of the representative subset.

We develop three algorithms to generate representative subsets for MOIP problems. Our first algorithm is an improvement on the algorithms developed by Masin and Bukchin [20] (Diversity Maximization Algorithm (DMA)), and Sylva and Crema [30]. The algorithms presented in these studies generate subsets having the same coverage gap provided that both algorithms start at the same initial point and the models solved have unique optimal solutions. The algorithms iteratively generate nondominated points such that a subset of diverse nondominated points is available at any iteration of the algorithm. However, the computational complexity increases since binary variables and constraints are added to the model for each nondominated point generated.

In order to handle this computational complexity, we develop a search procedure that finds the next nondominated point without additional binary variables. We use the nondominated subspace enumeration technique of Lokman and Köksalan [19] and search each subspace for the most diverse nondominated point. The improvement is not in the quality of the subsets but in the solution time of the algorithm. If the initial point is the same, then our algorithm generates subsets having the same coverage gap with the other algorithms provided that there exist unique optimal solutions to the models solved at each iteration. However, the complexity of the models does not increase in our algorithm as the cardinality of the subset increases. Therefore, the solution time does not increase exponentially which is the case in the previous two algorithms. Throughout the algorithm, the DM can monitor the cardinality and the coverage gap of the subset, and stop the algorithm whenever he/she is satisfied with the results. We name this algorithm as Algorithm 1.

In our second algorithm, Territory Defining Algorithm (TDA), we generate a representative subset to ensure a coverage gap value specified a priori. Since the desired coverage gap value is known, we define territories around the available nondominated points such that we guarantee that the nondominated points in these regions do not exceed the desired coverage gap value. Therefore, we do not conduct search in these regions. In TDA, we ensure achieving the desired coverage gap value with less computation than our first algorithm. Use of TDA is suitable for the cases where the DM can give a coverage gap value threshold before the solution stage.

Our last algorithm, Surface Projection Algorithm (SPA), is based on the idea of approximating the nondominated frontier by hypersurfaces. This idea is used in Köksalan [13] for bi-criteria scheduling problem and applicable for any multi-objective combinatorial optimization (MOCO) problem with any number of objectives. Köksalan and Lokman [14] use the same idea to converge the preferred regions in the objective space. Even in the interactive algorithm, many nondominated points have to be generated until finding the most preferred solution. However, generating even a single nondominated point could be computationally hard for MOCO problems. Therefore, they propose to approximate the nondominated frontier by a hypersurface and then generate hypothetical points on this surface easily. After the most preferred hypothetical point is found, they search for the true nondominated points around that point. This procedure is a practical way to find preferred nondominated points for computationally hard to solve problems like MOCO problems.

Similarly, we approximate the nondominated frontier by fitting a surface and then generate a discrete set on this surface. Among these discrete points, we choose the most diverse subset with a specified cardinality. So, we have a diverse subset of hypothetical points which are expected to be close to some true nondominated points. Therefore, we find the projection of the hypothetical points onto the true nondominated frontier in order to achieve a diverse subset of the true nondominated points. To find the projection, we solve a Tchebycheff program considering the hypothetical point as the reference vector. SPA can be used in cases where the DM expresses the desired number of nondominated points.

We test the performance of the algorithms on random test instances of Multi-objective Knapsack Problem (MOKP) and Multi-objective Assignment Problem (MOAP). Algorithm 1 provides substantial solution time improvements compared to the DMA. We also compare the cardinalities of subsets generated by Algorithm 1 and TDA for the same coverage gap values. Computational results show that TDA can provide the same coverage gap value with fewer solutions than Algorithm 1. Finally, SPA can generate subsets having smaller coverage gap values than Algorithm 1 if the desired number of solutions is reasonable. However, for large representative subsets, Algorithm 1 generates higher quality subsets.

In Chapter 2, we review the approaches in the literature to solve MOIP problems. We mention both exact and heuristic algorithms developed for generating the full or partial nondominated frontier. We review the quality measures and calculation methods in Chapter 3. We present our algorithms in Chapter 4. In Chapter 5, we report the results of computational experiments and we conclude in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

In this section, we review some of the studies on MOIP problems. We briefly summarize both exact and heuristic approaches to generate or approximate nondominated points for MOIP problems. Some of those algorithms are applicable to any MOIP problem, and some are problem specific. We focus on the generic methods which can be used to solve different MOIP problems. Furthermore, we present the studies that define some quality measures for the performance assessment of approximation algorithms and the studies which concentrate on generating only a subset of nondominated points.

Ehrgott and Gandibleux [5] present a comprehensive review of exact and heuristic methods to solve MOCO problems. They emphasize the exponential increase in the number of unsupported nondominated points with the problem size whereas the linear increase in supported nondominated points. In addition, they argue that the difficulty of solving MOCO problems arises from the difficulty of generating unsupported nondominated points. They classify the methods in the literature based on the combinatorial structure of the problem, the number and type of the objectives, the type of the problem such as generating the efficient set, the subset of the efficient set, the supported efficient set, etc. and the solution method applied. They give concluding remarks on the gap between biobjective and multi-objective solution methods and the need to study theoretical propoerties of MOCO problems more. Furthermore, they point out some future research areas such as the computation of the nadir point, generation of lower and upper bound sets and the evaluation of the quality of approximations.

Ehrgott [4] discusses scalarization techniques for MOIP problems and points out that it may be computationally difficult to generate all nondominated points using current scalarization techniques. Based on this observation, he proposes the method of elastic constraints that combines the advantages of weigted sum and $\epsilon-$ constraint scalarization. It can reduce the computational effort to solve scalarized problem and finds all efficient solutions.

There are not many studies in the literature for generating all nondominated points for a MOIP problem with more than two objectives. One general article in this area is by Sylva and Crema [29]. In their algorithm, they generate a nondominated point at each iteration by adding m binary variables and $m + 1$ linear constraints to the model for each nondominated point found in the previous iterations where m is the number of objectives. Therefore, computation of the next nondominated point becomes harder as the number of nondominated points found increases.

Laumans et al. [15] developed an algorithm based on a variaton of the $\epsilon-$ constraint method. The $m - 1$ dimensional space is partitioned into cells with the already found nondominated points. Each cell is defined with lower and upper bounds on the $m - 1$ objectives. The cells are searched in a particular order to prevent generating dominated points.

Özlen and Azizoğlu [34] propose an improvement over the classical $\epsilon-$ constraint method. The method can generate all nondominated points for any number of objectives. The method identifies objective efficiency ranges and uses this information to define tighter bounds for $\epsilon-$constraint scalarization. The method reduces the number of models solved significantly compared to the classical $\epsilon-$ constraint method. Özlen et al. [35] further reduce the number of IP models by storing a list of already solved subproblems and searching this list before solving a new subproblem.

Lokman and Köksalan [19] developed an exact algorithm to generate all nondominated points of MOIP problems. At each iteration, the previously generated nondominated points as well as the subspace dominated by those points are eliminated from the search space. The remaining feasible space is decomposed into subspaces by introducing lower bounds for each objective by means of a sorting mechanism. Then, nondominated points are searched in these subspaces. The size of the model to be solved does not increase during the algorithm. Computational results show that the algorithm is superior in performance compared to Sylva and Crema [29] and Özlen and Azizoğlu [34].

Kirlik and Sayın [12] propose another algorithm to generate all nondominated points for multi-objective discrete optimization problems. They project each nondominated point found to $m-1$ dimensional space where m is the number of objectives. Then, they create $m-1$ dimensional rectangles in which nondominated points exist. The method conducts search in the rectangles and updates them afterwards. The comparison of the algorithm to previous studies is also given for MOKP and MOAP for $m = 3$ and $m = 4$. The method outperforms the previous studies Sylva and Crema [29], Laumans et al. [15] and Özlen and Azizoğlu [34].

A common approach to solve MOIP problems is the two-phase approach. Przybylski et al. [22] applied the two-phase approach to bi-objective assignment problem. In the first phase, supported nondominated points are generated. Using these supported points, triangles on which unsupported nondominated points lie are constructed. These unsupported points can be found with different approaches including problem specific algorithms.

Generating all supported nondominated points is not straightforward for more than two objectives as it is for bi-objective problems. Özpeynirci and Köksalan [36] and Przybylski et al. [23] developed algorithms based on weight space decomposition to generate all extreme supported nondominated points for multi-objective mixed-integer programming (MOMIP) problems with more than two objectives.

Przybylski et al. [24] developed a two-phase algorithm for more than two objectives and applied it to the assignment problem with three objectives. They use the supported nondominated point set generated by Przybylski et al. [23] to define the search areas for the second phase. The method performs much faster than the methods of Sylva and Crema [29], Tenfelde-Podehl [31] and Laumanns et al. [15] in the instances of the three objective assignment problem. They point out that this performance is expected due to use of problem specific methods in the second phase.

Lemestre et al. [16] propose a new exact algorithm to solve bi-objective problems.

The method splits the search space in the first phase and finds a nondominated point in each split if exists. If the nondominated frontier is sparse, then these points are expected to become well-sparse points. In the next stage, all the remaning nondominated points are searched in the triangles defined by the adjacent points found in the previous stage. Dhaenens et al. [3] improved this method to handle more than two objectives. For a problem with m objectives, the method iteratively finds the nondominated point sets of problems with $m-k$ objectives, $1 \leq k \leq m-1$. Thus, nadir point is computed and used to split the search space.

Delort and Spanjaard [2] use bound sets in a two-phase approach for biobjective knapsack problem. Ponte et al. [21] extend beam search algorithm to multi-objective combinatorial optimization problems. Chergui et al. [1] propose a branch-and-cut framework to solve MOIP problems and compare the method with Sylva and Crema [29].

Multi-objective branch and bound methods compose a significant part of the exact solution approaches for MOIP problems, but most of them are restricted to the bi-objective problems. Stidsen et al. [28] developed a bi-objective branch and bound method which is applicable to mixed integer programs where the integer variables are binary and continuous variables appear in at most one objective. A comparison to generic two-phase method is given on different MOMIP problem instances.

Sourd and Spanjaard [27] developed a branch-and-bound method where the main idea is to separate the feasible solutions at a node from the upper bound set via a hypersurface. If this occurs, then the corresponding node can be fathomed.

Although there exist efficient exact methods to solve MOIP problems, they are generally far from being practical for large size problems. Therefore, approximate methods for multi-objective problems have attracted the attention of many researchers in the MCDM area. Ehrgott and Gandibleux [7] present a review of approximate methods for MOCO problems. Extension of metaheuristics to multi-objective case is well-studied by the researchers. Ehrgott and Gandibleux [8] review hybrid metaheuristics combining good features of evolutionary algorithms and neighborhood search algorithms.

In MOEAs, a diversity operator is needed to prevent crowding of the solutions along some parts of the nondominated frontier. Instead of using a diversity operator, Karahan and Köksalan [10] define territories around the individuals to prevent crowding in a region. Any offspring inside these territories is rejected. This mechanism provides well - spread population along the nondominated frontier.

In order to assess the quality of approximate solution sets, Sayın [25] proposes coverage, uniformity and cardinality measures. While smaller values of coverage measure are required to represent all parts of the nondominated frontier, larger values of uniformity measure are desired not to generate too close points in the objective space.

Zitzler et al. [33] discuss the quality indicators in the literature in terms of invariance to scaling, monotonicity and computational effort. They review the quality indicators for the assessment of solution sets produced by deterministic and stochastic algorithms separately.

Faulkenberg and Wiecek [9] classify the measures in the literature as the measures of cardinality, the measures of coverage and the measures of spacing. It also groups the generating methods into three classes according to stage in which quality measure information is integrated to produce representation of the nondominated frontier. A priori methods generate representative solutions such that it is guaranteed that the desired quality level is satisfied while a posteriori methods filter the generated solution set at the end to satisfy desired quality level.

Sayın [26] developed a procedure to find a discrete representation of MOLP problems. One can either generate desired number of points and report the resulting coverage error or guarantee a specified coverage error generating as many points as needed.

Karasakal and Köksalan [11] use a surface to approximate nondominated frontier of a MOLP problem and then generated well distributed reference points on the surface. Then, finding projection of the reference points onto the nondominated frontier, a representative subset of is generated. The method performs well in terms of coverage measure and computation time.

We are aware of two studies in the literature which aim to develop methods to generate representative subsets for MOIP problems. Sylva and Crema [30] modify [29] to generate only a subset of nondominated point set. At each iteration of the algorithm, the nondominated point which is the most distant point to the dominated space is generated. Masin and Bukchin [20] define a diversity measure and develope a method which finds the most diverse nondominated point at each iteration.

CHAPTER 3

QUALITY MEASURES

In this chapter, we first give necessary background in multi-objective decision making. Then, we define some measures to calculate the quality of a representative subset of nondominated point set in MOIP problems. These measures include the coverage error, uniformity and cardinality measures proposed by Sayın [25] as the properties of good representative sets. We focus on the coverage gap measure proposed by Masin and Bukchin [20]. In addition to the definitions of these measures, we discuss some additional properties of the coverage gap measure and the alternative ways to calculate it.

3.1 Background

A MOIP problem can be stated as follows:
(MOIP) "Max" **z = f(x)**, subject to **x** $\in$ **X**,
where **f(x)** $= \{f_1(\mathbf{x}), ..., f_m(\mathbf{x})\}$ is *m* dimensional point, **x** is the decision vector and **X** $\subseteq \mathbb{Z}^n$ is the feasible decision space. **Z** is the feasible objective space made up of the images, **f(x)**, of all points **x** $\in$ **X**.

Definition 3.1. *A feasible decision point* $\boldsymbol{x}^k$ *of the MOIP problem is an efficient solution if there exists no* $\boldsymbol{x}^j$ *such that* $f_i(\boldsymbol{x}^j) \geq f_i(\boldsymbol{x}^k)\ \forall i$ *and* $f_i(\boldsymbol{x}^j) > f_i(\boldsymbol{x}^k)$ *for at least one i. The image of* $\boldsymbol{x}^k$ *in the objective space,* $\boldsymbol{f}(\boldsymbol{x}^k)$*, is said to be nondominated point. Otherwise,* $\boldsymbol{x}^k$ *is an inefficient solution and* $\boldsymbol{f}(\boldsymbol{x}^k)$ *is a dominated point.*

Definition 3.2. *For some point* $\boldsymbol{x}^j$*, if* $f_i(\boldsymbol{x}^j) > f_i(\boldsymbol{x}^k)\ \forall i$*, then* $\boldsymbol{x}^k$ *is a strictly inefficient solution and* $\boldsymbol{f}(\boldsymbol{x}^k)$ *is a strictly dominated point. If there exists no such* $\boldsymbol{x}^j$*, then* $\boldsymbol{x}^k$ *is* *called a weakly efficient solution and $\boldsymbol{f}(\boldsymbol{x}^k)$ is called weakly nondominated point.*

We denote the nondominated point set as $\mathbf{Z}_{ND}$ and the set of efficient solutions as $\mathbf{X}_E$ for the MOIP problem. Let $|\mathbf{Z}_{ND}| = N$.

Definition 3.3. *The ideal point, $\boldsymbol{z}^{IP}$, is a vector such that $z_i^{IP} = Max\,\{z_i, \boldsymbol{z} \in \mathbf{Z}\}$.*

Definition 3.4. *The nadir point, $\boldsymbol{z}^{NP}$, is a vector such that $z_i^{NP} = Min\,\{z_i, \boldsymbol{z} \in \mathbf{Z}_{ND}\}$.*

In case of a minimization problem, $z_i^{IP} = Min\,\{z_i, \mathbf{z} \in \mathbf{Z}\}$ and,
$z_i^{NP} = Max\,\{z_i, \mathbf{z} \in \mathbf{Z}_{ND}\}$.

Definition 3.5. *The distance between two points $\boldsymbol{z}$ and $\boldsymbol{y} \in \boldsymbol{R}^m$ in terms of L_p metric is calculated as follows:*
$||\mathbf{z} - \mathbf{y}||_p = [\sum\limits_{i=1}^{m} |z_i - y_i|^p]^{1/p}, \quad p \in \{1, 2, ...\}$.

Definition 3.6. *The Tchebycheff metric is an L_∞ metric and is defined as*
$||\mathbf{z} - \mathbf{y}||_\infty = [\max\limits_{i=1,\dots,m} |z_i - y_i|]$, $\boldsymbol{z}, \boldsymbol{y} \in \boldsymbol{R}^m$.

3.2 Coverage Error Measure

According to Sayın [25], a representative subset of a nondominated point set should cover all parts of the nondominated frontier well. For this purpose, Sayın [25] defines the "coverage error" as follows:

Definition 3.7. *Let $\boldsymbol{R}$ be a representative subset of all nondominated points. The coverage error, ϵ, of set $\boldsymbol{R}$ in terms of Tchebycheff metric for a "max" type problem can be calculated as*
$\epsilon = \max\limits_{\boldsymbol{z} \in \mathbf{Z}_{ND}} \min\limits_{\mathbf{y} \in \mathbf{R}} ||\mathbf{z} - \mathbf{y}||_\infty$.

Example 3.1:

Consider a bi-objective problem and let $\mathbf{Z}_{ND}$ be the set of all nondominated points of the problem, $\mathbf{Z}_{ND} = \{(1, 10), (3, 9), (5, 8), (6, 5), (8, 4), (9, 3), (10, 1)\}$. Suppose a representative subset of cardinality 3 is chosen as $\mathbf{R} = \{(3, 9), (5, 8), (9, 3)\}$. The worst represented point is shown on Figure 3.1. The closest representative nondominated point to $(6, 5)$ is $(5, 8)$ or $(9, 3)$ which have the same Tchebycheff distance to

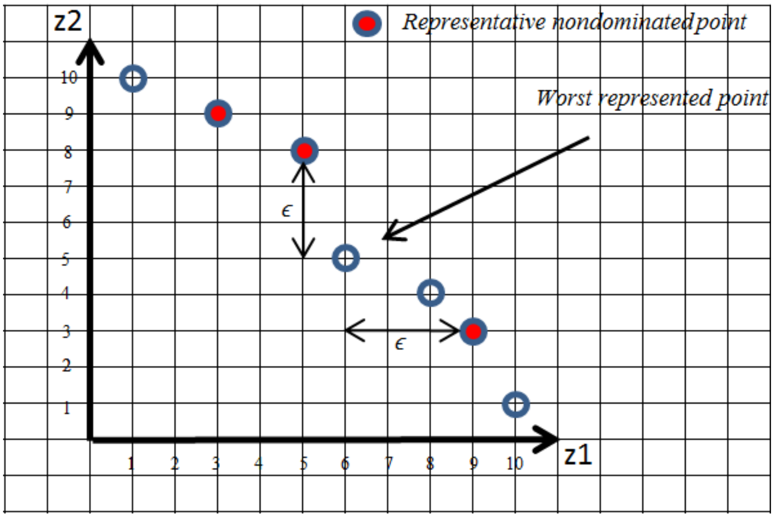

Figure 3.1: A representative subset with $\epsilon = 3$.

point $(6, 5)$ which is 3. Other solutions are closer to their closest representative points. Therefore, point $(6, 5)$ is the worst represented point determining the coverage error of the set **R**.

