MULTIPLE CRITERIA PROJECT SELECTION PROBLEMS

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

BY

MUSA ÇAĞLAR

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

SEPTEMBER 2009

Approval of the thesis:

MULTIPLE CRITERIA PROJECT SELECTION PROBLEMS

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

Prof. Dr. Canan Özgen _____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Nur Evin Özdemirel _____________________
Head of Department, **Industrial Engineering**

Assoc. Prof. Dr. Esra Karasakal _____________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Assist. Prof. Dr. Z. Müge Avşar _____________________
Industrial Engineering Dept., METU

Assoc. Prof. Dr. Esra Karasakal _____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Banu Yüksel Özkaya _____________________
Industrial Engineering Dept., Hacettepe University

Assist. Prof. Dr. Seçil Savaşaneril _____________________
Industrial Engineering Dept., METU

Dr. Ahmet Reha Botsalı _____________________
ARDEB, TUBITAK

Date: 11.09.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: **Musa ÇAĞLAR**

Signature:

ABSTRACT

MULTIPLE CRITERIA PROJECT SELECTION PROBLEMS

Çağlar, Musa
M.S., Department of Industrial Engineering
Supervisor: Assoc. Prof. Esra Karasakal

September 2009, 88 pages

In this study, we propose two biobjective mathematical models based on PROMETHEE V method for project selection problems. We develop an interactive approach (ib-PROMETHEE V) including data mining techniques to solve the first proposed mathematical model. For the second model, we propose NSGA-II with constraint handling method. We also develop a Preference Based Interactive Multiobjective Genetic Algorithm (IMGA) to solve the second proposed mathematical model. We test the performance of NSGA-II with constraint handling method and IMGA on randomly generated test problems.

Keywords: Project Selection Problem, PROMETHEE V, data mining, Preference Based Multiobjective Genetic Algorithm, Interactive Approach

ÖZ

ÇOK KRİTERLİ PROJE SEÇİMİ PROBLEMLERİ

Çağlar, Musa
Yüksek lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç Dr. Esra Karasakal

Eylül 2009, 88 sayfa

Bu çalışmada, proje seçimi problemleri için PROMETHEE V yöntemini temel alan iki amaçlı iki farklı matematiksel model önerilmiştir. İlk önerilen matematiksel modeli çözmek için veri madenciliği tekniklerini içeren etkileşimli bir yaklaşım (ib-PROMETHEE) geliştirilmiştir. İkinci önerilen matematiksel modele ilk olarak kısıtları dikkate alan NSGA-II yöntemi uygulanmıştır. Ayrıca, ikinci matematiksel modeli çözmek için Tercihe Dayalı Çok Amaçlı Genetik Algoritma (IMGA) geliştirilmiştir. Kısıtları dikkate alan NSGA-II yöntemi ve IMGA'nın rassal üretilmiş test problemleri üzerinde performansı test edilmiştir.

Anahtar Kelimeler: Proje Seçimi Problemi, PROMETHEE V, veri madenciliği, Tercihe Dayalı Çok Amaçlı Genetik Algoritma, Etkileşimli Yaklaşım

To My Precious Family and My Wife Özden

ACKNOWLEDGEMENTS

I would like to express profound gratitude to my supervisor Assoc. Prof. Dr. Esra Karasakal for her invaluable support, encouragement, supervision and useful suggestions throughout this study. Her moral support and continuous guidance enabled me to complete my work successfully.

I am also highly thankful to my friends for their valuable suggestions throughout this study. I am as ever, especially indebted to my parents for their love and support throughout my life. I also wish to thank my wife Özden for her unfailing emotional support and understanding during my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In project selection problems, a set of projects is selected among large number of competing projects. Projects are evaluated according to some predefined criteria and considering the budget of each project, projects are selected. PROMETHEE V is one of the methods used in project selection problems. In PROMETHEE V, a single objective 0-1 integer programming model is solved to select the best subset of projects. The objective function is to maximize total net flow of the selected projects, which is a combined measure of superiority (positive flow) and inferiority (negative flow) between projects.

We propose two biobjective mathematical models based on PROMETHEE V method for project selection problems. In the first mathematical model, first objective corresponds to the maximization of the superiorority, and second objective corresponds to the minimization of the inferiority. We also develop an interactive approach (ib-PROMETHEE V) to solve the proposed model. ib-PROMETHEE V approach uses ε-constraint method to generate Pareto front of the model. This approach also incorporates data mining techniques by using SPSS Clementine software for post-optimality analysis.

In the second mathematical model, first objective corresponds to maximization of the total net flow, and second objective corresponds to the minimization of the cumulative budget of selected projects. We apply one of the well known heuristics NSGA-II to this model with constraint tournament mechanism. We also develop an interactive preference based multiobjective genetic algorithm (IMGA) to solve this model. We both test and compare the performance of NSGA-II with constraint tournament method and IMGA.

The thesis is organized as follows. In Chapter 2, we summarize the literature on PROMETHEE V method and preference based multi objective genetic algorithms. In Chapter 3, we give information on PROMETHEE V method, ε-constraint method, data mining classification and regression trees (CRT), genetic algorithms, and NSGA-II with constraint tournament method. In Chapter 4, we present the first proposed mathematical model and ib-PROMETHEE V approach. In Chapter 5, we illustrate ib-PROMETHEE V approach on example problems. In Chapter 6, we describe the second proposed mathematical model and IMGA. In Chapter 7, computational experiments of NSGA-II with constraint tournament method and IMGA are presented. In Chapter 8, conclusion and further research are discussed.

CHAPTER 2

LITERATURE SURVEY

2.1 PROMETHEE V METHOD

In the last three decades, different kinds of methods and techniques have been developed in order to aid the decision maker (DM) in multiple criteria decision making (MCDM) problems. MCDM problems can be classified as Multi-Attribute Decision Making (MADM), and Multi-Objective Decision Making (MODM) problems. MADM deals with discrete MCDM problems, whereas MODM is concerned with continuous MCDM problems.

One of the commonly used methods in the field of MADM is Preference Ranking Organization METHod for Enrichment Evaluations (PROMETHEE). PROMETHEE outranking method was introduced by Brans et al. (1985). Two kinds of preorders are generated: partial ranking in PROMETHEE I and complete ranking in PROMETHEE II. PROMETHEE V was introduced by Brans and Mareschal (1992) due to the fact that only ranking was not adequate in most of the problems, there were also some constraints to be considered. In PROMETHEE V, scores calculated in PROMETHEE II (net flows) are used in the single maximization objective, and some constraints are defined to represent the limitations of the problem and 0-1 integer programming model is solved to select the subset of best alternatives. Brans and Mareschal (1994) also introduced the PROMCALC & GAIA decision support system.

After the introduction of PROMETHEE V, studies using this method began to appear in the literature. Abu-Taleb and Mareschal (1995) solved a water resources planning problem in the Middle East with PROMETHEE V method; they selected the best subset of water resource designate options that supports the national policies. Al-Kloub et al. (1997) used PROMETHEE V in order to select the water projects in Jordan. Al-Shemmeri et al. (1997) developed a computer aided decision support system with PROMETHEE V for water strategic planning in Jordan. Al-Rashdan et al. (1999) selected the environmental (wastewater) projects in Jordan with PROMETHEE V method.

Pandey and Kengpol (1995) used the PROMETHEE V method in order to select the automated inspection devices for flexible manufacturing system. Mavrotas et al. (2006) solved the project selection problem under policy restrictions. They revised PROMETHEE V method in order to prevent the selection of relatively low rank projects with small budgets. Morais and Almeida (2007) selected the feasible options for leak management system with PROMETHEE V by group decision making.

2.2 MULTIOBJECTIVE GENETIC ALGORITHMS (MOGAs)

Multi-objective genetic algorithms (MOGAs) are used to solve MCDM problems. They have been so popular in the last decade. Ishibuchi et al. (2008) reviewed the MOGA methods.

Deb (2001) classified MOGAs into two parts: elitist and non-elitist MOGAs. In this classification, "elitism" refers keeping and transferring the good solutions to the following iterations. Thereby, better solutions can be produced and elitist MOGAs make less effort when compared to non-elitist MOGAs. There are many MOGAs proposed in the literature, however NSGA-II (Deb et al., 2000) and SPEA2 (Zitzler et al., 2001) are the most popular elitist MOGAs. One of the reasons of their popularity is that they deal with successfully both MADM and MODM problems.

There are also studies on incorporating preference information into MOGAs. Coello (2000) reviewed how the preferences were handled in GAs. Rachmawati and

Srinivasan (2006) reviewed the preference incorporation approaches in MOGAs. Basically, preferences of decision maker (DM) can be articulated in three ways; before the optimization process (A Priori Methods), during the optimization process (Interactive or Progressive Methods), and after the optimization process (A Posteriori Methods).

Branke et al. (2001) integrated the preferences into the MOGA by defining linear maximum and minimum trade-off functions. Phelps and Koksalan (2003) introduced an interactive evolutionary metaheuristic in which pairwise comparisons of the DM are used for incorporating preference information. Zitzler and Kunzli (2004) developed an indicator based MOGA to incorporate preference information. Deb et al. (2006) developed a reference point based MOGA; they combined the NSGA-II with reference point interactive approach. Deb and Kumar (2007) proposed a reference direction based MOGA. Deb and Kumar (2007) integrated NSGA-II with light beam search approach to articulate preferential information. Thiele et al. (2007) suggested a preference based interactive MOGA. Almes and Almeida (2007) proposed a MOGA based on the Tchebycheff scalarizing function. Beume et al. (2007) proposed a MOGA based on a dominated hypervolume. Koksalan and Phelps (2007) developed a MOGA for approximating preference-nondominated solutions. Soylu and Koksalan (2008) suggested favorable weight based evolutionary algorithm in which each solution finds its own weights for a weighed Tchebycheff distance function to get its fitness value. Karahan and Koksalan (2008) developed a territory defining evolutionary algorithm and incorporated preferences of the DM. Pfeiffer et al. (2008) proposed a reference point based MOGA for group decisions.

There are also a few genetic algorithms used for project selection problem in the literature. A MOGA based on NSGA-II was proposed by Medaglia et al. (2007) for project selection with partially funded projects and resource constraints. A Pareto Ant Colony optimization approach for investment projects was proposed by Doerner et al. (2004).

CHAPTER 3

THEORETICAL BACKGROUND

3.1. DEFINITIONS

The generic form of the MCDM problem can be stated as follows:

$$\text{“maximize”} \quad z = f(x)$$
$$s.t \quad x \in X$$

where $f(x) = (f_1(x),......,f_M(x))^T$ is a *M*-vector of objective functions, $f_m(x)$ represents the m^{th} objective function , $x = (x_1,.....,x_u)^T$ is the decision vector, $X \subseteq R^u$ represents the feasible decision space, $z = f(x)$ represents the objective vector, and $Z = f(X)$ represents the feasible objective space (solution space).

Some of the basic definitions of the MCDM are given below:

Definition 1. A decision vector, $x^k \in X$ is efficient if and only if there doesn't exist $x^j \in X$ such that $f_t(x^j) \geq f_t(x^k)$ for all *t* with a strict inequality for at least one *t*. Otherwise x^k is inefficient.

Definition 2. A decision vector, $x^k \in X$ is weakly efficient if and only if there doesn't exist $x^j \in X$ such that $f_t(x^j) > f_t(x^k)$ for all *t*. Otherwise x^k is strictly inefficient.

Definition 3. An objective vector, $z \in Z$ is non-dominated if and only if there doesn't exist $y \in Z$ such that $y_t \geq z_t$ for all *t* with a strict inequality for at least one *t*. Otherwise *z* is said to be dominated.

Definition 4. The set of all efficient solutions is called the efficient frontier or efficient set.

Definition 5 . The set of all non-dominated objective vectors is called non-dominated frontier or non-dominated set. Non-dominated set is also called as "Pareto-optimal set, "Pareto front" or "True Pareto Front" in the literature.

3.2. PROMETHEE I, II and V METHODS

PROMETHEE methods are the outranking methods which use pairwise comparisons of alternatives according to the preference functions and consist of two steps: constructing and exploiting an outranking relation (Brans and Vincke, 1985). The outranking relation is defined for alternative pairs, e.g. alternatives *a* and *b*, and *a* outranks *b* if and only if *a* is at least as good as *b* on the majority of the criteria whereas *a* is not considerably worse than *b* on any of the criteria. (Vincke, 1992).

In these methods, a weight is assigned to each criterion by DM and an outranking degree π (*a, b*) for alternatives *a* and *b* is formulated as follows:

$$\pi\,(a,b) = \sum_{i=1}^{n} w_i P_i(a,b)$$

where $P_i(a,b)$ is the number between 0-1 and calculated according to the preference function of criterion *i,* w_i is the weight of criterion i, *n* is the number of criteria. Preference function $P_i(a,b)$ is defined for each criterion and used to translate the difference between the alternatives in terms of the given criterion into a preference degree ranging from 0 to 1 ($0 \leq P_i(a,b) \leq 1$).

Let $d = v_i(a) - v_i(b)$ and $P(d) = \begin{cases} P_i(a,b) & d \geq 0 \\ P_i(b,a) & d < 0 \end{cases}$

where $v_i(a)$ is the i^{th} criterion value of alternative *a*, $v_i(b)$ is the i^{th} criterion value of alternative *b*, and $P_i(.)$ is the preference function for i^{th} criterion.

The form of the preference function is determined by the type of criteria and thresholds used; six types of criteria and three types of thresholds are defined (Brans and Vincke, 1985). Thresholds are indifference, $\mathrm{q_i}$, preference, $\mathrm{p_i}$ and Gaussian, σ_{i}, and they are applied in order to indicate the impreciseness in the criterion values. The type of the preference functions and necessary thresholds for each function are displayed in Figure 1. Using outranking degree π (*a, b*), leaving and entering flow of each alternative is calculated as follows;

$$\phi_a^+ = \sum_{b \in A} \pi(a,b) \quad , \quad \phi_a^- = \sum_{b \in A} \pi(b,a)$$

where A is the set of alternatives; ϕ_a^+ is the leaving flow of alternative *a*, representing the significance of alternatives outranked by *a*; ϕ_a^- is the entering flow of alternative *a*, representing the significance of alternatives outranking *a*. PROMETHEE methods use leaving and entering flows in order to build an outranking relation.

In PROMETHEE I, the partial ranking is obtained by using following relations: *a* outranks *b* if $\phi_a^+ \geq \phi_b^+$ and $\phi_a^- \leq \phi_b^-$ (one being strict); *a* is indifferent to *b* if $\phi_a^+ = \phi_b^+$ and $\phi_a^- = \phi_b^-$; *a* is incomparable to *b* otherwise. In PROMETHEE II, net flow is calculated by using leaving and entering flows. Net flow of alternative *a* is defined as $\phi_a = \phi_a^+ - \phi_a^-$. The complete ranking resulting from net flow is obtained using the following relations: *a* outranks *b* if $\phi_a > \phi_b$; *a* is indifferent to *b* if $\phi_a = \phi_b$

PROMETHEE V integrates PROMETHEE II with 0-1 integer programming. Mathematical model of PROMETHEE V is as follows:

Maximize $\sum_{l=1}^{u} \phi_l * x_l$ (Maximize Total Net Flows)

s.t.

Segmentation constraints (Policy, budget etc.)

$x_l \in \{0,1\}$

where ϕ_l is the net flow of alternative l, x_l is a binary variable, x_l=1 if alternative l is selected, x_l=0 otherwise.

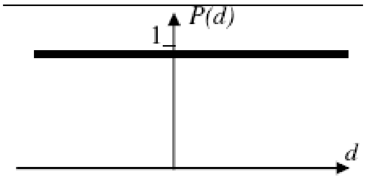
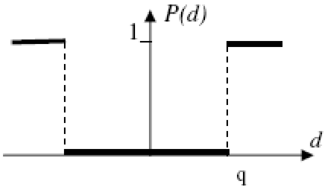
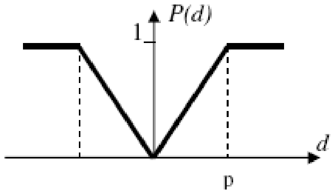
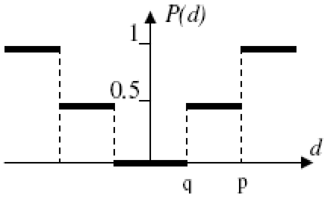
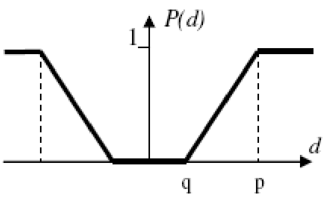
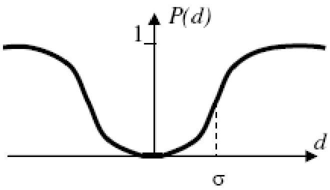
Preference Function Shape	**Properties**
	Type I, Usual Criterion, No threshold is defined.
	Type II, Quasi Criterion, indifference threshold (q) is defined.
	Type III, Linear Preference Criterion, preference threshold (p) is defined.
	Type IV, Level Criterion, both indifference threshold (q) and preference threshold (p) are defined.
	Type V, Linear Preference Criterion with indifference area, both indifference threshold (q) and preference threshold (p) are defined.
	Type VI, Gaussian Criterion, Gaussian threshold (σ) is defined.

Figure 1 *Preference Function Types*

3.3. EPSILON (ε) CONSTRAINT METHOD

"ε-constraint" method was introduced by Haimes et al (1971). It finds the non-dominated set of MCDM problems by converting the MCDM problem to its corresponding single objective problem. The method basically consists of two steps:

1. Find the range of objective functions (construct pay-off table).
2. Convert the MCDM problem into its corresponding "ε-constraint" problem.

Illustration of the method in the two dimensional space for discrete problems is provided below. Consider the following problem:

Maximize $f_1(x)$

Maximize $f_2(x)$

s.t

$x \in X$

x are integer.

An example problem is displayed in Figure 2. To find the range of the objective functions, following two problems are solved independently.

(M_1)

Maximize $f_1(x) + \rho * f_2(x)$

s.t

$x \in X$

(M_2)

Maximize $f_2(x) + \rho * f_1(x)$

s.t

$x \in X$

where X is the feasible region, and ρ is a sufficiently small nonnegative number. Augmentation term $\rho * f_1(x)$ is added to objective function to prevent obtaining weakly efficient solutions. Let A(a_1,0) be the optimal solution to the problem M_1 and B(0,b_2) be the optimal solution to the problem M_2. Range of the objective functions are obtained, they are displayed in Table 1.

Table 1 *Payoff table of objective functions*

Objectives	$f_1(x)$	$f_2(x)$
$f_1(x)$	a_1	0
$f_2(x)$	0	b_2

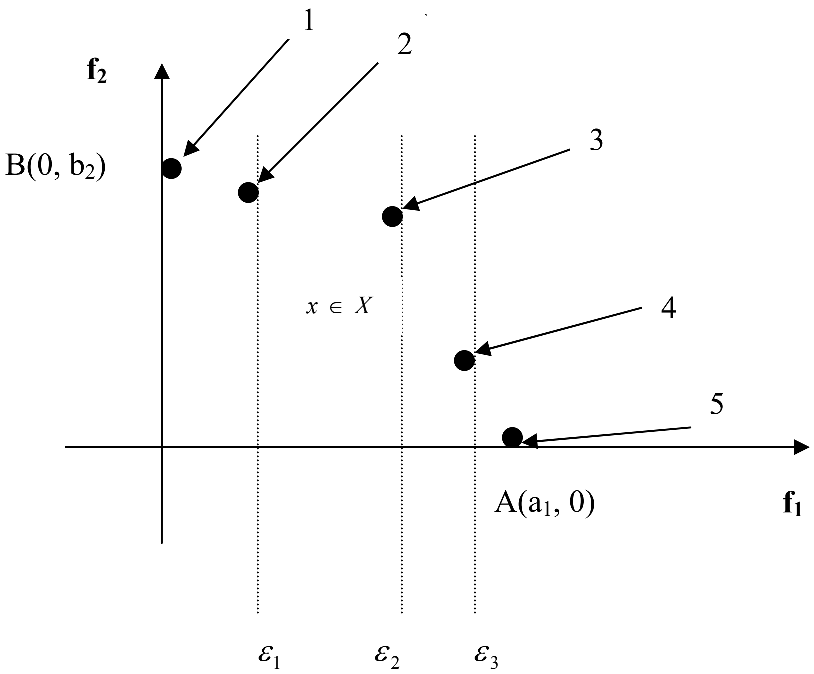


Figure 2 *Graphic representation of ε-constraint problem*

After obtaining the range of objectives, corresponding "ε-constraint" problem is formulated by defining one of the objectives as a constraint.

Maximize $f_2(x) + \rho * f_1(x)$

s.t

$f_1(x) \geq \varepsilon$

$x \in X$

Assume that discrete non-dominated set consists of five solutions (see Figure 2). $f_1(x)$ is between 0 and a_1. First of all, "ε-constraint" problem solved with $\varepsilon = 0$, the non-dominated solution B (arrow 1) is obtained. At the second iteration, ε-constraint is increased with very small number to find the neighboring non-dominated solution, the problem solved with $\varepsilon = 10^{-6}$, and the non-dominated solution pointed by the arrow 2 is obtained (ε_1 is also obtained). At the third iteration, the problem solved with $\varepsilon = \varepsilon_1 + 10^{-6}$, and the non-dominated solution pointed by the arrow 3 is obtained and so on. This method continues until no new efficient solution is generated.

3.4. DATA MINING CLASSIFICATION AND REGRESSION TREES

Classification and Regression Trees (CRT) is a data mining technique which was originally described by the Breiman et al. in 1984. CRT partitions the data into two subsets so that the records within each subset are more homogeneous than in the previous subset. It is a recursive process, each of those two subsets is then split again, and the process repeats until the homogeneity criterion is reached or until some other stopping criterion is satisfied. The mathematical algorithms that work behind the CRT are not in the scope of this study.

To the best of our knowledge, there was an attempt to do a post-optimality analysis using data mining techniques in biobjective traveling salesman problems with profits (Karademir, 2008). The cities and their marginal effects over the Pareto front are analyzed to give supportive information to the DM.

3.5. GENETIC ALGORITHMS (GAs)

Genetic algorithms (GAs) are the optimization methods that are inspired by the principles of natural genetics. Some basic concepts of genetics are artificially embedded into these algorithms. In GAs, a solution is represented by a chromosome that is coded with binary or real bits according the characteristic of the problem under consideration. After the representation of solutions, a fitness value is assigned to each solution by various techniques. Fitness represents the survival of a solution in a population.

