BALANCE IN RESOURCE ALLOCATION PROBLEMS: A CHANGING REFERENCE APPROACH

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

INDUSTRIAL ENGINEERING

By

Hale Erkan

May 2018

BALANCE IN RESOURCE ALLOCATION PROBLEMS: A CHANGING REFERENCE APPROACH

By Hale Erkan

May 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Özlem Karsu(Advisor)

Oya Karaşan

Meral Azizoğlu

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

BALANCE IN RESOURCE ALLOCATION PROBLEMS: A CHANGING REFERENCE APPROACH

Hale Erkan
M.S. in Industrial Engineering
Advisor: Özlem Karsu
May 2018

Fairness has become one of the primary concerns in resource allocation problems, especially in settings which are associated with public welfare. Using a pure efficiency maximizing approach may not be applicable while distributing resources among entities, hence we propose a novel structure for integrating balance into the allocation process. In the proposed approach, balance is defined and measured as the deviation from a reference distribution determined by the decision maker. We acknowledge that what is considered balanced by the decision maker might change with respect to the level of total output distributed. To provide an allocation policy that is in line with this changing structure of balance, we allow the decision maker to change her reference distribution depending on the total amount of output (benefit).

We illustrate our approach using a project portfolio selection problem. We formulate a mixed integer mathematical programming model for the problem with maximizing efficiency and minimizing imbalance objectives. The bi-objective model is initially solved with the epsilon constraint method. However for larger problem instances this approach fails to find solutions within reasonable time limits. Hence we implement metaheuristic algorithms and report on their performance. As an alternative solution method, an interactive algorithm is presented and used to find the most preferred solution of the decision maker. The proposed resource allocation model provides important insights to decision makers regarding the tradeoff between efficiency and fairness, and provides a useful tool to incorporate specific balance concerns into the problem.

Keywords: Decision support systems, Biobjective resource allocation problem, Fairness, Knapsack Problem, Balance.

ÖZET

KAYNAK DAĞILIM PROBLEMLERİNDE DENGE FAKTÖRÜ: DEĞİŞEN REFERANS YÖNTEMİ

Hale Erkan
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Özlem Karsu
Mayıs 2018

Denge, kaynak dağıtım problemlerinde önemli bir kriter haline gelmektedir. Özellikle toplumsal refahı etkileyen konularda denge faktörüne karşı hassasiyet daha yüksek ölçüdedir. Bu tarz kaynak dağılım problemlerinde yalnızca verimliliğin ençoklandığı bir yöntem kullanıcılar tarafından uygun bulunmayabilir. Bu nedenle, bu çalışmada denge ve verimlilik kriterlerinin birlikte ele alındığı bir kaynak dağıtım mekanizması önerilmektedir. Denge ölçümü yapılırken gerçekleşen dağılımın karar verici tarafından önceden belirlenen referans dağılımdan ne kadar saptığı hesaplanmaktadır. Denge, farklı çıktı miktarları söz konusu olduğunda farklı şekillerde algılanabilir. Bu değişken denge algısıyla uyumlu bir mekanizma geliştirebilmek için farklı çıktı miktarlarında karar vericinin referans dağılımını değiştirmesine olanak verilmiştir.

Önerilen yöntem proje portföy seçimi problemi üzerinde uygulanmıştır. Problem, karışık tamsayı matematiksel modelleme kullanılarak iki amaç fonksiyonlu bir şekilde formüle edilmiştir. Amaç fonksiyonlarından biri verimliliği ençoklamakken, diğeri dengesizliği enazlamak olarak belirlenmiştir. İki amaç fonksiyonlu problem öncelikle epsilon kısıt yöntemi kullanılarak çözülmüştür. Problem boyutu büyüdükçe hızla artan çözüm süreleri sebebiyle büyük problemler için metasezgisel algoritmalar kullanılmıştır ve performansları raporlanmıştır. Ardından alternatif bir çözüm yöntemi olarak interaktif bir algoritma uygulanmıştır. Önerilen yöntem karar vericiye adaletli dağıtım konusunda içgörü kazandıracak ve probleme özel denge kısıtlarını yansıtmakta kullanışlı olacaktır.

Anahtar sözcükler: Karar verme mekanizması, İki amaçlı kaynak dağılım problemi, Adillik, Sırt Çantası Problemi, Denge.

Acknowledgement

I would like to thank my advisor Asst. Prof. Özlem Karsu for her invaluable support during my graduate study. I feel very lucky to have such an insightful supervisor.

Furthermore, my thanks and appreciation go to Prof. Oya Karaşan and Prof. Meral Azizoğlu for accepting to read and review my thesis and their valuable comments.

I am grateful to my mother and father, Öznur Erkan and Ziya Erkan, for their self-sacrifice throughout my education life. They have always supported me to accomplish my dreams in every part of my life. I cannot thank enough for their encouragement and endless love.

I would like to express my gratitude to Sertaç Güneri Yazgı. He has been always there to cheer me up in difficult times and we have shared many great memories throughout the years. He has brightened my life in many ways and always motivated me to overcome problems that I have encountered. I feel his love and support constantly at my side that give me the strength to handle things more easily.

Above all, I would like to thank my sister Hande İrez, her husband Yiğit İrez, and especially my dear niece Lale İrez. With her kind loving hearth, Lale has been one of my major sources of joy throughout my graduate study. It may seem like she has been the one exploring world and learning new skills; but in fact, witnessing her journey has thought me a lot about not giving up and made me realize the endless potential and energy we all have inside. I am constantly fascinated by her resilient attitude towards the difficulties and passion for learning.

I also want to thank Harun Avcı, Merve Bolat, Başak Erman, Utku Karaca, Eren Özbay, and Farzad Shams along with my friends from EA-327 for the supportive and joyful environment they have provided during our graduate studies.

I would like to thank TUBITAK (The Scientific and Technological Research Council of Turkey) for their financial support throughout this study (under Grant no:215M713).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Allocating resources across multiple entities is a problem encountered in many real life settings, hence resource allocation problems are widely studied in the operational research (OR) literature [1]. Various decision support systems have been proposed to help the decision makers allocate resources so that system efficiency is maximized.

Typically, resources are allocated so that the entities will enjoy benefits (outputs), i.e. any resource allocation to entities is associated with an output allocation. Resources and outputs can change based on the problem type: for example in a health care resource allocation problem, resource can be budget and benefit can be measured with the number of people who benefit from a healthcare project or it can be measured by the quality adjusted life years gained by the target population. Throughout this thesis we use the terms output and benefit and also resource and input interchangeably.

One widely-used approach in resource allocation is efficiency maximization, also known as the *utilitarian* approach. This method focuses on maximizing the total output regardless of how it is distributed across the entities. It achieves the best result in terms of efficiency; however it may fail to sustain fairness (balance) in the allocation. This is because this approach may ignore allocating resources to entities that are not as good as the others at converting resources to output. For example, in a project portfolio selection problem, if some project categories are more productive than the others, a pure efficiency maximizing model would lead the decision maker to use the entire budget on them. However, such a solution may not be acceptable, especially when categories provide benefits in different technology areas, or to different population groups (as in the health care settings). Consider a health care project selection problem, where projects are categorized by target patients' ages; resource usage is measured with the cost of a project and benefit is the resulting quality adjusted life years for a patient group. In this case, the utilitarian approach can result in a highly unfair resource distribution among categories. Elderly patients may not receive any resource, since these patient groups obtain relatively less benefit per unit resource devoted to them. Hence, using a utilitarian approach could be considered a highly controversial decision.

One approach to ensure fairness can be equally dividing the resources among entities, if possible. However, such an allocation may also be inapplicable, since it may result in a high efficiency loss. Consider the same problem in healthcare policy making: a utilitarian approach may not give any resources to elderly (or terminally ill patients) and hence is undesirable. On the other hand, a policy maker may also find a completely equal resource allocation to infants and elderly unfair or unacceptable on the grounds that the society should provide more resources to younger generations. Between these two extremes (an efficiency maximizing allocation and fairness maximizing allocation), there are other allocations, i.e. the problem is a bi(multi) objective problem. Moreover, as the example shows, fairness itself is a subjective term and different people may have different definitions for a "fair" allocation. A good decision support approach should be able to accommodate this difference.

In this thesis, we will suggest an approach that will address fairness in resource allocation problems. The aim is to create a decision support system that will guide decision makers to incorporate balance concerns into the allocation decisions. Proposed method will enable decision makers to define a perfectly balanced (a reference allocation) allocation differently at different levels of output. In order to achieve this structure, a bi-objective mixed integer mathematical formulation is used. In addition, we will engage both a posteriori and interactive solution approaches, which differs in terms of decision maker's participation throughout the solution process.

In the following chapter, Chapter 2, the underlying idea of changing reference approach is introduced. Then the basis of this study, measurement of balance while using changing reference allocations is explained with an example setting.

Chapter 3 reviews the most relevant literature on this topic. Literature review is divided into 3 main groups and each group is discussed separately. First we address the literature on fairness in OR problems, and provide applications and categorizations. Then, we refer to the literature on multi objective portfolio selection problems. Lastly, solution methods used for bi-objective programming models are presented. Furthermore, main focus and distinctive characteristics of this thesis are specified and novelty of this study is emphasized.

In Chapter 4 we exemplify the use of the proposed approach on knapsack type discrete resource allocation problems. We provide the corresponding mixed integer mathematical model and a variation of it. We assume that the decision maker provides reference proportions, which indicate the proportions that she wants to achieve in an ideally balanced allocation. The first model uses a discrete reference distribution approach, meaning that reference proportion is constant for each interval. As long as distributed total output is between two thresholds balance is measured with a constant reference proportion. Its variation uses a moving reference distribution approach which enables the reference proportion to move along the threshold values. Details of the mathematical models are explicitly explained in this chapter.

Chapter 5 is devoted to solution methods used to solve proposed bi-objective resource allocation problem. We discuss the exact algorithms; Epsilon constraint method and an interactive algorithm as well as metaheuristic algorithms for finding approximate solutions; SPEA2, NSGA-II and MOCell. For the metaheuristic algorithms performance metrics are defined to asses the quality of solutions.

In Chapter 6 we present the results of our computational experiments. We conclude in Chapter 7 by summarizing our results and pointing out some further research directions that could be pursued.

Chapter 2

Problem Definition

In this chapter we discuss the idea of changing reference points with respect to which balance is measured. This way, we allow the decision maker to change her desired proportions as the total output or the total input changes.

The underlying idea is similar to that of the well-known allocation rule from Babylonian Talmud, which divides a given resource in different proportions, changing with respect to the total amount of resource and demand (see [2] for a detailed description). The story is as follows: a man dies leaving a heritage and debts to three different creditors with amounts of 100, 200, 300 units. According to Talmud's rule, if the heritage is 100 units, each creditor gets 33.3 units (equal amounts); if heritage is 200 units then creditors receive 50, 75, 75 units respectively and if it is 300 units, they receive 50, 100, 150 units respectively. In this allocation policy the received shares change depending on the total amount of heritage. When heritage is small, each creditor receives equal proportions; however, as the amount on hand increases their shares become proportional to their debt. If the decision maker keeps the equal proportion policy in the third scenario, 100% of creditor one's debt will be paid while only 33.3% of creditor three's debt will be covered. From creditor three's perspective this allocation may not be acceptable, therefore in such situations shifting the input distribution from equal amounts to a distribution proportional to the debts may improve fairness.

A similar situation may also be seen in investment planning. [3] and [4] state that investors may become more risk tolerant when the potential output is higher. As the expected benefit gets higher, the investor becomes more prudent to invest in more risky projects, hence the reference allocation changes.

We will explain the novel approach that we suggest for imbalance measurement in resource allocation settings using a small example as follows:

Assume that a local healthcare provider is planning the annual budget allotment for a set of proposed projects, each of which targets a different population group. For simplicity we are going to assume that there are two population groups, which are significantly different with respect to their life styles. Let us assume that the first group represents people who are negligent about their health while the second group consists of people that take care of their health (Such behavioral differences could emerge due to other attributes of the groups such as income.). The provider measures health gain in terms of the total quality adjusted life years (QALYs). Assume that the total QALYs are normalized to obtain a score.

If the total score that would result from the allocation decisions is relatively low (e.g between 0 and 40), the policy makers may prefer to use the resources such that benefit is evenly distributed across population groups. Making the allocation in favor of the group that returns more benefit per unit resource (in our example this corresponds to the second group) would create a significant health gap; thus it may not be appropriate for ethical reasons. Conversely, if total score is relatively high (e.g. between 70 and 100), the overall community health may be considered to be above standards. Since the overall health state is already good, having an imbalanced distribution between population groups may be more acceptable. Both groups already have above satisfactory health status; hence, when benefit distribution shifts towards one of the groups, the difference would not be as striking as the first case. Therefore the policy maker may prefer an allocation where the second population group has higher returns in order to increase total health score. To summarize, distributing a good in an equal manner may be more important when the total welfare is already low. As the total welfare increases, the allocator may want to shift resources to more productive entities.

We illustrate the changing references in the example in Figure 2.1, where there are two categories (population one and population two). The two axes represent the total benefit enjoyed by the categories, i.e. the total health score of each population. As long as total benefit (tb) is below 40 units, decision maker uses the reference proportion α_1. This reference proportion is a vector showing the desired benefit percentages for each group, in this example α_1=(0.5,0.5), i.e. if the total benefit is below 40 units, the policy maker would like to distribute it evenly across the two groups. When the total benefit exceeds 40 units she changes the reference distribution vector from α_1 to α_2 =(0.4,0.6) and in this representation she prefers to see that the bigger portion of the benefit is in category 2. As the output increases reference distribution becomes less even between categories.

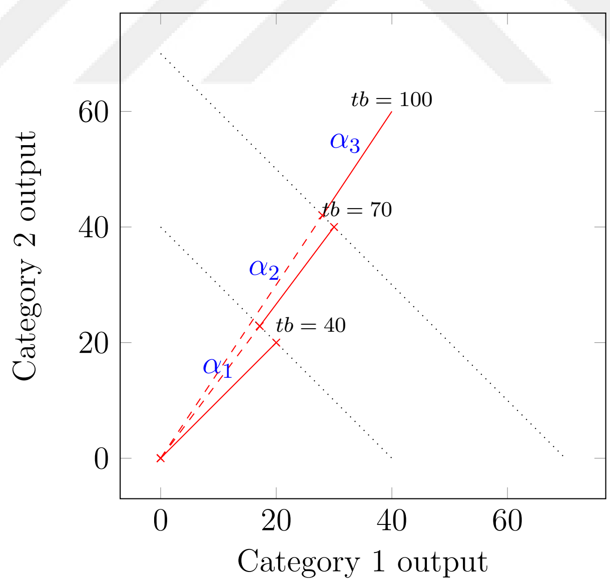

Figure 2.1: Changing reference distributions

This methodology can be applied to incorporate balance concerns in various resource allocation environments which can be formulated as; e.g. *continuous knapsack* problems, where benefits are obtained with respect to some production function, *discrete knapsack* problems, where costs and benefit parameters are explicitly given, or in *assignment problems* with capacity limitations. We illustrate the proposed approach using a discrete knapsack problem. Knapsack problem is a well known combinatorial optimization problem and it is in the class of NP-hard problems [5]. The problems that we formulate will trade balance off against efficiency while making discrete project portfolio selection decisions, hence they can be classified as bi-objective discrete knapsack problems with fairness concerns.

We will discuss two methods to incorporate the notion of changing reference points: The discrete and moving reference distribution methods. In the discrete method, we will assume that the reference proportion changes at predefined threshold values and is constant within each interval as in the example given above. In the moving reference point approach we will allow the reference proportion to move along between two threshold proportions, meaning that, depending on the exact value of total output, reference proportion vector is calculated particularly for each allocation, rather than assigning a single reference proportion vector by an interval based partition.

Chapter 3

Literature Review

The problem considered in this thesis can be classified as a bi-objective project portfolio selection problem with fairness concerns. As the description reveals the problem is related to three main areas; (i) fairness, (ii) project portfolio selection and (iii) bi-objective optimization.

In the first section of this chapter, integration of fairness in OR problems is discussed. [6] states that fairness has become an important criterion in many OR applications. Fairness is a subjective concept, hence it does not have a de facto definition. Depending on the problem setting and the decision maker several different criteria can be listed to describe a fair operation. In this part, we will discuss different implementations of fairness and their application areas studied in the literature. In the second section, the focus is on the multi-objective project portfolio selection problems. The project portfolio selection problem has been studied in the OR literature for over 40 years [7]. It became a very popular problem because it has an impact on various fields and it has a broad range of extensions that yields to new research areas. In the last section, literature on solution methods used for bi-(multi) objective programming problems are analysed. First, a classification of solution methods depending on timing of elicitation of preferences are described. We then discuss some exact methods used for solving bi-objective programming problems. Later we discuss, applications of metaheuristic approaches used in multi-objective resource allocation literature.

3.1 Fairness in OR Problems

OR practitioners started to consider fairness as an important criterion along with efficiency in many applications, such as resource allocation, airline scheduling and energy distribution. Especially in decisions affecting social welfare, fairness becomes an almost mandatory criterion [6].

Karsu and Morton [6] categorize fairness-related concerns as equitability and balance. Equitability aims to achieve an even distribution of the resources (outputs) since it assumes that the entities are indistinguishable and anonymity holds, i.e. the identities of the entities do not affect the decision. Balance concerns, however, are relevant in cases, where the entities have different characteristics. They may, for example, differ in terms of productivity, need or claims. In such cases a completely equal allocation may be undesirable and the decision maker may want to allocate the resource in different proportions. Note that there are two relevant distributions over which balance can be ensured: the resource distribution and the output distribution. Note also that, a balanced resource distribution may not lead to a balanced output distribution and vice versa as seen in the healthcare resource allocation example discussed in Chapter 2.

