GENERATING EVENLY DISTRIBUTED EQUITABLY EFFICIENT SOLUTIONS IN MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

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

Bashir Abdullahi Bashir

September 2018

Generating Evenly Distributed Equitably Efficient Solutions in Multi-objective Optimization Problems
By Bashir Abdullahi Bashir
September 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)

Emre Nadar

Ceren Tuncer Şakar

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

GENERATING EVENLY DISTRIBUTED EQUITABLY EFFICIENT SOLUTIONS IN MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Bashir Abdullahi Bashir
M.S. in Industrial Engineering
Advisor: Özlem Karsu
September 2018

We consider multi-objective optimization (MOP) problems where the decision maker (DM) has equity concerns. We assume that the preference model of the DM satisfies properties related to inequity-aversion, hence we focus on finding efficient solutions in line with the properties of inequity-averse preferences, namely the equitably efficient solutions.

We discuss two algorithms for finding good subsets of equitably efficient solutions. In the first approach, we propose an algorithm that generates an evenly distributed subset of the set of equitably efficient solutions to be considered further by the DM. The second approach is an extension of an interactive approach developed for finding efficient solutions in the rational dominance sense and finds equitably efficient solutions in the preferred region of the DM.

We illustrate these algorithms on equitable multi-objective knapsack problems that fund projects in different categories subject to a limited budget. We perform experiments to show and discuss the performances of the algorithms for three and five criteria settings. The experiments show that the first algorithm generates an evenly distributed subset in reasonable time, hence is advantageous in terms of solution time, compared to an approach that aims to find the whole set of equitably efficient solutions. The second approach is also shown to be a computationally efficient one that could be used in settings where the DM is willing to provide preference information.

Keywords: multi-objective knapsack problem, equitable preferences, equitable efficiency, generalized Lorenz dominance.

ÖZET

ÇOK AMAÇLI OPTİMİZASYON PROBLEMLERİNDE EŞİT DAĞILIMLI EŞİTLİKÇİ VERİMLİ ÇÖZÜMLER BULUNMASI

Bashir Abdullahi Bashir
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Özlem Karsu
Eylül 2018

Karar vericinin eşitlikçilik (adillik) kaygılarının olduğu durumlarda kullanılan çok amaçlı optimizasyon problemleri (ÇAOP) ele alınmıştır. Karar vericinin tercih modelinin eşitsizlikten kaçınma ile ilgili özellikler taşıdığı varsayılarak, bu özellikler duğrultusunda verimli çözümler, yani eşitlikçi verimli çözümler bulmaya odaklanılmıştır.

Eşitlikçi verimli çözüm kümesinden iyi alt kümeler bulunması için iki algoritma tartışılmıştır. İlk yaklaşımda, eşitlikçi verimli çözümler kümesinde eşit olarak dağıtılmış çözümler bulan bir algoritma önerilmektedir. İkinci yaklaşım, rasyonel baskınlık anlamında verimli çözümler bulmak için geliştirilmiş interaktif bir yaklaşımın bir uzantısıdır ve karar vericinin tercih ettiği bir bölgede eşitlikçi verimli çözümler bulmaktadır.

Bu algoritmaların kullanımı, sınırlı bir bütçe dahilinde farklı kategorilerdeki projeleri finanse eden eşitlikçi çok amaçlı sırt çantası problemleri üzerinde örneklenmiştir. Algoritmaların, üç ve beş kriterli problemlerde performanslarını gösterilmesi ve tartışılması için deneyler yapılmıştır. Önerilen ilk algoritmanın, dengeli şekilde dağılmış eşitlikçi verimli çözümler bulduğu ve bütün eşitlikçi verimli çözümleri bulan bir yaklaşıma göre süre açısından avantajlı olduğu gösterilmiştir. İkinci yaklaşımın da, karar vericinin tercih bilgisi sağlayabileceği durumlarda kullanılabilecek, çözüm süresi açısından verimli bir yaklaşım olduğu gözlemlenmiştir.

Anahtar sözcükler: çok amaçlı sırt çantası problemi, eşitlikçi tercihler, eşitlikçi verimlilik, genelleştirilmiş Lorenz baskınlığı.

Acknowledgement

I would like to express my deepest gratitude to my advisor Asst. Prof. Özlem Karsu whose unwavering support, patience and guidance has been invaluable throughout my masters programme. I have to say, I am most fortunate to have been her student.

Also, I would like to thank Asst. Prof. Ceren Tuncer Şakar and Asst. Prof. Emre Nadar for accepting to read and review my thesis. Their comments and feedbacks have been most valuable.

I would like to take a moment and remember perhaps the two most influential people in my life. My late grandparents Justice Bashir Ma'aji Ishaq and Alhaji Balarabe Husseini (Dan Majen Zazzau) whose lives have been sources of immense lessons, inspirations and benchmarks. They lived a good life, were fearless and steadfast in righteousness. Their life stories have shaped who I am and make me be a better version of myself everyday. And to my living grandparents, I love you very much and pray for your good health.

I would like to express my gratitude to my parents, Alhaji Abdullahi Bashir Ishaq, CNA, MBA and Hajiya Mariya Balarabe for the immeasurable support and sacrifices they made for me and my siblings. They blessed us with the necessary upbringing to prepare us for the challenges in life and to pursue our dreams. There are no sufficient words for me express how thankful I am to them, but I have to say, I am who I am today because of them. So thank you very much. I would also like to show my appreciation to my best friend, my brother, Aliyu Abdullahi Bashir who has been my confidant, my punching bag to whom I constantly express my challenges. Furthermore, to my uncles and aunts thank you very much for always being there for me.

I would like to thank the Industrial Engineering department of Bilkent University for providing the most conducive environment for me to conduct my research and their financial support throughout my postgraduate programme.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multi objective optimization (MOP) problems have been studied for many years. Different techniques have been used to successfully solve and analyse these problems in a wide range of application areas such as engineering design, medical treatments, logistics, resource allocation and facility location ([1],[2],[3]). In a typical MOP, multiple objective functions that correspond to decision criteria are simultaneously optimized over a feasible region. There are trade-offs between the multiple objectives considered, hence, usually no single solution optimizes all of the objective functions simultaneously. Due to these trade-offs, the concept of optimality is replaced with the concept of Pareto optimality (nondominance).

Equity concerns arise in various real life problems and it is vital to handle them for the recommended solutions to be admissible [4]. Addressing these concerns is challenging. The motivation for these concerns and the decision maker (DM)'s understanding of fairness may lead to different equity-related concerns like equitability and balance. Moreover, equity is rarely the only concern. The decision makers usually consider the trade-offs between equity and efficiency.

Researchers have started to consider extensions of several classical problems like knapsack, assignment and location problems to incorporate equity concerns. The notion of equity is usually studied in allocation settings where we try to attain a "fair" allocation of the resources or outcomes by treating the involved entities in an impartial manner.

In general, any system serving multiple users where the service quality for every individual user is taken into consideration can be assessed with equity concerns. The users or entities involved can be departments of an organization, people of different social classes, customers at different locations, etc. For example, public service location models strive to provide equitable access to different demand points (customers).

The need for inequity averseness naturally occurs in various operational research (OR) applications, including but not limited to vehicle routing problems during disaster relief [5], workload allocation, queuing systems, bandwidth allocation and healthcare service provision (See [4] and the references therein). Another application shown in [6] is that of partitioning physical electrical grids into companies and incorporating equity concerns for the companies earnings.

We consider problem settings where equity concerns over multiple categories/entities are involved. Hence the problems we consider are different in the sense that all objectives are of the same type (a single type of benefit), and it is the concern of maximizing the benefit received by each category (entity) that makes the problem a multi-objective one. We call this problem multi-objective optimization problem with equity concerns (E-MOP). Unlike a classical MOP, in E-MOP the values of the objective functions are comparable. Furthermore, the criteria are considered impartially, which makes the distribution of the criteria values more important than the assigned outcome to a specific criterion. For example, in a setting with three entities, the decision maker would be indifferent between $(90, 100, 70)$ and $(100, 70, 90)$.

Incorporating equity concerns into the preference model of the DM makes some solutions which are non-dominated (in classical dominance sense) unattractive. Therefore, rather than focusing on the Pareto efficient solutions, we focus on the more relevant *equitably efficient* solutions. Equitable efficiency was defined in [7] and an approach to find non-dominated points for MOP with equity concerns by aggregating the objective functions has been studied in [8]. A two-step method to find equitable efficient solutions for MOP was developed in [9].

Motivated by the observation that it may be computationally too expensive to find the whole set of equitably efficient solutions, we consider two algorithms that find a subset of it. The algorithms we consider can be applied to find solutions to any such MOP. We exemplify their use for project portfolio selection problems where decision makers have fairness concerns.

One example is the investment decision problem, in which projects that will provide different benefits to different beneficiary groups (different population groups or different geographical zones) are considered. Each project is associated with an output vector, showing the amount of benefit it provides to these different groups, which we call entities throughout the text. In such cases, a typical concern for decision makers is ensuring an equitable benefit allocation over the multiple entities and a total benefit maximizing approach is usually considered inapplicable as it may result in extreme inequity in the benefit distribution.

Another example occurs when project proposals that belong to different categories are evaluated and it is important to ensure a balanced funding over the multiple categories involved. A total value maximizing approach may result in imbalanced funding decisions in the sense that the majority of the funded proposals might belong to a single category [10].

The rest of this thesis is structured as follows:

Chapter 2: We provide review of the related literature on MOP problems and equity concerns in the literature.

Chapter 3: We discuss the concept of equitable efficiency, alongside the underlying assumptions on the decision maker's preference model and provide mathematical models that can be used to find the set of equitably efficient solutions.

Chapter 4: We propose a novel approach to generate evenly distributed equitable non-dominated points.

Chapter 5: We utilize an interactive approach that generates highly preferred equitable non-dominated points.

Chapter 6: We provide the summary of our computational experiments, in which we demonstrate the performance of the algorithms using an equitable knapsack problem.

Chapter 7: We conclude our discussion and list some future research directions.

Chapter 2

Literature Review

In this thesis, we study MOP problems where the decision maker (DM) has equity concerns. We consider two algorithms to solve these problems. We apply these algorithms to project portfolio problems that are formulated as multi-objective binary knapsack problems where the DM has fairness concerns. Hence our study is mainly related to equity concerns and multi-objective optimization. In the first section of this chapter, we discuss the solution methods used in solving multi-objective optimization (MOP) problems. We focus mainly on the works in the literature that consider generating exact efficient solutions for MOP problems with more than two objectives. In the second section, we discuss equity concerns in operational research.

2.1 Multi Objective Optimization Problems

Multi objective optimization or vector optimization problems are problems where we consider optimizing a collection (at least two) of objective functions simultaneously. A general MOP with m objectives is formulated as follows:

$$Max\text{“}z_1(x), z_2(x), \cdots, z_m(x)\text{”}$$
$$s.t. x \in \mathbb{X} \tag{2.1}$$

x denotes the vector of decision variables and $\mathbb{X}$ is the feasible decision space. $(z_1(x), z_2(x), \cdots, z_m(x))$, is the vector of the m objective functions considered.

The quotation marks in the objective function implies that the "*max*" operator used in these settings is not well-defined. Due to the conflicting nature of the objective functions there usually is not a single feasible point that maximizes all the objectives. Therefore, solving these problems refers to finding the whole set (or a subset) of the efficient solutions:

Definition 1 (Maximization setting) *Let $x' \in \mathbb{X}$, be a feasible solution to the generic MOP model above, x' is efficient if $\nexists x \in \mathbb{X}$ such that $z_k(x') \leq z_k(x)$ $\forall k = 1, 2, \cdots m$ and $z_k(x') < z_k(x)$ for at least one k. The image of x' in the criteria space $z(x')$ is non-dominated.*

There is a vast literature on approaches to find efficient solutions for various MOP problems. The solution methods can be classified into two groups according to the type of solution generated, namely, exact and approximate (usually referred to as heuristic or meta-heuristic) solution methods. The exact solution methods are further classified into two; those that generate a representation of the whole efficient frontier and those that generate a single efficient solution as shown in Figure 2.1.