In GAs, firstly, an initial population (parent population) is randomly generated and evaluated according to fitness values. Then reproduction (selection) operator is used for eliminating bad solutions and duplicating good solutions. Solutions that survive in the population constitute the mating pool. Binary tournament selection is usually used as a reproduction strategy in the literature. Simply, in binary tournament selection, two solutions are randomly picked from the population and the better one is selected and placed into the mating pool. New solutions (offspring population) are created from mating pool by performing crossover (recombination) and mutation (injection) operators. In crossover operator, two solutions are randomly taken from the mating pool and some bits of the solutions are exchanged between themselves by using different techniques such as single-point crossover and two-point crossover. In mutation operator, some bits of the solutions are changed with a predefined probability.

Reproduction operator favors the good solutions in the population. However, crossover and mutation operators create new and diverse solutions. Once the offspring population is created, its solutions are evaluated according to the fitness values and solutions of the parent population are updated. This generic loop goes on until a stopping condition is achieved. In Figure 3, working principle of a generic GA is depicted.

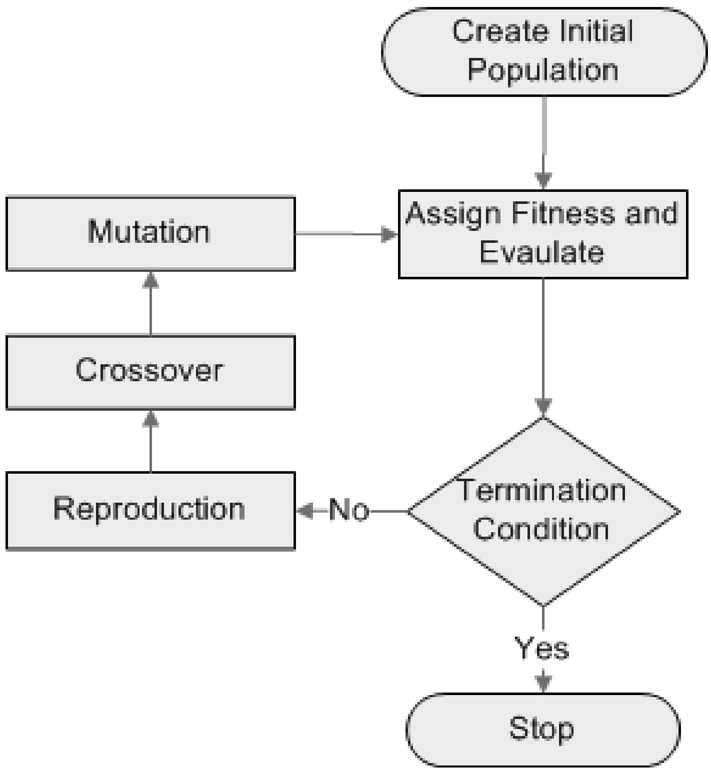


Figure 3 *Working Principle of a Generic GA*

3.6. ELITIST NON-DOMINATED SORTING GENETIC ALGORITHM (NSGA-II) WITH A CONSTRAINT HANDLING MECHANISM

Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) was developed by Deb et al. (2000). They also proposed constraint tournament method that can be embedded into genetic algorithms for constraint handling purposes. In fact, in some problems, constraint handling can be somehow performed with adhoc problem-specific approaches. However, structured constraint handling methods are usually compulsory for most of the constrained MCDM problems.

In this study, we apply NSGA-II with constraint tournament method. Therefore, we explain NSGA-II that uses the constraint tournament mechanism instead of NSGA-II itself. In generic NSGA-II, binary tournament selection with a crowded tournament operator is used for reproduction and non-dominated sorting is applied for fitness assignment. However, in NSGA-II with constraint tournament method, binary tournament selection with a constraint tournament operator is used for reproduction and constraint-non-dominated sorting is applied for fitness assignment. In both approaches of NSGA-II, an explicit diversity preserving mechanism is used.

Illustration of the constraint-non-dominated sorting concept is displayed in Figure 4. In Figure 4, population consists of nine solutions (objectives are to be minimized). Constraint-non-dominated sorting is performed as follows; firstly, feasible solutions are determined and dominance check is applied to feasible solutions. Each feasible solution is compared with every other feasible solution in the population for checking whether it is dominated by any feasible solution or not. If it is not dominated by any of the feasible solution in the population, it belongs to first constraint-non-dominated front. In Figure 4, only one solution constitutes the first constraint-non-dominated front.

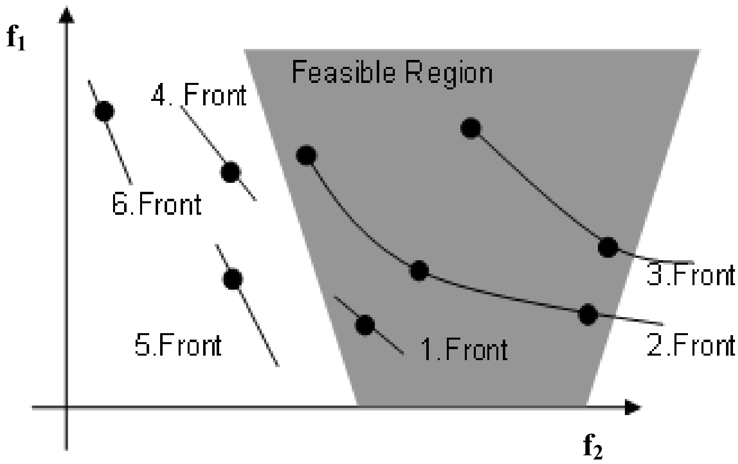


Figure 4 *Determining Fronts in NSGA-II with Constraint Tournament Method*

After the determination of the first constraint-non-dominated front, solutions in the first constraint-non-dominated front are removed temporarily from the population and domination check is performed for remaining feasible solutions and second constraint-non-dominated front is constructed by three solutions as seen in Figure 4. These three solutions are also temporarily removed from the population and third constraint-non-dominated front is determined from the remaining two feasible solutions. Finally, three constraint-non-dominated fronts are determined for feasible solutions. However, there are also some infeasible solutions in the population. Overall constraint violations are calculated for infeasible solutions and infeasible solution that has the minimum violation constitute the next constraint-non-dominated front. As seen in Figure 4, three constraint-non-dominated fronts are determined for infeasible solutions. Finally, six constraint-non-dominated fronts are determined in Figure 4.

NSGA-II also uses a mechanism called crowding distance to maintain diversity among solutions. Crowding distance can be interpreted as an estimate of the density of solutions in the neighborhood of a specific solution. It finds the total scaled distance of two solutions on either side of a solution along each objective. Firstly, for each front, solutions are sorted according to each objective in ascending order and sorted lists are formed. Then, boundary solutions are determined and very large numbers are assigned to them. For remaining solutions in the front, crowding distance of each solution (d_j) is calculated using following formula:

$$d_j = \sum_{m=1}^{M} \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}} \qquad \forall j = 2,....,(l-1)$$

where I_j is the solution index of the j^{th} solution in the sorted list of m^{th} objective. $f_m^{(I_{j+1}^m)}$ and $f_m^{(I_{j-1}^m)}$ are the neighbors of the j^{th} solution according to the m^{th} objective. $f_m^{\max}$ and $f_m^{\min}$ denote the maximum and minimum values for the m^{th} objective respectively. M is the number of objectives; l is the number of solutions exists in each front. In the formula, $f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}$ is the difference between neighboring solutions of j^{th} solution with respect to m^{th} objective. This difference is scaled with the range of m^{th} objective. Total scaled differences constitute the crowding distance of j^{th} solution.

In Figure 5, crowding distance concept is illustrated; five solutions (a, b, c, d, and e) constitute the front. Cuboid (rectangle) distance of solution c is shown with dashed lines. Solution “a” and “e” are the boundary solutions to which larger crowding distance values are assigned. Solution “b” and “d” are the neighboring solutions of solution “c”. Crowding distance of solution “c” is calculated as follows;

$$d_c = \frac{f_1^d - f_1^b}{f_1^{\max} - f_1^{\min}} + \frac{f_2^b - f_2^d}{f_2^{\max} - f_2^{\min}}$$

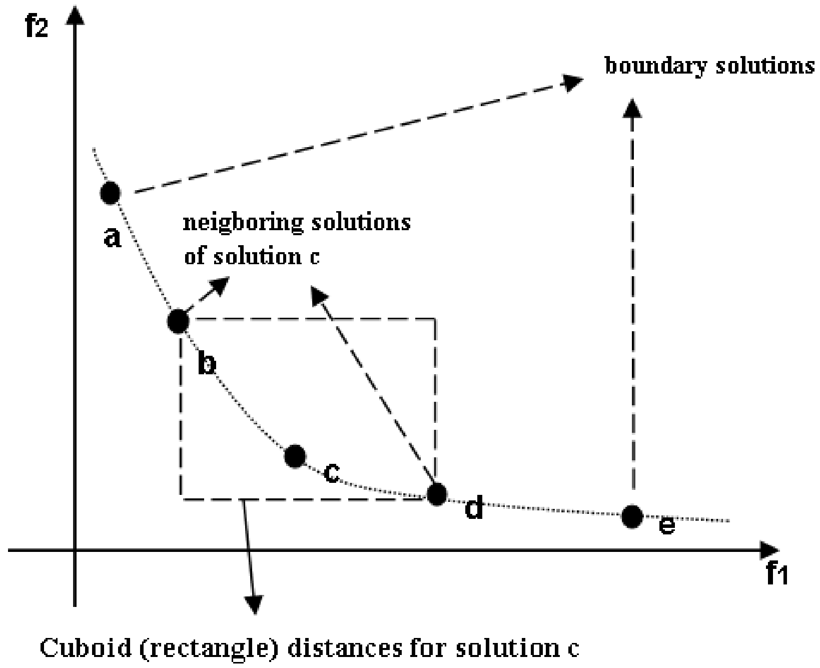


Figure 5 *The Cuboid (rectangle) Distances of a Solution in a Front*

In NSGA-II with constraint tournament method, an initial population (P_0) is randomly generated. Constraint-non-domination sorting is applied, and constraint-non-dominated fronts are determined. Each solution in the population is assigned a fitness value indicating its belonging constraint-non-dominated front. Binary tournament selection with a constraint tournament operator, crossover and mutation operators are used to create an offspring population Q_0 of size N.

In Figure 6, schematic of the NSGA-II procedure with constraint tournament method is depicted. As seen in Figure 6, after offspring population Q_t is created from initial population, they are combined into R_t with size 2N and constraint-non-dominated sorting is applied to R_t. From R_t population, N solutions are selected based on the constraint-non-domination sorting and crowding distance; and the parent population of the next step P_{t+1} is created. As seen in Figure 6, four fronts exist for R_t, the last front is directly discarded due to the size N restriction. However, number of solutions in the first two front are smaller than size N, whereas number of solutions in the first three front are bigger than size N. Thus, third front (F_3) somehow has to be split to fill up the P_{t+1}. At this point, crowded distance sorting is applied and solutions with the larger crowding distances are placed into the parent population P_{t+1}. The reason of using crowding distance is favoring the solutions that reside in a less crowded region.

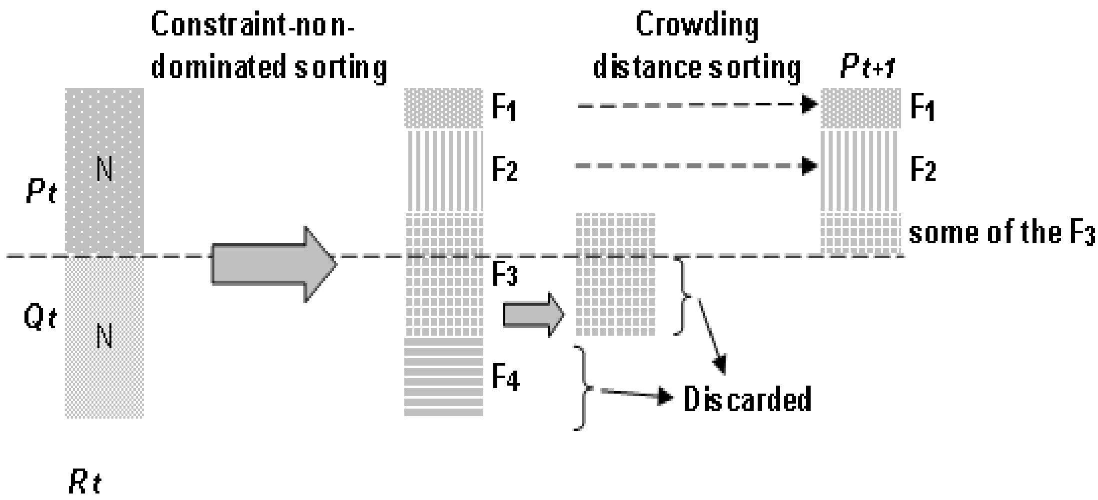


Figure 6 *Schema of the NSGA-II with Constraint Tournament Method*

After the production of P_{t+1}, binary constraint tournament selection operator is used for filling up the mating pool. Two solutions are randomly selected from the parent population P_{t+1}, solution belongs to better front wins the tournament and placed into the mating pool; in the case of belonging the same front, solution with the larger crowding distance wins the tournament and placed into the mating pool. Once the mating pool is formed by the selection operator, crossover and mutation operators are performed to create offspring population Q_{t+1}. This process goes on until a termination condition is achieved. In Figure 7, steps of NSGA-II with constraint tournament method are presented.

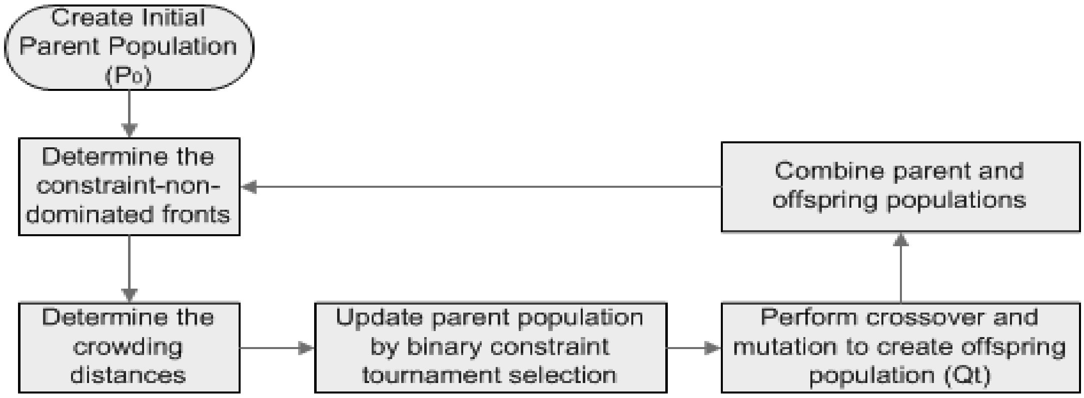


Figure 7 *Steps of the NSGA-II with constraint tournament method*

CHAPTER 4

AN INTERACTIVE BIOBJECTIVE PROMETHEE V APPROACH (ib-PROMETHEE V)

4.1 MOTIVATION

The main motivation behind the proposed approach is that "while calculating net flows some useful information about incomparabilities is lost" (Brans and Vincke 1985; Ozerol and Karasakal 2008). In order to capture these incomparabilities, namely leaving and entering flows, ib-PROMETHEE V is proposed. As formulated in Section 3, PROMETHEE V combines PROMETHEE II with 0-1 integer programming. In ib-PROMETHEE V, we integrated PROMETHEE I with 0-1 integer programming to capture the trade-offs between leaving and entering flows.

Mavrotas et al. (2006) stated that PROMETHEE V formulation usually violates the project ranking scores (ranking of PROMETHEE II) due to the budget constraint, i.e, the projects that have relatively higher budgets may not be selected although they have good ranking scores. Mavrotas et al. (2006) demonstrated this drawback in a simple example problem (see Table 2).

Table 2 *Example of Misleading Results*

Projects	Net Flow Score	Budget
A	65	50
B	30	20
C	40	10

In Table 2, assume three projects applied for funding. PROMETHEE V objective function prefers combination of projects B and C to project A since net flow score of combination of B and C is greater than that of A (70>65), and budget of combination of B and C is lower than that of A (30<50). In order to overcome this drawback, Mavrotas et al. (2006) proposed two approaches. First approach is the revision of the PROMETHEE V objective as follows:

$$\max \quad \frac{\sum_{i=1}^{n} \phi_i * X_i}{\sum_{i=1}^{n} X_i} \qquad (1)$$

However, Mavrotas et al. (2006) claimed that objective (1) complicates the solution procedure due to nonlinearity. Thus, they secondly proposed adding the following two constraints instead of the budget constraint to PROMETHEE V formulation:

$$\sum_{i=1}^{n} X_i \leq PR(k) \qquad (2)$$

$$\sum_{i=1}^{n} Budget_i * X_i - totbudg(k) = 0 \qquad (3)$$

where *PR(k)* is the maximum number of selected projects, *totbudg(k)* is the cumulative budget of *PR(k)* projects. They solve the problem *k* times for different *PR*(*k*) parameters and tried to find the appropriate range on the number of selected projects by both considering *totbudg(k)* and total available budget. They called this approach "parametric solution". To clarify this approach, assume that total available budget is 100 monetary units for a specific problem, and there are 20 projects apply for funding. In this case, assume that problem is solved with *PR(1)*=5, *PR(2)*=10, *PR(3)*=15 parameters and *totbudg(1)*=40, *totbudg(2)*=70, *totbudg(3)*=120 are obtained. Since total available budget is 100 monetary units, a suitable range on the number of selected project is between 10 and 15 (70<100<120). Once the suitable range is obtained, problem is solved with *PR(k)*=10,11,…15 and six solutions are generated. Finally, a single solution is selected from these six solutions. However, determination of *PR*(*k)* parameter is not well structured in Mavrotas et al. (2006). Mavrotas et al. (2006) also stated that they used the normalized net flow values of projects since negative values of net flows are opposite to the maximization objective of PROMETHEE V model.

4.2 PROPOSED APPROACH

In order to overcome the drawback of PROMETHEE V formulation, we simplified the idea of parametric solution. We propose a simple way to determine the upper and lower bounds on the number of selected projects by using budget information. We define constraint set (4) instead of budget constraint.

$$lb \leq \sum_{i=1}^{n} X_i \leq ub \qquad (4)$$

where *lb* is a lower bound and *ub* is an upper bound on the number of projects. We use these bounds to be able to generate a reasonable set of solutions that may convey valuable information about the projects and show trade-offs between the objectives and the budget. To determine *lb* and *ub* on the number of projects, we propose the following heuristic:

Step 1. Rank the projects in decreasing order of their budgets.

Step 2. Add up the budgets of the projects starting from the highest budget until reaching the available budget. Set the *lb* to the number of projects added up in this step.

Step 3. Add up the budgets of the projects starting from the lowest budget until reaching the available budget. Set the *ub* to the number of projects added up in this step.

We develop a two stage algorithm to solve project selection problems. In the first stage, we generate the Pareto front of ib-PROMETHEE V model. In the second stage, we obtain useful information about Pareto front by using data mining techniques and present this useful information to the DM to help him/her during the project selection process. Mathematical model of ib-PROMETHEE V for project selection problem is as follows:

$$\text{Max} \quad \sum_{i=1}^{n} \phi_i^{+} * X_i \qquad (5)$$

$$\text{Min} \quad \sum_{i=1}^{n} \phi_i^{-} * X_i \qquad (6)$$

Subject to:

Constraint set (4).

Segmentation constraints if any… (7)

$$X_i \in \{0,1\} \qquad (8)$$

where X_i is a binary variable, X_i = 1, if the i^{th} project is selected; and X_i = 0 otherwise. ϕ_i^{+} is the leaving flow, and ϕ_i^{-} is the entering flow of i^{th} project, respectively. Objectives functions (5) and (6) correspond to maximization of total leaving flow and minimization of total entering flow of selected projects, respectively. Constraint set (7) implies the segmentation constraints such as policy, logical, labor, and time. Constraint set (8) refers to binary decision variables.

We solve the ib-PROMETHEE V model by ε-constraint method. After generating the Pareto front, we ask the decision maker (DM) allowable deviation percentage (α) on the total available budget (*TAB*). According to the DM's answer, allowable budget range (*ABR*) is calculated as follows:

$$ABR = \frac{TAB * (100 \pm \alpha)}{100}$$

The solutions which are in the *ABR* are determined as candidate solutions. We assume that DM selects his/her preferred solution from the candidate solutions.

In a posteriori MCDM problem, DM selects the most preferred solution from the solution set by evaluating the objective values of solutions. However, this brings cognitive burden to DM. Moreover, s/he may want to know the differences among solutions in terms of decision variables, such as selection frequencies of projects in the solution set.

In our proposed approach, when we obtain the candidate set, we determine the projects that are different in the set and call these projects as critical projects. Selecting the most preferred solution from candidate solution set is equivalent to deciding on which critical projects to include in the solution. However, how can the DM decide on critical projects? How much knowledge does DM have about critical projects? At that point, data mining analysis of critical projects helps the DM to decide among critical projects. We propose following three post-optimality analyses by using SPSS Clementine data mining software :

Analysis 1. Frequency analysis: Selection frequencies of critical projects over Pareto front are analyzed. For example, if A is a critical project, number of solutions including the project A over the Pareto front gives the frequency of project A. It is expected that DM prefers the solutions whose frequencies are high.

Analysis 2. Classification and Regression Tree (CRT) analysis: Marginal impacts of critical projects on cumulative budget, entering flow, and leaving flow are analyzed by using CRT. Marginal impact refers to how the inclusion or exclusion of critical projects in the generated part of the Pareto front affects the cumulative budget, entering and leaving flows. In a problem, suppose Pareto front consist of 20 solutions, for example 11 solutions including project A have an average cumulative budget of $1000, whereas 9 solutions excluding project A have an average cumulative budget of $1600. So, Pareto front is classified into two sets according to the project A and DM has a chance to see the significant impact of project A on cumulative budget. Classification goes on in this manner and subsets are determined with the inclusion or exclusion of other projects. Thus, DM has a chance to see the classification of Pareto front in terms of both critical projects and objectives (cumulative budget, leaving flow, entering flow). We illustrate this analysis in detail in Section 5.

Analysis 3. Frequency distribution analysis: Frequency distribution of any critical project interested by DM over the range of cumulative budget, entering flow, and leaving flow is analyzed. In a problem, suppose the entering flow range of candidate solutions is between 21 and 25, for example, selection frequency of project A is low when the entering flow objective value is between 15 and 20, whereas selection frequency of project A is high when the entering flow objective value is between 21 and 25. It is expected that DM may prefer the projects whose frequencies are high in the desired range of cumulative budget, entering flow, and leaving flow.

