AN INTERACTIVE METHOD FOR INVERSE MULTIPLE CRITERIA SORTING PROBLEM

BURCU ÖZMEN

JANUARY 2017

İZMİR UNIVERSITY OF ECONOMICS
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN INTERACTIVE METHOD FOR INVERSE MULTIPLE CRITERIA SORTING PROBLEM

BURCU ÖZMEN

JANUARY 2017

`Approval of the Graduate School of Natural and Applied Sciences

(Assoc. Prof. Dr. Devrim ÜNAY)

Acting Director

I certify that this thesis satisfies all the requirements for the degree of Master of Science in Industrial Engineering.

(Assoc. Prof. Dr. Selin ÖZPEYNİRCİ)

Head of Departmant

We have read this thesis entitled An Interactive Method for Inverse Multiple Criteria Sorting Problem prepared by Burcu ÖZMEN under supervision of Assoc. Prof. Dr. Selin ÖZPEYNİRCİ and we hereby agree that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science in Industrial Engineering.

(Assoc. Prof. Dr. Selin ÖZPEYNİRCİ)

Advisor

Examining Committee Members:
(Chairman, Supervisor, Members)

Assoc. Prof. Dr. Selin ÖZPEYNİRCİ
(Industrial Engineering Dept., IUE)

Asst. Prof. Dr. Özgür ÖZPEYNİRCİ
(Logistic Management Dept., IUE)

Assoc. Prof. Dr. Gözde YAZGI TÜTÜNCÜ
(Mathematics Management Dept., IUE)

Asst. Prof. Dr. Metehan Feridun SORKUN
(Business Administration Dept., IUE)

Assoc. Prof. Dr. Serhat BURMAOĞLU
(Business Administration Dept., IKC)

ABSTRACT

AN INTERACTIVE METHOD FOR INVERSE MULTIPLE CRITERIA SORTING PROBLEM

ÖZMEN, BURCU

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

Supervisor: Assoc. Prof. Dr. Selin ÖZPEYNİRCİ

January 2017

This thesis work addresses the interactive method for inverse multiple criteria sorting problem. Multiple criteria sorting methods aim to assign the alternatives to ordered classes. The interactive method involves a person who called as the decision maker. The inverse case intends to re-assign the alternatives to the higher classes by taking necessary set of actions, if possible.

Basically, the alternatives were already assigned in the inverse sorting problems. The actions with their contributions to the alternatives are also defined. By choosing proper set of actions, the alternatives are tried to re-assign better classes with respect to the problem limitations.

The developed model relies on the Majority Rule Sorting (MR-Sort) Method which is a simplified version of Electre Tri. In this study, two cases are examined: the case of two classes and the case of three classes. The experiments are done with the developed algorithms and their results are analysed.

Keywords: MR-Sort, Multiple Criteria, MCDA, Interactive Algorithm, Inverse Algorithm

ÖZ

TERS ÇOK AMAÇLI SINIFLANDIRMA PROBLEMLERİ İÇİN ETKİLEŞİMLİ BİR ALGORİTMA

ÖZMEN, BURCU
FEN BİLİMLERİ ENSTİTÜSÜ
Danışman: Doç. Dr. Selin ÖZPEYNİRCİ
Ocak 2017

Bu tez ters çok amaçlı sınıflandırma problemleri için etkileşimli bir algoritmaya değinmektedir. Çok amaçlı sınıflandırma problemlerinin amacı, nesneleri önceden tanımlanmış olan sıralı sınıflara atamaktır. Etkileşimli algoritmalar ise literatürde karar verici olarak tanımlanan bireyin ilgili probleme dahil edilmesiyle oluşmaktadır. Ters çok amaçlı sınıflandırma problemi ise belli eylemler altında nesnelerin, mümkünse, halihazırda bulundukları sınıflardan daha iyi sınıflara atnamasıdır.

Temel olarak ters çok amaçlı sınıflandırma problemlerinde nesnelerin atandığı sınıflar halihazırda bellidir. Eylemler ve nesnelere olan etkileri de tanımlanmıştır. Uygun eylem seti seçilerek, problem kısıtları dahilinde nesnelerin daha iyi sınıflara ataması hedeflenir.