Every point $\mathbf{z}$ is represented by the closest point $\mathbf{y}' \in \mathbf{R}$ and the coverage error for point $\mathbf{z}$ is $||\mathbf{z} - \mathbf{y}'||_\infty$. The maximum of all such errors gives the coverage error of the representative subset **R**. The worst represented point is the one which has the coverage error equal to ϵ.

If the representative point set is available, then Sayın [25] gives a 0-1 mixed integer programming formulation to compute the coverage error. The formulation includes $2mN$ binary and $1+m+(2m+1)N$ continuous variables and $(4m+2)N$ constraints in addition to the constraints defining the feasible objective space, **Z**.

Calculation of the coverage error requires the availability of the nondominated point set, otherwise the coverage error cannot be computed exactly. Sayın[26] uses the coverage error measure for the MOLP problem to generate representative points on the nondominated faces. After a point is generated on a nondominated face, then the worst represented point on that face is found by solving a 0-1 mixed integer programming problem. The procedure requires the generation of the nondominated faces prior to finding the representative points.

A coverage error value, ϵ, guarantees that a nondominated point, **z**, not included in the representative set has at most ϵ difference in any of the objectives compared to the closest representative point to it, **y**. Point **z** is worse than point **y** in at least one objective since both points are nondominated points. Therefore, the difference in the objective where point **z** is worse than point **y** is also accounted in the calculation of the coverage error.

However, there should be no interest in the objectives of point **z** where it is worse than its representative point. Masin and Bukchin [20] propose *coverage gap* measure which accounts for the representation error only in the objectives where point **z** is better than its representative point.

3.3 Coverage Gap Measure

Definition 3.8. *Given the nondominated set* $\mathbf{Z}_{ND}$ *and a representative nondominated subset* $\boldsymbol{R} \subseteq \mathbf{Z}_{ND}$*, the coverage gap of set* $\boldsymbol{R}$ *for a "max" type problem is defined as follows:*

$$\alpha = \max_{z \in \mathbf{Z}_{ND}} \left\{ \min_{\mathbf{y} \in \mathbf{R}} \left\{ \max_{1 \leq i \leq m} z_i - y_i \right\} \right\}.$$

In their definition, they also integrate a scaling coefficient to the calculation of the measure. At this point, we assume that all objectives have approximately equal range of values. If the scales of the objectives are different, than the integration of a scaling coefficient to the the measure is straighforward. Furthermore, a weighted coverage gap measure might also be considered in some cases. Then, we can scale the objectives into the ranges proportional to the weights given.

If the coverage gap of a representative set is α, then it means that there exist at least one nondominated point not included in the representative set such that it is α better than its representative point in at least one objective. That point is called "most diverse point". However, *most diverse point* might be more than α worse in as many as $m-1$ objectives.

Definition 3.9. *The nondominated point* $\mathbf{y}^* \in \mathrm{R}$ *is called the representative of point* $\mathbf{z}'$ *if*

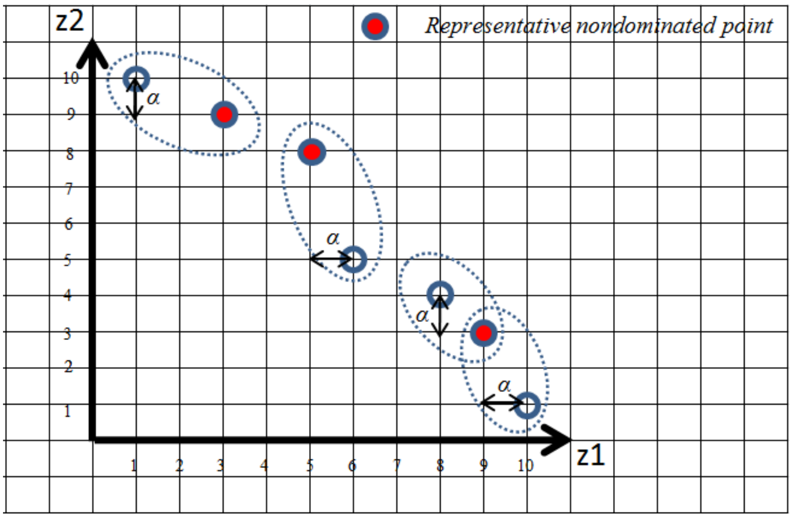

Figure 3.2: A representative subset with $\alpha = 1$.

$$\max_{1 \leq i \leq m} z'_i - y^*_i = \min_{\mathbf{y} \in \mathbf{R}} \left\{ \max_{1 \leq i \leq m} z'_i - y_i \right\}.$$

Example 3.2:

Consider the same problem in Example 3.1. The coverage gap of set **R**, $\alpha_{\mathbf{R}}$, is equal to 1. As it can be seen on Figure 3.2, all nondominated points except for the representative ones have the same maximum criterion difference from their representative points.

From the definition of the coverage gap measure, Masin and Bukchin [20] present the following lemma:

Lemma 3.1. *Let* $\mathbf{z}' \in \mathbf{Z}$ *and* $\alpha_{\boldsymbol{R}}(\mathbf{z}') = \min_{\mathbf{y} \in \mathbf{R}} \left\{ \max_{1 \leq i \leq m} z'_i - y_i \right\}$. *Then,*

1. *if* $\alpha_{\boldsymbol{R}}(\mathbf{z}') < 0$, *then* $\mathbf{z}'$ *is dominated by at least one point in* ***R***,
2. *if* $\alpha_{\boldsymbol{R}}(\mathbf{z}') > 0$, *then* $\mathbf{z}'$ *is not dominated by any point in* ***R***,
3. *if* $\alpha_{\boldsymbol{R}}(\mathbf{z}') = 0$, *then* $\mathbf{z}' \leq \mathbf{y}$ *for some point* $\mathbf{y} \in \mathbf{R}$ *and* $z'_i = y_i$ *for some objective* i,
4. *if* $\alpha_{\boldsymbol{R}}(\mathbf{z}') > \alpha_{\boldsymbol{R}}(\mathbf{z}'')$ *for some* $\mathbf{z}'' \in \mathbf{Z}_{ND}$, *then* $\mathbf{z}'$ *is not dominated by* $\mathbf{z}''$,
5. *if* $\boldsymbol{R}_1 \subseteq \boldsymbol{R}_2 \subseteq \mathbf{Z}_{ND}$, *then* $\alpha_{\boldsymbol{R}_1}(\mathbf{z}') \geq \alpha_{\boldsymbol{R}_2}(\mathbf{z}')$.

We present the following corollary based on the characteristics of the most diverse point:

Corollary 3.1. *There exists at least one nondominated most diverse point.*

Proof. Let $\mathbf{z}'$ be the most diverse point, $\alpha_{\mathbf{R}}(\mathbf{z}') \geq \alpha_{\mathbf{R}}(\mathbf{z}) \quad \forall \mathbf{z} \in \mathbf{Z}_{ND}$. If it is strictly dominated, then there exists at least one point $\mathbf{z}'' \in \mathbf{Z}_{ND}$ such that $z_i' < z_i'' \quad \forall i$. Then, $\alpha_{\mathbf{R}}(\mathbf{z}'') > \alpha_{\mathbf{R}}(\mathbf{z}')$ contradicting with the fact that $\alpha_{\mathbf{R}}(\mathbf{z}') \geq \alpha_{\mathbf{R}}(\mathbf{z})\ \forall \mathbf{z} \in \mathbf{Z}_{ND}$.
Then, if there exist more than one most diverse point, they are either weakly nondominated but dominated or nondominated points. Let $\mathbf{z}'$ one of those points. If $\mathbf{z}'$ is a weakly nondominated but dominated point, then there exists at least one nondominated point $\mathbf{z}''$ such that $\mathbf{z_i'} \leq \mathbf{z_i''} \quad \forall i$ and $\mathbf{z}''$ is also a most diverse point. $\square$

The following model finds a new most diverse nondominated point when a representative set **R** is already available. Although Masin and Bukchin [20] give the formulation for a "min" type problem, we change it to "max" type in order not to confuse the reader. Besides, we do not introduce the scaling coefficient, Δ_{ie}, since, without loss of generality, one can scale the objectives and make the ranges approximately equal.

$$
\begin{aligned}
&\text{(P1):}\\
&\text{Max} \quad Z = lexmax(\alpha, \sum_{i=1}^{m} w_i z_i(\mathbf{x}))\\
&\text{s.to.}\\
&\qquad \alpha = \min_{\mathbf{y}\in\mathbf{R}} \left\{ \max_{1\leq i\leq m} z_i(\mathbf{x}) - y_i \right\}\\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned}
\tag{3.1}
$$

The lexicographic type objective function of model (P1) guarantees to find the most diverse nondominated point. In lexicographic maximizaton, the model is first solved with objective function α, then adding the objective values of the optimal **z(x)** point as the lower bounds, a weighted sum objective function is maximized where $w_i > 0 \quad \forall i$. Masin and Bukchin [20] linearize the nonlinear constraint and solve the mixed-integer linear programming (MILP) model (P2) to find the most diverse nondominated point in $\mathbf{Z}_{ND}$. In the model, β^j is a continuous variable which represents the distance to the point $\mathbf{y} \in \mathbf{R}$ and γ_{1ij} is a binary variable which takes 1 if $z_i(\mathbf{x}) - y_i^j = \max_{i'=1,2,\ldots,m} z_{i'}(\mathbf{x}) - y_{i'}^j$.

The second term in the objective function guarantees to find a nondominated point if ϵ is sufficiently small positive constant. The first constraint ensures that the optimal point is the most diverse one and $\beta^j = \max_{i=1,2,\ldots,m} z_i(\mathbf{x}) - y_i^j$ by the following three constraints. The remaining constraints is to ensure that $\gamma_{2ij} = z_i(\mathbf{x})\gamma_{1ij}$.

In this formulation, there are $m|\mathbf{R}|$ binary variables and $(m+1)|\mathbf{R}|+1$ continuous variables. Therefore, as the cardinality of the representative set increases, finding the most diverse point becomes harder. In their algorithm, *M*-DMA, they start with an initial nondominated point and then (P2) is solved in each iteration. The nondominated point found in each iteration is added to the representative set and (P2) is modified accordingly. The coverage gap value will be nonincreasing as the number of representative points generated increases and it is equal to zero when all nondominated points are generated.

(P2):

$$\text{Max} \quad Z = \alpha + \epsilon \sum_{i=1}^{m} \{\lambda_i z_i\}$$

s.to.

$$\begin{aligned}
&\alpha \leq \beta^j \quad j = 1,2,...,|\mathbf{R}| \\
&\beta^j \geq z_i(\mathbf{x}) - y_i^j \quad i = 1,...,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\beta^j = \sum_{i=1}^{m} \gamma_{2ij} - y_i^j \gamma_{1ij} \quad j = 1,2,...,|\mathbf{R}| \\
&\sum_{i=1}^{m} \gamma_{1ij} = 1 \quad j = 1,2,...,|\mathbf{R}| \\
&\gamma_{2ij} \geq z_i(\mathbf{x}) + (\gamma_{1ij} - 1)M \quad i = 1,..,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\gamma_{2ij} \leq z_i(\mathbf{x}) + (1 - \gamma_{1ij})M \quad i = 1,..,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\gamma_{2ij} \leq \gamma_{1ij}M \quad i = 1,..,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\gamma_{1ij} \in \{0,1\} \quad i = 1,..,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\gamma_{2ij} \geq 0 \quad i = 1,..,m, \quad j = 1,2,...,|\mathbf{R}| \\
&\mathbf{x} \in \mathbf{X}
\end{aligned} \tag{3.2}$$

Stopping criterion for their algorithm can be the number of representative nondominated points desired by the DM or an upper bound for the coverage gap of the generated set. Both stopping criteria may also be used such that at least one of them will hold. The authors argue that at the end of the algorithm, the nondominated points generated are well distributed over the complete nondominated frontier.

If the coverage gap measure is used to generate a subset of nondominated points, then we noticed that the coverage gap of the resulting set gives a valuable information. We state the following proposition to show the characteristic of the representative subset found when the algorithm terminates.

Proposition 3.1. *Let the coverage gap value in the last iteration of the algorithm be α^* and let the generated set be **R**. Then, α^* is the minimum Δ value to satisfy that for any nondominated point **z** $\in \mathbf{Z}_{ND}$ there exists at least one point $\mathbf{y}' \in$ **R** such that $z_i \leq y_i' + \Delta$, i=1,...,m. When this inequality holds, we say that point **y** $\Delta-$ dominates point **z**.*

Proof. Let $\mathbf{y}'$ be the representative point of **z**. If the coverage gap of set **R** is α^*, then
$\max\limits_{i=1,\ldots,m} \{z_i - y_i'\} \leq \alpha^*$,
$z_i - y_i' \leq \alpha^* \quad i = 1, \ldots, m$,
$z_i \leq y_i' + \alpha^* \quad i = 1, \ldots, m$.
Let z^* be the most diverse point. Then, $\max\limits_{i=1,\ldots,m} \{z_i^* - y_i'\} = \alpha^*$ which implies that $\Delta \geq \alpha^*$. □

Another mathematical formulation to find a representative nondominated point set is proposed by Sylva and Crema [30]. They find the nondominated point which has the largest Tchebycheff distance from the dominated region. In their iterative algorithm, they solve model (P3) in each iteration and can generate the whole nondominated frontier, where $\mathbf{z}^k = \left\{z_1^k, \ldots, z_m^k\right\}$ denotes the k^{th} nondominated point in representative set **R**, M_i is the lower bound for $z_i(\mathbf{x})$, U is an upper bound to $||z(\mathbf{x}) - z(\mathbf{x}')||$

for any $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$, $\lambda_i > 0 \quad \forall i$ and ϵ is a small positive constant.

$$
\begin{aligned}
&\text{(P3):}\\
&\text{Max} \quad Z = \delta + \epsilon \sum_{i=1}^{m} \{\lambda_i z_i(\mathbf{x})\}\\
&\text{s. to.}\\
&\qquad z_i(\mathbf{x}) \geq z_i^k y_i^k + \delta - (M_i + U)(1 - y_i^k) \quad for \quad i = 1, ..., m; \quad k = 1, ..., n,\\
&\qquad \sum_{i=1}^{m} y_i^k = 1 \quad for \quad k = 1, ..., n,\\
&\qquad y_i^k \in \{0, 1\} \quad for \quad i = 1, ..., m; \quad k = 1, ..., n,\\
&\qquad \delta \geq 0,\\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned} \tag{3.3}
$$

Before showing that (P3) gives the point which maximizes the Tchebycheff distance to the region dominated by set **R**, they present the following two lemmas:

Lemma 3.2. *Let* $\boldsymbol{R} = \{\mathbf{z}^1, ..., \mathbf{z}^n\}$ *be the set of nondominated points and* $\mathbf{Z}_D^k = \left\{\mathbf{z} \in \mathbf{Z} : \mathbf{z} \leq \mathbf{z}^k\right\}$ *is the dominated portion of the feasible objective space by point* $\mathbf{z}^k$*. Also suppose that* $\hat{\mathbf{z}} \in \mathbf{Z} - \cup_{k=1}^{n} \mathbf{Z}_D^k$ *and* $\delta = \min_{k=1,...,n} \left\{ \max_{i=1,...,m} \left\{\hat{z}_i - z_i^k\right\} \right\}$*. Then,* $\delta = min \left\{||\hat{\mathbf{z}} - \mathbf{z}||_\infty | \mathbf{z} \in \cup_{k=1}^{n} \mathbf{Z}_D^k\right\}$.

Lemma 3.2. states that the Tchebycheff distance of a nondominated point $\hat{\mathbf{z}}$ from the region dominated by set **R** can be found by taking the minimum of the maximum criterion difference of point $\hat{\mathbf{z}}$ to each point in set **R**. In the following lemma, they show that the Tchebycheff distance of point $\hat{\mathbf{z}}$ from the region dominated by set **R** is the optimal objective function value of (P3) with the additional constraint of $\mathbf{z}(\mathbf{x}) = \hat{\mathbf{z}}$.

Lemma 3.3. *Let (P3($\hat{\mathbf{z}}$)) be the modified version of (P3) with the addition of constraint* $\mathbf{z}(\mathbf{x}) = \hat{\mathbf{z}}$ *and the optimal value of this model be* $\hat{\delta}$*. Then,*
$\hat{\delta} = min \left\{||\hat{\mathbf{z}} - \mathbf{z}||_\infty | \mathbf{z} \in \cup_{k=1}^{n} \mathbf{Z}_D^k\right\}$.

Following above two lemmas, they show that following proposition holds:

Proposition 3.2. *Let δ^* be the optimal value of (P3). If $\delta^* > 0$, then* $\delta^* = \max_{\mathbf{z}' \in \mathbf{Z} - \cup_{k=1}^{n} \mathbf{Z}_D^k} \left\{ min \left\{ ||\mathbf{z}' - \mathbf{z}||_\infty | z \in \cup_{k=1}^{n} \mathbf{Z}_D^k \right\} \right\}$. *If $\delta^* = 0$, then set **R** includes all nondominated points of the problem.*

Interested readers can find the proofs of the above two lemmas and the proposition in Sylva and Crema [30]. They use model (P3) to generate a well-dispersed subset of nondominated points with similar stopping conditions to the ones in Masin and Bukchin [20]. Note that (P3) guarantees to find the nondominated point which is at the maximum Tchebycheff distance to the dominated space by set **R**.

The model of Sylva and Crema [30] includes same number of binary variables with the model of Masin and Bukchin [20]. However, the number of continuous variables and constraints is much lower in (P3) compared to (P2).

The aforementioned two studies try to generate a subset of the nondominated frontier which is claimed to be well-distributed over the nondominated frontier. In fact, we can show that two algorithms generate the subsets with the same coverage gap provided that they start at the same point and there exists unique optimal solution at each iteration. In the following corollary, we state that the nondominated points corresponding to the optimal solutions of (P2) and (P3) are the same assuming that they are both feasible.