Equitability and balance are studied in various articles in OR literature and they are integrated in many classical OR problems. One can give examples from a vast area of applications including but not limited to air traffic flow management, slot assignment in airline planning [8],[9] and [10]; food re-distribution [11],[12]; public service provision [13]; public facility location [14] and even in cooperative advertising [15], in which fairness concern for channel members and retailers are taken into account.

There are also studies in the literature that focus on alternative ways of "measuring" fairness. [16] and [17] analyse inequity measures that can be used with the efficiency objectives in location problems. See also [18] for a discussion on different equity measures used in location problems. [19] considers a generic resource allocation setting and discusses mathematical implications of utilitarian and maxmin type (Rawlsian) welfare distributions with respect to equity.

[20] proposes a bi-criteria framework to handle balance and efficiency concerns in resource allocation, and exemplify the approach through a discrete project selection problem. They suggest using a reference distribution approach to measure how balanced a given distribution is. [21] uses a reference point approach to solve a real life multi objective resource allocation problem where quantity, quality and balance are the main concerns. Their balance definition is similar to the one used in [20], where a desired proportion of input for each category is used as a reference balanced distribution. In their problem categorization is based on several different attributes which are client groups, time horizons and alignment with management objectives. In order to engage fairness criteria into this multi dimensional setting they used a form of chi-squared statistic to measure imbalance.

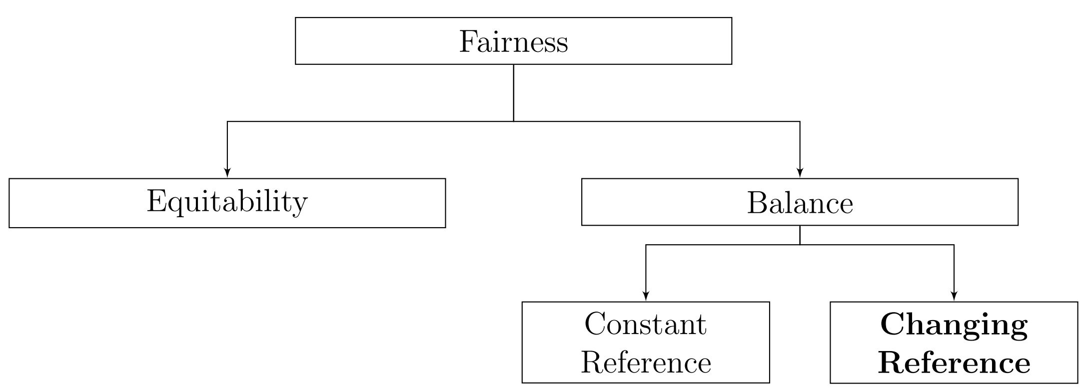

Figure 3.1: Fairness concept categorization in literature

Figure 3.1 illustrates the categorization used to describe fairness in the literature. As explained in Karsu and Morton [6] fairness can be identified as either equability or balance. In an equitable allocation, anonymity holds and most fair distribution gives each entity the same amount of output (or input). Main principle is distributing equal proportion of output (or input) among entities regardless of their characteristic properties. On the other hand we have the more generic concept of balance to define fairness, where decision maker can choose the ideal balance point to be different from equal proportions. In balance criteria, anonymity does not hold and the most desired distribution depends on decision maker. Decision maker distinguishes entities and desired allocation is affected from the characteristic properties of entities, such as productivity and needs, along with the structure of decision environment.

There are two ways of incorporating balance to a problem. First, definition of a balanced distribution can be unique and in all possible scenarios of the problem same ideal proportion vector can be used as a reference. This method is called "Constant Reference" on Figure 3.1 and it is studied in [20] and [21]. Another way of integrating balance to a problem is allowing the decision maker to change the definition of balanced distribution depending on certain characteristics of the problem such as amount of available resource, potential output level and types of goods. This method is referred as "Changing Reference" on the figure and in this thesis we focus on the mathematical formulation, implementation and solution methods of this problem.

In this thesis we also assume that the decision maker has balance concerns for the output distribution and provide a methodology to help decision makers to trade balance off against efficiency in resource allocation settings. We extend Karsu and Morton's [20] idea of balance measurement to a piecewise linear concept, which allows the decision maker to change her reference distribution depending on total output. This extension significantly changes the structure of the multi-objective problem models solved, as will be discussed in Chapter 4.

3.2 Multi Objective Project Portfolio Selection

Project portfolio selection problem is planning the allocation of a limited amount of resource to a set of projects, that will return some type(s) of output. There can be different preferences and priorities of decision makers in the process of project selection. Hence in many portfolio selection problems a set of diverse objective functions are simultaneously used. In many cases these objectives are conflicting, such as maximizing revenue, minimizing risk, minimizing cost and maximizing utilization. This multi-objective structure necessitates the use of multi-objective programming models. A multi-objective integer programming model, where all variables are binary, can be expressed in the following way:

$$\text{Optimize} \qquad Z(x) = \{z_1(x), z_2(x), ..., z_p(x)\} \tag{3.1}$$

$$\text{s.t.} \qquad x \in X \qquad (x_i \in \{0,1\}\,; \quad i = 1, 2, ..., n) \tag{3.2}$$

where x is the binary decision vector and in the project selection problem its dimension equals to the number of candidate projects $n = N$. X is the feasible set in the decision space. $x_i = 1$ indicates that project i is selected and $x_i = 0$ indicates the opposite. $z_j(x)$ is the j^{th} objective function, $j = 1, ..., p$, [22].

Multi-criteria project portfolio selection problem has been studied in a wide range of areas with several different methodologies. [23] studies long- and short-term valuation of IT investments with high outcome uncertainties. [24] and [25] integrate analytical network process and goal programming to solve the information system project selection problem with interdependencies. [26] develops a two phased method to solve project selection problem under uncertainty. [27] discusses the use of analytic network process to evaluate value of different R&D project selection settings. [28] uses goal programming to select projects in multi-objective healthcare management problems.

The task of selecting a set of projects with a certain amount of resource constitutes a knapsack problem. Knapsack problem is an NP-hard combinatorial optimization problem and it has many implementations in various fields. In the original knapsack problem, resource is the capacity of the knapsack; projects are the items, each entitled with a weight and benefit value. Capacity of the knapsack is not sufficient to fit all the items, hence a set of items has to be selected so as to maximize total benefit without exceeding the capacity. It is also called 0-1 knapsack problem [29], [30] and [5].

In the multi-objective extension of the problem, items are associated with multiple benefit values. Generic formulation of the multi-objective knapsack problem is provided below [30];

$$\text{Max} \quad z(x) = \{z_1(x), z_2(x), ..., z_p(x)\} \tag{3.3}$$

$$\text{s.t.} \qquad \sum_{i}^{n} c_i x_i \leq B \tag{3.4}$$

$$x_i \in \{0, 1\}\,; \quad i = 1, 2, ..., n \tag{3.5}$$

In the model, c_i is the weight of item $i = 1, 2, ..., n$. Constraint (3.4) ensures that capacity is not violated. b_{ij} is the benefit of item $i = 1, 2, ..., n$ for $j = 1, 2, ..., p$ and objective functions are in the following form;

$$z_j(x) = \sum_{i}^{n} b_{ij} x_i \quad j = 1, 2, ..., p \tag{3.6}$$

To solve multi-objective project portfolio selection problems, multi-objective knapsack formulation or its variations are frequently used in the literature, There is a vast number of articles in the literature using multi-objective knapsack formulation or a variation of it to solve project portfolio selection problems [26], [31], [32], [33], [34] and [20].

The problem encountered in this thesis can be generically described as a bi-objective knapsack problem. However, unlike the formulation provided above, in our problem there is a single resource and items are entitled with a particular cost and benefit values. The first objective function maximizes efficiency and the second objective function aims to minimize imbalance in the distribution. The mathematical formulation and details of our model is further discussed in Chapter 4.

3.3 Solution Methods Used for Bi-(Multi) Objective Programming Problems

Bi-objective optimization is a reduced version of multi-objective optimization to two objective functions. As a result of project portfolio selection problems' multifaceted structure, they are commonly solved with multi-objective programming models. Note that, optimizing a vector is not a well-defined operator, hence solving a multi-criteria decision making model may refer to finding all (or a subset of) the nondominated solutions of the problem or finding the solution that is most preferred by the decision maker. All definitions provided below are for maximization settings.

Definition 1: $z(x') = (z_1(x'), z_2(x'), ..., z_p(x'))$ is said to dominate $z(x) = (z_1(x), z_2(x), ..., z_p(x))$ is $z_j(x') \geq z_j(x) \quad \forall j$ and $z_k(x') > z_k(x)$ for at least one k.

Definition 2: If $\nexists x \in X$ such that $z(x')$ dominates $z(x)$ then $z(x)$ is nondominated.

Definition 3: If $z(x)$ is nondominated then x is efficient.

While solving a multi criteria decision making problem, a key point is the timing of articulation of preferences. As demonstrated in Figure 3.2 there are three main classifications depending on the timing of elicitation of preferences; (i) a priori, (ii) a posteriori and (iii)interactive (progressive) approaches [35].

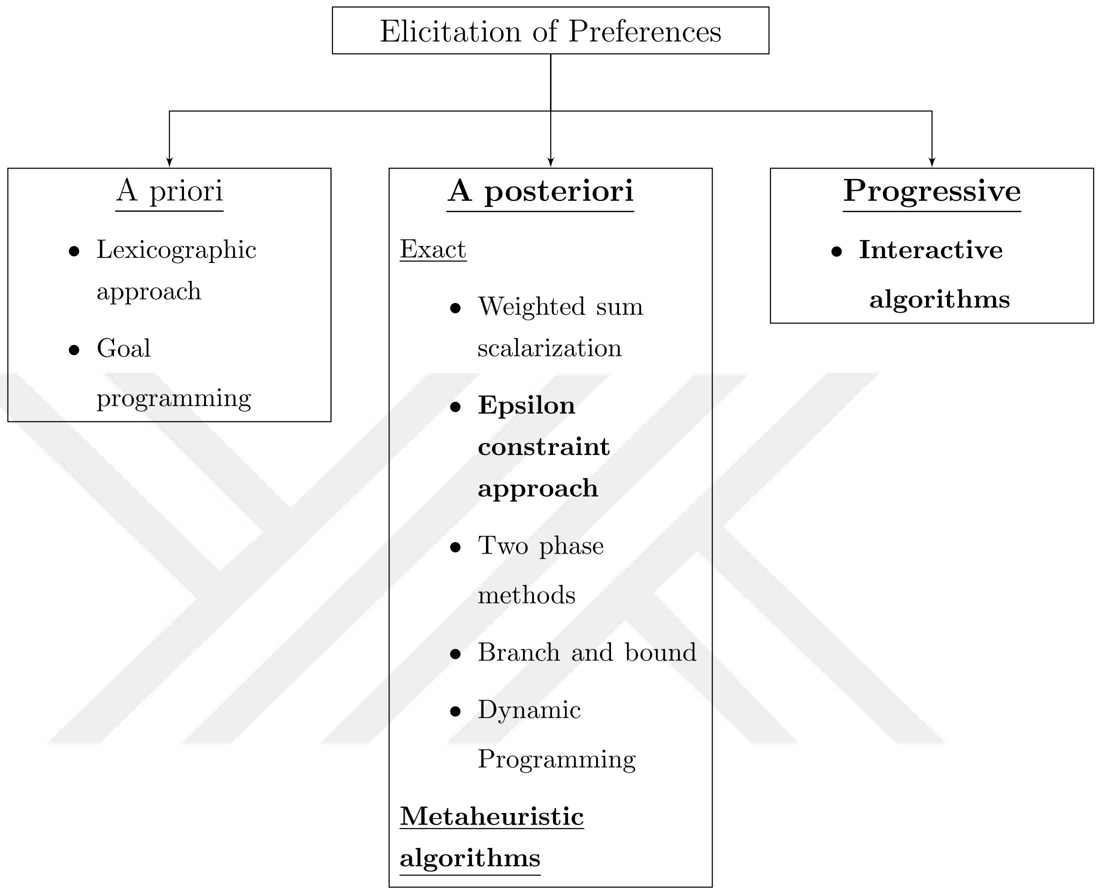

Figure 3.2: Categorization by timing of elicitation of preferences

Prior articulation of preferences requires the use of explicit value function of the decision maker. For instance, in lexicographic ordering, value function is assumed to be a lexicographic function of the criteria. In goal programming, there are some predetermined goals for each objective and value function is assumed to be a weighted minimization of deviations from these goals. Even though, these approaches are relatively easier to handle, it is very difficult to determine value function of the decision maker over multiple objectives. Hence these methods are not always suitable for practical use [35].

Algorithms using posterior articulation of preferences aim to present all the nondominated (Pareto) solutions of a problem to the decision maker. Then, the decision maker is expected to select the most preferred solution among all nondominated solutions. The main and only assumption of this method is that decision maker's value function satisfies monotonicity; in other words "more is better". Hence there is no need to know the exact value function to represent decision makers preferences. As long as monotonicity holds these methods can be employed to find the Pareto front. However this approach can be very complex for decision makers since the number of Pareto solutions can be very large to analyse. Some commonly used methods are weighted sum scalarization, epsilon constraint approach, two phase methods, branch and bound methods and metaheuristic algorithms [35].

In the progressive articulation of preferences the purpose is to find the most preferred alternative by iteratively using the preference information obtained from the decision maker. At each iteration the algorithm asks the decision maker questions about preference relations among a set of alternatives. Then using this information, solution space is reduced and a new set of solutions are presented to the decision maker. The algorithm stops when the most preferred (or a close enough) solution is obtained.

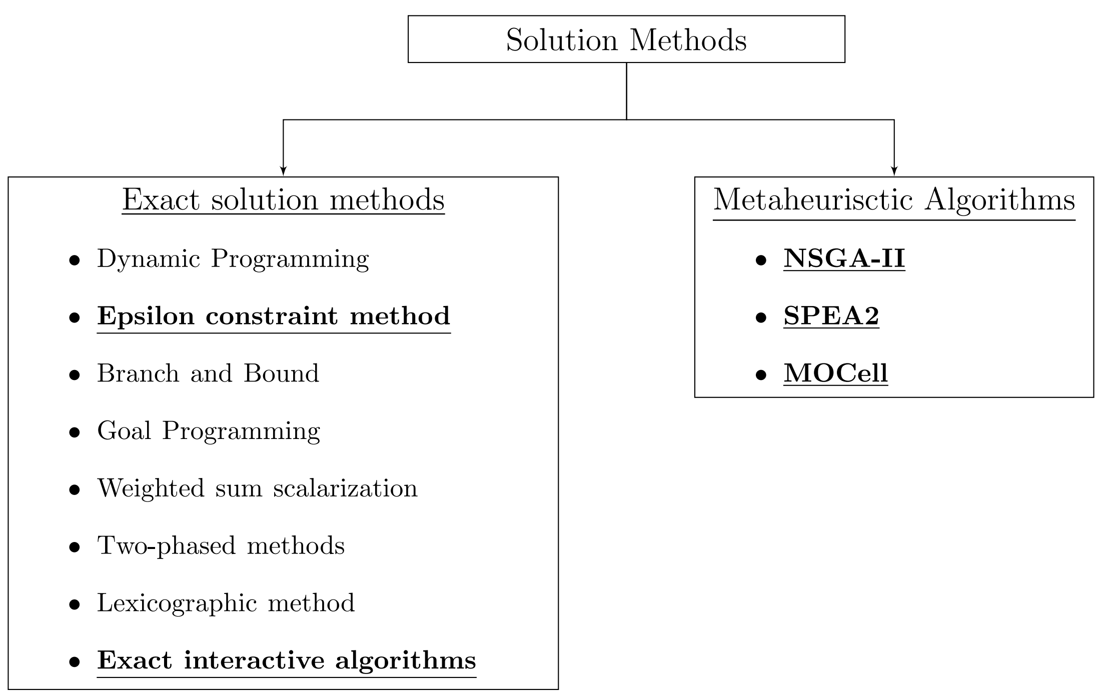

Figure 3.3: Solution methods for bi-objective programming problems

Solution methods used to solve multi-objective optimization problems can also be categorized into two main groups depending on whether true Pareto solutions are found or not; exact solution methods and metaheuristic algorithms. This categorization and frequently used methods are illustrated in Figure 3.3. Exact methods guarantee exact Pareto front however they require solving large number of optimization models and that can take considerable amount of time. On the other hand metaheuristic algorithms are computationally-efficient but they only provide an approximation of the real Pareto. Some frequently used metaheuristic algorithms are Non-dominated Sorting Genetic Algorithm (NSGA) [36], Strength Pareto Evolutionary Algorithm (SPEA) [37] and MOCell [38].

Metaheuristic algorithms can find multiple solutions with a single simulation run. They are flexible and can be easily adjusted or combined depending on the problem specifics to improve efficiency. Thus, metaheuristic algorithms are adopted to solve many multi-objective resource allocation problems in the literature. [39] develops a multi-objective generic algorithm to approximate Pareto front of a linear resource allocation problem with project interdependencies, stochastic objectives and uncertain parameters. [40] combines two metaheuristic algorithms, Tabu Search and Scatter Search, to solve the selection and scheduling problem of project portfolios. [41] solves the multi objective, multi constraint knapsack problem by using both exact (Branch and Bound) and metaheuristic algorithms (SPEA2 and NSGA-II).

As highlighted in the Figure 3.3, initially we used epsilon constraint method to find all exact nondominated solutions of our problem. However as the problem size grows it becomes difficult to obtain exact solutions in reasonable time limits. In order to overcome this computational difficulty we adopt metaheuristic algorithms; NSGAII, SPEA2 and MOCell to find approximate Pareto fronts. Then we employ an interactive algorithm so as to obtain the most preferred solution of the decision maker without obtaining the whole Pareto front. We demonstrate the details of these methods and corresponding performance analysis in Chapters 5 and 6 respectively.