Bu tez çalışması kapsamında geliştirilen model Electre Tri'nin basitleştirilmiş versiyonu olan MR-Sort (Çoğunluk Kuralı Sıralaması) metoduna dayanmaktadır. Bu çalışmada iki sınıflı ve üç sınıflı problemler olmak üzere iki problem incelenmiştir. İlgili deneyler geliştirilen algoritmaya göre tamamlanmış ve sonuçlar analiz edilmiştir.

Anahtar Kelimeler: MR-Sort, Çok Amaçlı, Çok Kriterli Karar Analizi, Etkileşimli Algoritma, Ters Sınıflandırma

Dedicated to my family and my love...

ACKNOWLEDGEMENT

I would like to give sincere thanks to my thesis advisor, Assoc. Prof. Dr. Selin Özpeynirci for her guidance and mental support. I would also like to acknowledge Asst. Prof. Dr. Özgür Özpeynirci for his great directions, expertise and very valuable comments on this thesis. It is my honour to work with them under their supervision. Their never-ending patience and sense of teaching are valuable virtues which are value added to my life.

I would like to thank my thesis committee members, Assoc. Prof. Dr. Gözde Yazgı Tütüncü, Asst. Prof. Dr. Metehan Feridun Sorkun and Assoc. Prof. Dr. Serhat Burmaoğlu for all of their guidance through this process; your discussion, ideas, and feedback have been absolutely invaluable.

Finally, I must express my very profound gratitude to my parents and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

List of Figures

List of Tables

1 INTRODUCTION

The focus of this thesis is developing an the interactive method for inverse multiple criteria sorting problem. Algorithms for inverse sorting are developed under the case that the assignments of alternatives to classes are known.

Multiple criteria decision analysis (MCDA) is a sub-field of operations research. MCDA aims providing aid to the decision maker for problems that involve multiple and conflicting criteria. In the daily life, one can see many examples of MCDA. Let us consider the example of purchasing a car. You would like to buy a car and you have many preferences. The car that you will purchase would be safe, has low fuel consumption, be a coupe style and your budget is as much as 30.000 euros. With or without brand constraint, there are several options in front of you. Here's a MCDA problem that we face and solve in our daily activities.

The earliest reference of MCDA is dated on Benjamin Franklin. He explained his way of decision to a friend of his, as below MacCrimmon (1973).

" To Joseph Priestley

London, September 19, 1772

Dear Sir,

In the Affair of so much Importance to you, wherein you ask my Advice, I cannot for want of sufficient Premises, advise you what to determine, but if you please I will tell you how.

When these difficult Cases occur, they are difficult chiefly because while we have them under Consideration all the Reasons pro and con are not present to the Mind at the same time; but sometimes one Set present themselves, and at other times another, the first being out of Sight. Hence the various Purposes or Inclinations that alternately prevail, and the Uncertainty that perplexes us.

To get over this, my way is, to divide half a sheet of paper by a line into two columns, writing over the one pro, and over the other con. Then during three or four days consideration I put down under the different heads short hints of the different motives that at different times occur to me for or against the measure. When I have thus got them all together in one view, I endeavour to estimate their respective weights; and where I find two, one on each side, that seem equal, I strike them both out: If I find a reason pro equal to some two reasons con, I strike out the three. If I judge some two reasons con equal to some three reasons pro, I strike out the five; and thus proceeding I find at length where the balance lies; and if after a day or two of farther consideration nothing new that is of importance occurs on either side, I come to a determination accordingly.

And tho' the weight of reasons cannot be taken with the precision of algebraic quantities, yet when each is thus considered separately and comparatively, and the whole lies before me, I think I can judge better, and am less likely to take a rash Step; and in fact I have found great advantage from this kind of equation, in what may be called moral or prudential Algebra.

Wishing sincerely that you may determine for the best, I am ever, my dear Friend,

Yours most affectionately

B. Franklin "

According to Roy Roy1985, there are 3 groups of MCDA problems as choice, ranking, classification/sorting. The choice problems aim to select a best alternative or a small group of alternatives by reducing the options. At the ranking problems, the alternatives are ordered from best to worst one. An example can be given as ranking the universities. Lastly, the classification/sorting problems sort the objects into predefined groups. These groups are also called as categories. The difference between classification and sorting problem is at the sorting problem the groups are identified in an ordinal way from best preferred and least preferred, however there is not such an obligation at the classification problem.

