ISSUES IN SELECTING A REPRESENTATIVE SET FOR MULTI-OBJECTIVE INTEGER PROGRAMS

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

BY

SAMİ SERKAN ÖZARIK

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

SEPTEMBER 2017

Approval of the thesis:

ISSUES IN SELECTING A REPRESENTATIVE SET FOR MULTI-OBJECTIVE INTEGER PROGRAMS

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

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

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

Assist. Prof. Dr. Banu Lokman
Supervisor, **Industrial Engineering Department, METU** ______________

Prof. Dr. Murat Köksalan
Co-supervisor, **Industrial Engineering Department, METU** ______________

Examining Committee Members:

Prof. Dr. Yasemin Serin
Industrial Engineering Department, METU ______________

Assist. Prof. Dr. Banu Lokman
Industrial Engineering Department, METU ______________

Assist. Prof. Dr. Mustafa Kemal Tural
Industrial Engineering Department, METU ______________

Assist. Prof. Dr. Bahar Çavdar
Industrial Engineering Department, METU ______________

Assist. Prof. Dr. Ceren Tuncer Şakar
Industrial Engineering Department, Hacettepe Uni. ______________

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: SAMİ SERKAN ÖZARIK

Signature :

ABSTRACT

ISSUES IN SELECTING A REPRESENTATIVE SET FOR MULTI-OBJECTIVE INTEGER PROGRAMS

ÖZARIK, Sami Serkan

M.S., Department of Industrial Engineering

Supervisor : Assist. Prof. Dr. Banu Lokman

Co-Supervisor : Prof. Dr. Murat Köksalan

September 2017, 67 pages

Multi-objective Integer Programs (MOIPs) have many areas of application in real life since it allows the decision makers to consider conflicting objectives simultaneously. However, the optimal solution is not unique for MOIPs and the number of nondominated points of multi-objective integer programs increases exponentially with the problem size. Therefore, finding all nondominated points is computationally hard and not practical for the decision maker. Instead of generating all nondominated points, it is reasonable to generate a set of points that represents the nondominated set with a desired quality level. In this thesis, we develop algorithms to generate representative sets for different MOIPs using a new quality measure that considers the distribution of points over the nondominated set. We first introduce a density measure and analyze typical distributions of nondominated points for different MOIPs. We then develop an approach that approximates the nondominated set, categorizes the approximate nondominated set into regions based on their estimated densities and generate distribution-based representative sets.

Keywords: Nondominated point, representative point, quality measures, multi-objective integer programs, distribution-based representative set, density-based quality measure

ÖZ

ÇOK AMAÇLI TAM SAYI PROBLEMLERİNDE TEMSİLCİ KÜMESİ SEÇİMİ

ÖZARIK, Sami Serkan
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Banu Lokman
Ortak Tez Yöneticisi : Prof. Dr. Murat Köksalan

Eylül 2017 , 67 sayfa

Karar vericilere birbiri ile çelişen farklı amaçların birlikte değerlendirebilinmesini sağlayan Çok Amaçlı Tamsayı Programları (ÇATP), gerçek yaşamda birçok uygulama alanına sahiptir. Bununla birlikte, ÇATP'ler için optimal çözüm tek değildir ve çok amaçlı tamsayı programlarındaki baskın noktalarının sayısı, problem büyüklüğü ile birlikte üssel olarak artmaktadır. Bu nedenle tüm baskın noktaların bulunması, karar verici için hesaplanması zor ve pratik olmayan bir yöntemdir. Tüm baskın noktaları üretmek yerine, belirli kalite ölçütlerini sağlayan baskın noktalar kümesi üretmek uygulanabilir bir yöntemdir. Bu tezde, baskın noktaların uzaydaki dağılımını da dikkate alan yeni bir kalite ölçüsüne göre, baskın nokta kümesini iyi temsil edecek baskın noktalar üreten algoritmalar geliştirilmektedir. Bu kapsamda, öncelikle yoğunluk ölçüsü tanımlanmakta ve farklı ÇATP'ler için baskın noktaların tipik dağılımları analiz edilmektedir. Ardından, baskın noktaların bulundukları bölgeleri yaklaşık olarak tanımlayan, yoğunluk ölçüsüne göre kategorilere ayıran ve buna göre temsilci baskın noktalar üreten bir yaklaşım geliştirilmektedir.

Anahtar Kelimeler: Baskın nokta, temsili baskın nokta, kalite ölçüleri, çok amaçlı tamsayı programları, dağılım temelli temsilci kümesi yoğunluk temelli kalite ölçütü

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisors Banu Lokman and Murat Köksalan for their brilliant guidance and comments throughout this study. Without their guidance, it would not be possible to finish my studies in such a short time.

I am also grateful to all my professors and friends at METU Industrial Engineering Department for their kindness.

I would also like to express my gratitude to Scientific and Technological Research Council of Turkey (TÜBİTAK) for the supports provided during my graduate study. This thesis was supported by TÜBİTAK 1001 Program, Project No: 215M844.

I would like to thank all my friends at ASELSAN Research Center for their help and for being such supportive and cheerful friends. I am particularly grateful for the assistance given by Erhan Gündoğdu, Veysel Yücesoy and Alper Güngör.

My deepest thanks go to my sister Nurdan Özarık Göksu and her husband Emrah Göksu. They have offered invaluable support and humor during my studies. I always knew that they believed in me in every decision I had made and wanted the best for me.

Finally, I must express my very profound gratitude to my parents, Ahmet and Zuhal, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Their support and encouragement was worth more than I can express on paper. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DM	Decision Maker
MCDM	Multi-Criteria Decision Making
MILP	Mixed Integer Linear Program
MOAP	Multi-Objective Assignment Problem
MOCO	Multi-Objective Combinatorial Optimization
MOEA	Multi-Objective Evolutionary Algorithm
MOIP	Multi Objective Integer Program
MOKP	Multi-Objective Knapsack Problem
MOLP	Multi-Objective Linear Program
MOMIP	Multi-Objective Mixed-Integer Program
MOSPP	Multi-Objective Shortest Path Problem
MOSTP	Multi-Objective Spanning Tree Problem

CHAPTER 1

INTRODUCTION

In real life problems, Decision Makers (DMs) are usually faced with multiple objectives while solving problems. Most of the time, there is a trade-off between these objective that is to say, in order to gain more from one of the objectives, the DM should sacrifice from another. This situation in multi objective decision making problems, leads to a set of solutions which provide almost equal benefits without being able to definitely dominate any other solution in the set. These points in the solution space are called as "nondominated points" and the set generated by them is called "efficient frontier".

In general a MOIP problem can be defined as follows:

$$\textbf{\textit{"Max"}} z = f(x), \text{ subject to } \boldsymbol{x} \in \boldsymbol{X} \tag{1.1}$$

where $f(x) = (z_1(x), z_2(x), ..., z_m(x))$ is in m dimensions, $\boldsymbol{x}$ is a decision variable vector and $\boldsymbol{X}$ is the feasible decision space, $\boldsymbol{X} \subseteq \mathbb{Z}^p$.

A feasible decision point x_i is said to be an *"efficient solution"* if there is not any x_j which satisfies;

- $z_k(x_j) \geq z_k(x_i) \; \forall k \quad and \quad z_k(x_j) > z_k(x_i) \; for\, at\, least\, one\, k.$

The image of x_i in the objective space, $f(x_i)$, is said to be *"nondominated point"*. If there exists any , x_j which satisfies the above condition, the point x_i is said to be an *"inefficient solution"* and $f(x_i)$ is said to be *"dominated point"*.

An *"ideal point"*, $z^{IP} = (z_1^{IP}, z_2^{IP}, ..., z_m^{IP})$, consists of the best values which a point can take at each objective separately. Here the term "best" is variant according to the nature of the problem. Minimum or maximum values at each objective in the efficient set may be preferred for different problems. With other words, the ideal point can be found by;

- either maximizing

  e.g., $z_i^{IP} = \max_{x \in X}(z_i(x)), i = 1, 2..., m$ for a maximization type problem,

- or minimizing

  e.g., $z_i^{IP} = \min_{x \in X}(z_i(x)), i = 1, 2..., m$ for a minimization type problem each objective individually with respect to the problem type.

Unlike the ideal point, *"nadir point"* $z^{NP} = (z_1^{NP}, z_2^{NP}, ..., z_m^{NP})$, comprises the worst objective values in the efficient set, X_E. That is;

- for a maximization type problem, $z_i^{NP} = \min_{x \in X_E}(z_i(x)), i = 1, 2..., m$
- for a minimization type problem, $z_i^{NP} = \max_{x \in X_E}(z_i(x)), i = 1, 2..., m.$

As the size of the problem increases in the MOIPs, it is difficult to find any nondominated solution. For this reason, many intuitive, metaheuristic and heuristic approaches have been developed for MOIPs. Ehrgott and Gandibleux (2004) [8] examine these algorithms developed specifically for Multi-Object Combinatorial Optimization (MOCO) problems, which are special types of MOIPs, while Ruzika and Wiecek (2005) [24] provide an overview of multi-objective optimization problems.

Recently, effective algorithms have been developed to find all nondominated points; but it is both difficult and impractical to find all the nondominated points and present them to the DM. In all of the studies done by Lokman and Köksalan (2013) [15], Mavrotas and Florios (2013) [18], Kırlık and Sayın (2014) [11], Özlen et al. (2014) [20] and Dächert and Klamroth (2015) [4] it is stated that as the number of objectives increase, computational complexity when identifying the nondominated set increases substantially too. The number of nondominated points may be exponential in the problem size and in this case, finding all nondominated points may become highly difficult. Therefore, instead of finding all the nondominated points, finding a subset that best represents the entire nondominated set of points based on certain quality measures is important for MOIPs.

In this thesis, we develop algorithms to generate representative sets for different MOIPs using a new quality measure that considers the distribution of points over the nondominated set. Working with a representative set for the nondominated points set might be an easier and practical approach since it is usually very time and effort consuming to generate and work on all nondominated points of a MOIP. So far, there have been some algorithms developed in order to form representative point sets that ensure certain quality measures. It is observed that many alternative representative sets may satisfy existing performance measures equally well with these algorithms. Thus, we first introduce a density-based quality measure and analyze typical distributions of nondominated points for different MOIPs. We then develop an approach that approximates the nondominated set and categorizes the points in the approximate nondominated set into regions based on their estimated densities and generate distribution-based representative sets.

In Chapter 2, we will review the exact and heuristic approaches in the literature to solve MOIP problems. In Chapter 3, we introduce a density measure and analyze the distribution properties of MOIPs. In Chapter 4, we present an approach that estimates the distribution of the points on the nondominated set. We develop an algorithm to generate distribution-based representative sets and present the results of our experiments in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

MOIPs are very useful for many areas of application such as capital budgeting, facility location, scheduling, transportation and network design problems. However, the literature is rather limited when compared with the MOLPs since the introduction of discrete variables make these problems hard to solve. Although there exists supported nondominated points that could be obtained using a weighted sum problem, Ehrgott and Gandibleux (2000) [7] note that there is an exponential increase in the number of unsupported nondominated points for MOCO problems, special MOIPs. They exhibit a thorough survey of exact and heuristic techniques to tackle MOCO problem issues.

In the literature, most of the studies that generate the whole nondominated point set focus on the bi-criteria case. There are very few studies which identify all nondominated points for a MOIP with more than two objectives. The method proposed by Sylva and Crema (2004) [26] identify a nondominated point at each iteration by adding binary variables and linear constraints to the model for each non- dominated point found in the previous iterations when number of objectives in the problem can be more than two. In this way, calculation of the following nondominated point tends to be considerably harder as the number of nondominated points discovered increases.

Ehrgott (2006) [6] examines scalarization strategies for MOIP problems and brings up that it can be computationally hard to create all nondominated points by utilizing the current scalarization strategies. In light of this study, he proposes the method with elastic constraints that consolidates the benefits of weigted sum and epsilon constraint scalarization methods. With this, one can decrease the computational efforts to solve scalarized problems and identify all the points in nondominated point set.

Laumans et al. (2006) [14] develops an algorithm based on methods for generating or approximating the Pareto set of multi-objective optimization problems by solving a sequence of constrained single-objective problems. They discuss the drawbacks of the original epsilon-constraint methods by showing that the running times of the original epsilon-constraint method is exponential in the problem size, although the size of the Pareto set is growing only linearly.

Özlen and Azizoğlu (2009) [19] propose an improvement (by reducing the number of models solved) over the classical epsilon-constraint method which can generate all nondominated points for any number of objectives. The method reduces the number of models solved significantly compared to the classical epsilon-constraint method. Özlen et al. (2014) [20] introduces an improved recursive algorithm to generate the set of all nondominated objective vectors for MOIPs. Improvement is done by storing a list of already solved subproblems and making a search in this list before solving any new subproblem. This way it is possible to further reduce the number of IP models to be solved. As their experiments show, the improvement becomes more significant as the problems grow larger in terms of the number of objectives.

Identifying all supported nondominated points is not simple for more than two objectives as it is for bi-objective problems. Özpeynirci and Köksalan (2010) [21] and Przybylski et al. (2010) [23] develop algorithms based on changing weights systematically to generate all extreme supported nondominated points for multi-objective mixed integer programming problems with more than two objectives.

Lokman and Köksalan (2013) [15] develop an exact algorithm to generate all nondominated points of MOIPs. The algorithm iteratively identifies the nondominated points and exclude the regions that are dominated by the previously-identified nondominated points. They provide an algorithm which generates new points by solving models with additional binary variables and constraints. The other algorithm employs a search procedure and solves a number of smaller models to find the next point without using any additional binary variables or constraints. Both algorithms guarantee to find all nondominated points for any MOIP. Computational experiments show that the algorithm performs better compared to Sylva and Crema (2004) [26].

Kırlık and Sayın (2014) [11] suggest an algorithm which generates the entire nondominated set for multi-objective discrete optimization problems. The algorithm uses two stage epsilon-constraint formulation to generate nondominated points. The method conducts searches in rectangles and later on updates them. The comparison of the algorithm to previous studies is also given for MOKPs and MOAPs.

As a recent approach, Dächert and Klamroth (2015) [4] propose a search algorithm. They present a procedure for finding the entire nondominated set of tri-criteria optimization problems for which the number of sub-problems to solve is bounded linearly depending on the number of nondominated points. The approach updates iteratively the search region that, given a subset of nondominated points, describes the area in which additional nondominated points may be located.

In spite of the fact that there exist effective exact techniques to solve MOIPs, they are often not practical when the size of the problem gets larger. Hence, approximation techniques for multi-objective problems are employed most of the times. Ehrgott and Gandibleux (2004) [8] present a review of approximate methods for MOCO problems.