Chapter 4

Mathematical Programming formulations

In this chapter, we first introduce the structure and mathematical formulation of Discrete Reference Distribution Model. Then, characteristics of Pareto solutions obtained with this model are briefly discussed with an example. In the following section, we provide an improved approach to formulate the balance concerns in the problem, which is referred to as Moving Reference Distribution Model.

4.1 Discrete Reference Distribution Model (DRDM)

We consider a bi-objective discrete knapsack problem to illustrate our resource allocation mechanism. In the problem setting, suppose there are N proposed projects (N is the set of projects) and each project incurs a cost of c_i units and returns a benefit of b_i units. Projects are divided in different categories such that each project belongs to one (and only one) category. Let K be the set of categories and let $K \leq N$. We assume that the decision maker has a limited budget B, which is not enough to fund all of the projects ($\sum_{i=1}^{N} c_i > B$).

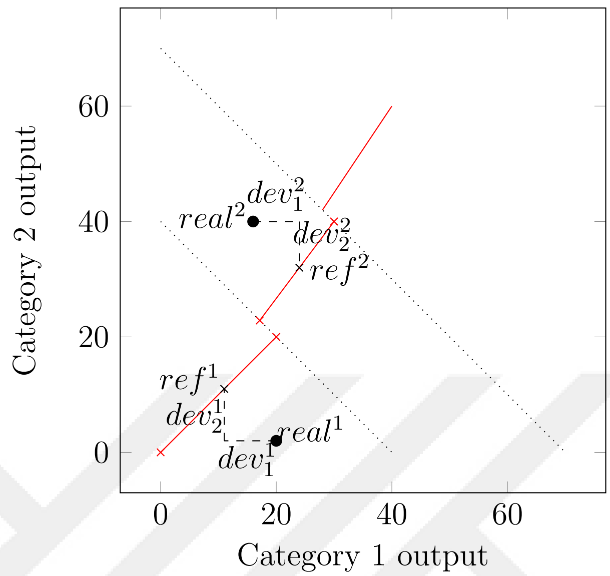

Figure 4.1: Distances from reference distributions

We define the binary variable x_i for $i = 1, ..., N$ taking the value 1 if project i is selected in the portfolio and 0 otherwise. We assume that the degree of balance of any distribution is measured by its distance to a reference allocation, which distributes the benefit according to decision maker's desired (reference) proportions. We allow the DM to determine different reference proportions for different total benefit intervals. Assume that there are M intervals, each of which is defined by two threshold levels (interval m is defined by threshold levels T_{m-1} and T_m). α_{mk} is the reference proportion showing the desired proportion of total benefit allocated to category $k = 1, ..., K$ if the total benefit is in interval $m = 1, ..., M$. α_{mk} can increase/decrease as we move along on the intervals to shift the allocation in favor of particular categories.

Imbalance of a realized allocation is measured as the ratio of total componentwise deviation from reference distribution to the total benefit of that allocation. Figure 4.1 shows the componentwise deviations of two realized allocations to their reference allocation vectors. Realized allocations are represented with $real^1$, $real^2$ and x^1, x^2 are the corresponding decision variable vectors. Note that, these two realized allocations have total benefits lying in different intervals and their imbalance level should be calculated with respect to their own reference proportion vectors. By using the corresponding α values we calculate reference distributions as follows; $ref_k^1 = \sum_{i=1}^{N} \alpha_{1k} b_i x_i^1$ and $ref_k^2 = \sum_{i=1}^{N} \alpha_{2k} b_i x_i^2$, $k = 1, 2$. Componentwise deviation from reference distribution can be measured as $dev_k^j = |real_k^j - ref_k^j| \quad j = 1, 2 \quad k = 1, 2$. Then, $imbalance^j = \dfrac{\sum_{k=1}^{K} dev_k^j}{\sum_{i=1}^{N} b_i x_i} \quad j = 1, 2$. This measure allows us to favor the allocation with a higher benefit when two solutions have the same deviation from their reference distributions.

Notations used throughout this thesis for the mathematical formulations are introduced below;

Problem Parameters

- N: number of projects
- K: number of categories
- M: number of thresholds
- B: total budget
- c_i: cost of project $i = 1, ..., N$
- b_i: benefit of project $i = 1, ..., N$
- T_m: threshold values defining intervals for total benefit values $m = 1, ..., M$
- α_{mk}: reference (desired) proportion for category k in interval m
  $\sum_{k=1}^{K} \alpha_{mk} = 1 \quad \alpha_m \in R^K \quad k = 1, ..., K \quad m = 1, ..., (M-1)$
- g_{ik}: 1 if project $i = 1, ..., N$ belongs to category $k = 1, ..., K$, and 0, otherwise.
- TB: total benefit of all the suggested projects $\sum_{i=1}^{N} b_i$
- ΔT_m: Difference between two consecutive threshold values, $T_{m+1} - T_m$ $m = 1, ..., (M-1)$

Decision variables

- x_i: 1 if project i is accepted, and 0, otherwise, $i = 1, ..., N$.
- $\overline{X}_m$: amount of benefit gained within the interval $m = 1, ..., (M-1)$
- y_m: 1 if the total amount of benefit is higher than T_{m+1}, $m = 1, ..., (M-2)$, and 0, otherwise
- α_{0k}: selected reference proportion for each category $k = 1, ..., K$, $\sum_{k=1}^{K} \alpha_{0k} = 1$
- $Imbalance$: Imbalance indicator of the benefit distribution

Discrete Reference Distribution Model (DRDM)

The proposed bi-objective Discrete Reference Distribution Model (DRDM) is provided below;

$$\max \quad \sum_{i=1}^{N} b_i x_i, \quad \min \quad Imbalance \tag{4.1}$$

$$\sum_{i=1}^{N} c_i x_i \leq B \tag{4.2}$$

$$\sum_{i=1}^{N} b_i x_i = \sum_{m=1}^{M-1} \overline{X}_m \tag{4.3}$$

$$\overline{X}_1 \leq \Delta T_1 \tag{4.4}$$

$$\overline{X}_m \geq \Delta T_m y_m \quad m = 1, ..., (M-2) \tag{4.5}$$

$$\overline{X}_m \leq \Delta T_m y_{m-1} \quad m = 2, ..., (M-1) \tag{4.6}$$

$$\Delta T_m - \overline{X}_m \leq \Delta T_m (1 - r_m) \quad m = 1, ...(M-2) \tag{4.7}$$

$$\Delta T_m - \overline{X}_m \geq (1 - r_m) \quad m = 1, ...(M-2) \tag{4.8}$$

$$r_m \leq y_m \quad m = 1, ..., (M-2) \tag{4.9}$$

$$\alpha_{0k} = \alpha_{1k}(1 - y_1) + \sum_{m=1}^{M-2} \alpha_{m+1k}(y_m - y_{m+1}) + \alpha_{M-1k} y_{M-2} \quad k = 1, ..., K \tag{4.10}$$

$$Imbalance = \frac{\sum_{k=1}^{K} \left| \sum_{i=1}^{N} b_i \alpha_{0k} x_i - \sum_{i=1}^{N} b_i g_{ik} x_i \right|}{\sum_{i=1}^{N} b_i x_i}, \tag{4.11}$$

$$x_i \in \{0, 1\} \quad i = 1, ..., N \tag{4.12}$$

$$\overline{X}_m \geq 0, \quad m = 1, ..., (M-1) \tag{4.13}$$

$$y_m \in \{0, 1\} \quad m = 1, ..., (M-2) \tag{4.14}$$

Constraint (4.2) ensures that the budget is not exceeded. Constraint (4.3) is used to calculate the total benefit of the projects which are selected in the portfolio. Constraints (4.4) - (4.6) are used to identify the interval of total benefit. If total benefit corresponds to interval m then $\{y_j = 1 \forall j < m\}$ and $\{y_j = 0 \ \forall j \geq m\}$. Constraints (4.7), (4.8) and (4.9) are used to indicate the interval of the threshold values. Whenever total benefit is exactly equal to a threshold value, T_m, it is included to interval $m + 1$ and α_{m+1k} is assigned to this solution. Constraint (4.10) makes sure that α_0 is equal to the reference proportion vector of the corresponding interval. Imbalance is measured in (4.11), it is calculated as the total proportional deviation from the reference benefit distribution, ([20]). $\sum_{i=1}^{N} b_i \alpha_{0,k} x_i$ is the desired (reference) benefit in category $k = 1, ..., K$ and $\sum_{i=1}^{N} b_i x_i g_{ik}$ is the actual benefit in category $k = 1, ..., K$. We add up the absolute difference between desired and realized benefits over all categories and obtain a relative measure by dividing it to the total realized benefit. Since total benefit would differ in each feasible portfolio, using a relative measure is more meaningful.

Note that constraint (4.11) involves nonlinear terms. We linearise them using additional auxiliary variables; w_{ki}, f_i, p_k, and d_k. Then we obtain the following model, Constraints (4.1)-(4.10), (4.12)-(4.14)

$$w_{ki} \leq x_i \quad k = 1, ..., K \quad i = 1, ..., N \tag{4.15}$$

$$w_{ki} \leq \alpha_{0k} \quad k = 1, ..., K \quad i = 1, ..., N \tag{4.16}$$

$$w_{ki} \geq \alpha_{0k} - (1 - x_i) \quad k = 1, ..., K \quad i = 1, ..., N \tag{4.17}$$

$$\sum_{i=1}^{N} b_i w_{ki} - \sum_{i=1}^{N} b_i g_{ik} x_i \leq d_k \quad k = 1, ..., K \tag{4.18}$$

$$\sum_{i=1}^{N} b_i g_{ik} x_i - \sum_{i=1}^{N} b_i w_{ki} \leq d_k \quad k = 1, ..., K \tag{4.19}$$

$$d_k - (\sum_{i=1}^{N} b_i g_{ik} x_i - \sum_{i=1}^{N} b_i w_{ki}) \leq 2TB(1 - p_k) \quad k = 1, ..., K \tag{4.20}$$

$$d_k - (\sum_{i=1}^{N} b_i w_{ki} - \sum_{i=1}^{N} b_i g_{ik} x_i) \leq 2TBp_k \quad k = 1, ..., K \tag{4.21}$$

$$f_i \leq Imb^{UB} x_i \quad i = 1, ..., N \tag{4.22}$$

$$f_i \leq Imbalance \quad i = 1, ..., N \tag{4.23}$$

$$f_i \geq Imbalance - Imb^{UB}(1 - x_i) \quad i = 1, ..., N \tag{4.24}$$

$$\sum_{k=1}^{K} d_k = \sum_{i=1}^{N} b_i f_i \tag{4.25}$$

$$w_{ki} \geq 0, \quad k = 1, ..., K \quad i = 1, ..., N \tag{4.26}$$

$$p_k \in \{0, 1\} \quad k = 1, ..., K \tag{4.27}$$

$$d_k \geq 0, \quad k = 1, ..., K \tag{4.28}$$

$$f_i \geq 0 \quad i = 1, ..., N \tag{4.29}$$

Constraints (4.15)-(4.17) are used to linearise the product of α_{0k}(continuous) and x_i(binary) decision variables such that w_{ki} is equal to this product ($w_{ki} = \alpha_{0k} x_i$). With constraints (4.18)-(4.21) and the auxiliary variables d_k and p_k, we rewrite the absolute deviation value term with linear inequalities. New variable d_k is the deviation in category k from the reference distribution. TB is an upper bound on this absolute value. Constraint (4.11) involves the term $Imbalance \sum_{i=1}^{N} b_i x_i$: the product of a continuous and a binary variable. We define $f_i = Imbalancex_i$ for $i = 1, ..., N$, and we linearise this term in (4.22)-(4.24). We take the lower bound on imbalance as 0 and the upper bound (Imb^{UB}) as $K \cdot TB/min_i c_i$. After the linearisation we obtain constraint (4.25). In the linearised model there are $5M + 3NK + 5K + 3N - 6$ constraints and $2N + 3K + 3M + KM - 4$ variables, $2M + N + K - 4$ of which are binary.

As an example, we solve a model with $N = 30, K = 4$ using the well-known epsilon-constraint approach, described in Chapter 5. Figure 4.2 shows the set of all nondominated solutions. At one end of the Pareto front we have a completely balanced allocation with 280 units of total benefit, on the other hand there is the solution with 576 units of benefit and 0.33 unit of imbalance. The reference proportion vector used for the most balanced solution is $\alpha_1 = (0.25, 0.25, 0.25, 0.25)$. As the total benefit increases reference proportion vector changes and it becomes harder to achieve better imbalance values. Figure 4.3 represents the change in realized benefit distributions with respect to their references as we move towards the most efficient solution. Categories 3 and 4 are more productive than category 1 and 2, it is clear that as we move towards the efficiency extreme their outcomes are significantly higher than those of the first two categories'.

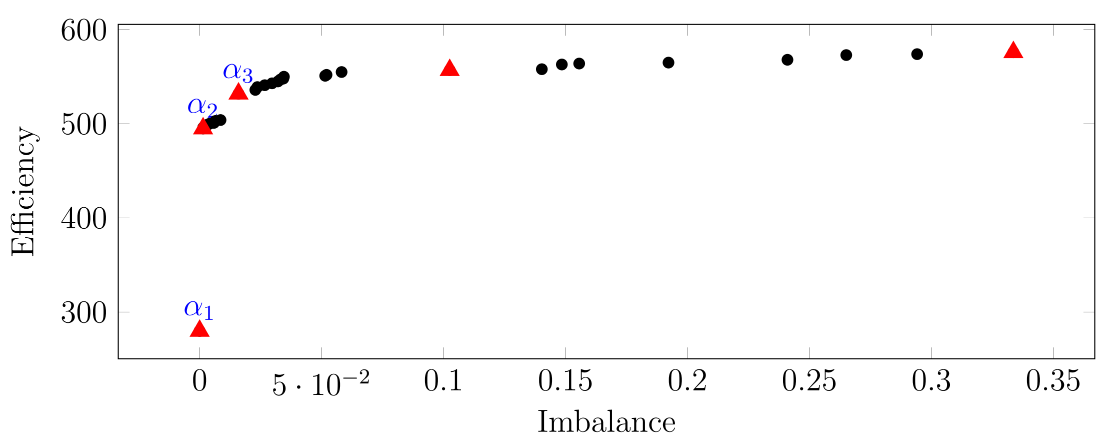

Figure 4.2: Pareto solutions of DRDM Type C $N = 30, K = 4$

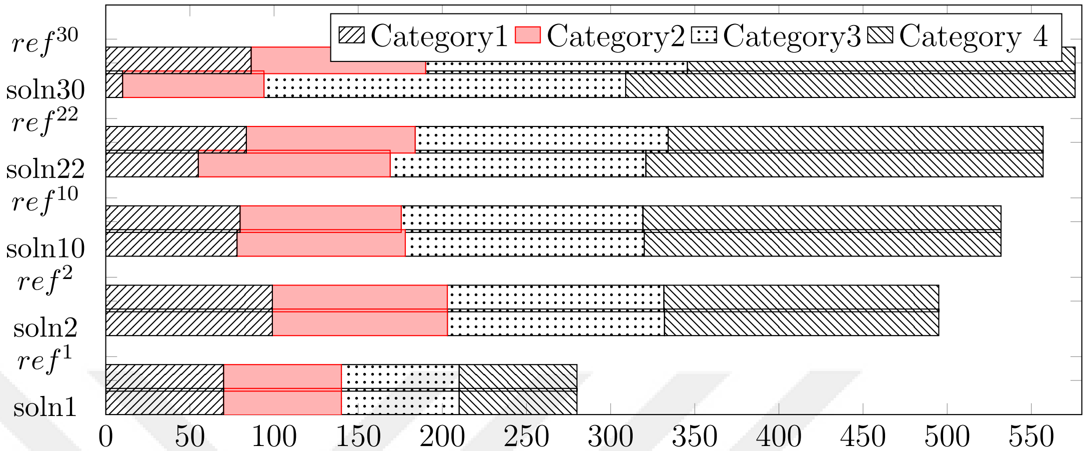

Figure 4.3: Reference benefit distribution (ref) vs. Realized benefit distribution ($soln$)

In this thesis we made the analysis assuming that as the total benefit increases decision maker prefers to make the allocation in favor of more productive categories. However in certain problem settings decision maker may prefer a more even distribution when total benefit is higher. As the total benefit increases, uneven distributions may result in significant differences among categories. For instance in a setting where there are two categories and reference proportion vector is set to (0.3,0.7) when total benefit is 10 units, categories get 3 and 7 units of benefit, respectively. On the other hand in the same setting if total benefit is 1000 units, categories get 300 and 700 units of benefit, respectively. As seen in the example when total benefit is higher, difference between categories' benefit level is significantly higher.

In order to reflect the affect of reversing balance definition in different levels of benefit, we resolved the problem instance presented in Figures 4.2 and 4.3 with this reversed policy and its results are provided in Figures 4.4 and 4.5. In the given example reference proportion vectors are reversed over the intervals; meaning that as the total benefit increases, more even reference proportion vectors are assigned. In this policy the most balanced solution has a higher efficiency score since more productive categories are promoted in lower benefit levels.