Corollary 3.2. *Let $\mathrm{R} = \{\mathbf{y^1}, \mathbf{y^2}, ..., \mathbf{y^n}\}$ be the already generated subset of the all nondominated points. If (P2) and (P3) have unique optimal solutions, then the optimal solutions of (P2) and (P3) are the same.*

Proof. Assume that we know all nondominated points of the problem, $\mathbf{Z}_{ND} = \left\{\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^N\right\}$. The optimal value of (P2) will satisfy the following equality:

$$\alpha^* = \max_{k=1,2,...,N} \left\{ \min_{j=1,2,...,n} \left\{ \max_{i=1,2,...,m} \left\{ z_i^k - y_i^j \right\} \right\} \right\}$$

Let $\alpha(k) = \min_{j=1,2,...,n} \left\{ \max_{i=1,2,...,m} \left\{ z_i^k - y_i^j \right\} \right\}$ and $\alpha^* = \alpha(k')$. On the other hand, optimal value of (P3) is:

$\delta^* = \max_{k=1,2,...,N} \left\{ \min(||\mathbf{z}^k - \mathbf{z}||_\infty | \mathbf{z} \in \cup_{j=1}^{n} \mathbf{y}_D^j) \right\}$ where $\mathbf{y}_D^j = \{\mathbf{z} \in \mathbf{Z} : \mathbf{z} \leq \mathbf{y}^j\}$. Let $\delta(k) = \min(||\mathbf{z}^k - \mathbf{z}||_\infty | \mathbf{z} \in \cup_{j=1}^{n} \mathbf{y}_D^j)$ and $\delta^* = \delta(k'')$. Lemma 3.2. shows that $\alpha(k) = \delta(k)$. Then, $\alpha^* = \max_{k=1,2,...,N} \alpha(k) = \delta^* = \max_{k=1,2,...,N} \delta(k)$.

Since we assume that the models have unique optimal solutions, both models find the same most diverse nondominated point at each iteration.

□

These two formulations become impractical as the size of the representative set increases. With the addition of new binary variables and constraints for each representative point found, the models become very difficult to solve. To be able to find the coverage gap of large representative sets, we propose to decompose the problem into smaller ones and to solve models without additional binary variables and constraints. If we divide the nondominated feasible space into smaller nondominated subspaces, we can search for the nondominated point in each subspace such that the solution found is at maximum Tchebycheff distance to the dominated region. Then, we can pick the most distant point among all the nondominated points generated and that solution will be the one giving the coverage gap of the subset. With the following propositions, we show that this procedure can be used to find the coverage gap of large representative sets.

Proposition 3.3. *Let $lb_1, lb_2, ..., lb_m$ be the lower bounds defining the nondominated subspace* **s** *of the feasible objective space,* **Z** *and let ϵ be a small positive constant. Then, the following model finds a nondominated point in* **s** *if the model is feasible:*

$$
\begin{aligned}
&P(\alpha_{\mathbf{s}}): \\
&\quad Max \quad \alpha + \epsilon \sum_{i=1}^{m} \{\lambda_i z_i\} \\
&\quad s.\ to. \\
&\qquad z_i(\boldsymbol{x}) \geq lb_i + \alpha \quad \forall i = 1, 2, ..., m \\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned}
\tag{3.4}
$$

Proof. Let P($\alpha_{\mathbf{s}}$) be feasible and $(\mathbf{z}^*(\mathbf{x}), \alpha^*)$ be the optimal solution of P($\alpha_{\mathbf{s}}$). At least one of the constraints $z_i(\mathbf{x}) \geq lb_i + \alpha$ will be binding at optimal solution. Let i^* be the objective at which the corresponding constraint is binding. Then, $i^* = \arg \min_{i=1,2,\ldots,m} \{z_i^*(\mathbf{x}) - lb_i\}$ and $z_{i^*}^*(\mathbf{x}) = lb_{i^*} + \alpha^*$. Since the objective function is augmented with weighted sum of the criterion values, $\mathbf{z}^*(\mathbf{x})$ cannot be weakly nondominated but dominated solution. Suppose $\mathbf{z}^*(\mathbf{x})$ is a strictly dominated solution. Then, there should be at least one nondominated point $\hat{\mathbf{z}}$ such that $\hat{z}_i > z_i^* \quad \forall i = 1, 2, ..., m$. This implies that $\hat{z}_{i^*} - lb_{i^*} > \alpha^*$ and contradicts with the optimality of α^*. Therefore, $\mathbf{z}^*$ is a nondominated point in subspace $\mathbf{s}$. $\square$

Proposition 3.4. *Suppose **R** is the nondominated point set already generated and $\mathbf{Z}_D$ is the portion of the feasible objective space dominated by **R**. Let $\mathbf{z}^{\mathbf{s}}$ be the nondominated point in subspace **s** found by P(α_s) and the corresponding solution is $(\mathbf{z}^{\mathbf{s}}, \alpha^s)$. Then, α^s is the maximum Tchebycheff distance in subspace **s** to the dominated region.*

Proof. Let $\bar{i}$ be the objective where the corresponding lower bound constraint is binding at the optimal solution, $\alpha^{\mathbf{s}} = z_{\bar{i}}^{\mathbf{s}} - lb_{\bar{i}}$. For any $\mathbf{z} \in \mathbf{Z}_D$, $||\mathbf{z}^{\mathbf{s}} - \mathbf{z}||_\infty \geq \max_{i=1,2,...,m} \{z_i^{\mathbf{s}} - z_i\} \geq \min_{i=1,2,...,m} \{z_i^{\mathbf{s}} - lb_i\} = z_{\bar{i}}^{\mathbf{s}} - lb_{\bar{i}} = \alpha^{\mathbf{s}}$. So, $\min_{\mathbf{z} \in \mathbf{Z}_D}(||\mathbf{z}^{\mathbf{s}} - \mathbf{z}||_\infty) = \alpha^{\mathbf{s}}$. Since $\alpha^{\mathbf{s}} \geq \alpha_{\mathbf{R}}(\mathbf{z}') \quad \forall \mathbf{z}' \in \mathbf{s}$ due to optimality of $\alpha^{\mathbf{s}}$, $\mathbf{z}^{\mathbf{s}} = \operatorname{argmax}_{\mathbf{z}'}(\alpha_{\mathbf{R}}(\mathbf{z}'))$ and therefore $\mathbf{z}^{\mathbf{s}} = \operatorname{argmax}_{\mathbf{z}' \in \mathbf{s}} \{min(||\mathbf{z}' - \mathbf{z}||_\infty | \mathbf{z} \in \mathbf{Z}_D)\}$. $\square$

Proposition 3.5. *Let **S** be the set of all nondominated subspaces, $\mathbf{s}^*$= $\operatorname{argmax}_{\mathbf{s} \in \mathbf{S}} \{\alpha^s | \mathrm{s} \in \mathrm{S}\}$ and $(\mathbf{z}^*, \alpha^*)$ be the optimal solution of $P(\alpha_s)$ in $\mathbf{s}^*$. Then, α^* gives the coverage gap of the nondominated point set, **R**, and $\mathbf{z}^*$ is the most diverse nondominated point.*

Proof. If $\alpha^* = \max_{\mathbf{s} \in \mathbf{S}} \{\alpha^{\mathbf{s}}\}$, then $\alpha^* = \max_{\mathbf{s} \in \mathbf{S}} \left\{ \max_{\mathbf{z}' \in \mathbf{s}} \{min(||\mathbf{z}' - \mathbf{z}||_\infty | \mathbf{z} \in \mathbf{Z}_D)\} \right\} = \max_{\mathbf{z}' \in \mathbf{Z}_{ND}} \{min(||\mathbf{z}' - \mathbf{z}||_\infty | \mathbf{z} \in \mathbf{Z}_D)\}$. By Proposition 3.2., α^* is the coverage gap of set **R** and $\mathbf{z}^*$ is the most diverse nondominated point. $\square$

This procedure can be used instead of solving models (P2) or (P3) at each iteration of the subset generating algorithms developed by Sylva and Crema [30] and Masin and Bukchin [20]. If there exist unique optimal solution for the model P($\alpha_{\mathbf{s}}$) for each subspace, then the same representative subset of nondominated points will be generated.

In order to assess the quality of a representative set in terms of coverage gap measure, we develop a mathematical model (P4) to find the optimal representative subset for a given number of representative points, |**R**|. That is, the model chooses the best |**R**| representative nondominated points that minimizes the coverage gap measure.

Decision Variables:

α: coverage gap of the representative subset

y_k: 1 if the k^{th} nondominated point is selected as a representative point, 0 otherwise

u_{jk}: 1 if the k^{th} nondominated point is the representative point of j^{th} nondominated point, 0 otherwise. (k^{th} nondominated point is the closest representative point to j^{th} nondominated point.)

β_j: coverage gap of the j^{th} nondominated point

We also define a distance metric between each two nondominated points:

$$d_{jk} = \max_{i=1,2,\ldots,m} \left\{ z_i^j - z_i^k \right\} \quad j,k = 1,2,...,N. \tag{3.5}$$

The model (P4) finds the best subset of nondominated points of cardinality |**R**|:

$$\begin{aligned}
&\text{(P4):}\\
&\text{Min} \quad \alpha\\
&\text{s. to.}\\
&\quad \alpha \geq \beta_j \quad j = 1,2,...,N\\
&\quad \beta_j = \sum_{k=1}^{N} d_{jk} u_{jk} \quad j = 1,2,...,N\\
&\quad \sum_{k=1}^{N} u_{jk} = 1 \quad j = 1,2,...,N\\
&\quad \sum_{k=1}^{N} y_k = |\mathbf{R}|\\
&\quad u_{jk} \leq y_k \quad j,k = 1,2,...,N\\
&\quad \alpha, \beta_j \geq 0, y_k, u_{jk} \in \{0,1\}
\end{aligned} \tag{3.6}$$

This problem is very similar to p-center problem in the literature except that our distance metric is not symmetric (i.e. $d_{jk} \neq d_{kj}$). The model has $N + N^2$ binary variables, N continuous variables and $N^2 + 3N + 1$ constraints. However, we can relax the constraint $y_{jk} \in \{0,1\}$ since each point will be assigned to its closest point in the optimal solution. In other words, partial assignment of a point to representative points cannot be optimal. With this property, number of binary variables is reduced to N.

3.4 Uniformity Measure

Sayın [25] defines the uniformity measure to assess the closeness of representative points as follows:

Definition 3.10. *The uniformity of set **R** is,* $\delta_{\boldsymbol{R}} = \min_{\boldsymbol{y},\boldsymbol{z}\in\boldsymbol{R}|y\neq z} ||\boldsymbol{y}-\boldsymbol{z}||_\infty$

It is expected that good representative subsets have higher uniformity values. If the uniformity of a representative set is too low, then some parts of the nondominated frontier is overrepresented compared to other parts. If the uniformity of a set is low, then the coverage gap of the set can be expected to be improved by selecting another point instead of the one of the closest representative points in the current set. Representative point set can be regarded as a scarce resource. If we spent most of it for some parts of the nondominated frontier, low uniformity, then there will be some parts of the nondominated frontier which are not well represented, high coverage gap.

3.5 Cardinality Measure

Cardinality is another measure proposed by Sayın [25] to assess the quality of the representative subsets. Since the main motivation is to help the DM to analyze the tradeoff information easily tolerating some error and reduce the computational effort, it does not make sense if the cardinality of the subset is large. Therefore, it is expected that the number of representative points is small enough to be able to generate in a reasonable computation time.

There are many other quality measures in the literature to assess the performance of approximation algorithms. Some of the measures are very closely related with each other such that high quality in terms of one measure implies higher qualities in another measures. However, there are also measures that are conflicting with each other. One of the main concerns is the ease of the computation of the measure. Other desired features of quality measures can be reviewed in Zitzler et al. [33].

We only consider coverage gap and cardinality measures to assess the quality of the subsets generated by the algorithms which we propose in the next chapter. We did not consider the uniformity measure explicitly since we observed that smaller values of coverage gap measure imply higher uniformity values and it is not very clear that why a DM is supposed to desire higher uniformity values for a subset. However, the interpretation of the coverage gap measure by the DM is straighforward as we discussed before. Furthermore, generating high quality representative subset can be considered as a bi-objective problem with the objectives to minimize cardinality and coverage gap. These two objectives are highly conflicting with each other. With our proposed approaches, we attack the different portions of the nondominated frontier of this bi-objective problem.

CHAPTER 4

APPROACHES FOR GENERATING REPRESENTATIVE NONDOMINATED POINT SETS

In this chapter, we present three approaches for generating representative subsets of nondominated points. We use the coverage gap measure given in the previous section to evaluate the quality of representative nondominated points generated. We are first going to review the algorithm proposed by Sylva and Crema [30] and Masin and Bukchin [20]. In the previous section, we showed that both algorithms generate same coverage gap values when they start with the same initial solution and the models solved have unique optimal solutions. Therefore, we give only the algorithm of Masin and Bukchin [20] which they call as Diversity Maximization Algorithm (DMA). Then, we are going to present our algorithms specifying the cases in which they can be best used.

We first propose a computation time improvement on DMA by using the nondominated subspace generation technique proposed in Lokman and Köksalan [19]. In the second algorithm, we ask the DM to set a coverage gap threshold value and try to satisfy this threshold value with minimum number of representative points. Lastly, we develop an algorithm based on an L_p function fitted to approximate the nondominated frontier. In this algorithm, the number of representative points is assumed to be given by the DM.

4.1 Diversity Maximization Algorithm (DMA)

Step 1. Find an initial nondominated point, $\mathbf{y}^*$. Initialize the representative nondominated point set, $\mathbf{R} = \{\mathbf{y}^*\}$.
Step 2. Solve problem (P2).
Step 3. If $\alpha^* > \Delta$, then $\mathbf{R} = \mathbf{R} \cup \mathbf{y}^*$, go to Step 2; else stop.

If $\Delta = 0$, DMA can find all the nondominated points of the problem. Otherwise, the nondominated points in set **R** at the termination of the algorithm $\Delta-$*dominate* all the nondominated points. Maximum number of nondominated points can also be a termination condition for the algortihm.

In each iteration of the algorithm, the coverage gap value of set **R** is at least as good as the one in the previous iteration. The reason is that the algorithm adds the current most diverse nondominated point at each iteration. As one more point is added to the set **R**, m (the number of objectives) binary variables and $4m + 3$ linear constraints are added to the model. Therefore, the computation time increases as the number of representative point increases.

Next, we are going to propose our first algorithm which achieves same coverage gap values when it starts with the initial solution of DMA and has unique optimal solutions to the models solved. However, as we will show in the computational experiments, this algorithm reduces the computation time significantly as the size of the representative set increases.

4.2 Algorithm 1

In the DMA, dominated objective space by the current nondominated points is eliminated with the use of binary variables and additional linear constraints. We refer the feasible space which is not dominated by the current nondominated points as *not-yet dominated space*. Lokman and Köksalan [19] enumerate all not-yet dominated subspaces and conduct search in those subspaces. These subspaces are defined as the set of feasible points in the objective space satisfying a set of lower bounds.

In this approach, only $(m-1)$ lower bounds are added to the model to define a not-yet dominated subspace. Thus, the size of the models to be solved does not increase. On the other hand, the model has to be solved for each subspace and the number of subspaces can be high. However, the results in the computatinal experiments show that solving many simpler models instead of solving a complex model improves the solution time of the DMA substantially as the cardinality of the representative set increases.

In their approach, one of the objective functions is selected at the beginning of the algorithm and that objective function is maximized in each model solved. Let this objective be p. Although the method can find many nondominated points at each iteration when all the subspaces are searched, only the nondominated point having the highest value in objective p is added to the generated nondominated point list. Other nondominated points are stored in a separate list in order not to solve one more model to obtain the same point in the succeeding iterations. Then, the generated nondominated points are in nonincreasing order of the p^{th} objective values. Therefore, only $(m-1)$ lower bounds are required to identify not-yet dominated solution space.

However, in Algorithm1, we find the most distant point from the already generated nondominated points. Therefore, we require m lower bounds. We next present how to generate these lower bounds:

Generation of lower bounds:
The model to be solved in each subspace is as follows:

$$
\begin{aligned}
&(P^{\mathbf{lb}^{\mathbf{k}},n}): \\
&\quad \text{Max} \quad \alpha + \epsilon \sum_{i=1}^{m} \{\lambda_i z_i\} \\
&\quad \text{s. to.} \\
&\qquad z_i(\mathbf{x}) \geq lb_i + \alpha \quad i = 1, 2, ..., m \\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned}
\tag{4.1}
$$

Suppose that $\mathbf{R}_n$ is the set of already generated nondominated points with cardinality of n, $\mathbf{R}_n = \{\mathbf{z}^j : 1 \leq j \leq n\}$, and $\mathbf{z}^j$ be the j^{th} nondominated point in the generation sequence. Let $\mathbf{lb}$ be the vector of lower bounds, $\mathbf{lb} = \{lb_1, lb_2, ..., lb_m\}$.

Let k_i be the index of the nondominated point that is used to set a lower bound for the i^{th} objective and $\mathbf{k} = (k_1, k_2, ..., k_{m-1})$. $0 \leq k_i \leq n$ and if $k_i = 0$ there is no lower bound set for the i^{th} objective. $\mathbf{R}_n^{k_1} = \left\{\mathbf{z}^j : z_1^j \geq z_1^{k_1}\right\}$ and $\mathbf{R}_n^{k_i} = \left\{\mathbf{z}^j \in \mathbf{R}^{k_{i-1}} : z_i^j \geq z_i^{k_i}\right\} \quad i = 2, 3, ..., m.$

If $i < i'$, then $z_i^{k_{i'}} \geq z_i^{k_i}$ where $k_i, k_{i'} > 0$ and $k_i \neq k_{i'}$. We denote a lower bound vector as $\mathbf{lb}^{\mathbf{k},n} = (lb_1^{\mathbf{k},n}, lb_2^{\mathbf{k},n}, ..., lb_m^{\mathbf{k},n})$ such that:
$lb_1^{\mathbf{k},n} = z_1^{k_1} + 1$,
$lb_2^{\mathbf{k},n} = z_2^{k_2} + 1$, where $z_1^{k_2} \geq z_1^{k_1}$,...
$lb_{i+1}^{\mathbf{k},n} = z_{i+1}^{k_{i+1}} + 1$, such that $\mathbf{z}^{k_{i+1}} \in \mathbf{R}_n^{k_i}$. If $\mathbf{R}_n^{k_i} = \emptyset$, then $lb_{i+1} = -M$.
Lastly, we set the lower bound for the m^{th} criterion as follows:
$lb_m^{\mathbf{k},n} = \max\limits_{\mathbf{z}^j \in \mathbf{R}_n^{k_{m-1}}} z_m^j + 1$

Let the optimal value of model ($P^{\mathbf{lb}^{\mathbf{k},n}}$) be $\alpha^{\mathbf{k},n}$ and the corresponding nondominated point be $\mathbf{z(x)}^{\mathbf{k},n}$. Let $\mathbf{K}$ be the set of possible $\mathbf{k}$ vectors. Then, $\alpha^* = \max\limits_{\mathbf{k} \in \mathbf{K}} \alpha^{\mathbf{k},n}$ is the coverage gap of the set $\mathbf{R}$ and the corresponding nondominated point $\mathbf{z}^*$ is the most diverse point.