Different performance measures have been defined in evaluating the quality of the representative set (Sayın, 2000 [25]; Wu and Azarm, 2001 [1]; Zitzler et al., 2003 [28]). Sayın (2000) [25] proposes coverage, uniformity and cardinality measures. While coverage measure should be decreasedin order to represent all parts of the efficient frontier, larger value for the uniformity measure is also wanted not to generate the nondominated representative points close to each other in the objective space. Faulkenberg and Wiecek (2009) [9] evaluate quality measures that could be used for general multi-objective optimization problems in three different groups, as suggested by Sayın (2000) [25].

Sylva and Crema (2007) [27] and Masin and Bukchin (2008) [17] propose algorithms that find representative sets for nondominated points in MOIPs and MOMIPs. In these algorithms, at each step the worst represented nondominated point is found. With other words, the nondominated point which has the largest representation error is found. Both algorithms use similar mathematical models in order to find new nondominated points such that the distance from the yet-known nondominated point set is maximized.

Boland et al. (2016) [2] use a different measure to evaluate the distribution of the nondominated points represented. They present a new criterion space search method, the L-shape search method, for finding all nondominated points of a tri-objective integer program. After that in order to assess the performance of representative sets, the closest representative for each yet unknown nondominated point is determined. Then, the number of nondominated points represented by each representative is found, and the average number of points represented and the deviation from the average are calculated. As a measure of quality, a distribution measure is used, and when this ratio is small, it is said that the representative cluster has better distribution.

Ceyhan (2014) [3] proposes computational improvements from Sylva and Crema (2007) [27] and Masin and Bukchin (2008) [17] by employing decomposition methods and search alghorithms. In none of these studies, distribution characteristics of nondominated point sets are considered. Therefore, in this thesis firstly distribution characteristics of nondominated point sets are taken into consideration for further analysis. Afterwards, a new density based performance measure is going to be presented in order to assess the performance of representative points sets.

CHAPTER 3

DISTRIBUTION PROPERTIES OF MULTI-OBJECTIVE INTEGER PROGRAMS

In this chapter, a brief information about existing performance measures will be provided. The results of several mathematical models used during experiments will show the pitfalls of existing performance measures and create a basis for a density based performance measure.

Before starting any further analysis, the method that nondominated points are generated with will be provided. We conduct experiments on MOKP, MOAP, MOSPP, and MOSTP problems for which all nondominated points are generated using the algorithm of Lokman and Köksalan (2013) [15] for comparison purposes. The instances are generated as it is in Köksalan and Lokman (2009) [12]. Mathematical models that generate the nondominated points in three objective space for MOKPs, MOAPs, MOSPPs and MOSTPs are provided in Appendix A with the values of parameters used.

3.1 Representing Nondominated Set with Existing Quality Measures

All efficient solutions of a problem must be well represented in order to improve coverage measures. This study mostly focuses on the quality measures which include "coverage error," "uniformity" and "cardinality" proposed by Sayın (2000) [25] or "diversity measure" proposed by Masin and Bukchin (2008) [17].

At the coverage gap calculations, unlike Sayın (2000) [25], we ignore the differences in the objective values for which the nondominated point is worse than its representative point. Thus, for a nondominated set $\boldsymbol{S}$ and a representative set $\boldsymbol{R} \subseteq \boldsymbol{S}$ the coverage gap α of a maximization type MOIP problem can be calculated as follows,

$$\alpha = \max_{x \in S}(\min_{y \in R}(\max_{1 \leq i \leq M}(x_i - y_i))) \tag{3.1}$$

Here, the model assigns a representative point, $y \in \boldsymbol{R}$, to each nondominated point, $x \in \boldsymbol{S}$, based on a Tchebycheff-based distance metric. Therefore, coverage gap, α, corresponds to the distance of the worst represented point to its representative. In an objective, if a representative point is better than the nondominated point it represents, then the difference in this objective value should not be considered in the calculation of the representation error.

In order to analyze the properties of the representative sets, we initially solve models using existing quality measures. For comparison purposes, we find optimal representative sets of problems (for which all nondominated points are available to us) according to different performance measures. The following mathematical model (which will be called as, ***Model-Gap***, for the rest of the thesis) can be used in order to select K representative points from the set of all nondominated points that minimize the coverage gap. The objectives for ***Model-Gap*** should be maximization type (the larger the better).

Indices:

i,j = number of points in nondominated point set (1,2,...,N)

p = number of objectives of the problem (1,2,...,M)

Decision Variables:

$$x_i = \begin{cases} 1, & \text{if point i is selected to be representative point} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if point j is represented by point i} \\ 0, & \text{otherwise} \end{cases}$$

Parameters:

z_{ip} = the p'th objective value of point i

K = number of representative points

Model-Gap:

$$
\begin{aligned}
\min \ & \alpha && && (3.2)\\
\text{s.t. } & \alpha && \geq (z_{jp} - z_{ip}) y_{ij} && \forall i, j, p && (3.3)\\
& N x_i && \geq \sum_{j=1}^{N} y_{ij} && \forall i && (3.4)\\
& \sum_{i=1}^{N} y_{ij} && = 1 && \forall j && (3.5)\\
& \sum_{i=1}^{N} x_i && = K && && (3.6)\\
& x_i, y_{ij} && \in \{0, 1\} && \forall i, j && (3.7)
\end{aligned}
$$

Model-Gap finds the optimal subset that represents the nondominated set with minimum coverage gap value. In this model, we scale the objectives such that all objective values are to be between 0 and 100.

In ***Model-Gap***, only coverage gap measurement has been taken into account. This means there might be some alternative solutions to the model that may give better solutions in terms of different performance measures while keeping coverage gap as the same.

In order to take the other performance measures into consideration, density-based mathematical models that take the distribution properties into account are developed. For each representative point to be selected, we define an upper bound for the number of nondominated points it will represent. This way, from the dense parts of nondominated solution sets, more representative points can be selected and more information from dense parts of the set can be generated to present to the decision maker.

Here, the definition of "density" implies "the percentage of all nondominated solutions that are represented by a representative point". To illustrate; let's assume a nondominated points set with cardinality 100, if any representative point represents 15 nondominated points, the density around the representative point is said to be 15 percent.

By setting an upper limit for all representative points, ***Model-Gap*** can be improved in terms of other performance measures. ***Model-Gap*** can be modified easily by just changing the Constraint 3.4 as stated below in order to set a density based upper limit;

$$dNx_i \geq \sum_{i=1}^{N} y_{ij} \qquad \forall i \tag{3.8}$$

where ***d*** is an upper density limit for any representative point (i.e. 18%) and ***N*** is cardinality of the nondominated points set. The new model,***Model-Gap-Cap*** , finds the optimal representative set that minimizes the coverage error when the number of nondominated points that can be represented by any representative point is limited. If d is not large enough, Model-Gap-Cap turns out to be infeasible since some nondominated points will not be represented due to the capacity restriction. When ***d*** gets closer to 1 (i.e. 100%), ***Model-Gap-Cap*** behaves like there is no upper limit. Then ***Model-Gap-Cap*** would work in the same way as ***Model-Gap***. Because of that, ***d*** should be chosen a small number without violating the condition mentioned above.

Both original and modified models are run with the same nondominated points set and some of the results are as follows:

Table 3.1: Comparison of two suggested models

	Coverage Gap Values		
Number of Representative Points	***Model-Gap***	***Model-Gap-Cap***	
K=10	9.26	d=12	12.61
		d=14	10
		d=16	9.26
		d=20	9.26

From Table 3.1 it can be seen that as ***d*** increases, ***Model-Gap-Cap*** starts to behave like ***Model-Gap***. Also it can be said that the number of represented points per representative point can be decreased at the expense of an increase in the coverage gap.

Both ***Model-Gap*** and ***Model-Gap-Cap*** need number of representative points as an input parameter. However, we observe that there are instances for which the coverage gap value does not change with small decreases in the number of representative points. Therefore, it may sometimes be a wise idea to let the model to find the number of representative points since we would prefer to have small representative sets to save computational effort and make it practical for the DM. We next develop a new model, (***Model-Rep-Gap-Dens***) as follows:

Indices:

i,j = number of points in nondominated point set (1,2,...,N)

p = number of objectives of the problem (1,2,...,M)

Decision Variables:

$$x_i = \begin{cases} 1, & \text{if point i is selected to be representative point} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if point j is represented by point i} \\ 0, & \text{otherwise} \end{cases}$$

$r =$ number of representative points

$d =$ density around the most dense representative point

Parameters:

$z_{ip} =$ the p'th objective value of point i

$gap =$ maximum accepted coverage gap value

$w_{gap} =$ the weight of coverage gap in the objective function

$w_{rep} =$ the weight of number of representative points in the objective function

$w_{den} =$ the weight of density in the objective function

Model-Rep-Gap-Dens:

$$\min \; w_{rep}r + w_{gap}\alpha + w_{den}d \tag{3.9}$$

$$
\begin{aligned}
\text{s.t.}\quad & \alpha && \geq (z_{jp} - z_{ip}) y_{ij} && \forall i, j, p && (3.10)\\
& Nd && \geq \sum_{i=1}^{N} y_{ij} && \forall i && (3.11)\\
& Nx_i && \geq \sum_{i=1}^{N} y_{ij} && \forall i && (3.12)\\
& \sum_{i=1}^{N} y_{ij} && = 1 && \forall j && (3.13)\\
& \sum_{i=1}^{N} x_i && = r && && (3.14)\\
& \alpha && \leq gap && && (3.15)\\
& x_i, y_{ij} && \in \{0, 1\} && \forall i, j && (3.16)
\end{aligned}
$$

Model-Rep-Gap-Dens aims to find the optimal representative set that minimizes the number of representative points that satisfy the maximum coverage gap value. Coverage gap value is also included in the objective function as a secondary objective since it is possible to find different values of coverage measure for a specific value of cardinality measure. Among the alternative optimal representative sets with minimum cardinality value and coverage gap value, we favor the ones that minimize the maximum number of points to be represented by a single representative point. It allows the DM to have representative points for each of which the number of points they represent are similar. To achieve these goals, we choose an objective function as in 3.9 where the weight parameters are selected as follows:

$$
w_{rep} \gg w_{gap} \gg w_{den} \tag{3.17}
$$

In order to show the effect of density in terms of better representation, two versions of ***Model-Rep-Gap-Dens*** will be compared. Comparison has been done with the same nondominated solution set that is used for previous models. The first version of the model takes the weight of the density measure as 0. That is to say that, densities around the representative points are not taken into the consideration. The weights of each measure in the objective function can be seen in Table 3.2.

Model-Rep-Gap-Dens is run for 15 times for both of the versions. At each run the

Table 3.2: Weight parameters of **Model-Rep-Gap-Dens**

Model-Rep-Gap-Dens		
Weights	Version 1	Version 2
w_{rep}	10^6	10^6
w_{gap}	10^3	10^3
w_{den}	0	1

parameter gap is updated with previous run's coverage gap minus an ϵ value (0.001). Then, the minimum number of representative points to satisfy the upper level of accepted coverage gap measure is found.

The first thing that needs attention for both versions on Table 3.3 is the number of representative points on runs 8 and 9. It shows that in order to gain from coverage gap at run 9, one additional representative point is not enough. So in order to come up with a coverage gap value better than "10", two additional representative points are needed. This is where ***Model-Rep-Gap-Dens*** is more useful rather than ***Model-Gap*** and ***Model-Gap-Cap***.

We also observe that the same coverage gap value obtained with 11 representative points can also be obtained with 10 representative points. When two versions are compared, it is easy to see there are some alternative solutions to the model for certain coverage gap and number of representative points. Both versions find the same coverage gap and number of representative points when inputed with the same gap parameter. The only difference between the versions is that, Version 2 tries to lower the maximum number of represented points for any representative point, whereas Version 1 does not. The difference of both versions can be understood better with Figure 3.1.

In Figure 3.1, numbers placed below the bars show the number of representative points at each run. The portions in the bars show the percentages of all nondominated points represented by each representative point. The largest portion (the maximum number of points represented by any representative point, defined as d) is placed at

Table 3.3: Comparison of two versions of **Model-Rep-Gap-Dens**

No of Runs	Version 1			Version 2		
	Number of Representatives	Coverage Gap	Max d	Number of Representatives	Coverage Gap	Max d
1	1	37.83	52	1	37.83	52
2	2	28.52	27	2	28.52	26
3	3	23.04	22	3	23.04	20
4	4	16.30	18	4	16.30	15
5	5	14.81	18	5	14.81	12
6	6	12.88	24	6	12.88	11
7	7	12.61	30	7	12.61	10
8	8	10	24	8	10	12
9	10	9.26	8	10	9.26	8
10	11	9.15	8	11	9.15	8
11	12	8.89	14	12	8.89	6
12	13	7.83	14	13	7.83	9
13	14	6.67	15	14	6.67	7
14	15	6.52	13	15	6.52	5
15	16	5.76	10	16	5.76	6

the bottom of the bar and the smallest portion is placed at the top of the bar. Since we minimize the maximum number of points to represent, each representative point represents almost equal number of nondominated points, and hence the more equal sized portions obtained at each bar the better it is. This shows using a density based quality measure can improve the representation of nondominated points by assigning each representative point similar number of nondominated points. This can be interpreted as "equal representation" of each nondominated point.

Figure 3.1 and Table 3.3 show that; despite of the use of different quality measures (cardinality and coverage gap), alternative solutions may still exist. To break the ties and represent the nondominated point set better, a new quality measure is needed.