This thesis focuses on the inverse multiple criteria sorting problem and proposes interactive solution approaches. In interactive methods, a decision maker is invited for simple pairwise comparison. In an interactive method, a decision maker provides her preferences in an iterative way and the method finally suggests a solution based on the collected preference information. Based on the interactive method, the DM may express her preferences in several ways including pairwise comparison of alternative solutions, providing reference points, setting upper or lower bounds. The inverse sorting means that we intend the classes of alternatives by changing their scores with necessary actions.

We assume that the inverse sorting problem includes objects (or alternatives), groups (categories), criteria, scores and actions. The set of alternatives and criteria define the given problem. The universities can be considered as alternatives and teaching quality, number of institutes etc. can be considered as criteria. In the sorting/classification problems, the alternatives are assigned to a class (or categories). These classes should be identified according to their order of preference. Each class has its own upper and lower bounds. If an alternative fits these limits, this alternative is assigned corresponding class.

There are numerous methods for solving multiple criteria sorting problem such as UTADIS Zopounidis and Doumpos (1999), ELECTRE Bernard and Bouyssou (1993), MR-Sort (Majority Rule Sorting) Leroy, Mousseau, and Pirlot (2011) and AHP Sort Saaty (1990). This thesis follows the MR-Sort method. This method is a simplified version of ELECTRE TRI.

ELECTRE stands for “Elimination and Choice Expressing Reality" (original in French: “Elimination Et Choix Traduisant la Realite"). At mid 1900’s, SEMA (Systems Engineering Management and Assessment), a European consultancy company focused on multiple-criteria problems for activities in firms. With Bernard Roy’s consultancy, ELECTRE was born in 1965. Based on the first work, more outranking methods were developed such as ELECTRE I, ELECTRE Iv, ELECTRE IS, ELECTRE TRI, ELECTRE IV, and ELECTRE A.

Consider a set of hotels whose classes are already known. The intent of this problem is reassigning hotels to upper classes, if it’s possible. While doing this, corresponding action set is taken into account. For example, an hotel could be assigned to an upper class if its number of swimming-pools is increased from 1 to 3 or green area size is doubled.

An example of the case of three classes could be the supplier evaluation system. Nowadays, almost all companies create systems for evaluating their suppliers. Such a system may consider several criteria simultaneously. Consider an example where the supplier performance value is the sum of its quality and delivery time performance values, each range between 0 to 50. If a supplier performance value is smaller than 40, then this supplier can be assigned a class of unworkable, if its performance is between 40 and 70, then the customer will ask from its supplier to take some actions to increase the performance. Lastly, if the performance is greater than 70, the customer probably thanks the supplier for the outstanding efforts.

We consider a set of predefined actions that affect the performances of the objects, hotels, suppliers etc. These actions are changed according to each objects, each criteria and each class. Let us consider the above supplier evaluation example. The supplier at the middle class would like to increase his level. He decides that he can only be upgraded by using a better quality raw material to reach quality needs of his customer, by hiring new employees to deliver the order within time, or producing more than order quantity to make stock so that, when the customer places an order he could deliver some amount immediately. As you can imagine, all these actions has different value-added to the company. This example stands for just making an increment of one class, from mid to best class. There would be more than these considering to upgrade a supplier from worst to mid class, from worst to best class or from mid to best class. For the nature of our problem, the supplier indicates the object, the criteria indicate quality and delivery time performances, and the classes are assigned according to the suppliers' notes.

Since each action has an associated value indicating the cost of taking this action, the algorithm considers a subset of actions with the minimum cost to upgrade the object from its original class. There are some actions who do not have any effect on the class level. Since we are trying to find a subset of action with the minimum cost value, this type of actions are forbidden to take. Likewise, it is assumed that the starting point is the worst case. So, an object cannot be assigned to a worse class than it currently belongs to. This initial point is also referred as do-nothing case since if there is not any action taken, we are still at the starting point.