Thereby, DM has a chance to learn more about critical projects with the guidance of above analyses. According to the data mining results on critical projects, DM may prefer some of the critical projects, then candidate solutions are updated and normalized values of leaving flow, entering flow, and cumulative budget of the candidate set in the interval [0,1] are graphically presented to the DM. Finally, the DM either selects the most preferred solution from updated candidate solutions or continues to do data mining analysis with the remaining critical projects.

We summarize the ib-PROMETHEE V below:

Stage I: Generating Pareto front

Step 1.1. Ask the DM to determine the type of preference function for each criterion, criterion weights, and thresholds. Compute entering and leaving flows of alternatives.

Step 1.2. Ask the DM about segmentation constraints if any.

Step 1.3. Ask the DM to determine *TAB* and α . Calculate the *ABR*.

Step 1.4. Compute the lower and upper bounds on number of selected projects by using *TAB*.

Step 1.5. Solve the mathematical model of ib-PROMETHEE V with ε-constraint method, and generate Pareto front.

Stage II: Data mining analysis and selecting the most preferred solution.

Step 2.1. Determine the candidate solution set.

Step 2.2. Determine the critical projects.

Step 2.3. Display the frequencies of the critical projects over the Pareto front.

Step 2.4. Display the marginal impacts of the critical projects on cumulative budget, entering flow, and leaving flow by using CRT.

Step 2.5. Display the frequency distribution of any critical project interested by DM over the range of cumulative budget, entering flow, and leaving flow.

Step 2.6. Ask DM which projects s/he prefers.

Step 2.7. Update the candidate solution set and display the normalized graph of the set.

Step 2.8. Ask DM whether s/he decides on selecting one of the solutions. If yes stop, otherwise go to step 7.

The flowchart of the ib-PROMETHEE V is shown in Figure 8.

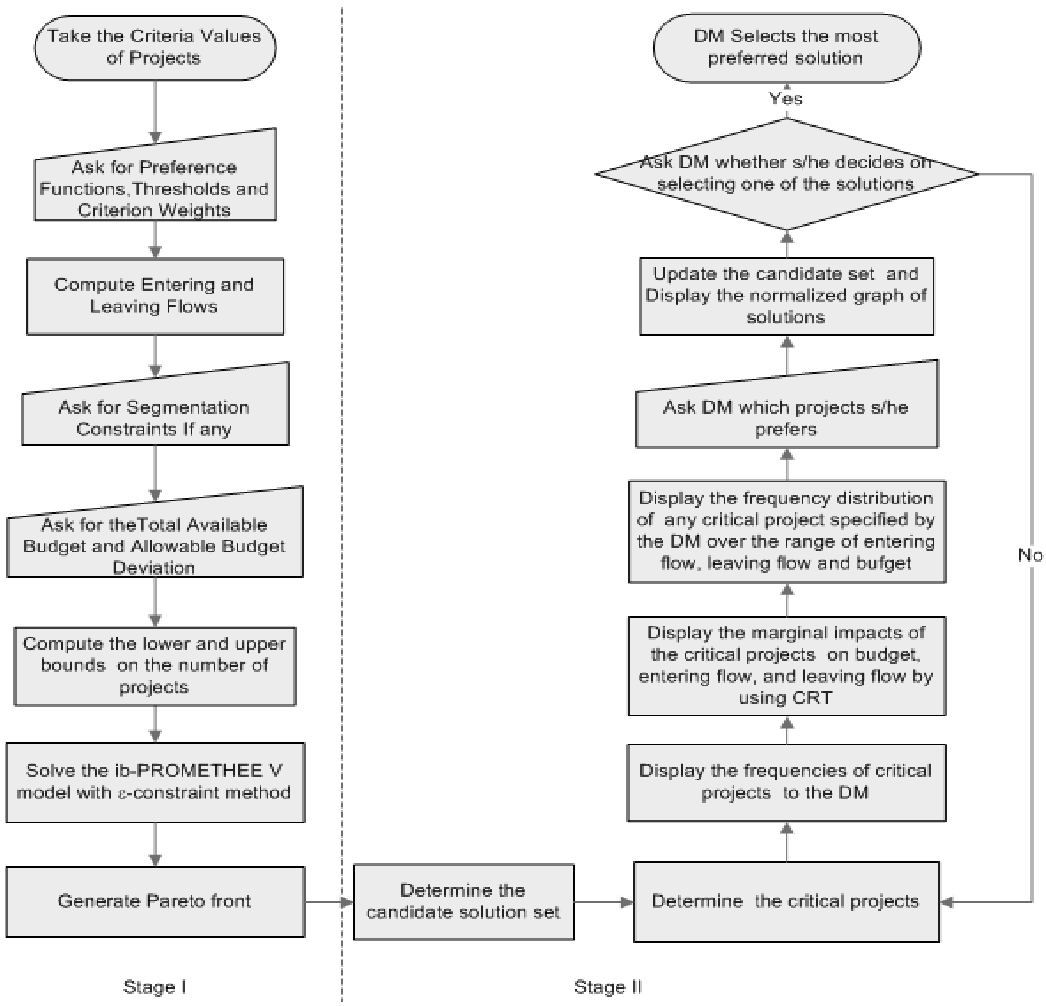


Figure 8 *Flowchart of ib-PROMETHEE V*

CHAPTER 5

ILLUSTRATIVE EXAMPLES

To illustrate ib-PROMETHEE V, firstly an example from literature is solved, and then a big-sized randomly generated example is solved and analyzed.

5.1 AN EXAMPLE FROM LITERATURE

Mavrotas et al. (2006) solve the following problem with their parametric PROMETHEE V solution method which is discussed in Section 4. We solve the same problem with the ib-PROMETHEE V algorithm. Our approach finds the results of Mavrotas et al. (2006) as well as different alternative solutions.

There are 20 industrial firms that apply for funding their projects. These firms belong to two different regions and to three different sectors and the submitted projects have different budgets. The projects are evaluated and scored according to the six criteria (to be maximized) that are shown in Table 3. Data of the literature example is given in Table 4.

***Table 3** Criteria characteristics*

Criteria	Criteria Definitions*	Weight	Preference Function Type	Threshold Values
Criterion 1 (C1)	Sales2000/Sales1999	0.14	Type V	q=0.05, p=0.2
Criterion 2 (C2)	Employees2000/employees1999	0.14	Type V	q = 0.05, p = 0.2
Criterion 3 (C3)	Profit margin/sales	0.14	Type V	q = 0.02, p = 0.1
Criterion 4 (C4)	Net profit/equity	0.14	Type V	q = 0.1, p = 0.3
Criterion 5 (C5)	Sales/employees	0.14	Type V	q = 5, p = 20
Criterion 6 (C6)	Quality of application	0.3	Type II	q = 1

* Criteria are explained in detail in Mavrotas et al. (2006).

Total available budget (*TAB*) is 4000k€ and there are some additional policy constraints in order to avoid the accumulation of funds in a specific region or sector. These constraints are stated below:

Regional constraint: The approved projects in metropolitan areas (Athens and Thessaloniki) must not exceed 75% of the total approved projects.

Sectoral constraint: The approved projects from each sector must be between 20% and 50% of the total approved projects.

***Table 4** Data for the literature example (Mavrotas et al. 2006)*

No	Firm	Region*	Sector*	Budget (k€)	C1	C2	C3	C4	C5	C6	ϕ_i^+	ϕ_i^-
1	A	A	TX	356	1.02	0.84	0.36	0.56	42.00	4.00	4.33	8.60
2	B	A	FD	256	1.04	0.91	0.17	0.35	28.00	3.00	0.94	11.71
3	C	A	CH	189	1.10	0.97	0.14	0.27	16.00	2.00	0.83	13.08
4	D	O	CH	203	1.37	1.10	0.26	0.22	25.00	5.00	4.65	7.66
5	E	A	FD	380	1.09	1.01	0.30	2.31	19.00	4.00	5.03	7.88
6	F	A	TX	114	2.19	1.20	0.04	0.11	32.00	5.00	5.22	8.61
7	G	O	FD	121	1.36	1.41	0.22	1.08	48.00	8.00	11.38	3.91
8	H	O	TX	376	1.74	1.40	0.17	0.88	34.00	9.00	10.40	3.85
9	I	A	FD	494	1.64	0.84	0.21	0.71	26.00	6.00	5.27	7.13
10	J	A	CH	116	1.04	1.00	0.02	0.07	33.00	2.00	0.79	13.31
11	K	A	FD	94	1.62	2.00	0.39	2.85	49.00	10.00	15.72	0.83
12	L	O	CH	116	1.11	1.82	0.21	0.77	16.00	6.00	5.50	7.09
13	M	O	TX	225	1.01	2.23	0.34	0.91	89.00	8.00	12.36	3.30
14	N	A	FD	1117	1.24	0.94	0.09	0.41	48.00	5.00	4.19	8.43
15	O	O	TX	475	1.10	1.00	0.27	0.72	32.00	5.00	3.82	6.88
16	P	A	TX	583	1.14	1.58	0.50	1.90	32.00	10.00	12.32	2.66
17	Q	A	FD	416	1.25	0.77	0.39	0.48	24.00	6.00	5.19	7.69
18	R	A	FD	156	1.16	1.12	0.26	0.65	44.00	6.00	6.10	5.91
19	S	O	TX	99	2.84	1.94	0.20	2.02	29.00	9.00	12.14	2.48
20	T	A	FD	1021	1.20	0.94	0.31	1.34	61.00	7.00	9.43	4.62

* The region A means Athens or Thessaloniki and O means others. The sector TX is for Textile, FD for Food and Beverage and CH for Chemicals.

We solve the above problem with ib-PROMETHEE V as follows:

Stage I: Generating Pareto front

Step 1.1: Preference function types, criterion weights, threshold are given in Table 3. Entering and leaving flows are calculated and displayed in the last two columns of Table 4.

Step 1.2: Regional and sectoral constraints are specified above.

Step 1.3: *TAB* is 4000k€, suppose DM indicates that allowable deviation percentage (α) is 10%.

$$ABR = \frac{TAB*(100 \pm \alpha)}{100} = \frac{4000*(100 \pm 10)}{100} \Rightarrow ABR \text{ is in the interval } [3600 - 4400]$$

Step 1.4: Lower and upper bounds on the number of selected projects are calculated.

***Table 5** Computing lower and upper bounds*

Firm	Budget (k€)	lb	Cumulative Budget Top to Bottom (k€)	ub	Cumulative Budget From Bottom to Top (k€)
N	1117	1	1117	20	6907
T	1021	2	2138	19	5790
P	583	3	2721	18	4769
I	494	4	3215	17	4186
O	475	5	3690	16	3692
Q	416	6	4106	15	3217
E	380	7	4486	14	2801
H	376	8	4862	13	2421
A	356	9	5218	12	2045
B	256	10	5474	11	1689
M	225	11	5699	10	1433
D	203	12	5902	9	1208
C	189	13	6091	8	1005
R	156	14	6247	7	816
G	121	15	6368	6	660
J	116	16	6484	5	539
L	116	17	6600	4	423
F	114	18	6714	3	307
S	99	19	6813	2	193
K	94	20	6907	1	94

In Table 5, "Budget" column presents the ranking of projects in decreasing order of their budgets. In the "Cumulative Budget Top to Bottom" and "Cumulative Budget Bottom to Top" columns, budget of projects calculated cumulatively. lb (lower bound) and ub (upper bound) columns show that how many projects are considered in calculating the corresponding cumulative budget value. For example, in the "Cumulative Budget Top to Bottom" column, the highest cumulative less than 4000 is 3690; this corresponds to the number 5 in the lb column. Similarly, in the "Cumulative Budget Bottom to Top" column, the highest budget less than 4000 is 3692; this corresponds to the number 16 in the ub column. Thereby, lb is set to 5, and ub is set to 16, respectively.

Step 1.5: ib-PROMETHEE V model of the problem is as follows:

$$\text{Max} \quad \sum_{i=\text{A}}^{\text{T}} \phi_i^+ * X_i \qquad i = \text{A, B,……., T} \qquad (9)$$

$$\text{Min} \quad \sum_{i=\text{A}}^{\text{T}} \phi_i^- * X_i \qquad i = \text{A, B,……., T} \qquad (10)$$

$$\sum_{i \in \text{Region A}} X_i \le 0.75 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (11)$$

$$\sum_{i \in \text{Sector TX}} X_i \le 0.5 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (12)$$

$$\sum_{i \in \text{Sector TX}} X_i \ge 0.2 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (13)$$

$$\sum_{i \in \text{Sector FD}} X_i \le 0.5 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (14)$$

$$\sum_{i \in \text{Sector FD}} X_i \ge 0.2 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (15)$$

$$\sum_{i \in \text{Sector CH}} X_i \le 0.5 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,……., T} \qquad (16)$$

$$\sum_{i \in \text{Sector CH}} X_i \geq 0.2 * \sum_{i=\text{A}}^{\text{T}} X_i \qquad i = \text{A, B,......, T} \qquad (17)$$

$$\sum_{i=\text{A}}^{\text{T}} X_i \leq 16 \qquad i = \text{A, B,......, T} \qquad (18)$$

$$\sum_{i=\text{A}}^{\text{T}} X_i \geq 5 \qquad i = \text{A, B,......, T} \qquad (19)$$

$$X_i \in \{0,1\} \qquad i = \text{A, B,......, T} \qquad (20)$$

where X_i is a binary variable, X_i = 1, if the i^{th} project is selected; and X_i = 0 otherwise. ϕ_i^+ is the leaving flow, and ϕ_i^- is the entering flow of i^{th} project, respectively. Objective function (9) maximizes the total leaving flow of the selected projects, and objective function (10) minimizes the total entering flow of selected projects. Regional constraint is represented in constraint set (11). Sectoral constraints are represented in constraint sets (12)-(17). Constraint sets (12) and (13) correspond to textile sector, (14) and (15) correspond to food and beverage sector, (16) and (17) correspond to chemical sector. Constraint sets (18) and (19) give upper and lower bounds on the number of selected projects. Constraint set (20) represents the binary decision variables.

Above model is solved with ε-constraint method via C program which calls the GAMS model iteratively. 25 non-dominated solutions are generated (see Table 6). In Table 6, each column corresponds to one solution, 1 and 0 values in each column shows whether the project is selected or not in the solution. Cumulative budget, total leaving flow, total entering flow, and number of selected projects of each solution is displayed at the end of each column. Frequency column shows the number of solutions including the project in the corresponding row (explained in Step 2.3). The gray columns in Table 6 shows the solutions found by Mavrotas et al. (2006). Pareto front of the problem in the objective space is also shown in Figure 9.

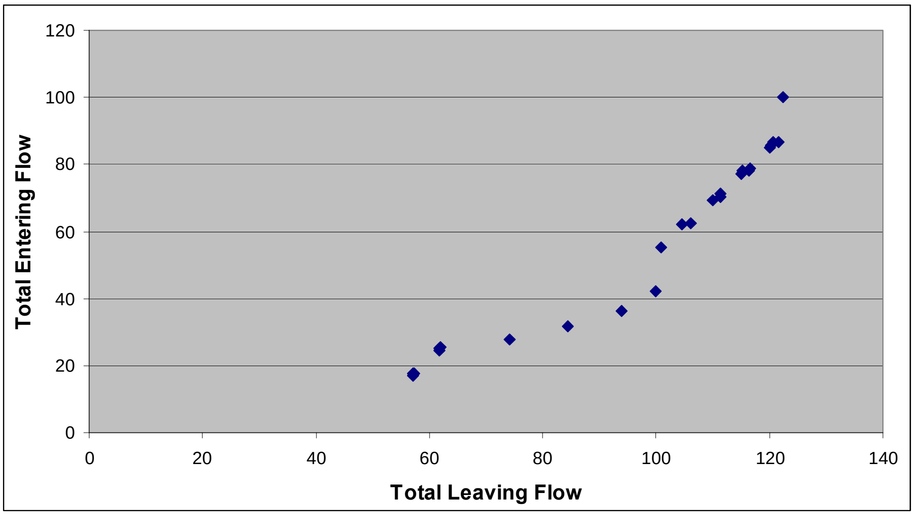


Figure 9 *Pareto front of the illustrative problem*

***Table 6** Non-dominated Solutions of Literature Example*

			Solution Number												
#	**Name**	**Budget**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	A	356	0	0	1	0	0	0	0	0	0	0	0	0	0
2	B	256	0	0	0	0	0	0	0	0	0	0	0	0	0
3	C	189	1	1	1	1	1	1	1	1	1	1	1	1	1
4	D	203	1	1	1	1	1	1	1	1	1	1	1	1	1
5	E	380	1	1	1	0	1	0	1	0	0	0	0	0	0
6	F	114	1	1	0	1	0	1	0	1	0	1	0	0	0
7	G	121	1	1	1	1	1	1	1	1	1	1	1	1	1
8	H	376	1	1	1	1	1	1	1	1	1	1	1	1	1
9	I	494	1	1	1	1	1	1	1	1	1	1	1	1	1
10	J	116	1	0	0	0	0	0	0	0	0	0	0	0	0
11	K	94	1	1	1	1	1	1	1	1	1	1	1	1	1
12	L	116	1	1	1	1	1	1	1	1	1	1	1	1	1
13	M	225	1	1	1	1	1	1	1	1	1	1	1	1	1
14	N	1117	0	0	0	0	0	0	0	0	0	0	0	0	0
15	O	475	0	0	0	1	1	0	0	1	1	0	0	1	0
16	P	583	1	1	1	1	1	1	1	1	1	1	1	1	1
17	Q	416	1	1	1	1	1	1	1	0	1	0	1	0	0
18	R	156	1	1	1	1	1	1	1	1	1	1	1	1	1
19	S	99	1	1	1	1	1	1	1	1	1	1	1	1	1
20	T	1021	1	1	1	1	1	1	1	1	1	1	1	1	1
Cumulative Budget			4703	4587	4829	4682	4948	4207	4473	4266	4568	3791	4093	4152	3677
Total Leaving Flow			122.33	121.54	120.65	120.33	120.14	116.51	116.32	115.14	115.11	111.32	111.29	109.92	106.1
Total Entering Flow			100.01	86.7	86.69	85.7	84.97	78.82	78.09	78.01	77.09	71.13	70.21	69.4	62.52
Number of Selected Projects			16	15	15	15	15	14	14	14	14	13	13	13	12

Table 6 *Non-dominated Solutions of Literature Example (Continues)*

			Solution Number												
#	**Name**	**Budget**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**Frequency**
1	A	356	0	0	0	0	0	0	0	0	0	0	0	0	1
2	B	256	0	0	0	0	0	0	0	0	0	0	0	0	0
3	C	189	1	1	0	0	0	0	0	0	0	0	0	0	15
4	D	203	1	1	1	1	1	1	1	1	1	0	0	0	22
5	E	380	0	0	0	0	0	0	0	0	0	0	0	0	5
6	F	114	0	0	0	0	0	0	0	0	0	0	0	0	6
7	G	121	1	1	1	1	1	1	1	1	1	1	1	1	25
8	H	376	1	1	1	1	1	0	0	0	0	0	0	0	18
9	I	494	0	0	0	0	0	0	0	0	0	0	0	0	13
10	J	116	0	0	0	0	0	0	0	0	0	0	0	0	1
11	K	94	1	1	1	1	1	1	1	1	1	1	1	1	25
12	L	116	1	1	1	1	1	1	1	1	1	1	1	1	25
13	M	225	1	1	1	1	1	1	1	1	0	1	1	0	23
14	N	1117	0	0	0	0	0	0	0	0	0	0	0	0	0
15	O	475	1	0	0	0	0	0	0	0	0	0	0	0	6
16	P	583	1	1	1	1	1	1	1	0	1	1	0	1	23
17	Q	416	0	0	0	0	0	0	0	0	0	0	0	0	9
18	R	156	1	1	1	0	0	0	0	0	0	0	0	0	16
19	S	99	1	1	1	1	1	1	0	1	1	0	1	1	23
20	T	1021	1	1	1	1	0	0	0	0	0	0	0	0	17
Cumulative Budget			3658	3183	2994	2838	1817	1441	1342	858	1216	1139	655	1013	
Total Leaving Flow			104.65	100.83	100	93.9	84.47	74.07	61.93	61.75	61.71	57.28	57.1	57.06	
Total Entering Flow			62.27	55.39	42.31	36.4	31.78	27.93	25.45	25.27	24.63	17.79	17.61	16.97	
Number of Selected Projects			12	11	10	9	8	7	6	6	6	5	5	5	

Stage II: Data mining analysis and selecting the most preferred solution.

Step 2.1: Solutions 6, 8, 10, 11, 12, 13, and 14 are in the *ABR*. Hence, these seven solutions form the candidate set. Candidate set is shown in Table 7.

***Table 7** Candidate solution set*

	Name	Budget	6	8	10	11	12	13	14	Frequency
1	A	356	0	0	0	0	0	0	0	0
2	B	256	0	0	0	0	0	0	0	0
3	C	189	1	1	1	1	1	1	1	7
4	D	203	1	1	1	1	1	1	1	7
5	E	380	0	0	0	0	0	0	0	0
6	F	114	1	1	1	0	0	0	0	3
7	G	121	1	1	1	1	1	1	1	7
8	H	376	1	1	1	1	1	1	1	7
9	I	494	1	1	1	1	1	1	0	6
10	J	116	0	0	0	0	0	0	0	0
11	K	94	1	1	1	1	1	1	1	7
12	L	116	1	1	1	1	1	1	1	7
13	M	225	1	1	1	1	1	1	1	7
14	N	1117	0	0	0	0	0	0	0	0
15	O	475	0	1	0	0	1	0	1	3
16	P	583	1	1	1	1	1	1	1	7
17	Q	416	1	0	0	1	0	0	0	2
18	R	156	1	1	1	1	1	1	1	7
19	S	99	1	1	1	1	1	1	1	7
20	T	1021	1	1	1	1	1	1	1	7
Cumulative Budget			4207	4266	3791	4093	4152	3677	3658	
Total Leaving Flow			116.51	115.14	111.32	111.29	109.92	106.1	104.65	
Total Entering Flow			78.82	78.01	71.13	70.21	69.4	62.52	62.27	
Number of Selected Projects			14	14	13	13	13	12	12	

Step 2.2: Critical projects are determined from the frequency column of Table 7. Frequency column shows the number of solutions including the project in the corresponding row, i.e, project A is not selected in any of the candidate solution since its frequency value is zero, and project C is selected in all candidate solutions since its frequency value is seven. We look for the projects which are selected in some solutions; i.e., projects whose frequencies are strictly between zero and seven. Thus, projects F, I, O, and Q are the critical projects since their frequencies are 3, 6, 3, and 2, respectively.