3.2 Distribution Properties of Multi-Objective Integer Programs

The existing quality measures do not consider the distribution properties of the nondominated points. Furthermore, our experiments show that there may have alternative

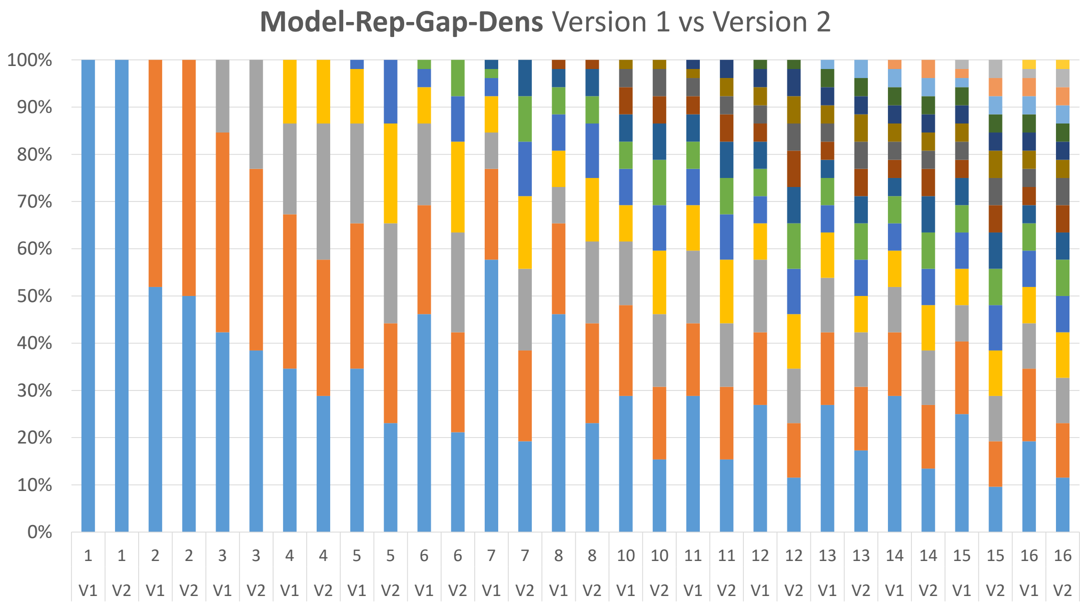

Figure 3.1: Model-Rep-Gap-Dens Version 1 vs. Version 2 - nondominated point distributions with respect to number of representatives

optimal subsets that minimize the coverage gap value and the number of representative points. In order to produce representative sets that also reflect the distribution characteristics of the nondominated set, we employ a density metric that is proposed by by Parzen (1962) [22]. We estimate the density function around any point using "Parzen Windows" which is a very commonly used non-parametric classification method. It is also used in pattern recognition, classification and identification applications (see Duda (1973) [5]).

Parzen Windows approach estimates the distribution of points in the space by calculating each point's contribution to the density measure around a point, "x", with a given kernel function. In fact, kernel function is used for interpolation to calcute the contribution of each nondominated point to the density measure around "x". Each point's contribution is consistent with its distance or L^p-norm to "x". In order to find the parzen windows estimate around the point "x" following formula is used:

$$P(x) = \frac{1}{Nh} \sum_{i=1}^{N} K \left(\frac{||x_i - x||_q}{h} \right) \tag{3.18}$$

where h is the windows size (the window width or bandwidth parameter that corresponds to the width of the kernel), q is the norm of the distance vector and $K(x)$ is a kernel function in p dimensional space.

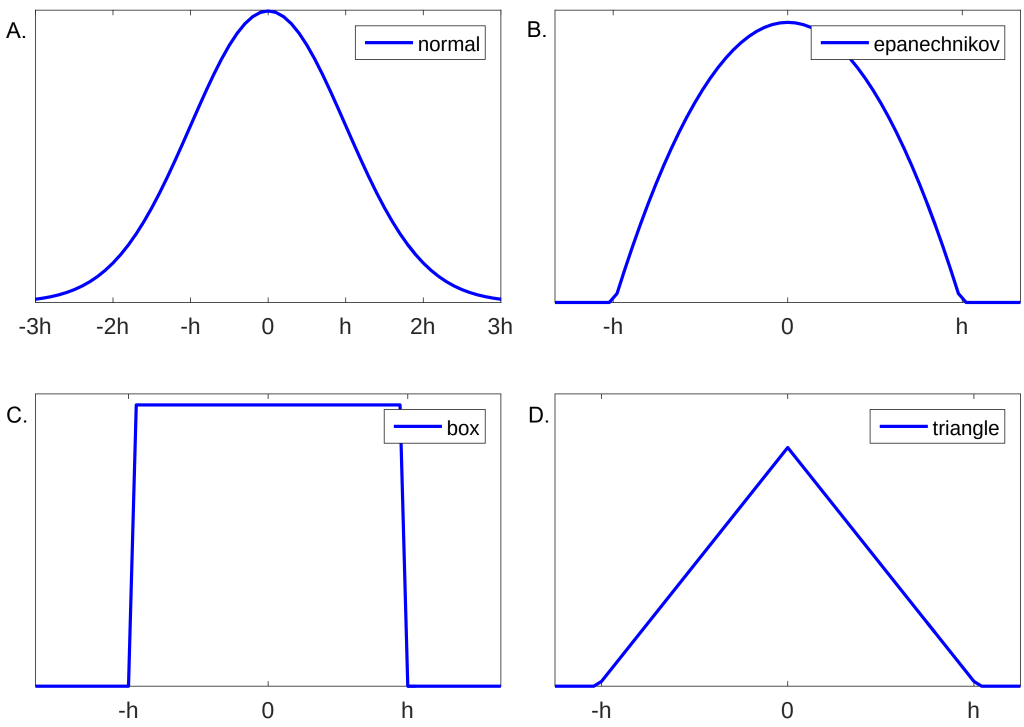

Figure 3.2: Available Kernels

Definition: A Kernel is a non-negative real-valued integrable function, $K(u)$, such that:

$$\int_{-\infty}^{+\infty} K(u)du = 1 \tag{3.19}$$

For most of the applications it is desirable to define the function to satisfy Equation 3.20.

$$K(u) = K(-u) \quad \forall u \tag{3.20}$$

Commonly used Kernels can be found in Figure 3.2.

CHAPTER 4

DENSITY ESTIMATION FOR MULTI-OBJECTIVE INTEGER PROGRAMS

4.1 Application of Parzen Windows in Multi-Objective Integer Programs

In this section, we will employ Parzen Windows method to analyze the distribution properties of MOIPs. As we did in our previous experiments, we scale the objective values such that the minimum and the maximum values of each objective is the same (i.e. objective values lie between 0 and 1). This not only allows comparison of distribution properties for different objectives in the same problem or comparison of distribution properties for two different problems but also is required for the calculations to make sense.

In the calculations of this study, $K(u)$ represents a point's contribution to the density measure according to its distance, u, to the reference point. The distance u can be calculated with $p-$norm distance functions. In general, the length of the distance vector between any two points in the M-dimensional real vector space is given by Euclidean norm. In preference based calculations of MCDM problems, Tchebycheff distance (L^{∞} or L^{p} while $p \to \infty$) is also being used quite often. Simply, $p-$norm or $L^{p}-$norm between points $X(x_1, x_2, ..., x_M)$ and $Y(y_1, y_2, ..., y_M)$ in M dimensional space can be calculated as follows:

$$
\begin{aligned}
||X - Y||_p &= \left(\sum_{i=1}^{M} (|x_i - y_i|)^p \right)^{1/p} \\
||X - Y||_{rec} &= \left(\sum_{i=1}^{M} (|x_i - y_i|)^1 \right)^{1}
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
||X-Y||_{euc} &= \left(\sum_{i=1}^{M}(|x_i-y_i|)^2\right)^{1/2} \\
||X-Y||_{tch} &= \lim_{p\to\infty}\left(\sum_{i=1}^{M}(|x_i-y_i|)^p\right)^{1/p} \\
&= max\,(|x_1-y_1|,|x_2-y_2|,...,|x_M-y_M|)
\end{aligned}
\tag{4.2}
$$

For certain values of q, distance functions are called with special names. These are;

- 1-norm is the norm that corresponds to the **rectilinear distance**,
- 2-norm is the norm that corresponds to the **euclidean distance**,
- maximum norm or $L^{\infty}-$norm is the norm that corresponds to the **Tchebycheff distance**.

The most basic Kernel function that can be used in any of MOIP problems is Uniform (Boxcar) function. Uniform function is zero except for a single interval where it is equal to a constant other than zero (see the plot at Figure 3.2.(C)). In general, Uniform function as a Kernel function can be shown as follows:

$$
K\left(\frac{||z_i-z||_p}{h}\right) = \begin{cases} 1, & \frac{||z_i-z||_p}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}
$$

Equation 4.3 indicates whether z_i is inside the window area (centered at z with width h) or not. If the point is closer to the reference than the distance of $h/2$, it contributes 1 unit, otherwise it does not contribute at all. Summing over all possible values of z_i as in the Equation 3.18, gives the density estimate around point z.

As an example, let's consider the case shown in Figure 4.1 and calculate each point's contribution to the reference point $R(0.50, 0.50)$. Large circle around R shows the border of 0.20 unit Euclidean distance. It represents a window centered at $R(0.50, 0.50)$ with width 0.4 units.

Points $A(0.55, 0.65)$ and $C(0.40, 0.45)$ lie in the circle, so according to Equation 4.3, both contribute 1 unit to the density measure around R. This result can also be seen

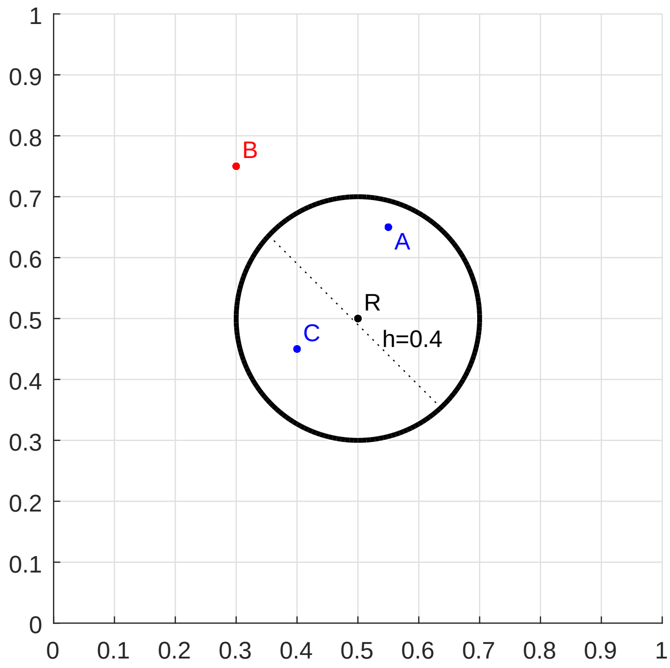

Figure 4.1: Kernel Example

when the Euclidean distances of both points are calculated as below:

$$
\begin{aligned}
||A-R||_2 &= \left((|x_A - x_R|)^2 + (|y_A - y_R|)^2\right)^{\frac{1}{2}} \\
&= \left((|0.55-0.50|)^2 + (|0.65-0.50|)^2\right)^{\frac{1}{2}} \\
&\approx 0.158 \\
\frac{||A-R||_2}{h} &\approx \frac{0.158}{0.40} \leq \frac{1}{2} \\
||C-R||_2 &= \left((|x_C - x_R|)^2 + (|y_C - y_R|)^2\right)^{\frac{1}{2}} \\
&= \left((|0.40-0.50|)^2 + (|0.45-0.50|)^2\right)^{\frac{1}{2}} \\
&\approx 0.112 \\
\frac{||C-R||_2}{h} &\approx \frac{0.112}{0.40} \leq \frac{1}{2}
\end{aligned}
\tag{4.4}
$$

Now let's consider the point outside of the window, $B(0.30, 0.75)$, and do the distance calculations:

$$\begin{aligned}
||B-R||_2 &= \left((|x_B-x_R|)^2+(|y_B-y_R|)^2\right)^{\frac{1}{2}} \\
&= \left((|0.30-0.50|)^2+(|0.75-0.50|)^2\right)^{\frac{1}{2}} \\
&= (0.1025)^{\frac{1}{2}} \\
&\approx 0.320 \\
\frac{||B-R||_2}{h} &\approx \frac{0.320}{0.40} \geq \frac{1}{2}
\end{aligned} \tag{4.5}$$

As seen in the calculations above, B does not contribute to the density measure around R. So far, the distance calculations between two points are done according to Euclidean distances. Tchebycheff distance calculations for the same points are as follows:

$$\begin{aligned}
||A-R||_\infty &= max\left(|x_A-x_R|,|y_A-y_R|\right) \\
&= max\left(|0.55-0.50|,|0.65-0.50|\right) \\
&= 0.15 \\
\frac{||A-R||_\infty}{h} &= \frac{0.15}{0.40} \leq \frac{1}{2} \\
\\
||B-R||_\infty &= max\left(|x_B-x_R|,|y_B-y_R|\right) \\
&= max\left(|0.30-0.50|,|0.75-0.50|\right) \\
&= 0.25 \\
\frac{||B-R||_\infty}{h} &= \frac{0.25}{0.40} \geq \frac{1}{2} \\
\\
||C-R||_\infty &= max\left(|x_C-x_R|,|y_C-y_R|\right) \\
&= max\left(|0.40-0.50|,|0.45-0.50|\right) \\
&= 0.10 \\
\frac{||C-R||_\infty}{h} &= \frac{0.10}{0.40} \leq \frac{1}{2}
\end{aligned} \tag{4.6}$$

If Tchebycheff distance was preferred in the calculations, then the window border in Figure 4.1 would look like a square with the length of each side equal to h. The corners of that square would be at $\{(0.3, 0.3), (0.7, 0.3), (0.3, 0.7), (0.7, 0.7)\}$. Again with Tchebycheff distance, one could say A and C contribute 1 unit to the density measure around R, whereas B does not lie in the window area, so it does not contribute to the density measure.

So far, Uniform function is used in the calculation of each point's contribution to the density measure around the reference point. One other mostly used Kernel Function is Triangular Kernel. The shape of the Triangular Kernel can be seen in the right bottom corner of Figure 3.2. Triangular Kernel allows nondominated points with different distances to have different contributions to the reference point's density measure. The contribution of each point on the density is linearly decreasing as the distance from the reference point increases. In general, Triangular function as a Kernel function is shown in Equation 4.7:

$$K\left(\frac{||z_i - z||_p}{h}\right) = \begin{cases} 1 - \frac{||z_i - z||_p}{h/2}, & \frac{||z_i - z||_p}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

Now, let's check the difference between Triangular Kernel and Uniform Kernel with the example previously shown in Figure 4.1:

$$\begin{aligned} \frac{||A - R||_\infty}{h} &= \frac{0.15}{0.40} \leq \frac{1}{2} \\ K\left(\frac{||A - R||_\infty}{h}\right) &= \begin{cases} 1 - \frac{||A - R||_\infty}{h/2}, & \frac{||A - R||_\infty}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \\ &= 1 - \frac{||A - R||_\infty}{h/2} \\ &= 1 - \frac{0.15}{0.80} = 1 - 0.1875 = \mathbf{0.8125} \end{aligned} \tag{4.8}$$

$$\begin{aligned} \frac{||C - R||_\infty}{h} &= \frac{0.10}{0.40} \leq \frac{1}{2} \\ K\left(\frac{||C - R||_\infty}{h}\right) &= \begin{cases} 1 - \frac{||C - R||_\infty}{h/2}, & \frac{||C - R||_\infty}{h} \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \\ &= 1 - \frac{||A - C||_\infty}{h/2} \\ &= 1 - \frac{0.10}{0.80} = 1 - 0.1250 = \mathbf{0.8750} \end{aligned} \tag{4.9}$$

Results shown in bold in Equation 4.8 and 4.9 are the contributions of A and C to the density measure around reference point R. Unlike Uniform Kernel, using Triangular Kernel for these points yielded different contributions to the density measure around R.

