REPRESENTING THE NONDOMINATED SET WITH A SMALL SUBSET IN MULTI-OBJECTIVE MIXED INTEGER PROGRAMS

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

BY

ILGIN DOĞAN

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

JUNE 2018

Approval of the thesis:

REPRESENTING THE NONDOMINATED SET WITH A SMALL SUBSET IN MULTI-OBJECTIVE MIXED INTEGER PROGRAMS

submitted by **ILGIN DOĞAN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Yasemin Serin
Head of Department, **Industrial Engineering** ________________

Assist. Prof. Dr. Banu Lokman
Supervisor, **Industrial Engineering Dept., METU** ________________

Prof. Dr. M. Murat Köksalan
Co-supervisor, **McDonough Bus. Sch., Georgetown Univ., US** ________________

Examining Committee Members:

Assoc. Prof. Dr. Canan Ulu
McDonough Bus. Sch., Georgetown Univ., US ________________

Assist. Prof. Dr. Banu Lokman
Industrial Engineering Dept., METU ________________

Prof. Dr. Murat Köksalan
McDonough Bus. Sch., Georgetown Univ., US ________________

Assist. Prof. Dr. Özgen Karaer
Industrial Engineering Dept., METU ________________

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

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: ILGIN DOĞAN

Signature :

ABSTRACT

REPRESENTING THE NONDOMINATED SET WITH A SMALL SUBSET IN MULTI-OBJECTIVE MIXED INTEGER PROGRAMS

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

June 2018, 86 pages

Multi-Objective Mixed Integer Programs (MOMIPs) have a wide variety of application areas in real-life decision making problems. Since the number of nondominated points grows exponentially with the problem size and finding all nondominated points is typically hard and impractical in MOMIPs, generating a subset having "desired properties" rises as an important problem. Motivated with this fact, we observe that the distribution of nondominated points may be critical in defining the desired properties of the representative subset to be generated. Based on our observations, we develop algorithms to generate a small subset of nondominated points that represents the nondominated set with a prespecified coverage gap. Our computational experiments show that our algorithms outperform the existing algorithms in terms of the cardinality of the generated representative set and the solution time.

Keywords: Multi-Objective Mixed Integer Programming, Representative Subset, Nondominated Point, Supported Point, Coverage Gap

ÖZ

ÇOK AMAÇLI KARIŞIK TAMSAYI PROBLEMLERİNDE BASKIN KÜMENİN KÜÇÜK BİR ALTKÜME İLE TEMSİL EDİLMESİ

Doğan, Ilgın
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üyesi Banu Lokman
Ortak Tez Yöneticisi : Prof. Dr. M. Murat Köksalan

Haziran 2018 , 86 sayfa

Birçok karar verme probleminde geniş bir uygulama alanı bulunan Çok Amaçlı Karışık Tamsayı Problemlerinde (ÇAKTP) problem boyutu arttıkça baskın nokta sayısı da üssel olarak artmaktadır. Bu sebeple bütün baskın noktaları üretmek genellikle zor ve kullanışsız olup istenilen özelliklere sahip küçük bir altküme ile bütün noktaları temsil etmek amaçlanmaktadır. Bu motivasyon ile üretilecek altküme için istenilen özelliklerin tanımlanmasında baskın noktaların dağılımının belirleyici olabileceğini gözlemledik. Gözlemlerimize bağlı olarak geliştirdiğimiz algoritmalar verilen bir temsil hatası ile tüm baskın noktaları temsil edecek küçük bir alt küme üretmektedir. Deneylerimiz, üretilen alt kümenin kardinalitesi ve çözüm süreleri açısından geliştirdiğimiz algoritmaların mevcut algoritmalardan daha iyi çalıştığını göstermiştir.

Anahtar Kelimeler: Çok Amaçlı Karışık Tamsayı Programlama, Temsili Küme, Baskın Nokta, Destekli Nokta, Kapsama Hatası

To my mother,

Hülya Ülger

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisors Banu Lokman and Murat Köksalan for their constant support and guidance throughout this study. It was a great honor to work with both of them. Our cooperation influenced my academical and world view highly. They not only supported and motivated me on my research but also provided that I feel welcome and personally attended all my needs and problems.

This work is also supported by TÜBİTAK-ARDEB Scientific and Technological Research Project Program (1001) (Project No: 215M844).

A lot of people influenced and supported me during my six years in the Department of Industrial Engineering, METU. I am so thankful to all my professors. I would also thank Gökhan Ceyhan who supplied a lot of important material for the progress of my thesis studies. Moreover, I would like to thank the members of my thesis committee Assoc. Prof. Dr. Canan Ulu, Assist. Prof. Dr. Özgen Karaer and Assist. Prof. Dr. Sakine Batun for their valuable feedbacks.

My deepest thanks to Semih Barutçu who was always a ray of sunshine for me.

Last but not least, my family always provided invaluable support for me. I would like to thank specially my mother Hülya Ülger for all her love and encouragement throughout my life. She is the person who deserves endless thanks. I dedicate this thesis to my mother. All my accomplishments would be impossible without her.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DM	Decision Maker
MOIP	Multi-Objective Integer Program
MOMIP	Multi-Objective Mixed-Integer Program
MOLP	Multi-Objective Linear Program
MCDM	Multi-Criteria Decision Making
MOCO	Multi-Objective Combinatorial Optimization
MOKP	Multi-Objective Knapsack Problem
MOAP	Multi-Objective Assignment Problem

CHAPTER 1

INTRODUCTION

In almost every real life problem, decision makers (DMs) encounter multiple objectives that are usually conflicting with each other. Due to this conflict, these problems do not have a single optimal solution but have a set of preferable solutions. The main characteristic of these solutions is that in order to improve one objective, the DMs must sacrifice from at least one of the other objectives. Then, the most preferred solution is chosen from this set of solutions by the DMs.

In this thesis, we study Multi-Objective Mixed Integer Programs (MOMIPs). In the MOMIP literature, these preferable solutions in the decision space are called as the "efficient solutions" while their images on the objective space are called as the "nondominated points". The corresponding sets of these solutions are called as the "efficient frontier" and the "nondominated frontier", respectively. Multi-Objective Integer Programs (MOIPs) are a special case of MOMIPs where all variables are integers and there is a finite number of nondominated points.

In large-sized practical problems, as the number of objectives increases, the number of nondominated points increases exponentially. That is why, generating all nondominated points is typically hard. Furthermore, trying to find the most preferred solution among such a huge set is not practical for the DM. Therefore, to avoid unnecessary computational effort, finding a subset of these points rises as an important research area in MOMIPs. In the literature, there are some approaches developed to generate subsets of nondominated points considering certain quality measures specified by the DMs. Such a subset can also be called as a representative set since it is used to represent the whole nondominated frontier in terms of some desired quality measures.

In this study, we develop algorithms to generate representative sets with desired properties. The desired properties could naturally differ from application to application. Özarık (2017) observes that the distribution of nondominated points may be critical in defining the desired properties of the representative set to be generated. Once the distribution of nondominated points is known, one may want to generate more points from the densely populated regions. Alternatively, one may wish to positively discriminate less dense regions in order to capture the properties of rare solutions in addition to typical solutions. Considering all these possible implementations, our algorithms are based on the typical properties of the distributions of nondominated points in MOMIPs.

In MOMIPs, the points that can be found by solving a single-objective problem, that is a weighted sum problem, are called as the "supported nondominated points". All other nondominated points that cannot be generated by solving a weighted sum problem are called as the "unsupported nondominated points". In our algorithms, we iteratively reduce the feasible set by excluding the regions that are dominated by the previously found nondominated points. Then, we solve a weighted sum problem over the reduced feasible set. While it generates a supported nondominated point defined on the reduced region, the generated point may be an unsupported nondominated point with respect to the original problem.

In the literature, there are some algorithms developed to generate representative sets of nondominated points in MOMIPs. Some quality measures are defined to assess the performance of these representative sets. Sayın (2000) suggests that a representative set should cover each and every nondominated point of a MOMIP and how well each point is covered can be measured by the *coverage gap*. There are some approaches proposed to generate representative sets for a given coverage gap value. There is an exact algorithm called as the Diversity Maximization Algorithm (DMA) proposed by Masin and Bukchin (2008) and a similar approach developed by Sylva and Crema (2007). These approaches try to find the most diverse set of nondominated points for a given coverage gap value. Throughout the algorithm, as the number of generated points increases, the number of binary variables and constraints added to their models increase substantially. Therefore, the computational complexities of these algorithms may be undesirable, especially in large-sized real-life problems.

As an improvement of these algorithms, Ceyhan et. al. (2014) propose two algorithms called as the Subspace Based Approach (SBA) and the Territory Defining Algorithm (TDA). These are also based on the same idea of iteratively generating the worst represented nondominated point in terms of the coverage gap measure. SBA and TDA present improvement in terms of the computational efficiency since they use the search method proposed by Lokman and Köksalan (2013) that is based on the enumeration of the nondominated subspaces. Due to this decomposition method, their solution times are much less than the solution times of Sylva and Crema (2007) and Masin and Bukchin (2008).

In this study, we propose two new algorithms that provide improvements to these existing approaches in terms of both the solution quality and the computational efficiency. Our purpose is to use the common properties of the density distributions of the nondominated points in MOMIPs. We try to generate nondominated points from the dense regions of the nondominated frontier in order to represent more points by less number of representatives. To achieve this, while developing our algorithms, we observed that the nondominated points which could represent more number of points may have better weighted sum values due to the shape of the frontier.

In our first algorithm (called as the Territory-Excluded Supported Generating Algorithm, TSGA), given a specific coverage gap value, we define some regions (called as the *territories*) around each generated nondominated point and after excluding these regions from the feasible space, we search for a new nondominated point by solving a weighted sum problem over the reduced objective space. TSGA has a better performance than the existing approaches in terms of both the solution quality and the solution time. In our second algorithm (called as TSGA-II), we yield better representative sets in terms of the cardinality but the solutions times are longer than TSGA. In this algorithm, we again work on the reduced objective space. We randomly select an objective and iteratively find the nondominated point that has the best value in this objective. Then, we generate the representative point that will cover this point by solving a weighted sum problem (excluding the selected objective) in each iteration. The major characteristic of this algorithm is that we always guarantee to cover all nondominated points in terms of the selected objective and for a problem with m objectives, we search for the nondominated points in the $(m-1)$-dimensional space.

Our computational experiments are performed on the randomly generated test instances of the Multi-Objective Knapsack Problem (MOKP), Multi-Objective Assignment Problem (MOAP) and mixed-integer knapsack problem. In order to assess the performance of our algorithms, we generate all nondominated points of these problems. Then, we compare the quality of our representative sets with the quality of the representative sets generated by Masin and Bukchin (2008) & Sylva and Crema (2007) and Ceyhan et. al. (2014). We implement all algorithms by using the recently proposed decomposition method by Dächert et. al. (2017). Our results show that our algorithms outperform the existing ones in terms of both the cardinality of the generated representative sets and the solution times. In addition, we solve for the representative sets with the optimal cardinality for a given coverage gap value. Results indicate that our algorithms converge to the optimal cardinality better than the existing approaches.

The organization of this thesis is as follows. In Chapter 2, we present the relevant preliminaries including the quality measures and the description of the scaling method used in our algorithms. In Chapter 3, we briefly review some recent approaches developed to generate representative nondominated sets and some approximation algorithms for MOMIPs. In Chapter 4, we first present the existing approaches that we compare the performance of our algorithms against. Then, we describe our algorithms TSGA and TSGA-II with their interactive applications. Finally, we report the results of our computational experiments in Chapter 5 and we summarize our conclusions in Chapter 6.

CHAPTER 2

PRELIMINARIES

In this chapter, we provide the relevant background in Multi-Criteria Decision Making (MCDM) and definitions of the quality measures proposed by Sayın (2000) for representative sets of nondominated points in MOMIPs.

2.1 Background - Definitions

As a mathematical model, a MOMIP can be defined as follows:

$$\text{(MOMIP)} \qquad \text{“}Max\text{”} \;\; \boldsymbol{z} = \boldsymbol{f(x)}, \quad \text{subject to} \quad \boldsymbol{x} \in \boldsymbol{X},$$

where $f(x) = \{z_1(x), z_2(x), ..., z_m(x)\}$ is m-dimensional point in the objective space, $\boldsymbol{x}$ is a vector in decision space. $\boldsymbol{X} \subseteq \mathbb{Z}^n$ is the feasible decision space and $\boldsymbol{Z}$ is the feasible objective space.

Definition 2.1. A feasible decision point $\boldsymbol{x}^i \in \boldsymbol{X}$ is an *efficient solution* if there does not exist any $\boldsymbol{x}^j \in \boldsymbol{X}$ such that

$$z_k(\boldsymbol{x}^j) \geq z_k(\boldsymbol{x}^i) \;\; \forall k \in \{1, 2, ..., m\} \text{ and } z_k(\boldsymbol{x}^j) > z_k(\boldsymbol{x}^i) \; for \; at \; least \; one \; k.$$

If $\boldsymbol{x}^i$ is an efficient solution, then its image in the objective space, $\boldsymbol{f}(\boldsymbol{x}^i)$, is said to be a *nondominated point*.

For a MOMIP, we denote the set of efficient solutions as $\boldsymbol{X}_E$ and the set of nondominated points as $\boldsymbol{Z}_{ND}$.

Definition 2.2. The *ideal point* of a MOMIP, $\boldsymbol{z}^{IP} = \left\{z_1^{IP}, z_2^{IP}, ..., z_m^{IP}\right\}$, is defined as a m-dimensional vector whose components are the best possible values of each objective. More specifically, it can be defined as follows for different types of MOMIPs:

- For a maximization problem:

$$z_k^{IP} = \max_{\boldsymbol{x} \in \boldsymbol{X}} (z_k(\boldsymbol{x})) \ \ \forall k$$

- For a minimization problem:

$$z_k^{IP} = \min_{\boldsymbol{x} \in \boldsymbol{X}} (z_k(\boldsymbol{x})) \ \ \forall k$$

Definition 2.3. The *nadir point* of a MOMIP, $\boldsymbol{z}^{NP} = \left\{z_1^{NP}, z_2^{NP}, ..., z_m^{NP}\right\}$, is defined as a m-dimensional vector consists of the worst objective values in the set of efficient solutions, $\boldsymbol{X}_E$. Specifically,

- For a maximization problem:

$$z_k^{NP} = \min_{\boldsymbol{x} \in \boldsymbol{X}_E} (z_k(\boldsymbol{x})) \ \ \forall k$$

- For a minimization problem:

$$z_k^{NP} = \max_{\boldsymbol{x} \in \boldsymbol{X}_E} (z_k(\boldsymbol{x})) \ \ \forall k$$

2.2 Quality Measures

In real-life multi-objective decision making problems, presenting all nondominated points to the DM gets harder and time consuming as the problem size increases. Instead, working with a subset of nondominated points is more practical and easier. While generating such a subset of nondominated points, our purpose is to represent all nondominated points according to quality levels desired by the DM. In order to specify the quality of a representative subset, Sayın (2000) defines three performance measures which are coverage gap, cardinality and uniformity measures.

2.2.1 Coverage Gap Measure, α

Sayın (2000) suggests that a representative set should cover every portion of the efficient frontier. In order to measure this coverage attribute, Sayın (2000) defines the following coverage measure:

Definition 2.4. Let $\boldsymbol{R} \subset \mathbf{Z}_{ND}$ be a representative set of a maximization MOMIP. The *coverage gap* for $\boldsymbol{R}$ can be calculated as follows:

$$\alpha = \max_{\boldsymbol{z} \in \mathbf{Z}_{ND}} \min_{\boldsymbol{y} \in \boldsymbol{R}} d(\boldsymbol{z}, \boldsymbol{y})$$

where $d(\boldsymbol{z}, \boldsymbol{y})$ is a distance metric.

To calculate the coverage gap of a representative set, Sayın (2000) uses a Tchebycheff distance metric, i.e. $\max_{k=1,2,\ldots,m} |z_k - y_k|$. Each nondominated point ($\boldsymbol{z}$) is assigned to its closest representative point $\boldsymbol{y}$ in terms of this metric which is defined as the maximum of the differences in objective values of these two points. The distance between a nondominated point and its closest representative point is the coverage gap of this specific nondominated point. Then, among all nondominated points, the one with the maximum coverage gap determines the coverage gap of the representative set, $\alpha_{\boldsymbol{R}}$. This point is called as the the worst represented nondominated point.

Masin and Bukchin (2008) suggests a different distance metric considering only the objectives in which the nondominated point ($\boldsymbol{z}$) is better than the representative point ($\boldsymbol{y}$). In this case, coverage gap of the representative set can be calculated as follows:

$$\alpha = \max_{\boldsymbol{z} \in \mathbf{Z}_{ND}} \min_{\boldsymbol{y} \in \boldsymbol{R}} \max_{1 \leq k \leq m} (z_k - y_k)$$

In our studies, we use the distance metric defined by Masin and Bukchin (2008). If the coverage gap of a representative set $\boldsymbol{R}$ is $\alpha_{\boldsymbol{R}}$, then there exists a nondominated point which is α better than its representative in at least one objective. Also, by definition, all nondominated points are at most α better than their representatives in at least one objective. Then, a nondominated point is said to be $\alpha - dominated$ by its representatives.

Definition 2.5. Let $\boldsymbol{R}$ be a representative set of a maximization MOMIP. Let $\boldsymbol{y} \in \boldsymbol{R}$ represent $\boldsymbol{z} \in \mathbf{Z}_{ND}$ such that $z_k \leq y_k + \alpha \; for \; all \; k = 1, 2, ..., m$. Then, $\boldsymbol{z}$ is said to be $\alpha - dominated$ by $\boldsymbol{y}$.

In our computational experiments, we used a scaling coefficient so that the coverage gap value specified by DM is defined on the interval [0, 1]. The scaling coefficient for objective k is $\frac{1}{R_k}$ where R_k is the range of objective k on the efficient frontier for a maximization problem, i.e.

$$R_k = \max_{\boldsymbol{x} \in \boldsymbol{X}_E}(z_k(\boldsymbol{x})) - \min_{\boldsymbol{x} \in \boldsymbol{X}_E}(z_k(\boldsymbol{x}))$$

2.2.2 Uniformity Measure, δ

The points in a representative set are desired to be uniformly distributed over the nondominated frontier in order to present to the DM as much information as possible. In the ideal case of uniformity, the representatives should be located as equally-spaced from each other with respect to a given distance measure. In terms of uniformity, it is undesired to generate representative points which are mostly located as clusters in specific regions of the nondominated frontier. Sayın (2000) defines the uniformity measure as follows:

Definition 2.6. Let $\boldsymbol{R}$ be a representative set of a maximization MOMIP. Then, $\boldsymbol{R}$ is said to be δ-*uniform* if

$$\min_{\boldsymbol{y}^1, \boldsymbol{y}^2 \in \boldsymbol{R}} d(\boldsymbol{y}^1, \boldsymbol{y}^2) \geq \delta.$$

2.2.3 Cardinality Measure

Another quality measure for a representative set is cardinality defined by Sayın (2000). The main purpose of generating a representative set of nondominated points is to avoid the inefficiency of generating all points. Therefore, while satisfying the quality level desired by the DM, our aim is to achieve this level with the minimum number of points to minimize the computational effort.

These three quality measures are closely interrelated with each other. For example, we expect coverage gap to improve as the cardinality and uniformity increases. However, especially in large-sized real life problems, our main concern is to decrease the computational effort which is directly related with the cardinality.

Example for Quality Measures

Consider a bi-objective problem with 8 nondominated points which are shown on the scaled objective space in Figure 2.1. Suppose the generated representative set is $\boldsymbol{R} = \{(0.1, 0.9), (0.6, 0.6), (0.9, 0.3)\}$. Accordingly, the values of three quality measures are given in Figure 2.1. The worst represented point is also shown since it is represented with the maximum coverage gap among all nondominated points and it determines the coverage gap of $\boldsymbol{R}$.

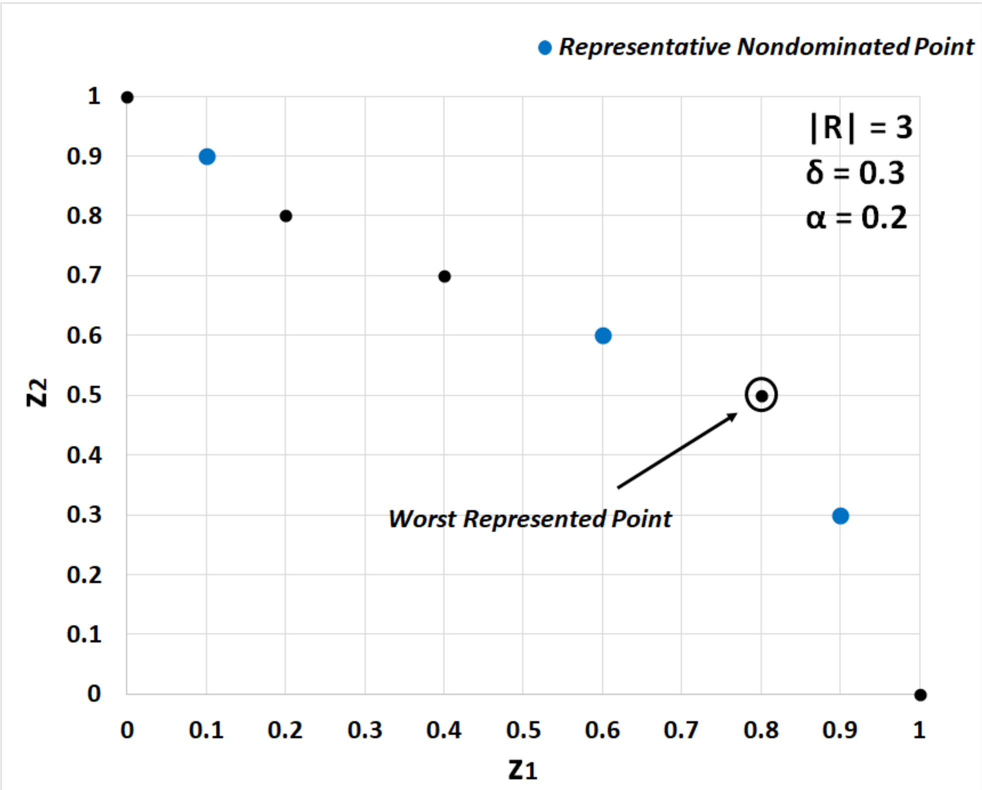


Figure 2.1: Example of quality measures for a representative set

CHAPTER 3

LITERATURE REVIEW

In this chapter, we review some exact and heuristic approaches in the MOMIP literature. There are so many approaches which have been proposed in order to generate either all nondominated points in MOIPs or good representative subsets of nondominated points satisfying certain quality measures. Instead of the problem-specific ones, we focus on general methods which can be applied to any problem type.

Finding all nondominated points of MOIPs requires high computational efforts especially in large-sized practical problems. Ehrgott and Gandibleux (2000) discuss that as the problem size grows, the number of supported nondominated points increases exponentially whereas the number of unsupported nondominated points increases linearly. Therefore, while generating all nondominated points, the difficulty of generating the supported ones increases the solution times mostly. As the number of nondominated points increases, finding all of them becomes computationally hard. As a result, several approaches have been developed in the literature to generate a subset that represents all nondominated points for some desired quality measures.

Sylva and Crema (2007) propose an exact algorithm that can be used to generate all nondominated points or a subset satisfying a given coverage gap value. They iteratively solve a single model by adding m binary variables and $(m + 1)$ linear constraints for an m-objective problem. Masin and Bukchin (2008) also develop a very similar approach. They define a diversity measure that is maximized for minimization problems whereas Sylva and Crema (2007) use a measure that is minimized for maximization problems. Although they use different mathematical formulations, the computational complexities of both approaches increase significantly as the problem size grows.

Özlen and Azizoglu (2009) present a recursive algorithm which can be used to generate either the whole set or a subset of the nondominated points. It is developed as an improvement to the classical constraint method. They increase efficiency by identifying efficiency ranges for objectives. Özlen et. al. (2014) further improves the efficiency of this algorithm by avoiding to solve previously solved submodels throughout the solution process.

Lokman and Köksalan (2013) develop another exact algorithm for generating all nondominated points of MOIPs. In each iteration, after generating a nondominated point, they reduce the search space by eliminating the regions dominated by the generated point. Then, they decompose the reduced search space into subspaces by an enumeration technique. This procedure outperforms Sylva and Crema (2007) and Özlen and Azizoglu (2009) in terms of the computational complexity.

Kirlik and Sayın (2014) also propose a method to generate all nondominated points. For an m-objective problem, they project each generated nondominated point to an $(m-1)$-dimensional objective space. Then, they create $(m-1)$-dimensional regions around the generated points and search for the next nondominated point within these regions. This approach again outperforms Sylva and Crema (2007) and Özlen and Azizoglu (2009) with respect to the solution times.

Boland et. al. (2015) present an objective space search method, called as the Balanced Box method, which is designed to find all nondominated points of bi-objective integer programs. This method enhances the efficiency significantly and shows a fast approximation of the efficient frontier. They report that the performance of this method is comparable with the ε-constraint method. Similarly, Boland et. al. (2016) suggest the L-Shape Search method (LSM) which aims to find all nondominated points of a tri-objective integer program. They define rectangles around each generated point and sort them in the non-increasing order of their areas. In each iteration, by inducing an L-shape within each rectangle, they search for a new nondominated point whose projection is in the rectangle. They state that the efficiency of LSM comes from its reliance on solving single-objective integer programs (IPs). They also try to avoid solving unnecessary IPs and increase the overall efficiency of LSM. They show that LSM is 17% faster on the average than Kirlik and Sayin (2014).

In large-sized practical problems, instead of generating all nondominated points, presenting a small subset to the DM is more useful and efficient. There are some quality measures defined in the literature in order to assess the performance of the generated small subset of nondominated points. Sayın (2000) suggests three main measures which are the coverage gap, uniformity and cardinality measures. The generated representative subset should cover each and every nondominated point and should be well dispersed over the nondominated frontier. For a given coverage or uniformity level, the cardinality of the representative set should be minimized in order to avoid unnecessary computational efforts.

Ceyhan et. al. (2014) develop three algorithms to produce representative sets of nondominated points for MOMIPs. Firstly, they developed an improved approach of Sylva and Crema (2007) and Masin and Bukchin (2008). Their first approach (the Subspace-Based Approach) generates a representative set for a given coverage gap value or for a given cardinality level. They use the decomposition method of Lokman and Köksalan (2013) which eliminate the additional binary variables and linear constraints in Sylva and Crema (2007). By this way, SBA improves the computational efficiency. Secondly, they develop the Territory Defining Algorithm (TDA) which generates a representative set for a given coverage gap value by the DM. TDA reduces the search space by excluding the territories contructed for each generated point from the search space. Although the solution quality may be worse than SBA, TDA requires less solution times.