Heuristics are usually problem-dependent techniques that aim to generate near optimal solutions (without guarantee of feasibility or optimality) at a reasonable computational cost. On the other hand, meta-heuristics are stronger techniques applicable to various problems. They are powerful algorithmic frameworks providing sets of guidelines to develop other heuristics by combining different concepts to explore the search region. [11] presents a survey on the heuristics and meta-heuristics methods used for multi objective combinatorial optimization (MOCO) problems in the literature. In Figure 2.1, we show some examples of these methods.

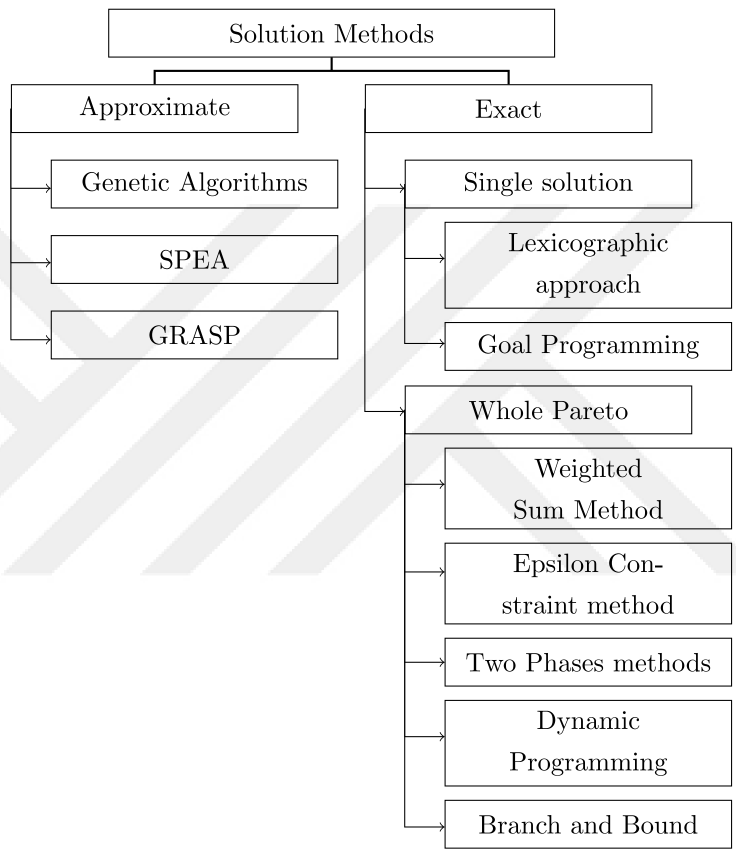


Figure 2.1: Classification of solution methodologies based on the types of solutions generated

Typically, there are many efficient solutions in an MOP of which one is preferred at least as good as the rest by the decision maker (assuming the aim is to find a single most preferred solution). When there are at least two efficient solutions, more information (preference information) is needed from the decision maker to further reduce the solution space and converge to the most preferred solution. The information taken can be in the form of a predetermined goal for each objective function, an explicit value function or pairwise comparisons of the efficient solutions. Figure 2.2 shows the main classifications of the solution methods in terms of the time the preference information is obtained[12].

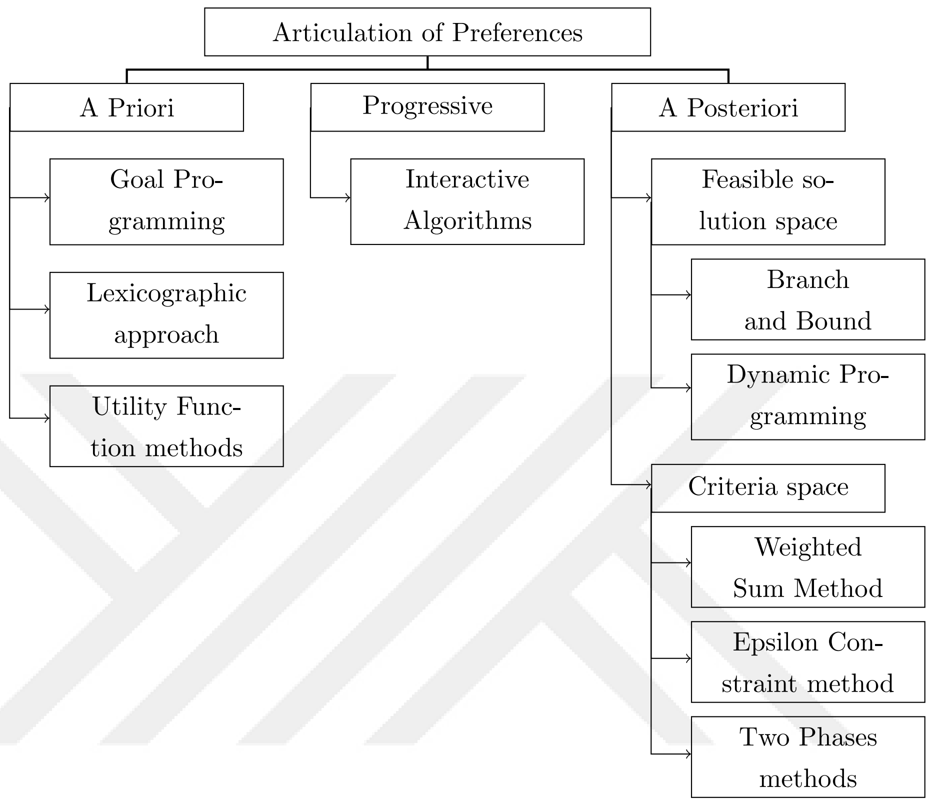


Figure 2.2: Classification of solution methodologies based on the timing of the articulation of preferences

The methods that employ a prior articulation of preferences require the decision maker to specify preferences (in terms of goals, relative importance of the objectives, etc) before the optimization. To reflect the decision maker's preferences, these methods incorporate parameters in terms of weights or constraint bounds to optimize the decision maker's value function. The optimization models in these methods are usually straight forward, however, the decision maker may face challenges in determining the preference information needed.

Approaches based on the progressive articulation of preferences involve iteratively using the preference information obtained from the decision maker during the optimization. At each iteration, the decision maker's responses are used as inputs in the next iterations to reduce the solution space until her best solution is attained. Usually, the aim is finding one best solution in line with the decision maker's preference information.

[13] presents a detailed review on the interactive methods used in handling multi-objective integer (MOIP) and mixed integer programming (MOMIP) problems. Some interactive approaches assume the existence of a utility function that represents the decision maker's preferences and they work on finding the solution that optimizes the utility function ([14], [15]). However, some other approaches do not assume the existence of a utility functions [16]. Instead they pursue a progressive approach where they help the decision maker avoid non-dominated solutions she is not interested in. They do not necessarily try to find the best solution but help in finding satisfactory solutions.

In general, these interactive methods help reduce the computational effort and ease the decision maker's decision process.

We propose an algorithm that finds a subset of the set of equitably efficient solutions. Therefore, our study is most relevant to the posteriori approaches discussed in the literature. We provide a detailed discussion on these approaches.

Approaches classified as posteriori aim to generate all the non-dominated points (or a subset) and present these solutions to the decision maker for further consideration. These methods are classified into two; those that work on the feasible solution space and those that work on the criteria space as shown in Figure 2.2. In general, when generating the non-dominated points, the only assumption made about the decision maker value function is monotonicity ("more is better"). Moreover, the decision maker is expected to choose a most preferred solution among these points.

These methods are not without their challenges. In general, solving MOP problems using these methods is computationally intractable and expensive. Moreover, due to the cardinality of the set of the pareto solutions, it is strenuous for the decision maker to compare or rank all the solutions (in an attempt to choose a best solution). Our approach presented in Chapter 4 addresses these issues by generating a well spread subset of the non-dominated points for the decision maker to consider.

Most studies in the literature deal with generating the Pareto solutions for bi-objective programming problems. In this review, we consider papers that generate exact solutions where more than two objectives are considered.

[18] presents a branch and bound algorithm to generate the efficient set in MOMIP problems where the integer variables are binary. They extend the classical branch and bound algorithm to handle this problem. They use a depth first search algorithm where vector comparisons instead of scalar ones are used in the fathoming tests. To handle large problems, they implement interactive approaches guided by the decision maker's preferences.

[19] develops an approach for generating all non-dominated points of Multi objective integer linear programming problems. The algorithm starts with a selected weighted vector used in linearising the multiple objectives and sequentially generates non-dominated solutions by adding more constraints at every iteration until the model becomes infeasible. At any iteration l, the algorithm uses the solution found in the previous iteration $(l-1)$ and adds m binary variables and $m+1$ constraints to eliminate previously generated solutions, where m is the number of objectives. Hence, for any iteration, the generation of the next non-dominated point is computationally harder.

[20] develops an approach to generate all non-dominated points for MOIP problems. This approach is an improvement over the classical epsilon constraint method. For any MOIP problem with m objectives, they add one of the objectives as a constraint bounded by its global upper bound value over all the feasible solutions and solve the problem as MOIP with $m-1$ objectives. They generate all the non-dominate points by systematically decreasing the upper bound and solving the models until the model becomes infeasible. [21] significantly improves this approach by utilizing the set of already solved subproblems. As the algorithm progresses, they reuse the solutions to problems solved in previous iterations. This algorithms improves that of [20] both in terms of CPU time and the number of models solved.

[22] proposes two exact algorithms for MOIP problems. Like the algorithm in [19], their first algorithm also generates non-dominated points by adding constraints and binary variables. At each iteration, the algorithms generate new non-dominated points by eliminating the previously generated points and the regions they dominate from the search space. Assuming there are n non-dominated points and m objectives, the first algorithm reduces the number of binary variables added from nm to $n(m-1)$ and constraints from $n(m+1)$ to nm. However, the model still grows large when there are a lot of non-dominated points. They develop a second algorithm which improves the first where at any iteration to find the n^{th} point, the model is decomposed into $(n+1)$ submodels. The second algorithm outperforms the first algorithm and the algorithms of [19] and [20].

[23] considers the problem of generating all non-dominated points for multi objective discrete optimization (MODO) problems. They propose an algorithm which searches the feasible region over $(m-1)$ dimensional rectangles where m is the number of objectives. For each rectangle two-stage optimization problems are solved to avoid generating weakly efficient solutions. This algorithm outperforms those of [19] and [20] in terms of both the CPU time and the number of models solved.

[24] presents an approach for generating the exact Pareto set in MOP problems. They present AUGMECON2, an improvement on AUGMECON, an approach previously developed by the authors. AUGMECON is an augmented epsilon constraint method where one of the objective functions is maximized and the other objective functions are used as constraints. They use the surplus variables of the constraints to form the augmented part. In AUGMECON2 the objective function is modified in order to perform a lexicographic optimization on the augmented part of the objective function. They also use the information from the surplus or slack variables of the constrained objectives using a bypass coefficient to avoid redundant iterations, reduce the number of subproblems solved and accelerate the algorithm.

Another variation (SAUGMECON) of AUGMECON is studied in [25] which is found to be more efficient. They accelerate the algorithm using bouncing steps and early exit strategies. They perform experiments on randomly generated MOIP problems and use the average number of integer programming models solved as the criterion for performance comparisons with other works in the literature. Their approach outperforms those of [20] and [22].

2.2 Equity Concerns in Operational Research

In this section we present literature review on equity concerns in operational research. Equity concerns have been studied and applied in various areas such as postdisaster humanitarian relief [26],[27],[28] distribution of donated food [29], food rescue and distribution [30], healthcare facility location and allocation [31] and air traffic flow management [32]. The presence of equity concerns is perhaps felt strongest in the public sector where different groups or classes of people are to be treated in a fair manner. However, equity is seldom the only criterion considered. Most applications consider efficiency concerns alongside these equity concerns. Therefore, many researchers model these problems as multi criteria decision making problems.