Step 2.3: In this step, we present the frequencies of projects F, I, O, and Q over the Pareto front to the DM. In Table 8, the frequencies of these projects over Pareto front are shown. This information is taken from the frequency column of Table 6. Suppose the DM realizes that frequencies of project I and Q are relatively high.

Table 8 *Frequency analysis of critical projects*

Critical Projects	Frequency (out of 25 solutions)	Percentage
F	6	24%
I	13	52%
O	6	24%
Q	9	36%

Step 2.4: The marginal impacts of the projects F, I, O, and Q on cumulative budget, leaving flow, and entering flow are depicted in Figure 10, Figure 11, and Figure 12, respectively. In these figures, CRT branches on the critical projects. In SPSS Clementine, once we specify the projects, software automatically place the projects into the CRT branches. Application of the software is described in Appendix A.

A variety of useful information can be extracted from Figures 10, 11, and 12. For instance, in Figure 10 “node 0” shows that there are 25 solutions and their average cumulative budget is 3165.2; “node 4” shows that average cumulative budget is 4382.769 for the solutions including project I, and 1846.167 for the solutions excluding project I. Therefore, project I has a great impact on the cumulative budget and project I’s budget is relatively high compared to those of other projects. “Node 12” depicts that average cumulative budget is 4565.556 for the solutions both including projects I and Q. In Figures 11, and 12, leaving flow and entering flow classification are presented, respectively. Suppose that DM evaluates “node 12” in Figures 10, 11, and 12; and realizes that frequency of the solutions including both project I and Q is 9 out of 25.

Step 2.5: Suppose that DM has some idea about critical project, s/he thinks that frequencies of project I and Q are high and solutions including both project I and Q constitute the 36% of the Pareto front.

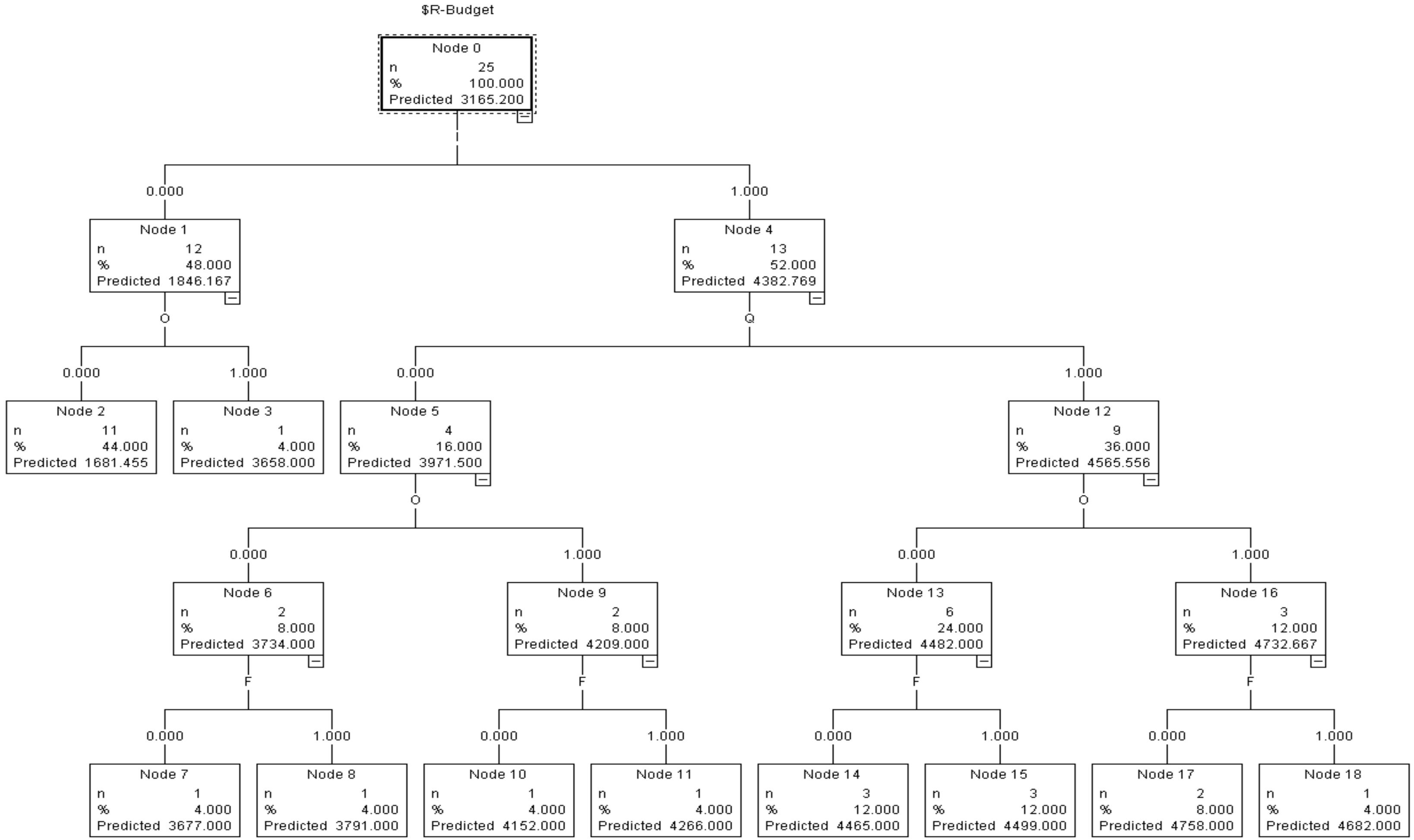


Figure 10 *CRT of the cumulative budget*

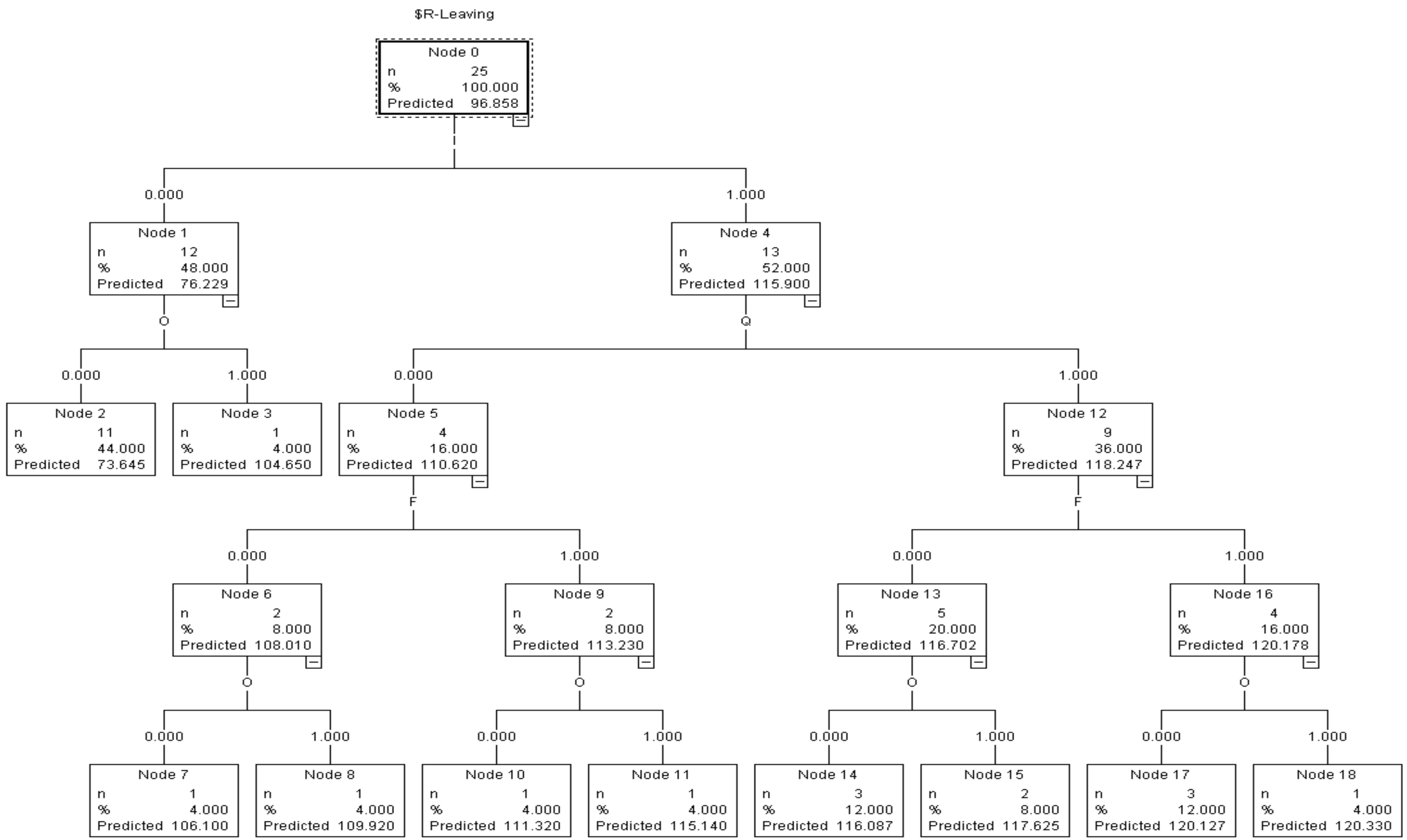


***Figure 11** CRT of Leaving Flow*

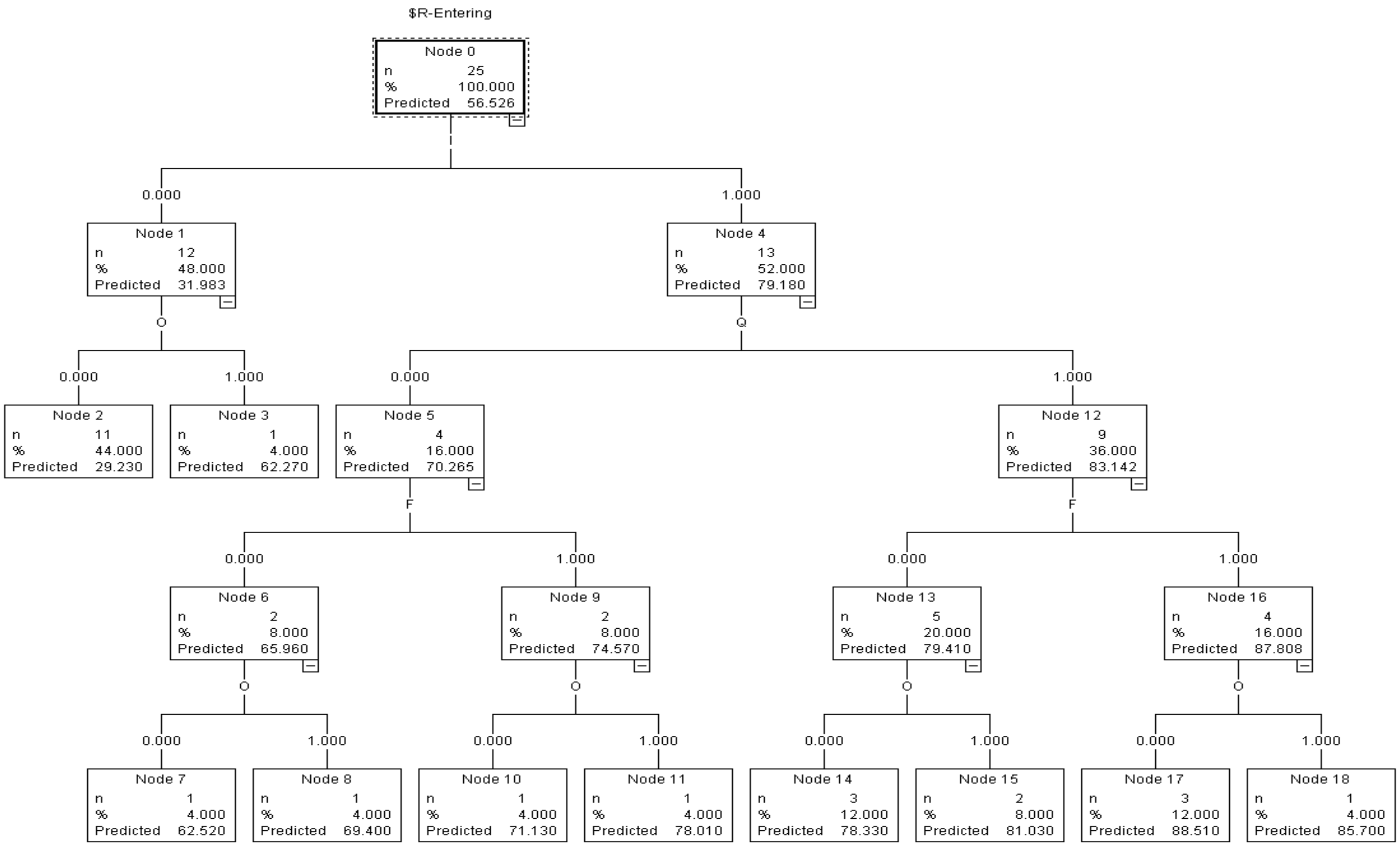


Figure 12*CRT of Entering Flow*

However, suppose that DM thinks that project I has a great impact on cumulative budget, and wants to see the frequency distribution of this project over the range of the cumulative budget, leaving flow, and entering flow to make the final decision on it. In Figure 13, frequency distribution of project I over the range of cumulative budget, leaving flow, and entering flow is depicted, respectively.

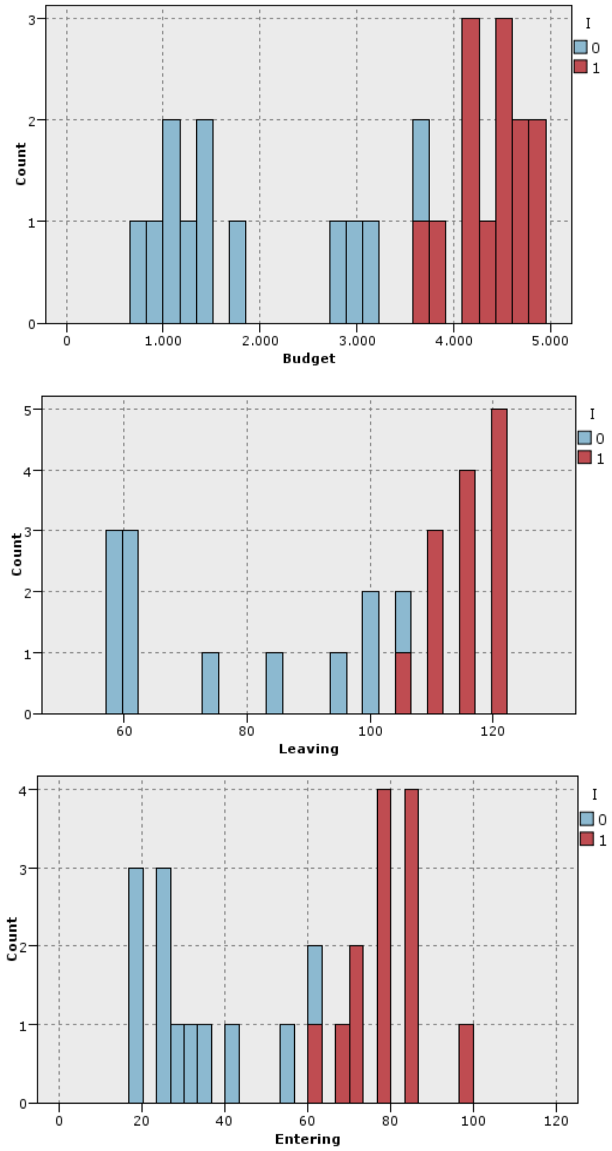


***Figure 13** Distribution of project I over cumulative budget, leaving and entering flows*

In Figure 13, at the right top corner of the graphs, I represents the project I, 1 and 0 represent whether the project I is selected or not. Suppose DM evaluates Figure 13, and realizes that frequency of the project I is high in the cumulative budget range [3658-4207]; in the leaving flow range [104-116]; and in the entering flow range [62-78]. These ranges correspond to the ranges of candidate solutions (see Table 7).

Step 2.6: In the light of above analyses suppose DM prefers project I and Q. Thus, DM wants to see the project I and the project Q in his/her most preferred solution.

Step 2.7: We update the candidate set in which project I and project Q selected simultaneously. Only solutions 6 and 11 are left. We display the normalized graph of updated candidate set to DM in Figure 14. In Figure 14, cumulative budget is normalized according to the available budget (0 shows the available budget in the y axis); leaving flow and entering flow are normalized according to their maximum and minimum values in the candidate set.

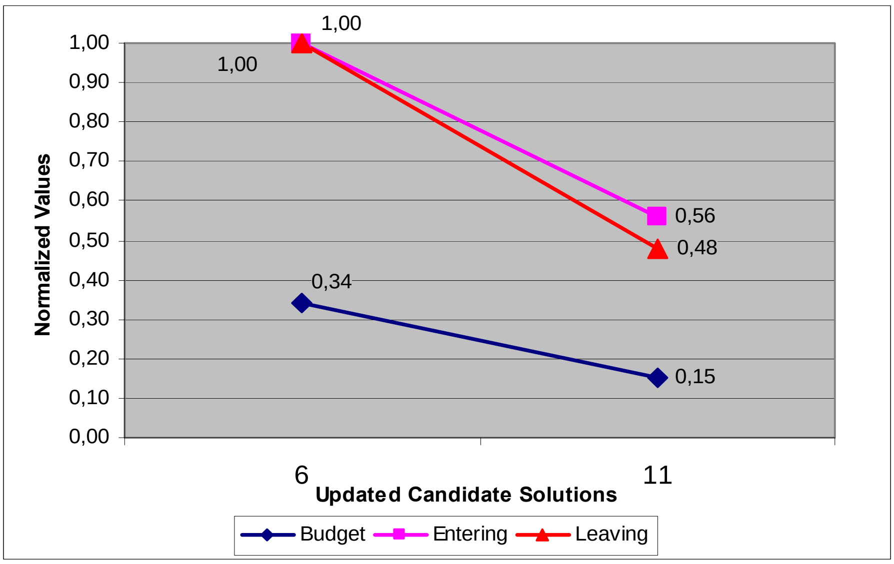


Figure 14 *Normalized Graph of Updated Candidate Solutions*

Step 2.8: There is a trade off between solution 6 and solution 11 in terms of cumulative budget, leaving and entering flow. Suppose DM does not want to exceed available budget too much, solution 11 is selected as the most preferred solution.

To summarize, “ib-PROMETHEE V” approach gives alternative solutions for the specific number of selected projects (i.e. what are the solutions if we select 15 projects?), however parametric PROMETHEE V approach (Mavrotas et al., 2006) gives only one solution for the specific number of selected projects. Thus, our approach not only finds the solutions of PROMETHEE V, but also finds the alternative solutions because of the biobjective formulation. Note that our approach finds all the efficient solutions including convex dominated ones.

5.2 RANDOMLY GENERATED EXAMPLE

In this section, we generate an example problem. Criterion, budget, threshold values are uniformly generated. Criterion weights are taken equal. There are 100 projects and 10 criteria. We assume that all criteria are to be maximized and PROMETHEE function type is V for all criteria.

Stage I: Generating Pareto front

Step 2.1: Preference function types, criterion weights, thresholds information are given above. Entering and leaving flows are calculated.

Step 2.2: Suppose DM does not specify any segmentation constraints.

Step 2.3: Suppose total available budget (*TAB*) is 5000 monetary units, suppose DM indicates that allowable deviation percentage (α) is 2%.

$$ABR = \frac{TAB*(100 \pm \alpha)}{100} = \frac{5000*(100 \pm 2)}{100} \Rightarrow ABR \text{ is in the interval } [4900 - 5100]$$

Step 2.4: Lower (*lb*) and upper bounds (*ub*) on the number of selected projects are calculated as 21 and 46, respectively.

Step 2.5: ib-PROMETHEE V model of the problem is as follows;

Max $\quad \sum_{i=1}^{100} \phi_i^+ * X_i$

Min $\quad \sum_{i=1}^{100} \phi_i^- * X_i$

s.t

$$\sum_{i=1}^{100} X_i \leq 46$$

$$\sum_{i=1}^{100} X_i \geq 21$$

$$X_i \in \{0,1\} \qquad (5)$$

328 efficient solutions are generated by ε-constraint method. Pareto front of the problem in the objective space is shown in Figure 15.

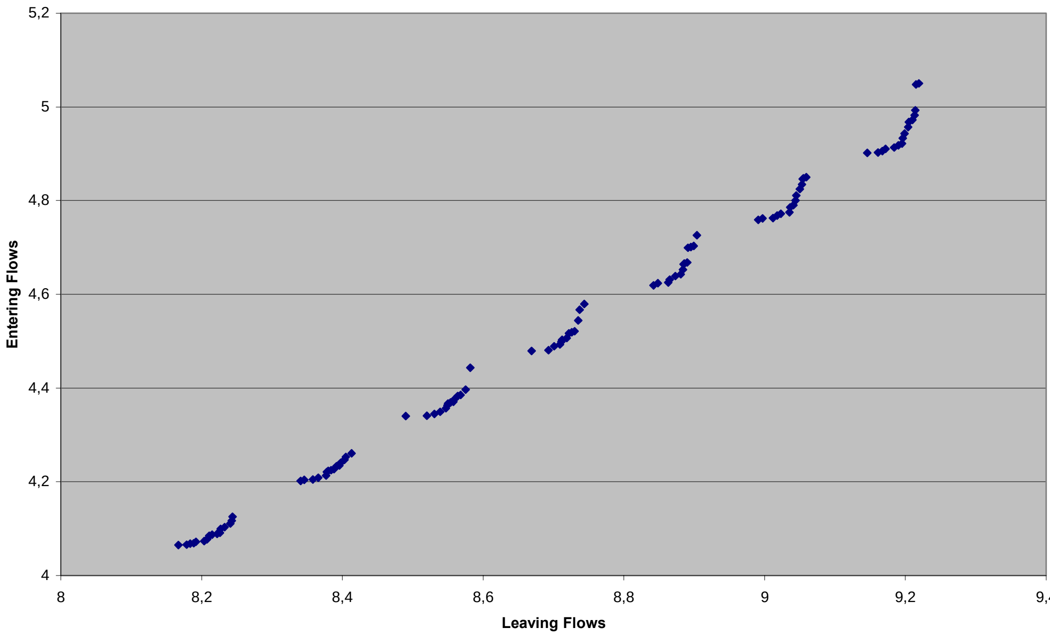


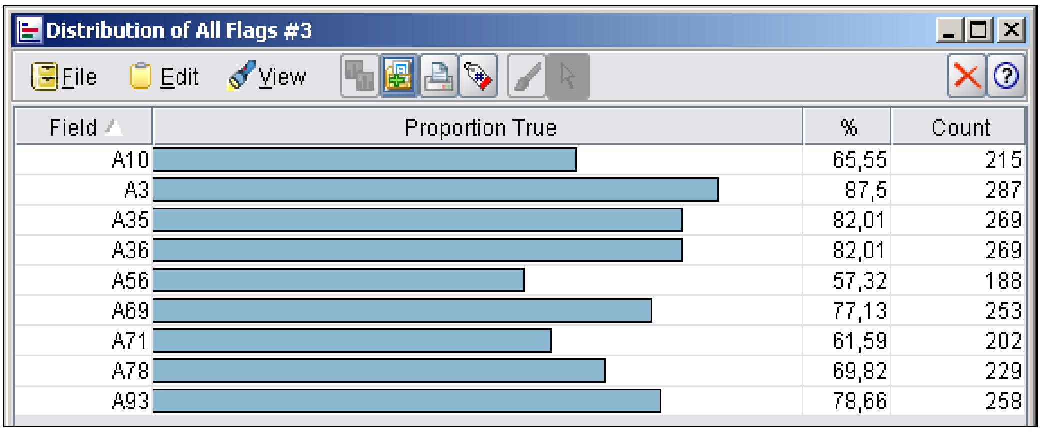
Figure 15 *Pareto front of the randomly generated example*

Stage II: Data mining analysis and selecting the most preferred solution.