Vaz et. al. (2015) propose several algorithms to find representative sets for bi-objective discrete optimization problems. They consider uniformity, cardinality and ε-indicator (similar to the coverage gap) measures. They develop several algorithms which are either based on solving several subproblems or solving a sequence of feasibility problems. They formulate these problems as special types of one-dimensional facility location problems. For instance, the representation problem which tries to maximize the uniformity for a given cardinality level, is defined as a k-dispersion facility-location problem. They also develop algorithms which try to optimize two performance measures simultaneously for a given number of representatives. They conclude that since the related k-dispersion and k-center problems are generally NP-hard, they say that their methods are not applicable for more than two objectives.

Vassilvitskii and Yannakakis (2005) introduce the problem of finding a lower bound on the cardinality of the representative set which will satisfy a certain coverage level. Then, Bazgan et. al. (2015) work on this problem and for the bi-objective problems, they guarantee that their approximation method computes at most 3 times the cardinality of the optimal representative set, which is called as a 3-approximation algorithm. For the tri-objective optimization problems, they propose a greedy approach under the assumption that all nondominated points are known in advance.

Filippi and Stevanato (2013) develop two approximation algorithms for bi-objective combinatorial optimization problems. Their algorithms find a representative subset such that each nondominated point is within a specific factor from a representative point in terms of both objectives. They also show that the cardinality of their representative sets is at most three times worse than the optimal cardinality. The first algorithm is called as the ABE algorithm which iteratively finds a new nondominated point and partitions the objective space into four subspaces. The second algorithm is called as AEC method which is stated as a modification of the well-known ε-constraint method. They conduct some experiments on the Travelling Salesman Problem with profits where maximizing profit and minimizing cost are the two conflicting objectives. Their results show that their algorithms yield a guaranteed approximation to the results of exact methods in an efficient way.

Shao and Ehrgott (2016) propose a method to generate representative sets for the continuous but non-convex nondominated sets while guaranteeing a specific coverage gap and uniformity level. Specifically, they try to solve the NP-hard problem of finding a number of evenly-distributed nondominated points of a MOLP (Multi-Objective Linear Program) for a given coverage gap value. Their method is stated as a combination of the global shooting and the normal boundary intersection (NBI) methods. By combining these methods, they take the advantage of the global shooting method in satisfying the coverage property and the advantage of NBI in generating evenly-distributed nondominated points (which satisfies the uniformity property). They call their new approach as "Revised Normal Boundary Intersection" (RNBI) method. They show that the generated representative points are indeed evenly distributed and the RNBI method is applicable for MOLPs with up to 8 objectives.

Köksalan and Lokman (2009) approximate the nondominated frontiers in MOCO problems. After scaling all objectives, they try to fit a smooth hypersurface, called as an L_q surface, to the nondominated frontier which passes through all the hypothetical extreme nondominated points and a centrally located nondominated point. Based on their observations, they make analyses related with the typical characteristics of the shapes of nondominated frontiers in MOCO problems.

Özarık (2017) develops algorithms to generate representative sets for different types of MOIPs by defining a new quality measure. They introduce a density measure and make analyses for identifying the typical distribution properties of nondominated points over the frontier. Their approaches first approximate the nondominated frontier by using the method of Köksalan and Lokman (2009). Then, they categorize this approximated nondominated set based on the estimated density measures in each subregion. By this way, they generate density-based representative sets for MOIPs.

Dächert et. al. (2017) develop one of the most recent decomposition methods in the MOMIP literature. They decompose the whole feasible objective space into subspaces which are defined by lower bound vectors. Then, they define a specific neighborhood relation between these lower bounds and update the search space in each iteration by updating these neighborhood relations. They show that due to the efficiency of their update procedure, they outperforms all existing search methods in the literature (such as Lokman and Köksalan (2013), Kirlik and Sayın (2014)) in terms of computational efficiency.

CHAPTER 4

ALGORITHMS FOR GENERATING REPRESENTATIVE SETS OF NONDOMINATED POINTS

In this chapter, we present two approaches for representation of nondominated sets in MOMIPs in terms of the quality measures defined in Chapter 2. Our main concern is to cover all parts of the nondominated frontier by generating a small representative subset of the nondominated points. Therefore, our main quality measures are the coverage gap and the cardinality. Specifically, we aim to find a small subset that represents the whole nondominated set with a prespecified coverage gap value. The uniformity measure is also controlled implicitly since the algorithms are designed to produce representative points within at least a certain distance from each other.

Before presenting our approaches, we briefly describe the existing algorithms that we compare our experimental results against. These algoritms are the Diversity Maximization Algorithm (DMA) proposed by Masin and Bukchin (2008), the Subspace Based Approach (SBA) and the Territory Defining Algorithm (TDA) developed by Ceyhan et. al. (2014). Given a coverage gap value, our approaches are compared with these algorithms in terms of the cardinality, solution times and number of models solved. All of these approaches can be used either to generate a representative subset or the entire nondominated set.

4.1 Existing Approaches

4.1.1 Diversity Maximization Algorithm (DMA)

DMA is an exact algorithm developed by Masin and Bukchin (2008). It is very similar to the approach proposed by Sylva and Crema (2007). These approaches iteratively generate the worst represented nondominated point by the previously generated representatives. Starting with an initial supported nondominated point, they solve a model in each iteration which generates the nondominated point with the maximum coverage gap value (which is the worst represented nondominated point by the current representatives). The model (P_0) shows the mathematical formulation proposed by Sylva and Crema (2007). Let $\boldsymbol{R} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{|\boldsymbol{R}|}\}$ be a representative set of a maximization MOMIP with m objectives. Then,

(P_0):

$$Max \quad \alpha \; + \; \epsilon \sum_{k=1}^{m} \lambda_k z_k$$

$$s.to.$$

$$z_k(\boldsymbol{x}) \geq y_k^i \, p_k^i + \alpha - (M_k + U)(1 - p_k^i) \quad \forall\, i = 1, 2, ..., |\boldsymbol{R}| \quad \forall\, k = 1, 2, ..., m$$

$$\sum_{k=1}^{m} p_k^i = 1 \quad \forall\, i = 1, 2, ..., |\boldsymbol{R}|$$

$$p_k^i \in \{0, 1\} \quad \forall\, i = 1, 2, ..., |\boldsymbol{R}| \quad \forall\, k = 1, 2, ..., m$$

$$\alpha \geq 0$$

$$\boldsymbol{x} \in \boldsymbol{X}$$

where
M_k is the lower bound for $z_k(\boldsymbol{x})$,
U is an upper bound for $d(z_k(\boldsymbol{x}) - z_k(\boldsymbol{x}'))$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{X}$ where d is a Tchebycheff distance metric,
$\lambda_k > 0 \,\forall\, k = 1, 2, ..., m$,
ϵ is a sufficiently small positive number.

In (P_0), the first constraint ensures that the optimal α value will be the maximum of the Tchebycheff distances from the optimal solution to each representative point.

Sylva and Crema (2007) solves (P_0) in each iteration by adding the newly generated nondominated point to the set $\boldsymbol{R}$. This corresponds to adding m binary variables and $(m + 1)$ linear constraints to the model. The mathematical formulation used in DMA also have the same number of binary variables and linear constraints. Therefore, in terms of computational complexity, DMA and the approach of Sylva and Crema (2007) turn out to be very inefficient as the number of generated representative points increases.

Both approaches have two alternative stopping conditions. They continue until the coverage gap value of the last generated point hits to a lower bound on the coverage gap of $\boldsymbol{R}$. In this case, all nondominated points are at most α better in at least one objective than their closest representatives in $\boldsymbol{R}$. If this lower bound set to zero, then their algorithms generate all nondominated points of a MOIP. Another stopping condition can be an upper bound on the cardinality of the representative set, $|\boldsymbol{R}|$, desired by the DM. Both of these stopping criteria can be used at the same time such that algorithm continues until at least one of them holds.

4.1.2 Subspace Based Approach (SBA) and Territory Defining Algorithm (TDA)

Ceyhan et. al. (2014) propose the Subspace Based Approach (SBA). Similar to the Diversity Maximization Algorithm (DMA), SBA iteratively generates the worst represented point that is the nondominated point with the maximum coverage gap. However, in order to eliminate the dominated regions by the already generated nondominated points, they use a decomposition method proposed by Lokman and Köksalan (2013) instead of adding new binary variables as in DMA. Lokman and Köksalan (2013) enumerate all subspaces that are not dominated by the current representative points. Each subspace is defined by lower bounds on objectives where $\boldsymbol{lb} = \{lb_1, lb_2, ..., lb_m\}$ is a lower bound vector. Then, SBA iteratively solves the following model (P_1) for a maximization problem:

(P_1):

$$Max \;\; \alpha \;\; + \;\; \epsilon \sum_{k=1}^{m} \lambda_k z_k$$

$$s.to.$$

$$z_k(\boldsymbol{x}) \geq lb_k + \alpha \quad \forall\, k = 1, 2, ..., m$$

$$\boldsymbol{x} \in \boldsymbol{X}$$

In each iteration, SBA solves as many models as the number of nondominated subspaces. After searching all subspaces and finding the nondominated points which yield the maximum coverage gap in each subspace, they select the one with the maximum coverage gap among all found points and add this selected point to their representative set. Similar to DMA, the stopping condition of SBA can be either a lower bound on the coverage gap or an upper bound on the cardinality of the representative set.

Although using the search method of Lokman and Köksalan (2013) provides a computational advantage to SBA, the number of models solved in each iteration may increase the solution times substantially as the number of generated points increases. However, their computational experiments show that when both algorithms start with the same initial nondominated point, solution times of SBA are significantly lower than those of DMA especially as the cardinality of the representative set increases. They argue that SBA solves much simpler models than DMA and the size of these models do not increase as the algorithm proceeds. In addition, they keep the solutions of the previously searched subspaces in order to avoid solving the same models unnecessarily. As a result, as the cardinality increases, solution times of SBA increase linearly while solution times of DMA increase exponentially.

In addition to SBA, Ceyhan et. al. (2014) develop the Territory Defining Algorithm (TDA) which outperforms DMA and SBA in terms of solution times. TDA uses the same subspace search method used in SBA and solves the model (P_1) for each subspace. However, in each iteration, TDA does not search all subspaces and does not try to find the nondominated point with the maximum coverage gap. Instead, it only solves the largest subspace and chooses its solution as the next representative point. In other words, TDA solves only a single model in each iteration.

They also suggest that instead of solving the largest subspace in each iteration of TDA, different subspace selection techniques can be applied throughout the algorithm. They also provide experimental results of selecting and solving a random subspace in each iteration. According to their results, both cardinality and the solution times do not change significantly for two different subspace selection techniques applied.

In order to satisfy a prespecified coverage gap, TDA defines specific regions around each generated nondominated point called as *territories*. Specifically, all nondominated points that are α-dominated by a representative point $\boldsymbol{y}$ are included in its territory. Territory of $\boldsymbol{y}$ is defined by the hyperspace ($\mathbf{H}$) in the m-dimensional objective space such that $\mathbf{H} = \{y_k - \Delta \leq z_k(\boldsymbol{x}) \leq y_k + \Delta \quad \forall k \in \{1, ..., m\},\ \boldsymbol{x} \in \boldsymbol{X}\}$. If the region that is dominated by the hyperspace $\mathbf{H}$ is denoted as $\mathbf{H}_D$, then the territory of a representative point $\boldsymbol{y}$ is defined by excluding two spaces from $\mathbf{H}_D$ which are the space dominated by $\boldsymbol{y}$ ($\mathbf{y}_D$) and the space dominating $\boldsymbol{y}$ ($\mathbf{y}_U$), i.e. $\mathbf{T_y} = \mathbf{H}_D \setminus \{\mathbf{y}_D \cup \mathbf{y}_U\}$. The territories defined in two and three dimensional spaces are illustrated in Figures 4.1 and 4.2, respectively.

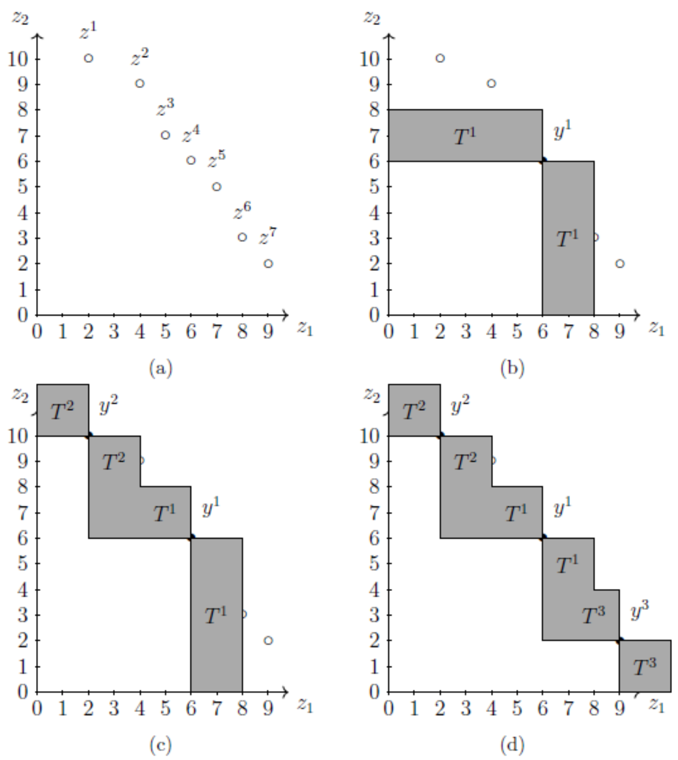


Figure 4.1: Territories in a two dimensional objective space

Source: Ceyhan et. al. (2014)

In order to eliminate the territories from the search space, TDA creates an artificial representative point ($\boldsymbol{y}'$) for each nondominated point ($\boldsymbol{y}$) generated such that $y'_k = y_k + \alpha \ \ \forall k \in \{1, ..., m\}$ for a maximization problem. Then, they define lower bounds of new subspaces by using the objective values of these artificial points. This way, they avoid searching the regions that are α-dominated by the true points generated by TDA. At the end of the algorithm, they show that all nondominated points are α-dominated by the final $\boldsymbol{R}$ and the coverage gap of this representation does not exceed the desired coverage gap value, α.

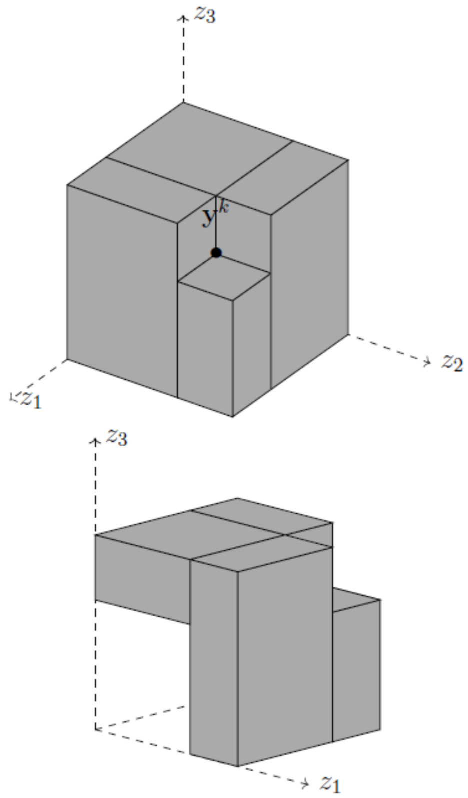


Figure 4.2: Graphical display of the three dimensional territory constructed around point $\mathbf{y}^k$

Source: Ceyhan et. al. (2014)

The main difference of TDA is that since it does not search all subspaces in an iteration, it does not guarantee to generate the worst represented point as in SBA and DMA. Although the solution quality (in terms of cardinality) may be worse than DMA and SBA, Ceyhan et. al. (2014) show that the solution times decrease significantly in TDA. It is much more efficient than DMA and SBA since it searches only a single subspace and eliminates the territories from the search region in each iteration.

4.2 Territory-Excluded Supported Generating Algorithm (TSGA)

In all of these approaches, assuming a coverage gap value specified by the DM, the quality measure for a generated representative set is its cardinality. During an algorithm, in order to reduce the cardinality of the final representative set, we should iteratively generate nondominated points that will represent as many points as possible. Focusing on this purpose, we define a new attribute, called as the *individual representation power (irp)*, for each nondominated point that can be selected as a representative point.

Definition 4.1. Let $\boldsymbol{z} \in \boldsymbol{Z}_{ND}$ be a representative point of a maximization MOMIP. Then, the number of nondominated points which are α-dominated by $\boldsymbol{z}$ is said to be the *individual representation power* of $\boldsymbol{z}$, $P_{\boldsymbol{z}}$.

In developing TSGA, our main purpose is to generate the nondominated points with higher irps in each iteration. Our main motivation is based on some important observations related with the general characteristics of the shapes and the density distributions of the nondominated frontiers in MOMIPs.

Firstly, we use the observations of Köksalan and Lokman (2009) where they approximate the nondominated frontiers of MOCO problems by fitting smooth L_q hypersurfaces. Similar to our scaling method, they scale the objective space using the ranges of the objectives on the nondominated frontier. Specifically, for a maximization problem, a nondominated vector $(z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), ..., z_m(\boldsymbol{x}))$ is transformed to a scaled vector $(z'_1(\boldsymbol{x}), z'_2(\boldsymbol{x}), ..., z'_m(\boldsymbol{x}))$ where $z'_k(\boldsymbol{x}) = \dfrac{z_k(\boldsymbol{x}) - z_k^{NP}(\boldsymbol{x})}{z_k^{IP}(\boldsymbol{x}) - z_k^{NP}(\boldsymbol{x})}$ and $0 \leq z'_k(\boldsymbol{x}) \leq 1 \ \forall k \in \{1, ..., m\}$. In the scaled objective space, they define hypothetical extreme nondominated points such that $z'_k(\boldsymbol{x}) = 1$ and $z'_j(\boldsymbol{x}) = 0 \ \forall j \neq k$. They also generate a centrally located nondominated point that has the minimum Tchebycheff distance from the ideal point by solving $\min_{\boldsymbol{x} \in \boldsymbol{X}}(\max_k(z_k(\boldsymbol{x}) - z_k^{IP}))$. Then, they fit a hypersurface to the nondominated frontier that passes through the central point and all hypothetical extreme points. This hypersurface is called as the L_q surface where $q > 0$ satisfies the following equation for each and every hypothetical nondominated vector: $L_q(\boldsymbol{z}') = (z'_1)^q + (z'_2)^q + ... + (z'_m)^q = 1$.

According to their results, the L_q surface fitting the nondominated frontier of a minimization problem generally has a convex shape whereas it is concave for a maximization problem. An example is shown in Figure 4.3 for a bi-objective knapsack problem with 200 items and 431 nondominated points.

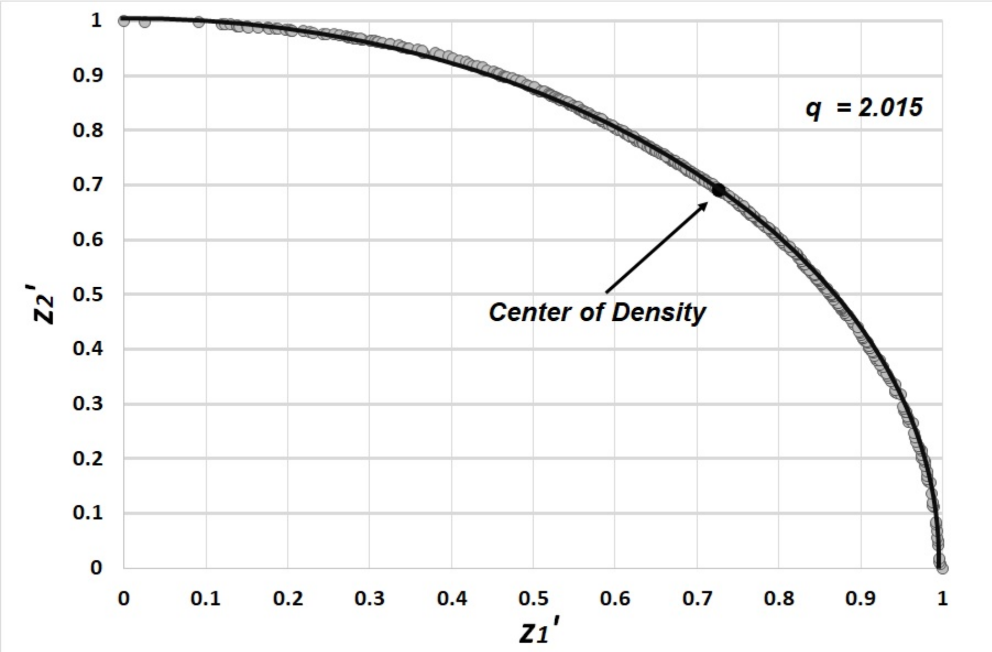


Figure 4.3: The fitted L_q surface of a bi-objective knapsack problem instance

Secondly, Özarık (2017) makes observations related with the typical density distributions of the nondominated frontiers of MOMIPs. They introduce a density measure, approximate the frontiers by L_q surfaces as in Köksalan and Lokman (2009) and make analysis of the common properties of the nondominated sets. They define the central point as the "center of density" and show that the value of their density measure decreases as we move away from this point through the edges of the frontier. Especially in large-sized problems with many nondominated points, this observation seems to be more typical. As an example, the density distribution of a knapsack problem with three objectives and 5652 nondominated points is shown in Figure 4.4 where the density of the nondominated points increases as the color turns from blue to red. According to their findings, 25% of the closest points to the "center of density" contribute approximately 70% to the total density and the rest 75% of these points contribute only approximately 30% to the total density. In other words, if we define the parts around the central point as the "central region", their analyses show that the "central region" of the nondominated frontier is typically the most dense region.

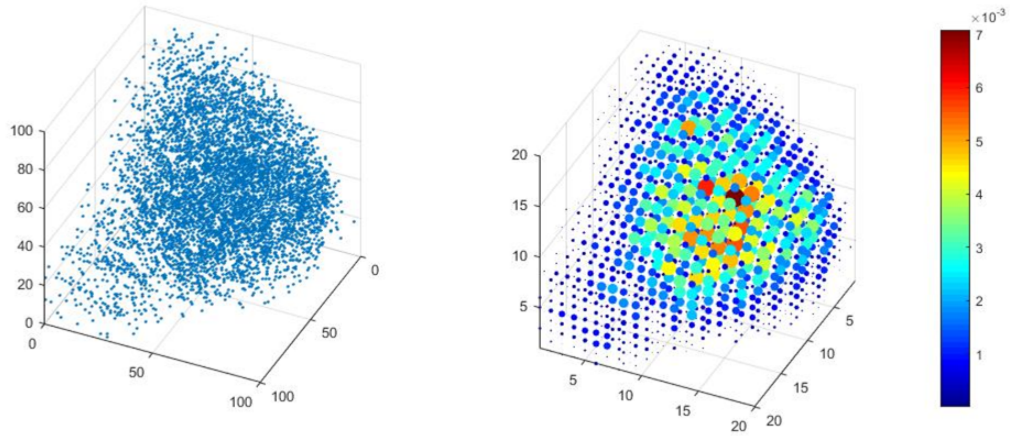
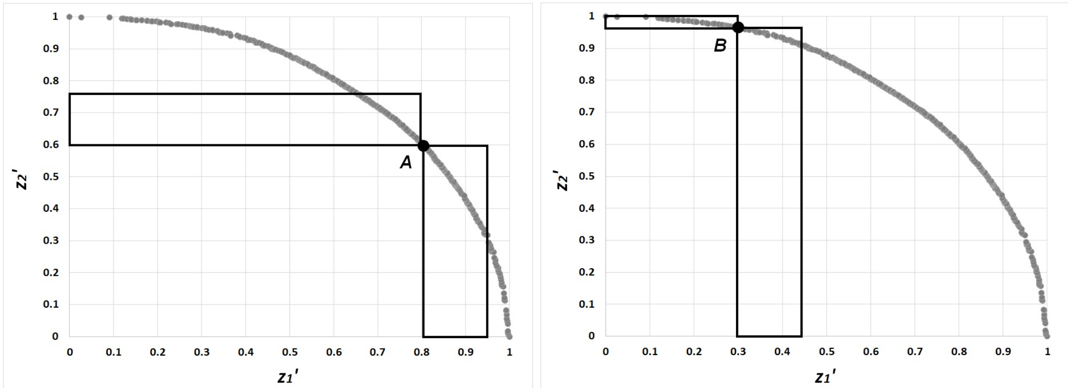


Figure 4.4: Density distribution of a 3-objective knapsack problem instance
Source: Özarık (2017)

If a nondominated point (z) has a high irp, then the number of nondominated points that are α-dominated by z is high. If we remember the territory definition in TDA, this corresponds to a high number of nondominated points located in the territory of z. Therefore, in order to generate the representative points with high irps, we should consider the number of nondominated points within the territories of these representative points. When we consider the findings of Özarık (2017) related with the distributions of nondominated frontiers, we can conclude that if we generate non-dominated points that are located closer to the dense regions of the frontier, their territories will obviously include more number of points and their irps will be high. This is illustrated in Figure 4.5 for a 2-objective knapsack instance with 5652 non-dominated points assuming a coverage gap value of 0.15. As can be seen from this figure, territory of point B includes fewer number of nondominated points than territory of Point A which is located in a dense region of the frontier. Particularly, we can say that the nondominated points located in the dense regions of the frontier (that are closer to the "center of density") have high irps. Considering our main motivation of reducing the cardinality of our final representative set, we desire to generate these nondominated points in our algorithms.

When the typical shapes of the nondominated frontiers of MOMIPs are considered, it can be seen that we can generate nondominated points from the dense regions of the frontier by optimizing a weighted sum of objectives. This is shown in Figure 4.6 for a bi-objective knapsack problem with 431 nondominated points. Since it is a maximization problem, when we maximize a weighted sum of two objectives (shown by the black line), it hits the central region of the frontier and generates a nondominated point with a high irp due to the shape of the frontier.



(a) High irp in the dense region

(b) Low irp in the sparse region

Figure 4.5: Territories in the dense and sparse regions of the nondominated frontier

After making all these observations, we performed additional analyses that support our arguments. We examine the correlation between irps of the nondominated points and the weighted sum of their objective values. The corresponding relationship is shown in Figure 4.7 for a 3-objective knapsack problem instance with 3253 nondominated points. The irp of each nondominated point is calculated for $\alpha = 50$. When we consider the individual data points in Figure 4.7a, although there are some deviations, there is a strong positive correlation between the irps and weighted sum values of the nondominated points. This can be observed more clearly in the histogram in Figure 4.7b which shows that the nondominated points with greater weighted sum values represent more nondominated points on the average.