Equity being a relative and ethical concept is perceived differently among decision makers. [33] presents an overview of the equality measures in location problems and propose new properties that such measures should possess. [4] discusses equitability and balance concerns; two notions related to equity concerns in the literature. Equitability considers achieving an even allocation of resources among indistinguishable entities or groups. In this case, the identity of the entities are not relevant (anonymity holds). Whereas balance concerns arise when the characteristics and needs of the entities are recognised and relevant in evaluating different allocations. As mentioned earlier, we consider a problem setting where anonymity holds.

In mathematical modelling, the three main approaches used in handling equitability concerns are:

- The Rawlsian approach: this approach incorporates equitability concerns by focusing on the minimum amount allocated across the entities, i.e., the better the worst-off entity the better the distribution.

- The inequality index based approaches: These approaches incorporate equitability concerns by using inequality indices. These indices evaluate and show the level of inequality of a given distribution. The indices can be integrated into the models either as an objective to be optimized or in a constraint.

- The inequity-averse aggregation function based approaches: In these approaches, an aggregation function of the distribution vector that encourages equitable distributions is used in the model. An inequity-averse function should be symmetric (anonymity holds) and should emulate concerns with regards to inequity-aversion.

[7] introduces the concept of equitable efficiency. They define a set of axioms namely; reflexivity, transitivity, strict monotonicity, anonymity and Pigou-Dalton principle of transfers on the decision makers's preference model in defining equitable dominance. [8] calls an aggregation function in accordance with this set of axioms an equitable aggregation function. Such functions must be strictly increasing, symmetric and satisfy the principles of transfers (a strictly increasing and strictly Schur-concave function). Any solution that maximizes an equitable aggregation function is equitably efficient.

[8] presents aggregations for generating equitably efficient solutions to both linear and nonlinear multi criteria optimization problems. [34] studies the ordered weighted averaging (OWA) aggregation applied to multi criteria problems where the criteria are considered to be uniform and the distribution of their values are important. They analyse the solution procedures for linear programs with OWA objective functions. [9] uses a cone based approach to study the equitable preference structure. To find the equitably efficient solutions, they developed a two step method. An epsilon constraint scalarization of the MOP is used to find an efficient solution in the first step which is used in the second step to generate an equitably efficient solution by solving a nonlinear problem.

In this study, we propose a novel approach for generating evenly dispersed equitably efficient solutions for E-MOPs. We also discuss an interactive approach that extends the algorithm of [17] for equitable MOP. We perform experiments and compare these approaches with an approach that finds the whole set of equitably efficient solutions. Both algorithms are shown to be computationally efficient. The first one indeed finds evenly spread equitably efficient solutions and hence provides an informative subset of the Pareto frontier to the DM. The second one can be used when there is access to preference information from the DM.

Chapter 3

Equitable Dominance and Equitable Efficiency

Consider a generic multiobjective programming model with m objectives:

$$\begin{aligned} &Model\ 1 \\ &Max``z_1(x), z_2(x), \cdots, z_m(x)" \\ &s.t. x \ \in \mathbb{X} \end{aligned} \tag{3.1}$$

x denotes the vector of decision variables and $\mathbb{X}$ is the feasible decision space. In the problems we consider, each objective function, $z_j(x)$, denotes the total output received by entity j in a feasible solution x.

Note that the "*max*" operator used in these settings is not a well-defined operator. Hence, solving these models refers to finding the most preferred solution or a set of "good" solutions that are candidates to be the most preferred solution. As mentioned in the previous chapter, the solution concepts applied in multiple criteria decision making literature rely mainly on three ideas, namely: aggregating the multiple objectives into one and maximizing this aggregate function; using interactive methods that take preference information from the DM and reduce the solution space based on her responses; and finding the non-dominated frontier (or a subset of it) and presenting it to the DM for further consideration.

Unlike a classical multiobjective programming setting, we assume that the preference model of the DM reflects inequity-aversion, therefore we are interested in finding the set of equitably efficient solutions. We now explain the equitable dominance relation that we use in this study and define equitable efficiency.

The following dominance relation is used for a rational decision maker whose preferences can be modeled with a weak preference relation, which is reflexive, transitive and monotonic.

Definition 2 (Weak Classical (Rational) Dominance) *Consider two solutions to Model 1 with output vectors z, z'. z rationally dominates z' ($z' \preccurlyeq_r z$) if and only if z is preferred to z' by all rational decision makers. i.e*
$z' \preccurlyeq_r z \iff z'_j \leq z_j$ *for all* $j \in M = \{1, 2, ..., m\}$.

An output vector z is non-dominated if there is no z' that dominates it. The corresponding solution is called efficient. Note that any rational decision maker's preference relation is assumed to be in line with the classical dominance.

However, in our problem setting, we assume that the DM has equity concerns. To reflect these concerns, we assume two more properties for the preference model, namely: symmetry and Pigou-Dalton principle of transfers.

1. Symmetry: This property states that the decision maker is indifferent between a feasible solution with an output vector z and any other feasible solution whose output vector is a permutation of the vector z. For example, the DM is indifferent among $(3, 5, 8)$, $(5, 3, 8)$ and any other permutation of these.

2. Pigou-Dalton principle of transfers: This property states that for any two solutions that have same total output, if one solution is obtained by transferring output from a better-off entity to a worse-off one from the other solution, then it is considered better. For example, the DM prefers $(5, 5, 6)$ to $(3, 5, 8)$.

A rational preference relation, which additionally satisfies symmetry and Pigou-Dalton principle of transfers properties, is called an equitable preference relation [7].

Definition 3 *Consider two solutions to model 1 with output vectors z, z'. z Equitably dominates z' ($z' \preccurlyeq_e z$) if and only if z is preferred to z' by all decision makers with equitable preference relations.*

A feasible solution with output vector z is equitably efficient if there is no z' that equitably dominates z. Note that equitable dominance is the generalized Lorenz (GL) dominance discussed in the economics literature [35]. Hence we will refer to z as *equitably non-dominated* (meaning nondominated in the GL sense).

Theorem 4 (Weak Equitable Dominance) $z' \preccurlyeq_e z \iff \sum_{j=1}^{k} \overrightarrow{z'}_j \leq \sum_{j=1}^{k} \overrightarrow{z}_j$ *for all $k \in M$. The vector $\overrightarrow{z}$ is the ordered permutation of z with elements ordered in a non-decreasing fashion, i.e, $\overrightarrow{z}_j$ is a vector whose elements express respectively: the minimum outcome, the second minimum outcome, the third minimum outcome, etc. of the outcome vector z.*

Utilizing Theorem 4, finding equitably efficient solutions to $Model$ 1 is equivalent to finding efficient solutions to $Model$ 2 below.

$$Model\ 2$$

$$Max \text{ ``}\overrightarrow{z}_1, \overrightarrow{z}_1 + \overrightarrow{z}_2, \cdots, \sum_{j=1}^{m} \overrightarrow{z}_j\text{''}$$

$$s.t.\ x \in \mathbb{X}$$

$$z_j = z_j(x)$$

$Model$ 2 above is not linear due to the use of the ordering operator $\overrightarrow{(.)}$. However, it has been shown in [34] that for any given output vector z, the cumulative ordered elements $\sum_{j=1}^{k} \overrightarrow{z}_j$ for any $k \in M$ can be found by solving the model below:

$Model\,3$

$$\sum_{j=1}^{k} \overrightarrow{z}_j = Max\; k * r_k - \sum_{j=1}^{m} d_{kj}$$

$$s.t \quad r_k - z_j \leq d_{kj} \qquad \forall j \in M \qquad (3.2)$$

$$d_{kj} \leq BM * t_{kj} \qquad \forall j \in M \qquad (3.3)$$

$$\sum_{j=1}^{m} t_{kj} \leq k - 1 \qquad (3.4)$$

$$d_{kj} \geq 0 \qquad \forall j \in M \qquad (3.5)$$

$$t_{kj} \in \{0,1\} \qquad \forall j \in M \qquad (3.6)$$

where BM is a sufficiently large constant and

$$t_{kj} = \begin{cases} 1, \; if \;\; \overrightarrow{z}_k > z_j \\ 0, \; otherwise \end{cases}$$

Moreover, it is shown in [34] that the optimization problem defining $\sum_{j=1}^{k} \overrightarrow{z}_j$ above can be greatly simplified since all the binary variables and their related constraints can be eliminated without loss of generality as shown in the theorem below.

Theorem 5 *For any given vector* $z \in \Re^m$*, the cumulative ordered elements* $\sum_{j=1}^{k} \overrightarrow{z}_j$ *for any* $k \in M$ *can be found by solving the problem below:*

$Model\,4$

$$\sum_{j=1}^{k} \overrightarrow{z}_j = Max\; k * r_k - \sum_{j=1}^{m} d_{kj}$$

$$s.t. \;\; ineq\,(3.2), (3.5)$$

It is noteworthy to state that model 4 has alternate optimal solutions (see

[36]). An optimal solution is as follows; Let $r^*_k = \overrightarrow{z}_k$ and

$$d^*_{kj} = \begin{cases} \overrightarrow{z}_k - z_j, \ if \ \overrightarrow{z}_k \geq z_j \\ 0, \ if \ \overrightarrow{z}_k < z_j \end{cases}$$

Hence, we have the optimal value as $kr^*_k - \sum_{j=1}^{m} d^*_{kj} = k\overrightarrow{z}_k - \sum_{j:\overrightarrow{z}_k \geq z_j} (\overrightarrow{z}_k - z_j) = \sum_{i=1}^{k} \overrightarrow{z}_i$. Other alternative optimal solutions can be found by making $r^*_k = \overrightarrow{z}_k + c$ where c is a positive constant. Consequently, we have $d^*_{kj} = \overrightarrow{z}_k + c - z_j$ for $j : \overrightarrow{z}_k \geq z_j$. Utilizing Theorem 5, $Model\ 2$ can be reformulated as follows:

$Model\ 5$

$$Max \text{``} y_1, y_2, \cdots, y_m \text{''}$$

$$s.t. \quad y_k - \left(k * r_k - \sum_{j=1}^{m} d_{kj} \right) = 0 \quad \forall k \in M \tag{3.7}$$

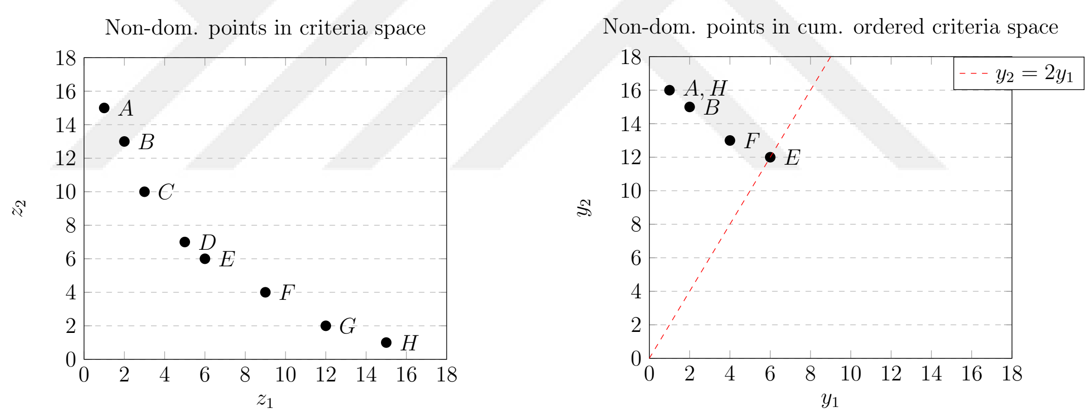
$$x \in \mathbb{X}$$

$$z_j = z_j(x)$$

$ineq.$ (3.2), (3.5)

Here, y_k is the sum of the "k" smallest components of any output vector z. For simplicity, let the feasible set of such y vectors in $Model\ 5$ be represented by $\{y \in \Re^m : y \in \mathcal{Y}\}$. $Model\ 5$ above transforms the criteria space into cumulative ordered criteria space. Finding the non-dominated solutions to this model is equivalent to finding equitably efficient solutions to Model 1 (by Theorem 4). Such a transformation is illustrated in Figure 3.1 below.