Step 2.1: 11 solutions are in the ABR out of 328 solutions. Candidate set is shown in the Appendix B, Table 20. In Appendix B, Table 20 each column corresponds to one solution, 1 and 0 values in each column shows whether the project is selected or not in the solution; cumulative budget, total leaving flow, total entering flow, and number of selected projects of each solution are displayed at the end of each column; "F" column shows the number of solutions including the project in the corresponding row.

Step 2.2: Critical projects can be easily determined by looking at the "F" column of Appendix B. There are 9 critical projects; project A3, A10, A35, A36, A56, A69, A71, A78 and A93.

Step 2.3: The frequencies of the critical projects over the Pareto front are shown in Figure 16. "%" column represents the frequency in percentage. Suppose DM does not prefer project A56 since its frequency percentage (57%) is the lowest among the critical projects. Note that frequency analysis is more meaningful in this example since the Pareto front size is big enough to make comprehensive judgments.



Field	Proportion True	%	Count
A10		65,55	215
A3		87,5	287
A35		82,01	269
A36		82,01	269
A56		57,32	188
A69		77,13	253
A71		61,59	202
A78		69,82	229
A93		78,66	258

Figure 16 *SPSS Clementine frequency analyses of critical projects*

Step 2.4: The marginal impacts of the critical projects on cumulative budget, leaving flow, and entering flow are depicted in Figures 17, 18 and 19, respectively. In these Figures, effect means that the difference between the current node and its parent node, numbers in the parenthesis indicate the number of projects in the corresponding nodes.

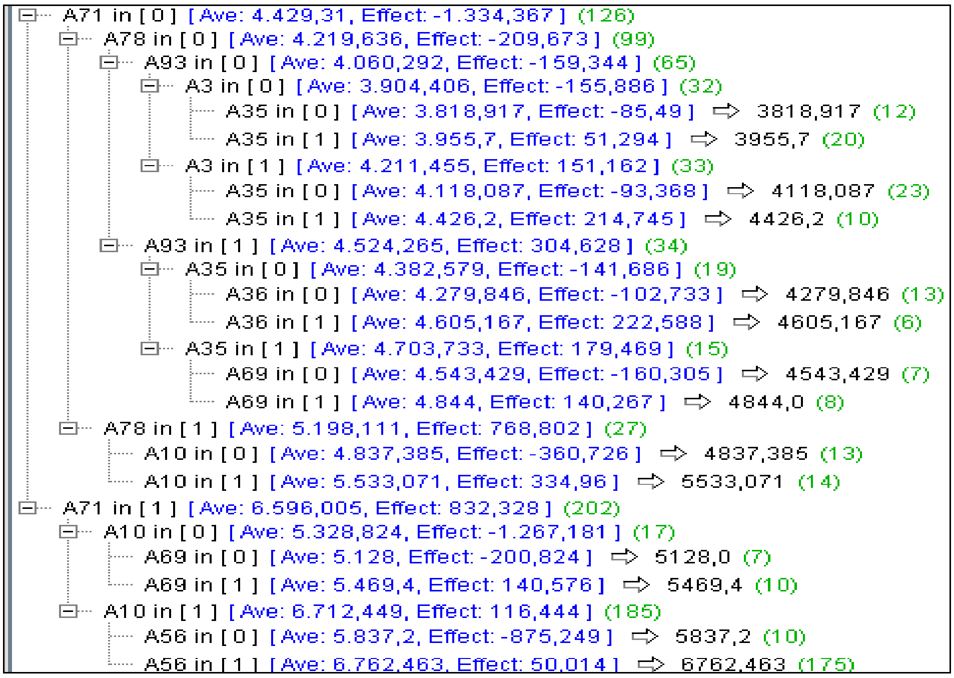


Figure 17 *CRT list of cumulative budget*

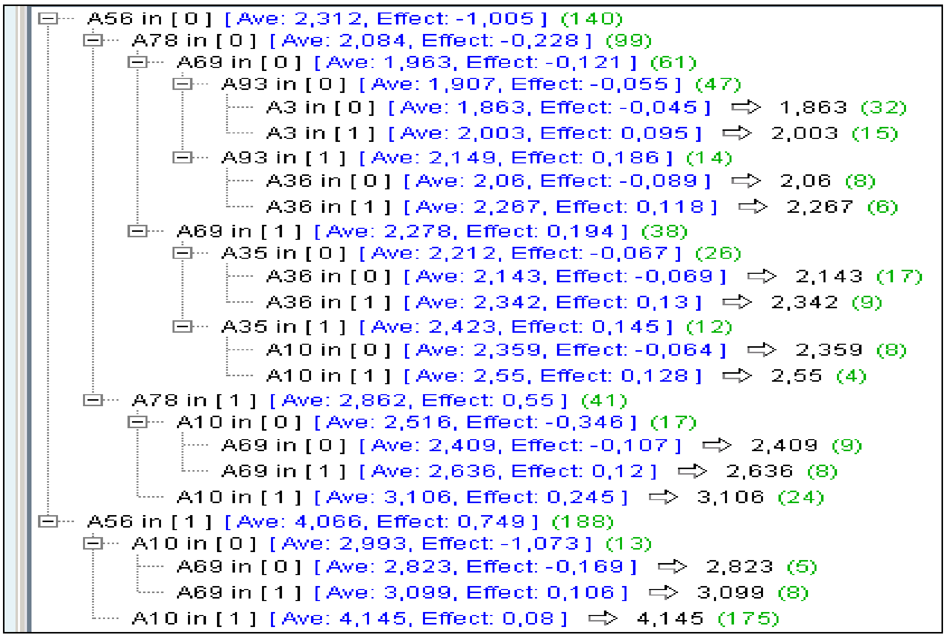


Figure 18 *CRT list of leaving flow*

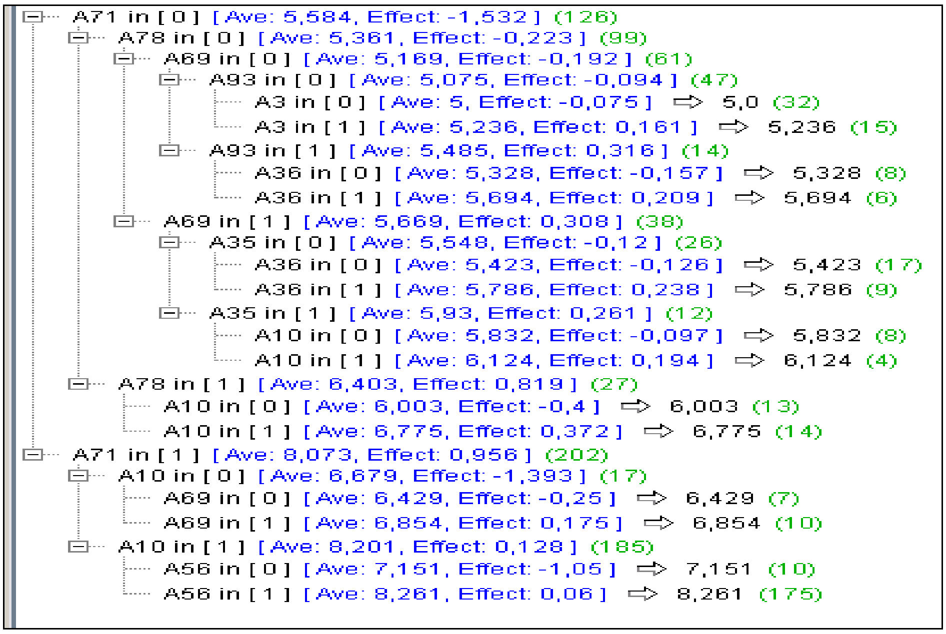


Figure 19 *CRT list of entering flow*

From Figure 17-19, it is seen that project A71 has a great marginal impact on cumulative budget and entering flow and does not have any significant impact on leaving flow. In Figure 17, it is seen that solutions excluding project A71 have an average cumulative budget of 4429, and solutions including project A71 have an average cumulative budget of 6596. In Figure 19, it is seen that solutions excluding project A71 have an average entering flow of 5.584, and solutions including project A71 have an average entering flow of 8.073. Figure 17 shows that frequency of project A78 in the solutions excluding project A71 is 27 out of 328 solutions; whereas there are 99 solutions excluding both projects A71 and A78.

Step 2.5: Frequencies of project A56, A71, and A78 are relatively low when compared to other critical projects. Also including A71 into the solution increases cumulative budget and entering flow which is not desirable and does not have significant impact on leaving flow.

In Figure 20, frequency distribution of project A71 over the range of cumulative budget, leaving flow, and entering flow is depicted, respectively. Suppose DM evaluates Figure 20, and realizes that frequency of the project A71 in the budget range [4915-5093]; in the leaving flow range [2.51-2.67]; and in the entering flow range [6.10-6.31] is very low. These ranges correspond to the candidate solutions range (see Appendix B).

Step 2.6: According to the data mining results, suppose the DM does not prefer projects A56, A71, and A78.

Step 2.7: We update the candidate set in which projects A56, A71, and A78 are not selected simultaneously. Solutions 2, 8 and 10 are left. We display the normalized graph of updated candidate set to DM in Figure 21. In Figure 21, budget is normalized according to the available budget; leaving flow and entering flow are normalized according to their maximum and minimum values in the candidate set.

Step 2.8: Suppose, the DM selects solution 8 from Figure 21.

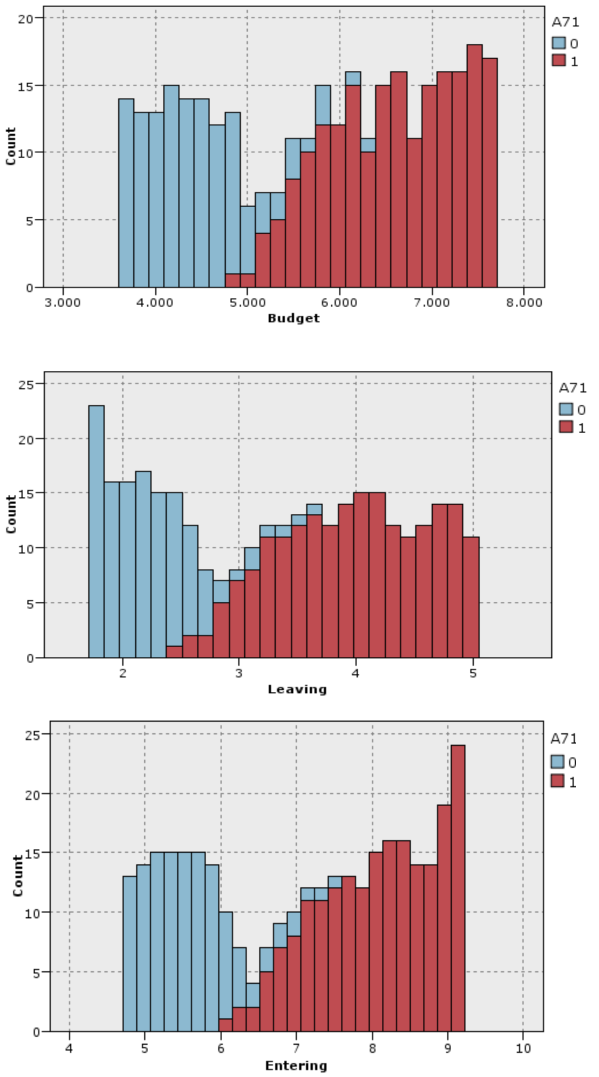


Figure 20 *Distribution of project A71 over budget, leaving and entering flows*

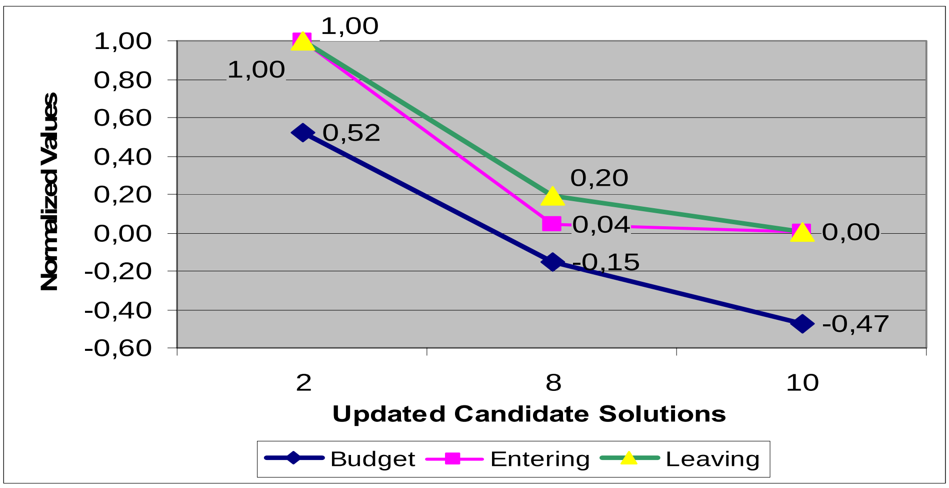


Figure 21 *Normalized graph of updated candidate solutions*

CHAPTER 6

AN INTERACTIVE MULTIOBJECTIVE GENETIC ALGORITHM (IMGA)

We propose another mathematical model for project selection problem to capture the trade-offs between net flow scores and budget of projects; and develop an interactive multiobjective genetic algorithm (IMGA) to solve the proposed model. We put lower and upper bounds to the budget objective not to generate the whole Pareto front.

Our proposed mathematical model is:

Max $z_1 = \sum_{i=1}^{n} \phi_i * X_i$ (21)

Min $z_2 = \sum_{i=1}^{n} Budget_i * X_i$ (22)

s.t

$$\sum_{i=1}^{n} Budget_i * X_i \geq \frac{TAB*(100-\alpha)}{100} \quad (23)$$

$$\sum_{i=1}^{n} Budget_i * X_i \leq \frac{TAB*(100+\alpha)}{100} \quad (24)$$

$$X_i \in \{0,1\} \quad (25)$$

where X_i is a binary variable, X_i = 1, if the i^{th} project is selected; and X_i = 0 otherwise. ϕ_i is the normalized net flow of the i^{th} project, $Budget_i$ is the budget of i^{th} project. TAB is the total available budget of the problem. α is the allowable budget deviation percentage. Objective (21) corresponds to maximizing total normalized net flow. We normalized the net flow of projects as in the Mavrotas et al (2006). Objective (22) corresponds to minimizing cumulative budget of the selected projects. Constraint set (23) and (24) correspond to lower and upper bounds on the objective (22), respectively. Constraint set (25) corresponds to binary decision variables.

Non-dominated solutions of the above mathematical model are obtained by using ε-constraint method in a reasonable time for small and moderate-sized problems. However, when the number of projects increases, solution time increases exponentially; and problem becomes a combinatorial optimization problem. Thus, using a multi-objective genetic algorithm (MOGA) is appropriate for large-sized problems.

Our proposed algorithm is a preference-based multi-objective genetic algorithm which is based on the pairwise comparisons of the DM and the reference point determined by the DM.

Our **I**nteractive **M**ultiobjective **G**enetic **A**lgorithm (IMGA) is a steady state algorithm in which one solution is produced and evaluated in each iteration. The IMGA works with a fixed size population of solutions (individuals) and interacts with the decision maker (DM) to learn his or her preferences in terms of pairwise comparisons. The population converges toward preferred solutions.

Determining the reference point: Total available budget (*TAB*) and allowable deviation percentage (α) are obtained by interacting with the DM. In the beginning, we set the reference point to $R(r_1, r_2)$ where

$$r_1 = \text{total net flows and } r_2 = -\frac{TAB*(100-\alpha)}{100} \qquad (26)$$

Reference point of cumulative budget is negative since we maximize all objectives. Reference point is allowed to be updated by the DM throughout the algorithm.

Creation of initial population: Initial population should have a predetermined fixed size. In the IMGA**,** a unique solution is randomly generated and its budget is checked, if its budget is in the allowable budget range (*ABR*) it is inserted into the population. The process repeats until the population reaches its predetermined fixed size. We set the population size to 25 in our algorithm.

Fitness function: Our algorithm depends on the reference point provided by the DM. Our fitness function is an achievement scalarizing function that ensures minimization of the deviation from the reference point. Solutions close to the reference point in the desired direction are directly favored by our fitness function. Achievement scalarizing fitness function value of a solution is calculated as follows:

$$\underset{j}{\text{Max}} \; \left(\lambda_j * (r_j - z_j)\right) - \varepsilon \sum_{j=1}^{2} z_j \qquad \textit{for} \;\; j = 1,2 \qquad (27)$$

where λ_j is the weight, r_j is the reference point, and z_j is the objective value of the j^{th} objective. ε is a sufficiently small positive constant and $\sum_j \lambda_j = 1$. Augmentation term $\varepsilon \sum_{j=1}^{2} z_j$ is included in the function to prevent the generation of weakly efficient solutions. λ_j, r_j, and z_j values are scaled in the interval [0, 1] to prevent bias of different ranges.

Selection of the parents: Once the initial population is created, the objectives and fitness function of the solutions are calculated and ranked according to their fitness function values. The best two solutions (the incumbent and the second best) are picked from the population for crossover and mutation.

Creation of the offspring and interactions: Offspring is produced by using crossover and mutations operators. The single point crossover is used. In single point crossover, a cross site on a chromosome is randomly selected, and left side bits of the first parent and right side bits of the second parent of the selected site are combined to produce the new chromosome (offspring) (see Figure 22). In Figure 22, third site along the chromosome length is randomly selected, and first three bits of the first parent and last (fourth) bit of the second parent are combined to produce the offspring. In genetic algorithms, crossover operators work with crossover probability p_c (usually taken between 0.9 and 1 in the literature) to preserve some of the good solutions produced with the reproduction operator. However, we set to p_c=1 since we do not use any reproduction operator in the IMGA.

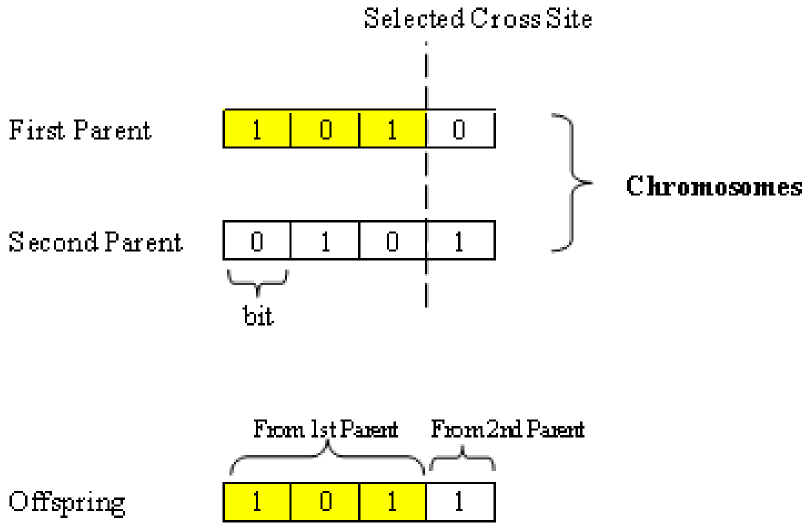


Figure 22 *Generation of the offspring by single point crossover operator*

Besides, we use a bit-wise mutation operator in which each bit has an independent mutation probability p_m. In Figure 23, the bit-wise mutation operator is depicted.. In the literature small mutation probability is used (usually taken 0.01) to maintain diversity as well as not to destroy good solutions. Thus, p_m is set to 0.01.

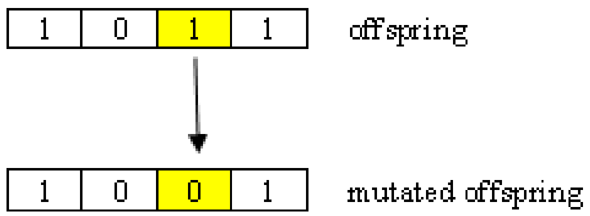


Figure 23 *The bit-wise Mutation Operator*

In the IMGA, once an offspring is created, its budget is checked. If its budget is in the allowable budget range (*ABR*), it is inserted into the population; otherwise a new offspring is created. This process goes on until creating an offspring whose budget is in the *ABR*. After inserting the offspring into the population, fitness function values of the solutions are updated and solutions are ranked according to their fitness values. The solution that has the worst fitness value is eliminated from the population. The incumbent and the second best solution are picked from the population and following four conditions are checked before interacting with DM in order to decrease cognitive load being placed on the DM:

Condition 1. This comparison should not have been done before by DM: If this comparison is previously done by DM, the same solutions are not compared again.

Condition 2.The best two solutions should not be so close to each other: The solutions which are so close to each other according to objective values are not presented to the DM. The incumbent is said to be close to the second best solution if

$$\left| z_k(\text{the incumbent}) - z_k(\text{the second best solution}) \right| \le \mu_k \qquad \forall\, k = 1,2$$

where μ_1 and μ_2 are closeness thresholds for objectives 1 and 2 respectively. These thresholds may either be provided by the DM or set to a small value.