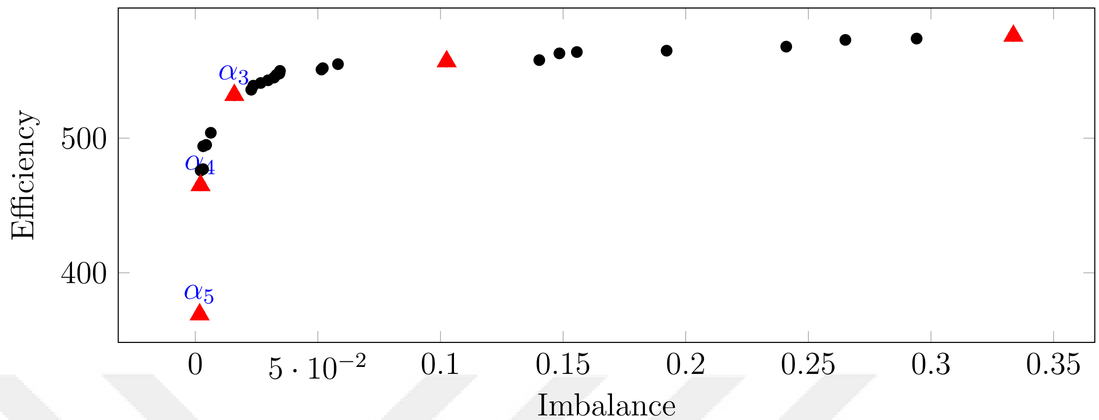

Figure 4.4: Pareto solutions of DRDM Type C $N = 30, K = 4$ with reversed reference proportion vector

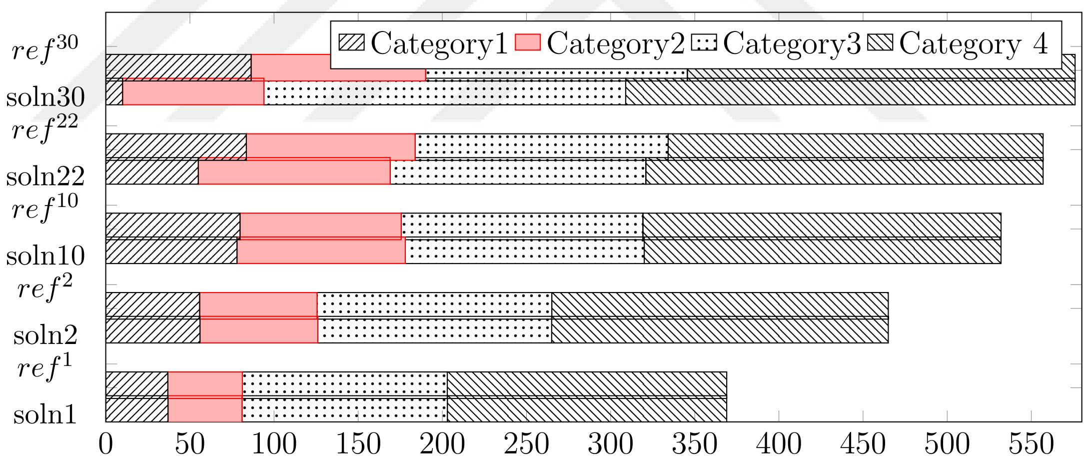

Figure 4.5: Reference benefit distribution (ref) vs. Realized benefit distribution ($soln$) with reversed reference proportion vector

4.2 Moving Reference Distribution Model (MRDM)

In this chapter we present a modified version of our initial model. We begin with pointing out a noteworthy observation from the solutions of DRDM that brings out the need for this modification. Following this observation, we discuss the necessary adjustments for this new approach and provide the reformulation of the model.

In DRDM there are multiple reference proportion vectors and the model assigns an α_{0k} based on the interval of the total benefit. Note that the same reference distribution is used for the entire interval between two thresholds, regardless of its distance from the threshold points. Although this may seem reasonable, in some cases it may result in abrupt changes in the solutions, especially for the ones with total benefit values closer to the thresholds.

Let us consider the allocation policy in Figure 4.6 with two categories. Until total benefit reaches $T_1 = 100$ units, α_0 is $(0.4, 0.6)$ and afterwards it changes to $(0.3, 0.7)$. Suppose that there are two Pareto solutions with efficiency values 96 and 102 units. In the first solution, with 96 units of benefit, benefits are distributed as (38,58) units between categories. In the second solution total benefit is in the second interval, hence reference distribution vector shifts to (0.3,0.7) and benefit distribution becomes (31,71) units. Note that in the second solution the benefit of category 1 decreases by 18% while benefit of category 2 increases by 22 %. Even though total benefit scores of these two solutions are close to each other, their distributions have significant differences. It may be expected that as the total benefit increases, each categories' benefit will also increase; however, this may not be possible for the entire Pareto frontier. In the neighborhood of thresholds, results do not coincide with this expectation. For a fact, these increases/decreases are inevitable due to the selected reference distribution coefficients: while one category' share is increasing others' have to decrease. Thus in the threshold neighborhoods there are sudden jumps from one distribution to the other, which may be undesired.

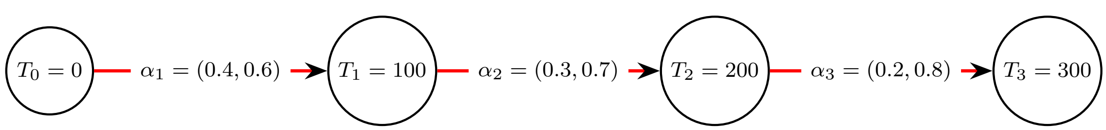

Figure 4.6: Example setting

In order to address this drawback, we enhanced our model to move α_{0k} values from one breaking point to the other by taking the convex combination of the two reference distribution vectors depending on the exact position of total benefit. This way, benefits of entities will gradually increase/decrease as the total benefit changes. In the previous example; where we have 96 units of benefit the corresponding reference distribution will be calculated with the equation $\alpha_{0k} = \overline{X_1}\dfrac{(\alpha_{2k} - \alpha_{1k})}{T_1 - T_0} + \alpha_{1k}$, in this example it is equal to (0.304,0.696). In the second solution, with 102 units of benefit, reference distribution becomes (0.298,0.702). Consecutively benefit distributions are (29,71) and (30,70), respectively. By moving the reference distribution vector from one breaking point to the other we can achieve a more smooth change in distributions.

We modify DRDM to formulate this change. Reference distribution constraint (4.10) is reformulated as a piecewise linear continuous equation provided below and this adjusted model is referred to as the Moving Reference Distribution Model (MRDM);

$$\alpha_{0k} = (1 - y_1)\left[\frac{\overline{X}_1(\alpha_{2k} - \alpha_{1,k})}{\Delta T_1} + \alpha_{1k}\right] + \\ \sum_{m=1}^{M-3}(y_m - y_{m+1})\left[\frac{\overline{X}_{m+1}(\alpha_{m+2k} - \alpha_{m+1k})}{\Delta T_m} + \alpha_{m+1k}\right] + y_{M-2}\alpha_{M-1k} \quad k = 1, ..., K \tag{4.30}$$

This term is non-linear due to the product of variables y_m (binary) and $\overline{X_m}$ (continuous). We linearise it with the additional decision variable t_{ij} and the new formulation contains the constraints (4.1)-(4.9) and (4.12)-(4.29) with the following ;

$$\alpha_{0k} = \left[\overline{X_1}\frac{(\alpha_{2k} - \alpha_{1k})}{\Delta T_1} + \alpha_{1k}\right] - \left[t_{11}\frac{(\alpha_{2k} - \alpha_{1k})}{\Delta T_1} + y_1\alpha_{1k}\right] + \\ \sum_{m=1}^{M-3}\left[\left(\frac{t_{mm+1}(\alpha_{m+2k} - \alpha_{m+1k})}{\Delta T_m} + y_m\alpha_{m+1k}\right) - \right. \\ \left.\left(t_{m+1m+1}\frac{(\alpha_{m+2k} - \alpha_{m+1k})}{\Delta T_m} + y_{m+1}\alpha_{m+1k}\right)\right] + y_{M-2}\alpha_{M-1k} \quad k = 1, .., K \tag{4.31}$$

$$t_{mm} \leq \overline{X}^{UB} y_m \quad m = 1, ..., (M-2) \tag{4.32}$$

$$t_{mm+1} \leq \overline{X}^{UB} y_m \quad m = 1, ..., (M-2) \tag{4.33}$$

$$t_{mm} \leq \overline{X}_m \quad m = 1, ..., (M-2) \tag{4.34}$$

$$t_{mm+1} \leq \overline{X}_{m+1} \quad m = 1, ..., (M-2) \tag{4.35}$$

$$\overline{X}_m - \overline{X}^{UB}(1 - y_m) \leq t_{m,m} \quad m = 1, ..., (M-2) \tag{4.36}$$

$$\overline{X}_{m+1} - \overline{X}^{UB}(1 - y_m) \leq t_{m,m+1} \quad m = 1, ..., (M-2) \tag{4.37}$$

$$t_{mm} \geq 0 \quad m = 1, ..., (M-2) \tag{4.38}$$

$$t_{mm+1} \geq 0 \quad m = 1, ..., (M-2) \tag{4.39}$$

In linearisation upper bound on $\overline{X_m}$, $\forall m = 1, ..., (M-1)$ is set as $\overline{X}^{UB} = \max_{m=1}^{M} (\Delta T_m)$ and the lower bound is zero. This model has $11M + 3NK + 5K + 3N - 18$ constraints and $2N + 3K + 5M + KM - 8$ decision variables.

Chapter 5

Solution Methods

In this chapter solution methods used to solve the proposed bi-objective discrete resource allocation problem are introduced. As mentioned in Chapter 3 we used the Epsilon constraint method and an interactive algorithm to find the exact Pareto front (or a subset of it) and metaheuristic algorithms SPEA2, NSGA-II and MOCell for the approximate results.

5.1 Epsilon Constraint Method

In a multi-objective optimization problem suppose there are p objective functions. As stated in Chapter 3, multi-objective optimization problems (for a maximization setting) can be modelled as follows;

$$\max \quad Z(x) = \{z_1(x), z_2(x), ..., z_p(x)\} \tag{5.1}$$

$$\text{s.t.} \quad x \in X \tag{5.2}$$

where X is the feasible set of decision space and $z(x)$ is the feasible criteria space.

Epsilon constraint method is categorized within the methods using a posteriori preference articulation, meaning that it aims to find all the nondominated solutions of a given problem. The method is based on solving single objective models repetitively. The generic form of these models is shown below:

$$\max \qquad Z(x) = z_j(x) + \delta \sum_{i \neq j}^{p} z_i(x) \tag{5.3}$$

$$\text{s.t.} \qquad x \in X \tag{5.4}$$

$$z_i(x) \geq \epsilon_i \quad i \neq j \quad i = 1, 2, ..., p \tag{5.5}$$

The method, by systematically changing the ϵ_i values and solving the updated model iteratively, obtains different nondominated solutions. In the bi-objective setting, when the decision space is discrete, it is possible to obtain all nondominated solutions of a problem. Note that, this is an augmented formulation of the method. We use a lexicographic approach instead of this augmented form. The approach we use for our bi-objective mixed integer problem setting is provided in Algorithm 1.

Algorithm 1 Pseudocode of Epsilon Constraint Method

```
Initialize: Set ϵ = 0 and f = 1
while f=1 do
  P(1) :    min  imbalance
             s.t  efficiency ≥ ϵ
               x ∈ X
  if the problem is infeasible then
    f = 0
  else
      objective value vector=(imbalance*, efficiency*). Solve,
    P(2) :    max  efficiency
               s.t  imbalance ≤ imbalance*
                 x ∈ X
      objective value vector=(imbalance**, efficiency**)
    Add this vector to the set of non-dominated solutions.
    Set ϵ = efficiency** + step_size
  end if
end while
We assume that the cost and benefit parameters are all integer, therefore setting
step_size = 1 guarantees that all Pareto solutions are generated.
```

We use this algorithm to solve our bi-objective problems. We have two single objective models $P(1)$ and $P(2)$. $P(1)$ minimizes imbalance while restricting efficiency with a lower bound and $P(2)$ maximizes efficiency with a constraint on imbalance value. We start from the initial solution obtained from $P(1)$ that has the minimum imbalance value, then in each iteration we find a new solution by restricting the efficiency by an ϵ- constraint and solving the given model. In each iteration, both $P(1)$ and $P(2)$ models are solved to ensure that a nondominated solution is found. Algorithm terminates when model $P(1)$ is infeasible, meaning that we reached the solution with the maximum efficiency value and all Pareto front is obtained.

5.2 Metaheuristic Algorithms

Metaheuristic algorithms are frequently used to solve multi-objective optimization problems. They do not guarantee the exact Pareto front however they are computationally very efficient and very flexible to fit different problem types.

As it will be demonstrated in Chapter 6, it is challenging to obtain exact solutions for bigger problem instances with the epsilon-constraint method. Thus, with the hope of finding good solutions in reasonable time, we adopt three metaheuristic algorithms: SPEA2 [42], NSGAII [36] and MOCell [38].

We modified the source codes of these heuristic approaches available at MOEA Framework, which is a free and open source java framework for multi objective optimization [43].

5.2.1 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

Strength Pareto Evolutionary Algorithm (SPEA) is a metaheuristic algorithm first introduced by Zitzler et.al [30] to find a set of approximate Pareto solutions of multi objective optimization problems with a single run. SPEA2 is developed to enhance the strategies used in its predecessor SPEA [42].

SPEA2 starts with initializing a population and creating an archive. Then fitness values of all members in both population and archive set are evaluated. Following the fitness function evaluation, selection is made to create the mating pool. During the selection operation nondominated solutions are copied to the archive set of next generation and if they do not fit, in terms of size, a truncation operation is needed. Recombinations and mutations are then applied to the mating pool and resulting population is updated. The algorithm terminates when it reaches the predefined maximum number of iterations (termination condition). Flowchart of the algorithm is provided in Figure 5.2.1.

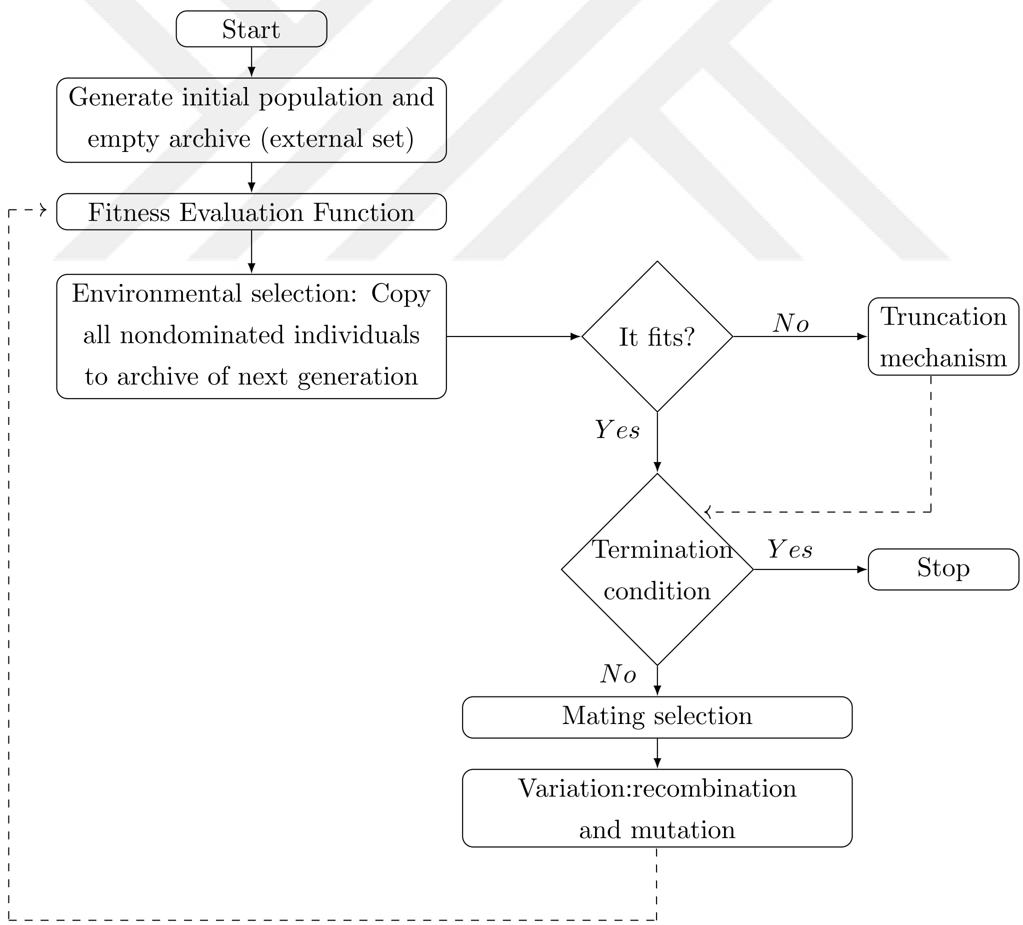

Figure 5.1: Flowchart of SPEA2

5.2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

Nondominated Sorting Genetic Algorithm II (NSGA-II) is a nondominated sorting based evolutionary algorithm proposed by Deb et.al [36]. NSGA-II is known by having good spread of solutions and fast convergence to the Pareto. First, the algorithm generates an initial population and checks whether these initial solutions are feasible or not. If an initial solution is not feasible a repair mechanism is used. Then fitness value of each member is evaluated and solutions are sorted. After the sorting procedure algorithm selects parents and by crossover and mutation operations new solutions are generated. Fitness values of these new solutions are evaluated and sorted to select new parents. Procedure continues until maximum number of generations is attained. Figure 5.2.2 shows flowchart of the algorithm.