Figure 3.1a below shows eight non-dominated points in the classical dominance sense plotted in the criteria space (in terms of the variables z_k). These points are Pareto optimal points in the rational dominance sense. However, we are interested in finding the equitably efficient points among these. To achieve this, we transform the space into the cumulative ordered criteria space (in terms of the variables y_k) and find the non-dominated cumulative ordered vectors (y's) as shown in Figure 3.1b. It can be seen that the number of non-dominated points in Figure 3.1b is less than that of Figure 3.1a (the set of equitable non-dominated points is a subset of the set of non-dominated points (see [9]). This is a direct result of the symmetry and Pigou-Dalton principle of transfers properties of the equitable preference relation. The points $(1, 15)$ and $(15, 1)$ in Figure 3.1a correspond to the point $(1, 16)$ in Figure 3.1b. The points $(5, 7)$ and $(3, 10)$ in Figure 3.1a correspond to the points $(5, 12)$ and $(3, 13)$ respectively, in the cumulative ordered space. These points are dominated by $(6, 12)$ and $(4, 13)$ respectively. Moreover, it can be seen from Figure 3.1b that for the case when $m = 2$ (in 2D) the transformed points lie in $y_2 \geq 2y_1$ region of the space.



(a)

(b)

Figure 3.1: Non-dom. points in criteria and cum. ordered criteria space

Note that the algorithms developed to find all Pareto optimal points for classical MOP can be used to find equitably efficient solutions. However, any such algorithm should be modified so that one works on the cumulative space, leading to the equitable MOP (Model 5). This modification is not always trivial due to the ordering operator.

There are numerous efficient solutions (both in the classical and equitable dominance sense) in large MOP problems and it may not be practical or useful to generate them all. Therefore, approaches that generate a "good " subset of the Pareto solutions are crucial.

One way of handling this computational challenge would be generating an evenly distributed subset of the equitably efficient solutions and present them to the DM. Another approach could be finding the solutions that are of interest to the DM by incorporating her preferences into the solution procedure. We could employ interactive approaches that take the DM's preferences into account and use the information to converge to a single most preferred non-dominated point. In this paper, we discuss two such algorithms.

We first develop an algorithm that finds a set of evenly distributed equitably efficient solutions to be presented to the DM. We then discuss the application of another interactive algorithm that finds the set of equitably efficient solutions in the preferred region of the DM. In both approaches, we work on the cumulative ordered space since finding nondominated points in this space is equivalent to finding equitably nondominated points in the original space.

In the next section we explain the algorithm we propose to generate evenly distributed equitably non-dominated points.

Chapter 4

Algorithm for Generating Evenly Distributed Equitably Non-Dominated Points (The GEND Algorithm)

Our approach is based on fitting a hyperplane function that is close to the non-dominated frontier in the cumulative ordered criteria space (hence nondominated frontier in the Generalized Lorenz sense). We then select representative points on the hyperplane, generate regions around those points and search those regions for non-dominated points. This way, we generate a subset of the set of equitably non-dominated solutions that is well spread.

The hyperplane could be placed at different positions relative to the non-dominated frontier. Figure 4.1 below shows the hyperplane placed above the non-dominated frontier for a maximization setting.

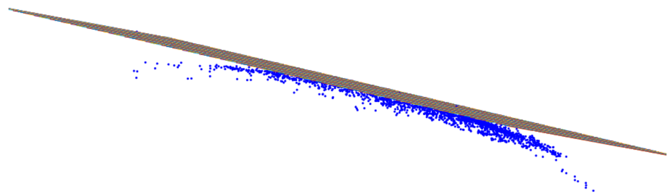


Figure 4.1: Hyperplane fitted using point with maximum total outcome value

In the next three subsections we explain the three main parts of the algorithm, fitting the hyperplane, defining the regions to be explored and finding the solutions within the specified regions.

4.1 Fitting the hyperplane

The hyperplane we fit is of the form $\sum_{k=1}^{m} y_k = p$. We explore the strategy of fitting a hyperplane that passes through the solution that has the maximum total outcome value. Hence, we set $p = \{max \sum_{k=1}^{m} y_k : y \in \mathcal{Y}\}$ to fit the hyperplane above the frontier.

Alternatively, one could set $p = \sum_{k=1}^{m} yc_k$, where $yc \in \Re^m$ is the centrally located equitably non-dominated point found by solving the equal-weighted Tchebycheff program, M_{ch} discussed in section 5.2.

4.2 Defining the regions

We define well spread regions in the cumulative ordered criteria space around some selected representative points on the hyperplane fitted in section 4.1. As shown in Figure 3.1b, in $\Re^2$, the non-dominated points in the cumulative ordered criteria space are restricted to the region defined by the polyhedron $Q = \{y \in \Re^2 : y_2 \geq 2y_1\}$. A similar analogy can be made for higher dimensional real spaces. For example, in $\Re^3$ the region is defined by $Q = \{y \in \Re^3 : y_2 \geq 2y_1,\ y_3 \geq 3y_1,\ 2y_3 \geq 3y_2\}$. In general, we can state the proposition below:

Proposition 6 *For any real space $\Re^m$, let $M = \{1, 2, \cdots, m\}$. The non-dominated points in the cumulative ordered criteria space are restricted to the polyhedron $Q = \{y \in \Re^m : jy_k \geq ky_j \;\; forall\, j, k \in M : k > j\}$.*

The proof of Proposition 6 is provided in Appendix B. To define the regions, we select a number of representative points yr, from the restricted polyhedron Q that lie on the hyperplane defined in section 4.1, since proposition 6 implies that there is no need to focus on the region outside Q.

Any representative point $yr \in \Re^m$ on the fitted surface will most likely be an infeasible or dominated point. We define a region around it and find equitably non-dominated points in the region. Note that the region defined around yr may or may not contain any equitably non-dominated point in it. To assure obtaining at least one such point, we generate the non-dominated point $yrt \in \Re^m$, that is at minimum Tchebycheff distance from the ideal point y^{IP} in the direction of the reference point yr by solving the problem below:

$$(M_{chev})$$

$$Min \;\; \rho_{max} + \varepsilon_1 * \sum_{k=1}^{m} y_k$$

$$s.t. \;\; \rho_{max} \geq \lambda_k^{chv}(y_k^{IP} - y_k) \qquad \forall k = 1, 2, \cdots, m. \tag{4.1}$$

$$y \in \mathcal{Y}$$

where ρ_{max} measures the maximum component-wise weighted distance from the ideal point y^{IP} in the direction of the reference point yr and ε_1 is a small enough positive number.

The weight vector $\lambda^{chv} \in \Re^m$, corresponds to the diagonal direction for the reference point yr from the ideal point y^{IP}. The general form of λ^{chv} given in

[37] (p.425) is as follows:

$$\lambda_k^{chv} = \begin{cases} \frac{\frac{1}{(y_k^{IP} - yr_k)}}{} \left[\sum_{j=1}^{m} \frac{1}{(y_j^{IP} - yr_j)} \right]^{-1} & \text{if } \; yr_k \neq y_k^{IP} \;\; \forall k \in M \\ 1 & \text{if } \; yr_k = y_k^{IP} \\ 0 & \text{if } \; yr_k \neq y_k^{IP} \text{ but } \exists j \in M : yr_j = y_k^{IP} \end{cases}$$

additionally, we also generate the Pareto optimal point yrl that is at minimum equal weighted linear distance from the ideal point y^{IP} by solving the problem below:

$$(M_{linr})$$

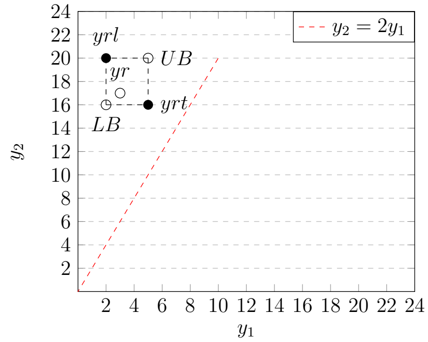
$$Min \sum_{k=1}^{m} \lambda_k^{lnr} (y_k^{IP} - y_k) + \varepsilon_1 * \sum_{k=1}^{m} y_k$$

$$s.t. \;\; y \in \mathcal{Y}$$

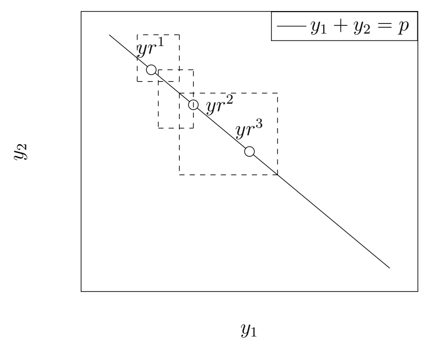
where $\lambda_k^{lnr} = \frac{1}{m}$ for all $k = 1, 2, \cdots, m$.

Note that for all $k \in M$, the element of the ideal point in the cumulative ordered criteria space, y_k^{IP}, is found by solving $max \; \{y_k : y \in \mathcal{Y}\}$.

For any representative point yr, let yrt and yrl be the optimal solutions obtained from solving M_{chev} and M_{linr} respectively. We determine the region by defining upper and lower bound as in [17]. We define the k^{th} element of the upper bound UB_k to be $max(yrt_k, yrl_k)$ and that of the lower bound LB_k to be $min(yrt_k, yrl_k)$ as illustrated in Figure 4.2a below. Note that we have multiple reference points, hence there is a possibility of generating intersecting regions as shown in Figure 4.2b below. However, if the reference points are chosen in a nice fashion (as shown in section 5), the size of the intersecting areas of the regions can be mitigated. Furthermore, we could unify the intersecting regions in order to eliminate the possibility of generating any solution from two or more generated regions.



(a) Generating a region around yr



(b) Intersecting regions

Figure 4.2: Generating regions around reference points

In the next section, we explain the exact algorithm we used to generate the equitably non-dominated points in the regions defined in section 4.2.

4.3 Finding the equitably non-dominated points in the defined regions

The exact algorithm we use in this section is based on the epsilon-constraint scalarization method (see [22]). The algorithm generates equitably non-dominated points in evenly distributed subsets of the feasible set, i.e. the regions generated in section 4.2. To explore each region, we solve scalarization models with additional constraints $LB_k \leq y_k \leq UB_k$ $k = 1, 2, \cdots, m$, where LB_k and UB_k are as explained in section 4.2.

We initialize the algorithm by arbitrarily choosing a region, r to begin with and a criterion, n. Then we solve the model below (M_n^0) to obtain the solution that maximizes the *nth* criterion in the region.

$$
\begin{aligned}
&(M_n^0) \\
&Max \;\; y_n + \varepsilon_1 * \sum_{k \neq n} y_k \\
&s.t. \\
&y_k \geq LB_k \qquad\qquad \forall k = 1, 2, \cdots, m \qquad (4.2) \\
&y_k \leq UB_k \qquad\qquad \forall k = 1, 2, \cdots, m \qquad (4.3) \\
&y \in \mathcal{Y}
\end{aligned}
$$

where ε_1 is as in section 4.1 and the augmented part of the objective function (term with ε_1) is used to make sure the model does not return a dominated point in the region. The optimal solution to the model (M_n^0) above, denoted by $\hat{y}^0 \in \Re^m$, may or may not be dominated by a feasible point outside the region. We solve the (MD_n^ω) model below with $\omega = 0$ to check whether the point $\hat{y}^0$ is dominated or not.

$$
\begin{aligned}
&(MD_n^\omega) \\
&Max \;\; y_n + \varepsilon_1 * \sum_{k \neq n} y_k \\
&s.t. \\
&y_k \geq \hat{y}_k^\omega \qquad\qquad \forall k = 1, 2, \cdots, m \qquad (4.4) \\
&y \in \mathcal{Y}
\end{aligned}
$$

Let the optimal solution to (MD_n^0) be $\overline{y}^0 \in \Re^m$. If $\overline{y}_k^0 = \hat{y}_k^0$ for $k = 1, 2, \cdots, m$, then $\hat{y}^0$ is a non-dominated point. $\hat{y}^0$ is then placed in Ω_r; the set that contains all the equitably non-dominated points in this region. We repeatedly generate new points in this region and check whether every obtained point is non-dominated or not. At every iteration ω, we use the epsilon constraint scalarization to find a new point y^ω. To make sure that the scalarization model provides a new solution, we utilize additional constraints that ensure that the new point is unique and not dominated by already generated points, including the ones generated outside the region.