Now that each point's contribution is calculated, next step is finding the cumulative density measure around the reference point. This can be done by adding every nondominated point's contribution. In Equation 3.18, this summation is scaled with the number of nondominated points N and the window size h.

In our density calculations in this thesis, this scalarization is not necessary since after finding density values for different reference points, the sum of density measures in all reference points will be scaled to 1 and each reference point's scaled density measure will be found accordingly. Also, setting h to a very small number may lead to peeks in density measures of some reference points whereas, setting h to a very large number may cause a uniformity among density measures of reference points. For these reasons Equation 3.18 is revised as Equation 4.10 below. Calculation method of the scaled density measure is shown in Equation 4.11.

$$P(x_r) = \sum_{i=1}^{N} K\left(\frac{||x_i - x_r||_p}{h}\right) \tag{4.10}$$

$$\bar{P}(x_r) = \frac{P(x_r)}{\sum\limits_{r=1}^{R} P(x_r)} \tag{4.11}$$

where R is the number of reference points in M dimensional space.

Now let's consider the previous example again and assume each point as a reference point. It is possible to calculate the contribution of each point to the other points. After the same calculations are done with Triangular Kernel for A, B, C and R, following results are obtained:

As one can see from Table 4.1, the results are symmetric.

In order to understand the density distributions of different problems and to test the usability of the Parzen Windows method, we first generate equal-spaced reference points over the solution space. We then calculate the density values at each point.

Table 4.1: Contributions between points

$K(\|\|x_r\|\|_\infty)$	**R**	**A**	**B**	**C**
R	-	0.8125	0	0.8750
A	0.8125	-	0	0
B	0	0	-	0
C	0.8750	0	0	-

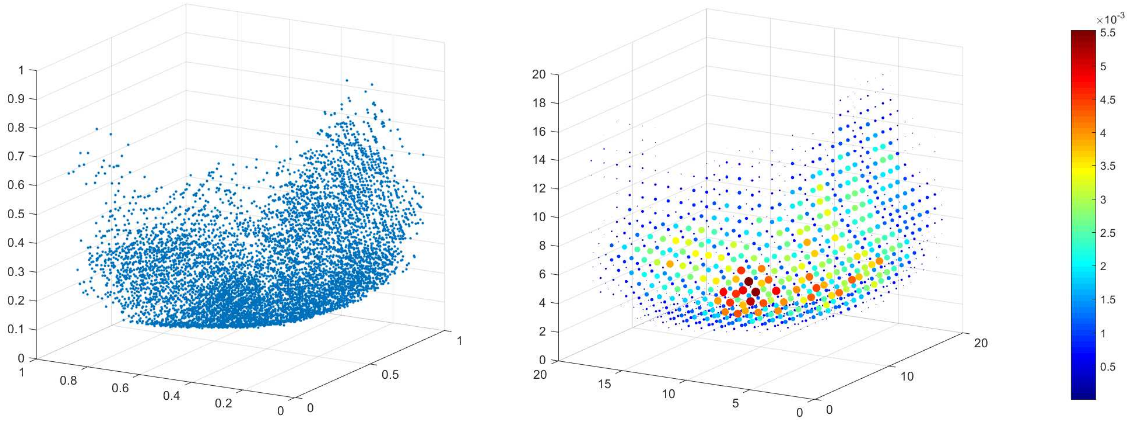

Figure 4.2: Left: scaled MOAP with 6573 nondominated points, Right: Density values at the reference points

To generate equal-spaced points, we first scale the objective values as in Köksalan and Lokman (2009)[12]: $\left(\frac{z_1 - z_1^{IP}}{z_1^{NP} - z_1^{IP}}, \frac{z_2 - z_2^{IP}}{z_2^{NP} - z_2^{IP}}, ..., \frac{z_M - z_M^{IP}}{z_M^{NP} - z_M^{IP}}\right)$ where $(z_1^{IP}, z_2^{IP}, ..., z_M^{IP})$ and $(z_1^{NP}, z_2^{NP}, ..., z_M^{NP})$ denote the ideal point and nadir point respectively corresponding to the problem.

After we scale each objective, we divide the range into 20 equally spaced intervals. The intersection points of these intervals create hypercubes in the solution space. A total of 8,000 hypercubes were created, each of which had a 0.05 units of edge length. We consider the center points of these hypercubes as reference points around which we calculate the density.

As an example, the distribution of the original nondominated points for a MOAP with 6573 nondominated points are shown on the left side of Figure 4.2. On the right side, density values are symbolized with the sizes and colors of the spheres. As the

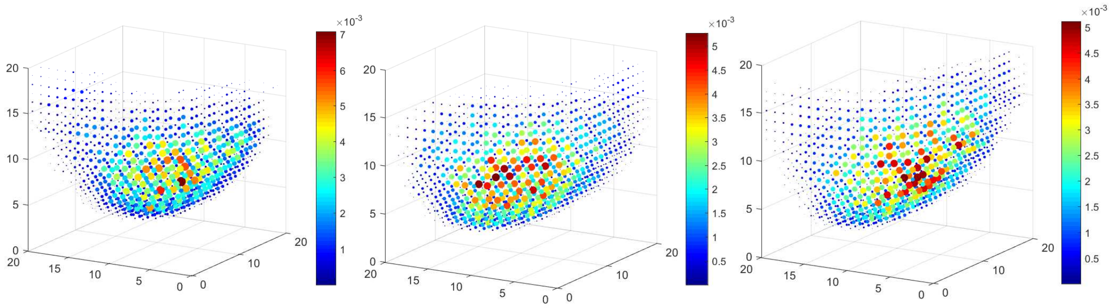

Figure 4.3: MOKPs with 5652, 6500 and 8288 nondominated points respectively

size of each sphere grows and the color changes from dark blue to dark red, density increases.

It is noteworthy that the density values of most of the reference points are quite small (zero or very close to zero). These representative points in some regions of the nondominated solution space become invisible since their size are very small relative to the others.

We also observe that there is an increase in density measure toward the center while density values approach zero or become negligible at the extreme points. These results are observed in different type of problems and different sized problems. Figure 4.3 demonstrates how the density values change over the solution space for different sized MOKPs and we observe common properties.

During the experiments, the effect of the problem type on the density values was investigated as well as the size. It can be seen in Figure 4.4 that similar sized problems in different types show similarities in density distribution. When Figure 4.4 is examined, it can be said that density of the nondominated points increases towards the center independently of the type of the problem, and density values approach zero when getting close to the extreme points.

In the studies described so far, reference points are placed in every part of the solution space and density values are calculated at these points. However, since the nondominated points set is studied on MOIP, there is no need to calculate the density values for the entire solution space. In both Figure 4.3 and Figure 4.4, it is observed that non-

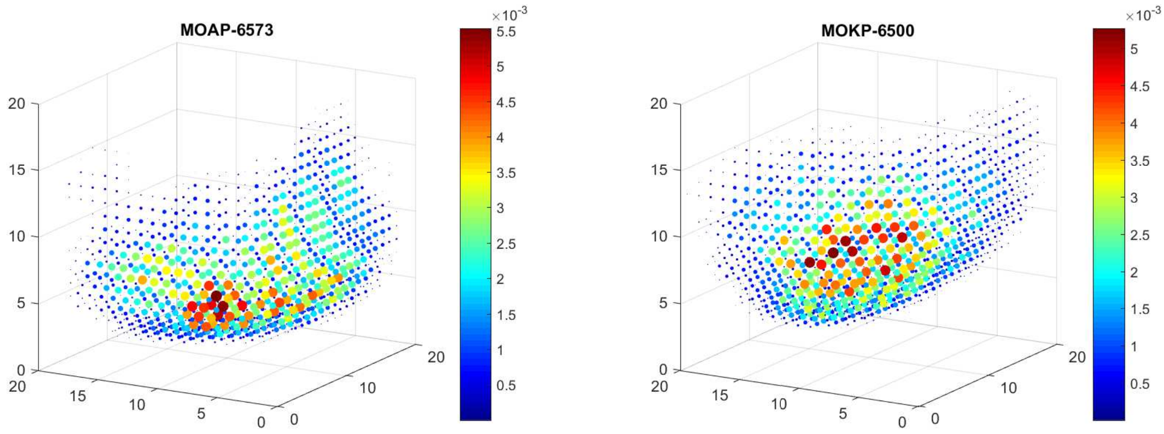

Figure 4.4: MOAP with 6573 and MOKP with 6500 nondominated points respectively

dominated points form an efficient frontier and the rest of the solution space is empty. Thus, density values at these regions are zero. For this reason, instead of calculating the density values over whole solution space, working with efficient frontier can save computational time by eliminating unnecessary calculations and help to avoid observation of non-zero density values on dominated or dominating (both empty) regions due to the large window sizes.

Because of the reasons specified above, density values should be calculated only for the regions that are expected to have nondominated points. To do that, we first approximate the possible locations of nondominated points.

4.2 Surface Fitting with Lp Functions (Lp-Fitting)

Köksalan (1999) [13] worked with L^p norms to estimate the nondominated frontiers of bicriteria scheduling problems. Later, Köksalan and Lokman (2009) [12] adapted this approach to approximate the nondominated frontiers of MOCO problems with any number of criteria. Lokman and Köksalan (2014) [16] defined preferred regions by approximating the nondominated set using a hyper-surface generated by L^p norms. Also, Karasakal and Köksalan (2009) [10] used L^p norms to estimate the shape of the nondominated frontier. With the help of the surface, they generated a set of points that represents the frontier. Computational experiments in these studies show that, using

L^p norms to approximate the nondominated frontiers performs quite well. Thus, in this study L^p norms are used in order to approximate the nondominated frontiers of MOIPs.

The algorithm of Köksalan and Lokman (2009) [12], finds a p value in order to fit a hypersurface to the nondominated points by using one reference point (i.e. mid-point of the problem) assuming that the hypersurface passes through the scaled vectors in set $S = \{(0,1,..,1),(1,0,1,...,1),...,(1,...,1,0)\}$. A specific hypersurface that passes through all the vectors in S can be defined by the following function:

$$(1-z_1')^p + (1-z_2')^p + ... + (1-z_q')^p = 1, \qquad p > 0 \tag{4.12}$$

In fact, this is a variation of L^p distance functions mentioned at the earlier parts of this thesis. This hypersurface will be refered as the "L^p surface" throughout this thesis.

"The central point" or "the reference point" for a MOIP can be defined as the point that has the minimum Tchebycheff distance to the ideal point.

$$i^* = arg \min_i (\max_q z_{iq}) \tag{4.13}$$

In equation 4.13, z_{iq} represents q'th criterion value of the i'th point. Once the point i^* is found, it is substituted into the following equation in order to find the p value for the L^p surface which passes through $S = \{(0,1,..,1),(1,0,1,...,1),...,(1,...,1,0)\}$ and Z_{i^*}:

$$L_p(Z_{i^*}) = \sum_q (1 - z_{i^*q})^p = 1 \tag{4.14}$$

After the L_p surface is obtained, hypothetical points are placed on the surface at approximately equal intervals to find the density measure at different regions on the surface. To place these almost equally spaced hypothetical points, the entire space is divided into small hypercubes, as it was done in the previous sections. Differently, we consider only the hypercubes that are close to the L_p surface in our density calculations. To do that, for each hypercube, we find a point on the L_p surface that is at minimum Euclidean distance from the center of the hypercube by solving ***Model-Dis***. If the distance is more than half of the edge length, the hypercube is excluded from our density analysis since we estimate that the density around this point as zero. Otherwise, we use the corresponding point on the L_p surface as one of our hypothetical points. These hypothetical points are approximately equally spaced on the L_p surface.

Indices:

q = number of objectives (1,2,...,M)

Parameters:

$a_q =$ q'th objective value of the center point of the hypercube

$p =$ previously known parameter of the L_p surface

Decision Variables:

$z_q =$ q'th objective value of the point on L_p surface

Model-Dis:

$$\min \quad \sqrt{\sum_{q=1}^{M}(a_q - z_q)^2} \tag{4.15}$$

$$\text{s.t.} \quad \sum_{q=1}^{M}(1 - z_q)^p = 1 \tag{4.16}$$

$$0 \leq z_q \leq 1 \qquad \forall q \tag{4.17}$$

In Figure 4.5, there is an example provided with a MOKP with 6500 nondominated points and fitted L_p surface related to the problem. Our experiments with 100 different MOIPs show that although the estimation error increases towards the edges, L_p surface fit is a good estimate for MOIPs since we expect more nondominated points in the center and the surface estimates central regions with better accuracy. Working with L_p surface estimates eliminates the unnecessary computational efforts in the regions that are not interesting for the decision maker such as dominated or dominating regions where we do not expect any nondominated points to exist. Thus, L_p surface fitting is very practical since it can be applied very fast to MOIPs by using one reference point. For further analysis see Köksalan and Lokman (2009) [12].

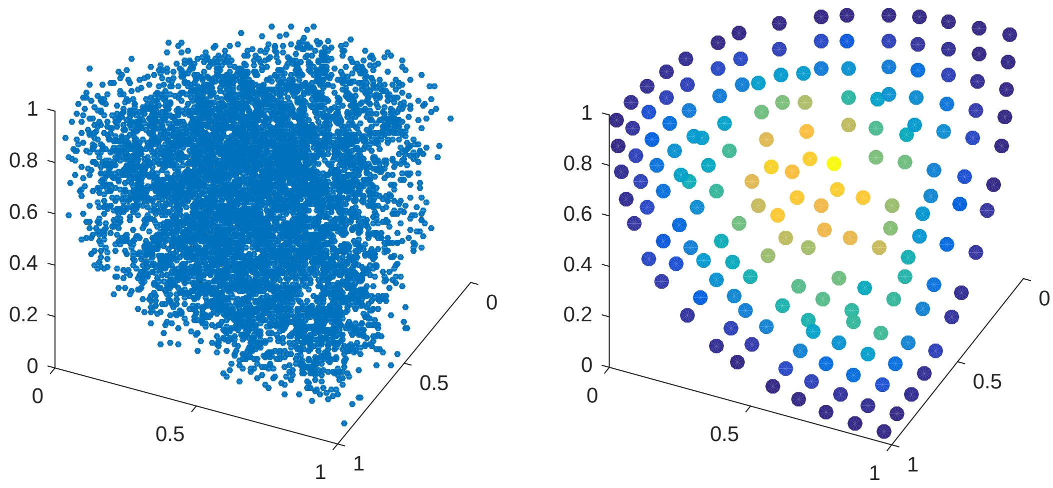

Figure 4.5: Left: MOKP with 6500 nondominated points, Right: hypothetical nondominated points placed on the fitted LP surface

4.3 Density Measure at Hypothetical Points on Lp Surface

For the MOIP problems, density values around each of the well-dispersed hypothetical points on the fitted L_p surface are calculated by Parzen Window (Equation 4.10) method using the triangular kernel function, $K(r)$, given in Equation 4.7 to perform a detailed density analysis on the L_p surface. To define a comparable density measure in different types of problems with different sizes, the density values for each hypothetical point are scaled so that the sum of cumulative density at all hypothetical nondominated points is 1. In order to scale the density values at each hypothetical point, R_i, following function is employed:

$$\overline{P(R_i)} = \frac{P(R_i)}{\sum_i P(R_i)} \tag{4.18}$$

where $\overline{P(R_i)}$ is the scaled density values at the hypothetical point R_i. This value can be interpreted as the contribution of point R_i to the cumulative density measure of the problem.