Based on all of these discussions, we propose the Territory-Excluded Supported Generating Algorithm (TSGA) in which we generate the nondominated point that has the maximum weighted sum value in the reduced search space at each iteration. We iteratively reduce the search space by excluding not only the dominated regions but also the regions that are α-dominated by the previously generated representative points. In order to achieve this, we define territories around each generated point as in TDA. In order to exclude these territories from the search space, we define artificial representative points that correspond to the upper corner points of the territories.

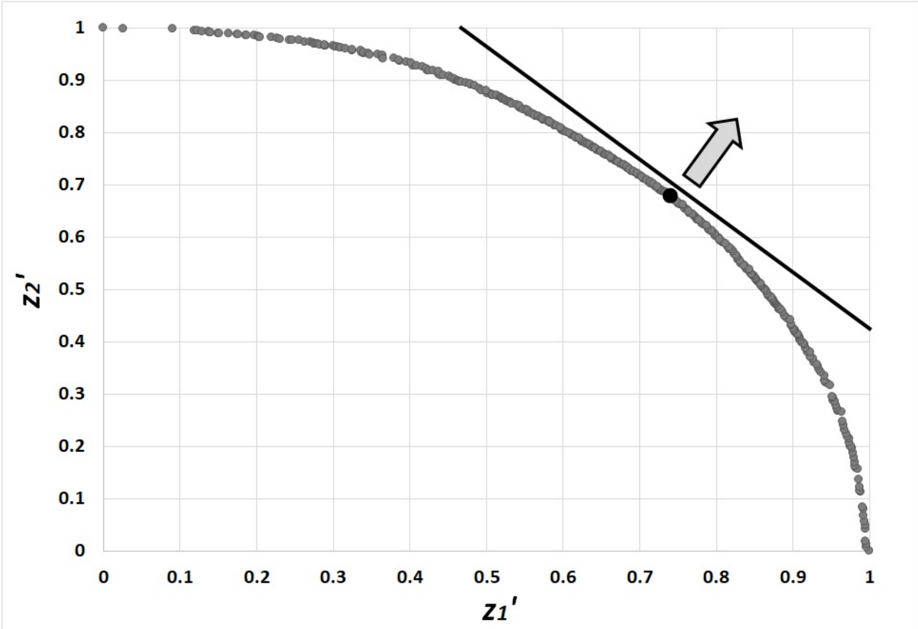
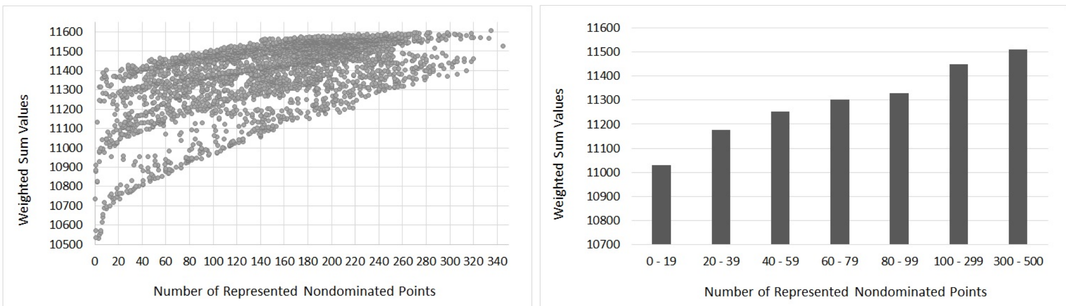


Figure 4.6: Generating a nondominated point with high irp

For a given coverage gap value (α) and a generated nondominated point (***y***), an artificial nondominated point (***y***′) is defined as $y'_k = y_k + \alpha \;\; \forall k \in \{1, ..., m\}$ for a maximization problem. Using these artificial points, we exclude the territories from the feasible objective space and obtain a reduced search space in each iteration. Then, by solving a weighted sum problem, we generate a supported nondominated point of this reduced problem.



(a) Individual Data Points (b) Histogram

Figure 4.7: Individual representation power vs. Weighted sum of objectives

In this weighted sum problem, we choose the weights based on the distribution of the nondominated points over the feasible objective space. To maximize the individual representation power, we target the dense regions. Özarık (2017) shows that the dense regions are typically located closer to the central point, $\boldsymbol{z}^C$, that is the nondominated point with the minimum Tchebycheff distance to the ideal point. Based on this observation, we calculate the objective weights by using the plane that is tangent to the L_q surface at the point $\boldsymbol{z}^C$. Specifically, the coefficients in the equation of this tangent plane are normalized and set as the objective weights in our weighted sum problem. For a bi-objective knapsack problem with 431 nondominated points, the L_q curve, the "center of density" and the tangent line at this point are illustrated in Figure 4.8. This way, the weight of each objective is chosen so that the weighted sum problem generates nondominated points from the dense regions of the nondominated frontier.

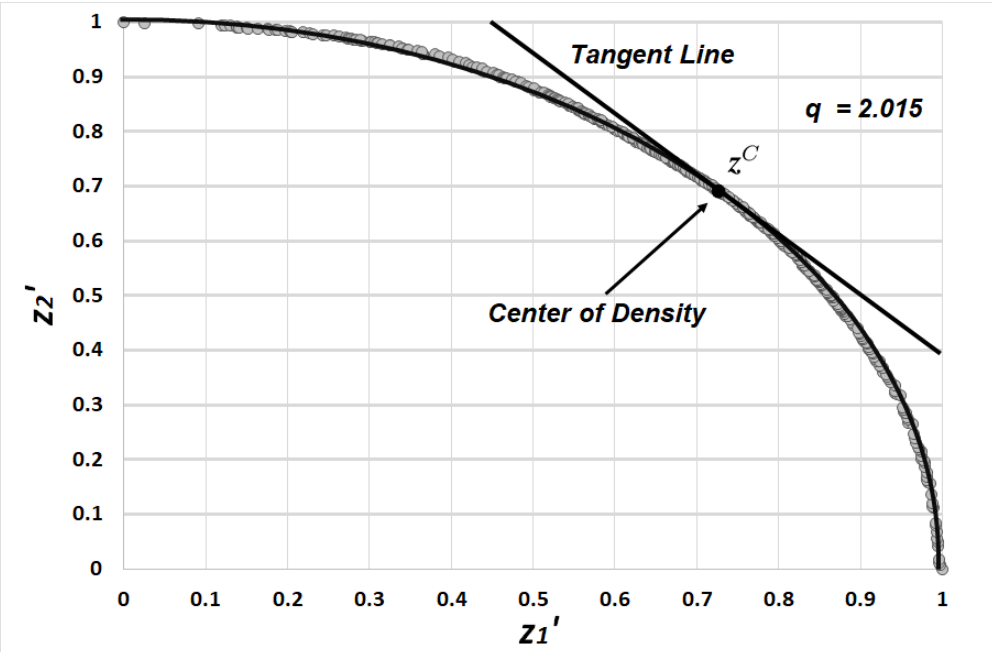


Figure 4.8: The fitted L_q surface and the tangent line at the center of density of a bi-objective knapsack problem instance

We search the objective space using the recently proposed decomposition method by Dächert et. al. (2017) which outperforms the decomposition method of Lokman and Köksalan (2013) in terms of computational efficiency. Similar to Lokman and Köksalan (2013), the search procedure of Dächert et. al. (2017) partitions the search space into subspaces and each subspace is defined by a set of lower bounds on objectives (for a a maximization problem), $\boldsymbol{lb} = \{lb_1, lb_2, ..., lb_m\}$. However, they introduce a specific neighborhood relationship between these lower bounds which are updated in each iteration. This update procedure is designed very efficiently and saves computation time. Another point which makes their procedure faster is that they avoid searching redundant search zones where redundancy is defined as follows:

Definition 4.2. Let $\boldsymbol{lb}^1$ and $\boldsymbol{lb}^2$ be two lower bound vectors of a maximization MOMIP. If $lb_k^1 = lb_k^2$ for some $k = 1, ..., m$ and $lb_j^1 > lb_j^2$ for $\forall j \neq k$, then $\boldsymbol{lb}^1$ is said to be a *redundant lower bound vector* which defines a *redundant search zone*.

By definition, it is known that the optimal solution in the search zone of $\boldsymbol{lb}^2$ will be either the same as or better than the optimal solution in the search zone of $\boldsymbol{lb}^1$. This is why there is no need to solve the submodel of the region defined by $\boldsymbol{lb}^1$. As the problem size grows, Dächert et. al. (2017) observe that the number of redundant search zones increases. Therefore, their search procedure results in a substantial decrease in solution times. Using this decomposition method, TSGA iteratively generates the nondominated point with the maximum weighted sum value in each subspace by solving the model (P_2). Then, among all these generated points, TSGA selects the one with the maximum weighted sum value as the new representative point.

(P_2):

$$
\begin{aligned}
& Max \ \sum_{k=1}^{m} \lambda_k z_k \\
& s.to. \\
& \quad z_k(\boldsymbol{x}) \geq lb_k \quad \forall\, k = 1, 2, ..., m \\
& \quad \boldsymbol{x} \in \boldsymbol{X}
\end{aligned}
$$

In the model (P_2), λ_k denotes the coefficient of objective k including a scaling factor and a weight factor. Specifically, we can define this coefficient as $\lambda_k = \dfrac{w_k}{R_k}$ where R_k is the range of objective k on the efficient frontier and w_k is the weight of objective k calculated from the center of density of the nondominated frontier. Since we work on the scaled objective space, the prespecified coverage gap value is defined on the interval [0, 1]. This way, the size of the territories constructed around each generated point is also scaled.

Although TSGA calculates the weights of the objectives based on the center of density of the nondominated frontier, some interactive applications can be designed so that these weights are adjusted considering the preferences of the DM. Accordingly, the territory sizes can be set inversely proportional to these weights. This way, we can discriminate the nondominated solutions that are more preferable by the DM. Examples for such applications are discussed in Section 4.5.

During the algorithm, we keep some sets and lists that are defined as follows: ***R*** is the set of the generated nondominated points, ***A*** is the set of the artificial nondominated points and ***LB*** is the list of the nonredundant lower bound vectors with the corresponding optimal solutions of (P_2). At the beginning of the algorithm, the DM specifies the coverage gap value, α^*. Then, the outline of the TSGA is given in the following algorithm:

Algorithm 1 Territory-Excluded Supported Generating Algorithm (TSGA)

Initialization: $\boldsymbol{R} = \emptyset$, $\boldsymbol{A} = \emptyset$, $\boldsymbol{LB} = \emptyset$. $n = 0$.

Step 0. Generate the initial nondominated point ($\boldsymbol{y}^1$) by solving model (P_2) for $lb_k = 0 \;\; \forall\, k = 1, 2, ..., m$. $\boldsymbol{R} = \{\boldsymbol{y}^1\}$, $\boldsymbol{A} = \{\boldsymbol{y}'^1\}$, $n = 1$. Update ***LB***.

Step 1. For each $\boldsymbol{lb} \in \boldsymbol{LB}$ **do**

Solve (P_2) if necessary. If (P_2) is feasible, let $\boldsymbol{z}^{\boldsymbol{lb}}$ be the nondominated point found. Update ***LB***.

If there exists any feasible solution **then**

$n = n+1$. $\boldsymbol{y}^n = \arg\max_{\boldsymbol{z}^{\boldsymbol{lb}}} \left\{ \sum_k \lambda_k z_k \right\}$. $\boldsymbol{R} = \boldsymbol{R} \cup \{\boldsymbol{y}^n\}$. $\boldsymbol{A} = \boldsymbol{A} \cup \{\boldsymbol{y}'^n\}$. Update ***LB***.

Else go to Step 2.

Step 2. Stop.

TSGA stops when the entire search space is infeasible which means that each nondominated point lies within the territory of at least one representative point. In other words, each nondominated point is α-dominated by at least one representative point at the end of the algorithm. This way, we guarantee that all nondominated points are α-dominated by the final representative set generated by TSGA.

4.3 Territory-Excluded Supported Generating Algorithm - II (TSGA-II)

In this section, we introduce our second algorithm, TSGA-II, that generates representative sets of nondominated points for MOMIPs for a given coverage gap value. Similar to TSGA, we generate nondominated points by solving a weighted sum problem in the reduced search space obtained by excluding the territories constructed around the points generated in each iteration. By eliminating these territories, we guarantee to represent all nondominated points by at most a prespecified coverage gap value.

The main advantage of this algorithm is that it provides us the opportunity to work on an $(m-1)$-dimensional objective space. We randomly choose one of the objectives (say the p^{th} objective) and we generate representative points in a nonincreasing order of z_p. After eliminating the nondominated points within the territories of current representatives in each iteration, we first generate the nondominated point whose p^{th} objective function value is the best among all nondominated points from the reduced space (denoted by $\boldsymbol{z}^{I_p}(\boldsymbol{x})$). Then, we generate the nondominated point from the reduced space that has the highest weighted sum of the remaining objectives (for a maximization problem) and α-dominates $\boldsymbol{z}^{I_p}(\boldsymbol{x})$. This generated nondominated point is selected as the new representative point.

In the first iteration of TSGA-II, the first representative point, $\boldsymbol{y}^1(\boldsymbol{x})$, is generated by solving model (P_4) so that it α-dominates $\boldsymbol{z}^{I_p}(\boldsymbol{x})$ which is found by solving model (P_3). We show that the p^{th} objective function values of all other nondominated points are at most α larger than the p^{th} objective function value of $\boldsymbol{y}^1(\boldsymbol{x})$. This enables us to define the territories in the $(m-1)$-dimensional objective space. As a result, we optimize the weighted sum of $(m-1)$ objectives in the reduced search space.

(P_3):

$$Max \quad \lambda_p z_p \; + \; \epsilon \sum_{k=1,\; k\neq p}^{m} \lambda_k z_k$$

$$s.to.$$

$$\boldsymbol{x} \in \boldsymbol{X}$$

(P$_4$):

$$Max \quad \sum_{k=1,\ k\neq p}^{m} \lambda_k z_k \ + \ \epsilon(\lambda_p \, z_p)$$

$$s.to.$$

$$z_k \geq z_k^{I_p} - \alpha R_k \quad \forall\, k = 1, 2, ..., m$$

$$\boldsymbol{x} \in \boldsymbol{X}$$

In (P$_3$) and (P$_4$), ϵ is a sufficiently small number and $\lambda_k = \dfrac{w_k}{R_k}$ where R_k is the range of objective k on the efficient frontier and w_k is the weight of objective k. Due to the scaling of objectives, the territory sizes are defined as $\alpha R_k \ \ \forall k \in \{1, ..., m\}$ where $\alpha \in [0, 1]$. In (P$_4$), the first constraint ensures that the generated point α-dominates $\boldsymbol{z}^{I_p}(\boldsymbol{x})$ that is the nondominated point corresponding to the optimal solution of (P$_3$).

Proposition 4.1. *In a maximization MOMIP, let* $\mathbf{y}^1 \in \mathbf{Z}_{ND}$ *be the first representative point generated by TSGA-II. Then, for any* $\boldsymbol{z} \in \mathbf{Z}_{ND}$*,* $y_p^1 \geq z_p - \alpha R_p$*.*

Proof. By definition, $z_p^{I_p} \geq z_p \ \ \forall\, \boldsymbol{z} \in \mathbf{Z}_{ND}$.
Since $\boldsymbol{z}^{I_p}$ is α-dominated by $\boldsymbol{y}^1$: $\quad y_k^1 \geq z_k^{I_p} - \alpha R_k \quad \forall\, k = 1, ..., p, ..., m.$
Then, it directly follows: $\quad y_p^1 \geq z_p^{I_p} - \alpha R_p \geq z_p - \alpha R_p \quad \forall\, \boldsymbol{z} \in \mathbf{Z}_{ND}.$

□

As stated in Proposition 4.1., since the first representative point generated by TSGA-II represents all nondominated points in the p^{th} objective, there is no need to construct territories in the p^{th} dimension of the objective space. This property is very significant since it allows us to work on an $(m-1)$-dimensional space throughout the algorithm.

In TSGA-II, we again search the objective space using the decomposition method of Dächert et. al. (2017). In iteration n, TSGA-II solves model (P$_5$) for each nonredundant lower bound vector, ***lb***. This model generates the nondominated point with the maximum p^{th} objective value in the search space defined by ***lb***. Then, TSGA-II selects the point with the maximum p^{th} objective value among the nondominated points corresponding to the optimal solutions of (P$_5$) for all lower bound vectors. Let the selected point be denoted as $\boldsymbol{z}^{I_p^n}$. Then, the algorithm solves (P$_6$) for each subspace in order to generate the nondominated point that α-dominates $\boldsymbol{z}^{I_p^n}$ and has the maximum weighted sum value for all objectives except z_p. Lastly, among all generated points, TSGA-II selects the one with the maximum weighted sum value (for $(m-1)$ objectives) as the new representative.

(P$_5$):

$$
\begin{aligned}
& Max \quad \lambda_p z_p \; + \; \epsilon \sum_{k=1,\; k \neq p}^{m} \lambda_k z_k \\
& s.to. \\
& \qquad z_k(\boldsymbol{x}) \geq lb_k \quad \forall\, k \neq p \\
& \qquad \boldsymbol{x} \in \boldsymbol{X}
\end{aligned}
$$

(P$_6$):

$$
\begin{aligned}
& Max \quad \sum_{k=1,\; k \neq p}^{m} \lambda_k z_k \; + \; \epsilon(\lambda_p \, z_p) \\
& s.to. \\
& \qquad z_k(\boldsymbol{x}) \geq z_k^{I_p^n}(\boldsymbol{x}) - \alpha^* R_k \quad \forall\, k = 1, 2, ..., m \\
& \qquad z_k(\boldsymbol{x}) \geq lb_k \quad \forall\, k \neq p \\
& \qquad \boldsymbol{x} \in \boldsymbol{X}
\end{aligned}
$$

where ϵ is a sufficiently small number, $\lambda_k = \dfrac{w_k}{R_k}$ where R_k is the range of objective k on the efficient frontier and w_k is the weight of objective k. The weight of each objective is assigned as in TSGA based on the density distribution of the nondominated frontier of a MOMIP. Specifically, these weights are the coefficients in the equation of the tangent plane to the L_q surface at the center of density.

Another property of TSGA-II which should be highlighted is that it generates the representative sets with the optimal cardinality for bi-objective mixed-integer problems. Since we work on an $(m-1)$-dimensional space, the original bi-objective problem reduces to a single objective problem. As a result, in addition to the solution quality, TSGA-II provides a significant computational advantage in bi-objective mixed-integer problems.

Proposition 4.2. *In a bi-objective mixed-integer problem, let* $\boldsymbol{R} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^n\}$ *be the current representative set in iteration n. Let the union of territories constructed around n representatives be* $\boldsymbol{T} = \{\boldsymbol{T}_{\boldsymbol{y}^1} \cup \boldsymbol{T}_{\boldsymbol{y}^2} \cup ... \cup \boldsymbol{T}_{\boldsymbol{y}^n}\}$. *Then, TSGA-II generates* $\boldsymbol{y}^{n+1}$ *with the maximum individual representation power in the reduced objective space, i.e.* $P_{\boldsymbol{y}^{n+1}} = \max\limits_{\boldsymbol{z} \in \mathbf{Z}_{ND} \setminus \boldsymbol{T}} P_{\boldsymbol{z}}$.

Proof. Without loss of generality, let $p = 1$.

Consider n = 0.

For a maximization problem, let $\boldsymbol{z}^{I_1}$ be the optimal solution of (P_3) and $\boldsymbol{y}^1$ be the optimal solution of (P_4).

To get contradiction, suppose that there exists $\boldsymbol{z}^* \neq \boldsymbol{y}^1 \in \mathbf{Z}_{ND}$ such that $P_{\boldsymbol{z}^*} = \max\limits_{\boldsymbol{z} \in \mathbf{Z}_{ND}} P_{\boldsymbol{z}}$.

Then, $P_{\boldsymbol{y}^1} < P_{\boldsymbol{z}^*}$ which implies that there exists $\boldsymbol{z} \in \mathbf{Z}_{ND}$ such that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^*}$ and $\boldsymbol{z} \notin \mathbf{T}_{\boldsymbol{y}^1}$. Since $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^*}$,

$$z_1 \leq z_1^* + \alpha^* R_1 \tag{4.1}$$

$$z_2 \leq z_2^* + \alpha^* R_2 \tag{4.2}$$

Since $\boldsymbol{z} \notin \mathbf{T}_{\boldsymbol{y}^1}$, at least one of the inequalities (4.3) and (4.4) must be satisfied:

$$z_1 > y_1^1 + \alpha^* R_1 \tag{4.3}$$

$$z_2 > y_2^1 + \alpha^* R_2 \tag{4.4}$$

By the construction of the algorithm,

$$z_1^{I_1} \leq y_1^1 + \alpha^* R_1 \tag{4.5}$$

$$z_2^* \leq y_2^1 \tag{4.6}$$

By definition of the ideal point, $z_1 \leq z_1^{I_1}$.

By substitution to (4.5), it implies to $z_1 \leq y_1^1 + \alpha^* R_1$ which contradicts with (4.3).

From (4.2) and (4.6), $z_2 \leq z_2^* + \alpha^* R_2 \leq y_2^1 + \alpha^* R_2$ which contradicts with (4.4).

Therefore, if there exists $\boldsymbol{z} \in \mathbf{Z}_{ND}$ such that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^*}$, this implies that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{y}^1}$.

Consider n > 0. Let $\boldsymbol{z}^{I_1^n}$ be the nondominated point such that $z_1^n = \max\limits_{\boldsymbol{z} \in \mathbf{Z}_{ND} \setminus \mathbf{T}} z_1$.

Then, TSGA-II generates $\boldsymbol{y}^{n+1}$ such that
$y_2^{n+1} = \max_{z \in \mathbf{Z}_{ND} \setminus \mathbf{T}} \{z_2 | \; z_k \geq z_k^{I_1^n} - \alpha^* R_k \;\; \forall k, \;\; \mathbf{x} \in \mathbf{X}, \; \epsilon > 0\}$.

Suppose that there exists $\boldsymbol{z}^{n*} \in \mathbf{Z}_{ND} \setminus \mathbf{T}$ such that $P_{\boldsymbol{z}^{n*}} = \max_{z \in \mathbf{Z}_{ND} \setminus \mathbf{T}} P_{\boldsymbol{z}}$.

Then, $P_{\boldsymbol{y}^{n+1}} < P_{\boldsymbol{z}^{n*}}$ which implies that there exists $\boldsymbol{z} \in \mathbf{Z}_{ND} \setminus \mathbf{T}$ such that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^{n*}}$ and $\boldsymbol{z} \notin \mathbf{T}_{\boldsymbol{y}^{n+1}}$.

Since $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^{n*}}$,

$$z_1 \leq z_1^{n*} + \alpha^* R_1 \tag{4.7}$$

$$z_2 \leq z_2^{n*} + \alpha^* R_2 \tag{4.8}$$

Since $\boldsymbol{z} \notin \mathbf{T}_{\boldsymbol{y}^{n+1}}$, at least one of the inequalities (4.9) and (4.10) must be satisfied:

$$z_1 > y_1^{n+1} + \alpha^* R_1 \tag{4.9}$$

$$z_2 > y_2^{n+1} + \alpha^* R_2 \tag{4.10}$$

By the construction of the algorithm,

$$z_1^{I_1} \leq y_1^{n+1} + \alpha^* R_1 \tag{4.11}$$

$$z_2^* \leq y_2^{n+1} \tag{4.12}$$

By definition of the ideal point, $z_1 \leq z_1^{I_1^n}$.
By substitution to (4.11), it implies to $z_1 \leq y_1^{n+1} + \alpha^* R_1$ which contradicts with (4.9).
From (4.8) and (4.12), $z_2 \leq z_2^{n*} + \alpha^* R_2 \leq y_2^{n+1} + \alpha^* R_2$ which contradicts with (4.10).
Therefore, if there exists $\boldsymbol{z} \in \mathbf{Z}_{ND} \setminus \mathbf{T}$ such that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{z}^{n*}}$, this implies that $\boldsymbol{z} \in \mathbf{T}_{\boldsymbol{y}^{n+1}}$.

□

Corollary 4.1. *In a bi-objective mixed-integer problem, let* $\boldsymbol{R} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{n_f}\}$ *be the final representative set generated by TSGA-II. Then,* n_f *is the minimum cardinality which satisfies a given coverage gap value.*

Proof. Let the union of territories constructed around n_f representative points be $\mathbf{T}_f = \{\mathbf{T}_1 \cup \mathbf{T}_2 \cup ... \cup \mathbf{T}_{n_f}\}$. In each iteration n, TSGA-II generates $\boldsymbol{y}^n$ such that $P_{\boldsymbol{y}^n} = \max_{z \in \mathbf{Z}_{ND} \setminus \mathbf{T}} P_{\boldsymbol{z}}$ where $z_k \geq z_k^{I_1} - \alpha^* R_k \;\; \forall k$. Therefore, since TSGA-II represents the maximum number of points in each iteration, it represents all nondominated points with the minimum number of representative points. □

The sets and lists that we keep throughout the algorithm are all the same with TSGA. ***R*** is the set of the generated representative nondominated points, ***A*** is the set of the artificial representative nondominated points and ***LB*** is the list of the nonredundant lower bound vectors with the corresponding optimal solutions of (P_5). For a coverage gap value (α^*) specified by the DM, the algorithm stops when all available search zones are infeasible. Similar to TSGA, since we exclude the territories constructed around each representative point throughout the algorithm, it is guaranteed that all nondominated points are α-dominated by at least one of the representatives when the algorithm stops. The outline of TSGA-II is provided in the following algorithm:

Algorithm 2 Territory-Excluded Supported Generating Algorithm-II (TSGA-II)

Initialization: ***R*** = Ø, ***A*** = Ø, ***LB*** = Ø. $n = 0$. Randomly choose an objective p.

Step 0. Solve (P_3). Let the optimal solution be $\boldsymbol{z}^{I_p}$. Then, solve (P_4). Let the optimal solution be $\boldsymbol{y}^1$. ***R*** = $\{\boldsymbol{y}^1\}$, ***A*** = $\{\boldsymbol{y}'^1\}$, $n = 1$. Update ***LB***.

Step 1. For each ***lb*** $\in$ ***LB*** **do**

Solve (P_5) if necessary. If (P_5) is feasible, let $\boldsymbol{z}^{\boldsymbol{lb}}$ be the nondominated point found. Update ***LB***.