At every iteration ω of the algorithm (for $\omega > 0$) a unique equitably non-dominated point is generated by solving model the model below (M_n^ω), until it becomes infeasible.

(M_n^ω)

$$Max \;\; y_n + \varepsilon_1 * \sum_{j \neq n} y_j$$

$s.t.$

$$y_k \geq (\hat{y}_k^\tau + 1) h_{\tau k} - BM(1 - h_{\tau k}) \qquad \forall \tau = 0, \cdots, \omega \quad \forall k \neq n \tag{4.5}$$

$$\sum_{k \neq n} h_{\tau k} = 1 \qquad \forall \tau = 0, \cdots, \omega \tag{4.6}$$

$$h_{\tau k} \in \{0, 1\} \qquad \forall \tau = 0, \cdots, \omega \quad \forall k \neq n, \tag{4.7}$$

$$y \in \mathcal{Y}$$

$ineq.$ (4.2), (4.3)

If (M_n^ω) is feasible, the solution found, $\hat{y}^\omega$ is a non-dominated point in this region and it is not identical to any of previously generated points in Ω_r. This is guaranteed by using auxiliary binary variables $h_{\tau k}$ and constraints 4.5 and 4.6, which ensure that the solution is better than the previously found solutions in at least one criterion. Then MD_n^ω is solved to see if $\hat{y}^\omega$ is a non-dominated solution of the original model and if so, it is added to Ω_r.

We implement the algorithm in every region and find the equitably non-dominated points in the regions. The set $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_p$ is the collection of generated equitably non-dominated points. Hence, we obtain subsets of the non-dominated frontier that lie on different parts of the frontier.

In a nutshell, the GEND algorithm is as follows:

Step 1: Fit a hyperplane to the frontier as shown in section 4.1

Step 2: Select p reference points on the fitted surface as shown in section 4.2.

Step 3: Generate non-dominated solutions in the neighbourhood of the selected points. Find the points yrt and yrl and define a region with upper and lower bounds as shown in section 4.2.

Step 4: Generate the equitably non-dominated points in the regions defined in step 3 above.

a: (Initialization). Enumerate the p regions. Select the first region (set $r = 1$) to explore and a criterion, n, to maximize. Set $\omega = 0, \Omega = \emptyset$ and $\Omega_1 = \emptyset$.

b: (Generating a new point). Solve the (M_n^ω) model. If (M_n^ω) is feasible, denote the optimal point as $\hat{y}^\omega \in \Re^m$ and go to step $4c$. Otherwise, go to step $4d$.

c: (Checking for dominance). Solve (MD_n^ω) to check whether $\hat{y}^\omega$ is non-dominated. Let the optimal solution be $\overline{y}^\omega$. If $\overline{y}_k^\omega = \hat{y}_k^\omega$ for $k = 1, 2 \cdots, m$, then $\Omega_r \cup \hat{y}^\omega \rightarrow \Omega_r$. Go to step $4b$.

d: Stop. Ω_r contains all the equitably non-dominated points in region r.

e: If $r = p$, stop, $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_p$. Else, set $r + 1 \longrightarrow r$, set $\omega = 0$, $\Omega_r = \emptyset$ and go to Step $4b$.

In the next section, we use an interactive approach to determine a highly preferred region containing attractive solutions for the DM. This approach is based on the algorithm proposed in [17]. We modify the algorithm so as to work in the cumulative ordered criteria space and obtain equitably efficient solutions to a MOP problem. We find a highly preferred reference point close to the non-dominated frontier in the cumulative ordered criteria space, define a region around it and generate the equitably non-dominated points in the region.

Chapter 5

Algorithm for generating the equitably non-dominated points in the preferred region (The GNDPR Algorithm)

In this approach, the DM chooses a reference point from a surface that is close to the real non-dominated set. The surface is obtained by fitting an L_q function, as in [38]. We interact with the DM to inquire about a preferred region and find the equitable Pareto optimal points lying in the region. This interactive algorithm applies the approach proposed by [17] to E-MOP.

5.1 The L_q function

The L_q function happens to be a very good approximation for the non-dominated set of multi-objective combinatorial optimization (MOCO) problems as shown in [39]. The hyper-surface we fit is an L_q function defined by $\sum_{k=1}^{m}(1 - y'_k)^q = 1$

where $q > 0$ and $y' = (y'_1, y'_2, \cdots, y'_m)$ is the scaled non-dominated point with $y'_k = \dfrac{y_k^{IP} - y_k}{y_k^{IP} - y_k^{NP}}$ such that $0 \leq y'_k \leq 1$ for all $k = 1, \cdots, m$. Here, y^{IP} and y^{NP} represent the ideal and nadir vectors, respectively. Let ND represents the set containig all the equitable Pareto optimal points of model 5, then $y_k^{NP} = min\{y_k : y \in ND\}$. The nadir vector is easily computed in the biobjective case but in general when $m \geq 2$ it is laborious since one has to generate all the points of the Pareto set for a given problem instance. In this study, we generated all the equitable Pareto optimal points for our computational purposes, hence we utilize the real nadir points. However, a practical approach to find exact nadir points for biobjective and triobjective optimization problems was developed in [40]. More recently, an efficient algorithm that finds the nadir points for MOIP problems was developed in [41] and could be used for the related problem.

After scaling the criteria, the points $(0, 0, \cdots, 0)$ and $(1, 1, \cdots, 1)$ correspond to the ideal and nadir points respectively and the transformed problem becomes a minimization problem regardless of the form of the original model. The better output vectors will have smaller values in the scaled output vectors; as the value of y_k approaches the value of y_k^{IP}, y'_k approaches zero.

5.2 Fitting the L_q Surface

It is empirically shown in [39] that a good approximation of the Pareto frontier is obtained by fitting the L_q surface using a middle-most non-dominated point. We follow the same idea. We find a centrally located point in a Tchebycheff distance manner. In our problem, we find the central point yc by solving M_{ch} below.

$$
\begin{aligned}
&(M_{ch})\\
&Min\ \ \alpha - \varepsilon_1 * \sum_{k=1}^{m} y_k\\
&s.t.\ \ \ \alpha \geq y_k^{IP} - y_k \quad \forall k = 1, 2, \cdots, m. \qquad (5.1)\\
&y \in \mathcal{Y}
\end{aligned}
$$

where $y^{IP} \in \Re^m$ is the ideal point whose kth component y_k, is found by solving $max\{y_k : y \in \mathcal{Y}\}$. It is well known that the augmented Tchebycheff program above returns a non-dominated point as shown in [37] (p.420).

Once we find the centrally located point yc, we solve the non-linear equation $\sum_{k=1}^{m}(1 - yc'_k)^q = 1$ for q. This assures that the L_q surface passes through the scaled point $yc' \in \Re^m$, where $yc' = (\frac{y_1^{IP} - yc_1}{y_1^{IP} - y_1^{NP}}, \frac{y_2^{IP} - yc_2}{y_2^{IP} - y_2^{NP}}, \cdots, \frac{y_m^{IP} - yc_m}{y_m^{IP} - y_m^{NP}})$.

5.3 Defining the region and finding the preferred equitably non-dominating points

Having fitted the L_q surface as shown in section 5.2, we define the DM's preferred region. Here, we try to select a preferred point, $yp' \in \Re^m$ on the surface. This point is typically infeasible and we interact with the decision maker to select it.

Most likely, the point yp' on the fitted surface will not be a feasible point, so it is used only as a reference point. We use it to define the preferred region from where we generate the equitably efficient solutions. Similar to the previous algorithm, we find two equitably non-domintaed points ypt' and ypl' that are closest to the scaled ideal point $y'^{IP} = (0, \cdots, 0)$.

ypt' is found by solving the (M'_{chev}) model below:

(M'_{chev})

$$Min\ \rho_{max} + \varepsilon_2 * \sum_{k=1}^{m} y'_k$$

$$s.t.\quad \rho_{max} \geq \lambda_k^{chv} y'_k \qquad \forall k = 1, 2, \cdots, m. \tag{5.2}$$

$$y'_k = (y_k^{IP} - y_k)/(y_k^{IP} - y_k^{NP}) \quad \forall k = 1, 2, \cdots, m. \tag{5.3}$$

$$y \in \mathcal{Y}$$

where ρ_{max} measures the maximum component-wise weighted distance from the ideal point in the direction of the reference point yp'.

The weight vector $\lambda^{chv} \in \Re^m$, is as in section 4.2. Since in our problem the scaled ideal point y'^{IP} is vector of zeros, the weight vector λ^{chv} reduces to the following:

$$\lambda_k^{chv} = \begin{cases} \frac{1}{yp'_k}\left[\sum_{k=1}^{m}\frac{1}{yp'_k}\right]^{-1} & \text{if } \; yp'_k \neq 0 \;\; \forall k \in M \\ 1 & \text{if } \; yp'_k = 0 \\ 0 & \text{if } \; yp'_k \neq 0 \text{ but } \exists j \in M : yp'_j = 0 \end{cases}$$

ypl' is found by solving the (M'_{linr}) model below:

(M'_{linr})

$$Min \sum_{k=1}^{m} \lambda_k^{lnr} y'_k + \varepsilon_2 * \sum_{k=1}^{m} y'_k$$

$s.t.$

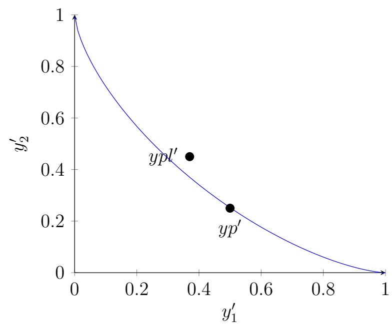
$y \in \mathcal{Y}$

$ineq.$ (5.3)

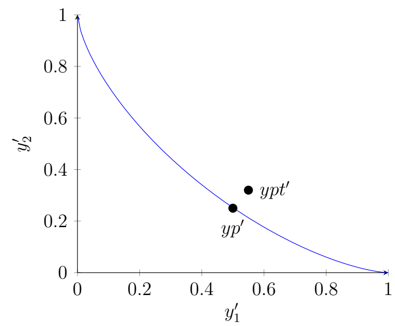
as shown in [37] (page 168) the weight vector $\lambda^{lnr} \in \Re^m$ is formulated as follows:

$$\lambda_k^{lnr} = \frac{(1 - yp'_k)^{q-1}}{\sum_{j=1}^{m}(1 - yp'_j)^{q-1}} \qquad \forall k = 1, 2, \cdots, m \text{ and } q > 1$$

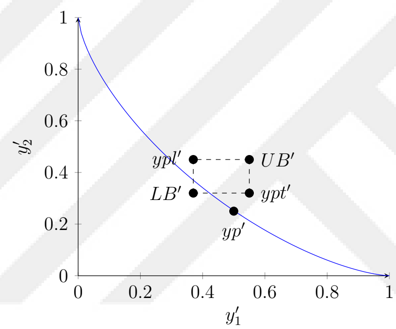
.