The calculations mentioned above are done for the hypothetical points placed on the

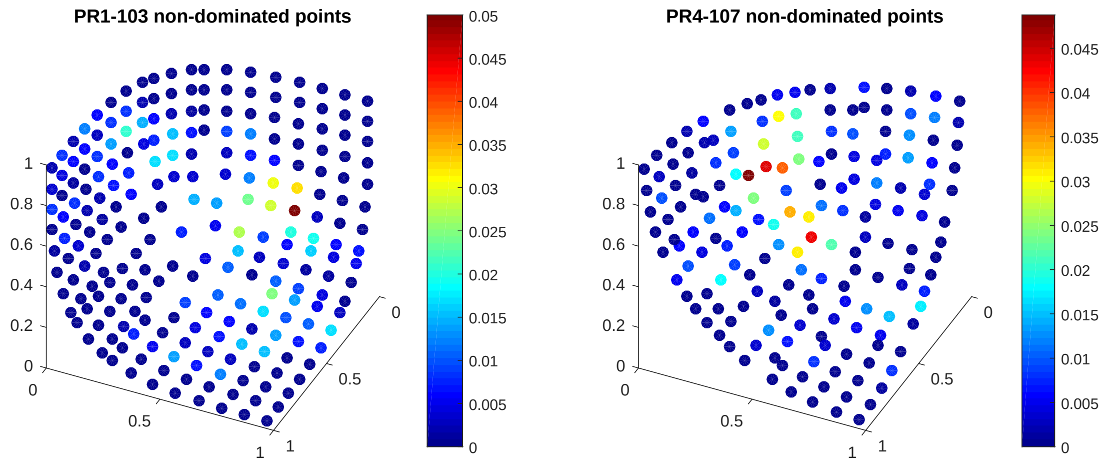

Figure 4.6: Density distributions of different MOAPs with 103 and 107 nondominated points respectively

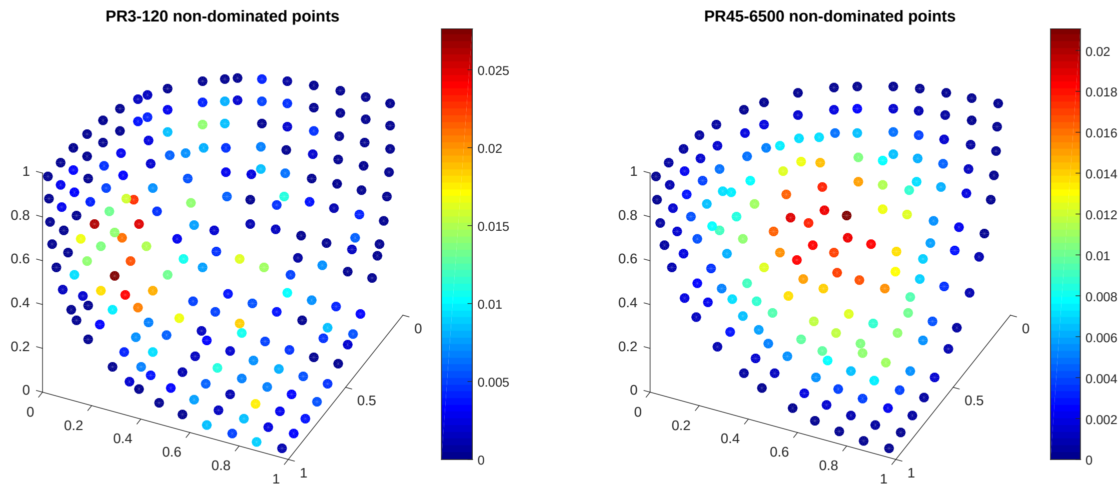

Figure 4.7: Density distribution differences between small and large sized problems

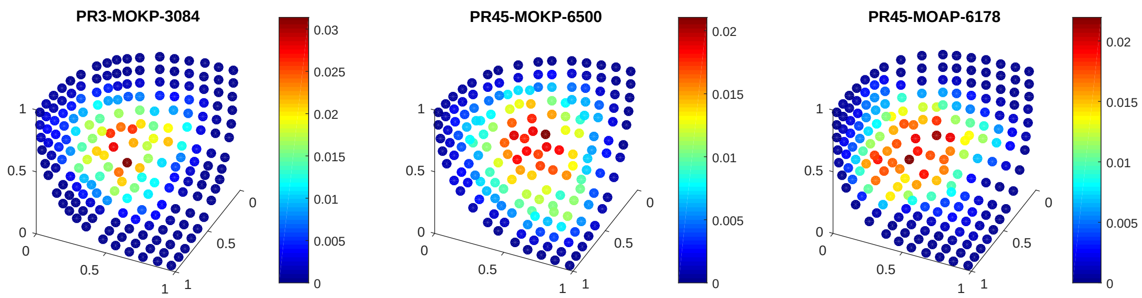

Figure 4.8: Density distribution differences between MOKP with 3084, MOKP with 6500, and MOAP with 6178 nondominated points respectively

fitted L_p surfaces of 100 different type and different sized problems. The complete list of problems which are worked with is provided in Appendix B; Table B.1, Table B.2, and Table B.3.

First observation in these 100 MOIPs is, when the size of the problem is small (i.e. problems with size less than 1000 nondominated points), it is not possible to interpret any acceptable common results among problems when the density distributions are examined (see Figure 4.6). The densities at certain regions of these problems are highly diversified. An example for density calculations with a MOAP having 120 nondominated points and a MOKP having 6500 nondominated points is presented in Figure 4.7 where colors dark red and dark blue represent the most dense and the least dense points respectively.

As the problem sizes get bigger (i.e. when the size gets more than 1000 nondominated points), density distributions of different problems tend to have common properties. As common features in bigger problems among the same type of MOIP, it is important to observe that the hypothetical point said to be the most dense point is located at or close to a region that can be called "the center" of the surface even though the sizes of the problems vary. Also, as one moves on the L_p surface from the "central point" to the one of the edges, density value for each hypothetical point decreases. These features can be observed for two different sized MOKP problems in Figure 4.8.

Another implication in the bigger sized MOIPs is that, different typed problems show similar density distributions regardless of the problem size. As an example for this property, Figure 4.8 is provided. In both problems, it is observed that the density values are decreasing from a point that can be called very dense towards the edges.

4.4 Density Based Categorization of Hypothetical Points

Based on the observations in Section 4.3, we conduct a further analysis to investigate the properties of the distributions of MOIPs. We categorize the hypothetical points based on their density values and search for common properties of MOIPs.

We study on different MOIPs with different sizes. The complete list of problems that are worked with includes MOKP, MOAP, MOSTP, and MOSPP. Problem sizes (the number of nondominated points) in these problems vary between 101 and 10,701. The results show that the studies do not reflect the distribution properties well, since the number of nondominated points is not large enough for the problems having less than 1000 nondominated points. For this reason, for the rest of the thesis the results will be discussed only for the problems having the size of over 1,000 nondominated points. The list of these problems are provided in Appendix C.

In the first method used in defining the categories , 4 different categories are created and their distribution percentages are examined. The highest category, Category 1, has the most dense hypothetical points, and the last category, Category 4, has the least dense hypothetical points. In order to understand the category which a hypothetical point falls into, the categories are formed with the algorithm provided below:

Step 0: Given the hypothetical points, R_i, for $i \in \{1, 2, ..., G\}$ when G is the total number of hypothetical points and category sets, $C_1, C_2, C_3, C_4 = \emptyset$

Step 1: Calculate $k_i = \overline{K(R_i)}$ where $\overline{K(R_i)} = \frac{K(R_i)}{\sum_{i=1}^{G} K(R_i)}$

Step 2: Sort k_i in descending order, call new list of points as $\hat{k}_i$

Step 3: Find $\max n_1$ such that $\sum_{i=1}^{n_1} \hat{k}_i \leq 0.4$ and form $C_1 = \{\hat{k}_1, \hat{k}_2, ..., \hat{k}_{n_1}\}$

Step 4: Find $\max n_2$ such that $\sum_{i=n_1+1}^{n_2} \hat{k}_i \leq 0.3$ and form $C_2 = \{\hat{k}_{n_1+1}, \hat{k}_{n_1+2}, ..., \hat{k}_{n_2}\}$

Step 5: Find $\max n_3$ such that $\sum_{i=n_2+1}^{n_3} \hat{k}_i \leq 0.2$ and form $C_3 = \{\hat{k}_{n_2+1}, k_{n_2+2}, ..., \hat{k}_{3_2}\}$

Step 6: Form $C_4 = \{\hat{k}_{n_3+1}, k_{n_3+2}, ..., \hat{k}_G\}$

The summarised results for the algorithm provided above is presented in Table 4.2. Table shows that the percentage of the nondominated points that fall into each category is similar in the first two categories although the problems differ from each other in problem type and problem size.

Table 4.2: Percentages of points in different types of MOIPs when the first method (4 categories) is used

Average Percentage of Points in each Category for Different Types of MOIPs

Problem Type	**Category 1**	**Category 2**	**Category 3**	**Category 4**
MOAP	0.11 %	0.15 %	0.19 %	0.55 %
MOSP	0.11 %	0.14 %	0.17 %	0.58 %
MOKP	0.10 %	0.14 %	0.17 %	0.59 %
Overall Average	0.11 %	0.14 %	0.18 %	0.57 %

However, because the difference between the percentages of points laying in **Category 4** between two different problems is greater than the other categories, the last two categories are combined and the number of categories is reduced to 3, as indicated below:

Step 0: Given the hypothetical points, R_i, for $i \in \{1, 2, ..., G\}$ when G is the total number of hypothetical points and category sets, $C_1, C_2, C_3 = \emptyset$

Step 1: Calculate $k_i = \overline{P(R_i)}$ where $\overline{P(R_i)} = \frac{P(R_i)}{\sum_{i=1}^{G} P(R_i)}$

Step 2: Sort k_i in descending order, call new list of points as $\hat{k}_i$

Step 3: Find $\max n_1$ such that $\sum_{i=1}^{n_1} \hat{k}_i \leq 0.4$ and form $C_1 = \{\hat{k}_1, \hat{k}_2, ..., \hat{k}_{n_1}\}$

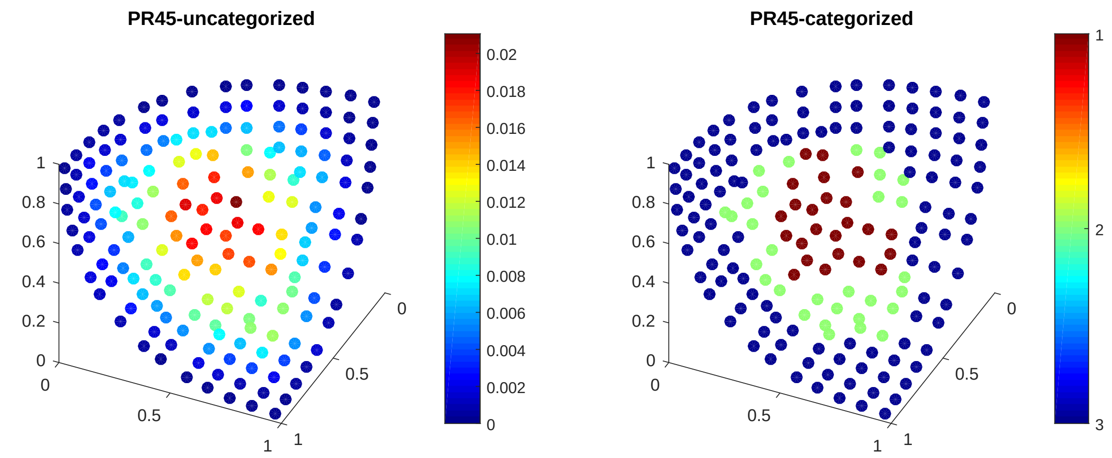

Figure 4.9: Categorization of hypothetical points

Step 4: Find $\max n_2$ such that $\sum_{i=n_1+1}^{n_2} \hat{k}_i \leq 0.3$ and form $C_2 = \{\hat{k}_{n_1+1}, \hat{k}_{n_1+2}, ..., \hat{k}_{n_2}\}$

Step 5: Form $C_3 = \{\hat{k}_{n_2+1}, k_{n_2+2}, ..., \hat{k}_G\}$

In Table 4.3, we present a summary table which shows the average number of points in each category for different types of MOIPs when the second method is used as shown in the algorithm provided above. The categorization of a MOKP problem is also presented in Figure 4.9.

Table 4.3: Percentages of points in different types of MOIPs when the second method (3 categories) is used

Average Percentage of Points in each Category for Different Types of MOIPs

Problem Type	**Category 1**	**Category 2**	**Category 3**
MOAP	0.11 %	0.15 %	0.74 %
MOSP	0.11 %	0.14 %	0.75 %
MOKP	0.10 %	0.14 %	0.76 %
Overall Average	0.11 %	0.14 %	0.75 %

The detailed analysis of categorization with the first method (4 categories) is provided in Tables C.1 and C.2 in Appendix C. Tables C.3 and C.4 in Appendix C also show the percentages of the hypothetical points at each category when the second method is used for different types and sizes of problems.

4.5 Interpretation of Density Measures, Categories and Results

Our analysis on the nondominated sets of different MOIPs show that the distributions have common properties. Based on our findings, we develop a method that estimates the possible locations of the nondominated points and categorize these regions based on their estimated densities for any given MOIP for which the nondominated set is not available. Before giving the details of our algorithm, let us summarize our findings on the distributions of MOIPs:

- the closest 11% of the points to the point with the highest density value, contribute to the total density about 40%.
- the closest 25% of the points to the point with the highest density value, contribute to the total density about 70%. So additional 14% of the points only contribute 30% to the total density.
- the rest 75% of the points contribute to the total density only about 30%.