The same nondominated point might be obtained as the optimal solution in the models solved for different subspaces. In order to prevent this, we keep the list of lower bounds, $\mathbf{lb}^{\mathbf{k},n}$, and the optimal solution, $(\mathbf{z}^{\mathbf{k},n}, \alpha^{\mathbf{k},n})$ in model $P^{\mathbf{lb}^{\mathbf{k},n}}$ after it is solved. Before solving a new model, $P^{\mathbf{lb}^{\mathbf{k}',n}}$, we check whether the new model will have the same optimal solution with a previously solved model by searching the list. The sufficient condition for $(\mathbf{z}^{\mathbf{k},n}, \alpha^{\mathbf{k},n})$ being equal to $(\mathbf{z}^{\mathbf{k}',n}, \alpha^{\mathbf{k}',n})$ is stated in the following corollary:

Corollary 4.1. *If $\boldsymbol{lb}_i^{\boldsymbol{k},n} \leq \boldsymbol{lb}_i^{\boldsymbol{k}',n} \leq z_i^{\boldsymbol{k},n} \forall i$ and*
$\min\limits_{i=1,2,...,m} \left\{z_i^{\boldsymbol{k},n} - \boldsymbol{lb}_i^{\boldsymbol{k}',n}\right\} = \min\limits_{i=1,2,...,m} \left\{z_i^{\boldsymbol{k},n} - \boldsymbol{lb}_i^{\boldsymbol{k},n}\right\}$, *then* $(\boldsymbol{z}^{\boldsymbol{k},n}, \alpha^{\boldsymbol{k},n}) = (\boldsymbol{z}^{\boldsymbol{k}',n}, \alpha^{\boldsymbol{k}',n})$.

Proof. Since $\mathbf{lb}_i^{\mathbf{k},n} \leq \mathbf{lb}_i^{\mathbf{k}',n} \quad \forall i$,
$\alpha^{\mathbf{k}',n} \leq \alpha^{\mathbf{k},n}$,
$\min\limits_{i=1,2,...,m} z_i^{\mathbf{k}',n} - lb_i^{\mathbf{k}',n} \leq \alpha^{\mathbf{k},n}$,
$\min\limits_{i=1,2,...,m} z_i^{\mathbf{k}',n} - lb_i^{\mathbf{k}',n} \leq \min\limits_{i=1,2,...,m} \left\{z_i^{\mathbf{k},n} - lb_i^{\mathbf{k}',n}\right\}$.
This proves the optimality of $(\mathbf{z}^{\mathbf{k},n}, \alpha^{\mathbf{k},n})$ for problem $P^{\mathbf{lb}^{\mathbf{k}',n}}$. $\square$

Besides, the number of infeasible models can also be reduced by keeping a list of the lower bounds resulted in infeasibility. We search this list before solving a new model to detect the infeasible models using Corollary 4.2.

Corollary 4.2. *If* $lb_i^{\mathbf{k},n} \leq lb_i^{\mathbf{k}',n} \quad \forall i$ *and* $P^{\mathbf{lb}^{\mathbf{k},n}}$ *is infeasible, then* $P^{\mathbf{lb}^{\mathbf{k},n}}$ *is also infeasible.*

Proof. Suppose that $(\mathbf{y}^{\mathbf{k}',n}, \Delta^{\mathbf{k}',n})$ is a feasible solution to $P^{\mathbf{lb}^{\mathbf{k}',n}}$. Then, $(\mathbf{y}^{\mathbf{k}',n}, \Delta^{\mathbf{k}',n})$ is also feasible solution to $P^{\mathbf{lb}^{\mathbf{k},n}}$ since $lb_i^{\mathbf{k},n} \leq y_i^{\mathbf{k}',n} \quad \forall i$. This contradicts with the infeasibility of $P^{\mathbf{lb}^{\mathbf{k},n}}$. □

As a result of the previous two propositions, the number of models to be solved can be reduced significantly. Besides, the performance of the algorithm is insensitive to the distribution of the nondominated points in the objective space. This is due to the fact that we use the generated nondominated points to define the subspaces and we update them each time a new nondominated point is generated.

Since we solve more than one model in each iteration and each model gives a nondominated point, the algorithm may generate more than n nondominated points in the n^{th} iteration. Let us assume that $n' > n$ nondominated points are generated although n representative nondominated points are desired by the DM. In this case, optimal subset of nondominated points with cardinality n can be found by solving model (P4) when m nondominated points are available.

Although we solve more than n models, the size of the model solved is always the same during the algorithm. This is the main advantage of Algorithm 1 over the DMA. DMA solves n models to generate n representative nondominated points, but the size of the model to be solved to find j^{th} nondominated point is bigger than the one for $(j-1)^{th}$ nondominated point. However, Algorithm 1 solves the same sized models more than n times.

For small n values, the number of additional binary variables and constraints in model (P2) is rather small and the complexity of model (P2) does not increase much. In this case, solution time of DMA may be lower than that of Algorithm 1. However, the complexity of model (P2) for larger n values increases the solution time exponentially. Therefore, there exists a breakeven cardinality after which the solution time of Algorithm1 is shorter than that of DMA. As the cardinality of the representative subset increases beyond this breakeven point, the differences increase substantially since the solution time increases linearly for Algorithm 1 whereas it is exponential in DMA.

In order to find an initial solution, we solve weighted sum objective function with equal weights. Since the algorithm will find the extreme nondominated points in the earlier iterations, the effect of different initial points on the quality of the subsets is not expected to be significant.

The Algorithm:

Initialization: Let **R** be the set of representative nondominated points. **C** is the list whose components are the lower bounds and corresponding optimal solutions of previously solved models. Similarly, **I** is the list whose components are the lower bounds of the previously solved infeasible models. Initially, $\mathbf{R} = \emptyset, \mathbf{C} = \emptyset, \mathbf{I} = \emptyset$. Let e_α and e_n be threshold coverage gap and cardinality levels, respectively.

Step 0: Let $\mathbf{z}^1$ be the optimal solution of $\max \left\{ \sum_{i=1}^{m} w_i z_i(\mathbf{x}) | \mathbf{x} \in \mathbf{X}, w_i = 1/m \right\}$. $\mathbf{R} = \{\mathbf{z}^1\}, n = 1$.

Step 1:

For each $\mathrm{k} \in \mathrm{K}$**, repeat:**

Search the set **I**. **If** Corollary 4.2 holds, then the model is infeasible. **Else, if** Corollary 4.1 holds, then optimal solution of $P^{\mathbf{lb}^{\mathbf{k},n}}$ is already known. **Else**, the model has to be solved.

Solve $P^{\mathbf{lb}^{\mathbf{k},n}}$. **If** the model is infeasible, then $\mathbf{I} = \mathbf{I} \cup \left\{ (\mathbf{lb}^{\mathbf{k},n}) \right\}$. **Else**, $C = C \cup \left\{ ((\mathbf{lb}^{\mathbf{k},n}), (\mathbf{z}^{\mathbf{k},n}, \alpha^{\mathbf{k},n})) \right\}$.

end

If at least one feasible solution exists, then let $\alpha^* = \max_{\mathbf{k} \in \mathbf{K}} \alpha^{\mathbf{k},n}$ be the coverage gap of the set **R** and $\mathbf{z}^*$ be the most diverse solution. $\mathbf{R} = \mathbf{R} \cup \{\mathbf{z}^*\}, n = n + 1$. Update **K**. **Else**, stop the algorithm.

Step 2: **If** $\alpha^* \leq e_\alpha$ or $n \geq e_n$, then stop the algorithm. **Else**, go to Step1.

Table 4.1: Nondominated points of the example problem

j	z_1	z_2	z_3	α
1	252	209	317	174
2	273	383	140	110
3	352	367	250	49
4	314	285	299	15
5	311	300	294	13
6	332	277	263	10
7	324	287	285	5
8	329	292	258	4
9	291	314	254	2
10	293	369	180	1
11	275	204	300	0

Demonstration of Algorithm1:

We demonstrate the algorithm on a 3-objective knapsack problem which has 11 nondominated points. When the DMA is run on this problem, the nondominated points are found in the sequence as it is given in Table 4.1. The rightmost column of the table gives the coverage gap of the set of already generated nondominated points up to j^{th} point.

Suppose that three nondominated points are already generated and we will next find $\mathbf{z}^4$. In order to generate the current not-yet dominated subspaces, **k** vectors in Table 4.2 are possible and used to set the corresponding lower bound values. Most diverse nondominated point in each subspace is given with its corresponding $\alpha^{\mathbf{k},3}$ value. The nondominated point with the highest $\alpha^{\mathbf{k},3}$ value will be the next representative point chosen. In this example, $\max\limits_{\mathbf{k}} \left\{\alpha^{\mathbf{k},3}\right\} = 49$ and $\operatorname*{argmax}\limits_{\mathbf{k}} \left\{\alpha^{\mathbf{k},3}\right\} = (0,1)$ or alternatively, $(1,0)$. Therefore, the next representative point is chosen as $\mathbf{z}^4$.

In the example given below, we show that if different most diverse points are selected in case of multiple most diverse points, then the subsets in the succeeding iterations and their coverage gaps may change.

Example:

Consider the following set of nondominated points:

$\{\mathbf{z}^1 = (5,5,5), \mathbf{z}^2 = (3,8,15), \mathbf{z}^3 = (9,15,4), \mathbf{z}^4 = (13,10,2)\}$

Suppose the initial nondominated point is $\mathbf{z}^1$ and 2 more nondominated points will be

Table 4.2: Generated subspaces and corresponding nondominated points in Algorithm 1 when n=3

k_1	$\mathbf{R}_3^{k_1}$	k_2	$\mathbf{R}_3^{k_2}$	$Bounds$	lb_1	lb_2	lb_3	$(\mathbf{z}^{\mathbf{k},3}, \alpha^{\mathbf{k},3})$
0	1,2,3	0	1,2,3	$lb_3 \geq \max\limits_{z^j \in R_3^{k_2}} z_3^j + 1$	-	-	318	(infeasible)
0	1,2,3	1	2,3	$lb_2 \geq z_2^1 + 1$ $lb_3 \geq \max\limits_{z^j \in R_3^{k_2}} z_3^j + 1$	-	210	251	((314, 285, 299), 49)
0	1,2,3	2	$\emptyset$	$lb_2 \geq z_2^2 + 1$	-	384	-	(infeasible)
0	1,2,3	3	2	$lb_2 \geq z_2^3 + 1$ $lb_3 \geq \max\limits_{z^j \in R_3^{k_2}} z_3^j + 1$	-	368	141	((293, 369, 180), 2)
1	2,3	0	2,3	$lb_1 \geq z_1^1 + 1$ $lb_3 \geq \max\limits_{z^j \in R_3^{k_2}} z_3^j + 1$	253	-	251	((314, 285, 299), 49)
1	2,3	3	2	$lb_1 \geq z_1^1 + 1$ $lb_2 \geq z_2^3 + 1$ $lb_3 \geq z_3^2 + 1$	253	368	141	((293, 369, 180), 2)
1	2,3	2	$\emptyset$	$lb_1 \geq z_1^1 + 1$ $lb_2 \geq z_2^2 + 1$	253	384	-	(infeasible)
2	3	0	3	$lb_1 \geq z_1^2 + 1$ $lb_3 \geq \max\limits_{z^j \in R_3^{k_2}} z_3^j + 1$	274	-	251	((314, 285, 299), 41)
2	3	3	$\emptyset$	$lb_1 \geq z_1^2 + 1$ $lb_2 \geq z_2^3 + 1$	274	368	-	((293, 369, 180), 2)
3	$\emptyset$	0	$\emptyset$	$lb_1 \geq z_1^3 + 1$	353	-	-	(infeasible)

generated. Initially, $\mathbf{R} = \{\mathbf{z}^1\}$. The corresponding coverage gaps of the remaining points are given below:
$\alpha_R(\mathbf{z}^2)$=10, $\alpha_R(\mathbf{z}^3)$=10, $\alpha_R(\mathbf{z}^4)$=8.
Both $\mathbf{z}^2$ and $\mathbf{z}^3$ are most diverse nondominated points in this case.

Case1: If we select $\mathbf{z}^2$, then
$\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2\}$ and $\alpha_R(\mathbf{z}^3)$=7, $\alpha_R(\mathbf{z}^4)$=8.
Then, $\mathbf{z}^4$ is the most diverse nondominated point and added to the representative set as the last generated point. The resulting subset is
$\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^4\}$ having a coverage gap of 5.

Case2: If we select $\mathbf{z}^3$, then
$\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^3\}$ and $\alpha_R(\mathbf{z}^2)$=10, $\alpha_R(\mathbf{z}^4)$=4.
Then, $\mathbf{z}^2$ is the most diverse nondominated point and added to the representative set as the last generated point. The resulting subset is
$\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3\}$ having a coverage gap of 4.

4.3 Algorithm 2: Territory Defining Algorithm

In this section, we introduce territory defining algorithm (TDA). This algorithm generates a representative set of nondominated points for a threshold value given by the DM such that the coverage gap measure of the representative set is guaranteed to be below the threshold value.

Similar to the DMA, nondominated points are generated iteratively. At each iteration, only one nondominated point is generated and added to the representative subset. But different than the DMA, the nondominated point at iteration n need not to be the worst represented nondominated point in the previous iteration. Instead, the algorithm searches different portions of the objective space at each iteration and generates a nondominated point as a representative of that portion.

Let Δ be the threshold value given by the DM and $\mathbf{y}$ be a nondominated point. We form a hyperspace $\mathbf{H}$ around point $\mathbf{y}$ in the m- dimensional objective space such that $\mathbf{H} = \{y_i - \Delta \leq z_i(\mathbf{x}) \leq y_i + \Delta \quad \forall i, \quad \mathbf{x} \in \mathbf{X}\}$.

Hyperspace $\mathbf{H}$ can be partitioned into 2^m subspaces where there can be nondominated points except two subspaces $\{y_i - \Delta \leq z_i(\mathbf{x}) \leq y_i \quad \forall i, \quad \mathbf{x} \in \mathbf{X}\}$ and $\{y_i \leq z_i(\mathbf{x}) \leq y_i + \Delta \quad \forall i, \quad \mathbf{x} \in \mathbf{X}\}$. The first subspace is dominated by point $\mathbf{y}$ while the second subspace dominates point $\mathbf{y}$. Since $\mathbf{y}$ is a nondominated point, both subspaces cannot contain any nondominated point. Remaining $2^m - 2$ subspace may contain nondominated points.

Proposition 4.1. *Let $\mathbf{y^H}$ be a nondominated point in **H** such that $y_i - \Delta \leq y_i^H \leq y_i + \Delta \quad \forall i$. Then, $\mathbf{y^H}$ is $\Delta-$ dominated by point **y**.*

Proof. It follows directly from the definition of $\mathbf{H}$. □

In this algorithm, we state a different definition of representativeness of point $\mathbf{y}$ for point $\mathbf{y^H}$:

Definition 4.1. *A point **y** is the representative point of $\mathbf{y^H}$ if **y** Δ-dominates $\mathbf{y^H}$.*

Using this definition, $\mathbf{y}$ is said to be the representative point of hyperspace $\mathbf{H}$. Since all nondominated points in this hyperspace are already represented by point $\mathbf{y}$, the search in this hyperspace is eliminated in future iterations. Not only the hyperspace $\mathbf{H}$, but also the space dominated by $\mathbf{H}$, $\mathbf{H}_D$ could also be eliminated from the search space since there cannot be any nondominated point in $\mathbf{H}_D$. For this purpose, we create an artifical point $\mathbf{y}'$ such that $y_i' = y_i + \Delta \quad \forall i$ and use this point instead of $\mathbf{y}$ in the following iterations of the algorithm. So, the space $\mathbf{H}_D$ is dominated by $\mathbf{y}'$ and can be defined as follows: $\mathbf{H}_D = \{\mathbf{z}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}, z_i(\mathbf{x}) \leq y_i' \quad \forall i\}$.

If we substract the space dominated by point $\mathbf{y}$, $\mathbf{y}_D$, and the space dominating point $\mathbf{y}$, $\mathbf{y}_U$, from $\mathbf{H}_D$, then $\mathbf{T_y} = \mathbf{H}_D \setminus \{\mathbf{y}_D \cup \mathbf{y}_U\}$ where hyperspace $\mathbf{T_y}$ is defined as the territory of point $\mathbf{y}$. In Figure 4.1, we show the territories constructed around the first three nondominated points in Table 4.1 with $\Delta = 20$.

Only one nondominated point will be generated in each territory. The addition of the generated points to the subset of already generated nondominated points and excluding the dominated region are handled as in Algorithm 1. However, instead of adding point $\mathbf{y}$ we add point $\mathbf{y}'$ to the subset $\mathbf{R}$ and exclude the region which is not dominated by $\mathbf{R}$ but $\Delta-$dominated by $\mathbf{R}$. Another difference from Algorithm 1 is that we do

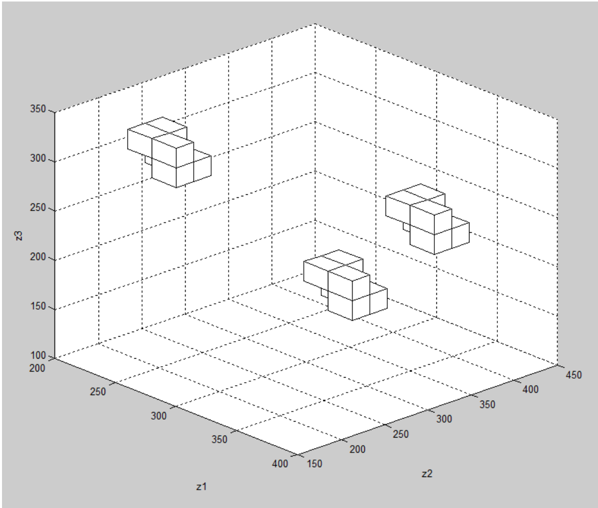

Figure 4.1: Example territories in three dimensional space

not need to solve models in all not-yet dominated subspaces **K** since the aim is not to find the most diverse nondominated point. Instead, any subspace **k** can be chosen and a nondominated point in that subspace will not be $\Delta-$dominated by subset **R**. If there does not exist any nondominated point in all subspaces, then objective space is $\Delta-$dominated by the set **R** as discussed in the following corollary:

Corollary 4.3. *Let R be the representative subset at the end of the algorithm. Then, every nondominated point is $\Delta-$dominated by at least one point in set **R**, i.e. $\forall \mathbf{z} \in \mathbf{Z}_{ND}, \exists \quad \mathbf{y} \in \mathbf{R} : z_i \leq y_i + \Delta \quad \forall i$. In addition, coverage gap of set **R** is less than or equal to Δ: $\alpha_{\boldsymbol{R}} \leq \Delta$.*

Proof. Let $\mathbf{R} = \{\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^n\}$ and the corresponding territory set be $\mathbf{T} = \{\mathbf{T}_{\mathbf{y}^1}, \mathbf{T}_{\mathbf{y}^2}, ..., \mathbf{T}_{\mathbf{y}^n}\}$.