(a) Non-dom. point in a weighted Tchebycheff distance manner



(b) Non-dom. point in a weighted linear distance manner



(c) Generated region

Figure 5.1: Generating the preferred region when $q > 1$

Let ypt' and ypl' be the optimal solutions obtained from solving M'_{chev} and M'_{linr} respectively. We determine the region by defining upper and lower bounds. Let the k^{th} element of the scaled upper bound UB'_k be defined as $max(ypt'_k, ypl'_k)$ and that of the scaled lower bound LB'_k be $min(ypt'_k, ypl'_k)$. This is illustrated in Figure 5.1 above.

In the rare case when we have $q \leq 1$, we set the k^{th} element of the upper bound as $UB'_k = ypt'_k$, and the define the region as follows: $y'_k \leq UB'_k$ for $k = 1, 2, \cdots, m$ (see Figure 5.2 below).

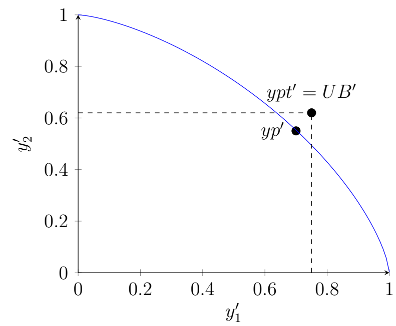


Figure 5.2: Region generated when $q \leq 1$

To find the equitably efficient solutions in this region, we transform the region to cumulative ordered space using $UB_k = y_k^{IP} - UB'_k * (y_k^{IP} - y_k^{NP})$ and $LB_k = y_k^{IP} - LB'_k * (y_k^{IP} - y_k^{NP})$ and obtain the additional constraint $LB_k \leq y_k \leq UB_k$ for $k = 1, 2, \cdots, m$. Now, we have a region similar to the ones generated in section 4.2. Hence, we can directly apply the algorithm in section 4.3 and generate the highly preferred equitably efficient solutions.

In a nutshell, the GNDPR algorithm is as follows:

Step 1: Find the y^{IP} and y^{NP} of the E-MOP.

Step 2: Find a centrally located non-dominated point (yc) by solving the equal-weighted Tchebycheff distance model, (M_{ch}) and scale it as shown in section 5.2.

Step 3: Fit an L_q function on the scaled centrally located point(yc') and ask the DM to choose a preferred point (yp).

Step 4: Find the non-dominated points ypt' and ypl'. Then define a region with upper bounds and lower bounds as shown in section 5.3

Step 5: Generate the equitably non-dominated points in the region defined in step 4 above.

a: (Initialization).Transform the region defining bounds to the cumulative criteria space and select a criterion, n, to be maximized. Set $\omega = 0, \Omega = \emptyset$.

b: (Generating a new point). Solve the (M_n^ω) model. If (M_n^ω) is feasible, denote the optimal point as $\hat{y}^\omega \in \Re^m$ and go to step 5c. Otherwise, go to step 5d.

c: (Checking for dominance). Solve (MD_n^ω) to check whether $\hat{y}^\omega$ is non-dominated. Let the optimal solution be $\overline{y}^\omega$. If $\overline{y}_k^\omega = \hat{y}_k^\omega$ for $k = 1, 2 \cdots, m$, then $\Omega \cup \hat{y}^\omega \to \Omega$ and go to step 5b. Otherwise, go to step 5b.

d: Stop. Ω contains all the equitably non-dominated points in the preferred region.

Chapter 6

Computational Experiments

In this section, we illustrate the two approaches on equitable multi-objective knapsack problems. The classical multi-objective binary knapsack problem is an extension of the single-objective binary knapsack problem where each item is associated with a vector of outputs instead of a single output value. The single objective knapsack problem is one of the classical problems studied in the operational research literature ([42],[43]). There have been recent studies aiming at developing fast and efficient exact solution algorithms to multi(bi)-objective knapsack problems ([44],[45],[46],[47]).

[44] considers bi-objective binary knapsack problems and proposes a two-phases method in generating the set of efficient solutions of these problems. The supported efficient solutions are generated in the first phase using a weighted sum scalarization approach. The second phase involves finding the unsupported efficient solutions using branch and bound based approaches. [45] studies multi-objective binary knapsack problems and proposes a dynamic programming based approach to generate the efficient solutions of these problems. [46] and [47] present improvements on the dynamic programming based approaches to solving multi-objective binary knapsack problems. They compare their approaches with the existing exact methods in the literature. Also, [46] performed experiments for the tri-objective case.

We will consider multi-objective binary knapsack problems where the decision maker has equity concerns (E-MOBKP). We assume that the preference model of the decision maker satisfies properties related to inequity-aversion and try to find the set of equitably efficient portfolios that result in non-dominated output vectors in the Generalized Lorenz sense.

Consider a setting where there are n project proposals that provide outputs to m entities. Let $M = \{1, 2, \cdots, m\}$ be the set of entities and $N = \{1, 2, \cdots, n\}$ be the set of proposals. Every project i is expected to generate an output value of o_{ij} for entity j and consumes c_i units of resources. Assume that the decision maker would like to select and fund a portfolio of these projects, which results in an equitable distribution of outputs across the m entities. The total amount of available resource is denoted by B, which is generally not enough to initiate all the projects. The decision to be made here, is whether to initiate a project or not, *i.e.* partial funding is not possible. The decision variables are as follows:

$$x_i = \begin{cases} 1, \; if \; project \; i \; is \; initiated \\ 0, otherwise \end{cases}$$

The mathematical model of this problem is given below:

$$Max \text{``} y_1, y_2, \cdots, y_m \text{''}$$

$$s.t. \quad y_k - \left(k * r_k - \sum_{j=1}^{m} d_{kj} \right) = 0 \qquad \forall k \in M \tag{6.1}$$

$$\sum_{i=1}^{n} c_i * x_i \leq B \tag{6.2}$$

$$z_j - \sum_{i=1}^{n} o_{ij} * x_i = 0 \qquad \forall j \in M \tag{6.3}$$

$$r_k - z_j \leq d_{kj} \qquad \forall j, k \in M \tag{6.4}$$

$$d_{kj} \geq 0 \qquad \forall j \in M \tag{6.5}$$

$$x_i \in \{0, 1\} \qquad \forall i \in N \tag{6.6}$$

We performed experiments to see whether the proposed algorithms provide satisfactory results. The experiments were conducted on randomly generated multi-objective knapsack problems with three and five objectives (entities (m)). In these instances, the weights (costs) and the output values are generated randomly using uniform distributions. We generate c_i and o_{ij} in the range $[1, 1000]$.

Different values are used for the total number of items n. For each combination of m and n, 10 problem instances are generated. For every instance, the total budget B is set as $\sum_{i=1}^{n} c_i/2$.

The algorithms are coded in Visual C++ and solved by a computer with an Intel Xeon E5 3.60 GHz processor and 32 GB RAM. The solution times are expressed in central processing unit (CPU) seconds. All mathematical models are solved with CPLEX 12.7.

As stated before, there is a rich literature and very fast solution approaches for the two criteria setting. Unfortunately, this is not the case for problems with more than two criteria. This motivates us to apply our algorithms to the three and five criteria settings. We first performed experiments for the three criteria settings ($m = 3$). We generated the whole equitably non-dominated frontier for these problem instances using the epsilon constraint method [48] in the cumulative ordered space. The computational effort increases significantly as the number of projects (n) increases as shown in Table 6.1 below. Moreover, the average number of solutions found is also increasing. In the case where $n = 200$, we have a problem instance with 2143 solutions. Generating all the points may be computationally inefficient. Moreover, it may not be practical to expect the DM to make comparisons among such a large set of points to find the most preferred one.

In an attempt to remedy this situation, we could put a time limit on the epsilon constraint method, so as not to generate the whole Pareto set. However, this fails to return solutions that represent different sections of the Pareto. To that end, we apply the GEND algorithm.

For the GEND algorithm, we perform experiments for hyperplanes fitted using

Table 6.1: Results of the epsilon-constraint method when $m = 3$

	Solution times(secs)			**No of solutions**		
n	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**
50	0.86	13.66	62.4	5	19.9	58
100	0.33	89.2	257.02	2	44.5	105
150	0.15	180.71	1222.12	1	47.6	205
200	6.25	44535.85	429475.14	9	363.9	2143

the point with maximum total benefit ($\sum_{k=1}^{m} y_k$) as shown in section 4.1. Let the maximum total be p. The hyperplane we fit is of the form $\sum_{k=1}^{m} y_k = p$.

For every problem instance, we implemented the algorithm with a set of five reference points. Due to Proposition 6, the reference points we select on the hyperplanes are restricted to the polyhedron Q and should have a total benefit of p. For the tri-objective case, we have the set $\{yr_1, yr_2, yr_3 \in \Re_+ : yr_1 + yr_2 + yr_3 = p,\ yr_3 \geq 3yr_1,\ 2yr_3 \geq 3yr_2,\ yr_2 \geq 2yr_1\}$. Moreover, we can define each element of the reference point yr_k as a fraction of p, i.e, we define a weight vector $\mu \in \Re^3$ where $yr_k = \mu_k * p$, $\sum_{k=1}^{m} \mu_k = 1$ and $\mu_k \geq 0\ \ \forall k \in M$.

So in $\Re^3$, the problem reduces to that of finding μ values that lie in the polyhedron $Q_\mu = \{\mu \in \Re^3 : \sum_{k=1}^{3} \mu_k = 1,\ \mu_3 \geq 3\mu_1,\ 2\mu_3 \geq 3\mu_2,\ \mu_2 \geq 2\mu_1,\ \mu_k \geq 0\ \forall k \in M\}$. We search the weight space to find weight values lying in the Q_μ polyhedron.

Firstly, we solve $min\,\{s :\ \mu_3 \geq 3\mu_1 + s,\ 2\mu_3 \geq 3\mu_2 + s,\ \mu_2 \geq 2\mu_1 + s,\ \sum_{k=1}^{3} \mu_k = 1,\ \mu_k \geq 0\ \ \forall k \in M\}$ to find a feasible weight vector in Q_μ. Note that we have also tried different versions of this model, e.g. one that maximizes s so as to find a middlemost feasible solution. However, the performances of the resulting weights were inferior to those found by minimizing s. We obtain $\mu^1 = (0.167, 0.333, 0.5)$ which represents the weight that simultaneously maximizes yr_1 and yr_2 in Q. To generate points in other regions, we solve the first model with additional simple bounds on the values of μ_1 and μ_2. The second weight vector $\mu^2 = (0.158, 0.328, 0.514)$ is generated by adding the constraints $\{0.158 \leq \mu_1 \leq 0.16, 0.328 \leq \mu_2 \leq 0.33\}$.

Similarly, we add the constraints $\{0.16 \leq \mu_1 \leq 0.167, 0.328 \leq \mu_2 \leq 0.33\}$ to generate the third weight vector $\mu^3 = (0.16, 0.33, 0.51)$. Two more weights $\mu^4 = (0.1636, 0.3289, 0.5075)$ and $\mu^5 = (0.165, 0.33, 0.505)$ are generated in a similar fashion. Due to the structure of the weights, extreme weights like $(1, 0, 0)$ and $(0, 1, 0)$ will result in the generation of reference points outside Q and hence these weights are not in the feasible weight space Q_μ.

Table 6.2: Performance of the GEND algorithm on instances with $m = 3$

	Solution times(secs)			**No of Solutions found**		
n	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**
50	0.48	1.35	2.11	2 (25.86%)	7.70 (42.93.3%)	15 (68.75%)
100	0.43	10.36	49.25	1 (19.05%)	23.70 (47.06%)	92 (87.62%)
150	0.70	34.29	233.74	1 (4.65%)	21.40 (35.98%)	121 (100%)
200	1.54	767.94	4261.84	1 (3.57%)	113.80 (29%)	446 (72.52%)

Table 6.2 reports the performance of the GEND algorithm on instances with three entities. The solution times and number of equitably non-dominated solutions found are reported. We also report what percentage of the whole equitably non-dominated set is found by the GEND algorithm. The minimum (Min), average (Avg) and maximum (Max) of CPU times and solutions found are taken over the ten random instances generated for every value of n. We also calculated the percentages of solutions found for every problem instance solved and report the minimum (Min), average (Avg) and maximum (Max) of these percentages over the 10 problem instances for every n value (the percentages in the bracket). For $n = 200$, the average solution time is very high due to the problem instance with 2143 solutions. Removing this instance will reduce the average solution times to 1764.82 seconds and 514.10 seconds for the epsilon constraint method and the GEND algorithm respectively. We can observe that on average we generate a significant portion of the equitably non-dominated set in a fraction of the average time it takes to generate the whole set for different n values given in Table 6.1.