In line with these findings; it is possible to estimate the possible locations of the solutions and the density categories of these regions by finding the "center of density" for any given MOIP. We first approximate the nondominated set using an L_p surface. The corresponding p-value is calculated using a central reference point as done in Köksalan and Lokman (2009) [12]. We estimate this point as "the center of density". Once we approximate the nondominated set using an L_p surface is fit, a set of well dispersed hypothetical points are generated.

Hypothetical points' categories are then estimated according to the distances to the Central Point with the following criteria:

- the first 11% of the hypothetical points contributing to the total density about 40% are in **Category 1**.

- the next closest 14% of the hypothetical points contributing to the total density about 30% are in **Category 2**.

- the rest 75% of the points contributing to the total density only about 30% are in **Category 3**.

We next present the steps of our algorithm:

Step 0: **(Initialization)** Given the feasible space for true nondominated point set, X, set window width as $h = 0.10$,

Step 1: **(Fitting an L_p surface)** Find the real point, c^*, that has the minimum Tchebycheff distance to the ideal point: $c^* = \min_{x \epsilon X}(\max_q(z_q(x) - Z_q^{IP})$ and calculate the p value for the L_p surface: $L_p(Z_{c^*}) = \sum_{q=1}^{M}(1 - z_{c^*q})^p = 1$.

Step 2: **(Generation of the hypothetical points on the L_p surface)** Partition the objective space into hypercubes with an edge length of h. Solve ***Model-Dis*** for each hypercube to find its Euclidean distance, D_i, to the L_p surface. If $D_i < \frac{h}{2},\ R_i \in H$

Step 3: **(Estimation of the density categories for the hypothetical points on the L_p surface)** Calculate the Tchebycheff distance of each hypothetical point $R_i \in H$ to the central point c^* and place the first 11% of the points to Category 1, the following 14% to Category 2 and the rest to Category 3.

In order to evaluate the performance of this our algorithm, it is possible to compare the true density categories of each point in the case of knowing all the nondominated points with the category estimates found by our method. Figure 4.8 shows the comparison of true and fitted categories of hypothetical points for a MOKP.

Table 4.4 summarizes all the results of experiments on 51 different problems to measure the performance of our density estimation method. The results in Table 4.4 show that in the 51 different MOIP, our estimation method placed 86% of the hypothetical points to the exactly same density category. In addition, when there is a category estimation error, the difference between the estimated category and the true category

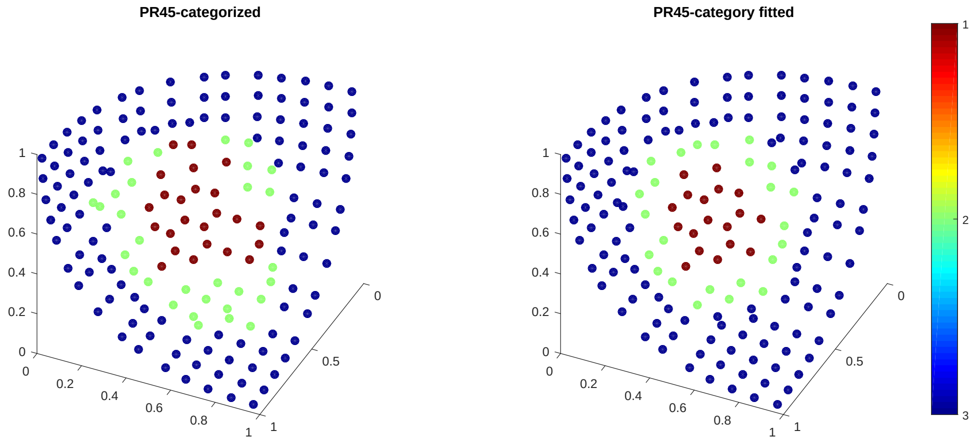

Figure 4.10: True vs. fitted categories of the hypothetical points

is not more than one level. Moreover, this method, which estimates the categories by generating only one nondominated point to define the L_p surface, can be considered as a fast and low cost method for difficult problems, although there is a margin of error in category estimates.

Table 4.4: Category Estimation Error

The Amount of Deviation From the Actual Category	**Percentage of the Hypothetical Points**
2 categories more	0 %
1 category more	6 %
Exact category	87 %
1 category less	7 %
2 categories less	0 %

CHAPTER 5

DISTRIBUTION-BASED REPRESENTATION FOR MULTI-OBJECTIVE INTEGER PROGRAMS

Although we have developed a method that generates hypothetical points and estimates the density categories, we need to generate the true nondominated points. In order to measure how these points represent the whole nondominated set, we also need to define a new quality measure that takes distribution of the nondominated points into consideration.

In this section, we propose a new performance measure, "weighted coverage gap" to evaluate the representation errors. Although it is similar to the coverage gap measure proposed by Sayın (2000) [25] and the representation errors are measured as in Masin and Buckin (2008) [17], we define a weight vector to represent the DM's preferences for different regions.

5.1 Finding Representative Sets with Weighted Quality Measures for Multi-Objective Integer Programs

As we discussed in Section 3.1, the preferences of the DM may be incorporated into the process of generating representative points. We may favor some regions and try to reduce the representation errors especially in these regions.

In order to measure how the nondominated set is represented based on the preferences of the DM, we first ask the DM to define weight values for each category. Then, we evaluate the representative set using:

$$\alpha = \max_{x \in H}(\min_{y \in H}(\max_{1 \le i \le M}((x_i - y_i)w_x))) \tag{5.1}$$

where H is the set of hypothetical points, M is the number of objectives and w_x is the weight of point x according to its category such that:

$$w_x = \begin{cases} W_1, & x \in C_1 \\ W_2, & x \in C_2 \\ W_3, & x \in C_3 \end{cases} \tag{5.2}$$

The values of W_1, W_2 and W_3 are determined according to the preferences of DM. That is, the measure which takes the distribution of the nondominated set into consideration. This also allows us to reduce the representation errors in regions that are of interest and the effect of representation errors in the disinterested regions.

In our algorithm the weights differ in the range [0,1]. If the weight for a point is 1, this means it is in the region of interest and we would like to see its exact coverage gap measure in the calculations. Hence, in the objective function even the smallest distance changes between its representative will play an important role. On the other hand, if we were to use 0 weight for any point, this means this point's coverage gap value will not be taken into consideration. The values between 0 and 1 determines the point's contribution level to the representation error.

Since we approximate the nondominated set using hypothetical points, we next develop a model that selects a subset of hypothetical points that will represent all hypothetical points based on our new quality measure.

Indices:

i,j = number of points in nondominated point set (1,2,...,N)

p = number of objectives of the problem (1,2,...,M)

Decision Variables:

$$x_i = \begin{cases} 1, & \text{if point i is selected to be representative point} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if point j is represented by point i} \\ 0, & \text{otherwise} \end{cases}$$

r = number of representative points

α = weighted coverage gap value

Parameters:

z_{ip} = the p'th objective value of point i

gap = maximum accepted coverage gap value

w_i = the weight of coverage gap for point i

Model-Weighted-Gap:

$$\min\ r \tag{5.3}$$

$$
\begin{aligned}
\text{s.t.}\quad \alpha &\geq (z_{jp} - z_{ip}) w_i y_{ij} && \forall i, j, p && (5.4)\\
N x_i &\geq \sum_{i=1}^{N} y_{ij} && \forall i && (5.5)\\
\sum_{i=1}^{N} y_{ij} &= 1 && \forall j && (5.6)\\
\sum_{i=1}^{N} x_i &= r && && (5.7)\\
\alpha &\leq gap && && (5.8)\\
x_i, y_{ij} &\in \{0, 1\} && \forall i, j && (5.9)
\end{aligned}
$$

Model-Weighted-Gap, is solved for hypothetical points in order to find representatives among hypothetical points. In ***Model-Weighted-Gap***, coverage gap values for hypothetical points are weighted according to their importance. Afterwards, in the inequality 5.8 an upper limit is set for the weighted gap measure.

In order to understand the effect of the weights, we experimented with the extreme values of these weights. We solved different MOIPs with the following weight sets:

- Weight set 1: $W_1 = 1, \quad W_2, W_3 = 0$
- Weight set 1: $W_2 = 1, \quad W_1, W_3 = 0$
- Weight set 3: $W_3 = 1, \quad W_1, W_2 = 0$

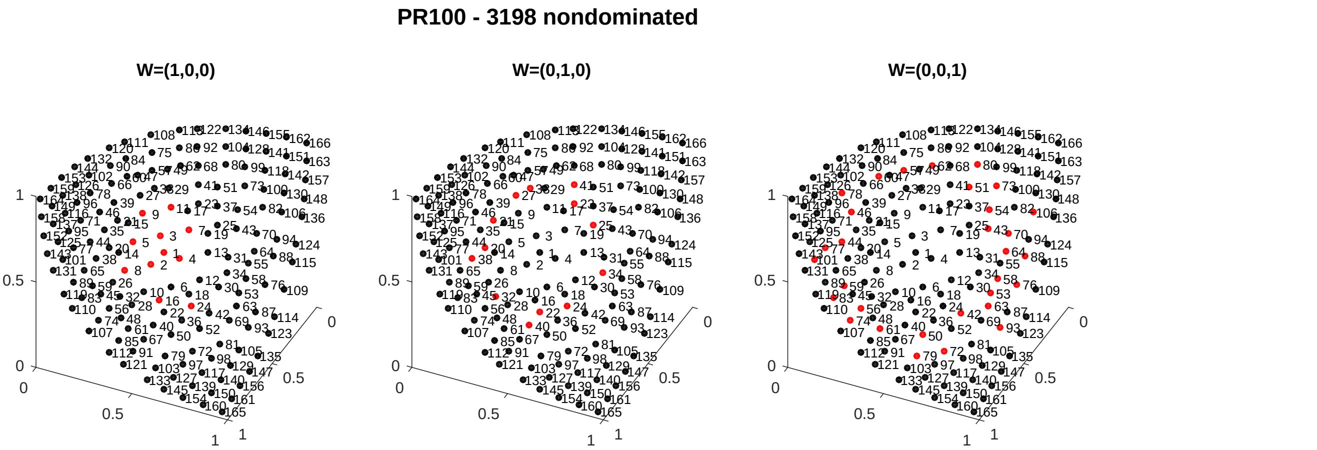

Figure 5.1: Representative point selection among the hypothetical points

As it can be seen in Figure 5.1, working with extreme values of weights shows the effect of the weights. While the representative points are selected in the most dense regions for W=(1,0,0), the hypothetical points in extreme regions are favored in representative selection when W=(0,0,1).

Once the steps mentioned above are followed, a representative point set can be formed as it is seen in Figure 5.2. In this example, a total of 32 representative points are selected among 172 hypothetical points. While selecting the representatives higher weights are given to the most dense regions (Category 1). This resulted in more number of representatives in that region.

Once the optimal set of hypothetical points is found using the estimated densities, we generate the true nondominated points closest to the hypothetical representatives in the optimal subset. For each hypothetical representative point, we search for the real point with the minimum Tchebycheff distance to the representative hypothetical point with the following model:

$$x^* = \min_{x \in X}(\max_{q} |x_q - R_{iq}|) \text{ such that } R_i \in H \tag{5.10}$$

To sum up the process of finding real representative nondominated points following alghoritm, ***FRRP***, is provided:

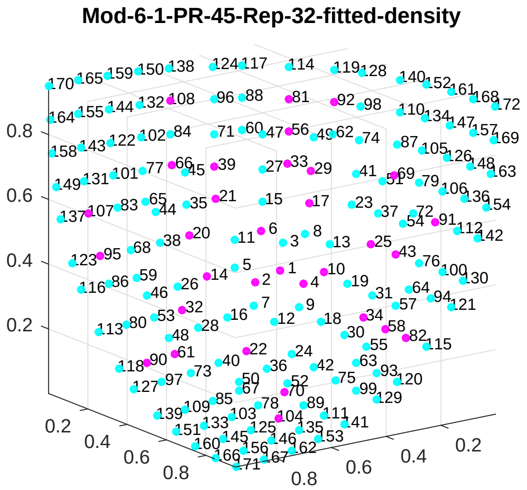

Figure 5.2: Representative point selection among the hypothetical points

Step 1: Find the central point of the MOIP (See Section 4.2)

Step 2: Fit a L_p surface using the central point (see Section 4.2)

Step 3: Generate hypothetical nondominated points on the L_p surface (see Section 4.2)

Step 4: Estimate the category values for the hypothetical nondominated points (See Section 4.5)

Step 5: Use estimated category information for hypothetical nondominated point set to find representatives among the nondominated hypothetical points by using ***Model-Weighted-Gap***

Step 6: Find $x^* = \min_{x \in X}(\max_q |x_q - R_{iq}|)$ such that $R_i \in H$

5.2 Computational Experiments

For a better understanding of estimated and true categories of real nondominated points, we present Figure 5.3. In the Figure, larger circles represent the hypothetical nondominated points whereas, small dots represent the real nondominated points for the same problem. Color of each dot and circle show the category of the nondominated point. In order to assign the category of each real nondominated point, we

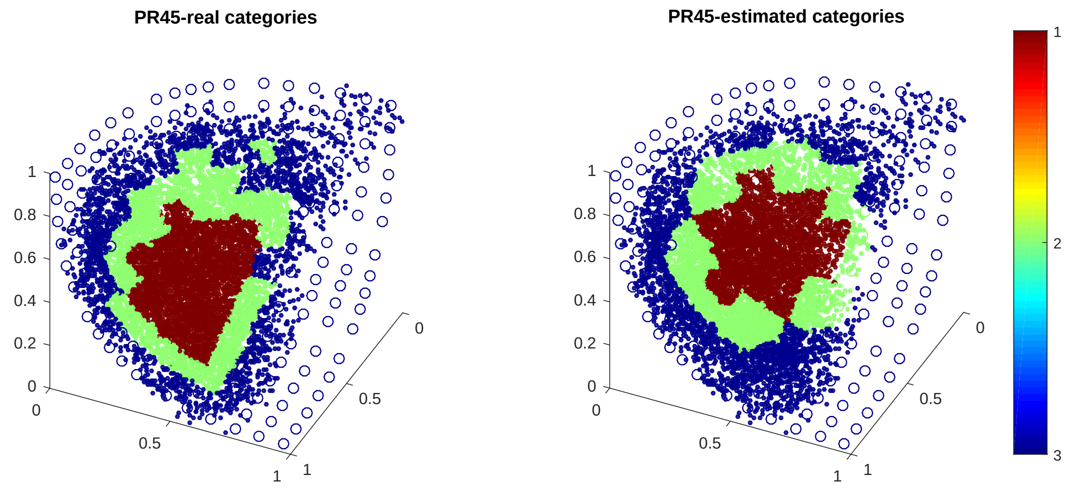

Figure 5.3: Estimation of categories on real nondominated points

used closest hypothetical point's category information. We checked the real and estimated categories for 51 different MOIPs and the results show that we approximate the category values quite well.