Condition 3. The second best solution should not be close to the solutions not preferred to the current incumbent: DM should not make comparisons that contribute little information to the preference (desired) direction, thus a list of solutions which are not preferred previously kept in memory. The second best solution that is very close to a solution not preferred previously is not presented to the DM. The second best solution is said to be close to one of the previously not preferred solutions of the incumbent, $x^{not-preferred}$, if

$$\left| z_k(\text{the second best solution}) - z_k(x^{not-preferred}) \right| \le \mu_k \qquad \forall\, k = 1,2$$

Condition 4. The best solution should not dominate the second best solution: If the second best solution is dominated by the incumbent, achievement scalarizing weights of the fitness function are updated in favor of the incumbent without interacting with the DM.

If these four conditions are not satisfied, DM is asked to compare the best two solutions. Favorable achievement scalarizing weights of the preferred solution ($\bar{z}$) are calculated according to below formula (Steuer, 1986) and weights of the algorithm updated.

$$\overline{\lambda_k} = \begin{cases} \dfrac{1}{(\overline{z_k} - r_k)} \left[\displaystyle\sum_{j=1}^{2} \dfrac{1}{(\overline{z_j} - r_j)} \right]^{-1} & if \quad \overline{z_j} < r_j \quad for \quad j = 1, 2. \\ 1 & if \quad \overline{z_k} = r_k \\ 0 & \text{Otherwise} \end{cases} \tag{28}$$

In the IMGA, DM responses are used to update the achievement scalarizing weights of the preferred solutions. Thus, solutions in the desired directions (preferred regions) are favored and the algorithm converges toward the preferred solutions. Evolution and interactions repeat in this manner until the DM concludes the search or some predetermined termination condition is realized.

The IMGA is summarized below:

Step 0. Initialization

Ask the DM total available budget (*TAB*) and allowable deviation percentage (α), Set reference point to R (r_1, r_2) according to (26), and create the initial population in the allowable budget range (*ABR*).

Step 1. Evaluation of the initial population

1.1 Calculate the objective value of each solution.

1.2 Calculate the fitness value of each solution by using initial weights according to (27). Equal weights are taken as initial weights.

1.3 Sort the solutions in the population in increasing order of their fitness values. The solution that has the minimum fitness value is determined as the incumbent.

Step 2. Selection of the parents

Select the incumbent and the second best solution for crossover and mutation from the population.

Step 3. Crossover

Perform single point crossover to create an offspring.

Step 4. Mutation

Perform mutation on the offspring with mutation probability.

Step 5. Budget Check

Check the budget of the offspring. If it is in the *ABR*, insert the offspring into population, and eliminate solution with the maximum fitness value. Otherwise go to Step 3.

Step 6. Interactions with the DM.

6.1 Ask the DM to compare the best two solution in the population if

- this comparison was not done before by DM or
- the best two solutions are not so close to each other or
- the second best solution is not so close to the previously not preferred solutions of current incumbent or

- the best solution does not dominate the second best solution,
- Otherwise go to Step 3.

6.2 Suppose DM prefers one solution to another, update the achievement scalarizing function weights in the favor of preferred solution ($\bar{z}$) according to (28).

6.3 Ask the DM whether s/he wants to change the reference point, if yes, update the reference point.

Step 7. Updating fitness function values

Calculate the fitness value of all solutions in the population according to (27).

Step 8. Present the current incumbent to the DM, if

8.1 the DM is satisfied with the incumbent or

8.2 the DM has made a predetermined number of pairwise comparisons or

8.3 a predetermined number of crossovers have been realized, then stop.

Otherwise go to step 2.

CHAPTER 7

COMPUTATIONAL EXPERIMENTS FOR THE IMGA

In this chapter, we use the IMGA, NSGA-II with constraint tournament method, and ε-constraint method to solve our second proposed mathematical model discussed in Chapter 6. We compare results of IMGA with those of ε-constraint method for small size problems and with those of NSGA-II with constraint tournament method for large size problems. The IMGA is coded with the Java programming language on the Net Beans IDE 6.1 software. C codes of NSGA-II with constraint tournament method are available at www.iitk.ac.in/kangal/codes.shtml. C Codes are used with some modifications on BloodShed DevC++ software version 4. Runs are performed on Intel Pentium IV, 2.30 GHz, and 512 MB of RAM computer.

7.1 GENERATION OF THE PROBLEMS

We test the performance of the IMGA and NSGA-II with constraint tournament method on four problem sets. Each problem set consists of ten different problems. Each problem has ten criteria, weights of the criteria are assumed to be equal, and all criteria are assumed to be maximized. Linear (Type V) PROMETHEE preference function is used for all criteria. Budget, criteria, indifference and preference threshold values are generated according to the uniform distribution. The properties of problem sets are given in Table 9. Net flow values are calculated and normalized.

Pareto front of the problem sets 1, 2, and 3 are generated with ε-constraint method. Table 10 reports averages and standard deviations of the solution times of the ε-constraint method for each problem set. Average solution time of problem set 1 is 5.4 minutes, problem set 2 is 35.91 minutes, and problem set 3 is 77.76 minutes. However, problem set 4 cannot be solved with ε-constraint method in 120 minutes.

Table 9 Problem Sets Characteristics

	Problem Set 1	Problem Set 2	Problem Set 3	Problem Set 4
Number of Problems in Each Set	10	10	10	10
Number of Projects in Each Problem	100	250	350	500
Number of Criteria in Each Problem	10	10	10	10
Criterion Weight Values in Each Problem	0.1	0.1	0.1	0.1
Preference Function Type for All Criteria	Type V	Type V	Type V	Type V
Available Budget in Each Problem	7500	18500	27500	50000
Allowable Budget Deviation Percentage in Each Problem	10%	10%	10%	10%

Table 10 also presents the average number of non-dominated solutions found by the ε-constraint method. The number of non-dominated solutions increases considerably when the problem size increases. On the average, 279.6 solutions are found for problem set 1, 1365.7 solutions are found for problem set 2 and 2635.4 solutions are found for problem set 3. Note that only Pareto front solutions falling in allowable budget range are generated in these experiments. Thus, generation of the whole Pareto front is expected to take much longer time.

Table 10 *Results of the initial experiments*

Problem Set	Solution Time (in minutes)		Average Number of non-dominated solutions
	Average	Standard Deviation	
1	5.40	1.32	279.6
2	35.91	10.70	1365.7
3	77.76	13.06	2635.4
4	*	*	*

* Not solved in 120 minutes.

7.2 PERFORMANCE METRICS

We test the performance of the IMGA with the following formula:

$$\delta = 100 * \left[\frac{U(x^{IMGA}) - U(x^{optimal})}{U(x^{optimal})} \right]$$

where δ is the deviation from optimal (in percentage), $x^{optimal}$ is the optimal solution, and x^{IMGA} is the solution obtained by the IMGA, $U(x)$ is the underlying utility function value of solution x.

We first test the performance of the NSGA-II with constraint tournament method with Hyper Volume Ratio (HVR), and Inverted Generational Distance (IGD) metrics.

Hypervolume (HV) metric is proposed by Zitzler and Thiele (1998). In our problem, HV gives the total area in the two dimensional objective space dominated by the non-dominated solutions with respect to a given reference point. HV metric provides both convergence and diversity information of the population. A bad solution or the nadir point is usually taken as a reference point (N).

$$\text{HV} = \bigcup_{i=1}^{|nP|} A_i^{\text{N}}$$

where $|nP|$ is the number of non-dominated solutions in the NSGA-II final population, A_i^N is the area dominated by solution $i \in nP$ with respect to a reference point N. Area of non-dominated solutions for our proposed model is illustrated in Figure 24. In Figure 24, for instance, the area dominated by unique solution e is shown in the dashed area.

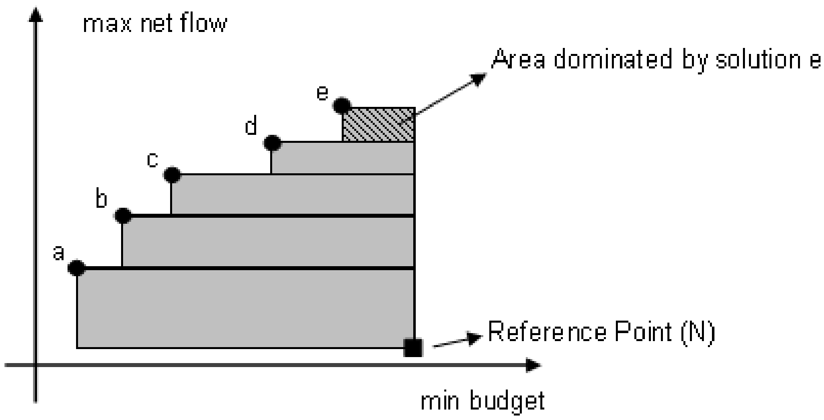


Figure 24 *Area enclosed by the non-dominated solutions*

Hyper Volume Ratio (HVR) is another metric which is computed based on HV metric. HVR is defined as follows:

$$\mathrm{HVR} = \frac{\mathrm{HV}^{nP}}{\mathrm{HV}^{PF}}$$

where HV^{nP} is the hypervolume of the non-dominated solutions of the final population (*nP*) of the NSGA-II, and HV^{PF} is the hypervolume of the Pareto front. Large HVR values (>0.95) imply a good approximation of the Pareto front.

Inverted Generational Distance (IGD) is developed by Bosman and Thiernes (2003). IGD finds the Euclidean distance of each non-dominated solution found by NSGA-II to its closest Pareto front solution. Small IGD values imply a good approximation of the Pareto front. IGD is defined as follows:

$$\mathrm{IGD} = \frac{\sum_{i \in nP} \left(\min_{j \in PF} \left\| z^i - z^j \right\|_2 \right)}{|nP|}$$

where *nP* is the non-dominated solutions found by the NSGA-II, *PF* is the Pareto front of the problem; z^i is the solution found by the NSGA-II, z^j is the solution in the Pareto front, and $\left\| z^i - z^j \right\|_2$ is the Euclidean distance between solutions z^i and z^j.

7.3 PARAMETER SETTING

We use a binary chromosome representation in the IMGA and NSGA-II with constraint tournament method. The length of chromosome is equal to *n*, which is the number of projects in the problem.

1	**0**	**0**	**1**

Figure 25 *Representation of a Solution*

In Figure 25, a chromosome (a solution) has four bits, which means that there are four projects in the problem. First bit is "1" indicating that first project is selected; second bit is "0" indicating that second project is not selected in the solution, and so on.

7.3.1 Parameter Setting for IMGA

In the IMGA, we use the following linear and Chebyshev utility functions to simulate the preferences of the DM:

$$\text{Linear} \qquad U(x) = \sum_{j=1}^{2} \lambda_j * (z_j^* - z_j(x))$$

$$\text{Chebyshev} \qquad U(x) = \underset{j=1,2}{Max} \left\{ \lambda_j * (z_j^* - z_j(x)) \right\}$$

where z_j^* is the ideal point and λ_j is the weight of the j^{th} objective; and $z_j(x)$ is the j^{th} objective value of the solution *x*. For each problem, objective (maximizing total net flow and minimizing total budget) weights of the utility function are generated according to the uniform distribution in the interval [0, 1].

As discussed in Section 6, crossover (p_c) and mutation probabilities (p_m) are set to 1 and 0.01, respectively; and population size is set to 25. The IMGA is sensitive to the closeness thresholds, hence preliminary experimental runs are done on problem sets 1 and 3 for parameter setting of IMGA. Five problems are selected from problem sets 1 and 3. 15 parameter sets are tested in problem set 1 and 9 parameter sets are tested in problem set 3 as shown in Table 11. For each parameter set and for each problem, five runs are performed; hence, totally 600 runs are performed. Parameter sets are given in Table 11.

Table 11 *Parameters for the IMGA*

	Number of Crossovers	*Closeness Thresholds*
Parameter set 1	10,000	0.005
Parameter set 2	10,000	0.01
Parameter set 3	10,000	0.02
Parameter set 4*	10,000	0.03
Parameter set 5*	10,000	0.04
Parameter set 6	25,000	0.005
Parameter set 7	25,000	0.01
Parameter set 8	25,000	0.02
Parameter set 9*	25,000	0.03
Parameter set 10*	25,000	0.04
Parameter set 11	50,000	0.005
Parameter set 12	50,000	0.01
Parameter set 13	50,000	0.02
Parameter set 14*	50,000	0.03
Parameter set 15*	50,000	0.04

* Only tested in problem set 1.

The deviations from the optimal, number of comparisons, and solution time of all these runs (averages and standard deviations) are given in Tables 12 and 13. As seen in Tables 12 and 13, for problem set 1, parameter 14 and for problem set 3, parameter 11 give the best results in terms of deviation from the optimal solution. For all problem sets, it is observed that 50,000 crossovers are suitable for the termination condition of the IMGA.

***Table 12** Results of the Parameter Setting Runs for Problem Set 1*

	Deviation from optimal		Number of Comparisons		Solution Time (in seconds)	
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**
Parameter 1	8.53	2.44	36.68	22.95	5.48	0.477
Parameter 2	5.95	1.71	13.7	9.44	4.84	0.45
Parameter 3	5.10	1.45	2.48	1.77	4.80	0.51
Parameter 4	3.43	1.63	0.49	0.33	5.28	0.69
Parameter 5	5.21	2.34	0.08	0.17	5.56	0.56
Parameter 6	7.05	2.29	47	20.93	11.96	1.14
Parameter 7	5.45	2.90	15.4	9.58	10.23	0.55
Parameter 8	3.85	2.38	1.44	1.17	10.16	1.38
Parameter 9	2.98	1.40	0.92	1.34	12.20	1.55
Parameter 10	5.35	1.20	0.12	0.26	12.04	1.06
Parameter 11	6.82	1.20	39.72	20.17	21.04	2.34
Parameter 12	4.39	1.65	13.32	8.86	19.00	1.46
Parameter 13	2.83	1.68	1.00	1.07	19.76	2.77
Parameter 14	1.71	1.70	0.56	0.57	21.24	2.78
Parameter 15	4.62	1.75	0.04	0.09	21.80	1.83

***Table 13** Results of the Parameter Setting Runs for Problem Set 3*

	Deviation from optimal		Number of Comparisons		Solution Time (in seconds)	
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**
Parameter 1	3.00	0.90	10.56	2.41	34.08	2.36
Parameter 2	3.98	1.46	1.68	1.20	32.48	2.94
Parameter 3	8.68	2.20	0.40	0.38	35.40	2.06
Parameter 6	2.06	1.46	11.95	6.13	81.52	10.00
Parameter 7	3.46	1.01	1.96	1.32	79.48	2.42
Parameter 8	7.04	2.87	0.40	0.66	82.56	3.75
Parameter 11	1.40	0.57	14.92	7.48	180.16	5.44
Parameter 12	2.87	0.84	1.40	1.08	176.96	9.25
Parameter 13	7.07	1.98	0.20	0.3085	191.12	9.109

For problem set 1, closeness thresholds are set to 0.03; for problem set 3 and 4, closeness thresholds are set to 0.005; for problem set 2, closeness thresholds are set to 0.01. Different closeness thresholds are used for different-sized problems since objectives are scaled in the interval [0, 1] in all problems, and same interval length corresponds to different number of projects for different problem sizes. For example, same interval length may represent the 100 projects in problem set 1, and 500 projects in problem set 4.

Number of comparisons considerably increases when the problem size is 350. Number of comparisons is on the average 0.56 with standard deviation 0.57 for problem set 1 and on the average 14.92 with standard deviation 7.48 for problem set 3. Number of comparisons of the selected parameter is one of the smallest for the problem set 1, whereas number of comparisons of the selected parameter is the highest for the problem set 3. Average solution time is 21.24 seconds for problem set 1 and 180.16 seconds for problem set 3. Solution times of the selected parameters are the highest ones for the problem sets 1 and 3. However, these solution times are considerably small when compared to the exact solution time of the problems.

7.3.2 Parameter Setting for NSGA-II with Constraint Tournament Method

In the literature, single point crossover and uniform crossover operators are usually performed in NSGA-II with probability between 0.9 and 1 and mutation operator is applied usually with probability between 0.002 and 0.02. In uniform crossover an offspring is produced by selecting the every bit with probability *p* (usually taken 0.5) from either parent (Deb, 2001). In the preliminary experiments, population size is set to 50 for problem set 1, 80 for problem set 2, and 100 for problem sets 3 and 4. Other parameters used in the runs are given in Table 14. Iteration size is taken 500.

Table 14 *Parameters for the NSGA-II with constraint tournament method*

	Crossover Type	**Crossover Probability**	**Mutation Probability**
Parameter Set I	Single Point Crossover	1	0.002
Parameter Set II	Uniform Crossover	1	0.01

We select one problem from problem set 1, and one problem from problem set 4. We only generate the Pareto front of the problem taken from the problem set 4 in 3 hours 53 minutes 25 seconds to have an idea about the NSGA-II performance on large-sized problems. For the problem taken from the problem set 1, we solve it with different allowable budget deviation percentages to have an idea about the performance of the constraint tournament mechanism of the NSGA-II. Each problem is solved 10 times with NSGA-II constraint tournament method. Totally 30 runs are performed. We know that number of test problems is not adequate for parameter settings, but they give us some idea about the performance of NSGA-II. In Table 15, average HVR values of the runs are shown. Parameter set I is selected to be used in the NSGA-II with constraint tournament method runs.

Table 15 *Parameter Settings for NSGA-II with constraint tournament method*

Problem Size	Allowable Budget Deviation Percentage	Average HVR Values	
		Parameter Set I	Parameter Set II
100	10%	0.9832	0.9815
100	20%	0.9884	0.9675
500*	10%	0.9881	0.7176

7.4 COMPUTATONAL RESULTS

We solve each problem in problem sets five times with the IMGA and NSGA-II with constraint tournament method using the above selected parameters. Totally, 400 runs are performed.

Performance of the NSGA-II with constraint tournament method for problem sets 1, 2, and 3 are reported in Table 16. Average of HVR is about 98% and average of IGD is about 0.0015. HVR and IGD values of the NSGA-II with constraint tournament method imply a good approximation to the true Pareto front. For problem set 4, five runs of the NSGA-II with constraint tournament method are combined and estimated Pareto front is obtained by the non-dominated solutions of the combined set.

Table 16 *HVR and IGD values of problem sets 1, 2 and 3*

	Problem Set I		Problem Set II		Problem Set III	
	HVR	IGD	HVR	IGD	HVR	İGD
Average	0.9838	0.0016	0.984	0.0014	0.9808	0.0017
Standard Deviation	0.005	0.0019	0.0020	0.0009	0.0095	0.0027

Computational results of the IMGA are reported in Table 17. Deviations from optimal or estimated front (for problem size 500) are less than 3.98% for linear utility functions and less than 5.45% for Chebyshev utility functions. Standard deviations of the deviations from optimal are less than 1.84% for linear utility functions, and less than 2.06% for Chebyshev utility functions. In linear utility functions, average deviation from optimal is 1.49% with standard deviation 0.98 for problem set 1, 1.82% with standard deviation 0.82 for problem set 2, 1.93% with standard deviation 0.83 for problem set 3, and average deviation from estimated front is 3.98% with standard deviation 1.84 for problem set 4. In Chebyshev utility functions, average deviation from optimal is 3.05% with standard deviation 1.06 for problem set 1, 3.26% with standard deviation 1.22 for problem set 2, 3.65% with standard deviation 1.14 for problem set 3, and average deviation from estimated front is 5.45% with standard deviation 2.06 for problem set 4. Deviations of both linear and Chebyshev utility functions imply good approximations to the optimal solutions. IMGA finds good solutions especially for problem set 1, 2, and 3. Besides, deviations of the linear utility functions are smaller than that of Chebyshev utility functions.

Number of comparisons is less than 14 for linear utility functions and less than 13 for Chebyshev utility functions. As we stated before, our IMGA is sensitive to closeness thresholds, number of comparisons of the problems with size 500, that is 8.67, is smaller than number of comparisons of the problems with size 350, 12.40. Note that we take the closeness thresholds of problems with size 350 and 500 equal, thus number of comparisons of the problems with size 500 decreases, but average deviation from optimal (estimated front) increases.

Table 17 Computational Results for IMGA

	Problem size	100		250		350		500*	
U(x)	**Measure**	**Avg**	**Std dev**	**Avg**	**Std dev**	**Avg**	**Std dev**	**Avg**	**Std dev**
Linear	**Deviation from optimal (δ)**	1.49	0.98	1.82	0.82	1.93	0.83	3.98	1.84
	Comparisons (total)	0.64	0.49	3.80	1.73	13.60	5.09	8.22	2.95
Chebyshev	**Deviation from optimal (δ)**	3.05	1.06	3.26	1.22	3.65	1.14	5.45	2.06
	Comparisons (total)	0.52	0.58	3.88	1.47	12.40	5.72	8.67	3.06

* Deviation from estimated front is calculated.

We also make an attempt to compare our algorithm with NSGA-II with constraint tournament method. For each problem in problem sets, five runs are performed with the NSGA-II with constraint tournament method and IMGA. We calculate the best deviation of each run of NSGA-II with constraint tournament method with respect to utility functions used in the IMGA. Average of the best deviations for NSGA-II with constraint tournament method and IMGA are reported in Table 18. Note that solutions found by NSGA-II are in the allowable budget range since NSGA-II uses constraint tournament method for constraint handling.

In Table 18, in linear utility functions, average of best deviation of NSGA-II with constraint tournament method is 0.64 for problem set 1, 2.41 for problem set 2, 3.23 for problem set 3 and average of best deviation of IMGA is 1.49 for problem set 1, 1.82 for problem set 2, 1.93 for problem set 3. So, in problem sets 2 and 3, averages of best deviation of IMGA are less than those of NSGA-II with constraint tournament method for linear utility functions. However, in problem set 1; averages of best deviation of IMGA are greater than that of NSGA-II with constraint tournament method for linear utility. In Chebyshev utility functions, average of best deviation of NSGA-II with constraint tournament method is 0.78 for problem set 1, 3.93 for problem set 2, 7.59 for problem set 3, and average of best deviation of IMGA is 3.05 for problem set 1, 3.26 for problem set 2, 3.65 for problem set 3. Averages of best deviation of IMGA are less than those of NSGA-II with constraint tournament method for Chebyshev utility functions in problem sets 2 and 3. However, average of best deviation of NSGA-II with constraint tournament method is better than that of IMGA for problem set 1. In both utility functions, performance of the IMGA deteriorates for the problem sets 1. This may be stem from the indifference thresholds since IMGA is very sensitive to them. For problems with size 500, deviation from estimated front is calculated, and average of best deviation of IMGA is 3.98 for linear utility functions and 5.45 for Chebyshev utility functions. NSGA-II with constraint tournament method finds better solutions for problem set 4 since deviation from estimated front is calculated for IMGA and these deviations are non-negative. In both utility functions, performance of the IMGA also deteriorates for the problem set 4. This may be stem from the indifference thresholds or termination condition used in the problem set 4.