As a sub-field of operations research, the sorting problems in MCDA studies are discussed while contributing a decision maker or not, and there is always considered the one directional sorting problems. In the literature, there is only one study about the inverse multiple criteria sorting problem Mousseau, Ö. Özpeynirci, and S. Özpeynirci (2017). This thesis states a new problem while addressing the interactive method of inverse multiple criteria sorting problem. This study combines the interactive approach with inverse sorting problem method. The analysed problem in this thesis can be useful to support decision makers who would like to re-assign the existing alternatives.

The another contribution of this thesis occurs by developing two new algorithms for each case of classes. The literature has general approaches which is independent from the number of classes of the problem. In this study, we separate the cases according to the number of classes and for each case, a different algorithm is constituted.

The organization of this thesis is as follows: Chapter 1 gives the introduction for the subject of this thesis. Chapter 2 is a literature review of Multiple Criteria Sorting Problem and, its interactive and inverse versions. Chapter 3 describes the notations and the mathematical models of developed algorithm. The solution approaches are detailed in Chapter 4 while explaining the flow charts of the algorithms. Chapter 5 focuses on the experimental studies and their detailed results. Chapter 6 concludes the thesis and present future research directions.

2 LITERATURE REVIEW

There are numerous works at the world of multiple criteria decision analysis. Bernard Roy is the one who first proposed the ELECTRE method at SEMA consultancy company. He mentioned this method in details at his book, Roy (1985). This book was originally written in French language, at the year 1996 the English version Roy (1996) was published. More detailed literature review can be found at Govindan and Jepsen (2015).

Binary outranking comparison is used for preference modelling at ELECTRE. Below list shows the possible relationships between two alternatives a and b where S stands for “as good as".

- aSb and not bSa, i.e., aP b (a is strictly preferred to b)
- bSa and not aSb, i.e., bP a (b is strictly preferred to a)
- aSb and bSa, i.e., aIb (a is indifferent to b)
- Not aSb and not bSa, i.e., aRb (a is incomparable to b)

The underlying sorting method of this thesis is MR-Sort whose ancestor is ELECTRE TRI method. The learning parameters of MR-Sort was published as the Algorithmic Decision Theory Conference paper, Leroy, Mousseau, and Pirlot (2011). The simplified procedure of ELECTRE TRI is detailed at Leroy, Mousseau, and Pirlot (2011) by considering a learning procedure. Their experiments were based on two, three categories; three, four, five criteria; and from 10 to 100 alternatives. In this thesis, the experiments are designed under the same categories with Leroy, Mousseau, and Pirlot (2011), three, five, seven criteria; and 50, 100, 200 alternatives.

The determinant factor of this thesis is involving a decision maker for the decision-making process and developing an interactive approach for the inverse sorting problem. Zionts and Wallenius (1976) present an interactive algorithm that the overall utility function is unknown and assumed to be a linear function. The question type asked to the decision maker is simple “yes or no" questions. The similarity between Zionts and Wallenius (1976) and this paper is, the method described by Zionts and Wallenius (1976) start with an arbitrary set of weights. Also the method of this paper follows the starting point references by Koksalan, Karwan, and Zionts (1984) which is the mid-point. In that way, the search region would be smaller and the solution time would be decreased either. With respect to the initial weights, an efficient solution is found. After that, it is asked to the decision maker to choose among a subset of these efficient variables. By following his/her response, there is added new constraint to the model which tries to construct the unknown weights. This very part is followed also in this thesis. The major difference between two studies is instead of asking to compare all subsets by the decision maker following Zionts and Wallenius (1976), this thesis finds efficient adjacencies to the incumbent value, chooses the appropriate one, and asks from the decision maker to prefer among these two solutions.

The effort of this thesis is based on the study of Mousseau, Ö. Özpeynirci, and S. Özpeynirci (2017). This article states their workings on inverse algorithm of multiple criteria sorting problems.

Their algorithm can be used with all methods, such as linear, UTADIS and Mr-Sort. Mousseau, Ö. Özpeynirci, and S. Özpeynirci (2017) analysis the problem within two versions; the simple and the robust version. The simple version assumes the underlying sorting method and all parameters of it are known. The robust version assumes that the underlying sorting method is known, however parameters are not. Instead, a number of reference objects and their assignments to classes are known.