If $\mathbf{z} \notin \mathbf{R}$ and $\mathbf{z}$ is a nondominated point, then $\mathbf{z} \in \mathbf{T}_{\mathbf{y}^j}$ for $\quad \exists \mathbf{T}_{\mathbf{y}^j} \in \mathbf{T}$. Otherwise, it is a dominated point. If $\mathbf{z} \in \mathbf{T}_{\mathbf{y}^j}$, then $z_i \leq y_i^j + \Delta \quad \forall i$ and $\max\limits_{i=1,2,...,m} \left\{z_i - y_i^j\right\} \leq \Delta$ by the definition of $\mathbf{T}_{\mathbf{y}^j}$. Therefore, $\mathbf{z}$ is $\Delta-$dominated by $\mathbf{y}^j$.

The coverage gap of set **R** is: $\alpha_{\mathbf{R}} = \max\limits_{\mathbf{z}\in\mathbf{Z}_{ND}} \left\{ \min\limits_{y\in R} \left\{ \max\limits_{i=1,2,...,m} \{z_i - y_i\} \right\} \right\}$.

Since $\max\limits_{i=1,2,...,m} \{z_i - y_i\} \leq \Delta \quad \forall \mathbf{z} \in \mathbf{Z}_{ND}, \forall \mathbf{y} \in \mathbf{R}$,

$\min\limits_{\mathbf{y}\in\mathbf{R}} \left\{ \max\limits_{i=1,2,...,m} \{z_i - y_i\} \right\} \leq \Delta \quad \forall \mathbf{z} \in \mathbf{Z}_{ND}$. Therefore, $\alpha_{\mathbf{R}} \leq \Delta$. $\square$

The Algorithm:

Initialization: Suppose the DM defines the territory size as Δ, $\Delta \geq 0$ (if we assume that objective values are scaled into [0,1], then $0 \leq \Delta \leq 1$). Let **R** be the set of representative nondominated points, $\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^n\}$, and $\mathbf{D} = \left\{\mathbf{y}^j : y_i^j = z_i^j + \Delta\right\}$. The set **C** consists of vectors corresponding to the lower bounds defining a subspace searched and the optimal solution found, $\mathbf{C} = \{(\mathbf{lb}, \mathbf{z}^*)\}$ where $\mathbf{z}^*$ is the optimal solution of problem $P^{\mathbf{lb}}$ in the subspace defined by lower bounds **lb**. Similarly, $\mathbf{I} = \{\mathbf{lb}\}$ where $P^{\mathbf{lb}}$ in the subspace defined by lower bounds **lb** is infeasible. $\mathbf{K} = \left\{\mathbf{k} = (k^1, k^2, ..., k^{m-1}) : lb_i^{\mathbf{k}} = z_i^{k^i} + 1 \quad \forall i = 1, 2, ..., (m-1)\right\}$. Initially, $\mathbf{R} = \emptyset, \mathbf{D} = \emptyset, \mathbf{K} = \emptyset, \mathbf{C} = \emptyset, \mathbf{I} = \emptyset$.

Step 0: Let $\mathbf{z}^1$ be the optimal solution of $\max\left\{\sum\limits_{i=1}^{m} w_i z_i(\mathbf{x}) | \mathbf{x} \in \mathbf{X}, w_i > 0\right\}$. Then, $n = 1, \mathbf{R} = \{\mathbf{z}^n\}$ and $\mathbf{D} = \{\mathbf{y}^n\}$. Generate all feasible **k** vectors and add to set **K**.

Step 1:
$n = n + 1$. Let $\mathbf{K}' = \mathbf{K}$ and choose $\mathbf{k} \in \mathbf{K}'$. Then, $\mathbf{K}' = \mathbf{K} \setminus \{\mathbf{k}\}$
do
Search set **I**. **If** Corollary 4.2. holds, then the model is infeasible. **Else, if** $lb_i^{'} \leq lb_i \leq z_i^*$ where $(\mathbf{lb}^{'}, \mathbf{z}^*) \in \mathbf{C}$, then $P^{\mathbf{lb}^{\mathbf{k},n}}$ has the optimal solution $\mathbf{z}^*$. Otherwise, we solve the model $P^{\mathbf{lb}^{\mathbf{k},n}}$. If the model is infeasible, then $\mathbf{I} = \mathbf{I} \cup \left\{(\mathbf{lb}^{\mathbf{k},n})\right\}$. Choose another $\mathbf{k} \in \mathbf{K}'$ and $\mathbf{K}' = \mathbf{K} \setminus \{\mathbf{k}\}$.
If $\mathbf{K}' = \emptyset$, **break** and go to Step 3. **Else**, $\mathbf{C} = \mathbf{C} \cup \left\{(\mathbf{lb}^{\mathbf{k},n}, \mathbf{z}^{\mathbf{k},n})\right\}$.
while $P^{\mathbf{lb}^{\mathbf{k},n}}$ is infeasible.

Step 2:
Let the new nondominated point generated be $\mathbf{z}^n$. Then, $\mathbf{R} = \mathbf{R} \cup \{\mathbf{z}^n\}$ and $\mathbf{D} = \mathbf{D} \cup \{\mathbf{y}^n\}$
Update **K**, go to Step 1.

Step 3:
Stop the algorithm. $\alpha_R \leq \Delta$.

There are some parts of the algorithm which should be highlighted. The first one is the model to be solved to find a nondominated point in a subspace **k**. In Algorithm 1, we solve model $P^{\mathbf{lb}^{\mathbf{k},n}}$ to find the most diverse point in subspace **k**. However, in TDA, there is no need to find the most diverse point at an iteration. Finding any nondominated point, which is guaranteed to be not $\Delta-$dominated by the generated representative nondominated points, is sufficient. Therefore, any subspace $\mathbf{k}$ can be chosen and a nondominated point can be found by not necessarily solving $P^{\mathbf{lb}^{\mathbf{k},n}}$.

The second point is that we do not need to solve a model in each subspace $\mathbf{k} \in \mathbf{K}$. In our algorithm, we choose the subspace $\mathbf{k}^*$ which has the highest upper bound on $\alpha^{\mathbf{k},n}$ such that $\mathbf{k}^* = \max_{\mathbf{k} \in \mathbf{K}} \left\{ \min_{i=1,2,\ldots,m} z_i^{IP} - lb_i^{\mathbf{k},n} \right\}$. However, different subspace selection methods can be proposed which may affect the cardinality of the representative set. Furthermore, we solve more than one model at an iteration only when the model is infeasible.

In our preliminary experiments, we compared the performance of our subspace selection rule with the case where subspaces are selected randomly. As long as the chosen subspace is infeasible, one of the remaining subspaces is chosen randomly. In Table 5.8, we reported the number of points generated and the solution time of the algorithm in both case.

Results indicate that, although there is not an obvious difference in the number of points generated, solution time of the case where the subspace selection rule is applied is lower than the other case. This difference is mainly due to the higher number of infeasible models when the subspaces are chosen randomly.

In generating a representative subset, there exists a tradeoff between the coverage gap and the cardinality. As discussed in Chapter 3, we could consider it as a bi-objective problem whose objectives are to minimize the cardinality and coverage gap, respectively. If the DM would like to capture this tradeoff information, we could solve the TDA for different δ values and report the cardinality of the representative subset.

It might be necessary to guide the DM to choose the desired coverage gap threshold. For this purpose, the TDA can first be solved with a high Δ value. According to the number of generated nondominated points, the DM may be satisfied with the result or prefer to improve the coverage gap tolerating more points to be generated. If the DM would like to decrease the threshold further, the previously generated nondominated points are kept and the territories around these points are updated with the new Δ value prior to rerunning the algorithm.

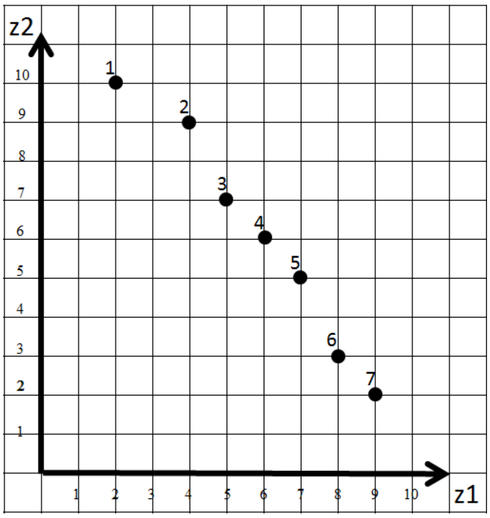

Figure 4.2: Nondominated points of the example problem

Illustration of the Algorithm

Let the following set be the set of nondominated points of a bi-objective integer programming problem:

$\mathbf{z}^1 = (2, 10), \mathbf{z}^2 = (4, 9), \mathbf{z}^3 = (5, 7), \mathbf{z}^4 = (6, 6), \mathbf{z}^5 = (7, 5), \mathbf{z}^6 = (8, 3), \mathbf{z}^7 = (9, 2)$.

In the objective space, the layout of the points of this set is shown on Figure 4.2.

Let $\Delta = 2$ and $\mathbf{y}^1 = (6, 6)$ be the first nondominated point generated. Let $\mathbf{R}$ be the representative subset of nondominated points. Initially, $\mathbf{R} = \{\mathbf{z}^4\}$. The territory defined by point $\mathbf{y}^1, \mathbf{T}_{\mathbf{y}^1}$ is shown on Figure 4.3. $\mathbf{y}^1_U$ is the upper right rectangle of $\mathbf{y}^1$ where there cannot be any nondominated point. In the left lower rectangle of $\mathbf{y}^1, \mathbf{y}^1_D$, all feasible points are dominated by $\mathbf{y}^1$. The rectangles denoted with $\mathbf{T}_{\mathbf{y}^1}$ contain nondominated points $\mathbf{z}^3, \mathbf{z}^5$ and $\mathbf{z}^6$ other than $\mathbf{z}^4$, but they are $\Delta-$dominated by $\mathbf{y}^1 = \mathbf{z}^4$. Therefore, the space $\mathbf{H}_D = \{z_1 \leq 8, z_2 \leq 8\}$ is the $\Delta-$dominated space by $\mathbf{y}^1$ and eliminated from the search space in the following iterations.

Let $\mathbf{D}$ be the $\Delta-$increased nondominated points obtained by adding Δ to each objective value of nondominated points generated. We denote $\Delta-$increased nondominated points with $\mathbf{Y}$. Initially, $\mathbf{Y}^1 = (y^1_1 + \Delta, y^1_2 + \Delta) = (8, 8)$ and $\mathbf{D} = \left\{\mathbf{Y}^1\right\}$. We regard set $\mathbf{D}$ as if it is the real representative nondominated pointset and generate the not-yet dominated subspaces accordingly. So, the next nondominated point will be generated in one of the following subspaces in Table 4.3.

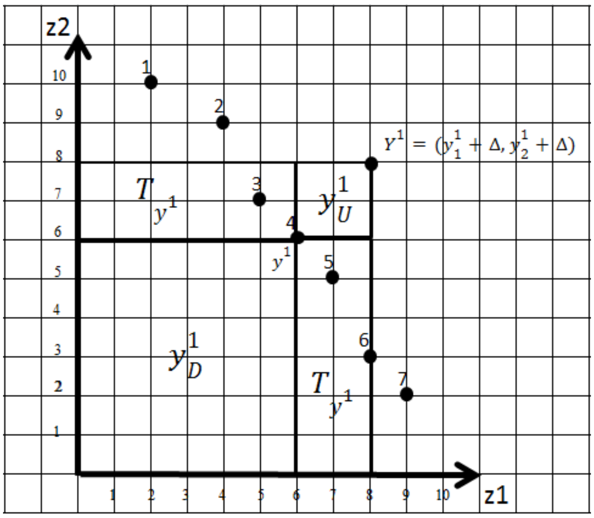

Figure 4.3: Territory around the first representative nondominated point, $\mathbf{z}^4$

Table 4.3: Generated subspaces when n=1 in TDA

k_1	$\mathbf{D}_1^{k_1}$	Bounds	lb_1	lb_2	$\mathbf{z}^j$
0	1	$lb_2 \geq \max_{Y^j \in D_1^{k_1}} Y_2^j + 1$	-	9	$\mathbf{z}^1 = (2, 10)$
1	$\emptyset$	$lb_1 \geq Y_1^1 + 1$	9	-	$\mathbf{z}^7 = (9, 2)$

Suppose that the first subspace is chosen and $\mathbf{y}^2 = \mathbf{z}^1$ is the next nondominated point found after solving $P^{\mathbf{lb}^{\mathbf{k},1}}$ where $\mathbf{k} = \{0, 1\}$. Then, $\mathbf{R} = \{\mathbf{z}^4, \mathbf{z}^1\}$, $\mathbf{Y}^2 = (4, 12)$ and $\mathbf{D} = \{(8, 8), (4, 12)\}$. The territory defined by point $\mathbf{y}^2$ is shown on the Figure 4.4.

Point $\mathbf{z}^2$ is $\Delta-$dominated by the representative point $\mathbf{y}^2$. The next search space is given in Table 4.4.

Table 4.4: Generated supspaces when n=2 in TDA

k_1	$\mathbf{D}_1^{k_1}$	Bounds	lb_1	lb_2	$\mathbf{z}^j$
0	1,2	$lb_2 \geq \max_{Y^j \in D_2^{k_1}} Y_2^j + 1$	-	13	infeasible
1	$\emptyset$	$lb_1 \geq Y_1^1 + 1$	9	-	$\mathbf{z}^7 = (9, 2)$
2	1	$lb_1 \geq Y_1^2 + 1, lb_2 \geq \max_{Y^j \in D_2^{k_1}} Y_2^j + 1$	5	9	infeasible

The only feasible subspace is the second one and $\mathbf{z}^7$ is the next representative nondom-

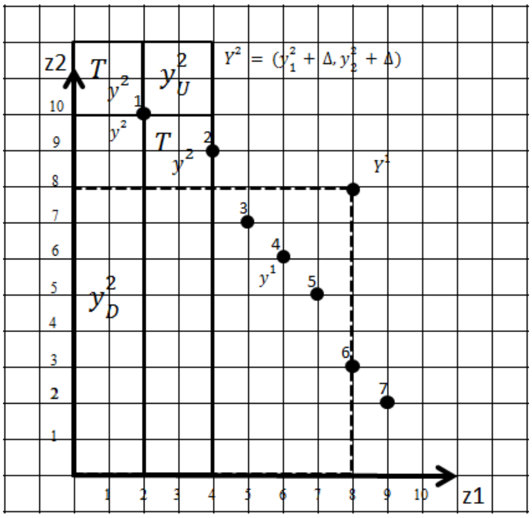

Figure 4.4: Territory around the second representative nondominated point, $\mathbf{z}^1$

inated point. After the generation of $\mathbf{y}^3 = \mathbf{z}^7$, we see that there is no nondominated point which is not $\Delta-$dominated by set $\mathbf{R} = \{\mathbf{z}^4, \mathbf{z}^1, \mathbf{z}^7\}$. If we define a territory around $\mathbf{z}^7$ and then enumerate the not-yet dominated subspaces, all the models will be infeasible. Therefore, the algorithm terminates at this point. The nondominated points $\mathbf{z}^2, \mathbf{z}^3, \mathbf{z}^5$ and $\mathbf{z}^6$ are not generated and represented by the closest point in set $\mathbf{R}$. The representative pointsof not generated nondominated points are as follows:
Let $\mathbf{R}_{\mathbf{z}^j}$ denote the representative of point $\mathbf{z}^j$. Then,

$$\mathbf{R}_{\mathbf{z}^2} = \operatorname*{argmin}_{\mathbf{y}^j \in \mathbf{R}} \left\{ \max_{i=1,2} z_i^2 - y_i^j \right\} = \mathbf{y}^2 = \mathbf{z}^1$$

$$\mathbf{R}_{\mathbf{z}^3} = \operatorname*{argmin}_{\mathbf{y}^j \in \mathbf{R}} \left\{ \max_{i=1,2} z_i^3 - y_i^j \right\} = \mathbf{y}^1 = \mathbf{z}^4$$

$$\mathbf{R}_{\mathbf{z}^5} = \operatorname*{argmin}_{\mathbf{y}^j \in \mathbf{R}} \left\{ \max_{i=1,2} z_i^5 - y_i^j \right\} = \mathbf{y}^1 = \mathbf{z}^4$$

$$\mathbf{R}_{\mathbf{z}^6} = \operatorname*{argmin}_{\mathbf{y}^j \in \mathbf{R}} \left\{ \max_{i=1,2} z_i^6 - y_i^j \right\} = \mathbf{y}^3 = \mathbf{z}^7$$

The point $\mathbf{z}^2$ is the one which has the largest $\max_{i=1,2} z_i^j - (R_{\mathbf{z}^j})_i$ value, which is 2, and it is the most diverse nondominated point. Accordingly, $\alpha_D = 2 \leq \Delta = 2$ as it is stated in Corollary 4.3.

We also tested the sensitivity of the TDA to different starting points. For this purpose, we conducted preliminary experiments whose results are reported in Table 5.7 in Chapter 5. Results indicate that, the starting point of the algorithm does not affect the number of nondominated points gerenated.

4.4 Algorithm 3: Surface Projection Algorithm

As discussed in Section 4.2, Algorithm 1 finds the most diverse point at each iteration given the already generated nondominated points. The algorithm can be stopped when the desired number of points are generated. The resulting subset of nondominated points have the coverage gap being equal to the coverage gap of the last nondominated point generated. The iterative nature of the algorithm causes a myopic search over the nondominated point set. However, if the desired number of nondominated points is known at the beginning of the algorithm, it might be possible to generate a subset with a better coverage gap value. With this in mind, we introduce the Surface Projection Algorithm (SPA). The purpose of this algorithm is to find a well-distributed subset with a minimum coverage gap given the desired number of nondominated points.

In order to generate a well-distributed subset of nondominated points, we need to know the possible locations of the nondominated points. Therefore, we developed the following idea: We approximate the nondominated frontier with a hypersurface. Then, we discretize this surface by generating points which represent the surface well. Among these hypothetical points, we choose as many solutions as the desired number of nondominated points such that they are well distributed on the surface. Finally, we project from these hypothetical representative points to the true nondominated points.

The quality of the subsets generated by SPA mainly depends on the quality of the approximation to the nondominated frontier and the coverage gap of the hypothetical nondominated points. If we can approximate the nondominated frontier well and find a diverse hypothetical point set, then we may expect that we obtain a well-distributed nondominated point set.