We conduct computational experiments algorithm on MOIPs assuming that the set of nondominated set as well as the density information are not available. We generate the representative set using our ***FRRP*** alghorithm in Section 5.1 and we report the weighted coverage gap values calculated using the estimated categories.

For comparison purposes, we also calculate the resulting coverage gap value, α_a, using the true densities of the nondominated points. This comparison shows how well the distribution is estimated by our algorithm. To evaluate how well our algorithm selects the representative hypothetical points, we also consider a case for which the true densities of hypothetical points are available throughout the algorithm, in both the representative point selection and weighted coverage gap value calculations. We report the weighted coverage gap values, α_t. We experiment on three-objective, MOAPs, MOKPs and MOSPPs with sizes between [1014, 10701]. The detailed results for each of 51 MOIPs can be found in Table 5.1. Table 5.2 includes the summarized information about weighted coverage gap values for these 51 MOIPs.

During calculations, weights are selected as 1 for Category 1 points, 0.75 for Category 2 points and 0.5 for Category 3 points. Also the upper limit for the weighted coverage

Table 5.1: Results for 51 MOIPs with more than 1000 nondominated points

Weighted Coverage Gap Values									
PR	**Type**	α	α_a	α_t	**PR**	**Type**	α	α_a	α_t
6	MOAP	0.0825	0.0825	0.0821	**76**	MOAP	0.1018	0.1018	0.0950
7	MOAP	0.1182	0.1182	0.1034	**77**	MOAP	0.1050	0.0978	0.1005
8	MOAP	0.0985	0.0985	0.1034	**78**	MOAP	0.1139	0.1139	0.0917
9	MOAP	0.1111	0.1111	0.1111	**79**	MOAP	0.1197	0.1197	0.0759
10	MOAP	0.1145	0.1145	0.1145	**80**	MOAP	0.0931	0.0931	0.0931
11	MOAP	0.0914	0.0914	0.0914	**81**	MOAP	0.1042	0.1042	0.1063
12	MOAP	0.0841	0.0841	0.0850	**82**	MOAP	0.0775	0.0941	0.1052
13	MOAP	0.1028	0.1028	0.0868	**83**	MOAP	0.0979	0.0979	0.0845
14	MOAP	0.1163	0.1131	0.0886	**84**	MOAP	0.0823	0.0879	0.1009
15	MOAP	0.1029	0.1029	0.1029	**85**	MOAP	0.1060	0.1060	0.1024
26	MOKP	0.0927	0.1023	0.1061	**86**	MOAP	0.1003	0.1003	0.0795
27	MOKP	0.1697	0.1697	0.1697	**87**	MOAP	0.0839	0.0839	0.0854
28	MOKP	0.0880	0.0880	0.0936	**88**	MOAP	0.0872	0.0872	0.0888
29	MOKP	0.0950	0.0950	0.1032	**89**	MOAP	0.0815	0.0807	0.1034
30	MOKP	0.0868	0.0807	0.0908	**90**	MOAP	0.1245	0.1245	0.1165
41	MOKP	0.1104	0.1104	0.1104	**91**	MOKP	0.0842	0.0842	0.1008
42	MOKP	0.1640	0.1640	0.1368	**92**	MOKP	0.1156	0.1156	0.1156
43	MOKP	0.1137	0.1137	0.1143	**93**	MOKP	0.1387	0.1387	0.1387
44	MOKP	0.1196	0.1196	0.1263	**94**	MOKP	0.1296	0.1296	0.1120
45	MOKP	0.1235	0.1235	0.1237	**95**	MOKP	0.1078	0.1078	0.1021
61	MOSP	0.0859	0.0859	0.0809	**96**	MOKP	0.0968	0.0968	0.1063
71	MOAP	0.0903	0.0903	0.1129	**97**	MOKP	0.0966	0.0966	0.0997
72	MOAP	0.0940	0.0940	0.0872	**98**	MOKP	0.0815	0.0932	0.0985
73	MOAP	0.1086	0.1086	0.1086	**99**	MOKP	0.1033	0.1033	0.1033
74	MOAP	0.1206	0.1206	0.1034	**100**	MOKP	0.1187	0.1187	0.1107
75	MOAP	0.1232	0.1232	0.1048					

gap measure is set to 0.1 units.

While we define the desirable weighted coverage gap value as 0.10 for these experiments, our algorithm produces a representative set whose performance measure is calculated by the algorithm as 0.1050 using the estimated density categories. The actual weighted coverage gap value of this representative set is actually equal to 0.1056. Although we slightly underestimate our representation error, the true weighted coverage gap value is also close to the desired performance level that shows our representative selection process works well. If the true density categories were available in our hypothetical representative selection process, we would select another representative

Table 5.2: Summary of results for 51 MOIPs with more than 1000 nondominated points

	Weighted Coverage Gap Values		
	α	α_a	α_t
min	0.0775	0.0807	0.0759
max	0.1697	0.1697	0.1697
avg	0.1050	0.1056	0.1031
std. dev.	0.0193	0.0187	0.0165

set with a weighted coverage value of 0.1031. That is, even when the true densities are known, we would achieve a coverage gap value of 0.1031. This implies that our density estimation procedure works well while the deviation from the target results from L_p surface approximation.

Table 5.3: Weighted coverage gap values of different MOKPs and MOAPs

Problem Type	**Average Number of Nondominated Points**	**Weighted Coverage Gap Values**		
		α	α_a	α_t
MOAP*	3788	0.1013	0.1016	0.0972
MOKP**	4400	0.1118	0.1126	0.1131

*Experiments are done on 30 instances of different MOAPs
**Experiments are done on 20 instances of different MOKPs

Results in Table 5.3 show that average values for weighted coverage gap values in different types MOIPs do not change substantially. Our average α value for 30 different MOAPs is 0.1013 where is that of 20 different MOKPs is 0.1118. The average values for α_a and α_t are also quite close for these two types of MOIPs. This means our estimation procedure performs well even when worked with different types of MOIPs.

In Figure 5.4, we show that our ***FRRP*** alghorithm in Section 5.1 also estimates the locations of worst represented points quite well. As an example, in Figure 5.4 hypothetical point 164 is the worst represented point. This means the weighted coverage value is determined by this point. When we generate the real points in order to com-

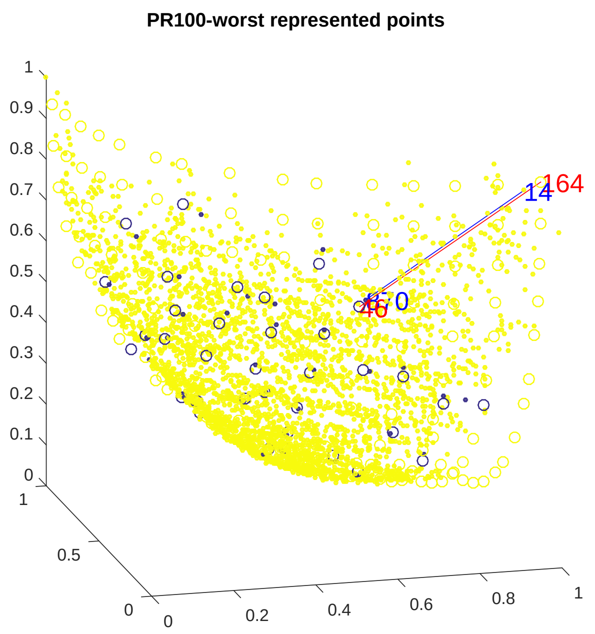

Figure 5.4: Estimation of the locations of worst represented points

pare the performance of our algorithm, we observe that real point 14 is the worst represented real point. In general both hypothetical and real worst represented non-dominated points are located at the same region in the solution space.

Although our resulting weighted coverage gap values deviate from the target level, the proposed methodology is applicable to any MOIP and is practical in terms of the solution times. It does not only generate a representative set, it also allows to the DM to change the weights and involve in the representative selection process.

CHAPTER 6

CONCLUSIONS

Finding a set of points that represent the nondominated set is useful for MOIPs. Incorporating the preferences of the DM into this process is also important for large-sized problems. In this thesis, we develop methods to generate a set of true nondominated points in a resaonable solution time. We consider the distrbution properties and the DM's preferences in our approach.

for comparison purposes, we study the problems for which we know all of the nondomianted points. We solve mathematical models that find the optimal set of representative points according to existing quality measures. We compare these optimal sets and show that taking the distribution of the nondominated set into account is important in representative selection. Therefore, we show that there is a need for a new quality measure.

We then introduce a density measure, and analyze how the density changes over the objective space. We find out common properties of MOIPs. Based on our analysis, we develop a solution framework that approximates the nondominated set, estimates the distribution of nondominated points over the approximate nondominated set and generates the true representative nondominated points using a density-based quality measure.

We experiment on different MOIPs and show that our approach works well. Since our experiments are limited to three-objective case, one extension would be to experiment on problems with more than three criteria. Another future work is to incorporate the decision maker not only in the weight selection process but also other processes throughout the algorithm to reduce the computational effort.

Finally although the experiments are done on MOIPs having three objectives, the algorithm can be further analysed with an expansion of problems with more than three objectives.

REFERENCES

[1] S. Azarm and J. Wu. Metrics for quality assessment of a multiobjective design optimization solution set. *ASME J. Mech. Des*, 123(1):18–25, 2001.

[2] N. Boland, H. Charkhgard, and M. Savelsbergh. The l-shape search method for triobjective integer programming. *Mathematical Programming Computation*, 8(2):217–251, 2016.

[3] G. Ceyhan. Generating representative nondominated point subsets in multi-objective integer programs. Master's thesis, METU, 2014.

[4] K. Dächert and K. Klamroth. A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems. *Journal of Global Optimization*, 61(4):643–676, 2015.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 1973.

[6] M. Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2006.

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

[8] M. Ehrgott and X. Gandibleux. Approximative solution methods for multiobjective combinatorial optimization. *Top*, 12(1):1–63, June 2004.

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

[10] E. Karasakal and M. Koksalan. Generating a Representative Subset of the Nondominated Frontier in Multiple Criteria Decision Making. *Operations Research*, 57(1):187–199, 2009.

[11] 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, Feb. 2014.

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

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

[14] M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, 2006.

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

[16] B. Lokman and M. Köksalan. Finding highly preferred points for multi-objective integer programs. *IIE Transactions*, 46(11):1181–1195, 2014.

[17] M. Masin and Y. Bukchin. Diversity Maximization Approach for Multiobjective Optimization. *Operations Research*, 56(2):411–424, 2008.

[18] G. Mavrotas and K. Florios. An improved version of the augmented ε-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18):9652–9669, 2013.

[19] 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.

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

[21] O. Ozpeynirci and M. Koksalan. An Exact Algorithm for Finding Extreme Supported Nondominated Points of Multiobjective Mixed Integer Programs, 2010.

[22] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[23] A. Przybylski, X. Gandibleux, and M. Ehrgott. A Recursive Algorithm for Finding All Nondominated Extreme Points in the Outcome Set of a Multiobjective Integer Programme. *INFORMS Journal on Computing*, 22(3):371–386, Aug. 2010.

[24] S. Ruzika and M. M. Wiecek. Approximation methods in multiobjective programming. *Journal of optimization theory and applications*, 126(3):473–501, 2005.

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

[26] J. Sylva and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158(1):46–55, 2004.

[27] 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.

[28] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

APPENDIX A

MODELS USED IN THE GENERATION OF NONDOMINATED POINTS

The Multi-Objective Assignment Problem - MOAP

Decision Variables:

$$x_{jk} \begin{cases} 1, & \text{if job j is assigned to person k} \\ 0, & \text{otherwise} \end{cases}$$

Parameters:

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

Model-MOAP:

$$\text{"max"}\ \{z_1(x), z_2(x)..., z_M(x)\} \tag{A.1}$$

$$\text{s.t.} \sum_{k=1}^{l} x_{jk} = 1 \quad \forall j \tag{A.2}$$

$$\sum_{j=1}^{l} x_{jk} = 1 \quad \forall k \tag{A.3}$$

$$x_{jk} \in \{0, 1\} \quad \forall jk \tag{A.4}$$

where $z_i(x) = \sum_{j=1}^{l} \sum_{k=1}^{l} c_{ijk} x_{jk} \quad \forall i.$

We generate c_{ijk} coefficients from discrete uniform distribution in the interval [1:20].

The Multi-Objective Knapsack Problem - MOKP

Decision Variables:

$$x_j \begin{cases} 1, & \text{if j is included in the knapsack} \\ 0, & \text{otherwise} \end{cases}$$

Parameters:

c_{ij}: the coefficient of item j in objective i

w_j: the weight of item j in the knapsack

W: the capacity of knapsack

Model-MOKP:

$$\text{"max"} \ \{z_1(x), z_2(x)..., z_M(x)\} \tag{A.5}$$

$$\text{s.t.} \ \sum_{j=1}^{n} w_j x_j \quad \leq W \tag{A.6}$$

$$x_j \quad \in \{0,1\} \quad \forall j \tag{A.7}$$

where $z_i(x) = \sum_{j=1}^{n} c_{ij} x_j \quad \forall i.$

We randomly generate objective function and weight coefficients from discrete uniform distribution such that $c_{ij}, w_j \in [10 : 100]$ and set the knapsack capacity to the half of the total weight of all items, $W = \frac{\sum_{j=1}^{n} w_j}{2}$.

The Multi-Objective Spanning Tree Problem - MOSTP

Decision Variables:

$$x_j \begin{cases} 1, & \text{if any flow exists from node i to node j} \\ 0, & \text{otherwise} \end{cases}$$

w_{ij}: total flow from node i to node j

f_{ij}^k: total flow of commodity k from node i to node j

Parameters:

c_{ijv}: unit cost of flow from node i to node j in objective v

Model-MOSTP:

$$\text{"}\max\text{"}\ \{z_1(w_{ij}), z_2(w_{ij})..., z_M(w_{ij})\} \tag{A.8}$$

$$\text{s.t.}\ \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} = 1 \tag{A.9}$$

$$\sum_{j=1}^{n} f_{ij}^k - \sum_{j=1}^{n} f_{ji}^k = \begin{cases} 1, & i = 1 \\ -1, & i = n \\ 0, & otherwise \end{cases} \quad \forall i,j \quad k = 2,3,...,n \tag{A.10}$$

$$f_{ij}^k \leq w_{ij} \quad \forall i,j \quad k = 2,3,...,n \tag{A.11}$$

$$w_{ij} + w_{ji} = x_{ij} \quad \forall i,j \tag{A.12}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \tag{A.13}$$

where $z_v(w_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}c_{ijv} \quad v = 1,2,...,M.$

We generate cost parameters, c_{ijv} as integers uniformly distributed such that $c_{ijv} \in [10:100]$.