We also report the average time it takes to generate a non-dominated point in

Table 6.3 below. We compare the average time it takes per solution in the GEND algorithm and the epsilon constraint method. As we can see, it takes considerably less time to generate a solution in the GEND algorithm compared to the epsilon constraint method for all values of n.

Table 6.3: Average time to generate a solution(in seconds)

n	Average time per solution (GEND)	Average of time per solution (whole Pareto)
50	0.20	0.48
100	0.44	1.35
150	0.96	1.88
200	2.95	26.01

To check the spread of the solutions, we divided the cumulative criteria space into 125 boxes of equal dimensions. As expected, only some of the boxes contain real equitably non-dominated points. For these boxes, we report in Table 6.4, the minimum (Min), average (Average) and maximum (Max) percentage of the solutions found by the GEND algorithm. In a similar fashion, we report on the percentage of the boxes that at least one solution is found by the GEND algorithm. As seen in Table 6.4, the algorithm is able to find representative solutions in 40% to 50% of the non empty boxes on average. Moreover, in each box covered, a satisfactory percentage of the solutions in that box is generated by the GEND algorithm. Also, the percentages tend to drop as the problem gets larger (from 40.06% to 26.41%)

Table 6.4: Spread of solutions

	% of solutions in boxes			% of boxes that contain solutions		
n	Min	Avg	Max	Min	Avg	Max
50	16.67	40.06	61.54	16.67	42.97	61.54
100	18.46	44.50	78.52	23.08	51.58	81.82
150	2.63	32.29	100	10.53	41.18	100
200	1.91	26.41	65.63	5.56	39.51	81.82

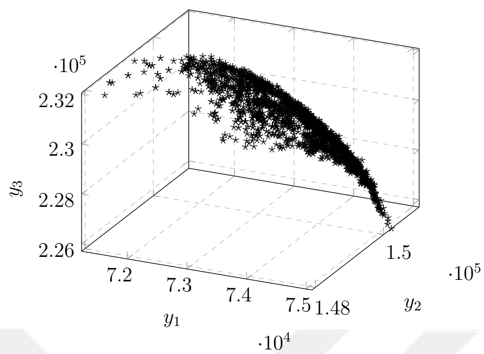
Also, we performed initial experiments with the hyper-surface fitted using a centrally located equitably efficient point. We observe that both strategies perform similar in terms of the solution time and the number of solutions found. This is due to the observation that the two surfaces are very close to each other.

In an attempt to converge to a single or a few most preferred points given preference information from the DM, we made experiments for the GNDPR algorithm. We randomly select the preferred point on the L_q function but the DM's responses could also be simulated using different value functions like weighted linear, Euclidean, or Tchebycheff functions to find a point on the L_q that is closest to the scaled ideal point $(0, 0, \cdots, 0)$. In practice, to determine such a preferred point, one could select a set of points that cover different regions of the L_q surface and ask the DM to select the best point among them. Table 6.5 below reports the performance of the GNDPR algorithm on the same set of problem instances used for the GEND and epsilon constraint algorithms. We report the average solution times, average number of solutions found and average time per solution for the different n values used. The percentages indicate these values as percentages of the solution time and the number of solutions in the whole equitably non-dominated set.

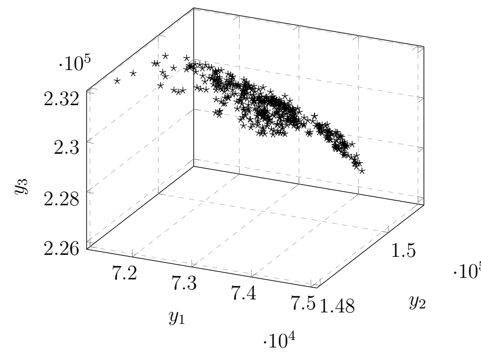
Table 6.5: Performance of the GNDPR algorithm on instances with $m = 3$

n	Avg solution time	Avg number of solutions found	Avg time per solution
50	1.05 (34.45%)	2.40 (15.58%)	0.53
100	0.61 (24.94%)	1.60 (11.30%)	0.42
150	1.01 (19.87%)	2.20 (16.13%)	0.49
200	12.91 (1.29%)	11.40 (4.36%)	0.68

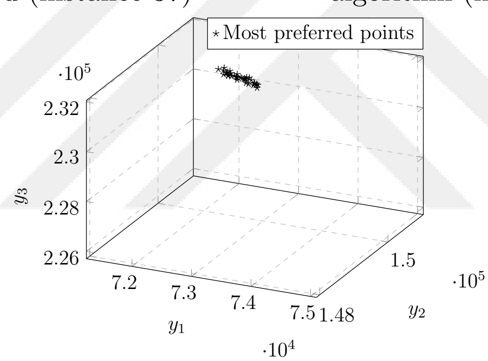
Figure 6.1 shows the solutions returned by the epsilon-constraint method, the GEND algorithm and the GNDPR algorithm for an example case (this is the instance with the highest solution time). As seen in Figure 6.1b the GEND algorithm returns solutions that represent different sections of the frontier. Also, Figure 6.1c shows the most preferred solutions of the DM returned by the GNDPR algorithm. In Figure 6.2 another example is shown. It is seen that the GEND algorithm performs quite well in terms of the representativeness. It actually finds more than 50% of the solutions in around 20% of the time it takes to generate the whole set.



(a) Equitably non-dominated points found using the epsilon-constraint method (instance 37)

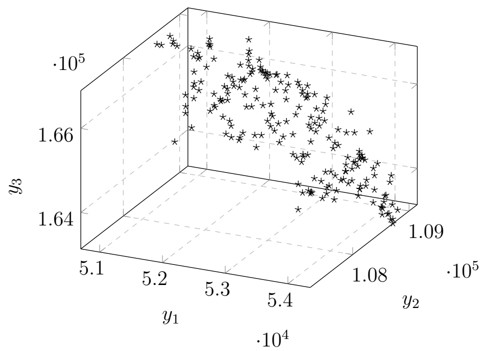


(b) Equitably non-dominated points found using the GEND algorithm (instance 37)

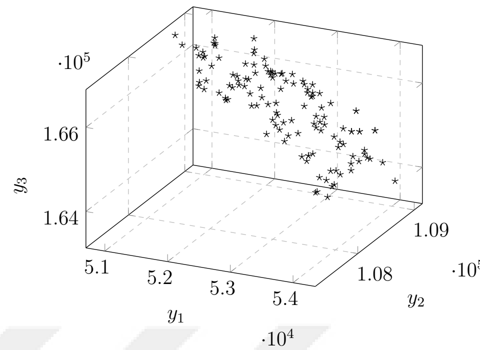


(c) Equitably non-dominated points found using the GNDPR algorithm (instance 37)

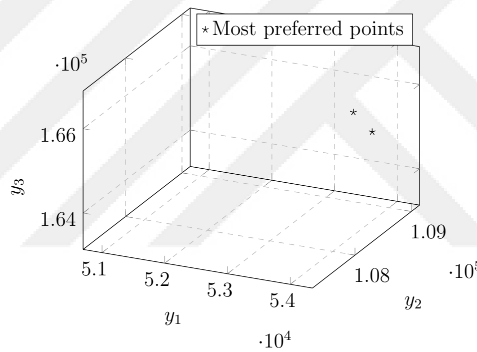
Figure 6.1: Equitably non-dominated points for an example case



(a) Equitably non-dominated points found using the epsilon-constraint method



(b) Equitably non-dominated points found using the GEND algorithm



(c) Equitably non-dominated points found using the GNDPR algorithm

Figure 6.2: Equitably non-dominated points for an example case

We also performed experiments for the five criteria setting. We use three reference points. We use the weights $(0, 0, 0, 0.22, 0.78)$, $(0, 0, 0.33, 0.25, 0.42)$ and $(0.05, 0.1, 0.15, 0.2, 0.5)$ generated in a similar fashion to that of the tri-objective case. Table 6.6 below reports the performance of this algorithm on the instances with five criteria. The minimum (Min), average (Avg), and maximum (Max) of computational times and number of solutions are reported.

Table 6.6: Performance of the GEND algorithm on instances with $m = 5$

	Solution times(secs)			No of Solutions found		
n	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**
50	0.27	1.05	2.96	2	3.5	6
100	0.69	17.29	112.01	3	19.6	87
150	1.61	740.36	6223.67	3	57.8	209
200	6.28	2177.34	16122.66	5	53.5	223

For the GNDPR algorithm, Table 6.7 below reports the average solution times, average number of solutions found and average time per solution. Since this algorithm only focuses on a single region around a selected point, the number of solutions found and solution times are smaller than those of the GEND algorithm.

Table 6.7: Performance of the GNDPR algorithm on instances with $m = 5$

n	**Avg solution time**	**Avg number of solutions found**	**Avg time per solution**
50	0.91	1.2	0.78
100	1.10	1.8	0.66
150	3.50	3.8	0.86
200	7.96	5.3	1.29

Note that even the single objective binary knapsack problem is NP-hard. Hence generating the equitable non-dominated frontier of the multi-objective version is computationally challenging, as seen in Table 6.1. We observed that the GEND algorithm generates a subset of the frontier in reasonable CPU time. The points generated are not concentrated in a specific area of the cumulative ordered criteria space, but rather located in different regions of this space. We generate a good representative subset in only a fraction of the time it takes to generate the whole frontier.

As seen in Tables 6.5 and 6.7, the number of solutions generated by the

GNDPR algorithm is relatively fewer than those of the GEND and the epsilon-constraint algorithms. This considerably reduces the computational effort. However, this method relies on preference information taken from the DM. For $m \geq 3$, the L_q function used in approximating the frontier may be computationally difficult to handle and visualize. This might make defining the preferred region a difficult task for the DM.

Chapter 7

Conclusion

We consider multi objective optimization problems where the decision maker is inequity averse, hence she is interested in finding equitably efficient solutions. These solutions are non-dominated in the Generalized Lorenz sense, and called equitably non-dominated. We discuss two solution approaches that differ in terms of the timing of preference articulation.

In the first approach, we aim to generate an evenly spread subset of the set of equitably non-dominated solutions to be presented to the DM for further consideration. We analyse the cumulative criteria space and fit a simple function close to the Pareto in the cumulative ordered criteria space. We then select reference points on the fitted function and generate regions around these points. Finally, we find the equitably efficient solutions in these regions. To the best of our knowledge, this is the first study proposing such an approach for equitable multi-objective programming problems.

The second approach is an interactive approach that relies on input from the DM during the solution process. We extend a recent approach proposed for finding non-dominated points in a "preferred" region such that equitably non-dominated points are found. This extension is not straightforward since we work on the cumulative ordered space. We discuss results on choosing appropriate reference vectors for such problems.

We illustrate the computational feasibility of the algorithms on equitable knapsack problems, that fund projects in different categories subject to a limited budget. Such problems are especially relevant in public service provision as categories may correspond to various population groups benefiting from the service. The experiments demonstrate that the proposed algorithm returns a well-spread set of solutions in reasonable time. The interactive algorithm is also computationally very efficient as it benefits from the information provided by the DM and generates a small number of solutions in the preferred region.