As we discussed in the previous sections, there is a tradeoff between the coverage gap and the cardinality of the subset in the problem of choosing a representative subset. In our preliminary experiments, we observed that the desired coverage gap values can be obtained even with a small percentage of all nondominated points. The preliminary results also show that there exist a "knee" point such that generating points after the knee point provides only slight improvements of the coverage gap of the set. Since generating a nondominated point in MOIP problems is computationally expensive, the knee point presents a good choice between the cardinality and the coverage gap of the subset.

In Figure 5.1, we can observe the knee point around 50 points for the MOKP100 problem. Generating about 50 representative points, the coverage gap can be reduced around 0.05. However, to improve this value further to 0.02 or below, many more nondominated points have to be generated. Then, we argue that that generating more than 50 points is not worthwhile even for large-sized problems like MOKP100 which has about 3000 nondominated points on the average. Therefore, in SPA, we aim to generate subsets with small cardinalities.

4.4.1 Fitting a Hypersurface to Approximate the Nondominated Frontier

Köksalan [13] used L_p functions to approximate the nondominated frontiers of bi-objective scheduling problems. Afterwards, Köksalan and Lokman [14] adapted this approach to approximate the nondominated frontiers of MOCO problems. Lokman and Köksalan [18] employed this approximation and developed an interactive procedure to generate highly preferred nondominated points for MOIP problems. Karasakal and Köksalan [11] used L_p functions to approximate the nondominated frontier of a MOLP problem and generate representative nondominated points by projecting from the chosen points on the approximate surface to the true nondominated frontier. Computational experiments in these studies showed that, use of L_p functions to approximate the nondominated frontiers performs well. Therefore, we also used L_p functions to approximate the nondominated frontiers of MOIP problems.

We fit the L_p function in the scaled objective space. In order to scale the objectives, we use the ideal and nadir points of the problem, $\mathbf{z}^{IP}, \mathbf{z}^{NP}$. If we denote the scaled vector $\mathbf{z} = (z_1, z_2, ..., z_m)$ with $\mathbf{v} = (v_1, v_2, ..., v_m)$, then $(v_1, v_2, ..., v_m) = (\frac{z_1 - z_1^{NP}}{z_1^{IP} - z_1^{NP}}, \frac{z_2 - z_2^{NP}}{z_2^{IP} - z_2^{NP}}, ..., \frac{z_m - z_m^{NP}}{z_m^{IP} - z_m^{NP}})$ so that $0 \leq v_i \leq 1 \quad \forall i$.

We define our hypersurface with the following equation:
$L_p(\mathbf{v}) = [\sum_{i=1}^{m} \lambda_i v_i^p]^{1/p} = 1, \quad p > 0$ and λ is a nonnegative weight vector. Given a

vector of weights, λ, we need to find the p value such that this equation is satisfied by a set of some reference points. We use the following points as the reference points to fit the hypersurface:
$\mathbf{v}^0 = (r_1, r_2, ..., r_m)$, $\mathbf{v}^1 = (1, 0, ..., 0)$, $\mathbf{v}^2 = (0, 1, ..., 0)$, ..., $\mathbf{v}^m = (0, 0, ..., 1)$, where $\mathbf{v}^0$ is found by solving equal weighted augmented Tchebycheff program which gives the nondominated point closest (in terms of Tchebycheff distance) to the ideal point. Let $\mathbf{S}$ be the set of those reference points used to fit the hypersurface. The reference points are chosen in this way to represent the middle and extreme parts of the nondominated frontier so that a general shape of the frontier is captured well.

The next step is to find the p value that satisfies the equation at the reference points. The reference points except for $\mathbf{v}^0$ correspond to extremes of the surface and satisfy the equation for every $p > 0$. Therefore, we need to find the p value such that $L_p(\mathbf{r}) = [\sum_{i=1}^{m} r_i^p]^{1/p} = 1$, where $\mathbf{r} = \mathbf{v}^0$, $p > 0$ and we take $\lambda = (1, 1, ..., 1)$.

In Figure 4.5 we show the L_p functions plotted for different values of p. Consider the nondominated frontier in Figure 4.2. In this case, $\mathbf{v}^0 = \mathbf{z}^3$ or $\mathbf{z}^4$. Lets take it as $\mathbf{z}^4$. In addition, $\mathbf{v}^1 = \mathbf{z}^1$ and $\mathbf{v}^2 = \mathbf{z}^7$. Then, p value satisfying the L_p equation for this set of reference points is equal to 1.11.

Köksalan and Lokman [14] used the same type of reference points and L_p function. For comparative purposes, they also approximate the nondominated frontier using all nondominated points as reference points. To do that, they solved a nonlinear program that assigns a representative point on the surface for each nondominated point that is at minimum Euclidean distance. The model finds the p value that minimizes the average Euclidean distance and they report this distance as a quality indicator for the approximation. Their results show that the quality of the surfaces when all nondominated points are used to fit the surface provides only very small improvements compared to the case where only set $\mathbf{S}$ is used. Therefore, they recommend to use only set $\mathbf{S}$ to fit a hypersurface since finding a nondominated point is difficult in MOCO problems.

Karasakal and Köksalan [11] used a slightly different way to fit the hypersurface. After fitting an L_p function with reference point set $\mathbf{S}$ and finding the corresponding p value, more reference points are used to better represent the nondominated

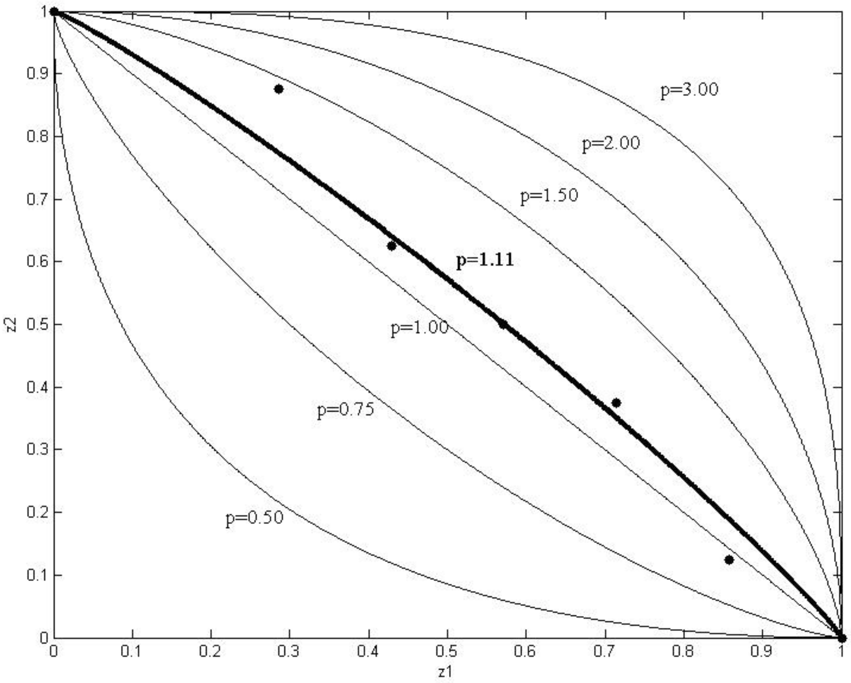

Figure 4.5: L_p functions plotted for different values of p

frontier. For this purpose, they generate reference points by solving equal weighted Tchebycheff programs by restricting the objectives. Let $\left\{\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^k\right\}$ be the set of reference points. Then, they find the weight vector λ^* such that it minimizes the sum of squared deviation of the L_p function from the reference points. That is, $E(\boldsymbol{\lambda}^*) = \min\limits_{\lambda>0}(\sum\limits_{j=1}^{k}(1 - \sum\limits_{i=1}^{m}\lambda_i(z_i^j)^p))$. Since the sum of squared deviation expression is twice differentiable with respect to $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^*$ can be easily found by differentiating with respect to λ and equating to zero.

However, fitting the surface with many reference points is contradictory to the idea of generating a small set of nondominated points. Therefore, we can use only a few reference points for the purpose of surface fitting.

Another critical part of surface fitting is the scaling of the objectives. Finding the nadir point is not straightforward for MOIP problems with more than two objectives. Lokman [17] developed exact methods to find the nadir point for MOIP problems. Therefore, the nadir point can be found prior to the surface fitting stage of the algorithm and used to scale the objectives. In addition, the nadir point can be approximated using the payoff table which is formed by storing the objective values of nondominated points giving the maximum value for each objective row by row. Then, the minimum value in each column gives the corresponding objective value of the approximated nadir point. The payoff nadir may overestimate (underestimate) the exact nadir point for maximization (minimization) type problems but Köksalan and Lokman [14] showed that use of the payoff nadir also works well for MOCO problems.

4.4.2 Finding a Diverse Set of Points on the Surface

In order to find a well-distributed set of points on the continuous surface, we first discretize the surface and then pick a diverse subset of the generated points. We solve the model (P4) in Chapter 3 to find the subset having the minimum coverage gap among the generated points on the surface.

Let K be the number of points we generate to discretize the surface. These points should be equally spaced along the surface in order to represent all parts of the surface equally. For this purpose, we form a grid of equally-spaced points in $(m-1)$ dimensional objective space where m is the number of objective functions. Since the surface is inside a unit hypervolume due to the scaling of objectives, we divide the range $[0,1]$ into equal parts for each objective. To be able to generate at least K grid points, we need $n = \lceil K^{\frac{1}{m-1}} \rceil$ points at each objective. If we denote the length of each interval with Δ, then $\Delta = \frac{1}{n-1}$. For criterion i, we pick the following points which are Δ unit distance apart.

$g_i^1 = 0,$
$g_i^2 = \Delta,$
$g_i^j = (j-1)\Delta,$
$g_i^n = (n-1)\Delta = 1,$
Then, we form the $(m-1)$ dimensional grid as $\mathbf{g}^j = (g_1^j, g_2^j, ..., g_{m-1}^j, 0)$.
To project $\mathbf{g}^j$ to the surface, we update g_m^j as follows: $g_m^j = 1 - L_p(\mathbf{g}^j)$.

It is possible that g_m^j is negative for some grid points. However, we are interested in the part of the surface in the nonnegative orthant. Therefore, we discard those points. Let $\mathbf{G}$ be the set of points on the fitted surface.

Since we have the set of points on the surface, we can find the optimal subset in terms of coverage gap measure at a particular cardinality. We use model (P4) for this purpose which we restate here for the sake of completeness:

(P4):

Min α

s. to.

$$
\begin{aligned}
&\alpha \geq \beta_j \quad j = 1, 2, ..., |\mathbf{G}| \\
&\beta_j = \sum_{k=1}^{|\mathbf{G}|} d_{jk} a_{jk} \quad j = 1, 2, ..., |\mathbf{G}| \\
&\sum_{k=1}^{|\mathbf{G}|} a_{jk} = 1 \quad \forall j = 1, 2, ..., |\mathbf{G}| \\
&\sum_{k=1}^{|\mathbf{G}|} g_k = |\mathbf{R}| \\
&a_{jk} \leq g_k \quad \forall j, k = 1, 2, ..., |\mathbf{G}| \\
&\alpha, \beta_j \geq 0, g_k, a_{jk} \in \{0, 1\}
\end{aligned}
\tag{4.2}
$$

If $g_k = 1$, then k^{th} point on the surface is selected as a representative hypothetical point. Optimal α value indicates the coverage gap of the representative hypothetical point subset over the approximated surface. So, if the surface fits well to the nondominated frontier, we could expect the true nondominated points to be very close to the representative hypothetical points. That is, if we replace the representative hypothetical points with the closest true nondominated points, we could expect the obtained set of nondominated points to be of high quality in terms of the coverage gap measure.

There exists a restriction on the generated number of points on the surface to make this procedure practical. If the cardinality of set **G** is large, then the solution time of (P4) can be quite high. According to our preliminary experiments, $|\mathbf{G}|$ should not be more than 200 to solve model (P4) in a reasonable solution time. In Chapter 5, we report the CPU time required to fit the surface and generate the hypothetical representative points in case the surface is discretized with approximately 200 points.

Due to the restriction on $|\mathbf{G}|$, we incur some error in the discretization of the surface.

We denote this error with $e_{\mathbf{G}}$. In addition, let α_s be the coverage gap of the representative hypothetical point subset with cardinality $|\mathbf{R}|$ if $e_{\mathbf{G}} = 0$. If the optimal solution of (P4) is α^*, then $\alpha_s = \alpha^*$. However, since we tolerate some error in the discretization of the surface, $e_{\mathbf{G}} > 0$ and $\alpha_s > \alpha^*$.

An alternative way to find a diverse set of hypothetical points would be to apply a heuristic method. If an efficient heuristic method can be developed, then more points can be generated on the surface. However, the quality of subset generated is not guaranteed to be higher than the case where the optimal subset is found with fewer points on the surface.

4.4.3 Generating Representative Nondominated Points

Let **H** be the representative subset of hypothetical points, $\mathbf{H} = \left\{\mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^{|\mathbf{R}|}\right\}$. Our aim is to find a nondominated point close to each hypothetical point in set **H**. For this purpose, we use achievement scalarizing functions, $a : \mathbb{R}^m \to \mathbb{R}$. Then, the problem is given by

$$\min_{\mathbf{x}\in\mathbf{X}} a(\mathbf{z(x)}). \tag{4.3}$$

We use the following achivement scalarizing function which is strongly increasing.

$$(ASF) : a(\mathbf{q}, \mathbf{z(x)}, \boldsymbol{\lambda}) = \max_{i=1,2,...,m} \lambda_i(q_i - z_i(\mathbf{x})) - \epsilon(\sum_{i=1}^{m} z_i(\mathbf{x})), \quad \mathbf{x} \in \mathbf{X} \tag{4.4}$$

In Equation 4.5, **q** is the reference point, $\boldsymbol{\lambda}$ is a vector of positive weights and $\epsilon > 0$. Then, the achievement scalarizing program (ASP) is as follows:

$$(ASP) : \min_{\mathbf{x}\in\mathbf{X}} \left\{ \max_{i=1,2,...,m} \lambda_i(q_i - z_i(\mathbf{x})) - \epsilon(\sum_{i=1}^{m} z_i(\mathbf{x})) \right\} \tag{4.5}$$

Since a is strongly increasing, (ASP) gives a nondominated point (Wierzbicki [32]).

The following linear program can be used to solve (ASP):

$$
\begin{aligned}
&(P_{\mathbf{q}}): \\
&\quad \text{Min} \quad \gamma - \epsilon(\sum_{i=1}^{m} z_i(\mathbf{x})) \\
&\quad \text{s. to.} \\
&\qquad \gamma \geq \lambda_i(q_i - z_i(\mathbf{x})) \quad \forall i \\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned}
\tag{4.6}
$$

We solve $(P_{\mathbf{q}})$ for each reference point $\mathbf{q} = \mathbf{h}^j, j = 1, 2, ..., |\mathbf{R}|$ with $\lambda_i = 1$. If the optimal objective value of $(P_{\mathbf{q}})$ is close to zero, then we obtain nondominated points close to representative hypothetical points. It means that, we can generate a diverse set of nondominated points with this procedure.

In the best case, we solve $(P_{\mathbf{q}})$ $|\mathbf{R}|$ times. However, it is possible that we generate the same nondominated point even when $(P_{\mathbf{q}})$ is solved with different reference points. In this case, we select a new hypothetical point on the surface and use it as the new reference point to generate a new nondominated point. We select this new point as the most diverse hypothetical point, $\mathbf{g}^d$ given set $\mathbf{H}$. Unless we generate $|\mathbf{R}|$ nondominated points, we update set $\mathbf{H}$ as $\mathbf{H} = \mathbf{H} \cap \left\{\mathbf{g}^d\right\}$ and find the next most diverse hypothetical point. The index of the most diverse hypothetical point can be found as:

$$
d = \arg \max_{k=1,2,\ldots,K} \left\{ \min_{j=1,2,\ldots,|\mathbf{H}|} \left\{ \max_{1 \leq i \leq m} g_i^k - h_i^j \right\} \right\} \tag{4.7}
$$

4.4.4 Demonstration of SPA on an example

We consider the following MOKP.

$$
\begin{aligned}
&\text{(MOKP):} \\
&\quad \text{"Max"} \quad \{\mathbf{c}_1 x, \mathbf{c}_2 x, ..., \mathbf{c}_m x\} \\
&\quad\quad \text{s. to.} \\
&\qquad \sum_{j=1}^{n} w_j x_j \leq W \\
&\qquad x_j \in \{0, 1\} \quad j = 1, 2, ..., n
\end{aligned}
\tag{4.8}
$$

Let $m = 3$ and $n = 25$. The objective function coefficient vectors, $\mathbf{c}_i$, and weight vector **w** are given in Table 4.5.

Table 4.5: Objective function and weight coefficients for the example problem

$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$	w
55	52	56	53
65	65	91	52
46	58	34	28
38	64	13	23
32	46	72	96
62	67	34	70
53	72	67	13
33	96	74	62
88	42	89	33
35	30	72	69
74	74	79	18
65	37	25	44
32	78	34	66
11	35	92	12
97	48	30	64
97	25	70	87
10	80	63	41
20	82	73	37
84	88	54	61
23	63	10	30
48	41	97	60
63	40	89	14
59	55	25	69
92	41	75	85
39	78	70	95

There are 21 nondominated points of this example problem. Suppose that we would like to generate only 5 nondominated points.

Stage1: Fitting an L_p surface

The ideal and nadir points of the problem are $\mathbf{z}^{IP} = (965, 1067, 1105)$ and $\mathbf{z}^{NP} = (701, 747, 848)$. Then, the efficient ranges for each objective are 264, 320 and 257, respectively. We first need to find a reference point to fit the surface. We solve the following augmented Tchebycheff program:

$$
\begin{aligned}
&\text{Min} \quad \gamma - \epsilon(\sum_{i=1}^{3} z_i(\mathbf{x})) \\
&\text{s. to.} \\
&\qquad \gamma \geq (965 - z_1(\mathbf{x})) \\
&\qquad \gamma \geq (1067 - z_2(\mathbf{x})) \\
&\qquad \gamma \geq (1105 - z_3(\mathbf{x})) \\
&\qquad \mathbf{x} \in \mathbf{X}
\end{aligned} \tag{4.9}
$$

We find the closest nondominated point to the ideal point as, $\mathbf{z} = (867, 945, 1012)$. The scaled version of the reference point is $\mathbf{r} = (0.63, 0.62, 0.64)$. Then, p value is found as 2.3666 when we solve the equation $L_p((0.63, 0.62, 0.64)) = 0.63^p + 0.62^p + 0.64^p = 1$.

We set $K = 250$ and generate $n = \lceil 250^{\frac{1}{2}} \rceil = 16$ grid points on each axis. When these grid points are projected on to the surface, we find that 200 of them are in the nonnegative orthant. So, we use $|\mathbf{G}| = 200$ points to discretize the surface. The fitted surface is shown on Figure 4.6.