If there exists any feasible solution **then**

$\boldsymbol{z}^{I_p^n} = \arg\max_{\boldsymbol{z}^{\boldsymbol{lb}}} \{z_p\}$. Update ***LB***. Go to Step 2.

Else go to Step 3.

Step 2. $n = n + 1$. **For** each ***lb*** $\in$ ***LB*** **do**

Solve (P_6). If (P_6) is feasible, let $\boldsymbol{y}^{\boldsymbol{lb}}$ be the nondominated point found. Update ***LB***.

If there exists any feasible solution **then**

$\boldsymbol{y}^n = \arg\max_{\boldsymbol{y}^{\boldsymbol{lb}}} \left\{ \sum_{k \neq p} \lambda_k z_k \right\}$. $\boldsymbol{R} = \boldsymbol{R} \cup \{\boldsymbol{y}^n\}$. $\boldsymbol{A} = \boldsymbol{A} \cup \{\boldsymbol{y}'^n\}$. Update ***LB***.

Else $\boldsymbol{y}^n = \boldsymbol{z}^{I_p^n}$.

Go to Step 1.

Step 3. Stop.

While generating a representative set, there is always a trade-off between the solution quality and the solution time. Although TSGA-II provides a significant improvement in the solution quality, TSGA is much more efficient in terms of the computational efforts. Since TSGA-II reduces one dimension in the objective space, the computation times could be expected to decrease. However, TSGA-II generates two nondominated points in each iteration by solving two models for each subspace. We keep the optimal solutions of (P_5) for each subspace that is solved in the previous iterations. However, the optimal solution of (P_6) for a subspace may change since the point $z^{I_p^n}$ is updated in each iteration n. Therefore, we need to resolve (P_6) for each feasible subspace in an iteration. As a result, the solution times may increase significantly especially in large-sized problems.

4.4 Illustration of All Algorithms

In this section, we demonstrate the procedures of TSGA, TSGA-II and DMA (which is equivalent to SBA such that both algorithms iteratively generate the nondominated point with the maximum coverage gap value).

Consider a bi-objective knapsack problem instance with 40 items and 14 nondominated points. The layout of the nondominated points in the scaled objective space is shown in Figure 4.9. Suppose that a representative set will be generated to satisfy a coverage gap value $\alpha = 0.25$.

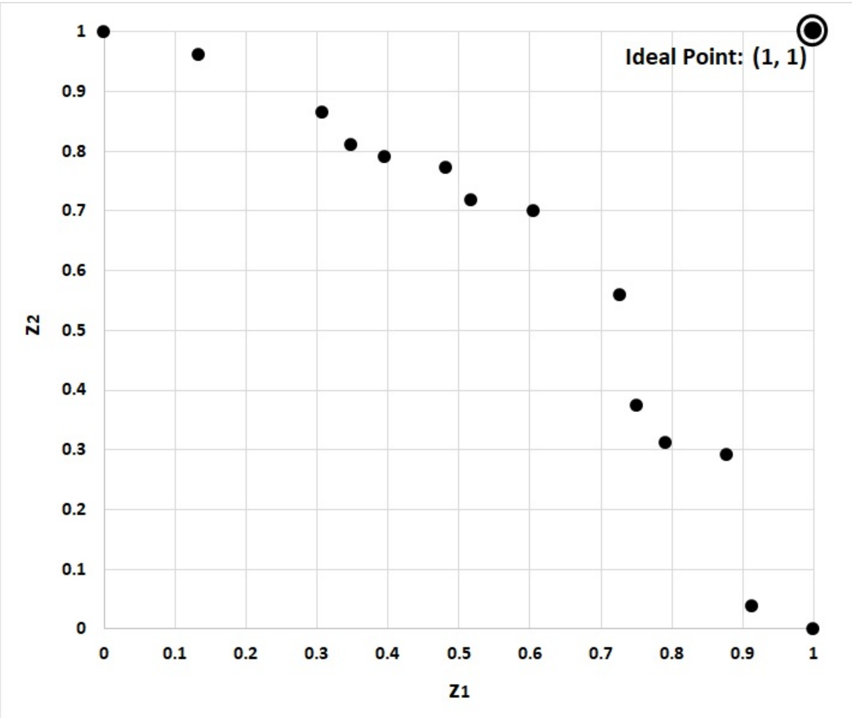


Figure 4.9: Nondominated points of the example bi-objective knapsack problem

We first need to calculate the weights of objectives for TSGA and TSGA-II. In terms of the Tchebycheff distance metric, the closest nondominated point to the ideal point is found as $(0.61, 0.70)$ that is the center of density ($\boldsymbol{z}^C$) of this problem. Then, the q value is calculated as 1.610 when we solve the equation $L_q((0.61, 0.70)) = 0.61^q + 0.70^q = 1$. The equation of the line that is tangent to the L_q curve at the point $\boldsymbol{z}^C$ is $z_2 = -0.919z_1$. When the coefficients in this equation are normalized, the weight of each objective is found as $w_1 = 0.479$ and $w_2 = 0.521$.

Both DMA and TSGA are started with the same initial point $(0.61, 0.70)$ that is the optimal solution of $\max_{\boldsymbol{x} \in \boldsymbol{X}}(w_1 z_1(\boldsymbol{x}) + w_2 z_2(\boldsymbol{x}))$. For TSGA-II, suppose the randomly selected objective is z_1, then it generates the first representative point by solving the model (P$_4$) where $\boldsymbol{z}^{I_1} = (1, 0)$. Then, the generated representative sets are as follows:

$\boldsymbol{R}_{DMA} \quad = \{(0,1),\ (0.61,0.70),\ (0.88,0.29),\ (1,0)\}$,

$\boldsymbol{R}_{TSGA} \quad = \{(0.13,0.96),\ (0.61,0.70),\ (0.88,0.29)\}$,

$\boldsymbol{R}_{TSGA-II} = \{(0.40,0.79),\ (0.75,0.37)\}$.

In Figures 4.10, 4.11 and 4.12, these sets are shown where the representative points are numbered according to their order of generation. The territories constructed around each generated point in TSGA and TSGA-II are also shown.

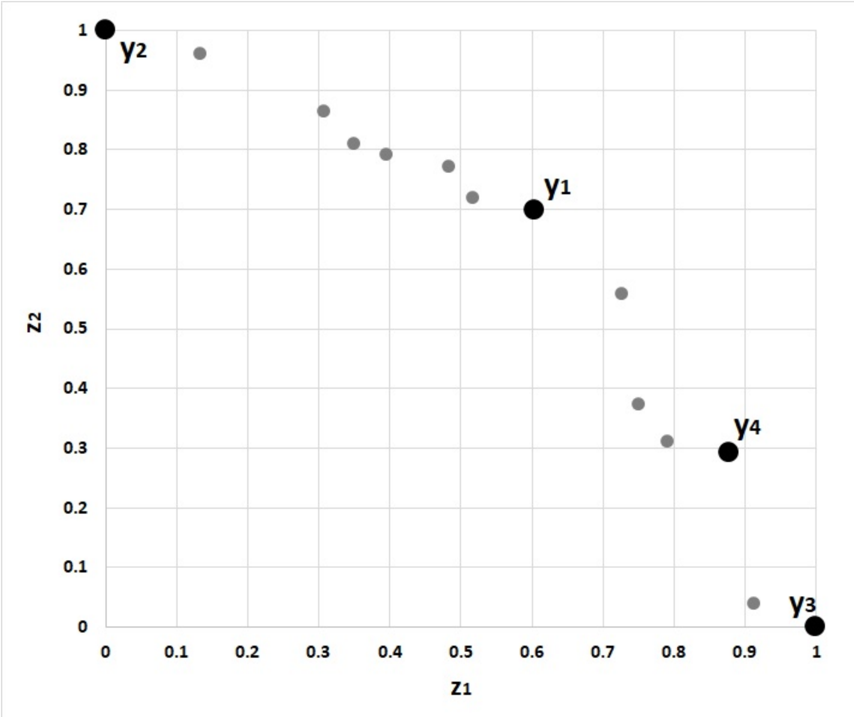


Figure 4.10: Illustration of DMA for the bi-objective example problem ($\alpha = 0.25$)

DMA generates the most diverse representatives located through the edges of the nondominated frontier. Although TSGA starts with the same initial point, it generates the nondominated points located closer to the central region of the frontier. This is why the cardinality of $\boldsymbol{R}_{TSGA}$ is less than the cardinality of $\boldsymbol{R}_{DMA}$. In Figure 4.12, it is clearly seen that TSGA-II satisfies the same coverage gap with only two representatives located in the dense regions of the frontier. As a result, TSGA-II generates the representative set with the minimum cardinality.

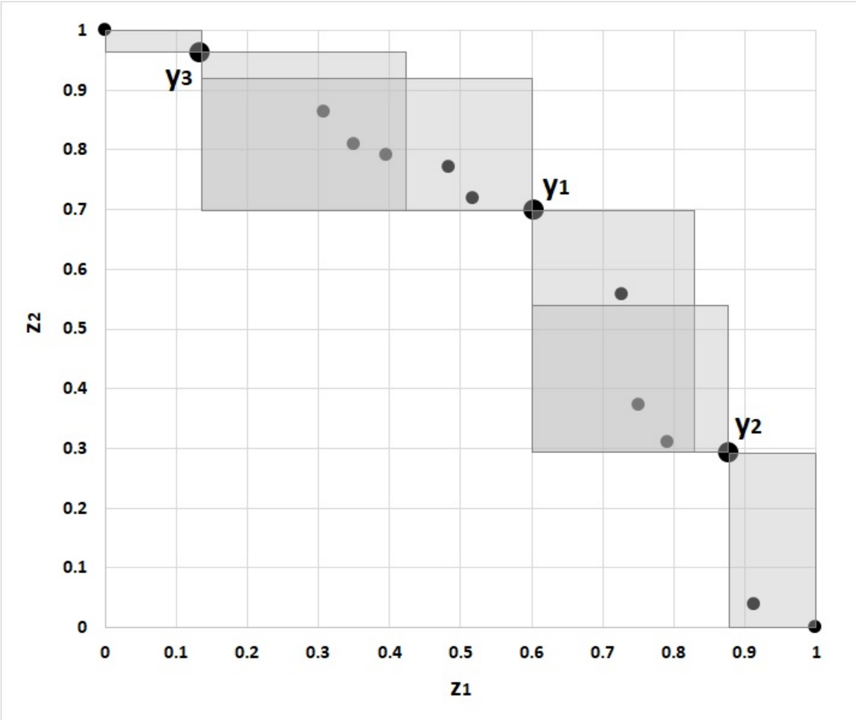


Figure 4.11: Illustration of TSGA for the bi-objective example problem ($\alpha = 0.25$)

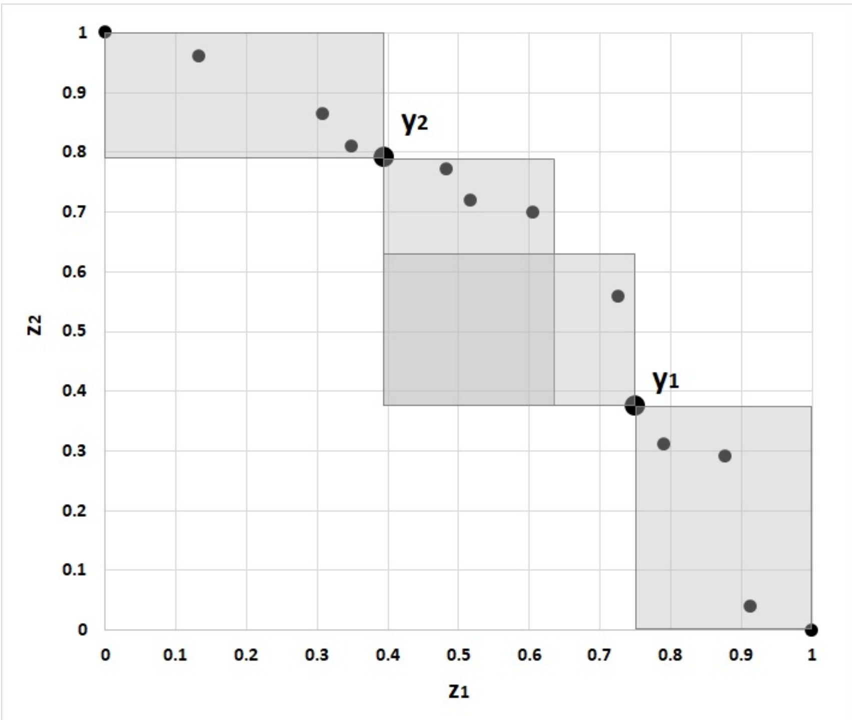


Figure 4.12: Illustration of TSGA-II for the bi-objective example problem ($\alpha = 0.25$)

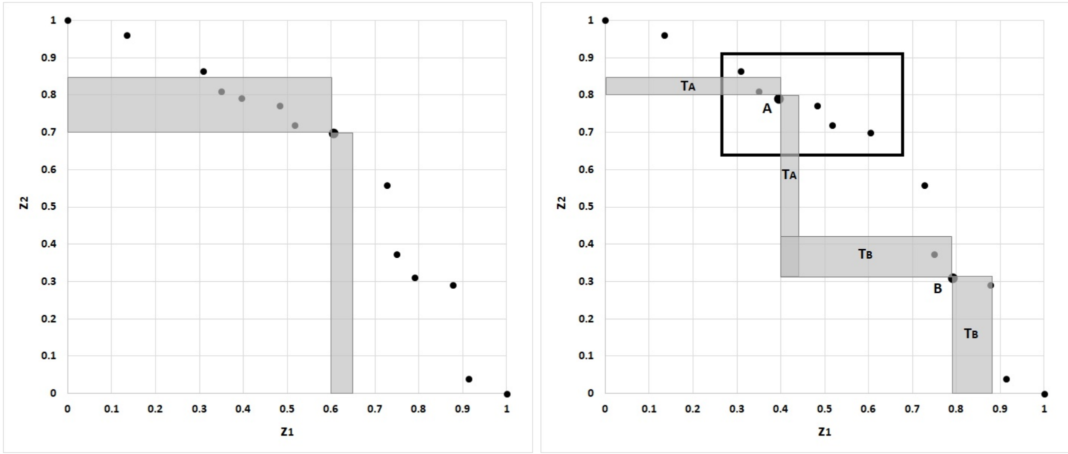
4.5 Interactive Applications of TSGA and TSGA-II

For a given coverage gap value α, TSGA and TSGA-II generate representative nondominated points by optimizing a weighted sum of objectives as in the models (P_2) and (P_6), respectively. As stated before, the coefficient of the k^{th} objective function, λ_k, in these models is composed of a scaling coefficient and a weight coefficient. Specifically, $\lambda_k = \dfrac{w_k}{R_k}$ where R_k is the range of objective k on the efficient frontier and w_k is the weight of objective k calculated based on the density distribution of the nondominated frontier. For $0 \leq w_k \leq 1 \ \forall k = 1, ..., m$, the size of the territories constructed around each representative nondominated point can be defined as $\dfrac{\alpha}{\lambda_k} \ \forall k = 1, ..., m$.

Particularly, we define our territory sizes as inversely proportional to the coefficients of objectives (λ_k). Using this relation, we can adjust the weights and territory sizes of objectives according to the preferences of the DM. We present two alternative scenarios for integrating the preferences of the DM to our algorithms.

4.5.1 Preference-Based Weight Selection in TSGA and TSGA-II

The DM may specify preferences on objectives such that the allowed coverage gap value is smaller for more important objectives. In this case, we consider to set the weights of objective functions in our models as directly proportional to the weights given by the DM. Since the territory size in an objective is defined as inversely proportional to its weight coefficient, this makes the territory size larger for an objective with less priority. For instance, for a bi-objective problem, if the DM gives more importance to the first objective than the second one, then we set our weight coefficients such that $w_1 > w_2$. Accordingly, $\lambda_1 > \lambda_2$ and the territory size in objective 1 is smaller than objective 2, i.e. $\dfrac{\alpha}{\lambda_1} < \dfrac{\alpha}{\lambda_2}$. These territory sizes are shown in Figure 4.13a.



(a) Preference-based territories ($w_1 > w_2$) (b) Desired region in the objective space

Figure 4.13: Illustrations of the interactive applications of TSGA and TSGA-II

4.5.2 Defining Indifference Regions in TSGA and TSGA-II

We may have the information related with the amounts for each objective that makes the DM indifferent between alternative solutions. The DM can state these amounts either at the beginning of the algorithm or can update these amounts based on the generated points throughout the algorithm. Then, our territories can be defined as *indifference regions* and their sizes can be set as directly proportional to these indifference amounts. Specifically, if the DM is known to be indifferent between y_k and $y_k + \Delta_k$ for the k^{th} objective, then our artificial points and the coefficients of objectives are defined as follows: $y'_k = y_k + \dfrac{\Delta_k}{R_k}$ and $\lambda_k = \dfrac{R_k}{\Delta_k} \ \forall k \in \{1, ..., m\}$.

Moreover, the DM can specify a desired range of values for each objective. Then, by using these ranges, we can define regions in the objective space such that the DM would prefer nondominated points generated from these regions. In our algorithms, we generate a subset of nondominated points which represents the entire frontier. However, we could adjust our territory sizes considering to generate more nondominated points from the desired regions. Specifically, we could define different territory sizes such that the smaller territories are constructed around the points generated from the desired region. This allows smaller coverage gap values in the regions desired by the DM. This way, we discriminate the nondominated points located in the desired regions.

An example is shown for the same bi-objective problem instance in Figure 4.13b where the desired region defined by the DM is shown by a rectangle. As can be seen, a smaller territory is constructed around the representative nondominated point A than the territory of point B.

To summarize, the preferences of the DM can be integrated in our algorithms by setting the objective weights and territory sizes accordingly. Therefore, our algorithms are convenient to be designed as interactive procedures for generating representative sets in MOMIPs.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, we present the results of our computational experiments performed on our algorithms (TSGA, TSGA-II) and three existing algorithms (DMA, SBA, TDA) that are described in Chapter 4. We also compare our results with the representative sets having the optimal cardinality for a given coverage gap value.

All experiments are conducted on randomly generated test instances of Multi-Objective Knapsack Problem (MOKP) and Multi-Objective Assignment Problem (MOAP) with three, four and five objectives (m = 3, 4, and 5). For each problem type and each m value, we consider different problem sizes (l) and we have a set of 10 instances for each problem size.

Additionally, in order to show that TSGA-II produces the same number of representative points with the optimal cardinality subsets, we also generate bi-objective (m = 2) knapsack and assignment problem instances. For each problem type, we have two different problem sizes (l) and a set of 5 instances for each problem size.

In MOKP experiments, we have the following problem sizes:

- For $m = 2$: 100 and 200 items (2MOKP100, 2MOKP200)
- For $m = 3$: 10, 20, 30, 40, 50, and 100 items (3MOKP10, 3MOKP20, 3MOKP30, 3MOKP40, 3MOKP50, 3MOKP100)
- For $m = 4$: 10, 20, 30, and 40 items (4MOKP10, 4MOKP20, 4MOKP30, 4MOKP40)
- For $m = 5$: 10 and 20 items (5MOKP10, 5MOKP20)

In MOAP experiments, we have the following problem sizes:

- For $m = 2$: 20 and 30 jobs (2MOAP20, 2MOAP30)
- For $m = 3$: 5, 10, 15, and 20 jobs (3MOAP5, 3MOAP10, 3MOAP15, 3MOAP20)
- For $m = 4$: 5, 10, and 15 jobs (4MOAP5, 4MOAP10, 4MOAP15)
- For $m = 5$: 5 and 10 jobs (5MOAP5, 5MOAP10)

We define MOKP and MOAP problems as given below:

Multi-Objective Knapsack Problem (MOKP):

$$\text{“}Max\text{”} \quad \{z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), ..., z_m(\boldsymbol{x})\}$$
$$s.to.$$
$$\sum_{j=1}^{l} w_j x_j \leq W$$
$$x_j \in \{0, 1\} \quad \forall\, j = 1, 2, ..., l$$

where

$$z_k(\boldsymbol{x}) = \sum_{j=1}^{l} c_{kj} x_j,$$

c_{kj}: Coefficient of item j in objective k,

w_j: Weight of item j in the knapsack,

W: Knapsack capacity,

x_j: Binary decision variable denoting whether item j is included in the knapsack.

For the knapsack problems with four and five objectives, we use test instances which are generated by Kirlik and Sayın (2014). For comparison purposes, all other knapsack test instances are generated as in Köksalan and Lokman (2009). The objective function coefficients and weight coefficients of the items are generated randomly from the discrete uniform distribution taking values between 10 and 100. Then, capacity of the knapsack is set to the half of the total of the weight coefficients of all items. To express mathematically,

$$W = \frac{\sum_{j=1}^{l} w_j}{2} \quad \text{where} \quad c_{kj}, w_j \in [10, 100].$$

Multi-Objective Assignment Problem (MOAP):

$$\text{“}Min\text{”} \quad \{z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), ..., z_m(\boldsymbol{x})\}$$

$s.to.$

$$\sum_{j=1}^{l} x_{ij} = 1 \quad \forall\, i = 1, 2, ..., l$$

$$\sum_{i=1}^{l} x_{ij} = 1 \quad \forall\, j = 1, 2, ..., l$$

$$x_{ij} \in \{0, 1\} \quad \forall\, i, j = 1, 2, ..., l$$

where

$$z_k(\boldsymbol{x}) = \sum_{i=1}^{l} \sum_{j=1}^{l} c_{kij} x_{ij},$$

c_{kij}: Coefficient for the cost of assignment of job i to worker j in objective k,

x_{ij}: Binary decision variable denoting whether job i is assigned to worker j or not.

Similar to the MOKP instances with two and three objectives, all MOAP test instances are generated just like in Köksalan and Lokman (2009) such that the assignment coefficients in objective functions are randomly generated from the Discrete Uniform distribution in the interval $[10, 100]$, i.e. $c_{kij} \in [10, 100]$.

The number of nondominated points for each of our MOKP and MOAP test instances are provided in Appendices. In order to generate all nondominated points of these problems, we used TSGA by setting the coverage gap value $\alpha = 0$ which makes the territory sizes equal to zero for all objectives.

Our algorithms TSGA and TSGA-II can also be implemented in Multi-Objective Mixed Integer Programs (MOMIPs). In order to show this, we made computational experiments with Mixed-Integer Knapsack test instances with three (3MIKP100), four (4MIKP40) and five objectives (5MIKP20). The models of these problems are same as the model **MOKP** defined above, except that half of the variables are binary variables and half of them are defined as continuous variables between [0, 1]. These instances are also generated as in Köksalan and Lokman (2009) such that $c_{kj}, w_j \in [10, 100]$ and $W = \dfrac{\sum_{j=1}^{l} w_j}{2}$.

As mentioned before, we decided to use the most recent search method developed by Dächert et. al. (2017) in all experiments performed for the five algorithms (DMA, SBA, TDA, TSGA, and TSGA-II), although SBA and TDA are originally designed with the subspace enumeration method of Lokman and Köksalan (2013). Dächert et. al. (2017) argue that their efficiency is based on avoiding to search redundant lower bounds whereas other existing approaches possibly include redundancies which makes them more complex. For further discussion, see Klamroth et. al. (2015) and Dächert et. al. (2017).

In our experiments, we generate representative sets for different coverage gap values on the interval [0, 1]. We use five different coverage gap values (α = 0.05, 0.10, 0.15, 0.20 and 0.25) which are defined as a percentage of the ranges of all objectives, where the range of an objective on the nondominated frontier of a maximization problem can be expressed as:

$$R_k = \max_{\boldsymbol{x} \in \boldsymbol{X}_E}(z_k(\boldsymbol{x})) - \min_{\boldsymbol{x} \in \boldsymbol{X}_E}(z_k(\boldsymbol{x}))$$

Since we generated all nondominated points of each of our MOIP test instances, we are able to calculate the range values by using the true ideal and nadir values in all objectives. However, in MOMIP experiments, since the true nadir point is not known, we use the payoff nadir point to calculate the range of each objective.

In addition to the algorithms, since we know all nondominated points of each problem instance, we solve the optimal cardinality model for each instance and α value:

Optimal Cardinality Problem:

$$Max \quad \sum_{i=1}^{n} r_i$$

$$s.to.$$

$$z_k^i - z_k^j t_{ij} - M(1 - t_{ij}) \leq \alpha R_k \quad \forall\, i, j = 1, 2, ..., n, \; \forall\, k = 1, 2, ..., m$$

$$\sum_{i=1}^{n} t_{ij} = 1 \quad \forall\, j = 1, 2, ..., n$$

$$\sum_{j=1}^{n} t_{ij} \leq M r_i \quad \forall\, i = 1, 2, ..., n$$

$$r_i \in \{0, 1\} \quad \forall\, i = 1, 2, ..., n$$

$$t_{ij} \in \{0, 1\} \quad \forall\, i, j = 1, 2, ..., n$$

where n is the number of nondominated points of the problem,
M is a sufficiently large number,
z_k^j : k^{th} objective function value of nondominated point j,
t_{ij}: Binary variable denoting whether nondominated point j is represented by point i,
r_i: Binary variable denoting whether nondominated point i is selected as a representative.

Using the source code of Dächert et. al. (2017), all algorithms are coded in C programming language by using the environment of Microsoft Visual Studio 2017 Professional. We run these algorithms on parallel computers with Intel(R)Core(TM)i7-477OS CPU @3.10 GHz, 16 GB RAM and Windows 10. As an optimization tool to solve our mathematical models, we use the callable library of IBM ILOG CPLEX 12.5. Lastly, we solve the optimal cardinality problems using the software General Algebraic Modeling System (GAMS) 23.9.5 and IBM ILOG CPLEX solver integrated in its portfolio.

In Appendices, we report the averages and standard deviations of the performance measures (cardinality, solution times and number of models solved) for all experiments conducted with five algorithms. Here, we summarize our results by graphical analyses. Firstly, in order to show the cardinality improvement of TSGA and TSGA-II for different problem sizes, we compare the average cardinalities for 3-objective knapsack problem instances in Figure 5.1. For the solution time improvement of TSGA, a similar graph is provided in Figure 5.2. In both graphs, the values corresponding to an algorithm are the averages of 10 test instances for each problem size.

Figure 5.1 shows that cardinality of our algorithms (TSGA and TSGA-II) are always smaller than the cardinality of the existing approaches (DMA, SBA, TDA) for all problem sizes. In addition, as the problem size increases, the gap between algorithms also increases which shows that the cardinality improvement becomes more significant in large-sized problems. Specifically, for 3MOKP100 instances, the average cardinality of TSGA is approximately 34% and 45% less than the average cardinalities of DMA and TDA, respectively. For the same instances, the average cardinality of TSGA-II is approximately 17% less than the average cardinality of TSGA.