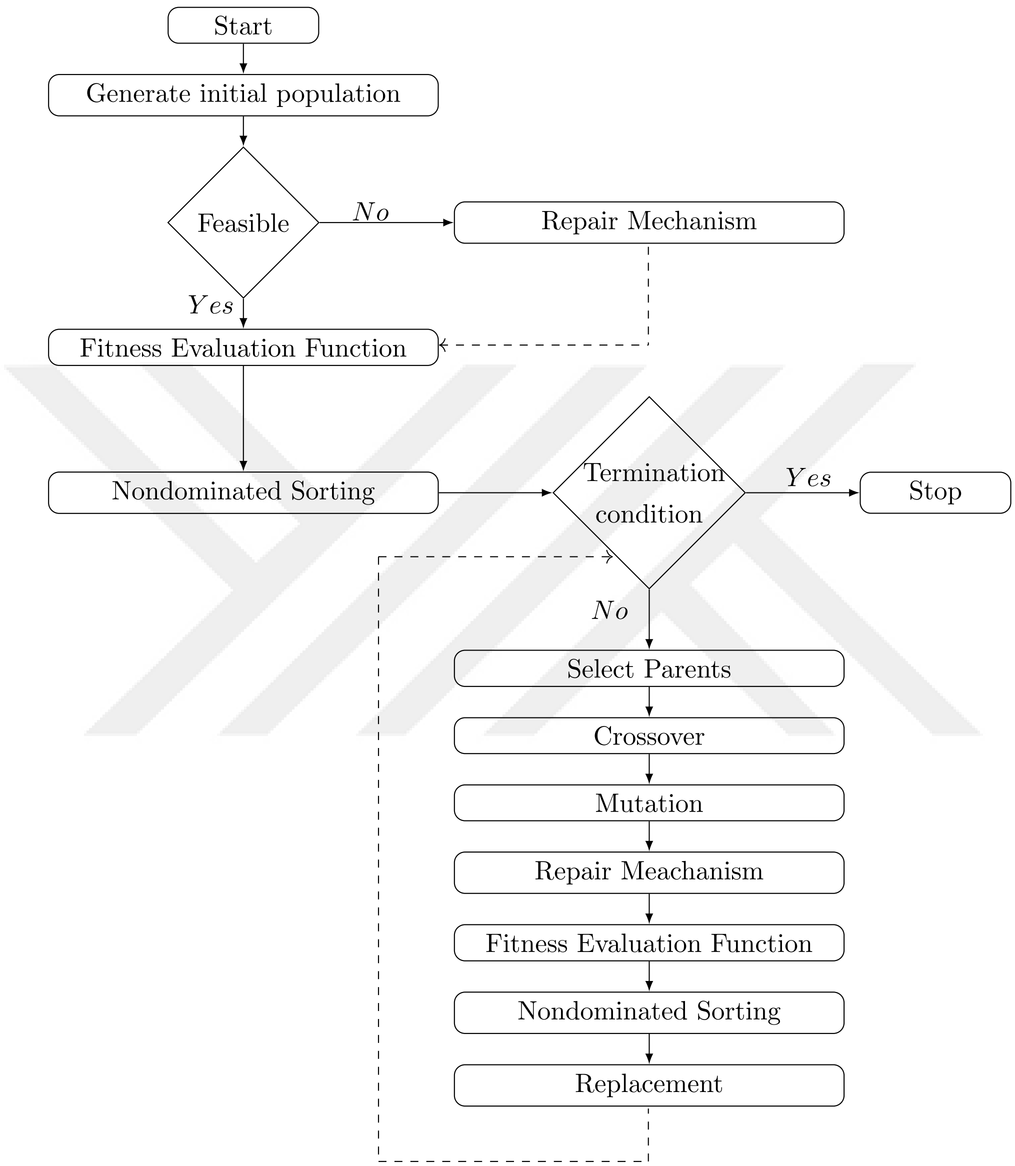

Figure 5.2: Flowchart of NSGA-II

5.2.3 MOCell

MOCell algorithm is developed as an adaptation of cellular model of genetic algorithms and multi-objective framework [38]. In cellular model of genetic algorithms a member in the population may only cooperate with solutions in its own neighbourhood. MOCell algorithm uses an external archive set to store non-dominated solutions, like SPEA2. The speciality of this algorithm is that at the end of each iteration a number of solutions are included back into the population from this external archive by a feedback mechanism. Flowchart of the algorithm is demonstrated in Figure 5.2.3.

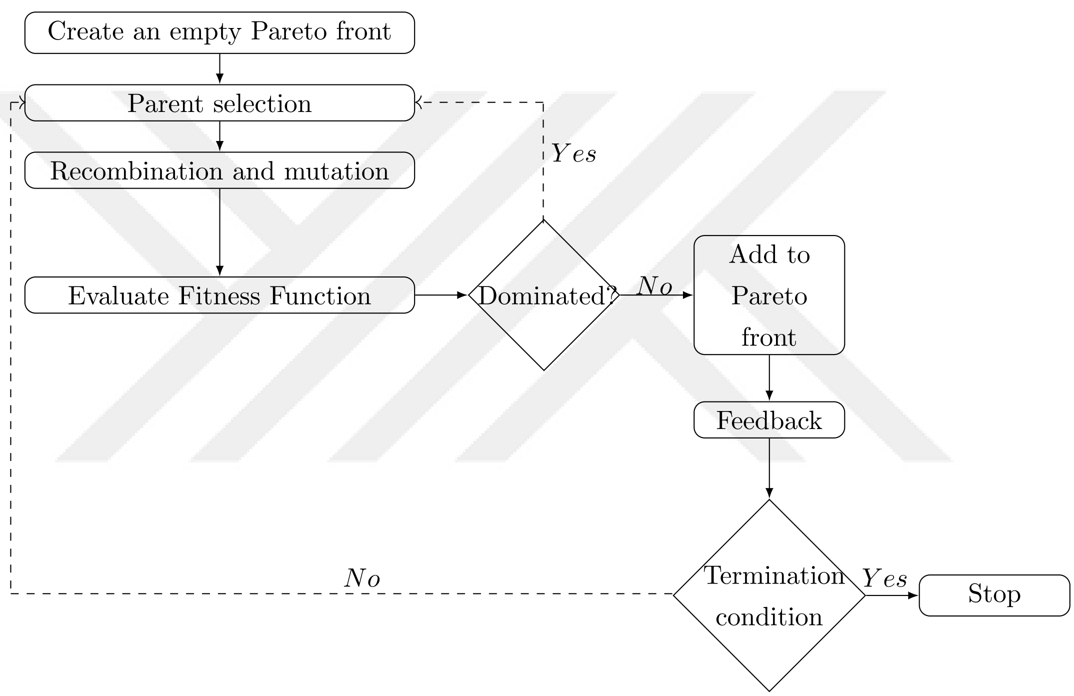

Figure 5.3: Flowchart of MOCell

5.2.4 Performance Metrics

In order to assess the performances of the algorithms in terms of solution quality, we use various performance metrics. Whenever possible, the comparison is based on the exact Pareto frontier. We use the following three metrics for this purpose: Let NS be the set of nondominated solutions and ANS be the solution set obtained from the heuristic algorithms.

P: Ratio of nondominated objective vectors returned by the heuristic to the total number of nondominated solutions.

$$P = \frac{|ANS \cap NS|}{|NS|}$$

P considers the number of nondominated solutions found but it does not measure closeness of heuristic solutions in ANS to their nondominated counterparts in NS.

We also use two distance measures to observe how close ANS is to NS, ([44]). To define the distances, we assume (E^r, I^r) is a vector from set NS and (E^q, I^q) is in set ANS where E and I are efficiency and imbalance scores of the vector, respectively. We calculate the range of the values in exact Pareto solutions.

$$R_1 = \max_{(E^r,I^r)\in NS} E^r - \min_{(E^r,I^r)\in NS} E^r$$

$$R_2 = \max_{(E^r,I^r)\in NS} I^r - \min_{(E^r,I^r)\in NS} I^r$$

$$f((E^r, I^r), (E^q, I^q)) = \max\{0, (E^r - E^q)/R_1, (I^q - I^r)/R_2\}$$

R_1 and R_2 are the component wise range of points in NS. f gives the maximum component wise normalized distance between point q and point r. If solutions q and r are the same, f takes value 0. Distances are defined as follows;

Distance 1 (D1): The average distance between the points of set NS and the points in set ANS.

$$D1 = \frac{1}{|NS|} \sum_{(E^r,I^r)\in NS} \min_{(E^q,I^q)\in ANS} \{f((E^r, I^r), (E^q, I^q))\}$$

Distance 2 (D2): The maximum distance between the points of set NS and the points in set ANS.

$$D2 = \max_{(E^r,I^r)\in NS} \left\{ \min_{(E^q,I^q)\in ANS} \{f((E^r, I^r), (E^q, I^q))\} \right\}$$

Performance is better when D1 and D2 values are smaller.

For problems where $N > 50$ it is not possible to find exact Pareto frontier therefore we compare the results of metahuristic algorithms within themselves. Performance metrics we used to compare different metaheuristic methods are listed below.

Spacing (S): This metric is proposed by [45] to measure distribution of solution vectors over the Pareto front for a single set. It is measured with the standard deviation of pairwise distances.

$$S = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (d_i - \overline{d})^2}$$

$$d_i = \min_j \sum_{k=1}^{m} |O_k^i - O_k^j| \quad i, j = 1, 2, ..., n$$

n is the number of nondominated solutions and m is the number of objectives. In our problem $O^i = (E^i, I^i)$ is a solution vector in set ANS, $\overline{d}$ is the mean value of all d_i. It is better when S value is smaller; when S is equal to zero all points are spread uniformly.

Maximum Spread (MS): The maximum extension covered by the nondominated solution set;

$$MS = \sqrt{\sum_{i=1}^{n} \max_{j=1,\ldots,n} (||O^i - O^j||)}$$

n is the number of solutions in the set, $||O^i - O^j||$ is the Euclidean distance between points O^i and $O^j \in ANS$. The distance of each nondominated solution to all other solutions in the set is calculated ([37]). Higher MS values means solutions can reach further edges of Pareto front indicating better performance.

Set Coverage (SC): Used for comparing two sets of solutions. It calculates the ratio of points that are dominated by at least one of the solutions in the other set.

$$SC(ANS_1, ANS_2) = \frac{|\left\{O \in ANS_2 | \exists \dot{O} \in ANS_1 : \dot{O}_i \leq O_i, \forall i\right\}|}{|ANS_2|}$$

Let ANS_1 and ANS_2 be two sets of solution vectors. SC(ANS_1, ANS_2) can take values between [0,1]. If it equals to 1, points in ANS_1 dominate all the points in ANS_2 (dominance in the weak sense). If SC(ANS_1, ANS_2) equals to 0, then points in ANS_1 are all dominated by solutions in ANS_2. Note that since there can be intersections between the sets, both SC(ANS_1, ANS_2) and SC(ANS_2, ANS_1) should be calculated separately.

Mutual points in the sets ANS_1 and ANS_2 can not indicate any superiority of one set over the other. Since they are identical points they will always dominate each other in a weak sense and their effect on both SC(ANS_1, ANS_2) and SC(ANS_2, ANS_1) will be the same. In order to calculate distinct points' set coverage on the opponent set, we eliminated common points from the analysis by using strong dominance as follows;

$$SC'(ANS_1, ANS_2) = \frac{|\left\{O \in ANS_2 | \exists \dot{O} \in ANS_1 : \exists i \quad \dot{O}_i < O_i\right\}|}{|ANS_2|}$$

We performed set coverage analysis with both SC and SC' approaches, results are provided in Chapter 6.

For further exploration we also made Hypervolume test for these heuristics.

- Hypervolume: Hypervolume is the volume of the dominated space surrounded by the nondominated points and the origin. In bi-objective settings it corresponds to the dominated area. This metric is first introduced by Zitzler and Thiele [30]. We modified the C code prepared by Eckart Zitzler based on the methods introduced in [46] for our bi-objective problem.

5.3 Interactive Approach

We now discuss a progressive approach that could be used to guide the decision maker to her most preferred solution without generating all the nondominated solutions beforehand. This approach has multiple advantages over a-posteriori approaches: It requires less time as it does not generate the nondominated solutions that would not be of interest to the decision maker. It also helps the decision maker by guiding her to the most preferred solution as opposed to just presenting her the set of Pareto solutions, which may be too large.

The interactive approach we adopted is based on the approach proposed by Lokman et.al [47] to find the most preferred alternative of a decision maker for multi-objective integer programming settings. We adjusted this method to our bi-objective mixed integer programming problem. We begin with providing necessary definitions, assumptions and theorems used throughout the algorithm. Then we will describe the adapted version of the algorithm to our problem in detail. Lastly we exemplify the algorithm by going through steps of an example solution.

All the definitions and theorems are provided below are for bi-objective maximization settings.

Note that in the original model we have two objective functions: one is efficiency maximization and the other one is imbalance minimization. In order to obtain maximization setting in the overall problem, we convert our imbalance term into a balance term. To do so, we subtract imbalance from its nadir value, meaning that $z_2 = z_2^N - z_2$, $z_2 \in \Re_+$. $z^N = \max_{x \in E} \{z_i(x)\}$ $i = 1, 2$ is the nadir point, where E denotes the set of efficient solutions.

Definition 4: Let $z = (z_1, z_2)$ be a solution vector in our proposed bi-objective problem setting, z_1 and z_2 denote efficiency and balance values of the solution, respectively.

Definition 5: $u(z) : \Re^2 \rightarrow \Re$ is a nondecreasing function if $u(z^m) \geq (z^k)$ for every $z^m = (z_1^m, z_2^m)$ and $z^k = (z_1^k, z_2^k)$ such that $z_i^m \geq z_i^k \quad i = 1, 2$.

Definition 6: $u(z) : \Re^2 \to \Re$ is a quasi concave function if $u(\sum_{i=1}^{n} \mu_i z^i) \geq min_{i=1,\dots,n} u(z^i)$ for $\sum_{i=1}^{n} \mu_i = 1, \quad \mu_i \geq 0$.

Definition 7: $\succ$ denotes the binary relation "is preferred to" .

Definition 8: For each pair of alternatives z^k and z^m such that $z^m \succeq z^k$, a 2-point cone is defined as;
$C(z^m, z^k) = \left\{ z : z = z^k + \mu(z^k - z^m), \mu \geq 0 \right\}$. z^m and z^k are called the upper and lower generators of the 2-point cone, respectively.

Definition 9: T is the set representing available preferences of decision maker, $T = \left\{ (z^m, z^k) : z^m \succ z^k \right\}$

Assumption: The fundamental assumption of the algorithm is that decision maker has a non-decreasing quasiconcave utility function, which is unknown.

Theorem 1: Let u be a nondecreasing quasiconcave function $u(z) : \Re^2 \to \Re$. Let $u(z^k) < u(z^m), \quad m \neq k$, where $z^k \in \Re^2$ and $z' \in \Re^2$. Consider;

$$max \quad \epsilon \tag{5.6}$$

$$\mu(z^k - z^m) - \epsilon \mathbf{1} \geq z' - z^k \tag{5.7}$$

$$\mu \geq 0 \tag{5.8}$$

If $\epsilon* \geq 0$; then $u(z^k) \geq u(z')$ [48].

Theorem 1 is first introduced by Korhonen et.al [48]. It implies that if the utility function is quasi concave and some prior knowledge on the preference of decision maker is known, one can infer that some parts of the feasible criteria space would not contain the most preferred solution; such areas are referred as cone dominated areas. This method is called the convex cone method and it allows to narrow down the feasible criteria space by eliminating more cone dominated areas as more preference information is obtained.

Mathematical model described in Theorem 1 checks whether alternative z' is inferior to the cone $z^k + \mu(z^k - z^m)$. If so, theorem implies alternative z' can be at most as preferred as alternative z^k.

Using the results from [48], Lokman et.al [47] introduced the following theorem that defines the characteristic of cone dominated solutions through 2-point cones.

Theorem 2: Let u be a nondecreasing quasiconcave function defined in Euclidean space $\Re^2$. Consider the two distinct points z^m and z^k such that $u(z^k) < u(z^m)$. Then a point z is cone dominated by cone $C(z^m; z^k)$ if and only if the following conditions hold:

1. $z_i \leq z_i^k, \quad i : z_i^k - z_i^m \leq 0$
2. $z_i(z_j^k - z_j^m) + z_j(z_i^m - z_i^k) \leq z_j^k z_i^m - z_i^k z_j^m, \quad i : z_i^k - z_i^m \leq 0, j : z_j^k - z_j^m > 0.$

We define Theorem 2 for bi-objective settings. Hence we provide a simplified version of the proof in Lokman et al [47] for these settings.

Proof: $\Rightarrow$We want to show that given two distinct points z^m and z^k such that $u(z^k) < u(z^m)$ conditions (1) and (2) are satisfied.

Note that, since this is a bi-objective setting if z^k and z^m are two distinct nondominated solutions, then they have to satisfy exactly one out of the following four conditions;

1. $z_1^k - z_1^m < 0 \wedge z_2^k - z_2^m \geq 0$
2. $z_1^k - z_1^m \leq 0 \wedge z_2^k - z_2^m > 0$
3. $z_1^k - z_1^m > 0 \wedge z_2^k - z_2^m \leq 0$
4. $z_1^k - z_1^m \geq 0 \wedge z_2^k - z_2^m < 0$

Without loss of generality assume $z_1^k - z_1^m \leq 0$ and $z_2^k - z_2^m > 0$. If z is cone dominated by $C(z^m; z^k)$ from Theorem 1 we know that $\exists \mu \geq 0$ satisfying;

$$z_i \leq z_i^k + \mu(z_i^k - z_i^m) \qquad i = 1, 2 \tag{5.9}$$

We can rewrite the inequalities in (5.9) separately as follows;

$$\mu(z_1^m - z_1^k) \leq z_1^k - z_1 \tag{5.10}$$

$$z_2 - z_2^k \leq \mu(z_2^k - z_2^m) \tag{5.11}$$

Now, both sides of (5.10) and (5.11) are multiplied by $(z_2^k - z_2^m)$ and $(z_1^m - z_1^k)$, respectively. Following inequalities are obtained;

$$\underline{\mu(z_1^m - z_1^k)(z_2^k - z_2^m)} \leq (z_1^k - z_1)(z_2^k - z_2^m) \tag{5.12}$$

$$(z_2 - z_2^k)(z_1^m - z_1^k) \leq \underline{\mu(z_2^k - z_2^m)(z_1^m - z_1^k)} \tag{5.13}$$

Since we already assumed $0 \leq z_1^m - z_1^k$, $z_2^k - z_2^m \geq 0$ and $\mu \geq 0$, inequality (5.12) implies the following;

$$z_1^k - z_1 \geq 0 \Rightarrow z_1 \leq z_1^k \tag{5.14}$$

(5.14) satisfies condition (1) in Theorem 2.