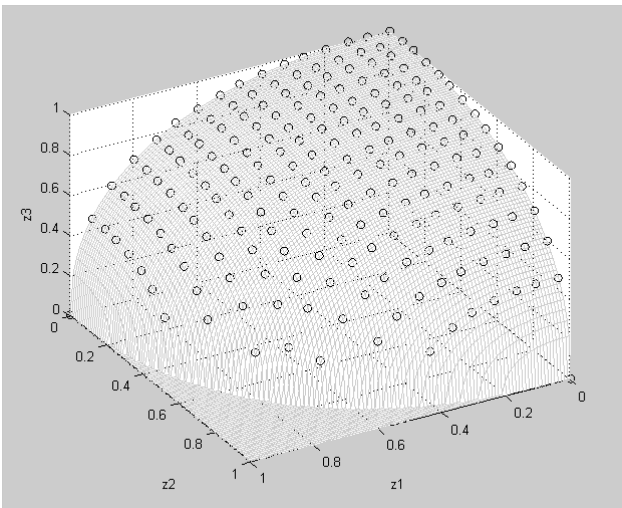

Figure 4.6: L_p surface with $p = 2.3666$ and discretization of the surface

Stage2: Finding a diverse set of hypothetical points

Since only 5 nondominated points are to be generated, we are going to solve model (P4) with $|\mathbf{R}| = 5$. We find the representative hypothetical points giving the minimum coverage gap value, $\alpha^* = 0.16$, and show them as filled hollows on Figure 4.7. Set $\mathbf{H}$ consists of following points: $(0.40, 0.87, 0.48)$, $(0.47, 0.47, 0.84)$, $(0.60, 0.60, 0.68)$, $(0.67, 0.67, 0.54)$, $(0.87, 0.40, 0.48)$.

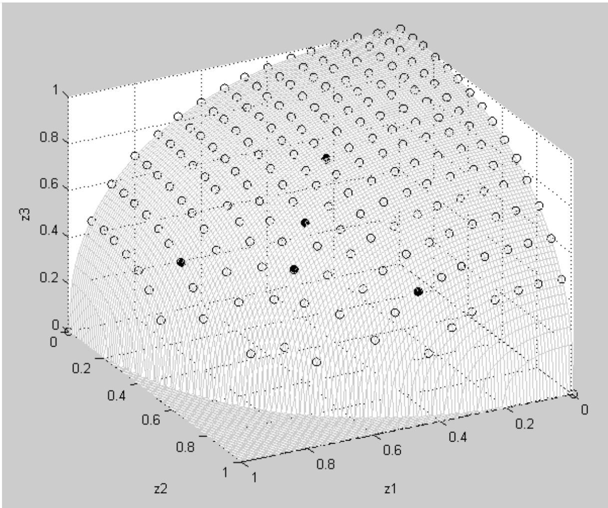

Figure 4.7: 5 representative hypothetical points with $\alpha^* = 0.16$

Stage3: Generating representative nondominated point subset

Once we have the representative hypothetical points, we solve model $(P_{\mathbf{q}})$ for each $\mathbf{q} = \mathbf{h}^j, \mathbf{h}^j \in \mathbf{H}$. For example, if we take the first hypothetical point $\mathbf{h}^1 = (0.47, 0.47, 0.84)$, we solve the model $(P_{\mathbf{h}^1})$. The optimal solution of $(P_{\mathbf{h}^1})$ gives the nondominated point $\mathbf{z}^* = (0.48, 0.87, 0.46)$. In the nonscaled objective space,$\mathbf{z}^*$ corresponds to the first representative nondominated point $\mathbf{y}^1 = (829, 1026, 967)$.

$(P_{\mathbf{h}^1})$

$$\text{Min} \quad \gamma - \epsilon(\sum_{i=1}^{3} z_i(\mathbf{x}))$$

s. to.

$$\begin{aligned} &\gamma \geq (0.40 - z_1(\mathbf{x})) \\ &\gamma \geq (0.87 - z_2(\mathbf{x})) \\ &\gamma \geq (0.48 - z_3(\mathbf{x})) \\ &\mathbf{x} \in \mathbf{X} \end{aligned} \tag{4.10}$$

We solve $(P_{\mathbf{h}^j})$ for $j = 1, 2, ..., 5$ and generate the representative nondominated points, $\mathbf{y}^1 = (829, 1026, 967)$, $\mathbf{y}^2 = (845, 967, 1071)$, $\mathbf{y}^3 = (858, 950, 1018)$, $\mathbf{y}^4 = (894, 974, 1004)$, $\mathbf{y}^5 = (927, 886, 969)$.

We show the distribution of the generated points together with all nondominated points on Figure 4.8. This subset of points achieve $\gamma = 0.18$ coverage gap level.

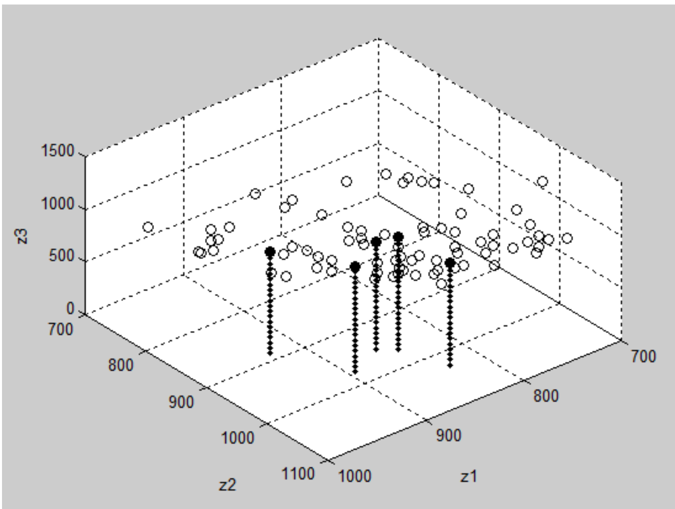

Figure 4.8: 5 representative nondominated points with $\gamma^* = 0.18$.

In this example, $(P_{\mathbf{q}})$ is solved only 5 times since we did not find any already generated nondominated point.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, we test the performance of the algorithms presented in the previous chapter. We conducted the experiments on multi-objective knapsack problem (MOKP) and multi-objective assignment problem (MOAP) with three objectives ($m = 3$). For both problems, we considered three problem sizes, l, with 25, 50 and 100 items for MOKP (MOKP25, MOKP50, MOKP100) and with 10, 20 and 30 jobs for MOAP (MOAP10, MOAP20, MOAP30). We define the problems as given below:

The Multi-objective Knapsack Problem

$(MOKP)$

$$
\begin{aligned}
&\text{"Max"} \quad \{z_1(\mathbf{x}), z_2(\mathbf{x}), ..., z_m(\mathbf{x})\} \\
&\quad \text{s. to.} \\
&\qquad \sum_{j=1}^{l} w_j x_j \leq W \\
&\qquad x_j \in \{0, 1\} \quad \forall j
\end{aligned}
\tag{5.1}
$$

where

$z_i(\mathbf{x}) = \sum\limits_{j=1}^{l} c_{ij} x_j$,

c_{ij} is the coefficient of item j in criterion i,

w_j is the weight of item j in the knapsack,

W is the capacity of the knapsack and,

x_j is the decision variable which takes the value of 1 if it is included in the knapsack, otherwise it is 0.

We generate our test instances as it is in Köksalan and Lokman [14]. We randomly generate objective function and weight coefficients from discrete uniform distribution such that $c_{ij}, w_j \in [10, 100]$ and set the knapsack capacity to the half of the total weight of all items, $W = \frac{\sum_{j=1}^{n} w_j}{2}$. We generate ten random instances for each problem size.

The Multi-objective Assignment Problem

$(MOAP)$

$$
\begin{aligned}
&\text{"Min"} \quad \{z_1(\mathbf{x}), z_2(\mathbf{x}), ..., z_m(\mathbf{x})\} \\
&\quad \text{s. to.} \\
&\qquad \sum_{k=1}^{l} x_{jk} = 1 \quad \forall j \\
&\qquad \sum_{j=1}^{l} x_{jk} = 1 \quad \forall k \\
&\qquad x_{jk} \in \{0, 1\} \quad \forall j, k
\end{aligned}
\tag{5.2}
$$

where

$z_i(\mathbf{x}) = \sum_{j=1}^{n} \sum_{k=1}^{n} c_{ijk} x_{jk}$,

c_{ijk} is the coefficient of the assignment of job j to person k in criterion i and,

x_{jk} is the decision variable which takes value 1 if job j is assigned to person k, otherwise it is 0.

We generate c_{ijk} coefficients from discrete uniform distribution in the interval [1,20]. Similar to the MOKP, we use ten random instances for each problem size.

We run the algorithms on a computer with Intel(R)Core(TM)i7-4770S CPU@ 3.10 GHz, 16 GB RAM and Windows 7. Algorithms are coded in C programming language using Microsoft Visual Studio 2010 Professional environment. We use IBM ILOG CPLEX 12.5 to solve mathematical models.

Firstly, we present the number of nondominated points for each instance in Table 5.1. We used Algorithm 1 to generate all nondominated points by putting the stopping condition as coverage gap and setting it to zero. For MOAP30, we stopped the algorithm when the number of nondominated points generated reached 4000 due to time considerations.

Table 5.1: Number of nondominated points of all test instances

	MOKP25	MOKP50	MOKP100	MOAP10	MOAP20	MOAP30*
1	21	161	1831	69	1009	2195
2	80	496	4236	192	1695	4000
3	78	696	4479	331	990	4000
4	53	354	4132	166	1309	4000
5	72	730	4431	412	2398	4000
6	19	219	2453	82	689	4000
7	36	216	3256	162	1527	4000
8	36	149	1455	116	1977	4000
9	71	580	3352	186	1483	3421
10	91	570	3264	136	2398	4000

* The algorithm was stopped after generating 4000 nondominated points.

In order to see the solution time improvement of our first algorithm, Algorithm 1, over DMA, we report solution time results of both algorithms in comparison with each other. We run both algorithms on MOKP50. Since the solution time of DMA exponentially increases, we generated only up to 120 nondominated points with this algorithm. For both algorithms, we recorded the time elapsed from the beginning of the algorithm at each iteration and compared them for different number of already generated nondominated points. We present the results in Table 5.2.

Table 5.2: CPU time (seconds) comparison of Algorithm 1 and DMA on MOKP50

	Algorithm 1		DMA	
$\|\mathbf{R}\|$	Avg.	Std. Dev.	Avg.	Std. Dev.
5	1.40	1.01	0.47	0.03
50	36.83	10.51	81.43	27.10
100	86.37	34.77	3863.25	1989.54
120	103.54	44.75	15291.65	8096.38

Table 5.2 shows that DMA requires excessive solution times even when the number of nondominated points to be generated is rather small. This is due to the increase in the number of additional binary variables and constraints at each iteration. Partitioning the not yet dominated feasible space into subspaces eliminates solving larger models

although much more model have to be solved. However, results reveal that solving same-sized model several times requires much less computation time than solving larger models.

As it is described in Chapter 4, Algorithm 1 finds the most diverse nondominated point at each iteration which also gives the coverage gap of the current subset of nondominated points. In order to observe how the coverage gap improves throughout the algorithm, we report the diversities of subsets at different cardinality levels in Table 5.3.

In Table 5.3, we report the averages and standart deviations of the performance measures for ten instances per problem. We used at most five cardinality levels to report the measures. If the number of nondominated points of an instance is smaller than the specified cardinality value, then that instance is excluded from the calculation of averages. For small problems, the results at high cardinality levels are also not reported.

We reported the coverage gap values in scaled values in [0,1] range using the efficiency ranges of objectives. The efficiency range, e_i, for objective i can be calculated as $e_i = z_i^{IP} - z_i^{NP}$. If the binding constraint when the most diverse solution found corresponds to objective i, then the optimal α value is scaled into [0,1] range by dividing it to the efficiency range of objective i. Since we continued the algorithm until it generated all nondominated points, we know the ideal and nadir points for each instance except for the ones belonging to MOAP30. For that problem, we use an approximate nadir value.

Table 5.3 shows that Algorithm 1 performs well in terms of the coverage gap measure, generating only few nondominated points even for large problems to obtain a diverse set of nondominated points. For instance, the algorithm represents the nondominated frontier with only five nondominated points at a coverage gap level of 0.25 to 0.30. That is, each remaining nondominated point is guaranteed to have at most 25%-30% better value than its representative in any objective. The coverage gap improves and takes a value between 0.05 and 0.10 when the number of representative points is between 25 and 50. Table 5.3 also presents the number of models to be solved and corresponding CPU times in order to generate a well-representative subset.

Table 5.3: Coverage gap of the subsets generated by Algorithm 1

		* Coverage Gap α		CPU time (sec.)		Num. of models	
Problem	**|R|**	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP25	5	0.22	0.04	0.64	0.17	16.60	1.84
	25	0.06	0.02	7.40	1.78	164.88	29.51
	50	0.02	0.01	15.87	1.31	356.50	32.50
MOKP50	5	0.24	0.02	1.40	1.01	18.40	1.07
	25	0.09	0.02	12.74	2.56	224.10	34.00
	50	0.05	0.02	36.83	10.51	639.80	161.49
	100	0.03	0.01	86.37	34.77	1471.90	518.16
MOKP100	5	0.28	0.05	1.24	0.51	18.60	1.43
	25	0.12	0.02	20.53	2.31	280.60	21.26
	50	0.06	0.01	76.75	9.50	1013.20	97.46
	100	0.04	0.00	227.33	29.56	2889.60	239.35
	500	0.01	0.00	1661.59	427.54	18570.20	3831.50
MOAP10	5	0.28	0.06	0.93	0.46	17.50	2.37
	25	0.08	0.02	11.23	2.44	212.10	31.66
	50	0.04	0.02	24.88	7.15	446.80	117.07
	100	0.02	0.01	48.81	18.21	857.25	274.26
MOAP20	5	0.27	0.03	1.61	0.55	18.40	1.71
	25	0.12	0.02	24.96	4.24	260.60	38.65
	50	0.07	0.01	82.88	15.92	829.70	139.71
	100	0.04	0.01	219.94	46.59	2116.50	386.92
	500	0.01	0.00	825.38	286.17	7556.50	2400.88
MOAP30	5	0.30	0.03	2.13	0.37	17.90	1.79
	25	0.12	0.01	46.04	7.74	300.00	34.65
	50	0.07	0.01	169.99	38.71	1034.10	164.17
	100	0.05	0.01	546.14	140.82	3175.60	544.03
	500	0.02	0.00	3376.59	1439.70	17939.40	5505.01

* Coverage Gap values are given in the scaled form, $0 \leq \alpha \leq 1$.

In Table 5.4, we present the percentages of the number of points in the subsets to the number of all nondominated points. We also give the percentage of the CPU time to generate the subset to the CPU time of generating all nondominated points. For large-sized problems, even the largest reported subsets constitute a small portion of all nondominated points. In addition, generating all nondominated points for those problems requires a substantial amount of computation time and representative subsets can be obtained in a rather small percentage it.

Table 5.4: Comparison of generated subsets to nondominated point set of the problems

		Pct. of all points (%)		Pct. of total CPU (%)	
Problem	**|R|**	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP25	5	11.95	7.56	7.30	6.51
	25	43.35	17.08	47.78	17.04
	50	69.29	13.47	77.48	7.79
MOKP50	5	1.66	1.02	1.30	2.01
	25	8.31	5.10	8.98	7.19
	50	16.62	10.20	22.85	14.14
	100	33.24	20.39	47.44	21.50
MOKP100	5	0.17	0.08	0.03	0.04
	25	0.87	0.39	0.50	0.36
	50	1.74	0.78	1.87	1.33
	100	3.48	1.56	5.41	3.55
	500	17.40	7.78	35.29	15.23
MOAP10	5	3.54	1.90	2.07	1.71
	25	17.72	9.50	25.14	14.52
	50	35.45	19.00	51.63	21.22
	100	55.25	20.56	75.39	17.74
MOAP20	5	0.37	0.16	0.17	0.11
	25	1.87	0.81	2.48	1.45
	50	3.74	1.62	8.02	4.18
	100	7.48	3.23	20.48	8.09
	500	37.38	16.16	70.59	12.89
MOAP30	5	0.14	0.03	0.02	0.02
	25	0.69	0.16	0.47	0.34
	50	1.37	0.32	1.65	0.99
	100	2.75	0.65	5.17	2.91
	500	13.74	3.25	28.56	9.94

In Table 5.5, we report the absolute gap between the coverage gap of the subset generated by Algorithm 1 and the optimal coverage gap for some cardinality levels. We made this comparison only for MOKP25 since finding optimal subset requires excessive computation time when the number of nondominated points is high. Instances having less than 50 nondominated points are excluded from the calculation of the statistics in Table 5.5.

In Figure 5.1, we plot the coverage gaps of the subsets generated by Algorithm 1 at some number of representative points up to 500 for MOKP100. The values on the plot are the averages of ten instances for the given number of points. According to the improvement trend of the average coverage gaps of the subsets, the marginal improvement of the coverage gap gets very small after some number of representative points which is a small precentage of all nondominated points.

Table 5.5: Difference of coverage gap values generated by Algorithm 1 and minimal possible coverage gaps on MOKP25

$\|\mathbf{R}\|$	Avg.	Std. Dev.
5	0.131	0.074
10	0.079	0.025
15	0.044	0.015
20	0.023	0.007
25	0.022	0.011
30	0.018	0.004
35	0.014	0.010
40	0.008	0.009
45	0.008	0.005
50	0.004	0.005

If only a small subset of nondominated points are to be generated, then quality of the subset generated by Algorithm 1 is substantially worse than the optimal subset. However, as the cardinality of the subset increases, the difference of coverage gaps between the subsets reduces to very low values. Therefore, Algorithm 1 is suitable in cases where the DM desires middle or large size subsets. On the other hand, there exists room for improvement if only a small subset of nondominated points is to be generated.

We present the performance of our second algorithm, TDA, in Table 5.6 in comparison with Algorithm1. All the values in Table 5.6 are calculated by subtracting the performance measure value of TDA from the corresponding value of Algorithm 1. We present the differences in cardinality, number of models solved and CPU time under the column names $|\mathbf{R}|_{A1} - |\mathbf{R}|_{TDA}$, $M_{A1} - M_{TDA}$ and $CPU_{A1} - CPU_{TDA}$, respectively. Since the smaller is the better for all performance measures reported, positive values indicate superiority of TDA over Algorithm 1.