Each version described in the paper Mousseau, Ö. Özpeynirci, and S. Özpeynirci (2017) has two cases: cost minimization and budget limitation. The final goal would be providing a solution with minimum cost or, providing a solution with maximum effort under the monetary tie. Each case is studied and explained for the method of linear, UTADIS and MR-Sort in simple version.

The subject of this thesis is the open research area of Mousseau, Ö. Özpeynirci, and S. Özpeynirci (2017).The authors mentioned that the interactive approach is worth to study for proceeding after their studies.

3 PROBLEM DESCRIPTION

In this chapter, we present the problem description, notations and mathematical models. As mentioned in the previous chapter, there are several methods developed to solve multiple criteria sorting problem. In this thesis, MR-Sort (Majority Rule Sorting) is considered as the underlying sorting model which is basically a simplified version of ELECTRE TRI.

In general, MR-Sort assigns alternatives according to their performances. The logic behind is assigning an alternative to a class if its performance is better than or equal to corresponding class' limit (referred also upper bound).

Triantaphyllou et al. (1998) defines that the alternatives represent the different choices of action available to the decision maker. Usually, the set of alternatives is assumed to be finite, ranging from several to hundreds.

The performance is referred as the outcome where each alternative is assessed using each criterion.

In ELECTRE Tri which is a basis of our underlying method MR-Sort, the categories (or classes) are ordered from worst to the best. Let C denote the set of categories.

$$C = \{C_1, C_2, ..., C_t\}$$

An alternative a can only be assigned to a category (class) C_h if its scores satisfies the limits of corresponding class. These limits are formed of upper and lower bounds. The upper bound of C_h be b_h and it is also the lower bound of the successor class C_{h+1}. This logic is valid for all class indexes from 1 to t while C_t refers to the best class and in parallel C_1 refers to the worst preferred class. The assignment rule processes determining credibility index of an alternative and controlling with the cutting level, λ. If the credibility index exceed the cutting level, this alternative is assigned to the corresponding class.

3.1 NOTATIONS

In this chapter, the notation that will be used in developed algorithms will be given.

Consider a set of objects we defined as $O = \{o_1, o_2, o_3, ..., o_n\}$. The score of each object o_i on each criterion is represented as $o_i = \{o_{i1}, o_{i2}, o_{i3}, ..., o_{in}\}$. The criteria set is represented as $N = \{q_1, q_2, ..., q_n\}$. The ordered class set is shown as $C = \{C_1, C_2, C_3, ..., C_t\}$, where C_t is the most preferred class.

Each class has its own upper bound value which is assumed to be known. The upper bound of class C_h for criterion q_j is b_j^h. It is expected that a class index h has a higher upper bound value than its lower index.

Corresponding mathematical notation is as follow:

$$b_j^h \geq b_j^{h-1} \text{ ; for } h = 2, .., t \text{ and for each } q_j$$

In this thesis, the defined problem is considered within two cases based on the number of classes. We can rewrite the generic notation for each case as below.

For the case of 2 classes:

$b_j^2 > b_j^1$; for each q_j

For the case of 3 classes:

$b_j^3 > b_j^2$; for each q_j

$b_j^2 > b_j^1$; for each q_j

An object o_i can only be assigned to the class C_h if and only if

$$\sum_{j \in N: o_{ij} \geq b_j^{h-1}} w_j \geq \lambda \quad and \quad \sum_{j \in N: o_{ij} \geq b_j^h} w_j < \lambda \tag{1}$$

where the parameter λ in $\{0.0, 1.0\}$ is the cut level (or threshold) and w_j is corresponding weight associated each criterion q_j, $q_j \in N$. Without loss of generality, we can assume: $\sum\limits_{j=1}^{n} w_j = 1$

Classic MR-Sort algorithm performs class assignments if performance of an object is at least as good as the upper bound of proper class under the criteria weight. In this thesis, it is assumed that this logic was already performed to each object, and assigned objects with their classes are already known. At this point, to re-assign the objects, an action set $A = \{a_1, a_2, ..., a_m\}$ should be taken. After taking a subset of actions, new scores of object o_i are calculated.

$o_i' = \{o_{i1}', o_{i2}', o_{i3}', ..., o_{in}'\}$

The upper bound and the threshold control is done according to new score value, and the algorithm makes assignments as defined in equation (1).