Now we will show that inequality 2 is also satisfied.

$\mu(z_2^k - z_2^m)(z_1^m - z_1^k)$ is a common term in inequalities (5.12) and (5.13), hence they together imply;

$$(z_2 - z_2^k)(z_1^m - z_1^k) \leq (z_1^k - z_1)(z_2^k - z_2^m) \tag{5.15}$$

By rearranging (5.15) can be written as;

$$z_1(z_2^k - z_2^m) + z_2(z_1^m - z_1^k) \leq z_2^k z_1^m - z_1^k z_2^m. \tag{5.16}$$

(5.16) is exactly same as condition (2) from Theorem 2.

We conclude that for any cone dominated solution both conditions (1) and (2) have to be satisfied.

$\Leftarrow$ We want to show that if a solution z satisfies conditions (1) and (2) then z is a cone dominated solution. To show this, one should prove that $\exists \mu$ for any z

dominated by cone $C(z^m; z^k)$ satisfying the following inequalities;

$$z_1 \leq z_1^k + \mu(z_1^k - z_1^m) \tag{5.17}$$

$$z_2 \leq z_j^k + \mu(z_2^k - z_2^m) \tag{5.18}$$

We need to discuss following two cases separately;
(a) $z_1^k - z_1^m = 0$ and $z_2^k - z_2^m > 0$
(b) $z_1^k - z_1^m < 0$ and $z_2^k - z_2^m > 0$

(a) For the cases where; $z_1^k - z_1^m = 0$ from condition (1) we can directly derive the following inequality

$$z_1 \leq z_1^k + \mu(z_1^k - z_1^m) = z_1^k \tag{5.19}$$

(b) Let us now consider the case when; $z_1^k - z_1^m < 0$ and $z_2^k - z_2^m > 0$.

We can add and subtract $z_1^k z_2^k$ from right hand side of condition (2);

$$z_1(z_2^k - z_2^m) + z_2(z_1^m - z_1^k) \leq z_2^k z_1^m - z_1^k z_2^m + (z_1^k z_2^k - z_1^k z_2^k) \tag{5.20}$$

By rearranging; $\dfrac{z_2 - z_2^k}{z_2^k - z_2^m} \leq \dfrac{z_1^k - z_1}{z_1^m - z_1^k}$.

From our initial assumption, $z_1^k - z_1^m < 0$ and $z_2^k - z_2^m > 0$, we know both denominators are positive; therefore $\exists \mu$ satisfying:

$$\frac{z_2 - z_2^k}{z_2^k - z_2^m} \leq \mu \leq \frac{z_1^k - z_1}{z_1^m - z_1^k} \tag{5.21}$$

μ^* can be set to $\dfrac{(z_1^k - z_1)}{(z_1^m - z_1^k)}$. Since we already assumed that $z_1^m > z_1^k$ and $z_2^k > z_2^m$ we can guarantee existence of such μ^*. We write (5.21) as follows,

$$z_1 \leq z_1^k + \mu^*(z_1^k - z_1^m) \tag{5.22}$$

$$z_2 \leq z_2^k + \mu^*(z_2^k - z_2^m) \tag{5.23}$$

(5.19), (5.22) and (5.23) together imply that z is cone dominated. $\square$

As a result of Theorem 2, any point that is not cone dominated by $C(z^m; z^k)$ has to satisfy at least one of the following conditions;

(i) $\exists \quad i : z_i^k - z_i^m \leq 0 \quad$ satisfying $z_i^k < z_i$.

(ii) $\exists \quad i : z_i^k - z_i^m \leq 0$ and $j : z_j^k - z_j^m < 0$ satisfying
$z_j^k z_i^m - z_i^k z_j^m < z_i(z_j^k - z_j^m) + z_j(z_i^m - z_i^k)$.

Figure 5.4 demonstrates a 2-point cone and the corresponding cone dominated region for our bi-objective problems. Assuming $z^m \succ z^k$ and $z_2^m > z_2^k$, it directly follows that $z_1^m < z_1^k$ (since z^k is nondominated). As shown by [47], any point z that is not cone dominated has to satisfy at least one of the conditions $(i), (ii)$, as otherwise it is in the cone dominated region (shaded).

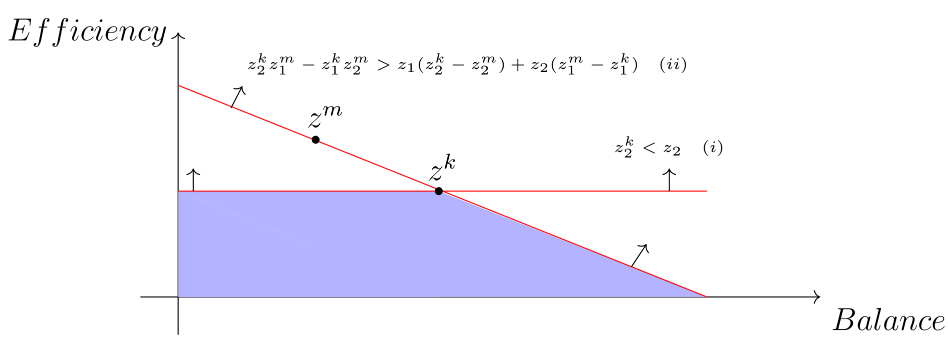

Figure 5.4: Cone dominated region when $z^m \succ z^k$ (shaded)

Following these results Lokman et.al [47] provide a mathematical model so as to obtain new solutions that are not dominated by any cone $C(z^m; z^k)$. Modified version of the problem to our bi-objective problem setting is as follows;

$$P^h \quad max \quad \lambda_1 z_1(x) + \lambda_2 z_2(x)$$

$$(z_i^k + \epsilon) r_i^{m,k} \leq z_i(x) \qquad (z^m, z^k) \in T, \quad i : z_i^k - z_i^m \leq 0 \tag{5.24}$$

$$(z_j^k z_i^m - z_i^k z_j^m + \epsilon) t_{ij}^{m,k} \leq z_i(x)(z_j^k - z_j^m) + z_j(x)(z_i^m - z_i^k)$$
$$(z^m, z^k) \in T, i : z_i^k - z_i^m \leq 0, \quad j : z_j^k - z_j^m > 0 \tag{5.25}$$

$$r_i^{m,k} + t_{ij}^{m,k} = 1, \quad (z^m, z^k) \in T, \quad i : z_i^k - z_i^m \leq 0, \quad j : z_j^k - z_j^m > 0 \tag{5.26}$$

$$(z_i^{inc} + \epsilon) y_i \leq z_i(x), \quad i = 1, 2 \tag{5.27}$$

$$y_1 + y_2 = 1 \tag{5.28}$$

$$t_{ij}^{m,k}, r_i^{m,k}, \in \{0, 1\}, \quad (z^m, z^k) \in T, \quad i : z_i^k - z_i^m \leq 0, \quad j : z_j^k - z_j^m > 0 \tag{5.29}$$

$$y_i \in \{0, 1\}, \quad i = 1, 2 \tag{5.30}$$

$$x \in X \tag{5.31}$$

where z^{inc} is representing the current incumbent which is the most preferred solution found so far, λ_1 and $\lambda_2 > 0$ are arbitrary weights and ϵ is a small positive constant. If the problem P^h is feasible, we refer to its optimal solution as z^h.

Constraints (5.24) and (5.25) guarantee that the obtained solution is not cone dominated by checking conditions (i) and (ii). When $r_i^{m,k} = 1$, for $(z^m, z^k) \in T, \quad i : z_i^k - z_i^m \leq 0$ constraint (5.24) becomes $z_i^k + \epsilon \leq z_i$ meaning that condition (i) is satisfied. In the same way, if $t_{ij}^{m,k} = 1, \quad (z^m, z^k) \in T, i : z_i^k - z_i^m \leq 0, \quad j : z_j^k - z_j^m > 0$ then constraint (5.25) implies $(z_j^k z_i^m - z_i^k z_j^m + \epsilon) < z_i(z_j^k - z_j^m) + z_j(z_i^m - z_i^k)$ so that condition (ii) will be satisfied. Constraint (5.26) is used to ensure that one of these conditions (i) or (ii) to be satisfied. At the beginning of the algorithm there is no preference information yet therefore no cones can be generated to obtain a distinct nondominated solution. (5.27) and (5.28) assert that z^h will have a strictly better objective function value than z^{inc} in at least one of the objectives; meaning that z^h is a distinct nondominated point.

The model creates 2 binary decision variables ($t_{ij}^{m,k}$ and $r_i^{m,k}$) and 3 constraints for each 2- point cone (5.24, 5.25 and 5.26), however the number of cones that needs to be generated throughout the algorithm can not be determined a priori.

Now we will discuss the adopted version of the algorithm to our problem setting. The algorithm begins with finding an initial incumbent by using a weighted sum scalarization model. Then a new nondominated solution is obtained by solving model P^h with different weight values. Since there is no preference information yet, $T = \emptyset$ and P^h only consists of constraints (5.27), (5.28), (5.30) and (5.31). This new solution, z^h is a challenger to the incumbent solution. To obtain preference information, decision maker is asked to make a pairwise comparison between z^h and z^{inc}. Depending on the choices of decision maker, preference information set T and z^{inc} are updated then model P^h is resolved with a new weight vector by eliminating all cone dominated regions. These steps are repeated until the most preferred solution is the only feasible solution to the problem. A pseudocode of the algorithm is provided in Algorithm 2.

Algorithm 2 Interactive Algorithm

Initialize: $T = \emptyset$, $f = 1$ and $h = 1$. Arbitrarily select weights $\lambda_i > 0$, $i = 1, 2$, ($\sum_{i=1}^{2} \lambda_i = 1$). Solve the initial model. (T is a set representing available preferences of DM)

$$\begin{aligned} \max \quad & \lambda_1 z_1(x) + \lambda_2 z_2(x) \\ & x \in X \end{aligned}$$

if the problem is infeasible **then**

  Stop

else

  $z^1 = (z_1^1, z_2^1)$ is the initial incumbent, $z^{inc} = z^1$

end if

while $f = 1$ **do**

  Set $h = h + 1$. Select new weights λ_i, by solving model P^w, and solve the model with additional cone domination constraints defined over set T

  if the problem is infeasible **then**

    Set $f = 0$. The incumbent point, $z^{inc} = (z_1^{inc}, z_2^{inc})$, is the most preferred solution of the DM.

  else

    Denote optimal solution as challenger, $z^h = (z_1^h, z_2^h)$.

    Compare the incumbent z^{inc} and the challenger z^h

    if $z^{inc} \succ z^h$ **then**

      $T = T \cup (z^{inc}, z^h)$

    else

      $T = T \cup (\cup_{k=1}^{h-1} z^{inc}, z^h)$

      Update incumbent, $z^{inc} = z^h$

    end if

  end if

end while

Assuming there are finite number of nondominated solutions the algorithm terminates in finite number of iterations. In our problem there is certain amount of projects, N, and we make 0-1 selection decisions therefore total number of possible allocation decision is bounded. In fact $\sum_{t=1}^{N} \binom{N}{t}$ is an upper bound on the maximum number of solutions that can be obtained. Hence we can also conclude that proposed algorithm will reach the most preferred alternative in finite number of iterations.

In our computational experiments initial incumbent weight vector is selected as (0.5,0.5). For each iteration of the algorithm model, P^h is solved with new weight vectors. To select weight vectors that will help us to find highly preferred distinct nondominated solutions, following mathematical model P^w is solved;

$$P^w \qquad \max \epsilon \tag{5.32}$$

$$\frac{\sum_{i=1}^{2}(z_i^m - z_i^k)\lambda_i}{\| z^m - z^k \|} \geq \sigma_{m,k}\epsilon, \quad (z^m, z^k) \in T \tag{5.33}$$

$$\lambda_1 + \lambda_2 = 1 \tag{5.34}$$

$$\lambda_i \geq \sigma_i \cdot \epsilon, \quad i = 1, 2 \tag{5.35}$$

$$\epsilon \geq 0 \tag{5.36}$$

$\sigma_{m,k}$ and σ_i are random parameters which are uniformly distributed between [0,1] and they allow us to find distinct weights from different parts of the feasible weight space.

5.3.1 Interactive Approach Example

Now we will exemplify the algorithm with an example demonstrated step by step with Figures 5.5-5.8. The example is provided for an instance from DRDM, Type C $N = 15$ and $K = 4$. When the instance is solved with epsilon constraint method, it is observed that there exists 132 nondominated solutions. These solutions are represented with the circles on the graphs. Solutions obtained throughout the algorithm are shown with triangles. At each iteration newly added cone constraints are drawn in red and previous ones are black.

Initially an incumbent solution is found by solving the weighted sum scalarization model with the initial weight vector, this solution is referred as $z^1 = z^{inc}$. Then by solving P^h, challenger z^2 is obtained. Recall that since this is the first iteration, set T is empty and there are no cone constraints in the model. It only aims to find a nondominated solution different from z^{inc}. z^2 is marked as the first challenger point. Later, decision maker is asked to make a pairwise comparison between z^{inc} and z^2 and she prefers z^2 over z^{inc}, $z^2 \succ z^{inc}$. This preference information is added to set T and cone dominated region is defined accordingly. Figure 5.5 demonstrates the cone dominated region (shaded) generated by z^2 and z^{inc}. All the solutions interior to this region are eliminated from further search. Incumbent solution is updated, $z^{inc} = z^2$ and algorithm moves to the second iteration.

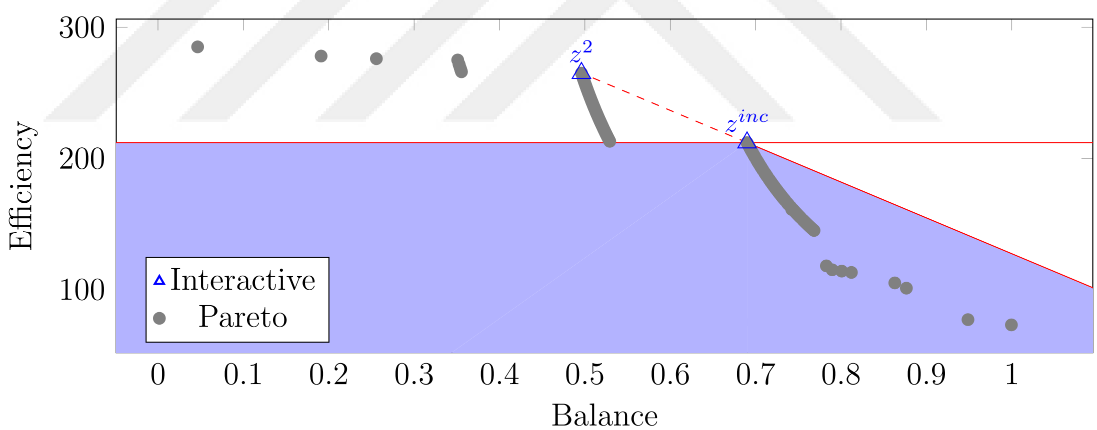

Figure 5.5: Interactive-DRDM Type C $N = 15$, Iteration 1

New weight vectors are obtained by solving the mathematical model defined with (5.32)-(5.36). P^h is solved with the information obtained in iteration 1 and a new challenger solution is found, z^3. This time, assume that decision maker prefers z^3 over z^{inc} (z^2). Since we already know $z^2 \succ z^1$, by transitivity we conclude that $z^3 \succ z^1$. As described in the algorithm both preference information $z^3 \succ z^{inc}$ (z^2) and $z^3 \succ z^1$ are added to set T. Figure 5.6 shows the new cone constraints and the overall cone dominated region. Incumbent solution is updated, $z^{inc} = z^3$ and algorithm proceeds with the iteration 3.

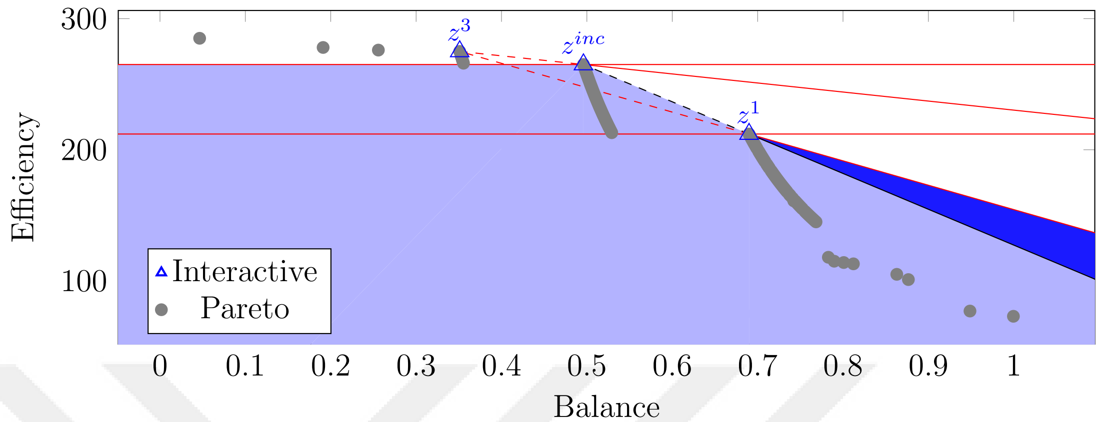

Figure 5.6: Interactive-DRDM Type C $N = 15$, Iteration 2

Model P^h is solved again with the additional cone constraints and a new weight vector. z^4 is the new challenger and decision maker prefers $z^{inc} \succ z^4$. A new cone is generated as shown in Figure 5.7. Incumbent does not change.