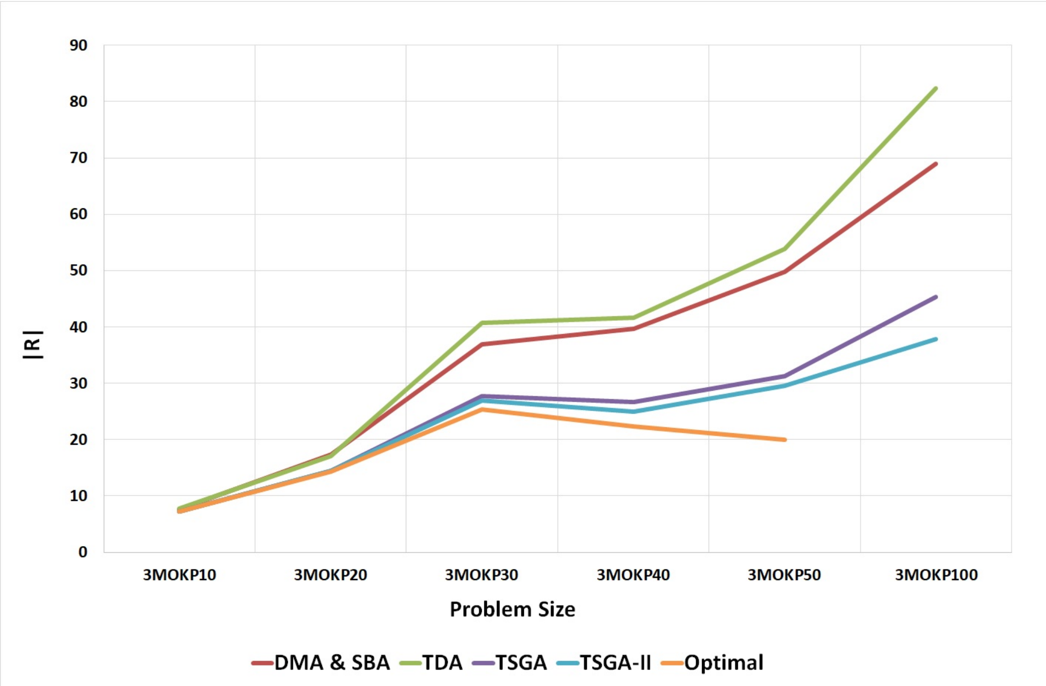


Figure 5.1: Cardinality comparison for different problem sizes of 3-objective knapsack problem (Average of 10 replications, $\alpha = 0.05$)

* Optimal cardinality model results could not be reported for 3MOKP100 instances since the models were not solvable within 12 hours.

Moreover, the average cardinality of TSGA-II is the closest to the optimal cardinality for all problem sizes. Particularly, for 3MOKP10 and 3MOKP20 instances, TSGA-II generates the same representative sets with the optimal cardinality model as can be seen in Figure 5.1.

Figure 5.2 shows that DMA is solved in excessive solution times especially in large-sized problems. In terms of computational efficiency, TSGA is the best approach among all other approaches. As the problem size increases, the solution time improvement of TSGA increases substantially. Although TDA solves a single model in each iteration and TSGA solves as many models as the number of subspaces, the average solution times of TSGA are less than the average solution times of TDA. For 3MOKP100 instances, the solution times of TSGA are approximately 41% and 46% lower than those of SBA and TDA on the average. Lastly, TSGA-II requires longer solution times than SBA and TDA due to the high number of models solved in each iteration. However, it provides a significant improvement in the cardinality as shown in Figure 5.1.

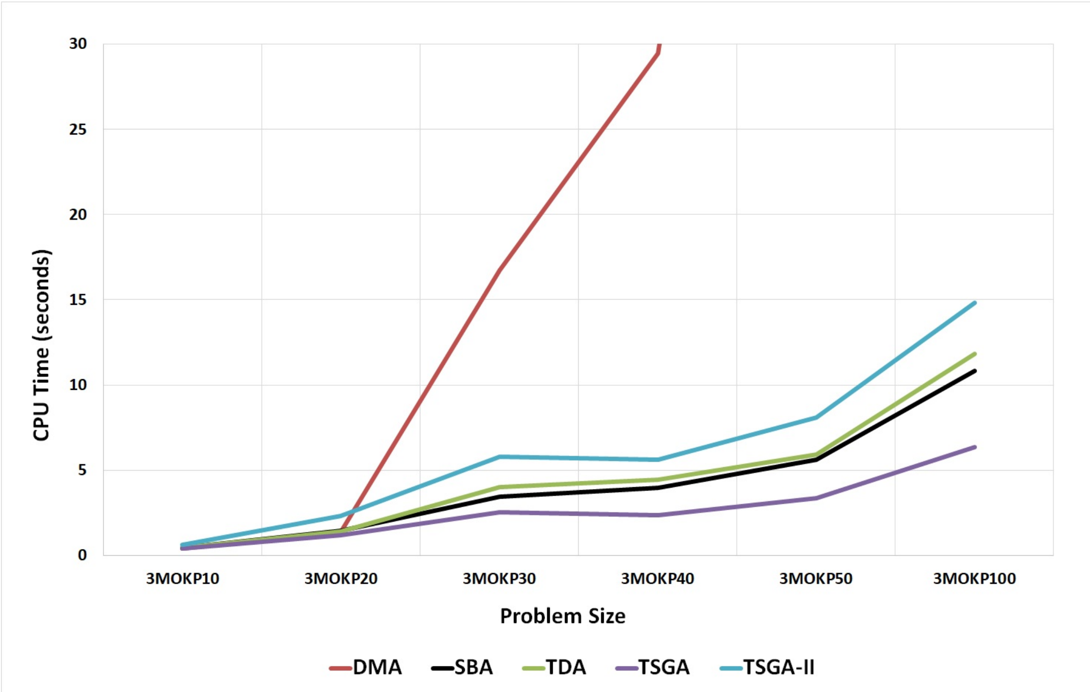


Figure 5.2: CPU time (secs) comparison for different problem sizes of 3-objective knapsack problem (Average of 10 replications, $\alpha = 0.05$)

In Figures 5.3 and 5.4, for a specific problem size, the cardinalities and solution times of each algorithm are compared for 3, 4 and 5-objective problems. The values plotted are the averages of 10 instances for 3MOKP20, 4MOKP20 and 5MOKP20 problems.

As can be seen in Figure 5.3, TSGA and TSGA-II generate representative sets with smaller cardinalities than DMA, SBA and TDA for all problem sizes. As the number of objectives increases, their cardinality improvements seem to be more significant. For 4MOKP20 instances, the average cardinality of TSGA is approximately 42% less than DMA and TDA. Moreover, the cardinalities of the representative sets generated by TSGA-II are always closer to the optimal cardinalities. On the average, the number of representative points generated by TSGA-II is approximately 7%, 10% and 16% higher than the optimal cardinalities for 3, 4 and 5-objective problems, respectively.

Figure 5.4 shows the time improvement of TSGA that increases as the number of objectives increases. For a knapsack problem with 20 items, as the number of objectives increases from three to four, the saving in solution times increases from approximately 33% to 56% with respect to SBA and TDA.

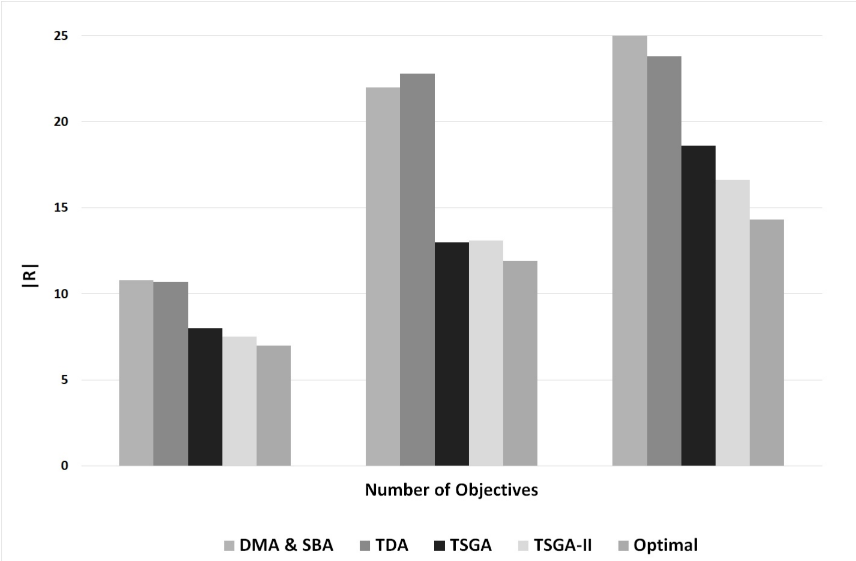


Figure 5.3: Cardinality comparison for 3, 4 and 5-objective knapsack problems (Average of 10 replications, $l = 20$, $\alpha = 0.10$)

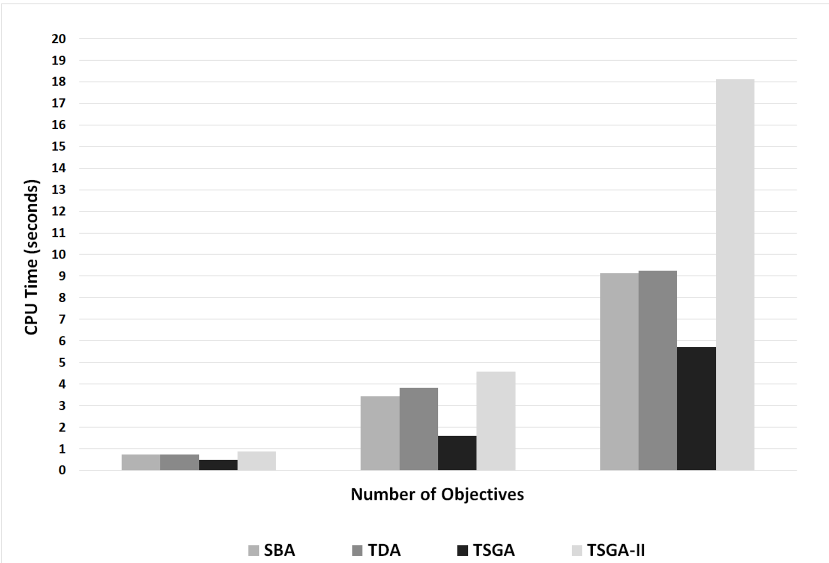


Figure 5.4: CPU time (secs) comparison for 3, 4 and 5-objective knapsack problems (Average of 10 replications, $l = 20$, $\alpha = 0.10$)

Similar results are obtained in our experiments with MOMIPs. As can be seen in Figures 5.5 and 5.6, TSGA and TSGA-II show better performances in MOMIPs. In terms of the average cardinality, TSGA generates approximately 45-55% less number of points than both SBA and TDA for all MOMIP instances. These cardinality improvements increases up to 65% for TSGA-II. (Since the problem sizes of 5MIKP20 instances are much smaller than 3MIKP100 and 4MIKP40 instances, the cardinality improvement of TSGA-II seems to be less significant for 5MIKP20 instances.) In terms of solution times, TSGA again outperforms all other approaches. Its solution times are approximately 57%, 73% and 65% less than those of SBA for 3MIKP100, 4MIKP40 and 5MIKP20 instances, respectively.

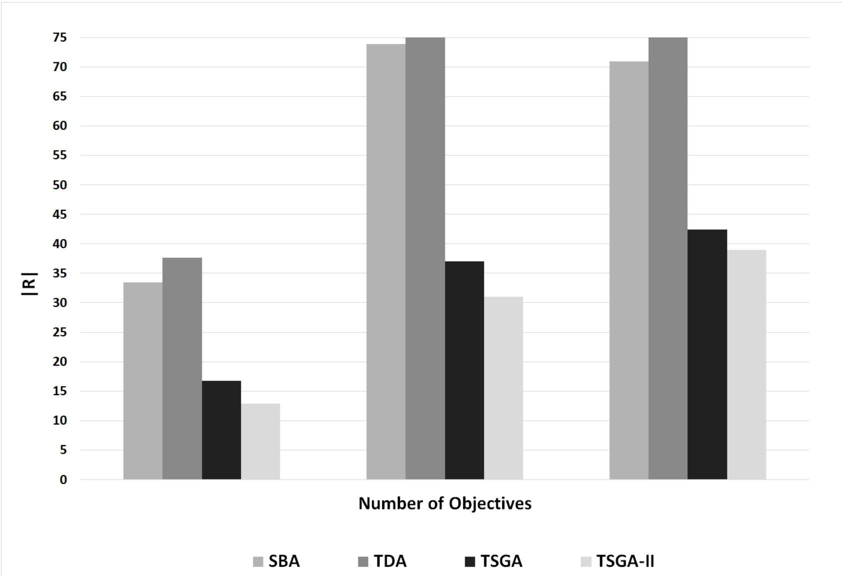


Figure 5.5: Cardinality comparison for 3, 4 and 5-objective mixed-integer knapsack problems (Average of 10 replications, $\alpha = 0.10$)

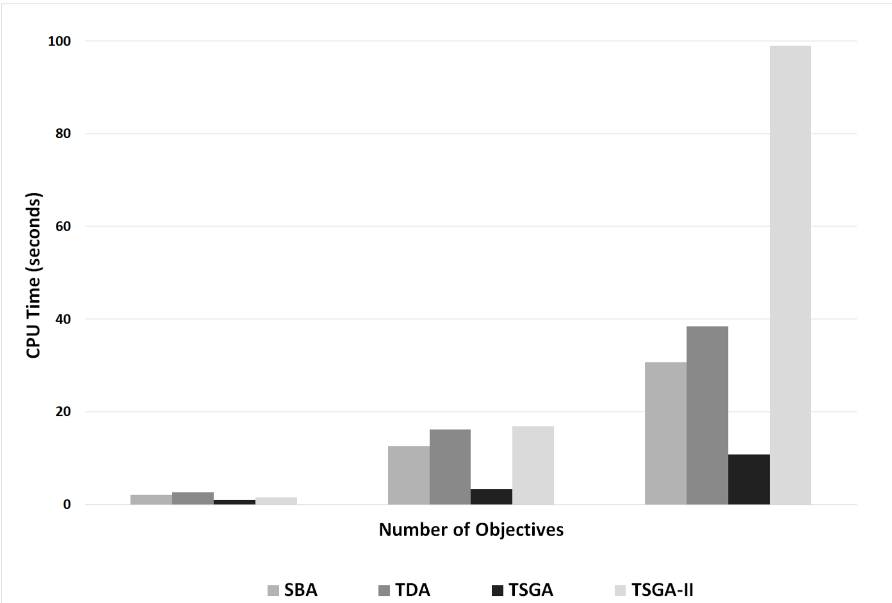


Figure 5.6: CPU time (secs) comparison for 3, 4 and 5-objective mixed-integer knapsack problems (Average of 10 replications, $\alpha = 0.10$)

As discussed in Chapter 4, TSGA and TSGA-II choose the objective weights based on the tangent plane to the L_q surface at the center of density. However, for the test instances used in our experiments, we observe that the calculated weights of objectives are approximately equal to each other. For these specific problem types, it can be concluded that we can generate points from the dense regions by assigning equal weights to all objectives without using the L_q surface. Based on this observation, we conducted additional experiments for TSGA and TSGA-II.

Results of the experiments conducted by using the original weights and equal weights are compared in Appendix F. For both TSGA and TSGA-II, the results of two variations are very similar to each other in terms of the cardinality of the generated representative sets, the number of models solved and the solution times. However, TSGA and TSGA-II use the problem-specific weights calculated based on the density distribution of the nondominated frontier of any type of MOMIP.

As shown in Section 4.3, TSGA-II generates the representative set with the minimum cardinality for a given coverage gap in bi-objective mixed-integer problems. In Table 5.1, the averages and standard deviations of the solution times of TSGA-II and the optimal cardinality model are provided for 2MOKP200 and 2MOAP30 test instances. As can be seen, for $\alpha = 0.10$, the solution times of TSGA-II are approximately 100% and 20% less than the solution times of the optimal cardinality model for the knapsack and assignment problems, respectively.

Table 5.1: CPU time (secs) comparison of TSGA-II and the optimal cardinality model for bi-objective problems

Problem	α	TSGA-II		Optimal	
		Avg.	StDev.	Avg.	StDev.
2MOKP200	0.05	0.69	0.06	633.37	708.08
	0.10	0.33	0.04	33.43	37.14
	0.15	0.20	0.09	26.61	29.00
	0.20	0.22	0.04	26.08	11.76
	0.25	0.17	0.07	8.88	4.68
2MOAP30	0.05	0.62	0.09	0.56	0.15
	0.10	0.36	0.07	0.46	0.11
	0.15	0.22	0.02	0.63	0.09
	0.20	0.18	0.06	0.37	0.16
	0.25	0.10	0.01	0.33	0.11

Table 5.1 also shows that as the coverage gap value decreases (and the number of generated points increases), the solution time improvement of TSGA-II increases significantly. In addition, since the problem sizes of 2MOKP200 instances are larger than the 2MOAP30 instances, we can conclude that the solution time improvement becomes more significant as the problem size grows. Especially for large-sized bi-objective mixed-integer problems, TSGA-II is very efficient since it generates the minimum cardinality representative sets much faster than the mathematical model.

Lastly, we test the sensitivity of TSGA-II to the objective function chosen randomly at the beginning of the algorithm. For this purpose, we conduct preliminary experiments with 3MOKP100 and 3MOAP20 test instances. For different coverage gap values, the averages and standard deviations of the cardinalities of the 10 instances for each problem type are reported in Table 5.2. Results indicate that different objectives chosen at the beginning of the algorithm does not affect the solution quality of TSGA-II significantly.

Table 5.2: The number of nondominated points generated for different objective functions chosen at the beginning of TSGA-II

Problem	α	z_1		z_2		z_3	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP100	0.05	37.80	6.55	38.10	7.85	36.70	6.80
	0.10	11.40	2.12	11.10	2.82	11.30	2.21
	0.15	5.70	1.06	5.30	1.16	5.40	0.70
	0.20	3.80	0.42	3.60	0.52	3.80	0.42
	0.25	3.00	0.00	3.00	0.47	3.00	0.00
3MOAP20	0.05	48.80	4.69	48.60	4.97	49.50	4.79
	0.10	14.00	1.63	14.00	1.63	13.80	0.92
	0.15	7.00	0.82	6.60	0.52	6.80	0.79
	0.20	3.90	0.57	4.20	0.63	3.70	0.68
	0.25	2.90	0.32	2.90	0.57	2.80	0.42

CHAPTER 6

CONCLUSIONS

In real-life multi-objective decision making problems, as the number of objectives and the problem size increase, the number of nondominated points increases substantially. In order to ease the decision making process of the DM and in order to decrease the computational effort, we prefer generating only a small subset of nondominated points. Considering the performance measures desired by the DM, presenting such a representative subset is more useful and practical. In this thesis, we develop two algorithms to generate representative sets of nondominated points in MOMIPs and we assess the quality of our sets by using the coverage gap and cardinality measures. For a prespecified coverage gap value, we show that our algorithms end up with less cardinality in a shorter amount of time than the existing approaches in the literature.

Our first approach, TSGA, is developed based on the observations on the density distribution and shape of the nondominated frontier. Instead of selecting the most diverse representative points as in DMA and SBA, we find out that if the representatives are selected closer to each other and located in the dense regions of the frontier, then more points could be represented with less number of representatives. In order to achieve this, we iteratively solve a weighted sum problem in our algorithms. Specifically, we observe that the nondominated points with better weighted sum values are able to represent more nondominated points. Furthermore, in order to guarantee the desired coverage gap value, we define territories around each generated point and search for the next representative after excluding these territories from the search space. Our experiments show that TSGA satisfies a given coverage gap value with less number of representatives and in a shorter time than the existing approaches.

Our second algorithm, TSGA-II, yields better results than TSGA in terms of the cardinality of the generated representative sets. Although TSGA-II requires longer solution times than the other approaches, it outperforms all of them in terms of the solution quality. In TSGA-II, we again define territories around each generated point and iteratively solve a weighted sum problem in the territory-excluded search space. TSGA-II is designed so that we can reduce one dimension of the origional problem. Specifically, for an m-objective problem, we optimize the weighted sum of $(m-1)$ objectives excluding a randomly selected one. The territories and subspaces are also defined for $(m-1)$ objectives throughout the solution process. Since we work on an $(m-1)$-dimensional objective space, the generated points are able to represent more nondominated points in the territory-excluded regions. As a result, for a given coverage gap value, TSGA-II represents the nondominated frontier of a MOMIP by a smaller number of representatives than the existing approaches. This way, it provides a significant improvement in the solution quality especially in large-sized problems.

For both TSGA and TSGA-II, we provide some application alternatives such that the DM can incorporate in the weight selection process. The DM can either define some preferences among objectives or specify an indifference range for each objective. In both cases, we can define the weight of each objective as inversely proportional to the territory size. Such an interactive process would be beneficial for presenting representative sets that are more suitable to the preferences of the DM.

We conduct our computational experiments with the randomly generated MOKP, MOAP and mixed-integer knapsack problem test instances with three, four and five objectives. Then, we compare our results with the results of the existing approaches in the literature that are called as DMA, SBA and TDA. To make a fair comparison, we implement all algorithms by using the decomposition method recently developed by Dächert et. al. (2017). These comparisons are made in terms of the cardinality of the generated representative sets, the solution times and the number of models solved by each algorithm. Our results show that TSGA and TSGA-II work well and outperform the existing approaches in terms of both the solution qualities and the solution times.

As future research, some interactive and problem specific solution strategies can be designed for TSGA and TSGA-II. The preferences of the DM can be incorporated throughout the solution process. The coverage gap value can be altered in each iteration based on the region where the generated nondominated point is located. Furthermore, in order to decrease our solution times (especially for TSGA-II), instead of searching all subspaces in each iteration, the representative points may be generated only from the specific regions desired by the DM. Lastly, an interesting work may be changing the structure of the territories constructed around each representative point. Especially for problems with more than two objectives, these territories may be defined so that more subspaces could be excluded from the search space in each iteration resulting in higher computational efficiency.

REFERENCES

[1] C. Bazgan, F. Jamain, and D. Vanderpooten. Approximate Pareto sets of minimal size for multi-objective optimization problems. *Operations Research Letters*, 43(1), 2015.

[2] N. Boland, H. Charkhgard, and M. Savelsbergh. A criterion space search algorithm for biobjective integer programming: The balanced box method. *INFORMS Journal on Computing*, 27(4):735–754, 2015.

[3] N. Boland, H. Charkhgard, and M. W. P. Savelsbergh. The L-shape search method for triobjective integer programming. *Math. Program. Comput.*, 8:217–251, 2016.

[4] G. Ceyhan, M. Köksalan, and B. Lokman. Finding a representative nondominated set for multi-objective mixed integer programs. *European Journal of Operational Research*, 2018.

[5] K. Dächert, K. Klamroth, R. Lacour, and D. Vanderpooten. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research*, 260(3):841–855, 2017.

[6] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multi-objective combinatorial optimization. *OR-Spektrum*, 22(4):425–460, 2000.

[7] C. Filippi and E. Stevanato. Approximation schemes for bi-objective combinatorial optimization and their application to the tsp with profits. *Computers & OR*, 40:2418–2428, 2013.

[8] G. Kirlik and S. Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3):479–488, 2014.

[9] K. Klamroth, R. Lacour, and D. Vanderpooten. On the representation of the search region in multi-objective optimization. *European Journal of Operational Research*, 245(3):767–778, 2015.

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

[11] B. Lokman and M. Köksalan. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57(2):347–365, 2013.

[12] M. Masin and Y. Bukchin. Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2):411–424, 2008.

[13] S. S. Özarık. Issues in selecting a representative set for multi-objective integer programs. Master's thesis, Middle East Technical University, Graduate School of Natural and Applied Science, Ankara, 2017.

[14] M. Özlen and M. Azizoğlu. Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1):25 – 35, 2009.

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

[16] S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.

[17] L. Shao and M. Ehrgott. Discrete representation of non-dominated sets in multi-objective linear programming. *European Journal of Operational Research*, 255(3):687–698, 2016.

[18] 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(3):1011–1027, 2007.

[19] S. Vassilvitskii and M. Yannakakis. Efficiently computing succinct trade-off curves. *Theoretical Computer Science*, 348(2):334–356, 2005.

[20] D. Vaz, L. Paquete, C. M. Fonseca, K. Klamroth, and M. Stiglmayr. Representation of the non-dominated set in biobjective discrete optimization. *Computers & Operations Research*, 63:172–186, 2015.