Table 18 *Comparison of NSGA-II with constraint tournament method and IMGA in terms of deviation from optimal*

		NSGA-II with Constraint Handling	IMGA
Problem Size	**Utility (U(x))**	**Best**	**Average**
100	Linear	0.64	1.49
	Chebyshev	0.78	3.05
250	Linear	2.41	1.82
	Chebyshev	3.93	3.26
350	Linear	3.23	1.93
	Chebyshev	7.59	3.65
500*	Linear	**	3.98
	Chebyshev		5.45

*: Deviation from estimated front is calculated
**: True Pareto front is not known

Comparison of solution times for NSGA-II with constraint tournament method and IMGA are presented in Table 19. Average solution time of IMGA is better than that of NSGA-II with constraint tournament method. Average solution time of IMGA is about 18 seconds for problem set 1, 77 seconds for problem set 2, 173 seconds for problem set 3, and 291 seconds for problem set 4. Average solution time of NSGA-II with constraint tournament method is 37.93 seconds for problem set 1, 168.79 seconds for problem set 2, 197.62 seconds for problem set 3, and 497.84 seconds for problem set 4. However, we definitely conclude that solution time of both NSGA-II with constraint tournament method and IMGA are very small compared to exact solution times.

Detailed results of the NSGA-II with constraint tournament method and IMGA are given in Appendix C.

Table 19 *Solution time of the IMGA and NSGA-II with constraint tournament method*

		NSGA-II with constraint tournament method		**IMGA**	
Problem Size	**Utility (U(x))**	**Average**	**Standart deviation**	**Average**	**Standart deviation**
100	Linear	37.93	2.27	19.04	2.07
	Chebyshev			17.32	1.68
250	Linear	168.79	14.84	77.36	7.39
	Chebyshev			77.62	7.10
350	Linear	197.62	6.49	174.94	3.99
	Chebyshev			172.54	4.52
500	Linear	497.84	94.04	293.86	4.89
	Chebyshev			289.97	4.46

CHAPTER 8

CONCLUSION AND FURTHER RESEARCH

PROMETHEE V is one of the most used methods in MCDM. In this study, we formulated two biobjective models based on PROMETHEE V method for project selection problems. We discussed the drawback of budget constraint in project selection problems and developed two mathematical models to avoid the drawback.

In the first proposed mathematical model, objective functions correspond to the PROMETHE I flows, namely leaving and entering flows. To solve this mathematical model, we develop an interactive approach, called ib-PROMETHEE. This approach consists of two stages. In the first stage, we solve the biobjective model using ε-constraint approach to generate the Pareto front. In the second stage, we use data mining techniques to obtain useful information about Pareto front and present this useful information to the decision maker to help her/him during the selection process.

In the second proposed mathematical model, one objective is to maximize the total net flow of the selected projects, and other objective is to minimize the cumulative budget of selected projects. We apply a well known heuristic algorithm, NSGA-II with constraint handling mechanism to this second model. We also develop a Preference Based Interactive Genetic Algorithm (IMGA) to solve this model. We test the performance of above algorithms on randomly generated test problems. Computational experiments show that solutions obtained by both NSGA-II with constraint handling method and IMGA imply good approximations to optimal solutions.

In this study, we assume that projects can be funded fully. As a future work, above biobjective models can be reformulated assuming that projects can be partially funded. Furthermore, three-objective mathematical models in which objectives correspond to PROMETHEE I flows and cumulative budget of selected projects may be studied as a future work.

REFERENCES

Abu-Taleb, M., Mareschal, B., 1995. Water resources planning in the middle east: Application of the PROMETHEE V multicriterion method. *European Journal of Operational Research* 81, 500–511.

Al-Kloub, B., A1-Shemmeri, T., Pearman, A., 1997. The role of weights in multi-criteria decision aid, and the ranking of water projects in Jordan. *European Journal of Operational Research 99*, 278-288

Al-Rashdan, D., Al-Kloub, B., Dean, A., Al-Shemmeri, T., 1999. Environmental impact assessment and ranking the environmental projects in Jordan. *European Journal of Operational Research* 118, 30-45

Al-Shemmeri, T., A1-Kloub, B., Pearman, A., 1997. Computer aided decision support system for water strategic planning in Jordan. *European Journal of Operational Research* 102, 455-472

Alves M.J., Almeida, M., 2007. MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem, *Computers & Operations Research* 34, 3458-3470.

Beume, N., Naujoks, B., Emmerich, M., 2007. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal of Operational Research* 181(3), 1653–1669.

Bosman P. A. N., Thiernes D., 2003. The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 7(2) 174-188.

Branke, J., Kaußler, T., and Schmeck, H., 2001. Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software* 32, 499–507.

Brans, J.P., Vincke, P., 1985. PROMETHEE: A new family of outranking methods in MCDM. *Management Science* 31(6), 647–656.

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. *Chapman& Hall*.

Brans, J.P., Mareschal, B., 1992. PROMETHEE V: MCDM problems with segmentation constraints. *INFOR* 30: 85–96.

Brans, J.P., Mareschal B., 1994. The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decision Support Systems* 12, 297-310.

Coello C.A.C, 2000. Handling Preferences in Evolutionary Multiobjective Optimization: ASurvey. *In Proc. of the 2000 Congress on Evolutionary Computation*, 30–37, Piscataway, NJ.IEEE Service Center.

Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. *Wiley, Chichester*.

Deb, K., Agrawal, S., Pratap, S., Meyarivan, T., 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, INDIA*.

Deb, K., Kumar, A., 2007. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. *Genetic and Evolutionary Computation Conference (GECCO'2007)* volume 1 , 781–788, London, UK. ACM Press.

Deb, K., Kumar, A., 2007. Light Beam Search Based Multi-objective Optimization using Evolutionary Algorithms. *Congress on Evolutionary Computation (GEC 2007)*, volume 1, 2125-2132

Deb, K., Sundar, J., 2006. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In et al., M. K., editor, 2006. *Genetic and Evolutionary Computation Conference (GECCO'2006)* volume 1, 635–642, Seattle, Washington, USA. ACM Press.

Doerner, K., Gutjahr, W.J., 2004. Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. *Annals of Operations Research* 131, 79-99.

Haimes, Y.Y., Ladson L.S., Wismer D.A., 1971. Bicriterion Formulation of Problems of Integrated System Identification and System Optimization. *IEEE Transactions On Systems Man And Cybernetics.* SMC1(3) 296-&.

Ishibuchi, H., Tsukamoto, N., Nojima, Y, 2008. Evolutionary many-objective optimization: A short review. *Proceedings of 2008 IEEE Congress on Evolutionary Computation*, 2424-2431, Hong Kong.

Karademir, S., 2008. The Biobjective Traveling Salesman Problem With Profits. *Master Thesis, Middle East Technical University, Dept. of Industrial Engineering.*

Karahan, I., Koksalan, M., 2008. Territory Defining Evolutionary Algorithm and Preference Incorporation, Technical Report, *Middle East Technical University, Dept. of Industrial Engineering.*

Koksalan, M., S., Phelps, 2007. An Evolutionary Metaheuristic for Approximating Preference-Non-dominated Solutions. *INFORMS Journal on Computing* 19 (2), 291-301.

Mavrotas, G., Diakoulaki, D., Caloghirou, Y., 2006. Project prioritization under policy restrictions. A combination of MCDA with 0–1 programming. *European Journal of Operational Research* 171, 296–308.

Medaglia, L., Graves, S.M, Ringuest, J.L., 2007. A Multiobjective Evolutionary Approach for Linearly Constrained Project Selection Under Uncertainty. *European Journal of Operation Research* 179, 869-894.

Morais, D.C, Almeida, A.T, 2007. Group decision-making for leakage management strategy of water network. *Resources, Conservation and Recycling* 52, 441–459.

Ozerol, G., Karasakal, E., 2008. Interactive outranking approaches for multicriteria decision-making problems with imprecise information. *Journal of the Operational Research Society* 59, 1253-1268.

Pandey, P.C, Kengpol, A., 1995. Selection of an automated inspection system using multiattribute decision analysis. *International Journal of Production Economics* 39, 289-298.

Pfeiffer, J., Golle, U., Rothlauf, F., 2008. Reference Point Based Multi-Objective Evolutionary Algorithms for Group Decisions. *In Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO-08)*, 697–704.

Phelps, S.P., Koksalan, M., 2003. An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization. *Management Science* 49(12), 1726–1738.

Rachmawati, L., Srinivasan, D., 2006. Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey, 2006. *2006 IEEE Congress on Evolutionary Computation*.

Soylu, B., Köksalan, M., 2008. A FavorableWeight Based Evolutionary Algorithm for Multiple Criteria Problems. *IEEE Transactions on Evolutionary Computation*. Forthcoming.

Steuer, R.E., 1986. *Multiple criteria optimization: theory, computation, and application.* John Wiley & Sons, Inc., New York.

Thiele, L., Miettinen, K., Korhonen, P.J, Molina, J., 2007. A Preference Based Interactive Evolutionary Algorithm for Multiobjective Optimization. *Helsinki School of Economics Working Papers*, W-412.

Vincke, P., 1992. *Multicriteria Decision-Aid. Wiley: Chichester*.

Zitzler, E., Thiele, L., 1998. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. *Parallel Problem Solving From Nature - Ppsn V* 1498 292-301.

Zitzler E., Laumanns M., and Thiele L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization, *Technical Report in Evolutionary Methods for Design, Optimization and Control*, *Barcelona.*

Zitzler, E., Kunzli, S., 2004. Indicator-based Selection in Multiobjective Search. In et al., X. Y., editor, Parallel Problem Solving from Nature - *PPSN VIII, volume 3242 of Lecture Notes in ComputerScience*, 832–842, Birmingham, UK. Springer-Verlag.

APPENDIX A

SPSS CLEMENTINE DATA MINING SOFTWARE

In this study, we use the CRT model and necessary graph tools. In Figure 26, the modeling interface of the software is depicted. "130eff.txt" icon represents the data (Pareto front) you feed into program; "Filter" icon represents selection of critical projects and objectives, "CRT Leaving" icon represents the construction of CRT modeling with the target objective leaving. Triangle icons represent the various graph forms that can be used in the analysis.

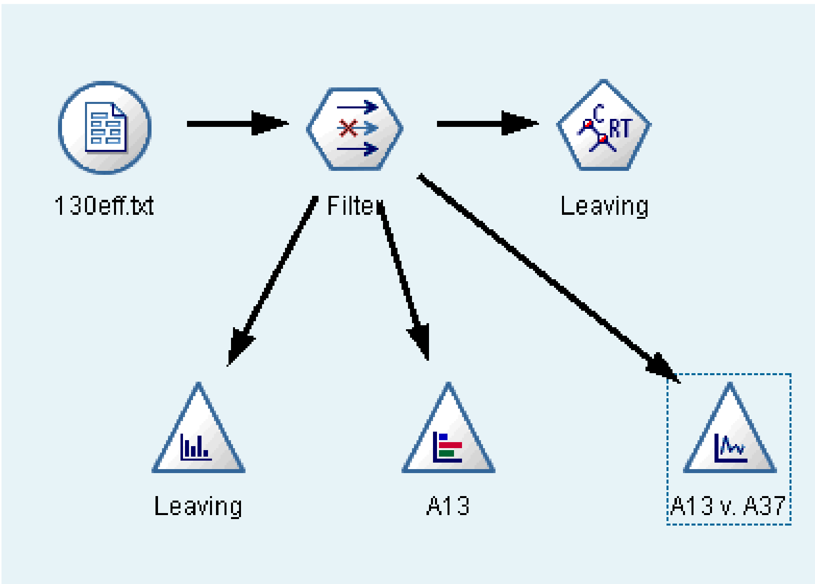


Figure 26 *SPPS Clementine modeling interface*

When you run the CRT, you obtain the decision tree (see Figure 27 in the graph form and Figure 28 in the list form). For example, in Figure 27, "node 0" reports that there are 130 solutions with the average leaving objective value of 10.628. "node 8" reports that the number of solutions including project A41 is 94 with the average leaving objective value of 10.841. In Figure 28, effect means that the difference between the current node and its parent (up) node in terms of leaving flow, numbers in the parenthesis indicate the number of solutions in the corresponding nodes.

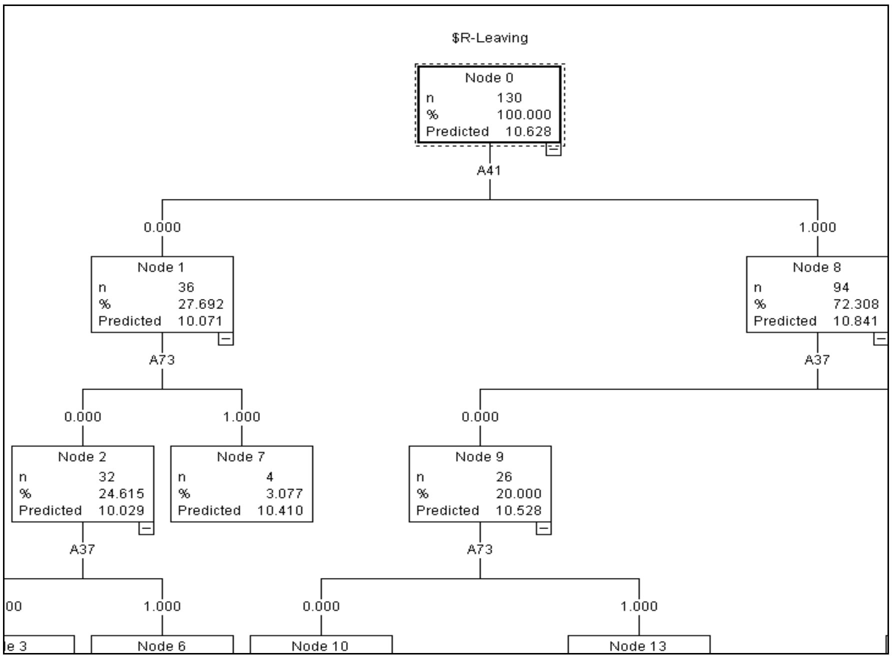


Figure 27 Illustration of CRT graph

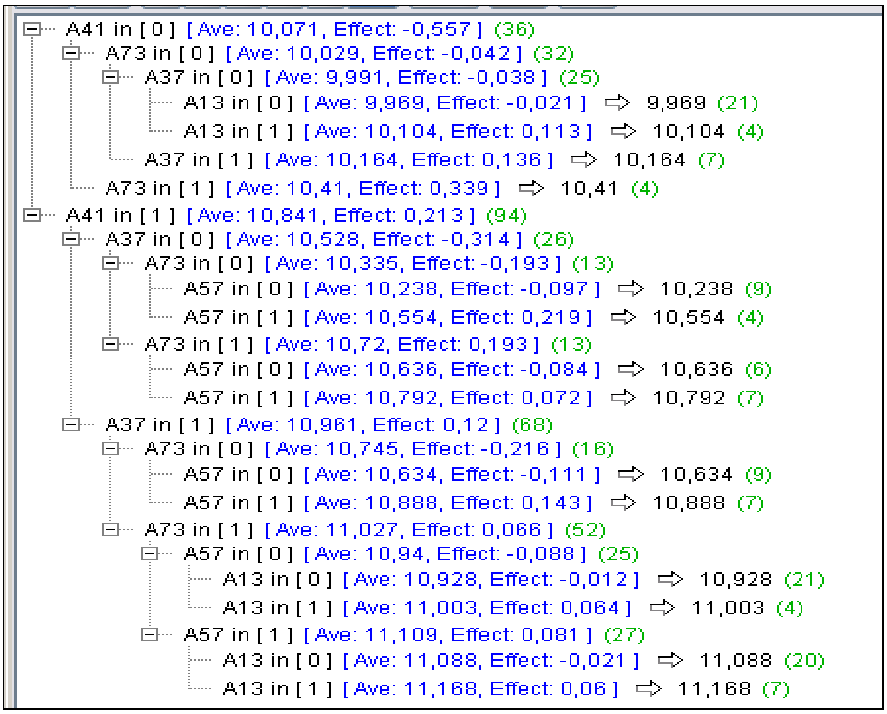


Figure 28 Illustration of the CRT list

APPENDIX B

RANDOMLY GENERATED EXAMPLE

Table 20 *Candidate Solutions of Randomly Generated Example*

Project\ Solution	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**F***
A1	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0	0	0	0	0
A3	1	1	1	1	1	0	1	1	1	1	1	10
A4	1	1	1	1	1	1	1	1	1	1	1	11
A5	1	1	1	1	1	1	1	1	1	1	1	11
A6	0	0	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	0	0	0	0	0	0
A9	1	1	1	1	1	1	1	1	1	1	1	11
A10	1	1	1	0	0	0	1	1	0	0	0	5
A11	0	0	0	0	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0
A17	1	1	1	1	1	1	1	1	1	1	1	11
A18	1	1	1	1	1	1	1	1	1	1	1	11
A19	0	0	0	0	0	0	0	0	0	0	0	0
A20	0	0	0	0	0	0	0	0	0	0	0	0
A21	0	0	0	0	0	0	0	0	0	0	0	0
A22	0	0	0	0	0	0	0	0	0	0	0	0
A23	0	0	0	0	0	0	0	0	0	0	0	0
A24	0	0	0	0	0	0	0	0	0	0	0	0
A25	0	0	0	0	0	0	0	0	0	0	0	0
A26	0	0	0	0	0	0	0	0	0	0	0	0
A27	0	0	0	0	0	0	0	0	0	0	0	0
A28	1	1	1	1	1	1	1	1	1	1	1	11
A29	0	0	0	0	0	0	0	0	0	0	0	0
A30	0	0	0	0	0	0	0	0	0	0	0	0
A31	0	0	0	0	0	0	0	0	0	0	0	0
A32	1	1	1	1	1	1	1	1	1	1	1	11
A33	0	0	0	0	0	0	0	0	0	0	0	0
A34	1	1	1	1	1	1	1	1	1	1	1	11
A35	0	1	1	1	1	1	0	1	1	1	1	9
A36	1	1	1	1	1	1	0	0	0	1	1	8
A37	0	0	0	0	0	0	0	0	0	0	0	0
A38	1	1	1	1	1	1	1	1	1	1	1	11
A39	0	0	0	0	0	0	0	0	0	0	0	0
A40	0	0	0	0	0	0	0	0	0	0	0	0
A41	0	0	0	0	0	0	0	0	0	0	0	0
A42	0	0	0	0	0	0	0	0	0	0	0	0

***Table 20** Candidate Solutions of Randomly Generated Example (Continues)*

Project\ Solution	1	2	3	4	5	6	7	8	9	10	11	F*
A43	0	0	0	0	0	0	0	0	0	0	0	0
A44	0	0	0	0	0	0	0	0	0	0	0	0
A45	0	0	0	0	0	0	0	0	0	0	0	0
A46	0	0	0	0	0	0	0	0	0	0	0	0
A47	1	1	1	1	1	1	1	1	1	1	1	11
A48	0	0	0	0	0	0	0	0	0	0	0	0
A49	0	0	0	0	0	0	0	0	0	0	0	0
A50	0	0	0	0	0	0	0	0	0	0	0	0
A51	0	0	0	0	0	0	0	0	0	0	0	0
A52	1	1	1	1	1	1	1	1	1	1	1	11
A53	1	1	1	1	1	1	1	1	1	1	1	11
A54	1	1	1	1	1	1	1	1	1	1	1	11
A55	1	1	1	1	1	1	1	1	1	1	1	11
A56	0	0	0	0	0	1	0	0	0	0	0	1
A57	0	0	0	0	0	0	0	0	0	0	0	0
A58	0	0	0	0	0	0	0	0	0	0	0	0
A59	0	0	0	0	0	0	0	0	0	0	0	0
A60	0	0	0	0	0	0	0	0	0	0	0	0
A61	0	0	0	0	0	0	0	0	0	0	0	0
A62	0	0	0	0	0	0	0	0	0	0	0	0
A63	1	1	1	1	1	1	1	1	1	1	1	11
A64	0	0	0	0	0	0	0	0	0	0	0	0
A65	0	0	0	0	0	0	0	0	0	0	0	0
A66	0	0	0	0	0	0	0	0	0	0	0	0
A67	0	0	0	0	0	0	0	0	0	0	0	0
A68	0	0	0	0	0	0	0	0	0	0	0	0
A69	1	1	1	1	0	0	1	1	1	1	0	8
A70	0	0	0	0	0	0	0	0	0	0	0	0
A71	0	0	0	0	1	1	0	0	0	0	0	2
A72	1	1	1	1	1	1	1	1	1	1	1	11
A73	0	0	0	0	0	0	0	0	0	0	0	0
A74	1	1	1	1	1	1	1	1	1	1	1	11
A75	1	1	1	1	1	1	1	1	1	1	1	11
A76	0	0	0	0	0	0	0	0	0	0	0	0
A77	0	0	0	0	0	0	0	0	0	0	0	0
A78	1	0	1	1	1	1	1	0	1	0	1	8
A79	1	1	1	1	1	1	1	1	1	1	1	11
A80	0	0	0	0	0	0	0	0	0	0	0	0
A81	0	0	0	0	0	0	0	0	0	0	0	0
A82	1	1	1	1	1	1	1	1	1	1	1	11
A83	0	0	0	0	0	0	0	0	0	0	0	0
A84	0	0	0	0	0	0	0	0	0	0	0	0
A85	0	0	0	0	0	0	0	0	0	0	0	0
A86	0	0	0	0	0	0	0	0	0	0	0	0
A87	0	0	0	0	0	0	0	0	0	0	0	0
A88	0	0	0	0	0	0	0	0	0	0	0	0
A89	0	0	0	0	0	0	0	0	0	0	0	0
A90	1	1	1	1	1	1	1	1	1	1	1	11
A91	0	0	0	0	0	0	0	0	0	0	0	0
A92	0	0	0	0	0	0	0	0	0	0	0	0
A93	1	1	0	1	1	1	1	1	1	1	1	10
A94	0	0	0	0	0	0	0	0	0	0	0	0
A95	0	0	0	0	0	0	0	0	0	0	0	0