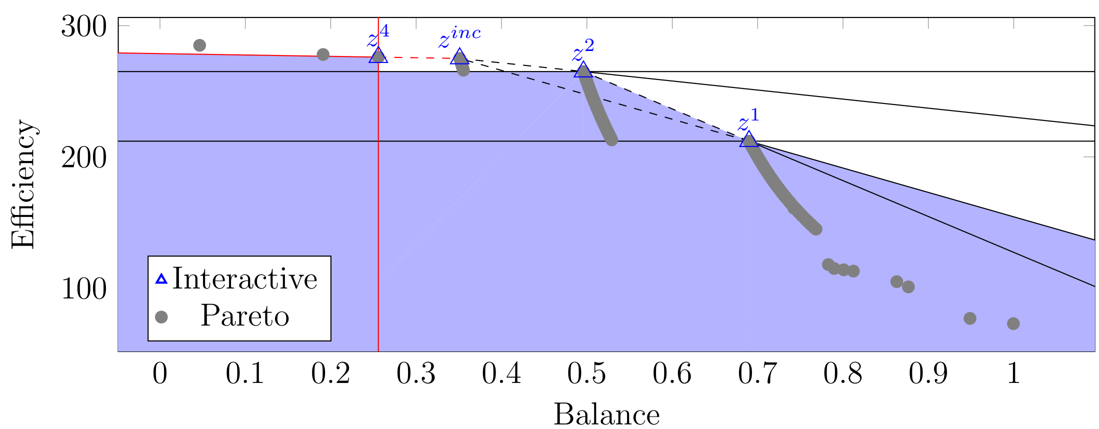

Figure 5.7: Interactive-DRDM Type C $N = 15$, Iteration 3

As seen on Figure 5.8, z^5 is returned to the decision maker for comparison as it is the only solution left exterior to any cone and she picks z^{inc} over z^5. After adding new cone constraints P^h fails to find a new nondominated solution since there are no solutions left. Algorithm terminates and $z^{inc} = z^3$ is the most preferred solution.

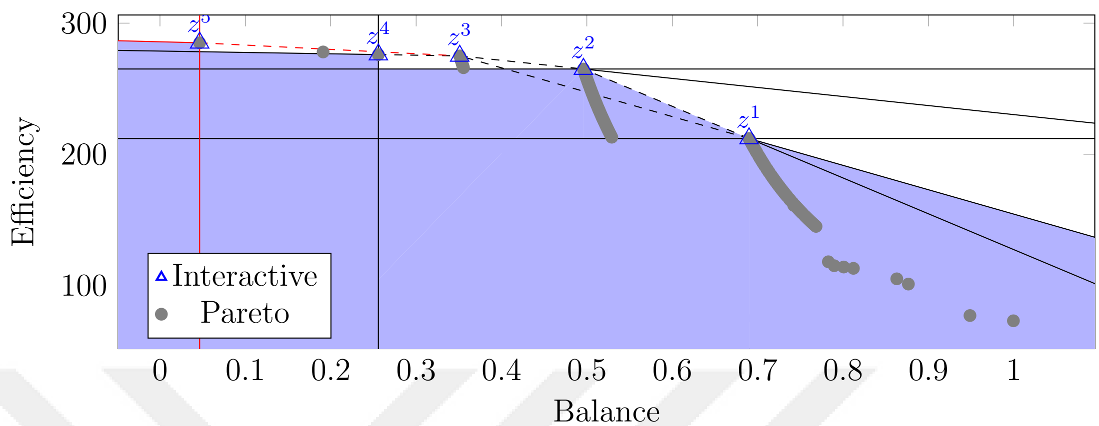

Figure 5.8: Interactive-DRDM Type C $N = 15$, Iteration 4

Interactive algorithm finds the most preferred solution among 132 solutions only by asking for 4 pairwise comparison questions to decision maker while solving 5 models. Of course, such an interactive algorithm would require the decision maker to be available to answer the preference questions. Further analysis on the performance of the algorithm will be made in Chapter 6.

Chapter 6

Computational Experiments

This chapter is dedicated to computational experiments that we performed to demonstrate the feasibility of proposed approach. To begin with, data generation and assumptions used in different types of problem instances are explained. Afterwards computational results obtained with the solution methods described in Chapter 5 are demonstrated and corresponding results are analysed in detail.

6.1 Data Generation and Assumptions

In different decision making environments, parameters may have certain characteristics; for instance output and input values may be correlated or productivity of projects may differ depending on the category of each project. In order to mimic different types of project portfolio selection problems that could be encountered in real life, we used 3 different problem types.

In all three problem types, cost values are generated randomly with Java random seed as follows:

$$c_i = rand + a \quad rand \in [0, 30] \quad i \in N \quad a \in \mathbb{Z}^+$$

Benefit values are generated with a different policy in each type. In type A instances, benefit is generated randomly, while in types B and C, it is generated using a linear function of cost. Problem types are described below.

Type A: Benefit values are generated randomly as follows:

$$b_i = rand + a \quad rand \in [0, 30] \quad i \in N \quad a \in \mathbb{Z}^+$$

In this type, there is no dependency between cost and benefit values. We use the constant value a to avoid having 0 benefit. In our case study we take $a = 5$. Note that this policy implies that productivity of the projects are also randomly determined.

Type B: In this type, benefit values are positively correlated with the cost values.

$$b_i = c_i \cdot t + a' \quad t \in \mathbb{Z}^+ \quad i \in N \quad a' \in \mathbb{Z}^+$$

The coefficient of the linear function t determines how productive a project is. $a' \in \mathbb{Z}^+$ is used to have some diversity among projects. In our case we took a' as a random integer taking values between 0 and 10 and t as 1.2.

Type C: In this type we assume that each category has a different level of productivity, hence use t_k instead of t. In our case study we took $t_k = \{1.2, 1.5, 1.8\}$ where $K = 3$ and $t_k = \{1.2, 1.5, 1.8, 2.0\}$ where $K = 4$.

for $k \leftarrow K$ **do**
  $b_i = c_i \cdot t_k + a' \quad i \in N \quad a' \in \mathbb{Z}^+$
end for

In all of the generated instances budget, B, is set to $\frac{\sum_{i \in N} c_i}{2}$. We generated problem instances with 3 and 4 categories and assumed there are 5 intervals. The threshold levels defining intervals are determined as a function of $TB = \sum_{i \in N} b_i$ (The total benefit that would have been gained if initiating all projects were possible). Used α_m and T values are provided below;

Table 6.1: Reference α_{mk} and corresponding T values

Interval(m)	α_m		T_m
	$K=3$	$K=4$	
1	(0.33, 0.33, 0.33)	(0.25,0.25,0.25,0.25)	0-TB*0.4
2	(0.30, 0.30, 0.40)	(0.20,0.21,0.26,0.33)	TB*0.4-TB*0.5
3	(0.20, 0.25, 0.55)	(0.15,0.18,0.27,0.40)	TB*0.5-TB*0.6
4	(0.15, 0.20, 0.65)	(0.12,0.15,0.30,0.43)	TB*0.6-TB*0.75
5	(0.10, 0.15, 0.75)	(0.10,0.12,0.33,0.45)	TB*0.75-TB

For each type we generated instances with $N = 30$ and $N = 50$. For each combination of these parameters we generated 10 instances.

Computational studies are performed on Eclipse Java Oxygen and solved using CPLEX 12.6. All of the instances are run in a computer with Intel Xeon CPU E5-1650 3.6 GHz processor and 32 GB RAM. Computation times are given in central processing unit seconds (CPU).

6.2 Epsilon-Constraint Algorithm Results

Initially we employ the epsilon constraint method explained in Chapter 5 to obtain the exact Pareto front of our problem. We run the algorithm for 10 instances for each problem type and report the average and maximum values of the solution time and number of exact nondominated solutions obtained, $|NS|$, for the DRDM and MRDM models, respectively.

Results of the DRDM are provided in Table 6.2. We used a time limit of 3 hours as follows: If the instance cannot be solved within 3 hours, that is, if the whole set of Pareto solutions cannot be returned, we terminate that instance and its values are not included in the performance metrics. Number of instances that can not be completed within the time limit is reported under the column "NF".

It is seen in Table 6.2 that Type A problems have the highest number of nondominated solutions and hence the largest solution time. In the cases where $N = 30$, increasing the number of categories from 3 to 4 does not create a significant difference in $|NS|$. Although solution times increase, they are still under 30 minutes on average. When $N = 50$, solution times increase significantly; most of the instances with $N = 50$, $K = 4$ cannot be solved within the time limit.

Table 6.2: Results of discrete reference distribution model

N	K	Type	CPU Times		$\|NS\|$		NF
			mean	max	mean	max	
30	3	A	37.8	70.0	71	128	
		B	29.2	55	20.5	41	
		C	22.3	46	27.2	58	
	4	A	102.3	210	69.9	146	
		B	71.7	159	36.5	110	
		C	111.5	257	52.7	122	
50	3	A	1565.6	5327	99.125	158	2
		B	178.5	752	20.8	41	2
		C	169.2	802	36.8	50	1
	4	A	1515.5	2261	184.5	240	8
		B	186.0	186	14.0	14	9
		C	717.6	1728	29.0	91	7

Table 6.3 shows the solution times of MRDM with the same data sets. Due to the additional constraints and binary variables, MRDM's solution times are significantly higher than DRDM. $|NS|$ values also tend to increase in this approach. Note that larger instances could not be solved within the time limit, hence we do not provide results for these cases.

Table 6.3: Results of moving reference distribution model

N	K	Type	CPU Times		$\|NS\|$	
			mean	max	mean	max
30	3	A	2275.9	5577	87.9	136
		B	2410.8	6901	34.7	84
		C	3692.8	10168	28.9	85
	4	A	505.5	2059	78.8	172
		B	599.7	3275	49.3	143
		C	1320.2	5260	65.8	150

Figure 6.1 shows the nondominated solutions found by the DRDM and MRDM approaches for a small-size Type A instance. The frontiers have a similar pattern; while MRDM has more solutions and its efficiency axis range starts from smaller values in general.

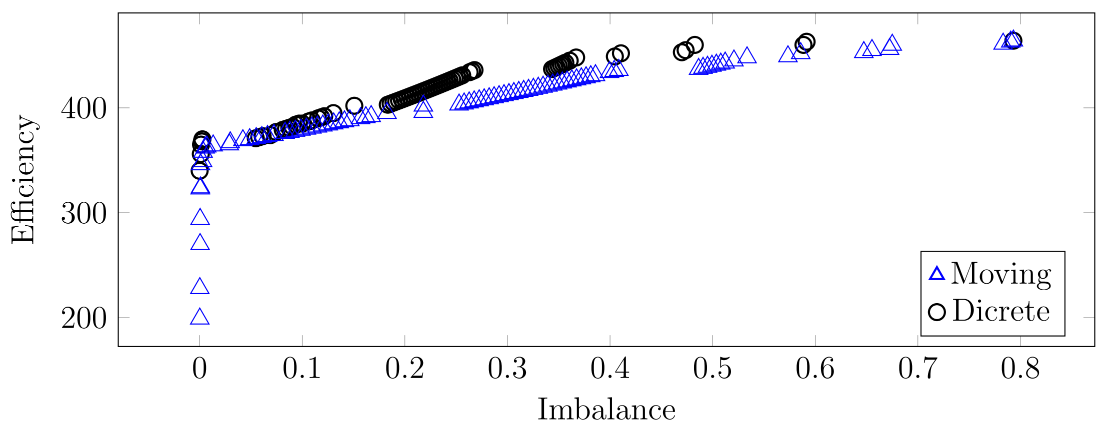

Figure 6.1: DRDM and MRDM solutions Type A results, N=30 K=3

6.3 Heuristic Algorithm Results

We conducted experiments by solving the problem instances with three meta-heuristic algorithms, SPEA2, NSGA-II and MOCell, and analysed quality of their solutions with the performance metrics introduced in Chapter 5.

We begin with discussing performance of metaheuristics in smaller sized problem instances, $N = 30$, where we can compare the results of metaheuristics with the true Pareto obtained with epsilon constraint method.

The results of the experiments are seen in Tables 6.4 and Table 6.5 for DRDM and MRDM approaches, respectively. For each algorithm, average and worst case values for P, D1 and D2 metrics are reported. The results of these metrics show that algorithms performed similarly in problem instances where $K = 3$. When the number of categories is increased, $K = 4$, it is clear that algorithms SPEA2 and MOCell have very close performance and they are better than NSGA-II.

Table 6.4: DRDM Heuristic results

				P		D1		D2	
DRDM			Type	mean	worst	mean	worst	mean	worst
N	K	SPEA2	A	0.846	0.384	0.000	0.001	0.016	0.076
30	3		B	0.674	0.000	0.013	0.067	0.068	0.333
			C	0.610	0.000	0.004	0.016	0.029	0.101
		NSGA-II	A	0.850	0.411	0.000	0.001	0.012	0.046
			B	0.605	0.000	0.011	0.04	0.051	0.116
			C	0.596	0.000	0.004	0.009	0.03	0.105
		MOCell	A	0.856	0.384	0.000	0.001	0.004	0.008
			B	0.784	0.000	0.003	0.008	0.023	0.057
			C	0.671	0.000	0.002	0.007	0.020	0.062
30	4	SPEA2	A	0.677	0.182	0.003	0.009	0.023	0.042
			B	0.556	0.033	0.004	0.013	0.025	0.041
			C	0.572	0.000	0.004	0.017	0.020	0.036
		NSGA-II	A	0.285	0.030	0.015	0.033	0.068	0.117
			B	0.144	0.000	0.038	0.181	0.096	0.348
			C	0.142	0.000	0.016	0.065	0.048	0.157
		MOCell	A	0.730	0.182	0.004	0.020	0.048	0.218
			B	0.560	0.033	0.004	0.013	0.020	0.034
			C	0.602	0.000	0.003	0.006	0.017	0.030

Table 6.5: MRDM Heuristic results

				P		D1		D2	
MRDM			Type	mean	worst	mean	worst	mean	worst
N	K	SPEA2	A	0.802	0.400	0.000	0.001	0.004	0.007
30	3		B	0.575	0.033	0.002	0.009	0.011	0.059
			C	0.576	0.000	0.002	0.007	0.008	0.017
		NSGA-II	A	0.793	0.400	0.000	0.002	0.007	0.016
			B	0.481	0.000	0.003	0.010	0.014	0.059
			C	0.548	0.000	0.002	0.005	0.009	0.012
		MOCELL	A	0.809	0.422	0.000	0.001	0.005	0.007
			B	0.260	0.000	0.001	0.004	0.004	0.023
			C	0.622	0.000	0.001	0.001	0.005	0.009
30	4	SPEA2	A	0.710	0.000	0.006	0.040	0.035	0.159
			B	0.697	0.000	0.001	0.040	0.015	0.159
			C	0.617	0.013	0.003	0.0146	0.022	0.085
		NSGAII	A	0.270	0.000	0.014	0.045	0.062	0.172
			B	0.185	0.000	0.011	0.045	0.037	0.172
			C	0.170	0.000	0.014	0.038	0.049	0.101
		MOCell	A	0.709	0.000	0.007	0.050	0.040	0.216
			B	0.669	0.000	0.001	0.050	0.014	0.216
			C	0.512	0.053	0.004	0.013	0.023	0.079

In Figure 6.2 exact Pareto solutions and the solutions found by SPEA2 of an instance are illustrated to give an idea about how close these two sets are. An instance with close to average results for the three metrics is chosen. SPEA2 succeeded in finding 77% of the exact nondominated solutions. Most of the solutions that could not be found are from the part of the Pareto front with better imbalance values. Solutions found by SPEA2 that has the best imbalance values are dominated by the exact Pareto, since exact solutions are able to achieve the same (or better) imbalance scores while obtaining more benefit. For the other regions of the graph, it is observed that the heuristic solutions are close to the exact Pareto set, as implied by the small D1 and D2 values.

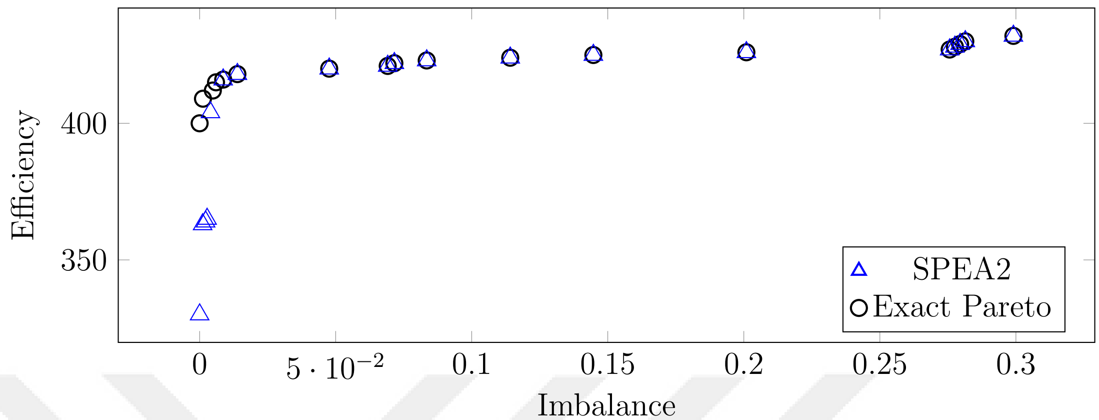

Figure 6.2: DRDM exact Pareto frontier and SPEA2 results, Type B $N = 30$, $K = 3$ P=0.77 D1=0.003 D2=0.024

Now we will discuss the performance of metaheuristics for the instances where exact Pareto front could not be obtained within the time limit. The comparison is made among the metaheuristic results based on their Spacing, Maximum Spread and Set Coverage values.