APPENDIX A

LISTS OF THE GENERATED MOKP AND MOAP TEST INSTANCES (ALL NONDOMINATED POINTS)

Table A.1: MOKP and MOAP Instances with $m = 2$

	2MOKP100	2MOKP200	2MOAP20	2MOAP30
1	85	296	69	89
2	172	417	56	80
3	94	336	56	88
4	112	431	53	80
5	119	408	65	77

Table A.2: MOKP Instances with $m = 3$

	3MOKP10	3MOKP20	3MOKP30	3MOKP40	3MOKP50	3MOKP100
1	6	54	125	245	221	3523
2	10	98	83	168	674	3114
3	14	57	198	175	750	2714
4	2	13	65	112	346	4773
5	6	25	206	212	736	2433
6	15	43	65	252	537	7203
7	3	37	88	269	379	3307
8	8	14	113	349	526	3062
9	9	17	147	90	127	4355
10	20	16	289	420	913	3198

Table A.3: MOAP Instances with $m = 3$

	3MOAP5	3MOAP10	3MOAP15	3MOAP20
1	20	104	578	1970
2	21	135	650	1247
3	20	169	940	1806
4	28	193	362	2150
5	18	270	640	2246
6	13	183	664	2813
7	15	249	554	1825
8	19	125	801	1591
9	6	325	299	1916
10	20	141	597	1521

Table A.4: MOKP Instances with $m = 4$ and $m = 5$

	4MOKP10	4MOKP20	4MOKP30	4MOKP40	5MOKP10	5MOKP20
1	10	60	563	901	20	220
2	13	143	535	5018	19	95
3	7	325	226	508	23	76
4	6	17	517	1248	28	89
5	9	74	281	2351	13	110
6	19	152	191	1920	9	87
7	15	175	480	1435	9	61
8	5	116	262	741	25	211
9	22	77	186	3409	6	237
10	10	229	735	555	10	426

Table A.5: MOAP Instances with $m = 4$ and $m = 5$

	4MOAP5	4MOAP10	4MOAP15	5MOAP5	5MOAP10
1	50	657	9463	37	3647
2	34	435	6254	42	2838
3	60	789	6032	42	3363
4	10	7099	4657	61	4832
5	45	1441	5748	48	2195
6	36	789	9921	50	4839
7	31	1234	8412	57	2622
8	24	671	4248	59	2775
9	40	1311	4437	60	3502
10	45	844	4346	44	3069

APPENDIX B

CARDINALITY COMPARISON FOR DMA, SBA, TDA, TSGA, TSGA-II AND THE OPTIMAL SUBSETS

Table B.1: Cardinality Comparison for MOKP ($m = 3$)

			DMA		SBA		TDA		TSGA		TSGA-II		Optimal*	
Problem	$\|\mathbf{Z}_{ND}\|$	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP10	9.30	0.05	17.60	8.57	17.40	8.78	17.10	8.65	14.40	6.28	14.40	6.33	14.30	6.29
		0.10	10.80	4.10	10.80	4.10	10.70	4.11	8.00	2.87	7.50	2.46	7.00	2.11
		0.15	8.10	3.11	8.10	3.11	7.90	2.92	5.30	1.77	4.80	1.48	4.60	1.43
		0.20	5.90	2.18	5.90	2.18	5.70	1.70	3.60	0.97	3.40	1.07	3.20	0.79
		0.25	4.20	1.62	4.20	1.62	4.20	1.55	2.80	0.79	2.70	0.67	2.50	0.53
3MOKP20	37.40	0.05	36.80	7.07	36.90	7.08	40.70	9.10	27.70	5.66	27.00	4.90	25.30	4.52
		0.10	19.00	3.06	19.00	3.06	19.40	3.72	12.20	2.25	11.70	1.89	9.80	1.32
		0.15	12.50	2.51	12.50	2.51	11.90	2.23	6.80	1.14	6.60	1.17	5.40	0.70
		0.20	8.80	1.32	8.80	1.32	8.60	1.78	4.90	0.88	4.00	0.82	3.50	0.53
		0.25	5.60	1.17	5.60	1.17	5.60	0.84	3.30	0.48	3.00	0.67	2.50	0.53
3MOKP30	137.90	0.05	36.80	7.07	36.90	7.08	40.70	9.10	27.60	5.58	27.00	4.81	25.30	4.52
		0.10	19.00	3.06	19.00	3.06	19.40	3.72	12.00	2.36	11.50	1.96	9.80	1.32
		0.15	12.50	2.51	12.50	2.51	11.90	2.23	7.10	1.29	6.60	1.07	5.40	0.70
		0.20	8.80	1.32	8.80	1.32	8.60	1.78	4.50	0.97	3.80	0.79	3.50	0.53
		0.25	5.60	1.17	5.60	1.17	5.60	0.84	3.10	0.74	3.00	0.67	2.50	0.53
3MOKP40	229.20	0.05	39.90	9.83	39.70	9.75	41.70	12.81	26.70	8.67	25.00	8.47	22.30	7.35
		0.10	20.60	4.45	20.60	4.45	20.40	3.84	10.10	2.42	9.40	1.96	7.50	2.01
		0.15	12.80	2.25	12.80	2.25	12.50	2.68	5.70	1.25	5.10	1.29	4.50	1.08
		0.20	8.60	2.59	8.60	2.59	8.30	2.26	3.90	0.74	3.70	0.48	3.00	0.67
		0.25	5.00	1.83	5.00	1.83	4.90	1.79	3.00	1.15	2.60	0.70	2.30	0.48
3MOKP50	520.90	0.05	50.10	13.47	49.80	13.16	53.80	15.21	31.30	9.17	29.50	7.43	20.00	4.24
		0.10	22.80	5.90	22.70	5.85	22.80	6.41	11.30	3.43	9.90	2.81	6.75	0.96
		0.15	13.60	4.30	13.60	4.30	13.60	4.22	6.20	1.93	5.40	1.71	3.25	0.50
		0.20	9.40	2.55	9.40	2.55	9.60	2.80	4.10	1.20	3.40	0.52	2.50	0.58
		0.25	6.00	1.49	6.00	1.49	6.00	1.05	3.00	0.67	2.80	0.42	2.00	0.00
3MOKP100	3768.20	0.05	68.90	10.10	69.00	10.45	82.30	10.77	45.30	8.97	37.80	6.55	-	-
		0.10	30.20	4.69	30.20	4.69	31.80	4.59	13.30	3.33	11.40	2.12	-	-
		0.15	19.10	2.38	19.10	2.38	20.00	2.87	6.80	1.40	5.70	1.06	-	-
		0.20	12.80	1.75	12.80	1.75	13.10	2.18	4.30	0.67	3.80	0.42	-	-
		0.25	7.90	1.66	7.80	1.62	7.90	1.37	3.90	0.32	3.00	0.00	-	-

* Results for 3MOKP100 instances are not reported since the models were not solvable within 4 hours.

Table B.2: Cardinality Comparison for MOKP ($m = 4$)

Problem	$\|\mathbf{Z}_{ND}\|$	α	DMA*		SBA		TDA		TSGA		TSGA-II		Optimal**	
			Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOKP10	11.60	0.05	8.50	3.95	8.50	3.95	8.40	4.06	7.90	3.84	7.80	3.65	7.80	3.65
		0.10	6.70	3.47	6.70	3.47	6.50	3.17	5.70	2.45	5.50	2.46	5.50	2.46
		0.15	5.20	2.35	5.20	2.35	5.40	2.50	4.50	1.51	4.40	1.58	4.20	1.14
		0.20	3.90	1.91	3.90	1.91	4.00	1.89	3.40	1.71	3.30	1.42	3.10	1.52
		0.25	2.90	1.20	2.90	1.20	3.00	1.15	2.90	0.99	2.80	0.92	2.40	0.84
4MOKP20	136.80	0.05	43.30	19.93	43.30	19.93	43.60	21.00	32.80	14.99	33.10	15.55	31.80	14.35
		0.10	22.00	8.45	22.00	8.45	22.80	9.31	13.00	4.69	13.10	4.61	11.90	4.31
		0.15	13.70	5.68	13.70	5.68	14.40	6.67	7.60	3.47	7.40	2.37	6.40	2.07
		0.20	8.60	2.91	8.60	2.91	8.40	3.37	5.00	1.76	4.30	1.34	4.00	1.15
		0.25	5.70	1.95	5.70	1.95	5.30	1.95	3.30	1.06	3.20	1.03	2.80	0.79
4MOKP30	397.60	0.05	75.60	22.47	76.10	22.73	82.60	25.75	54.10	15.98	52.60	14.66	48.00	12.46
		0.10	33.30	7.04	33.70	7.45	34.80	8.84	17.60	4.38	17.50	3.17	13.60	2.95
		0.15	18.60	4.27	18.70	4.24	19.00	3.80	9.00	1.76	8.40	1.65	6.38	1.30
		0.20	10.80	2.74	10.90	2.77	11.50	3.10	5.60	1.35	5.00	1.15	3.20	0.84
		0.25	6.60	1.71	6.60	1.71	7.00	1.76	3.80	0.92	3.10	0.88	2.40	0.55
4MOKP40	1808.60	0.05	86.25	9.95	120.20	37.93	128.50	42.84	76.30	25.36	72.00	23.82	39.50	5.26
		0.10	44.20	8.16	44.20	8.16	44.90	14.13	20.50	6.40	18.80	5.14	10.00	1.00
		0.15	23.00	4.40	23.00	4.40	22.90	4.65	10.40	3.06	8.00	2.49	-	-
		0.20	13.30	2.63	13.30	2.63	13.50	2.80	6.00	1.41	5.20	0.92	-	-
		0.25	7.80	2.62	7.80	2.62	7.90	2.28	3.70	0.82	3.30	0.82	-	-

* Fewer than 10 instances per cell are reported for 4MOKP40 since DMA could not be finalized within 12 hours.
** Results for 4MOKP40 instances ($\alpha \geq 0.15$) are not reported since the models were not solvable within 4 hours.

Table B.3: Cardinality Comparison for MOKP ($m = 5$)

Problem	$\|\mathbf{Z}_{ND}\|$	α	DMA		SBA		TDA		TSGA		TSGA-II		Optimal	
			Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOKP10	16.20	0.05	14.30	6.09	14.30	6.09	14.10	6.05	13.70	5.42	13.70	5.42	13.70	5.42
		0.10	10.50	3.84	10.50	3.84	10.70	4.08	9.30	3.47	9.30	3.47	9.30	3.47
		0.15	8.20	2.66	8.20	2.66	8.40	2.91	7.20	1.93	7.00	2.11	6.90	2.13
		0.20	6.80	2.20	6.80	2.20	6.60	2.27	5.70	1.42	5.30	1.64	5.20	1.48
		0.25	4.60	1.51	4.60	1.51	4.40	1.35	4.70	1.06	4.30	1.25	3.90	0.88
5MOKP20	161.20	0.05	55.00	24.78	55.20	24.99	54.40	24.80	46.60	20.71	46.50	20.94	44.40	19.69
		0.10	25.00	8.49	25.00	8.49	23.80	8.31	18.60	7.21	16.60	5.87	14.30	4.81
		0.15	14.90	3.90	14.90	3.90	12.90	4.53	9.40	2.41	8.80	2.10	7.00	2.00
		0.20	8.70	2.36	8.70	2.36	8.00	2.40	4.90	2.18	4.60	1.84	3.80	0.79
		0.25	6.40	2.27	6.30	2.11	6.30	2.36	3.60	1.65	3.50	1.08	2.40	0.52

Table B.4: Cardinality Comparison for MOAP ($m = 3$)

			DMA		SBA		TDA		TSGA		TSGA-II		Optimal*	
Problem	$\|\mathbf{Z}_{ND}\|$	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOAP5	18.00	0.05	14.20	3.88	14.10	3.98	14.30	4.22	13.30	4.11	13.10	4.15	13.10	4.15
		0.10	10.70	3.30	10.60	3.27	11.10	3.70	9.40	3.37	8.90	2.88	8.80	2.82
		0.15	8.30	2.45	8.20	2.44	8.30	2.71	6.80	1.75	6.70	1.77	6.60	1.78
		0.20	6.10	1.91	5.90	1.97	5.80	1.99	4.90	1.45	4.50	1.51	4.30	1.34
		0.25	4.60	1.90	4.40	1.90	4.50	1.96	4.00	1.63	3.50	1.08	3.10	0.99
3MOAP10	189.40	0.05	47.50	10.23	48.20	10.30	53.90	13.03	34.60	6.10	33.40	6.52	31.40	5.32
		0.10	22.30	4.40	22.40	4.22	23.80	3.94	14.00	3.65	12.30	1.83	10.30	1.83
		0.15	14.70	2.16	14.90	1.97	14.90	1.60	7.20	0.92	7.10	1.20	5.67	0.87
		0.20	9.30	1.64	9.20	1.62	9.10	1.97	5.10	1.52	4.10	0.88	3.56	0.53
		0.25	6.10	1.20	5.90	1.20	5.70	1.16	3.30	0.67	2.90	0.74	2.44	0.53
3MOAP15	608.50	0.05	62.90	10.06	61.50	10.41	68.40	13.49	39.30	9.87	37.20	6.63	26.00	3.83
		0.10	28.00	2.94	27.00	3.40	28.40	3.92	14.20	2.20	11.80	2.15	-	-
		0.15	16.30	2.79	16.10	2.69	17.10	2.13	7.30	1.57	5.90	0.99	-	-
		0.20	10.50	2.07	10.20	2.04	10.60	2.22	4.90	0.74	3.90	0.74	-	-
		0.25	6.00	1.25	6.00	1.25	6.10	1.29	3.00	0.67	2.80	0.42	-	-
3MOAP20	1908.50	0.05	78.60	6.31	78.80	7.74	92.40	7.24	56.70	6.29	48.80	4.69	-	-
		0.10	31.40	3.13	31.60	3.13	34.90	3.18	16.90	2.38	14.00	1.63	-	-
		0.15	21.20	3.22	21.50	3.44	21.40	3.27	8.60	1.17	7.00	0.82	-	-
		0.20	13.50	1.90	13.50	2.01	13.60	2.01	4.80	0.79	3.90	0.57	-	-
		0.25	8.60	1.71	8.50	1.65	8.50	1.35	3.70	0.67	2.90	0.32	-	-

* Fewer than 10 instances per cell are reported for 3MOAP15 ($\alpha = 0.05$). Other results are not reported for 3MOAP15 and 3MOAP20 instances since the models were not solvable within 4 hours.

Table B.5: Cardinality Comparison for MOAP ($m = 4$)

			DMA*		SBA		TDA		TSGA		TSGA-II		Optimal**	
Problem	$\|\mathbf{Z}_{ND}\|$	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOAP5	37.50	0.05	27.10	10.71	27.10	10.58	27.30	10.63	27.00	10.46	26.70	10.44	26.50	10.27
		0.10	19.20	8.07	19.10	8.16	19.30	7.73	16.90	6.14	16.30	6.40	16.00	6.38
		0.15	13.10	5.74	13.30	5.87	13.10	5.11	10.70	3.62	9.70	3.47	9.40	3.37
		0.20	9.40	3.98	9.40	4.01	9.60	4.22	7.80	2.53	7.10	2.42	5.70	1.83
		0.25	6.70	2.83	6.80	3.05	6.70	2.83	5.50	1.43	4.90	1.45	3.70	0.95
4MOAP10	1527.00	0.05	149.60	38.38	145.40	35.80	162.30	35.46	104.30	26.06	100.40	25.07	75.33	16.52
		0.10	51.30	9.96	50.90	9.21	55.20	11.08	29.50	5.91	26.50	6.13	-	-
		0.15	27.20	4.96	26.80	4.83	27.60	4.86	14.50	2.59	12.20	3.12	-	-
		0.20	15.50	3.37	15.30	3.68	15.90	3.31	8.00	1.25	6.70	1.49	-	-
		0.25	9.50	1.35	9.30	1.25	9.10	1.10	5.10	0.88	4.10	0.88	-	-
4MOAP15	6351.80	0.05	-	-	233.40	84.84	272.90	93.70	147.30	57.22	134.00	52.74	-	-
		0.10	68.30	23.73	69.60	23.56	76.40	27.25	33.50	12.20	29.40	10.74	-	-
		0.15	34.20	12.43	35.00	12.43	35.90	12.32	13.00	4.37	11.20	3.55	-	-
		0.20	19.50	7.41	19.70	7.75	20.60	7.66	7.50	2.68	6.20	1.99	-	-
		0.25	11.00	3.94	11.00	4.00	11.70	4.35	4.90	1.60	3.80	1.14	-	-

* Results are not reported for 4MOAP15 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.
** Fewer than 10 instances are reported for 4MOAP10 ($\alpha = 0.05$). Other results for 4MOAP10 and 4MOAP15 instances are not reported since the models were not solvable within 4 hours.

Table B.6: Cardinality Comparison for MOAP ($m = 5$)

Problem	$\lvert \mathbf{Z}_{ND} \rvert$	α	DMA*		SBA		TDA		TSGA		TSGA-II		Optimal**	
			Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOAP5	50.00	0.05	36.90	5.99	36.60	5.72	36.70	5.62	35.80	5.88	35.70	5.89	35.30	5.50
		0.10	21.90	4.86	22.10	4.82	23.30	5.29	21.00	3.83	20.30	3.74	20.00	3.37
		0.15	14.50	3.63	14.70	3.62	14.90	2.81	12.80	2.30	11.70	1.77	11.90	2.51
		0.20	10.60	2.27	10.30	2.41	11.20	2.39	8.30	1.49	8.00	1.49	7.10	1.45
		0.25	8.10	1.85	7.60	1.43	8.00	1.56	6.30	1.06	5.60	1.43	4.90	0.57
5MOAP10	3368.20	0.05	-	-	371.30	81.15	417.20	92.84	279.70	62.35	272.20	62.55	-	-
		0.10	96.00	16.32	99.00	16.93	108.90	18.93	60.60	10.43	56.60	12.84	-	-
		0.15	43.40	6.26	43.40	6.50	46.10	8.40	23.40	3.75	20.90	4.04	-	-
		0.20	24.10	5.70	24.10	6.06	25.60	6.40	12.30	2.58	10.90	1.37	-	-
		0.25	13.70	3.74	14.00	3.94	15.00	3.77	8.00	1.63	7.00	1.15	-	-

* Results are not reported for 5MOAP10 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.
** Results are not reported for 5MOAP10 instances since the models were not solvable within 4 hours.

APPENDIX C

CPU TIME COMPARISON FOR DMA, SBA, TDA, TSGA AND TSGA-II

Table C.1: CPU Time (secs) Comparison for MOKP ($m = 3$)

		DMA		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP10	0.05	0.39	0.20	0.43	0.25	0.44	0.27	0.42	0.23	0.64	0.45
	0.10	0.30	0.12	0.34	0.22	0.32	0.17	0.32	0.15	0.44	0.26
	0.15	0.24	0.07	0.22	0.10	0.25	0.13	0.25	0.10	0.32	0.16
	0.20	0.22	0.06	0.17	0.06	0.19	0.08	0.18	0.05	0.21	0.07
	0.25	0.20	0.06	0.15	0.06	0.17	0.08	0.15	0.03	0.19	0.06
3MOKP20	0.05	1.37	0.97	1.44	0.79	1.42	0.86	1.20	0.71	2.30	1.70
	0.10	0.67	0.34	0.74	0.30	0.74	0.32	0.49	0.22	0.86	0.47
	0.15	0.48	0.23	0.55	0.25	0.47	0.19	0.31	0.16	0.46	0.24
	0.20	0.34	0.15	0.36	0.18	0.32	0.13	0.20	0.08	0.29	0.14
	0.25	0.25	0.12	0.25	0.12	0.22	0.11	0.15	0.05	0.21	0.08
3MOKP30	0.05	16.73	19.80	3.45	0.94	4.01	1.34	2.54	0.74	5.77	1.91
	0.10	1.98	0.77	1.48	0.31	1.41	0.39	0.86	0.21	1.69	0.50
	0.15	0.90	0.24	0.85	0.21	0.73	0.14	0.45	0.12	0.77	0.19
	0.20	0.58	0.11	0.61	0.08	0.51	0.09	0.28	0.05	0.38	0.12
	0.25	0.35	0.09	0.36	0.07	0.29	0.05	0.19	0.04	0.28	0.11
3MOKP40	0.05	29.47	29.00	3.97	1.26	4.43	1.97	2.37	0.95	5.61	2.73
	0.10	3.17	2.25	1.69	0.48	1.60	0.43	0.73	0.24	1.36	0.51
	0.15	1.07	0.40	0.97	0.17	0.85	0.20	0.37	0.12	0.59	0.21
	0.20	0.58	0.21	0.61	0.18	0.52	0.14	0.23	0.04	0.35	0.07
	0.25	0.32	0.14	0.34	0.14	0.29	0.12	0.16	0.07	0.23	0.09
3MOKP50	0.05	89.75	82.06	5.61	1.88	5.92	2.22	3.37	1.20	8.11	3.28
	0.10	4.70	2.43	2.18	0.74	1.95	0.75	0.92	0.30	1.63	0.73
	0.15	1.45	0.80	1.15	0.39	0.92	0.29	0.43	0.15	0.74	0.32
	0.20	0.77	0.31	0.72	0.17	0.65	0.20	0.29	0.13	0.33	0.11
	0.25	0.46	0.18	0.48	0.12	0.38	0.07	0.20	0.06	0.31	0.08
3MOKP100	0.05	1029.92	1040.06	10.83	2.45	11.82	2.13	6.37	1.84	14.84	4.35
	0.10	11.74	3.83	3.64	0.50	3.38	0.58	1.35	0.43	2.35	0.69
	0.15	3.69	0.95	1.99	0.17	1.76	0.30	0.55	0.16	1.00	0.30
	0.20	1.62	0.51	1.23	0.15	1.04	0.17	0.35	0.05	0.50	0.08
	0.25	0.77	0.23	0.73	0.16	0.59	0.10	0.37	0.11	0.40	0.04

Table C.2: CPU Time (secs) Comparison for MOKP ($m = 4$)

		DMA*		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOKP10	0.05	0.45	0.23	0.78	0.48	0.67	0.45	0.73	0.34	1.46	0.82
	0.10	0.32	0.16	0.52	0.38	0.46	0.30	0.52	0.19	0.85	0.43
	0.15	0.25	0.11	0.35	0.26	0.35	0.23	0.41	0.12	0.65	0.24
	0.20	0.19	0.08	0.21	0.12	0.24	0.13	0.34	0.12	0.46	0.13
	0.25	0.15	0.05	0.14	0.07	0.16	0.07	0.30	0.11	0.38	0.13
4MOKP20	0.05	22.49	31.58	8.87	5.20	10.12	7.27	6.75	4.46	30.86	26.06
	0.10	2.33	1.94	3.43	1.88	3.82	2.60	1.59	0.81	4.58	2.67
	0.15	0.97	0.51	1.68	0.95	1.82	1.30	0.77	0.50	1.71	0.83
	0.20	0.51	0.18	0.88	0.36	0.84	0.52	0.40	0.20	0.76	0.38
	0.25	0.33	0.12	0.51	0.21	0.44	0.21	0.24	0.08	0.43	0.21
4MOKP30	0.05	375.46	469.28	18.65	6.73	25.73	11.79	13.40	5.22	56.08	26.82
	0.10	11.38	8.27	6.78	3.27	6.77	3.05	2.53	0.91	7.88	3.51
	0.15	2.15	1.19	18.26	26.98	2.73	1.03	0.85	0.25	2.06	0.81
	0.20	0.87	0.38	34.44	39.77	1.28	0.50	0.44	0.11	0.87	0.34
	0.25	0.47	0.15	0.77	0.37	0.67	0.26	0.28	0.07	0.41	0.16
4MOKP40	0.05	1987.14	2479.89	40.98	20.42	51.55	27.09	21.20	11.36	121.68	69.42
	0.10	90.30	82.17	10.25	3.19	10.68	5.89	3.13	1.59	10.07	5.38
	0.15	6.40	4.03	3.77	0.98	3.45	1.22	1.22	0.53	2.35	1.29
	0.20	1.61	0.70	1.77	0.47	1.64	0.68	0.52	0.16	0.98	0.32
	0.25	0.70	0.38	44.80	16.72	0.74	0.32	0.31	0.11	0.47	0.20

* Fewer than 10 instances per cell are reported for 4MOKP40 since DMA could not be finalized within 12 hours.

Table C.3: CPU Time (secs) Comparison for MOKP ($m = 5$)

		DMA		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOKP10	0.05	0.87	0.50	2.92	2.07	2.49	1.66	2.42	1.57	7.50	6.15
	0.10	0.51	0.24	1.65	0.89	1.51	0.81	1.33	0.67	3.20	2.07
	0.15	0.39	0.16	1.05	0.60	0.96	0.52	0.96	0.42	1.79	0.96
	0.20	0.32	0.12	0.76	0.37	0.68	0.32	0.61	0.20	1.06	0.62
	0.25	0.21	0.07	0.38	0.16	0.34	0.15	0.44	0.16	0.65	0.26
5MOKP20	0.05	45.74	61.82	31.81	27.10	37.56	37.39	28.68	28.97	177.36	181.65
	0.10	4.30	4.30	9.13	7.48	9.25	7.89	5.73	5.31	18.12	15.77
	0.15	1.23	0.69	3.46	2.29	2.69	2.35	1.51	0.98	3.97	2.53
	0.20	0.55	0.24	1.48	0.66	1.24	0.63	0.65	0.53	1.28	1.01
	0.25	0.38	0.16	0.87	0.44	0.80	0.50	0.41	0.27	0.69	0.32

Table C.4: CPU Time (secs) Comparison for MOAP ($m = 3$)

Problem	α	DMA		SBA		TDA		TSGA		TSGA-II	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOAP5	0.05	0.73	0.26	0.62	0.20	0.70	0.24	0.53	0.18	1.10	0.45
	0.10	0.48	0.19	0.39	0.14	0.48	0.22	0.36	0.14	0.63	0.27
	0.15	0.35	0.12	0.29	0.11	0.33	0.14	0.24	0.08	0.44	0.16
	0.20	0.25	0.09	0.20	0.07	0.27	0.11	0.18	0.06	0.26	0.11
	0.25	0.20	0.08	0.15	0.07	0.20	0.10	0.15	0.07	0.19	0.06
3MOAP10	0.05	12.36	10.79	4.72	1.19	5.35	1.58	2.74	0.69	8.41	2.84
	0.10	1.74	0.61	1.72	0.44	1.82	0.36	0.91	0.25	1.92	0.54
	0.15	0.93	0.20	1.05	0.25	0.96	0.14	0.44	0.07	0.89	0.22
	0.20	0.53	0.10	0.54	0.13	0.50	0.14	0.28	0.10	0.40	0.14
	0.25	0.33	0.07	0.33	0.07	0.28	0.08	0.18	0.04	0.26	0.09
3MOAP15	0.05	66.42	57.03	8.68	1.78	9.27	2.34	4.13	1.17	14.86	4.78
	0.10	4.10	0.99	3.16	0.38	3.20	0.75	1.24	0.23	2.60	0.85
	0.15	1.52	0.25	1.66	0.26	1.63	0.23	0.61	0.12	0.98	0.39
	0.20	0.85	0.19	1.00	0.25	0.88	0.19	0.40	0.07	0.50	0.10
	0.25	0.46	0.11	0.55	0.15	0.47	0.11	0.24	0.05	0.33	0.05
3MOAP20	0.05	217.16	101.19	16.13	1.90	17.89	1.40	8.05	1.26	29.44	3.61
	0.10	9.24	2.12	5.42	0.60	5.59	0.83	1.98	0.30	4.45	0.75
	0.15	3.52	0.87	3.17	0.62	2.78	0.46	0.94	0.16	1.67	0.32
	0.20	1.60	0.25	1.70	0.28	1.45	0.22	0.51	0.13	0.72	0.12
	0.25	0.90	0.23	0.96	0.20	0.79	0.12	0.35	0.06	0.50	0.07

Table C.5: CPU Time (secs) Comparison for MOAP ($m = 4$)

Problem	α	DMA*		SBA		TDA		TSGA		TSGA-II	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOAP5	0.05	2.21	1.56	3.86	2.43	3.50	2.01	2.70	1.40	11.03	7.45
	0.10	1.19	0.74	2.32	1.55	2.30	1.37	1.48	0.75	4.47	2.86
	0.15	0.68	0.37	1.48	1.02	1.32	0.76	0.85	0.35	1.86	0.98
	0.20	0.44	0.24	0.88	0.63	0.83	0.53	0.54	0.21	1.09	0.55
	0.25	0.33	0.18	0.53	0.42	0.49	0.33	0.36	0.13	0.59	0.27
4MOAP10	0.05	7596.01	8417.43	54.35	18.09	65.58	22.99	33.45	10.59	257.31	137.87
	0.10	32.10	20.26	13.65	3.86	14.93	4.43	5.44	1.93	22.19	10.30
	0.15	4.04	1.71	5.29	1.34	5.26	1.44	1.81	0.46	5.53	2.88
	0.20	1.28	0.38	2.38	0.73	2.30	0.85	0.75	0.16	1.68	0.67
	0.25	0.66	0.10	1.15	0.32	0.98	0.26	0.45	0.10	0.74	0.28
4MOAP15	0.05	-	-	151.03	64.26	176.79	75.15	52.24	26.66	668.63	495.34
	0.10	644.76	722.79	30.52	12.36	33.54	15.09	7.03	3.09	43.69	25.67
	0.15	22.63	14.32	11.12	4.47	11.14	4.90	1.97	0.75	6.95	3.17
	0.20	4.00	2.24	4.77	2.00	4.62	2.44	0.93	0.32	2.32	0.92
	0.25	1.22	0.60	2.26	1.03	1.82	0.82	0.55	0.17	0.97	0.38

* Results are not reported for 4MOAP15 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.