3.2 MATHEMATICAL MODEL

We consider two different cases. Inverse multiple criteria sorting problem with underlying sorting method of MR-Sort is studied with 2 and 3 classes. General parameters and variables of MR-Sort algorithm are mentioned below.

Parameters:

λ: threshold

W_j: weight of criterion q_j; for $j = 1, .., n$

b_j^h: upper bound of criterion q_j of class C_h; for $j = 1, .., n$ and $h = 1, .., t$

o_{ij}: score of object o_i, criterion q_j; for $i = 1, .., q$ and $j = 1, .., n$

δ_{ijk}:impact of object o_i , criterion q_j and action k; for $i = 1, .., q$, $j = 1, .., n$ and $k = 1, .., m$

c_k: cost of action k; for $k = 1, .., m$

To re-assign objects to upper classes, new score calculation is needed if one or more actions are taken. Each action has its own value-added. This is referred as δ value as defined above. The δ values alter according to object, criterion and action indexes. Let say there are 2 hotels, 1 criterion and 1 action. The action is adding 10 more rooms to both hotel. At first, it may seem adding equal number of rooms would impact equally. Adding new rooms requires to construct a small side building. However, hotel no.1 has lower area size, and new building would decrease the green area size. So, adding new rooms to hotel no.1 has lower effect than the second hotel.

The second list which is shown below contains corresponding decision variables. A binary variable is necessary to control adding impact value to new score calculation if an action is taken; x_{ik}.

Decision Variables:

o'_{ij}: new score of object o_i, criterion q_j; for $i = 1, .., q$ and $j = 1, .., n$

$$x_{ik} = \begin{cases} 1, & \text{if action } a_k \text{ is taken for object } o_i\text{; for } i = 1, .., q \text{ and } k = 1, .., m \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if object } o_i \text{ is assigned to the class } C_h\text{; for } i = 1, .., q \text{ and } h = 1, .., t \\ 0, & \text{otherwise} \end{cases}$$

$$\text{z}_{ik} = \begin{cases} 1, & \text{if object } o_i \text{ is assigned to the upper class under the criterion } q_j \text{ ; for} \\ & i = 1, .., q \text{ and } j = 1, .., n \\ 0, & \text{otherwise} \end{cases}$$

The general model for MR-Sort algorithm is as below:

(M1):

$$Minimize \quad \sum_{i} \sum_{k=1}^{m} c_k x_{ik} \tag{2}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{3}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{4}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda, \quad \forall i \tag{5}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{6}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{7}$$

This model aims to minimize the total cost (2). In constraint (3), new scores are calculated after corresponding actions are taken. Constraint (4) ensures that the value of z_{ij} is 1 if new score exceeds the lower bound of corresponding class. If weighted sum of z_{ij} is higher than δ value, constraint (5) returns z_{ij} as 1. Binary variables are declared at constraints (6) and (7).

The general model, M1, is the main MR-Sort algorithm. By using and/or modifying M1, we designed new models according to the logic of developed algorithm. Each and every models are explained at following chapters. The chapters 3.2.1 and 3.2.2 cover the models and flow chart of the case of 2 classes and 3 classes, respectively.

3.2.1 The Case of 2 Classes

Regardless of the total number of classes, the algorithm finds extreme points first by minimizing the number of objects in the worst class.

Model 1.1:

$$Minimize \quad B_w \tag{8}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{9}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{10}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M y_{ij}, \quad \forall i \tag{11}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{12}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{13}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{14}$$

$$B_w = \sum_{i=1}^{q} \sum_{j=1}^{n} y_{ij} \tag{15}$$

The decision variable of B_w represents the number of objects in the worst class at the best case. This output means also the lower bound of the worst class (8). The constraints are the same with the model M1. Model 1.1 is a version of M1, by just aiming to minimize the number of objects in the worst class. The number of objects in the worst class is the summation of binary variable which is y_{ij}. The assignments are controlled by the equation (11). Since there are only two classes at this case, N_b, the number of objects in the best class can be found by subtracting total number of objects, q, and the output of this model, B_w.

After finding lower and upper bounds of worst and best classes, respectively, the algorithm checks whether B_w, the minimum number of objects in the worst class, equals its current value, N_w or not. If it is, then the model is already at the best case and we stop. Else, by using Model 1.2, optimal cost for B_w are found.