Table 6.6: DRDM-Heuristic Comparisons

DRDM							
		Type A		Type B		Type C	
		mean	worst case	mean	worst case	mean	worst case
SPEA2	S	1.225	4.621	3.980	14.420	3.593	24.258
	MS	82.965	33.271	27.414	13.453	48.881	20.952
NSGAII	S	10.115	54.607	2.491	12.192	8.098	37.026
	MS	62.481	26.248	27.474	11.269	60.538	29.155
MOCell	S	3.434	10.392	4.559	32.106	1.638	6.375
	MS	88.564	48.114	27.087	15.199	51.994	23.36

Table 6.7: MRDM-Heuristic Comparisons

MRDM							
		Type A		Type B		Type C	
		mean	worst case	mean	worst case	mean	worst case
SPEA2	S	2.034	7.482	1.954	6.757	3.151	11.539
	MS	102.823	69.627	39.378	14.142	56.404	20.000
NSGAII	S	5.338	10.529	3.230	10.572	6.378	28.143
	MS	68.437	23.643	41.002	11.832	64.871	22.803
MOCell	S	1.5786	8.037	1.885	6.530	1.8726	5.630
	MS	95.747	63.624	40.612	16.733	52.904	21.118

In Tables 6.6 and 6.7 spacing and maximum spread metrics of these algorithms are provided. On average MOCell seems to be performing better than other two algorithms in terms of spacing. Especially in Type C problems MOCell is notably better than NSGA-II. In terms of maximum spread, SPEA2 has better performance in Type A problems but in Types B and C no significant difference is found.

For set coverage, as mentioned in Chapter 5, we performed two analyses: Table 6.8 shows the results in which mutual elements in two sets are considered and results in Table 6.9 are calculated with only using strong dominance between solutions. According to both analyses SPEA2 performs strictly better than the other two heuristics. MOCell seems to be better than NSGA-II in general with a few exceptions.

Table 6.8: Set coverage results-weak dominance

	DRDM			MRDM		
	Type A	Type B	TypeC	Type A	Type B	TypeC
SC(SPEA2,NSGAII)	0.879	0.844	0.852	0.800	0.848	0.792
SC(NSGAII,SPEA2)	0.381	0.657	0.732	0.336	0.716	0.684
SC(SPEA2,MOCell)	0.829	0.775	0.867	0.813	0.811	0.775
SC(MOCell,SPEA2)	0.825	0.642	0.711	0.768	0.739	0.671
SC(NSGAII,MOCell)	0.387	0.663	0.754	0.376	0.692	0.732
SC(MOCell,NSGAII)	0.891	0.686	0.777	0.851	0.765	0.636

Table 6.9: Set coverage results-strong dominance

			DRDM			MRDM
	Type A	Type B	TypeC	Type A	Type B	TypeC
SC'(SPEA2,NSGA-II)	0.086	0.293	0.205	0.089	0.256	0.263
SC'(NSGA-II,SPEA2)	0.001	0.128	0.068	0.004	0.143	0.127
SC'(SPEA2,MOCell)	0.049	0.306	0.278	0.029	0.243	0.224
SC'(MOCell,SPEA2)	0.016	0.126	0.079	0.010	0.149	0.132
SC'(NSGA-II,MOCell)	0.045	0.307	0.205	0.004	0.201	0.210
SC'(MOCell,NSGA-II)	0.108	0.287	0.179	0.085	0.247	0.185

Tables 6.10 and 6.11 show the hypervolume values for SPEA2, NSGAII and MOCell heuristics. V_1, V_2 and V_3 are areas of the spaces dominated by the objective vectors obtained by SPEA2, NSGAII and MOCell, respectively. $V_i \setminus V_j$ is the area of the space which is dominated by algorithm i's objective vectors but not by j's objective vectors $i, j = 1, 2, 3 \quad i \neq j$. It is seen that both SPEA2 and MOCell have very similar performance in terms of hypervolume while they are both better than NSGA-II in general.

Table 6.10: Hypervolume comparison for DRDM

DRDM	V_1	V_2	V_3	$V_1 \setminus V_2$	$V_2 \setminus V_1$	$V_1 \setminus V_3$	$V_3 \setminus V_1$	$V_2 \setminus V_3$	$V_3 \setminus V_2$
Type A	0.889	0.882	0.889	0.007	0.001	0.001	0.001	0.000	0.007
Type B	0.927	0.921	0.922	0.006	0.000	0.001	0.001	0.000	0.007
Type C	0.789	0.778	0.79	0.011	0.001	0.000	0.001	0.000	0.021

Table 6.11: Hypervolume comparison for MRDM

MRDM	V_1	V_2	V_3	$V_1 \setminus V_2$	$V_2 \setminus V_1$	$V_1 \setminus V_3$	$V_3 \setminus V_1$	$V_2 \setminus V_3$	$V_3 \setminus V_2$
Type A	0.886	0.873	0.877	0.013	0.000	0.009	0.000	0.004	0.007
Type B	0.930	0.929	0.931	0.002	0.000	0.001	0.001	0.000	0.002
Type C	0.798	0.798	0.798	0.000	0.001	0.001	0.001	0.001	0.001

6.4 Interactive Approach Results

In the epsilon constraint method, we have to solve 2 models to find each nondominated solution and we cannot solve many instances due to the high number of iterations required and rapidly increasing solution times. One way of handling this challenge is using metaheuristic algorithms that return a set of approximate Pareto solutions. Such algorithms are computationally very efficient, however they do not guarantee exact solutions. We discuss three such algorithms in Chapter 5. As an alternative, in Chapter 5 we proposed an interactive algorithm to find the most preferred solution of the decision maker without finding all nondominated solutions. Now we will present performance of this approach

During computational experiments, at each iteration of the algorithm we need some preference information from decision maker. In order to represent decision makers underlying utility function we used the following quadratic function: $\max \sum_{i=1}^{2} -w_i^2(z_i - z_i^I)^2$ where $w = (0.3, 0.7)$ and z^I denotes the ideal point $z_i^I = max_{x \in X} \{z_i(x)\}$, $i = 1, 2$.

The results of these experiments are summarized in Table 6.12 and 6.13. Instead of $|NS|$ we report the maximum and average number of pairwise comparisons made to achieve the most preferred alternative. Compared to the epsilon constraint approach (see Tables 6.2 and 6.3), this approach solves the problems in considerably less CPU time and number of iterations. Especially when N gets higher, greater number of instances can be solved and improvement in solution time is more significant.

Table 6.12: Performance of Interactive Approach on DRDM

N	K	Type	CPU Times		Comparison		NF
			mean	max	mean	max	
30	3	A	28.4	212	48.4	111	
		B	8.1	43	9.4	32	
		C	9.1	66	15.8	53	
	4	A	22.3	100	36.8	118	
		B	10.1	51	10.3	21	
		C	38.3	18.74	332.2	46	
50	3	A	146.2	464	61.8	97	3
		B	3.8	8	9.1	18	1
		C	18.1	104	14.0	18	1
	4	A	1944.1	8719	82.3	119	4
		B	208.7	1603	9	17	2
		C	249.6	1329	29.7	54	1

Table 6.13: Performance of Interactive Approach on MRDM

N	K	Type	CPU Times		Comparison	
			mean	max	mean	max
30	3	A	10.8	53	28.0	102
		B	30.3	271	271.0	68
		C	18.4	166	13.5	72
	4	A	533.2	5070	50.6	123
		B	591.8	5730	22.7	97
		C	224.1	2144	26.7	115

Chapter 7

Conclusion

In this thesis we consider resource allocation problems with fairness concern, in which an indivisible resource is allocated across multiple entities so that they will enjoy benefits. Resource allocation is a well known problem, which is faced by authorities in many real life decision making environments such as R&D project portfolio selection, healthcare and air traffic management. Decision maker can prefer different allocation policies depending on the problem setting, needs of entities and resource types. One commonly used approach is pure efficiency (total benefit) maximizing approach, which is entirely focused on the total amount of output received at the end. However efficiency maximization is not always necessarily the only objective in the allocation problems.

Our study is motivated by the fact that a pure efficiency maximizing approach may not be appropriate in terms of balance. In certain settings, balance can be considered as crucial as efficiency to both entities and decision maker. Especially when social welfare is the main scope of the allocation problem, it is expected to have balance in the distribution of benefits. In this thesis, we suggest a bi-objective optimization framework that trades balance off against efficiency by measuring how balanced a given distribution is based on reference proportions provided by decision maker.

Initially, we introduced a changing reference approach that enables decision maker to create different reference proportions for different levels of total output. We define imbalance as the deviation of actual benefit allocation from its reference distribution. As a novel extension of the current works in the literature, the proposed method allows different (moving) reference distributions to be used for different budget (benefit) allocations. For instance, when the total allocated amount is small, the decision maker may want the distribution to be equal among categories and as the total amount increases she may use a different reference allocation, which gives priority to some entities over the others.

We exemplify the proposed method on project portfolio selection problems, where the projects belong to different categories and there is a concern for ensuring a balanced allocation of the benefit as well as maximizing total benefit. We solve the resulting bi-objective optimization problems (with one efficiency and one imbalance objective) using the epsilon constraint method and generate Pareto solutions. For larger problems, which can not be solved in reasonable time, we use multi-objective metaheuristic approaches to find approximate Pareto solutions and report on their empirical performance.

As an alternative solution method, we implemented an interactive algorithm that finds the most preferred solution of a DM. Rather than presenting all nondominated solutions, this approach determines the most preferred solution and enables us to solve more instances in less time and number of iterations.

In addition to the solution methods we presented in Chapter 5, we have also tried a Tabu Search algorithm to generate approximate Pareto solutions to our bi-objective problem. However its performance was considerably worse than the metaheuristic algorithms. Hence its details and performance are not reported in this thesis.

We believe that considering balance concerns explicitly in the decision support systems will provide useful insights to the decision makers on the alternative allocation options and the trade-off between efficiency and balance concerns. Moreover we suggested different solution approaches to solve this problem. Among these solution methods decision maker can choose the one that suits the best to her particular decision environment.

Further research can be performed on the multi objective version of the problem, in which relationship among different balance and efficiency measures can be examined. The problem can be extended to a multiple resource and multiple output setting, which will result in more challenging models. Hence, efficient exact solution methods or customized heuristic algorithms that exploit problem structure can be designed.

Bibliography

[1] H. Luss, *Equitable Resource Allocation: Models, Algorithms and Applications*, vol. 101. John Wiley & Sons, 2012.

[2] H. P. Young, *Equity: in theory and practice.* Princeton University Press, 1995.

[3] B. Armbruster and E. Delage, "Decision making under uncertainty when preference information is incomplete," *Management Science*, vol. 61, no. 1, pp. 111–128, 2015.

[4] L. Eeckhoudt and H. Schlesinger, "Putting risk in its proper place," *The American Economic Review*, vol. 96, no. 1, pp. 280–289, 2006.

[5] H. Kellerer, U. Pferschy, and D. Pisinger, "Introduction to np-completeness of knapsack problems," in *Knapsack problems*, pp. 483–493, Springer, 2004.

[6] Ö. Karsu and A. Morton, "Inequity averse optimization in operational research," *European journal of operational research*, vol. 245, no. 2, pp. 343–359, 2015.

[7] S. Iamratanakul, P. Patanakul, and D. Milosevic, "Project portfolio selection: From past to present," in *Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on*, pp. 287–292, IEEE, 2008.

[8] D. Bertsimas and S. Gupta, "Fairness and collaboration in network air traffic flow management: an optimization approach," *Transportation Science*, vol. 50, no. 1, pp. 57–76, 2015.

[9] H. D. Sherali, J. M. Hill, M. V. McCrea, and A. A. Trani, "Integrating slot exchange, safety, capacity, and equity mechanisms within an airspace flow program," *Transportation Science*, vol. 45, no. 2, pp. 271–284, 2011.

[10] C. Barnhart, D. Bertsimas, C. Caramanis, and D. Fearing, "Equitable and efficient coordination in traffic flow management," *Transportation science*, vol. 46, no. 2, pp. 262–280, 2012.

[11] I. S. Orgut, J. Ivy, R. Uzsoy, and J. R. Wilson, "Modeling for the equitable and effective distribution of donated food under capacity constraints," *IIE Transactions*, vol. 48, no. 3, pp. 252–266, 2016.

[12] D. J. Nair, D. Rey, and V. V. Dixit, "Fair allocation and cost-effective routing models for food rescue and redistribution," *IISE Transactions*, vol. 49, no. 12, pp. 1172–1188, 2017.

[13] M. Yang, T. T. Allen, M. J. Fry, and W. D. Kelton, "The call for equity: simulation optimization models to minimize the range of waiting times," *IIE Transactions*, vol. 45, no. 7, pp. 781–795, 2013.

[14] R. Batta, M. Lejeune, and S. Prasad, "Public facility location using dispersion, population, and equity criteria," *European Journal of Operational Research*, vol. 234, no. 3, pp. 819–829, 2014.

[15] J. Yang, J. Xie, X. Deng, and H. Xiong, "Cooperative advertising in a distribution channel with fairness concerns," *European Journal of Operational Research*, vol. 227, no. 2, pp. 401–407, 2013.

[16] W. Ogryczak, "Inequality measures and equitable approaches to location problems," *European Journal of Operational Research*, vol. 122, no. 2, pp. 374–391, 2000.

[17] W. Ogryczak, "Inequality measures and equitable locations," *Annals of Operations Research*, vol. 167, no. 1, pp. 61–86, 2009.

[18] M. Barbati and C. Piccolo, "Equality measures properties for location problems," *Optimization Letters*, vol. 10, no. 5, pp. 903–920, 2016.

[19] J. N. Hooker, “Optimality conditions for distributive justice,” *International Transactions in Operational Research*, vol. 17, no. 4, pp. 485–505, 2010.

[20] Ö. Karsu and A. Morton, “Incorporating balance concerns in resource allocation decisions: a bi-criteria modelling approach,” *Omega*, vol. 44, pp. 70–82, 2014.

[21] T. J. Stewart, “Multiple objective project portfolio selection based on reference points,” *Journal of Business Economics*, vol. 86, no. 1-2, pp. 23–33, 2016.

[22] A. F. Carazo, “Multi-criteria project portfolio selection,” in *Handbook on Project Management and Scheduling Vol. 2*, pp. 709–728, Springer, 2015.

[23] B. Häckel, V. Isakovic, and F. Moser, “Integrated long-and short-term valuation of it innovation investments,” *Electronic Markets*, vol. 25, no. 1, pp. 73–85, 2015.

[24] J. W. Lee and S. H. Kim, “Using analytic network process and goal programming for interdependent information system project selection,” *Computers & Operations Research*, vol. 27, no. 4, pp. 367–382, 2000.

[25] J. W. Lee and S. H. Kim, “An integrated approach for interdependent information system project selection,” *International Journal of Project Management*, vol. 19, no. 2, pp. 111–118, 2001.

[26] M. Shakhsi-Niaei, S. A. Torabi, and S. H. Iranmanesh, “A comprehensive framework for project selection problem under uncertainty and real-world constraints,” *Computers & Industrial Engineering*, vol. 61, no. 1, pp. 226–237, 2011.

[27] L. M. Meade and A. Presley, “R&d project selection using the analytic network process,” *IEEE transactions on engineering management*, vol. 49, no. 1, pp. 59–66, 2002.

[28] M. A. Badri, D. Davis, and D. Davis, “A comprehensive 0–1 goal programming model for project selection,” *International Journal of Project Management*, vol. 19, no. 4, pp. 243–252, 2001.

[29] T. Erlebach, H. Kellerer, and U. Pferschy, "Approximating multi-objective knapsack problems," in *Workshop on Algorithms and Data Structures*, pp. 210–221, Springer, 2001.

[30] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[31] W. J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk, "Multi-objective decision analysis for competence-oriented project portfolio selection," *European Journal of Operational Research*, vol. 205, no. 3, pp. 670–679, 2010.

[32] C. McKenna, Z. Chalabi, D. Epstein, and K. Claxton, "Budgetary policies and available actions: a generalisation of decision rules for allocation and research decisions," *Journal of health economics*, vol. 29, no. 1, pp. 170–181, 2010.

[33] C. Bazgan, H. Hugot, and D. Vanderpooten, "Solving efficiently the 0–1 multi-objective knapsack problem," *Computers & Operations Research*, vol. 36, no. 1, pp. 260–279, 2009.

[34] M. E. Captivo, J. Clımaco, J. Figueira, E. Martins, and J. L. Santos, "Solving bicriteria 0–1 knapsack problems using a labeling algorithm," *Computers & Operations Research*, vol. 30, no. 12, pp. 1865–1886, 2003.

[35] G. W. Evans, "An overview of techniques for solving multiobjective mathematical programs," *Management Science*, vol. 30, no. 11, pp. 1268–1282, 1984.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[37] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," 1999.

[38] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "Mocell: A cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, vol. 24, no. 7, pp. 726–746, 2009.

[39] A. L. Medaglia, S. B. Graves, and J. L. Ringuest, "A multiobjective evolutionary approach for linearly constrained project selection under uncertainty," *European journal of operational research*, vol. 179, no. 3, pp. 869–894, 2007.

[40] A. F. Carazo, T. Gómez, J. Molina, A. G. Hernández-Díaz, F. M. Guerrero, and R. Caballero, "Solving a comprehensive model for multiobjective project portfolio selection," *Computers & operations research*, vol. 37, no. 4, pp. 630–639, 2010.

[41] K. Florios, G. Mavrotas, and D. Diakoulaki, "Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithms," *European Journal of Operational Research*, vol. 203, no. 1, pp. 14–21, 2010.

[42] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[43] "Source code." `http://moeaframework.org/features.html`. Accessed: December, 2017.

[44] A. Jaszkiewicz, "A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the pareto memetic algorithm," *Annals of Operations Research*, vol. 131, no. 1, pp. 135–158, 2004.

[45] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization.," tech. rep., Air Force Institution of Tech Wright-Patterson AFB OH, 1995.

[46] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *international conference on parallel problem solving from nature*, pp. 292–301, Springer, 1998.

[47] B. Lokman, M. Köksalan, P. J. Korhonen, and J. Wallenius, "An interactive algorithm to find the most preferred solution of multi-objective integer programs," *Annals of operations research*, vol. 245, no. 1-2, pp. 67–95, 2016.

[48] P. Korhonen, J. Wallenius, and S. Zionts, "Solving the discrete multiple criteria problem using convex cones," *Management Science*, vol. 30, no. 11, pp. 1336–1345, 1984.