Table C.6: CPU Time (secs) Comparison for MOAP ($m = 5$)

		DMA*		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOAP5	0.05	5.46	2.02	10.25	3.89	10.24	3.73	8.73	3.13	43.74	14.41
	0.10	1.71	0.76	19.94	44.59	5.87	2.98	3.68	0.99	14.12	5.42
	0.15	0.84	0.24	5.94	8.74	3.19	1.22	1.91	0.62	4.86	1.37
	0.20	0.55	0.13	1.77	0.82	2.06	1.08	0.97	0.24	2.30	0.83
	0.25	0.41	0.12	1.01	0.38	1.12	0.49	0.68	0.23	1.14	0.50
5MOAP10	0.05	-	-	709.61	274.31	848.09	373.67	445.72	209.64	7376.14	5066.06
	0.10	962.44	849.15	127.65	49.93	130.43	47.74	40.66	16.69	303.68	177.52
	0.15	30.48	14.75	33.47	12.48	30.93	14.20	7.46	3.20	36.95	19.56
	0.20	5.56	2.90	13.36	6.87	12.16	6.90	2.88	1.39	8.76	3.11
	0.25	1.57	0.90	5.08	2.75	4.90	2.93	1.39	0.51	3.46	1.53

[*] Results are not reported for 5MOAP10 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.

APPENDIX D

COMPARISON OF NUMBER OF MODELS SOLVED FOR DMA, SBA, TDA, TSGA AND TSGA-II

Table D.1: Comparison of Number of Models Solved for MOKP ($m = 3$)

		DMA		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP10	0.05	7.60	3.81	28.20	15.17	30.20	17.47	26.60	13.78	55.70	40.82
	0.10	6.30	2.54	23.10	10.54	24.50	12.17	21.20	9.48	36.80	22.58
	0.15	5.00	1.70	17.60	6.47	19.30	8.72	16.50	6.52	26.20	14.57
	0.20	4.30	1.25	14.90	4.75	15.90	6.49	12.60	3.53	17.70	6.78
	0.25	3.80	1.14	13.70	4.62	14.60	6.22	10.90	2.60	15.50	5.48
3MOKP20	0.05	17.60	8.57	68.60	36.52	70.00	38.14	53.00	25.49	142.10	94.94
	0.10	10.80	4.10	41.40	17.02	42.00	17.08	28.00	10.38	54.40	30.25
	0.15	8.10	3.11	30.90	13.58	30.40	12.31	18.50	6.70	28.00	12.66
	0.20	5.90	2.18	21.10	8.46	21.10	7.34	12.50	3.60	18.10	8.67
	0.25	4.20	1.62	14.70	5.72	15.10	5.88	9.80	2.94	13.30	5.06
3MOKP30	0.05	36.80	7.07	157.00	36.21	190.70	55.11	106.90	25.70	321.70	103.87
	0.10	19.00	3.06	75.70	13.78	81.30	21.08	44.20	9.87	97.00	26.20
	0.15	12.50	2.51	46.20	9.60	45.60	9.36	23.30	4.06	42.30	12.76
	0.20	8.80	1.32	32.00	4.00	31.80	5.85	16.70	3.47	21.50	6.54
	0.25	5.60	1.17	19.70	4.11	19.60	3.03	11.40	1.71	15.00	4.85
3MOKP40	0.05	39.90	9.83	167.10	47.61	192.00	71.91	97.70	34.10	298.00	136.60
	0.10	20.60	4.45	79.40	19.83	81.60	16.91	35.60	9.73	72.10	19.20
	0.15	12.80	2.25	46.90	8.31	46.50	11.51	19.70	4.57	31.30	10.88
	0.20	8.60	2.59	30.30	9.13	30.20	8.40	13.00	2.58	19.60	2.95
	0.25	5.00	1.83	17.40	6.22	17.10	6.12	10.20	3.88	12.80	4.21
3MOKP50	0.05	50.10	13.47	220.00	67.32	262.90	92.82	116.70	37.46	390.90	135.55
	0.10	22.80	5.90	90.00	27.76	98.80	34.89	40.10	13.43	83.20	36.35
	0.15	13.60	4.30	49.60	16.61	51.20	17.15	20.80	7.02	33.30	14.43
	0.20	9.40	2.55	33.00	8.26	34.50	10.36	13.80	4.39	17.10	4.38
	0.25	6.00	1.49	21.30	4.88	21.60	3.50	10.00	2.00	14.60	2.95
3MOKP100	0.05	68.90	10.10	303.80	50.17	416.40	58.60	167.90	36.88	546.80	120.74
	0.10	30.20	4.69	113.50	18.20	138.30	25.23	46.90	14.17	92.80	22.14
	0.15	19.10	2.38	68.40	8.42	77.90	13.12	22.60	5.27	36.20	9.48
	0.20	12.80	1.75	44.60	6.28	47.80	8.93	14.00	2.31	19.90	3.57
	0.25	7.90	1.66	27.10	5.38	27.60	4.53	12.70	0.95	14.70	1.49

Table D.2: Comparison of Number of Models Solved for MOKP ($m = 4$)

Problem	α	DMA*		SBA		TDA		TSGA		TSGA-II	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOKP10	0.05	8.50	3.95	49.20	26.64	49.80	27.31	42.80	24.52	120.00	80.35
	0.10	6.70	3.47	38.80	23.20	38.00	20.27	30.80	15.96	66.60	47.46
	0.15	5.20	2.35	28.00	16.28	30.60	17.17	23.00	10.07	46.20	24.97
	0.20	3.90	1.91	20.20	11.08	21.80	10.76	17.80	10.76	29.60	17.36
	0.25	2.90	1.20	14.20	6.88	15.40	7.04	14.40	5.58	22.20	9.94
4MOKP20	0.05	43.30	19.93	443.20	275.96	489.20	332.02	259.40	149.88	1723.00	1344.08
	0.10	22.00	8.45	194.20	112.95	225.00	136.98	86.00	39.33	314.60	171.98
	0.15	13.70	5.68	101.60	57.65	121.00	82.08	47.00	32.69	117.60	55.51
	0.20	8.60	2.91	55.40	22.03	58.40	34.80	26.80	11.68	48.20	20.96
	0.25	5.70	1.95	32.80	12.02	31.20	14.31	16.20	6.05	29.80	13.31
4MOKP30	0.05	75.60	22.47	848.20	309.90	1062.80	431.71	449.60	151.87	2848.40	1147.50
	0.10	33.30	7.04	308.60	87.49	367.00	149.28	114.40	37.35	456.80	206.77
	0.15	18.60	4.27	143.80	46.90	161.20	49.76	50.00	13.27	134.40	55.72
	0.20	10.80	2.74	73.80	22.47	82.60	31.22	26.60	6.52	54.60	24.56
	0.25	6.60	1.71	39.80	16.12	43.60	15.49	18.20	5.01	26.80	11.56
4MOKP40	0.05	86.25	9.95	1651.00	724.71	2097.00	977.34	702.00	341.66	6056.60	3240.56
	0.10	44.20	8.16	443.60	112.25	537.40	256.67	143.20	67.02	578.60	289.92
	0.15	23.00	4.40	172.00	37.64	194.60	55.91	61.80	25.77	135.00	74.02
	0.20	13.30	2.63	85.60	19.32	96.40	34.18	30.20	9.20	61.00	19.04
	0.25	7.80	2.62	0.93	0.36	47.80	17.44	18.20	5.90	29.60	13.00

* Fewer than 10 instances per cell are reported for 4MOKP40 since DMA could not be finalized within 12 hours.

Table D.3: Comparison of Number of Models Solved for MOKP ($m = 5$)

Problem	α	DMA		SBA		TDA		TSGA		TSGA-II	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOKP10	0.05	14.30	6.09	175.80	106.04	173.30	104.30	154.90	86.22	698.90	536.68
	0.10	10.50	3.84	118.10	56.28	120.60	58.36	97.00	44.31	320.50	205.53
	0.15	8.20	2.66	81.60	39.60	84.70	43.50	70.20	26.54	180.40	92.67
	0.20	6.80	2.20	63.40	28.48	62.30	27.15	49.30	15.11	108.60	73.88
	0.25	4.60	1.51	34.10	12.90	33.00	13.76	37.00	12.03	67.10	34.00
5MOKP20	0.05	55.00	24.78	1448.00	1257.49	1511.00	1367.14	923.30	819.24	9331.60	9104.69
	0.10	25.00	8.49	485.60	399.59	489.10	367.34	272.50	228.08	1185.70	972.87
	0.15	14.90	3.90	200.70	131.13	170.10	135.63	94.60	61.66	297.20	176.52
	0.20	8.70	2.36	90.30	43.46	84.30	42.28	43.20	39.51	95.80	81.37
	0.25	6.40	2.27	55.30	27.53	58.50	34.17	27.70	19.20	51.00	28.93

Table D.4: Comparison of Number of Models Solved for MOAP ($m = 3$)

		DMA		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOAP5	0.05	14.20	3.88	52.90	16.00	55.90	19.26	49.20	16.32	116.20	45.77
	0.10	10.70	3.30	40.10	13.51	43.20	17.00	34.40	13.54	66.90	29.24
	0.15	8.30	2.45	30.10	9.50	30.10	10.32	25.00	7.21	45.00	17.44
	0.20	6.10	1.91	21.60	7.28	21.20	7.77	17.80	6.29	26.70	12.08
	0.25	4.60	1.90	15.80	7.08	16.30	7.42	14.30	6.55	19.20	6.97
3MOAP10	0.05	47.50	10.23	209.30	50.50	254.30	68.61	130.20	26.47	497.10	165.85
	0.10	22.30	4.40	88.60	19.24	100.80	19.66	51.10	14.37	114.60	30.91
	0.15	14.70	2.16	54.30	8.25	56.80	7.16	24.10	3.38	53.20	13.36
	0.20	9.30	1.64	32.00	5.83	32.80	7.73	17.10	5.65	24.80	8.75
	0.25	6.10	1.20	20.50	4.22	20.00	4.78	11.10	2.02	15.10	5.65
3MOAP15	0.05	62.90	10.06	269.70	47.18	333.60	77.74	151.30	42.67	586.90	164.18
	0.10	28.00	2.94	106.30	12.07	124.70	21.80	51.10	10.35	108.00	35.44
	0.15	16.30	2.79	58.30	8.53	67.30	8.98	25.40	5.46	40.10	13.38
	0.20	10.50	2.07	35.90	7.78	38.60	8.25	16.70	2.83	22.10	5.74
	0.25	6.00	1.25	20.70	4.16	21.80	4.92	10.30	2.63	14.40	3.13
3MOAP20	0.05	78.60	6.31	355.50	38.05	455.70	32.35	219.20	25.76	818.90	98.59
	0.10	31.40	3.13	127.60	12.33	155.60	20.38	62.40	9.48	128.60	21.58
	0.15	21.20	3.22	80.20	13.75	84.60	13.72	30.60	4.30	48.80	7.66
	0.20	13.50	1.90	47.20	7.27	50.10	7.28	16.60	3.41	22.10	3.84
	0.25	8.60	1.71	28.80	5.09	29.70	4.60	12.20	2.10	15.20	2.20

Table D.5: Comparison of Number of Models Solved for MOAP ($m = 4$)

		DMA*		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOAP5	0.05	27.10	10.71	276.60	157.68	290.80	159.17	234.60	116.40	1131.40	743.57
	0.10	19.20	8.07	191.00	121.33	203.20	118.08	138.40	66.10	476.80	306.71
	0.15	13.10	5.74	124.00	81.18	120.20	68.31	81.00	33.41	190.60	102.79
	0.20	9.40	3.98	76.20	50.07	77.40	48.58	52.40	19.51	113.80	58.21
	0.25	6.70	2.83	46.80	32.20	47.20	31.16	34.80	12.87	60.40	30.07
4MOAP10	0.05	149.60	38.38	2077.60	630.21	2648.40	811.26	1078.20	333.31	11365.00	5809.17
	0.10	51.30	9.96	580.00	141.18	712.60	185.51	242.40	72.08	1165.40	552.84
	0.15	27.20	4.96	239.60	50.88	275.00	67.65	98.40	25.14	310.40	163.68
	0.20	15.50	3.37	116.00	32.47	129.60	41.53	44.60	8.15	100.80	38.20
	0.25	9.50	1.35	61.00	13.33	58.40	11.89	27.60	6.26	44.20	17.47
4MOAP15	0.05	-	-	3798.60	1530.84	5206.00	2053.43	1566.80	733.72	21795.60	15383.83
	0.10	68.30	23.73	838.40	320.61	1120.20	473.90	260.80	111.70	1574.60	916.54
	0.15	34.20	12.43	325.20	125.62	405.40	166.66	80.40	30.10	266.40	117.86
	0.20	19.50	7.41	147.60	58.07	184.40	88.33	40.20	14.52	96.40	40.49
	0.25	11.00	3.94	72.20	29.79	82.00	35.69	23.80	7.79	39.80	16.12

* Results are not reported for 4MOAP15 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.

Table D.6: Comparison of Number of Models Solved for MOAP ($m = 5$)

		DMA*		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOAP5	0.05	36.90	5.99	750.90	280.56	774.70	262.15	641.60	186.02	4467.80	1423.35
	0.10	21.90	4.86	427.10	180.77	500.20	238.22	337.00	94.11	1524.60	581.19
	0.15	14.50	3.63	252.00	106.63	282.70	108.05	182.70	58.19	512.00	150.90
	0.20	10.60	2.27	143.80	64.96	184.90	97.75	93.10	23.05	249.10	94.38
	0.25	8.10	1.85	87.20	30.92	102.10	44.40	64.30	21.63	122.30	56.63
5MOAP10	0.05	-	-	22859.50	8073.70	29379.30	11360.11	10408.50	4090.54	256990.30	162147.28
	0.10	96.00	16.32	4475.50	1596.49	5413.10	1763.32	1509.10	511.96	12793.70	6761.05
	0.15	43.40	6.26	1256.90	422.89	1430.90	565.93	354.10	127.90	1837.10	879.57
	0.20	24.10	5.70	517.20	250.51	615.30	335.27	161.50	68.76	485.00	170.97
	0.25	13.70	3.74	209.10	100.31	259.70	142.45	82.60	27.17	203.50	69.31

* Results are not reported for 5MOAP10 instances ($\alpha = 0.05$) since DMA could not be finalized within 12 hours.

APPENDIX E

COMPARISONS FOR EXPERIMENTS WITH MOMIPS

Table E.1: Cardinality Comparison for MOMIPs

		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MIKP100	0.05	93.10	16.60	112.80	18.48	63.90	14.43	47.30	11.00
	0.10	33.50	4.86	37.60	4.93	16.80	3.36	12.90	2.42
	0.15	22.00	4.03	22.10	4.48	7.60	1.78	6.20	0.92
	0.20	14.20	2.97	14.90	3.31	4.90	0.88	3.70	0.48
	0.25	9.60	1.96	9.90	1.97	3.60	0.52	3.00	0.47
4MIKP40	0.05	312.00	101.00	369.70	142.48	196.80	82.15	167.80	71.30
	0.10	73.90	21.67	86.60	27.58	37.00	12.25	31.00	9.80
	0.15	37.50	11.35	39.00	12.35	14.70	4.35	11.60	3.06
	0.20	23.30	6.31	23.70	7.93	8.00	1.94	6.30	1.70
	0.25	15.00	4.08	14.60	3.86	5.20	1.23	4.60	0.97
5MIKP20	0.05	335.40	299.97	351.50	317.07	238.70	223.06	206.70	210.07
	0.10	70.90	49.01	75.60	52.64	42.40	31.60	39.00	31.66
	0.15	33.30	16.55	34.70	21.13	17.10	10.82	14.60	9.71
	0.20	18.70	7.86	19.40	9.66	9.10	5.28	7.90	4.46
	0.25	12.70	6.70	11.70	6.04	5.60	2.50	4.80	1.93

Table E.2: CPU Time Comparison for MOMIPs

		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MIKP100	0.05	6.64	1.44	8.96	1.53	3.72	0.98	10.09	3.43
	0.10	2.13	0.36	2.63	0.48	0.93	0.22	1.54	0.41
	0.15	1.21	0.26	1.34	0.35	0.41	0.11	0.58	0.13
	0.20	0.72	0.14	0.81	0.18	0.26	0.05	0.31	0.06
	0.25	0.46	0.10	0.52	0.13	0.20	0.03	0.25	0.05
4MIKP40	0.05	73.13	29.91	98.30	50.41	25.67	13.05	312.82	210.27
	0.10	12.53	5.64	16.13	7.50	3.37	1.33	16.92	10.20
	0.15	5.05	1.88	5.30	2.43	1.17	0.38	3.23	1.70
	0.20	2.57	0.79	2.55	0.95	0.54	0.14	1.04	0.47
	0.25	1.33	0.43	1.34	0.53	0.33	0.09	0.62	0.24
5MIKP20	0.05	245.56	340.17	317.65	488.11	119.94	190.41	3135.88	6895.00
	0.10	30.67	36.78	38.39	48.18	10.76	14.52	99.03	198.39
	0.15	8.99	8.29	10.80	11.47	2.95	3.00	13.06	18.35
	0.20	3.94	3.35	4.20	3.40	1.17	1.23	3.45	4.36
	0.25	2.17	2.08	1.93	1.76	0.55	0.37	1.07	0.70

Table E.3: Comparison of Number of Models Solved for MOMIPs

		SBA		TDA		TSGA		TSGA-II	
Problem	α	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MIKP100	0.05	409.70	84.22	592.40	100.97	227.70	57.76	705.60	260.46
	0.10	131.40	23.21	175.10	30.56	57.60	12.82	104.50	29.04
	0.15	80.40	15.56	92.50	24.07	25.30	6.29	38.50	8.62
	0.20	51.20	10.79	56.10	12.74	16.10	3.07	18.60	3.37
	0.25	33.70	6.78	35.60	7.52	11.80	1.55	14.20	2.94
4MIKP40	0.05	4704.80	2071.14	7286.40	3761.62	1877.20	1012.74	27208.00	18091.66
	0.10	836.40	348.46	1232.60	565.96	250.80	96.90	1515.00	933.00
	0.15	333.60	119.58	412.60	188.33	86.60	32.78	262.40	145.23
	0.20	177.60	50.18	199.60	75.11	40.00	10.80	88.00	41.22
	0.25	100.80	30.35	105.00	40.69	25.20	6.00	52.60	21.88
5MIKP20	0.05	16626.00	21980.52	20440.40	28423.01	7662.00	10998.90	244550.50	526726.66
	0.10	2151.00	2446.02	2793.30	3304.24	816.30	1051.83	8455.20	16563.03
	0.15	655.30	567.13	830.80	858.13	230.70	223.25	1177.10	1643.77
	0.20	293.10	228.33	329.30	254.18	96.80	97.66	315.10	404.18
	0.25	163.40	143.56	153.80	137.25	45.90	30.94	94.70	65.43

APPENDIX F

COMPARISONS FOR VARIATIONS OF TSGA AND TSGA-II

Table F.1: TSGA Comparisons for MOKP ($m = 3$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP10	0.05	7.20	3.52	7.20	3.52	0.42	0.23	0.45	0.28	26.60	13.78	25.80	13.26
	0.10	5.70	2.21	5.60	2.27	0.32	0.15	0.35	0.19	21.20	9.48	20.00	8.81
	0.15	4.50	1.58	4.60	1.51	0.25	0.10	0.28	0.14	16.50	6.52	15.90	5.32
	0.20	3.50	0.85	3.60	0.84	0.18	0.05	0.25	0.16	12.60	3.53	12.40	2.76
	0.25	3.00	0.67	3.00	0.67	0.15	0.03	0.23	0.27	10.90	2.60	10.50	2.07
3MOKP20	0.05	14.40	6.28	14.50	6.43	1.20	0.71	1.64	0.95	53.00	25.49	53.90	26.42
	0.10	8.00	2.87	7.90	2.64	0.49	0.22	0.79	0.39	28.00	10.38	27.80	10.01
	0.15	5.30	1.77	5.20	1.81	0.31	0.16	0.44	0.25	18.50	6.70	18.30	7.06
	0.20	3.60	0.97	3.70	1.06	0.20	0.08	0.27	0.13	12.50	3.60	12.90	4.20
	0.25	2.80	0.79	2.90	0.74	0.15	0.05	0.22	0.08	9.80	2.94	10.00	2.62
3MOKP30	0.05	27.70	5.66	27.60	5.58	2.54	0.74	4.10	1.17	106.90	25.70	108.10	25.77
	0.10	12.20	2.25	12.00	2.36	0.86	0.21	1.25	0.40	44.20	9.87	43.30	11.00
	0.15	6.80	1.14	7.10	1.29	0.45	0.12	0.55	0.16	23.30	4.06	24.30	4.42
	0.20	4.90	0.88	4.50	0.97	0.28	0.05	0.39	0.15	16.70	3.47	15.50	4.01
	0.25	3.30	0.48	3.10	0.74	0.19	0.04	0.19	0.07	11.40	1.71	10.40	2.22
3MOKP40	0.05	26.70	8.67	26.80	8.83	2.37	0.95	3.71	1.61	97.70	34.10	98.10	35.40
	0.10	10.10	2.42	10.30	2.87	0.73	0.24	1.03	0.41	35.60	9.73	35.70	10.82
	0.15	5.70	1.25	5.50	1.08	0.37	0.12	0.42	0.16	19.70	4.57	18.90	4.28
	0.20	3.90	0.74	4.00	0.82	0.23	0.04	0.28	0.09	13.00	2.58	13.20	2.74
	0.25	3.00	1.15	2.90	0.99	0.16	0.07	0.17	0.06	10.20	3.88	9.70	2.98
3MOKP50	0.05	31.30	9.17	31.50	8.91	3.37	1.20	3.19	1.04	116.70	37.46	117.10	34.93
	0.10	11.30	3.43	11.20	3.12	0.92	0.30	0.91	0.32	40.10	13.43	40.90	13.25
	0.15	6.20	1.93	6.10	1.60	0.43	0.15	0.39	0.12	20.80	7.02	20.20	5.33
	0.20	4.10	1.20	4.20	1.23	0.29	0.13	0.27	0.10	13.80	4.39	14.20	4.24
	0.25	3.00	0.67	3.10	0.74	0.20	0.06	0.19	0.07	10.00	2.00	10.30	2.21
3MOKP100	0.05	45.30	8.97	44.80	8.65	6.37	1.84	4.99	1.39	167.90	36.88	166.00	36.97
	0.10	13.30	3.33	13.60	3.20	1.35	0.43	1.15	0.39	46.90	14.17	47.30	13.36
	0.15	6.80	1.40	6.70	1.06	0.55	0.16	0.45	0.11	22.60	5.27	22.40	4.48
	0.20	4.30	0.67	4.30	0.82	0.35	0.05	0.27	0.06	14.00	2.31	14.00	2.58
	0.25	3.90	0.32	3.80	0.42	0.37	0.11	0.25	0.04	12.70	0.95	12.40	1.26

Table F.2: TSGA Comparisons for MOKP ($m = 4$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
	0.05	7.90	3.84	7.80	3.65	0.73	0.34	0.63	0.47	42.80	24.52	41.40	21.68
	0.10	5.70	2.45	5.50	2.46	0.52	0.19	0.40	0.29	30.80	15.96	28.80	14.65
4MOKP10	0.15	4.50	1.51	4.30	1.25	0.41	0.12	0.27	0.11	23.00	10.07	21.00	7.66
	0.20	3.40	1.71	3.10	1.52	0.34	0.12	0.18	0.09	17.80	10.76	15.40	8.26
	0.25	2.90	0.99	2.70	1.06	0.30	0.11	0.15	0.06	14.40	5.58	13.20	5.29
	0.05	32.80	14.99	33.10	15.53	6.75	4.46	7.10	4.86	259.40	149.88	263.80	157.18
	0.10	13.00	4.69	13.20	5.20	1.59	0.81	1.71	0.98	86.00	39.33	88.20	47.95
4MOKP20	0.15	7.60	3.47	7.60	3.10	0.77	0.50	0.70	0.41	47.00	32.69	45.60	26.50
	0.20	5.00	1.76	5.00	1.76	0.40	0.20	0.40	0.20	26.80	11.68	27.80	13.80
	0.25	3.30	1.06	3.40	0.97	0.24	0.08	0.23	0.09	16.20	6.05	17.00	5.89
	0.05	54.10	15.98	54.00	15.85	13.40	5.22	13.01	5.19	449.60	151.87	445.60	154.81
	0.10	17.60	4.38	18.00	4.55	2.53	0.91	2.33	0.81	114.40	37.35	112.00	35.63
4MOKP30	0.15	9.00	1.76	8.80	2.04	0.85	0.25	0.82	0.24	50.00	13.27	47.80	14.55
	0.20	5.60	1.35	5.30	1.57	0.44	0.11	0.40	0.14	26.60	6.52	25.60	7.18
	0.25	3.80	0.92	3.70	0.67	0.28	0.07	0.24	0.06	18.20	5.01	17.80	4.24
	0.05	76.30	25.36	74.50	25.14	21.20	11.36	20.95	10.93	702.00	341.66	683.00	324.54
	0.10	20.50	6.40	20.30	5.79	3.13	1.59	2.97	1.35	143.20	67.02	134.20	54.48
4MOKP40	0.15	10.40	3.06	9.40	3.41	1.22	0.53	0.96	0.44	61.80	25.77	53.00	23.70
	0.20	6.00	1.41	5.60	1.35	0.52	0.16	0.44	0.16	30.20	9.20	26.60	7.76
	0.25	3.70	0.82	3.60	0.97	0.31	0.11	0.26	0.08	18.20	5.90	16.60	4.20