The Multi-Objective Shortest Path Problem - MOSPP

Decision Variables:

$$x_{ij} \begin{cases} 1, & \text{if arc between nodes i and j is used} \\ 0, & \text{otherwise} \end{cases}$$

Parameters:

c_{ijv}: unit cost of arc between nodes i and j in objective v

Model-MOSPP:

$$\text{"max"} \ \{z_1(x_{ij}), z_2(x_{ij})..., z_M(x_{ij})\} \tag{A.14}$$

$$\text{s.t.} \ \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = \begin{cases} 1, & i = 1 \\ -1, & i = n \\ 0, & otherwise \end{cases} \qquad \forall i \tag{A.15}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \tag{A.16}$$

where $z_v(x_{ij}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ijv} x_{i,j} \quad v = 1, 2, ..., M.$

The values for c_{ijv} are calculated as follows:

$$c_{ijv} = \begin{cases} UNIF(10, 50), & i \in Stage_k, \quad i < j, \quad k = 1, 2, ..., s \\ UNIF(30, 100), & i \in Stage_k, \quad j \in Stage_{(k+1)}, \quad k = 1, 2, ..., s - 1 \\ L, & otherwise \end{cases}$$

where L is a sufficiently large number to guarantee that the corresponding edge will not be included in the random graph.

APPENDIX B

COMPLETE LIST OF MOIPS

Table B.1: List of MOIPs (Part-1)

PR	Problem Type	P-value	Size of the Problem	Number of Hypothetical Points
1	MOAP	3.9573	103	232
2	MOAP	3.0947	162	207
3	MOAP	2.9711	120	201
4	MOAP	2.1442	107	174
5	MOAP	2.5478	117	169
6	MOAP	3.2179	1846	201
7	MOAP	2.8278	1617	196
8	MOAP	3.2183	1513	201
9	MOAP	2.7646	2007	190
10	MOAP	2.4907	3275	181
11	MOAP	3.6019	6369	211
12	MOAP	3.3835	5368	199
13	MOAP	3.0776	6654	204
14	MOAP	3.4712	6975	196
15	MOAP	3.4125	6573	202
16	MOKP	3.0402	79	207
17	MOKP	4.1975	82	217
18	MOKP	2.4288	60	178
19	MOKP	2.4014	18	178
20	MOKP	2.5394	52	169
21	MOKP	2.4619	405	178
22	MOKP	3.8204	378	229
23	MOKP	4.8056	92	222
24	MOKP	2.2909	676	171
25	MOKP	3.3470	313	195
26	MOKP	2.8080	2751	190
27	MOKP	2.5361	3837	169
28	MOKP	2.8742	3780	192
29	MOKP	2.8599	3084	195
30	MOKP	2.9494	2952	198
31	MOKP	2.3346	182	166
32	MOKP	2.3093	168	172
33	MOKP	2.9246	76	195
34	MOKP	1.9192	163	148
35	MOKP	2.2727	470	177

Table B.2: List of MOIPs (Part-2)

PR	Problem Type	P-value	Size of the Problem	Number of Hypothetical Points
36	MOKP	2.3120	784	172
37	MOKP	2.3462	912	166
38	MOKP	2.9717	519	201
39	MOKP	3.0248	280	207
40	MOKP	2.8790	356	192
41	MOKP	2.6780	2790	184
42	MOKP	2.3642	8288	172
43	MOKP	2.2769	10701	183
44	MOKP	2.3888	5652	172
45	MOKP	2.3898	6500	172
46	MOSPP	4.0531	49	226
47	MOSPP	2.1100	80	159
48	MOSPP	2.2199	119	174
49	MOSPP	2.8603	64	195
50	MOSPP	2.1137	45	159
51	MOSPP	2.9196	217	195
52	MOSPP	2.7456	169	190
53	MOSPP	2.7587	214	190
54	MOSPP	3.0873	325	207
55	MOSPP	2.8238	437	196
56	MOSPP	2.8238	437	196
57	MOSPP	3.6200	464	211
58	MOSPP	3.7270	510	226
59	MOSPP	3.9431	411	232
60	MOSPP	3.8152	316	232
61	MOSPP	4.1155	1014	220
62	MOSPP	3.8891	725	232
63	MOSPP	3.5615	874	205
64	MOSPP	4.2951	682	220
65	MOSPP	4.0050	795	232
66	MOSTP	2.4445	655	178
67	MOSTP	2.4555	486	172
68	MOSTP	2.2777	704	183
69	MOSTP	2.5129	549	184
70	MOSTP	2.5140	733	184

Table B.3: List of MOIPs (Part-3)

PR	Problem Type	P-value	Size of the Problem	Number of Hypothetical Points
71	MOAP	2.6147	1970	178
72	MOAP	2.9683	1247	198
73	MOAP	2.9469	1806	198
74	MOAP	2.9252	2150	195
75	MOAP	2.7078	2246	196
76	MOAP	2.8140	2813	196
77	MOAP	3.2177	1825	201
78	MOAP	2.7405	1591	190
79	MOAP	2.9497	1916	198
80	MOAP	2.8674	1521	195
81	MOAP	3.2186	6099	201
82	MOAP	3.3738	6178	196
83	MOAP	3.6803	3898	214
84	MOAP	3.8194	5394	232
85	MOAP	3.1313	5021	195
86	MOAP	3.7375	4066	226
87	MOAP	3.1454	4350	201
88	MOAP	3.5479	5670	205
89	MOAP	2.9280	6584	195
90	MOAP	3.2003	5090	204
91	MOKP	3.0002	3523	201
92	MOKP	2.9105	3114	195
93	MOKP	2.7196	2714	190
94	MOKP	2.5541	4773	175
95	MOKP	2.7240	2433	190
96	MOKP	2.7102	7203	190
97	MOKP	3.3894	3307	202
98	MOKP	3.1783	3062	195
99	MOKP	2.6522	4355	184
100	MOKP	2.3603	3198	166

APPENDIX C

COMPLETE LIST OF MOIPS HAVING MORE THAN 1000 NONDOMINATED POINTS

Table C.1: Percentage of the points in each of 4 categories for the MOIPs having more than 1000 nondominated points (Part-1/2)

PR	Problem Type	P-value	Problem Size	Hypo. Points	Percentage of Points in each Category			
					1	2	3	4
6	MOAP	3.2179	1846	201	13%	16%	19%	52%
7	MOAP	2.8278	1617	196	13%	17%	21%	48%
8	MOAP	3.2183	1513	201	12%	15%	18%	54%
9	MOAP	2.7646	2007	190	10%	14%	17%	58%
10	MOAP	2.4907	3275	181	12%	14%	16%	58%
11	MOAP	3.6019	6369	211	10%	12%	17%	61%
12	MOAP	3.3835	5368	199	11%	16%	22%	51%
13	MOAP	3.0776	6654	204	12%	16%	22%	50%
14	MOAP	3.4712	6975	196	12%	13%	16%	59%
15	MOAP	3.4125	6573	202	12%	13%	16%	59%
26	MOKP	2.8080	2751	190	7%	8%	13%	72%
27	MOKP	2.5361	3837	169	9%	12%	20%	59%
28	MOKP	2.8741	3780	192	9%	11%	15%	65%
29	MOKP	2.8599	3084	195	9%	11%	16%	64%
30	MOKP	2.9494	2952	198	10%	12%	14%	65%
41	MOKP	2.6780	2790	184	10%	13%	15%	62%
42	MOKP	2.3642	8288	172	13%	16%	20%	50%
43	MOKP	2.2769	10701	183	14%	16%	21%	49%
44	MOKP	2.3888	5652	172	12%	16%	19%	52%
45	MOKP	2.3898	6500	172	14%	17%	20%	49%
61	MOSPP	4.1155	1014	220	10%	14%	15%	60%
71	MOAP	2.6147	1970	178	15%	17%	20%	48%
72	MOAP	2.9682	1247	198	11%	15%	21%	53%
73	MOAP	2.9468	1806	198	14%	16%	16%	55%
74	MOAP	2.9252	2150	195	10%	15%	19%	56%
75	MOAP	2.7077	2246	196	11%	14%	17%	58%
76	MOAP	2.8140	2813	196	9%	11%	17%	63%
77	MOAP	3.2176	1825	201	12%	15%	17%	55%
78	MOAP	2.7404	1591	190	13%	16%	20%	51%
79	MOAP	2.9496	1916	198	11%	15%	18%	56%
80	MOAP	2.8674	1521	195	14%	14%	18%	54%

Table C.2: Percentage of the points in each of 4 categories for the MOIPs having more than 1000 nondominated points (Part-2/2)

					Percentage of Points in each Category			
PR	**Problem Type**	**P-value**	**Problem Size**	**Hypo. Points**	**1**	**2**	**3**	**4**
81	MOAP	3.2186	6099	201	13%	16%	20%	51%
82	MOAP	3.3737	6178	196	11%	12%	16%	61%
83	MOAP	3.6802	3898	214	10%	13%	17%	60%
84	MOAP	3.8194	5394	232	10%	13%	15%	63%
85	MOAP	3.1312	5021	195	12%	16%	23%	49%
86	MOAP	3.7374	4066	226	13%	16%	19%	52%
87	MOAP	3.1453	4350	201	11%	14%	19%	55%
88	MOAP	3.5479	5670	205	11%	15%	20%	54%
89	MOAP	2.9279	6584	195	13%	16%	20%	51%
90	MOAP	3.2003	5090	204	12%	15%	22%	52%
91	MOKP	3.0002	3523	201	9%	12%	14%	65%
92	MOKP	2.9105	3114	195	8%	12%	15%	65%
93	MOKP	2.7196	2714	190	7%	9%	15%	68%
94	MOKP	2.5541	4773	175	11%	14%	20%	55%
95	MOKP	2.7239	2433	190	10%	15%	21%	55%
96	MOKP	2.7101	7203	190	11%	13%	16%	60%
97	MOKP	3.3894	3307	202	11%	15%	20%	53%
98	MOKP	3.1782	3062	195	10%	13%	17%	60%
99	MOKP	2.6522	4355	184	10%	13%	15%	62%
100	MOKP	2.3603	3198	166	13%	14%	18%	55%

min	7%	8%	13%	48%
max	15%	17%	23%	72%
avg	11%	14%	18%	57%
std. dev.	1.8%	2.0%	2.5%	5.7%

Table C.3: Percentage of the points in each of 3 categories for the MOIPs having more than 1000 nondominated points (Part-1/2)

					Percentage of Points in each Category		
PR	**Problem Type**	**P-value**	**Problem Size**	**Hypo. Points**	**1**	**2**	**3**
6	MOAP	3.2179	1846	201	13%	16%	71%
7	MOAP	2.8278	1617	196	13%	17%	70%
8	MOAP	3.2183	1513	201	12%	15%	73%
9	MOAP	2.7646	2007	190	10%	14%	76%
10	MOAP	2.4907	3275	181	12%	14%	74%
11	MOAP	3.6019	6369	211	10%	12%	77%
12	MOAP	3.3835	5368	199	11%	16%	73%
13	MOAP	3.0776	6654	204	12%	16%	72%
14	MOAP	3.4712	6975	196	12%	13%	75%
15	MOAP	3.4125	6573	202	12%	13%	75%
26	MOKP	2.8080	2751	190	7%	8%	85%
27	MOKP	2.5361	3837	169	9%	12%	79%
28	MOKP	2.8741	3780	192	9%	11%	79%
29	MOKP	2.8599	3084	195	9%	11%	80%
30	MOKP	2.9494	2952	198	10%	12%	79%
41	MOKP	2.6780	2790	184	10%	13%	77%
42	MOKP	2.3642	8288	172	13%	16%	70%
43	MOKP	2.2769	10701	183	14%	16%	69%
44	MOKP	2.3888	5652	172	12%	16%	72%
45	MOKP	2.3898	6500	172	14%	17%	69%
61	MOSPP	4.1155	1014	220	10%	14%	76%
71	MOAP	2.6147	1970	178	15%	17%	69%
72	MOAP	2.9682	1247	198	11%	15%	74%
73	MOAP	2.9468	1806	198	14%	16%	71%
74	MOAP	2.9252	2150	195	10%	15%	75%
75	MOAP	2.7077	2246	196	11%	14%	75%
76	MOAP	2.8140	2813	196	9%	11%	80%
77	MOAP	3.2176	1825	201	12%	15%	73%
78	MOAP	2.7404	1591	190	13%	16%	71%
79	MOAP	2.9496	1916	198	11%	15%	74%
80	MOAP	2.8674	1521	195	14%	14%	72%

Table C.4: Percentage of the points in each of 3 categories for the MOIPs having more than 1000 nondominated points (Part-2/2)

PR	Problem Type	P-value	Problem Size	Hypo. Points	Percentage of Points in each Category		
					1	**2**	**3**
81	MOAP	3.21863	6099	201	13%	16%	71%
82	MOAP	3.373751	6178	196	11%	12%	77%
83	MOAP	3.680293	3898	214	10%	13%	77%
84	MOAP	3.81941	5394	232	10%	13%	78%
85	MOAP	3.131264	5021	195	12%	16%	72%
86	MOAP	3.737463	4066	226	13%	16%	71%
87	MOAP	3.145361	4350	201	11%	14%	75%
88	MOAP	3.547947	5670	205	11%	15%	74%
89	MOAP	2.927998	6584	195	13%	16%	71%
90	MOAP	3.200332	5090	204	12%	15%	74%
91	MOKP	3.000231	3523	201	9%	12%	79%
92	MOKP	2.910548	3114	195	8%	12%	79%
93	MOKP	2.719634	2714	190	7%	9%	84%
94	MOKP	2.554102	4773	175	11%	14%	75%
95	MOKP	2.723962	2433	190	10%	15%	75%
96	MOKP	2.710181	7203	190	11%	13%	76%
97	MOKP	3.389438	3307	202	11%	15%	74%
98	MOKP	3.178276	3062	195	10%	13%	77%
99	MOKP	2.65225	4355	184	10%	13%	77%
100	MOKP	2.360317	3198	166	13%	14%	73%

min	7%	8%	69%
max	15%	17%	85%
avg	11%	14%	75%
std. dev.	1.8%	2.0%	3.6%