This study can be extended by working on developing faster algorithms for generating non-dominated points in the defined regions for larger problem instances (in terms of m and n) in reasonable time. We can also investigate the application of some evolutionary and meta-heuristic approaches in approximating the equitably non-dominated frontier and generating diverse solutions. Also, another version of the GNDPR could be developed, where an easier-to-handle function than the L_q function is used to approximate the equitably non-dominated frontier.

Bibliography

[1] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

[2] E. L. Ulungu and J. Teghem, "Multi-objective combinatorial optimization problems: A survey," *Journal of Multi-Criteria Decision Analysis*, vol. 3, no. 2, pp. 83–104, 1994.

[3] D. White, "A bibliography on the applications of mathematical programming multiple-objective methods," *Journal of the Operational Research Society*, pp. 669–691, 1990.

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

[5] B. M. Beamon and B. Balcik, "Performance measurement in humanitarian relief chains," *International Journal of Public Sector Management*, vol. 21, no. 1, pp. 4–25, 2008.

[6] P. K. Bergey, C. T. Ragsdale, and M. Hoskote, "A simulated annealing genetic algorithm for the electrical power districting problem," *Annals of Operations Research*, vol. 121, pp. 33–55, Jul 2003.

[7] M. M. Kostreva and W. Ogryczak, "Linear optimization with multiple equitable criteria," *RAIRO-Operations Research*, vol. 33, no. 3, pp. 275–297, 1999.

[8] M. M. Kostreva, W. Ogryczak, and A. Wierzbicki, "Equitable aggregations and multiple criteria analysis," *European Journal of Operational Research*, vol. 158, no. 2, pp. 362–377, 2004.

[9] D. Baatar and M. M. Wiecek, "Advancing equitability in multiobjective programming," *Computers & Mathematics with Applications*, vol. 52, no. 1-2, pp. 225–234, 2006.

[10] Ö. Karsu and A. Morton, "Incorporating balance concerns in resource allocation decisions: A bi-criteria modelling approach," *Omega*, vol. 44, no. Supplement C, pp. 70 – 82, 2014.

[11] M. Ehrgott and X. Gandibleux, "A survey and annotated bibliography of multiobjective combinatorial optimization," *OR-Spektrum*, vol. 22, no. 4, pp. 425–460, 2000.

[12] C.-L. Hwang and A. S. M. Masud, *Multiple objective decision making—methods and applications: a state-of-the-art survey*, vol. 164. Springer Science & Business Media, 2012.

[13] M. J. Alves and J. Clímaco, "A review of interactive methods for multiobjective integer and mixed-integer programming," *European Journal of Operational Research*, vol. 180, no. 1, pp. 99–115, 2007.

[14] R. Ramesh, M. H. Karwan, and S. Zionts, "An interactive method for bicriteria integer programming," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 395–403, 1990.

[15] W. S. Shin and D. B. Allen, "An interactive paired comparison method for bicriterion integer programming," *Naval Research Logistics (NRL)*, vol. 41, no. 3, pp. 423–434, 1994.

[16] M. Köksalan and E. Karasakal, "An interactive approach for multiobjective decision making," *Journal of the Operational Research Society*, vol. 57, no. 5, pp. 532–540, 2006.

[17] B. Lokman and M. Köksalan, "Finding highly preferred points for multi-objective integer programs," *IIE Transactions*, vol. 46, no. 11, pp. 1181–1195, 2014.

[18] G. Mavrotas and D. Diakoulaki, "A branch and bound algorithm for mixed zero-one multiple objective linear programming," *European Journal of Operational Research*, vol. 107, no. 3, pp. 530–541, 1998.

[19] J. Sylva and A. Crema, "A method for finding the set of non-dominated vectors for multiple objective integer linear programs," *European Journal of Operational Research*, vol. 158, no. 1, pp. 46–55, 2004.

[20] M. Özlen and M. Azizoğlu, "Multi-objective integer programming: a general approach for generating all non-dominated solutions," *European Journal of Operational Research*, vol. 199, no. 1, pp. 25–35, 2009.

[21] M. Ozlen, B. A. Burton, and C. A. MacRae, "Multi-objective integer programming: An improved recursive algorithm," *Journal of Optimization Theory and Applications*, vol. 160, no. 2, pp. 470–482, 2014.

[22] B. Lokman and M. Köksalan, "Finding all nondominated points of multi-objective integer programs," *Journal of Global Optimization*, vol. 57, no. 2, pp. 347–365, 2013.

[23] G. Kirlik and S. Sayın, "A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems," *European Journal of Operational Research*, vol. 232, no. 3, pp. 479–488, 2014.

[24] G. Mavrotas and K. Florios, "An improved version of the augmented ε-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems," *Applied Mathematics and Computation*, vol. 219, no. 18, pp. 9652–9669, 2013.

[25] W. Zhang and M. Reimann, "A simple augmentedε-constraint method for multi-objective mathematical integer programming problems," *European Journal of Operational Research*, vol. 234, no. 1, pp. 15–24, 2014.

[26] W. J. Gutjahr and S. Fischer, "Equity and deprivation costs in humanitarian logistics," *European Journal of Operational Research*, vol. 270, no. 1, pp. 185–197, 2018.

[27] N. Al Theeb and C. Murray, "Vehicle routing and resource distribution in postdisaster humanitarian relief operations," *International Transactions in Operational Research*, vol. 24, no. 6, pp. 1253–1284, 2017.

[28] X. Li, R. Batta, and C. Kwon, "Effective and equitable supply of gasoline to impacted areas in the aftermath of a natural disaster," *Socio-Economic Planning Sciences*, vol. 57, pp. 25–34, 2017.

[29] I. S. Orgut, J. S. Ivy, R. Uzsoy, and C. Hale, "Robust optimization approaches for the equitable and effective distribution of donated food," *European Journal of Operational Research*, vol. 269, no. 2, pp. 516–531, 2018.

[30] 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.

[31] W. Zhang, K. Cao, S. Liu, and B. Huang, "A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as hong kong," *Computers, Environment and Urban Systems*, vol. 59, pp. 220–230, 2016.

[32] 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.

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

[34] W. Ogryczak and T. Śliwiński, "On solving linear programs with the ordered weighted averaging objective," *European Journal of Operational Research*, vol. 148, no. 1, pp. 80 – 91, 2003.

[35] A. Shorrocks, "Ranking income distributions," *Economica*, vol. 50, pp. 3–17, Feb. 1983.

[36] Ö. Karsu, A. Morton, and N. Argyris, "Capturing preferences for inequality aversion in decision support," *European Journal of Operational Research*, vol. 264, no. 2, pp. 686–706, 2018.

[37] R. Steuer, "Multiple criteria optimization: theory, computation, and application.," *Willey, New York*, 1986.

[38] E. Karasakal and M. Köksalan, "Generating a representative subset of the nondominated frontier in multiple criteria decision making," *Operations research*, vol. 57, no. 1, pp. 187–199, 2009.

[39] M. Köksalan and B. Lokman, "Approximating the nondominated frontiers of multi-objective combinatorial optimization problems," *Naval Research Logistics (NRL)*, vol. 56, no. 2, pp. 191–198, 2009.

[40] M. Ehrgott and D. Tenfelde-Podehl, "Computation of ideal and nadir values and implications for their use in mcdm methods," *European Journal of Operational Research*, vol. 151, no. 1, pp. 119–139, 2003.

[41] M. Köksalan and B. Lokman, "Finding nadir points in multi-objective integer programs," *Journal of Global Optimization*, vol. 62, no. 1, pp. 55–77, 2015.

[42] M. Silvano and T. Paolo, "Knapsack problems: algorithms and computer implementations," 1990.

[43] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems. 2004," 2003.

[44] M. Visée, J. Teghem, M. Pirlot, and E. L. Ulungu, "Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem," *Journal of Global Optimization*, vol. 12, no. 2, pp. 139–155, 1998.

[45] K. Klamroth and M. M. Wiecek, "Dynamic programming approaches to the multiple criteria knapsack problem," *Naval Research Logistics*, vol. 47, no. 1, pp. 57–76, 2000.

[46] 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.

[47] J. R. Figueira, L. Paquete, M. Simões, and D. Vanderpooten, "Algorithmic improvements on dynamic programming for the bi-objective {0, 1} knapsack problem," *Computational Optimization and Applications*, vol. 56, no. 1, pp. 97–111, 2013.

[48] M. Laumanns, L. Thiele, and E. Zitzler, "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method," *European Journal of Operational Research*, vol. 169, no. 3, pp. 932–942, 2006.

Appendix A

Nomenclature

y	A vector in the cumulative ordered space
y'	Scaled cumulative ordered vector
y^{IP}	Ideal point in the cumulative ordered space
y^{NP}	Anti-ideal (Nadir) point in the cumulative ordered space
yc	Centrally located y using equal-weighted Tchebycheff program
yp'	Highly preferred point on the L_q function
yr	Reference point on the hyperplane
$ypt', (yrt)$	Closest non-dominated point from y'^{IP} (y^{IP}) in a Tchebycheff distance manner
$ypl', (yrl)$	Closest non-dominated point from y'^{IP} (y^{IP}) in a linear distance manner
$\hat{y}$	A non-dominated point in the searched region
$\overline{y}$	A non-dominated point for the original problem

Appendix B

Proof of Proposition 6

We know that for $M = \{1, 2, \cdots, m\}$:
$y_1 = \overrightarrow{z}_1$
$y_2 = \overrightarrow{z}_1 + \overrightarrow{z}_2$
$\vdots$
$y_m = \overrightarrow{z}_1 + \overrightarrow{z}_2 + \cdots + \overrightarrow{z}_m$
and $\overrightarrow{z}_1 \leq \overrightarrow{z}_2 \leq \cdots \leq \overrightarrow{z}_m$
Lets prove Proposition 6 by induction. The base case is at $m = 2$ where $M = \{1, 2\}$. To show that Proposition 6 holds for the base case, we need to show that $y_2 \geq 2y_1$.
Since by definition $y_1 = \overrightarrow{z}_1$ and $y_2 = \overrightarrow{z}_1 + \overrightarrow{z}_2$ where $\overrightarrow{z}_2 \geq \overrightarrow{z}_1$, then $y_2 = \overrightarrow{z}_1 + \overrightarrow{z}_1 + \epsilon \geq 2\overrightarrow{z}_1 = 2y_1$ where $\epsilon \geq 0$. Hence $y_2 \geq 2y_1$.
Hypothesis 1: Lets assume that Proposition 6 holds for $m = s$.
To complete the proof, we need to show that Proposition 6 holds for $m = s + 1$. Due to Hypothesis 1, we just have to show that Proposition 6 holds for the following combinations of (j, k): $(1, s + 1), (2, s + 1), \cdots, (s, s + 1)$. We can show this by induction too.
First we show that $y_{s+1} \geq (s+1)y_1$. Due to Hypothesis 1, we know that $y_s \geq sy_1$. Since $\overrightarrow{z}_{s+1} \geq \overrightarrow{z}_1 = y_1$, then $y_s + \overrightarrow{z}_{s+1} \geq sy_1 + \overrightarrow{z}_1 = sy_1 + y_1 = (s+1)y_1$. Hence $y_{s+1} \geq (s + 1)y_1$.
Hypothesis 2: Lets assume that $(s - 1)y_{s+1} \geq (s + 1)y_{s-1}$.

We know that $\overrightarrow{z}_{s+1} \geq \overrightarrow{z}_s$.
Then $\overrightarrow{z}_{s+1} = y_{s+1} - y_s \geq \overrightarrow{z}_s = y_s - y_{s-1}$,
$\Rightarrow y_{s+1} \geq 2y_s - y_{s-1}$, multiply both sides by $s \geq 0$
$\Rightarrow sy_{s+1} \geq 2sy_s - sy_{s-1} \Rightarrow sy_{s+1} \geq (s+1)y_s + \underbrace{(s-1)y_s - sy_{s-1}}_{\geq 0,\text{ Due to Hypothesis 1}}$.
Hence $sy_{s+1} \geq (s+1)y_s$. Therefore, Proposition 6 holds in any space $\Re^m$.