Table F.3: TSGA Comparisons for MOKP ($m = 5$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
	0.05	13.70	5.42	13.70	5.42	2.42	1.57	2.51	1.79	154.90	86.22	151.00	87.27
	0.10	9.30	3.47	9.30	3.47	1.33	0.67	1.26	0.69	97.00	44.31	94.40	44.15
5MOKP10	0.15	7.20	1.93	6.90	2.13	0.96	0.42	0.73	0.36	70.20	26.54	62.40	27.77
	0.20	5.70	1.42	5.40	1.65	0.61	0.20	0.48	0.19	49.30	15.11	44.60	17.10
	0.25	4.70	1.06	4.30	1.25	0.44	0.16	0.31	0.09	37.00	12.03	30.00	9.03
	0.05	46.60	20.71	46.00	20.85	28.68	28.97	30.67	33.00	923.30	819.24	902.20	816.98
	0.10	18.60	7.21	16.70	5.33	5.73	5.31	4.69	4.10	272.50	228.08	225.30	164.78
5MOKP20	0.15	9.40	2.41	9.10	2.96	1.51	0.98	1.41	1.30	94.60	61.66	87.90	67.89
	0.20	4.90	2.18	4.60	1.96	0.65	0.53	0.51	0.37	43.20	39.51	35.50	24.65
	0.25	3.60	1.65	3.10	0.99	0.41	0.27	0.28	0.14	27.70	19.20	22.10	10.75

Table F.4: TSGA Comparisons for MOAP ($m = 3$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOAP5	0.05	13.30	4.11	13.20	4.05	0.53	0.18	0.71	0.30	49.20	16.32	49.00	15.42
	0.10	9.40	3.37	9.10	3.14	0.36	0.14	0.42	0.16	34.40	13.54	33.50	12.17
	0.15	6.80	1.75	6.80	1.87	0.24	0.08	0.34	0.10	25.00	7.21	24.90	7.89
	0.20	4.90	1.45	4.70	1.34	0.18	0.06	0.22	0.08	17.80	6.29	16.70	5.29
	0.25	4.00	1.63	3.60	1.26	0.15	0.07	0.17	0.07	14.30	6.55	12.50	4.22
3MOAP10	0.05	34.60	6.10	34.70	6.11	2.74	0.69	3.01	0.87	130.20	26.47	131.50	26.58
	0.10	14.00	3.65	13.90	3.41	0.91	0.25	0.91	0.25	51.10	14.37	49.90	13.75
	0.15	7.20	0.92	7.40	1.43	0.44	0.07	0.40	0.10	24.10	3.38	24.80	5.29
	0.20	5.10	1.52	4.80	1.55	0.28	0.10	0.26	0.10	17.10	5.65	16.20	5.67
	0.25	3.30	0.67	3.50	0.71	0.18	0.04	0.17	0.03	11.10	2.02	11.80	2.44
3MOAP15	0.05	39.30	9.87	39.60	9.75	4.13	1.17	4.07	1.14	151.30	42.67	150.90	40.58
	0.10	14.20	2.20	13.80	2.82	1.24	0.23	1.17	0.30	51.10	10.35	49.60	11.97
	0.15	7.30	1.57	7.00	1.25	0.61	0.12	0.54	0.12	25.40	5.46	24.10	4.31
	0.20	4.90	0.74	4.60	0.84	0.40	0.07	0.31	0.05	16.70	2.83	15.30	2.54
	0.25	3.00	0.67	3.20	0.63	0.24	0.05	0.22	0.06	10.30	2.63	10.80	2.20
3MOAP20	0.05	56.70	6.29	56.50	6.33	8.05	1.26	7.76	1.01	219.20	25.76	219.40	23.58
	0.10	16.90	2.38	17.30	2.36	1.98	0.30	1.95	0.26	62.40	9.48	64.70	9.57
	0.15	8.60	1.17	8.80	1.14	0.94	0.16	0.87	0.15	30.60	4.30	31.00	4.24
	0.20	4.80	0.79	5.10	0.99	0.51	0.13	0.45	0.10	16.60	3.41	17.20	3.65
	0.25	3.70	0.67	3.80	0.63	0.35	0.06	0.30	0.06	12.20	2.10	12.70	2.41

Table F.5: TSGA Comparisons for MOAP ($m = 4$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOAP5	0.05	27.00	10.46	26.50	10.47	2.70	1.40	3.01	1.92	234.60	116.40	228.40	122.94
	0.10	16.90	6.14	16.50	6.49	1.48	0.75	1.50	0.82	138.40	66.10	134.40	71.76
	0.15	10.70	3.62	10.40	3.84	0.85	0.35	0.80	0.39	81.00	33.41	76.20	35.79
	0.20	7.80	2.53	6.80	2.78	0.54	0.21	0.46	0.24	52.40	19.51	43.20	20.73
	0.25	5.50	1.43	4.80	1.62	0.36	0.13	0.31	0.14	34.80	12.87	28.40	12.44
4MOAP10	0.05	104.30	26.06	102.80	25.23	33.45	10.59	32.61	10.60	1078.20	333.31	1052.20	330.62
	0.10	29.50	5.91	30.50	6.04	5.44	1.93	5.37	1.83	242.40	72.08	246.80	67.49
	0.15	14.50	2.59	14.30	2.63	1.81	0.46	1.58	0.38	98.40	25.14	91.80	21.61
	0.20	8.00	1.25	8.00	1.05	0.75	0.16	0.71	0.15	44.60	8.15	43.80	8.34
	0.25	5.10	0.88	4.90	0.88	0.45	0.10	0.38	0.08	27.60	6.26	25.00	4.90
4MOAP15	0.05	147.30	57.22	148.90	57.23	52.24	26.66	52.30	25.80	1566.80	733.72	1583.60	711.04
	0.10	33.50	12.20	35.20	13.11	7.03	3.09	7.15	3.36	260.80	111.70	276.80	123.55
	0.15	13.00	4.37	14.10	4.91	1.97	0.75	2.13	0.92	80.40	30.10	92.20	38.62
	0.20	7.50	2.68	7.50	2.59	0.93	0.32	0.91	0.37	40.20	14.52	42.00	16.87
	0.25	4.90	1.60	4.90	1.60	0.55	0.17	0.50	0.17	23.80	7.79	24.20	7.79

Table F.6: TSGA Comparisons for MOAP ($m = 5$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOAP5	0.05	35.80	5.88	35.30	5.50	8.73	3.13	7.93	2.65	641.60	186.02	608.90	184.55
	0.10	21.00	3.83	20.40	3.41	3.68	0.99	3.54	1.07	337.00	94.11	315.90	91.48
	0.15	12.80	2.30	11.80	1.81	1.91	0.62	1.64	0.54	182.70	58.19	153.90	48.16
	0.20	8.30	1.49	7.90	0.88	0.97	0.24	0.85	0.20	93.10	23.05	84.20	19.38
	0.25	6.30	1.06	5.70	0.82	0.68	0.23	0.51	0.14	64.30	21.63	50.80	13.18
5MOAP10	0.05	279.70	62.35	279.00	61.82	445.72	209.64	432.46	197.51	10408.50	4090.54	10261.10	4131.62
	0.10	60.60	10.43	59.50	10.23	40.66	16.69	37.61	16.49	1509.10	511.96	1434.90	535.89
	0.15	23.40	3.75	23.30	4.60	7.46	3.20	7.88	3.98	354.10	127.90	375.80	162.38
	0.20	12.30	2.58	12.60	1.65	2.88	1.39	2.58	1.06	161.50	68.76	147.40	52.07
	0.25	8.00	1.63	7.50	1.51	1.39	0.51	1.07	0.51	82.60	27.17	65.50	25.11

Table F.7: TSGA Comparisons for MOMIP Experiments

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)		TSGA (orig.)		TSGA (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MIKP100	0.05	63.90	14.43	63.60	14.20	3.72	0.98	3.47	0.87	227.70	57.76	226.70	56.89
	0.10	16.80	3.36	17.00	3.09	0.93	0.22	0.87	0.18	57.60	12.82	57.40	11.46
	0.15	7.60	1.78	7.70	1.70	0.41	0.11	0.38	0.10	25.30	6.29	25.70	6.53
	0.20	4.90	0.88	4.60	0.70	0.26	0.05	0.22	0.04	16.10	3.07	14.90	2.33
	0.25	3.60	0.52	3.60	0.52	0.20	0.03	0.17	0.02	11.80	1.55	11.80	1.55
4MIKP40	0.05	196.80	82.15	193.20	75.62	25.67	13.05	25.76	14.21	1877.20	1012.74	1800.40	975.13
	0.10	37.00	12.25	37.80	12.49	3.37	1.33	3.44	1.45	250.80	96.90	249.00	107.22
	0.15	14.70	4.35	14.10	3.67	1.17	0.38	1.10	0.36	86.60	32.78	79.20	25.90
	0.20	8.00	1.94	7.30	1.95	0.54	0.14	0.47	0.12	40.00	10.80	35.20	9.45
	0.25	5.20	1.23	5.20	1.14	0.33	0.09	0.32	0.07	25.20	6.00	24.80	5.61
5MIKP20	0.05	238.70	223.06	234.80	233.01	119.94	190.41	107.48	168.66	7662.00	10998.90	6727.20	9722.02
	0.10	42.40	31.60	40.70	33.47	10.76	14.52	10.05	13.28	816.30	1051.83	719.60	904.59
	0.15	17.10	10.82	16.40	10.04	2.95	3.00	2.58	2.39	230.70	223.25	198.10	173.70
	0.20	9.10	5.28	8.60	4.60	1.17	1.23	0.96	0.82	96.80	97.66	77.40	64.34
	0.25	5.60	2.50	5.20	2.70	0.55	0.37	0.48	0.38	45.90	30.94	40.40	30.07

Table F.8: TSGA-II Comparisons for MOKP ($m = 3$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOKP10	0.05	7.20	3.52	7.20	3.52	0.64	0.45	0.65	0.47	55.70	40.82	55.70	40.82
	0.10	5.50	2.17	5.50	2.17	0.44	0.26	0.43	0.26	36.80	22.58	36.80	22.58
	0.15	4.30	1.42	4.30	1.42	0.32	0.16	0.30	0.15	26.20	14.57	26.20	14.57
	0.20	3.40	0.84	3.40	0.84	0.21	0.07	0.22	0.10	17.70	6.78	17.70	6.78
	0.25	3.10	0.74	3.10	0.74	0.19	0.06	0.19	0.07	15.50	5.48	15.50	5.48
3MOKP20	0.05	14.40	6.33	14.30	6.29	2.30	1.70	2.24	1.67	142.10	94.94	141.70	94.41
	0.10	7.50	2.46	7.50	2.46	0.86	0.47	0.84	0.48	54.40	30.25	54.40	29.97
	0.15	4.80	1.48	4.80	1.48	0.46	0.24	0.42	0.21	28.00	12.66	28.00	12.66
	0.20	3.40	1.07	3.40	1.07	0.29	0.14	0.27	0.13	18.10	8.67	18.20	8.64
	0.25	2.70	0.67	2.60	0.70	0.21	0.08	0.18	0.08	13.30	5.06	12.60	4.97
3MOKP30	0.05	27.00	4.90	27.00	4.81	5.77	1.91	5.48	1.81	321.70	103.87	322.60	99.62
	0.10	11.70	1.89	11.50	1.96	1.69	0.50	1.56	0.48	97.00	26.20	93.70	25.29
	0.15	6.60	1.17	6.60	1.07	0.77	0.19	0.68	0.16	42.30	12.76	42.10	10.73
	0.20	4.00	0.82	3.80	0.79	0.38	0.12	0.35	0.10	21.50	6.54	19.80	5.75
	0.25	3.00	0.67	3.00	0.67	0.28	0.11	0.24	0.07	15.00	4.85	15.00	4.85
3MOKP40	0.05	25.00	8.47	25.00	8.47	5.61	2.73	5.54	2.68	298.00	136.60	297.60	136.37
	0.10	9.40	1.96	9.20	1.93	1.36	0.51	1.31	0.53	72.10	19.20	72.00	20.39
	0.15	5.10	1.29	4.90	1.29	0.59	0.21	0.54	0.19	31.30	10.88	29.60	10.38
	0.20	3.70	0.48	3.60	0.52	0.35	0.07	0.30	0.05	19.60	2.95	18.70	2.67
	0.25	2.60	0.70	2.60	0.70	0.23	0.09	0.21	0.07	12.80	4.21	12.80	4.21
3MOKP50	0.05	29.50	7.43	29.40	7.44	8.11	3.28	8.17	3.42	390.90	135.55	395.90	141.39
	0.10	9.90	2.81	10.00	2.87	1.63	0.73	1.55	0.72	83.20	36.35	84.00	36.22
	0.15	5.40	1.71	5.40	1.71	0.74	0.32	0.64	0.29	33.30	14.43	33.30	14.43
	0.20	3.40	0.52	3.40	0.52	0.33	0.11	0.32	0.08	17.10	4.38	17.10	4.38
	0.25	2.80	0.42	2.80	0.42	0.31	0.08	0.27	0.06	14.60	2.95	14.30	2.95
3MOKP100	0.05	37.80	6.55	38.20	6.83	14.84	4.35	14.07	3.65	546.80	120.74	547.90	113.39
	0.10	11.40	2.12	11.50	2.22	2.35	0.69	2.28	0.62	92.80	22.14	94.40	19.25
	0.15	5.70	1.06	5.50	1.08	1.00	0.30	0.75	0.23	36.20	9.48	34.10	8.99
	0.20	3.80	0.42	3.70	0.48	0.50	0.08	0.43	0.09	19.90	3.57	19.30	2.75
	0.25	3.00	0.00	3.00	0.00	0.40	0.04	0.34	0.03	14.70	1.49	14.70	1.49

Table F.9: TSGA-II Comparisons for MOKP ($m = 4$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOKP10	0.05	7.80	3.65	7.80	3.65	1.46	0.82	1.26	0.92	120.00	80.35	120.00	80.35
	0.10	5.50	2.46	5.50	2.46	0.85	0.43	0.68	0.50	66.60	47.46	66.60	47.46
	0.15	4.40	1.58	4.20	1.14	0.65	0.24	0.42	0.20	46.20	24.97	42.40	16.47
	0.20	3.30	1.42	3.20	1.48	0.46	0.13	0.27	0.18	29.60	17.36	29.00	18.93
	0.25	2.80	0.92	2.60	0.97	0.38	0.13	0.20	0.10	22.20	9.94	20.40	10.59
4MOKP20	0.05	33.10	15.55	32.80	15.23	30.86	26.06	32.14	26.95	1723.00	1344.08	1694.40	1312.21
	0.10	13.10	4.61	13.10	4.91	4.58	2.67	4.60	2.67	314.60	171.98	321.20	183.07
	0.15	7.40	2.37	7.50	2.46	1.71	0.83	1.62	0.80	117.60	55.51	120.00	61.56
	0.20	4.30	1.34	4.70	1.89	0.76	0.38	0.74	0.44	48.20	20.96	57.60	36.45
	0.25	3.20	1.03	3.20	1.14	0.43	0.21	0.37	0.21	29.80	13.31	30.20	16.12
4MOKP30	0.05	52.60	14.66	51.70	14.35	56.08	26.82	53.90	27.43	2848.40	1147.50	2792.60	1159.67
	0.10	17.50	3.17	17.00	3.65	7.88	3.51	7.13	3.53	456.80	206.77	437.20	208.21
	0.15	8.40	1.65	7.90	1.52	2.06	0.81	1.75	0.60	134.40	55.72	118.00	36.57
	0.20	5.00	1.15	5.00	1.15	0.87	0.34	0.80	0.31	54.60	24.56	54.60	24.56
	0.25	3.10	0.88	3.10	0.88	0.41	0.16	0.40	0.20	26.80	11.56	26.80	11.56
4MOKP40	0.05	72.00	23.82	71.70	23.33	121.68	69.42	119.99	70.11	6056.60	3240.56	6061.20	3332.48
	0.10	18.80	5.14	18.50	5.44	10.07	5.38	9.77	5.39	578.60	289.92	559.60	277.56
	0.15	8.00	2.49	8.00	2.16	2.35	1.29	2.21	1.20	135.00	74.02	133.60	65.67
	0.20	5.20	0.92	4.90	0.99	0.98	0.32	0.86	0.29	61.00	19.04	54.40	18.36
	0.25	3.30	0.82	3.30	0.82	0.47	0.20	0.44	0.20	29.60	13.00	29.60	13.00

Table F.10: TSGA-II Comparisons for MOKP ($m = 5$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOKP10	0.05	13.70	5.42	13.70	5.42	7.50	6.15	7.32	6.09	698.90	536.68	698.90	536.68
	0.10	9.30	3.47	9.30	3.47	3.20	2.07	3.01	2.03	320.50	205.53	320.50	205.53
	0.15	7.00	2.11	6.90	2.13	1.79	0.96	1.59	0.90	180.40	92.67	173.60	95.45
	0.20	5.30	1.64	5.30	1.64	1.06	0.62	0.95	0.63	108.60	73.88	108.60	73.88
	0.25	4.30	1.25	4.30	1.25	0.65	0.26	0.57	0.27	67.10	34.00	66.70	33.93
5MOKP20	0.05	46.50	20.94	46.00	20.81	177.36	181.65	170.35	180.60	9331.60	9104.69	9233.90	9084.06
	0.10	16.60	5.87	16.20	5.81	18.12	15.77	17.11	16.14	1185.70	972.87	1153.40	974.22
	0.15	8.80	2.10	8.80	2.53	3.97	2.53	4.03	2.70	297.20	176.52	311.80	189.71
	0.20	4.60	1.84	4.50	1.65	1.28	1.01	1.09	0.81	95.80	81.37	87.10	65.67
	0.25	3.50	1.08	3.40	1.07	0.69	0.32	0.59	0.36	51.00	28.93	50.90	32.71

Table F.11: TSGA-II Comparisons for MOAP ($m = 3$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MOAP5	0.05	13.10	4.15	13.10	4.15	1.10	0.45	1.08	0.45	116.20	45.77	116.20	45.77
	0.10	8.90	2.88	8.80	2.82	0.63	0.27	0.59	0.26	66.90	29.24	65.70	28.89
	0.15	6.70	1.77	6.70	1.77	0.44	0.16	0.40	0.16	45.00	17.44	45.20	17.64
	0.20	4.50	1.51	4.60	1.51	0.26	0.11	0.25	0.11	26.70	12.08	27.40	11.79
	0.25	3.50	1.08	3.50	1.18	0.19	0.06	0.18	0.07	19.20	6.97	19.20	7.61
3MOAP10	0.05	33.40	6.52	33.70	6.58	8.41	2.84	8.23	2.62	497.10	165.85	508.20	160.38
	0.10	12.30	1.83	12.40	1.96	1.92	0.54	1.80	0.52	114.60	30.91	114.70	31.68
	0.15	7.10	1.20	6.90	1.20	0.89	0.22	0.80	0.22	53.20	13.36	50.70	14.51
	0.20	4.10	0.88	4.10	0.88	0.40	0.14	0.35	0.13	24.80	8.75	24.80	8.75
	0.25	2.90	0.74	2.90	0.74	0.26	0.09	0.21	0.09	15.10	5.65	15.20	5.51
3MOAP15	0.05	37.20	6.63	36.90	6.82	14.86	4.78	14.14	4.27	586.90	164.18	573.70	157.24
	0.10	11.80	2.15	11.60	1.96	2.60	0.85	2.45	0.82	108.00	35.44	101.90	35.52
	0.15	5.90	0.99	5.90	0.74	0.98	0.39	0.89	0.25	40.10	13.38	39.90	9.98
	0.20	3.90	0.74	3.60	0.52	0.50	0.10	0.41	0.08	22.10	5.74	19.50	3.41
	0.25	2.80	0.42	2.80	0.42	0.33	0.05	0.28	0.05	14.40	3.13	14.40	3.13
3MOAP20	0.05	48.80	4.69	48.20	4.78	29.44	3.61	28.86	3.68	818.90	98.59	818.30	100.76
	0.10	14.00	1.63	13.90	1.52	4.45	0.75	4.16	0.55	128.60	21.58	124.90	18.77
	0.15	7.00	0.82	7.10	0.88	1.67	0.32	1.65	0.34	48.80	7.66	50.60	9.56
	0.20	3.90	0.57	3.80	0.63	0.72	0.12	0.60	0.12	22.10	3.84	21.00	4.03
	0.25	2.90	0.32	2.90	0.32	0.50	0.07	0.41	0.05	15.20	2.20	15.20	2.20

Table F.12: TSGA-II Comparisons for MOAP ($m = 4$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
4MOAP5	0.05	26.70	10.44	26.60	10.45	11.03	7.45	10.35	7.08	1131.40	743.57	1122.20	745.08
	0.10	16.30	6.40	16.10	6.33	4.47	2.86	4.11	2.68	476.80	306.71	475.80	303.53
	0.15	9.70	3.47	9.60	3.53	1.86	0.98	1.63	0.92	190.60	102.79	189.20	108.53
	0.20	7.10	2.42	6.30	2.16	1.09	0.55	0.79	0.44	113.80	58.21	92.60	53.99
	0.25	4.90	1.45	4.30	1.57	0.59	0.27	0.43	0.25	60.40	30.07	48.80	30.29
4MOAP10	0.05	100.40	25.07	99.50	24.05	257.31	137.87	263.38	145.42	11365.00	5809.17	11170.60	5584.60
	0.10	26.50	6.13	26.20	4.71	22.19	10.30	20.66	7.73	1165.40	552.84	1113.20	431.96
	0.15	12.20	3.12	11.90	2.47	5.53	2.88	4.88	2.22	310.40	163.68	282.20	117.46
	0.20	6.70	1.49	6.40	1.26	1.68	0.67	1.46	0.53	100.80	38.20	93.00	33.05
	0.25	4.10	0.88	4.20	1.14	0.74	0.28	0.70	0.35	44.20	17.47	47.00	22.80
4MOAP15	0.05	134.00	52.74	138.50	55.31	668.63	495.34	687.92	494.61	21795.60	15383.83	23299.80	15931.96
	0.10	29.40	10.74	30.10	10.74	43.69	25.67	44.10	26.74	1574.60	916.54	1636.20	974.50
	0.15	11.20	3.55	11.20	3.91	6.95	3.17	6.67	3.13	266.40	117.86	265.40	121.17
	0.20	6.20	1.99	6.10	2.13	2.32	0.92	2.18	1.08	96.40	40.49	95.80	45.65
	0.25	3.80	1.14	4.10	1.20	0.97	0.38	0.98	0.37	39.80	16.12	45.40	17.33

Table F.13: TSGA-II Comparisons for MOAP ($m = 5$)

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
5MOAP5	0.05	35.70	5.89	35.30	5.50	43.74	14.41	44.49	21.38	4467.80	1423.35	4365.00	1405.62
	0.10	20.30	3.74	20.10	3.57	14.12	5.42	12.87	4.83	1524.60	581.19	1487.10	565.10
	0.15	11.70	1.77	11.60	1.78	4.86	1.37	4.30	1.22	512.00	150.90	503.00	145.94
	0.20	8.00	1.49	7.90	1.52	2.30	0.83	2.12	0.79	249.10	94.38	243.70	95.98
	0.25	5.60	1.43	5.70	1.34	1.14	0.50	1.07	0.43	122.30	56.63	126.20	54.54
5MOAP10	0.05	272.20	62.55	270.30	62.13	7376.14	5066.06	7113.76	4990.02	256990.30	162147.28	253655.40	159486.84
	0.10	56.60	12.84	54.30	11.14	303.68	177.52	323.63	177.87	12793.70	6761.05	11720.00	5524.03
	0.15	20.90	4.04	20.70	3.89	36.95	19.56	43.51	24.97	1837.10	879.57	1847.50	908.13
	0.20	10.90	1.37	10.40	1.35	8.76	3.11	8.78	3.33	485.00	170.97	436.50	124.55
	0.25	7.00	1.15	6.10	1.20	3.46	1.53	2.81	1.38	203.50	69.31	156.80	69.01

Table F.14: TSGA-II Comparisons for MOMIP Experiments

Problem	α	Cardinality				CPU Time (secs)				Number of Models Solved			
		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)		TSGA-II (orig.)		TSGA-II (eq.wt.)	
		Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
3MIKP100	0.05	47.30	11.00	47.40	11.11	10.09	3.43	9.49	3.42	705.60	260.46	709.30	260.90
	0.10	12.90	2.42	12.90	2.56	1.54	0.41	1.46	0.41	104.50	29.04	106.10	30.54
	0.15	6.20	0.92	6.00	0.94	0.58	0.13	0.51	0.10	38.50	8.62	36.50	8.18
	0.20	3.70	0.48	3.70	0.48	0.31	0.06	0.28	0.05	18.60	3.37	18.60	3.37
	0.25	3.00	0.47	2.90	0.32	0.25	0.05	0.20	0.03	14.20	2.94	13.50	2.12
4MIKP40	0.05	167.80	71.30	166.80	70.75	312.82	210.27	298.13	207.34	27208.00	18091.66	27217.80	18729.76
	0.10	31.00	9.80	29.60	9.64	16.92	10.20	14.38	8.90	1515.00	933.00	1362.20	851.22
	0.15	11.60	3.06	11.90	3.70	3.23	1.70	2.95	1.67	262.40	145.23	272.60	161.94
	0.20	6.30	1.70	6.60	1.71	1.04	0.47	1.11	0.50	88.00	41.22	100.40	45.65
	0.25	4.60	0.97	4.50	0.85	0.62	0.24	0.55	0.22	52.60	21.88	49.80	20.27
5MIKP20	0.05	206.70	210.07	207.00	217.84	3135.88	6895.00	3103.46	7147.06	244550.50	526726.66	253605.40	572271.84
	0.10	39.00	31.66	39.10	30.86	99.03	198.39	95.93	190.90	8455.20	16563.03	8666.10	16829.32
	0.15	14.60	9.71	15.10	10.62	13.06	18.35	14.33	20.63	1177.10	1643.77	1350.80	1942.35
	0.20	7.90	4.46	7.40	4.09	3.45	4.36	2.80	3.63	315.10	404.18	267.20	346.96
	0.25	4.80	1.93	5.00	2.49	1.07	0.70	1.21	1.16	94.70	65.43	117.00	113.29