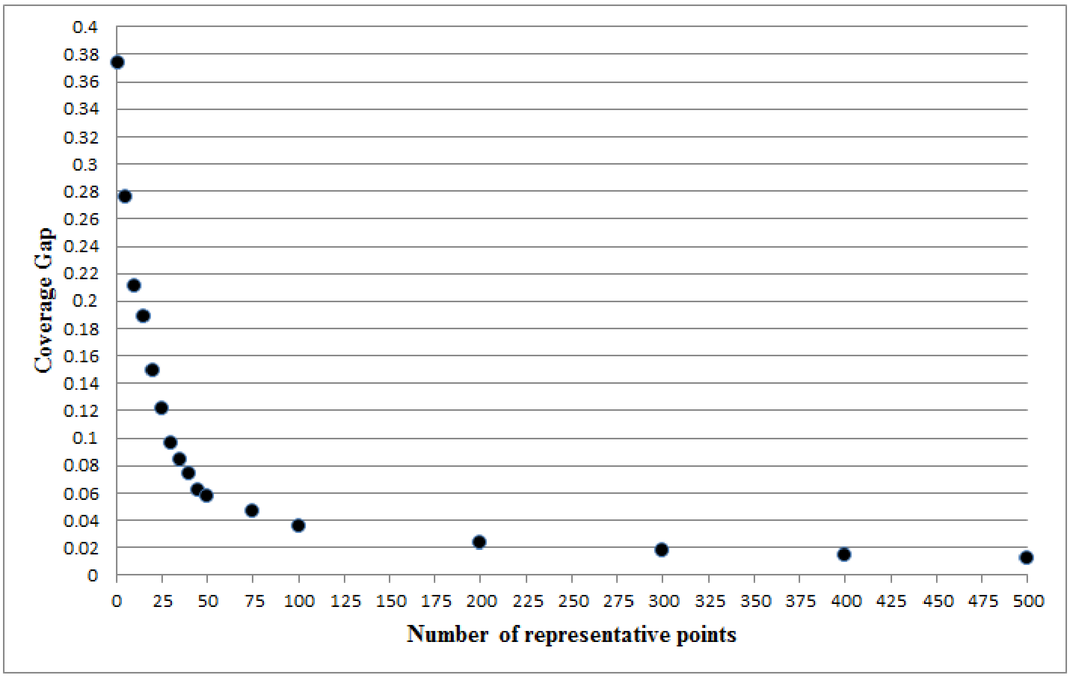

Figure 5.1: Average coverage gap values by Algorithm 1 for MOKP 100

We report the difference between number of nondominated points generated by both of the algorithms to satisfy a coverage gap threshold value. We report the performance measures for different threshold levels, Δ, which are instance specific. Actually, we set the coverage gap values returned by Algorithm 1 for given cardinality levels as the threshold values and run the TDA under these threshold values. (e.g. Δ_5 corresponds to the coverage gap when cardinality of the subset is 5.) The values of coverage gap thresholds are in the following order: $(\Delta_5 \geq \Delta_{25} \geq \Delta_{50} \geq \Delta_{100} \geq \Delta_{500})$.

In order to make a fair comparison, we also check whether Algorithm 1 satisfies the given threshold coverage gap with fewer number of points since coverage gap may remain the same for more than one iteration. If such a situation occurs, we take the minimum cardinality value while comparing it to the result of TDA.

If we examine the average cardinality values of the subsets returned to satisfy given threshold coverage gaps, there does not seem to exist a significant difference in the cardinality values except for the ones corresponding to the smallest thresholds of MOKP25, MOKP100 and MOAP30. In terms of the cardinality measure, both algorithms are competitive with each other. However, TDA outperforms Algorithm 1 in

Table 5.6: Differences in performace measure values produced by TDA and Algorithm 1

		$\lvert\mathbf{R}\rvert_{A1} - \lvert\mathbf{R}\rvert_{TDA}$		$M_{A1} - M_{TDA}$		$CPU_{A1} - CPU_{TDA}$	
Problem	Δ	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP25	Δ_5	-1.20	1.32	2.70	2.95	-0.07	0.46
	Δ_{25}	4.60	10.81	71.90	46.34	2.86	2.07
	Δ_{50}	20.80	25.26	120.50	106.73	4.16	3.74
MOKP50	Δ_5	-1.60	1.51	4.70	2.21	0.74	1.09
	Δ_{25}	1.90	4.72	159.20	40.37	9.72	2.69
	Δ_{50}	1.40	3.50	484.60	151.70	28.80	10.14
	Δ_{100}	5.00	6.00	1107.80	479.14	64.59	33.59
MOKP100	Δ_5	-2.50	0.97	3.50	2.37	0.35	0.60
	Δ_{25}	4.40	4.03	228.10	19.36	17.97	2.47
	Δ_{50}	-3.30	11.11	856.90	75.74	68.46	8.42
	Δ_{100}	-0.10	10.05	2585.40	229.21	209.11	29.19
	Δ_{500}	23.80	59.12	17043.60	3773.42	880.38	683.40
MOAP10	Δ_5	-1.20	1.23	4.00	2.75	0.43	0.48
	Δ_{25}	1.30	3.09	136.70	39.78	7.97	2.76
	Δ_{50}	3.00	3.27	285.40	111.31	16.16	8.29
	Δ_{100}	1.63	5.21	531.75	233.03	31.20	15.90
MOAP20	Δ_5	-2.20	1.40	3.50	4.97	0.70	0.54
	Δ_{25}	6.00	3.80	205.90	53.44	20.99	5.25
	Δ_{50}	4.90	5.88	622.20	135.82	65.10	15.36
	Δ_{100}	2.00	17.10	1625.40	402.23	173.58	46.60
	Δ_{500}	1.20	87.90	5638.80	2256.34	293.14	507.59
MOAP30	Δ_5	-2.00	1.41	4.70	3.62	0.86	0.56
	Δ_{25}	6.30	4.37	243.40	29.41	39.37	7.27
	Δ_{50}	6.80	5.96	853.10	176.00	144.69	35.88
	Δ_{100}	8.30	11.46	2760.40	638.65	486.41	150.27
	Δ_{500}	67.10	33.87	14699.30	4535.06	2623.85	1144.51

the number of models solved. This result is expected since a model is solved for each subspace in the current list in Algorithm 1 to find the next most diverse nondominated point. On the other hand, finding any nondominated point not in the territories of the previously generated points is sufficient in TDA. This results in solving fewer models in TDA compared to Algorithm 1. Similarly, TDA requires less solution time compared to Algorithm 1.

We also tested the performance of TDA to different initial nondominated points. For this purpose, we run the algorithm on MOKP50 with four different initial points. In

Table 5.7, the columns $Maxz_1$, $Maxz_2$ and $Maxz_3$ correspond to the cases where the algorithm starts with the point having the maximum value in the first, in the second and in the third objective, respectively. In the last column, $Maxz$, we use an equal weighted objective function to find the initial point. Based on these results, we cannot claim that different starting points affect the performance of the algorithm significantly.

Table 5.7: Number of nondominated points generated for different initial points in TDA

		$Maxz_1$		$Maxz_2$		$Maxz_3$		$Maxz$	
Problem	Δ	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP50	Δ_5	6.10	0.99	7.00	1.56	6.60	1.51	5.30	0.48
	Δ_{25}	22.30	4.24	23.60	4.38	22.60	4.55	20.90	2.64
	Δ_{50}	45.50	3.54	47.30	3.86	47.70	3.65	44.50	4.60
	Δ_{100}	90.10	5.11	90.70	4.67	90.40	5.87	90.40	5.40

As we mentioned at the end of Section 4.3, we tested the performance of TDA when there is no subspace selection rule in terms of the number of points generated. We give the results in Table 5.8. It seems that there is no significant improvement in terms of number of points but the solution time is reduced with the subspace selection rule given in Section 4.3.

Table 5.8: Effect of the subspace selection rule in TDA

Problem	Δ	Current Selection Rule				No Selection Rule			
		$\|\mathbf{R}\|$		CPU		$\|\mathbf{R}\|$		CPU	
		Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP50	Δ_5	6.60	1.51	0.66	0.32	4.50	0.71	0.51	0.19
	Δ_{25}	22.60	4.55	2.86	0.76	20.50	3.78	3.80	0.97
	Δ_{50}	47.70	3.65	7.11	1.08	47.60	7.99	13.63	2.34
	Δ_{100}	90.40	5.87	17.08	1.90	93.90	6.67	43.09	10.74

Lastly, we present the coverage gaps of the subsets generated by SPA and compare them to the ones produced by Algorithm 1. We separated the SPA algorithm into two stages: In the first stage, we fit a surface to approximate the nondominated frontier and find representative hypothetical point set. Afterwards, we use these hypothetical points and find representative nondominated points in Stage 2.

We report the coverage gaps of the subsets at the given cardinalities for both algorithms and give only the solution time of the SPA. Since we do not have the ideal and nadir points for MOAP30, we did not run the algorithm for that problem.

Table 5.9: Comparison of the subsets generated by SPA and Algorithm 1

Problem	$\|\mathbf{R}\|$	Algorithm 1		SPA					
		α		α		Stage 1 CPU		Stage 2 CPU	
		Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP25	5	0.22	0.04	0.18	0.05	179.50	79.91	0.17	0.06
	10	0.15	0.07	0.14	0.04	270.95	167.56	0.64	0.34
	25	0.06	0.02	0.11	0.04	81.05	68.45	2.05	1.18
MOKP50	5	0.24	0.02	0.20	0.04	211.72	89.45	0.29	0.13
	10	0.19	0.05	0.14	0.02	297.55	122.31	0.50	0.09
	25	0.09	0.02	0.11	0.02	95.19	43.37	1.41	0.26
	50	0.05	0.02	0.08	0.02	15.40	3.28	3.47	1.00
MOKP100	5	0.28	0.05	0.21	0.10	162.01	72.98	0.39	0.14
	10	0.21	0.04	0.16	0.13	643.82	260.18	0.85	0.21
	25	0.12	0.02	0.13	0.10	104.33	61.59	1.89	0.28
	50	0.06	0.01	0.11	0.10	17.20	4.60	3.54	0.34
MOAP10	5	0.28	0.06	0.21	0.04	183.06	118.62	0.23	0.07
	10	0.19	0.04	0.15	0.04	296.93	154.46	0.47	0.19
	25	0.08	0.02	0.11	0.03	102.15	72.65	1.69	0.69
	50	0.04	0.02	0.08	0.02	20.00	13.96	3.59	1.87
MOAP20	5	0.27	0.03	0.24	0.06	185.59	66.56	0.51	0.16
	10	0.21	0.03	0.17	0.04	364.98	205.13	1.04	0.21
	25	0.12	0.02	0.13	0.03	94.22	38.65	2.57	0.22
	50	0.07	0.01	0.11	0.04	22.82	15.82	5.59	1.06

According to the coverage gaps reported under the two algorithms in Table 5.9, SPA is able to generate higher quality subset if the cardinality of the subset is small. For 5 and 10 to 25 points, SPA finds subsets having lower coverage gap measure for all of the problems. However, the performance of SPA deteriorates as the cardinality of the subset increases. Although Algorithm 1 finds the most diverse point at each iteration and generate subsets having nonincreasing coverage gaps, the SPA cannot guarantee to find the most diverse point. If there exist a nondominated point which has a very large coverage gap compared to other nondominated points, then the DMA can find it after some iterations but the SPA might miss it at all. This is the main reason of better performance of the DMA for generating large sized subsets.

Performance of SPA is closely related with the structure of the nondominated frontier and the fitted L_p surface. It is known that some large size MOCO problems have thousands of nondominated points and the nondominated frontiers of them are very dense. If the nondominated frontier of the problem is dense, then it is highly probable to find very close nondominated points to the representative hypothetical points and to have higher quality subsets as a result.

The second determinant of the performance of SPA is the quality of the surface fitted. We used only one reference point to fit the L_p surface. It is possible to fit a better surface by using higher number of reference points. However, this number should not be too many since then it contradicts with the idea of generating a small subset of nondominated points. On the surface, we generate around 200 points and choose the optimal subset using only this many points. Although generating more points increases the quality of the discretization and quality of the optimal subset of hypothetical points, the number of points cannot be too many due to the incerased complexity of the model solved to obtain the hypothetical point subset. The solution time of that model determines the stage 1 CPU time of the algorithm which are reported in Table 5.9.

CHAPTER 6

CONCLUSIONS

We developed three methods to generate representative subsets of nondominated points for MOIP problems and tested the performance of them on random instances of MOKP and MOAP at different sizes. We used the coverage gap and cardinality measures to assess the quality of the subsets generated by the algorithms. The coverage gap of a representative subset shows us how much each remaining nondominated point could be better than its representative in any objective. Furthermore, if the coverage gap of a subset is sufficiently small, it implies that the subset almost dominates the nondominated frontier of the problem.

Our first algorithm, Algorithm 1, outperforms the DMA algorithm in terms of the solution time to generate a subset at the same quality. As the cardinality of the subset increases, it becomes prohibitive to solve DMA due to the number of additional binary variables and constraints while Algorithm 1 still works in a reasonable solution time. Furthermore, as the cardinality increases, Algorithm 1 performs well in terms of the coverage gap. While the coverage gap of a small representative subset generated by Algorithm1 is far from the best possible coverage gap value, the coverage gap approaches the optimal as the cardinality of the subset increases. Therefore, Algorithm 1 is well suited to the cases where a subset of moderate size is preferred.

In our second algorithm, we find a representative subset for a given coverage gap level such that we guarantee to achieve a coverage gap under the threshold value specified by the DM. Our results show that TDA satisfies the given coverage gap requirement with fewer points compared to Algorithm 1 especially for small threshold values. Furthermore, TDA solves 62% fewer models than Algorithm 1 on average.

Different than Algorithm 1 and TDA, our last algorithm, SPA, first considers the possible locations of the nondominated points. To do this, SPA first approximates the nondominated frontier by an Lp surface and then finds a diverse set of hypothetical points on this surface. Afterwards, SPA uses this set to produce a diverse set of true nondominated points. Our results show that the procedure works well for small subsets.Therefore, SPA may be practical in problems where solving the single objective problem is very difficult and only a small subset of nondominated points can be generated in a practical amount of time.

In order to find a diverse set of hypothetical points on the fitted surface, we discretize the fitted surface. As the number of discrete points increases, the quality of the discretization improves whereas the computational complexity increases considerably. As a future research, heuristic methods may be developed to find the representative hypothetical points on the surface. Since the problem solved is similar to $p-$center problem, the literature on this problem can be reviewed. If an efficient heuristic method can be developed, the L_p surface may be represented well using more discrete points and it may improve the performance of SPA.

An interesting research work could be studying the properties of optimal subsets in terms of the coevrage gap measure. It may be possible to obtain theoretical results related to the characterization of the most diverse points.

Finally, computational experiments can be expanded to more than three objectives and different MOCO problems.

REFERENCES

[1] M. E. Chergui, M. Moulaï, and F. Z. Ouaïl. Solving the multiple objective integer linear programming problem. In *MCO*, pages 69–76, 2008.

[2] C. Delort and O. Spanjaard. Using bound sets in multiobjective optimization: Application to the biobjective binary knapsack problem. In *SEA*, pages 253–265, 2010.

[3] C. Dhaenens, J. Lemesre, and E.G. Talbi. K-ppm: A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*, 200:45–53, 2010.

[4] M. Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147:343–360, 2006.

[5] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22:425–460, 2000.

[6] M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization. State of the art annotated bibliographic surveys*. Kluwer Academic, Dordrecht, 2002.

[7] M. Ehrgott and X. Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 12(1):1–63, June 2004.

[8] M. Ehrgott and X. Gandibleux. Hybrid metaheuristics for multi-objective combinatorial optimization. *Studies in Computational Intelligence*, 114:221–259, 2008.

[9] S.L. Faulkenberg and M.M. Wiecek. On the quality of discrete representations in multiple objectve programming. *Optimization and Engineering*, 11(3):423–440, 2010.

[10] İ. Karahan and M. Köksalan. A territory defining multiobjective evolutionary algorithms and preference incorporation. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 14(4):636–664, 2010.

[11] E. Karasakal and M. Köksalan. Generating a representative subset of the nondominated frontier in multiple criteria decision making. *Operations Research*, 57(1):187–199, 2009.

[12] G. Kirlik and S. Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232:479–488, 2014.

[13] M. Köksalan. A heuristic approach to bicriteria scheduling. *Naval Research Logistics*, 46(7):777–789, 1999.

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

[15] M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169:932–942, 2006.

[16] J. Lemesre, C. Dhaenens, and E.G. Talbi. Parallel partitioning method (ppm): A new exact method to solve bi-objective problems. *Computers & Operations Research*, 34:2450–2462, 2007.

[17] B. Lokman. *Converging Preferred Regions in Multi-objective Combinatorial Optimization Problems*. PhD thesis, Middle East Technical University, 2011.

[18] B. Lokman and M. Köksalan. Finding highly preferred points for multi-objective integer programs. in press.

[19] B. Lokman and M. Köksalan. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57:347–365, 2013.

[20] M.Masin and Y. Bukchin. Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2):411–4242, 2008.

[21] A. Ponte, L. Paquete, and J. R. Figueira. On beam search for multicriteria combinatorial optimization problems. In Nicolas Beldiceanu, Narendra Jussien, and Eric Pinson, editors, *CPAIOR*, volume 7298 of *Lecture Notes in Computer Science*, pages 307–321. Springer, 2012.

[22] A. Przybylski, X. Gandibleux, and M. Ehrgott. Two phase algorithms for the bi-objective assignment problem. *European Journal of Operational Research*, 185:509–533, 2008.

[23] A. Przybylski, X. Gandibleux, and M. Ehrgott. A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. *Informs Journal on Computing*, 22(3):371–386, 2010.

[24] A. Przybylski, X. Gandibleux, and M. Ehrgott. A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7:149–165, 2010.

[25] S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87:543–560, 2000.

[26] S. Sayın. A procedure to find discrete representations of the efficient set with specified coverage errors. *Operations Research*, 51(3):427–436, 2003.

[27] F. Sourd and O. Spanjaard. A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3):472–484, 2008.

[28] T. Stidsen, K.A. Andersen, and B. Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032, 2014.

[29] J. Sylva and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158:46–55, 2004.

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

[31] D. Tenfelde-Podehl. A recursive algorithm for multiobjective combinatorial optimization problems with q criteria. Technical report, Institut für Mathematik, Technische Universitat Graz, 2003.

[32] A. P. Wierzbicki. On the completeness and constuctiveness of parametric characterizations to vector optimization problems. *OR Spectrum*, 8(2):73–87, 1986.

[33] E. Zitzler, J. D. Knowles, and L. Thiele. Quality assessment of pareto set approximations. In *Multiobjective Optimization*, pages 373–404, 2008.

[34] M. Özlen and M. Azizoğlu. Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 25-35:347–365, 2009.

[35] M. Özlen, B.A. Burton, and C.A.G. MacRae. Multi-objective integer programming: An improved recursive algorithm. *Journal of Optimization Theory Applications*, 160(2):470–482, 2014.

[36] Ö. Özpeynirci and M. Köksalan. An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315, 2010.