Table 20 *Candidate Solutions of Randomly Generated Example (Continues)*

Project\ Solution	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**F***
A96	0	0	0	0	0	0	0	0	0	0	0	0
A97	0	0	0	0	0	0	0	0	0	0	0	0
A98	1	1	1	1	1	1	1	1	1	1	1	11
A99	0	0	0	0	0	0	0	0	0	0	0	0
A100	1	1	1	1	1	1	1	1	1	1	1	11
Cumulative Budget	5035	5093	5054	5090	5084	4928	4915	4973	4970	4916	4978	55036
Total Entering Flow	6.31	6.31	6.30	6.30	6.27	6.24	6.12	6.12	6.11	6.11	6.10	68.35
Total Leaving Flow	2.66	2.65	2.65	2.64	2.63	2.62	2.56	2.55	2.53	2.52	2.51	28.57
Number of Selected Projects	29	29	29	29	29	29	28	28	28	28	28	314

* Frequency of Projects

APPENDIX C

RESULTS OF THE IMGA AND NSGA-II

Table 21 *Results of the IMGA with Linear Utility*

Linear Utility		Problem Size											
		100			250			350			500		
P*	R*	D*	C*	T*	D	C	T	D	C	T	D	C	T
	1	0.25	0	27	0.89	7	71	1.01	13	178	1.24	11	272
	2	0.39	0	22	1.03	1	69	1.37	12	179	4.15	12	279
	3	0.32	0	22	0.75	4	78	1.04	4	180	5.91	8	284
1	4	0.25	0	30	1.32	6	70	0.9	18	185	2.27	6	280
	5	0.91	0	26	1.67	5	73	0.95	19	176	2.48	7	274
	Avg	0.424	0	25.4	1.132	4.6	72.2	1.054	13.2	179.6	3.21	8.8	277.8
	Std	0.28	0.00	3.44	0.37	2.30	3.56	0.18	5.97	3.36	1.84	2.59	4.82
	1	0.85	0	21	2.69	6	67	0.97	8	198	5.72	8	301
	2	0	0	23	1.08	3	80	1.42	32	200	4.3	11	287
	3	3.69	1	21	0.62	5	78	1.23	18	197	6.03	7	289
2	4	0.65	0	24	0.35	3	81	1.14	33	201	7.72	6	293
	5	0.6	0	26	1.91	2	80	0.44	19	189	1.15	8	294
	Avg	1.158	0.2	23	1.33	3.8	77.2	1.04	22	197	4.984	8	292.8
	Std	1.45	0.45	2.12	0.96	1.64	5.81	0.37	10.51	4.74	2.46	1.87	5.40
	1	2.66	1	21	1.15	2	89	0.94	14	175	3.28	7	305
	2	0.85	0	25	1.04	7	63	0.81	10	170	7.19	15	302
	3	0.1	0	20	3.27	5	62	1.02	9	181	2.21	8	309
3	4	0	0	19	2.1	3	81	1.07	13	177	4.38	7	301
	5	3.47	1	16	2.08	1	71	0.68	6	164	1.23	7	303
	Avg	1.416	0.4	20.2	1.928	3.6	73.2	0.904	10.4	173.4	3.658	8.8	304
	Std	1.57	0.55	3.27	0.90	2.41	11.67	0.16	3.21	6.58	2.30	3.49	3.16
	1	3.61	1	16	2.69	4	67	1.37	11	176	3.98	9	303
	2	2.28	0	19	0.69	6	86	1.58	13	172	5.23	8	302
	3	0.44	0	19	1.34	7	80	1.24	14	175	5.8	10	300
4	4	0.92	0	20	2.29	2	79	1.51	9	169	7.8	7	290
	5	1.35	0	21	2.36	3	86	1.14	10	170	2.08	4	293
	Avg	1.72	0.2	19	1.874	4.4	79.6	1.368	11.4	172.4	4.978	7.6	297.6
	Std	1.26	0.45	1.87	0.83	2.07	7.77	0.18	2.07	3.05	2.13	2.30	5.77
	1	0.81	4	23	1.13	7	91	2.1	14	169	5.68	6	297
	2	2.17	2	17	0.53	3	111	3.09	46	188	7.79	4	290
	3	1.92	0	21	0.51	8	101	5.63	2	171	2.87	5	299
5	4	3.14	2	16	0.77	5	90	2.12	24	175	1.21	7	303
	5	1.01	2	16	0.89	6	93	0.26	14	189	5.81	6	302
	Avg	1.81	2	18.6	0.766	5.8	97.2	2.64	20	178.4	4.672	5.6	298.2
	Std	0.94	1.41	3.21	0.26	1.92	8.84	1.96	16.49	9.48	2.61	1.14	5.17

__Table 21__ Results of the IMGA with Linear Utility (Continues)

Linear Utility		**Problem Size**											
		100			**250**			**350**			**500**		
	1	2.7	0	20	3.41	6	67	7.1	12	177	4.78	6	291
	2	0.36	1	16	3.62	4	72	4.27	16	171	3.94	6	288
	3	1.27	1	15	1.27	7	66	5.01	15	169	1.01	14	286
6	4	3.7	1	15	2.36	6	69	4.39	13	171	5.81	6	282
	5	3.72	1	15	0.33	2	66	4.04	13	179	3.01	9	286
	Avg	2.35	0.8	16.2	2.198	5	68	4.962	13.8	173.4	3.71	8.2	286.6
	Std	1.50	0.45	2.17	1.40	2.00	2.55	1.25	1.64	4.34	1.83	3.49	3.29
	1	2.28	1	15	0.21	7	78	2.05	10	171	4.25	8	308
	2	2.17	1	15	4.22	4	69	1.01	14	168	3.68	5	301
	3	2.28	1	14	5.02	6	100	2.09	12	170	3.94	5	298
7	4	1.75	1	14	1.35	8	94	2.01	10	168	3.15	15	303
	5	2.01	1	15	2.95	3	90	1.78	5	172	3.35	7	306
	Avg	2.098	1	14.6	2.75	5.6	86.2	1.788	10.2	169.8	3.674	8	303.2
	Std	0.22	0.00	0.55	1.99	2.07	12.54	0.45	3.35	1.79	0.44	4.12	3.96
	1	0.29	1	17	3.49	4	69	0.99	15	161	4.91	11	273
	2	0.34	0	21	3.16	1	70	1.33	10	163	5.61	9	281
	3	0.15	1	21	1.21	2	69	1.38	8	162	3.18	14	285
8	4	0.69	0	20	3.28	1	78	1.32	13	165	7.64	9	276
	5	0.25	1	20	2.87	1	89	5.07	9	166	5.58	5	288
	Avg	0.344	0.6	19.8	2.802	1.8	75	2.018	11	163.4	5.384	9.6	280.6
	Std	0.21	0.55	1.64	0.92	1.30	8.69	1.71	2.92	2.07	1.60	3.29	6.19
	1	2.99	0	20	0.95	1	70	2.06	9	169	0.23	18	297
	2	3.4	0	15	0.7	1	69	2.74	8	170	1.29	9	298
	3	0.61	1	17	0.59	2	86	1.23	9	168	1.36	11	303
9	4	1.79	1	16	0.74	1	81	0.56	14	172	2.55	7	302
	5	0.93	1	17	0.93	1	69	1.75	13	174	3.74	7	309
	Avg	1.944	0.6	17	0.782	1.2	75	1.668	10.6	170.6	1.834	10.4	301.8
	Std	1.23	0.55	1.87	0.15	0.45	7.97	0.83	2.70	2.41	1.35	4.56	4.76
	1	2.71	0	17	2.16	1	65	0.99	13	173	5.09	10	304
	2	3.11	1	16	2.58	2	70	1.27	16	171	1.16	6	301
	3	0.69	1	16	2.35	4	66	0.64	12	174	2.72	4	289
10	4	1.24	1	17	3.23	2	74	3.3	15	169	3.82	6	291
	5	0.45	0	17	2.68	2	75	2.9	11	170	5.74	10	295
	Avg	1.64	0.6	16.6	2.6	2.2	70	1.82	13.4	171.4	3.706	7.2	296
	Std	1.20	0.55	0.55	0.41	1.10	4.53	1.20	2.07	2.07	1.84	2.68	6.40

Avg	1.49	0.64	19.04	1.82	3.80	77.36	1.93	13.60	174.94	3.98	8.22	293.86
Std	0.98	0.49	2.07	0.82	1.73	7.39	0.83	5.09	3.99	1.84	2.95	4.89

* P: Problem, R: Run, D: Deviation from optimal, C: Number of Comparison, T: Solution Time (in seconds)

__Table 22__ Results of the IMGA with Chebyshev Utility

Chebyshev Utility		Problem Size											
		100			250			350			500		
P*	R*	D*	C*	T*	D	C	T	D	C	T	D	C	T
1	1	4.5	0	20	0.57	3	78	8.92	6	194	0.09	13	286
	2	2.3	0	19	1.16	4	75	8.9	7	187	0.35	9	278
	3	3.35	1	19	2.34	2	76	9.62	8	179	0.07	8	280
	4	3.55	0	19	1.9	6	77	9.26	5	171	0.81	15	283
	5	2.28	0	16	0.99	7	76	5.98	6	186	-2.18	14	290
	Avg	3.196	0.2	18.6	1.392	4.4	76.4	8.536	6.4	183.4	-0.17	11.8	283.4
	Std	0.93	0.45	1.52	0.72	2.07	1.14	1.46	1.14	8.73	1.16	3.11	4.77
2	1	0.72	0	20	1.81	4	71	1.12	14	169	9.71	6	269
	2	2.74	0	20	0.81	7	69	0.54	13	171	8.94	8	274
	3	1.44	0	10	4.79	3	68	2.37	9	168	7.57	7	281
	4	2.71	1	15	1.63	6	67	1.81	11	170	9.91	8	276
	5	0	1	25	1.01	4	70	0.61	14	181	8.87	10	277
	Avg	1.522	0.4	18	2.01	4.8	69	1.29	12.2	171.8	9	7.8	275.4
	Std	1.21	0.55	5.70	1.61	1.64	1.58	0.79	2.17	5.26	0.92	1.48	4.39
3	1	1.74	2	14	6.09	6	59	3.38	9	170	5.17	15	290
	2	0	0	19	4.13	3	60	3.19	29	169	8.24	6	291
	3	0.64	0	18	5.19	4	69	1.33	14	173	8.41	8	293
	4	1.58	0	19	3.93	3	68	0.68	10	176	2.81	13	289
	5	0.67	0	19	3.21	1	70	0.64	13	170	7.09	10	288
	Avg	0.926	0.4	17.8	4.51	3.4	65.2	1.844	15	171.6	6.344	10.4	290.2
	Std	0.72	0.89	2.17	1.13	1.82	5.26	1.35	8.09	2.88	2.36	3.65	1.92
4	1	4.07	1	14	1.16	5	81	5.18	15	177	7.61	7	279
	2	2.73	0	14	1.2	6	88	7.16	10	175	10.66	4	282
	3	4.23	1	14	1.28	3	82	5.56	12	171	5.73	9	285
	4	3.55	1	16	2.38	4	67	6.13	9	179	0.01	16	291
	5	3.03	0	15	3.41	5	71	5.07	15	168	8.95	7	295
	Avg	3.522	0.6	14.6	1.886	4.6	77.8	5.82	12.2	174	6.592	8.6	286.4
	Std	0.65	0.55	0.89	0.99	1.14	8.58	0.86	2.77	4.47	4.10	4.51	6.54
5	1	3.12	1	26	0.73	6	101	3.13	11	167	8.76	6	287
	2	4.28	0	25	3.66	4	106	0.91	6	173	7.08	3	279
	3	5.62	0	25	0.74	5	102	1.39	6	171	9.91	8	283
	4	5.65	0	23	2.31	5	97	3.81	55	169	10.33	4	286
	5	4.71	0	25	1.98	3	89	1.91	8	181	6.79	5	279
	Avg	4.676	0.2	24.8	1.884	4.6	99	2.23	17.2	172.2	8.574	5.2	282.8
	Std	1.05	0.45	1.10	1.22	1.14	6.44	1.21	21.23	5.40	1.61	1.92	3.77
6	1	3.28	1	14	3.94	6	65	5.15	11	165	4.19	8	279
	2	2.03	1	15	5.17	4	91	5.21	18	166	2.74	11	285
	3	2.01	0	14	2.71	3	84	3.78	16	167	8.41	8	286
	4	4.21	0	15	4.61	4	88	6.98	11	169	6.89	4	281
	5	4.09	1	16	2.31	5	81	3.58	19	171	7.74	5	279
	Avg	3.124	0.6	14.8	3.748	4.4	81.8	4.94	15	167.6	5.994	7.2	282
	Std	1.07	0.55	0.84	1.22	1.14	10.13	1.37	3.81	2.41	2.43	2.77	3.32

***Table 22** Results of the IMGA with Chebyshev Utility (Continues)*

Chebyshev Utility		Problem Size											
		100			250			350			500		
P*	R*	D*	C*	T*	D	C	T	D	C	T	D	C	T
7	1	4.5	1	15	5.53	4	92	4.14	9	178	3.42	4	289
	2	2.35	1	14	3.96	6	89	4.86	10	168	4.29	10	293
	3	0.04	1	15	5.65	3	79	3.03	13	171	6.33	7	300
	4	3.33	1	14	2.53	6	68	1.29	17	174	1.51	12	301
	5	5.2	1	14	5.43	4	73	2.03	11	167	0.01	18	309
	Avg	3.084	1	14.4	4.62	4.6	80.2	3.07	12	171.6	3.112	10.2	298.4
	Std	2.02	0.00	0.55	1.36	1.34	10.23	1.47	3.16	4.51	2.45	5.31	7.73
8	1	0.9	0	20	3.24	2	69	3.31	10	159	5.26	10	299
	2	0.88	0	19	4.11	3	70	3.4	7	163	7.49	4	293
	3	3.88	3	14	4.55	1	76	0.58	11	167	6.08	7	295
	4	0.83	1	19	4.97	6	81	2.08	8	169	3.33	9	296
	5	3.26	0	17	3.69	4	78	2.27	9	170	1.06	15	297
	Avg	1.95	0.8	17.8	4.112	3.2	74.8	2.328	9	165.6	4.644	9	296
	Std	1.50	1.30	2.39	0.68	1.92	5.17	1.14	1.58	4.56	2.51	4.06	2.24
9	1	6.1	1	16	2.77	4	69	5.52	11	177	1.14	11	302
	2	4.8	0	17	5.42	1	65	5.24	8	175	4.25	8	290
	3	5.84	1	17	4.49	2	64	7.08	13	167	3.62	8	293
	4	3.58	1	16	5.52	1	69	4.99	8	170	6.37	9	297
	5	5.03	0	15	5.9	2	89	7.48	12	171	1.7	10	300
	Avg	5.07	0.6	16.2	4.82	2	71.2	6.062	10.4	172	3.416	9.2	296.4
	Std	0.99	0.55	0.84	1.26	1.22	10.21	1.14	2.30	4.00	2.10	1.30	4.93
10	1	3.29	0	17	5.55	2	62	0.1	34	179	7.42	6	311
	2	4.02	0	17	5.98	3	96	0.18	10	176	6.81	9	303
	3	3.62	1	16	3.19	5	85	1.38	8	175	7.64	7	312
	4	3.31	0	15	1.27	2	81	0.23	9	177	5.74	7	291
	5	2.91	1	16	2.34	2	80	0.17	12	171	7.5	8	301
	Avg	3.43	0.4	16.2	3.666	2.8	80.8	0.412	14.6	175.6	7.022	7.4	303.6
	Std	0.41	0.55	0.84	2.04	1.30	12.28	0.54	10.95	2.97	0.78	1.14	8.53

Avg	3.05	0.52	17.32	3.26	3.88	77.62	3.65	12.40	172.54	5.45	8.67	289.97
Std	1.06	0.58	1.68	1.22	1.47	7.10	1.13	5.72	4.52	2.06	3.06	4.46

* P: Problem, R: Run, D: Deviation from optimal, C: Number of Comparison, T: Solution Time (in seconds)

***Table 23** HVR and IGD Results of NSGA-II with constraint tournament method*

		100			250			350		
Problem	**Runs**	**HVR**	**IGD**	**Time***	**HVR**	**IGD**	**Time***	**HVR**	**IGD**	**Time***
1	1	0.9859	0.0001	37	0.9839	0.0007	176	0.98091	0.00136	186
	2	0.9861	0.0001	36	0.9840	0.0007	165	0.98074	0.00137	193
	3	0.9861	0.0001	35	0.9838	0.0007	156	0.98096	0.00132	177
	4	0.9861	0.0001	39	0.9837	0.0008	191	0.98068	0.00132	190
	5	0.9858	0.0001	37	0.9837	0.0006	187	0.98099	0.00139	191
2	1	0.9885	0.0002	40	0.9866	0.0007	169	0.98203	0.00101	196
	2	0.9885	0.0002	41	0.9865	0.0007	187	0.98241	0.00100	192
	3	0.9887	0.0003	39	0.9868	0.0005	178	0.98166	0.00133	189
	4	0.9887	0.0001	36	0.9870	0.0005	195	0.98146	0.00124	193
	5	0.9885	0.0002	34	0.9870	0.0006	147	0.98197	0.00109	192
3	1	0.9881	0.0003	38	0.9797	0.0022	161	0.98480	0.00076	199
	2	0.9880	0.0003	37	0.9800	0.0019	147	0.98441	0.00068	189
	3	0.9881	0.0003	36	0.9796	0.0020	198	0.98490	0.00059	201
	4	0.9882	0.0001	39	0.9798	0.0021	176	0.98415	0.00078	200
	5	0.9858	0.0004	40	0.9798	0.0019	159	0.98455	0.00087	202
4	1	0.9826	0.0015	33	0.9834	0.0008	158	0.98376	0.00075	197
	2	0.9827	0.0015	37	0.9830	0.0009	177	0.98297	0.00083	196
	3	0.9836	0.0011	35	0.9834	0.0008	174	0.98323	0.00087	200
	4	0.9828	0.0012	34	0.9832	0.0010	169	0.98367	0.00069	201
	5	0.9831	0.0015	39	0.9832	0.0010	143	0.98393	0.00064	203
5	1	0.9780	0.0052	42	0.9834	0.0002	142	0.98393	0.00039	194
	2	0.9776	0.0052	37	0.9830	0.0001	187	0.94997	0.00045	199
	3	0.9781	0.0051	36	0.9834	0.0002	147	0.95008	0.00032	197
	4	0.9773	0.0054	39	0.9832	0.0001	155	0.94927	0.00065	200
	5	0.9780	0.0051	37	0.9832	0.0001	169	0.94974	0.00057	200
6	1	0.9854	0.0002	33	0.9854	0.0023	175	0.98283	0.00094	200
	2	0.9848	0.0003	34	0.9856	0.0026	157	0.98231	0.00093	209
	3	0.9846	0.0004	39	0.9850	0.0026	166	0.98285	0.00096	199
	4	0.9850	0.0003	38	0.9855	0.0024	186	0.98348	0.00074	198
	5	0.9851	0.0004	37	0.9853	0.0027	173	0.98322	0.00087	199
7	1	0.9894	0.0002	39	0.9865	0.0012	188	0.98578	0.00049	199
	2	0.9898	0.0003	41	0.9868	0.0014	167	0.98570	0.00048	200
	3	0.9888	0.0005	40	0.9864	0.0018	162	0.98634	0.00035	201
	4	0.9897	0.0002	40	0.9868	0.0019	159	0.98537	0.00054	200
	5	0.9889	0.0003	37	0.9869	0.0014	149	0.98518	0.00051	197
8	1	0.9695	0.0045	40	0.9836	0.0026	145	0.98452	0.00037	211
	2	0.9770	0.0045	39	0.9837	0.0021	169	0.98499	0.00038	208
	3	0.9779	0.0044	39	0.9833	0.0024	189	0.98502	0.00036	200
	4	0.9776	0.0046	34	0.9837	0.0024	181	0.98488	0.00040	188
	5	0.9769	0.0044	41	0.9838	0.0019	177	0.98469	0.00038	193
9	1	0.9764	0.0033	40	0.9836	0.0023	153	0.98245	0.00078	203
	2	0.9768	0.0034	40	0.9838	0.0020	150	0.98295	0.00089	205
	3	0.9767	0.0034	39	0.9841	0.0025	151	0.98351	0.00060	210
	4	0.9762	0.0035	39	0.9840	0.0028	160	0.98346	0.00071	191
	5	0.9763	0.0035	40	0.9842	0.0034	169	0.98337	0.00073	201
10	1	0.9868	0.0001	39	0.9831	0.0014	178	0.98453	0.00933	209
	2	0.9855	0.0006	41	0.9827	0.0015	182	0.98542	0.00930	201
	3	0.9869	0.0001	39	0.9832	0.0014	180	0.98374	0.00956	193
	4	0.9867	0.0003	39	0.9827	0.0012	177	0.98382	0.00960	194
	5	0.9868	0.0002	37	0.9829	0.0014	179	0.98409	0.00963	196
	Avg	0.9838	0.0016	37.90	0.9840	0.0014	169.02	0.9808	0.0017	197.53
	Std	0.0050	0.0019	2.28	0.0020	0.0009	14.9	0.0095	0.0027	6.5259

* in seconds.

***Table 24** Results of the NSGA II with constraint tournament method*

		Problem Size		
		100	**250**	**350**
Problem	**Utility U(x)**	**Best Deviation From Optimal**	**Best Deviation From Optimal**	**Best Deviation From Optimal**
1	Linear	0.02	0.97	0.60
	Chebyshev	0.16	1.14	7.81
2	Linear	0.14	0.61	7.28
	Chebyshev	0.26	1.21	7.94
3	Linear	0.08	5.34	0.49
	Chebyshev	0.25	7.85	7.69
4	Linear	0.49	1.15	0.37
	Chebyshev	0.59	1.59	7.76
5	Linear	0.86	0.21	6.60
	Chebyshev	0.32	0.31	7.90
6	Linear	0.41	7.73	0.59
	Chebyshev	0.46	8.08	7.53
7	Linear	0.33	5.14	4.92
	Chebyshev	0.45	8.08	8.25
8	Linear	2.67	0.65	5.68
	Chebyshev	4.02	1.36	7.61
9	Linear	0.86	0.62	0.33
	Chebyshev	0.22	7.67	5.74
10	Linear	0.49	1.65	5.38
	Chebyshev	1.11	2.02	7.70

Linear	Average	0.64	2.41	3.23
Chebyshev	Average	0.78	3.93	7.59