Model 1.2:

$$Minimize \quad \sum_i \sum_k x_{ik} c_k \tag{16}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{17}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{18}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M y_{ij}, \quad \forall i \tag{19}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{20}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{21}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{22}$$

$$I_w \leq B_w \tag{23}$$

$$I_w = \sum_{i=1}^{q} \sum_{j=1}^{n} y_{ij} \tag{24}$$

Model 1.2 aims to satisfy the condition (23) that optimal worst number of object in the worst class, I_w, should be less than or equal to its lower bound, B_w, with minimum necessary cost (16).

The figure 1 shows the small representation of relation between the number of objects in the worst class and the required cost value. As it is seen from the figure, the cost value increases while the number of objects in the worst class decreases and vice versa.

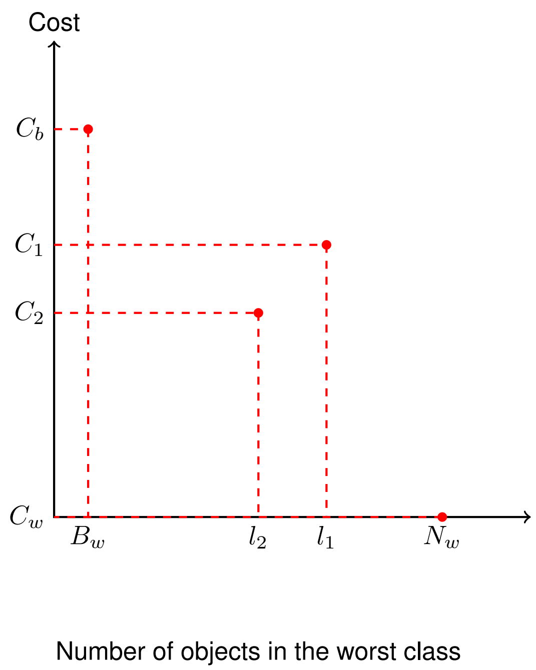

Figure 1: Example of a cost-worst diagram

The algorithm proceeds by finding the middle point between corresponding points of initial and best case of worst class. To do so, we find the ceil value of current number of objects in the worst class and number of objects in the worst class under the best case; N_w and B_w; and l_2 is its adjacency point. Let l_1 be the rounded up value.

$$l_1 = \left\lceil \frac{N_w + B_w}{2} \right\rceil \tag{25}$$

$$l_2 = l_1 - 1 \tag{26}$$

Model 1.2 runs again to find minimum optimal cost to make sure the number of objects in the worst class is less than or equal to l_i.

$$I_w \leq l_1 \tag{27}$$

$$I_w \leq l_2 \tag{28}$$

The minimum total cost necessary to obtain minimum number of objects at worst class, which are l_1 and l_2 are referred to as C_1 and C_2, respectively.

Up to this point, the number of object can be assigned from worst to best class with corresponding cost and same values of adjacency point are calculated. The algorithm continues with score calculation of these two points and their comparison. Score is composed of normalization values of the number of objects in the worst class and cost value.

$$S_1 = \alpha_1 \frac{C_1}{C_b - C_w} + \alpha_2 \frac{l_1}{N_w - B_w} \tag{29}$$

$$S_2 = \alpha_1 \frac{C_2}{C_b - C_w} + \alpha_2 \frac{l_2}{N_w - B_w} \tag{30}$$

The developed algorithm starts with calculating the extreme points, that are correspond to the worst and best cases, by using Model 1.1. The worst case is the one for the initial case, N_w. Since we cannot move worse solution than the initial one, initial case could be considered as the worst point. B_w is the one for the minimum number of objects in the worst class, which is called the best case. If those points are equal to each other, based on the number of objects, the final solution would occur. If not, we need to find optimal cost to reach minimum number of objects in the worst class, B_w and this cost is defined as C_b by Model 1.2. At the score calculation, C_w represents the cost value for N_w which is basically equals to 0 since there is no need to take any additional action to reach initial case. Finding extreme points and cost process is followed by calculating the middle-point of extreme points. This is basic midpoint formula which can be seen in equation (25). The adjacency point of midpoint is closest point based on the number of objects. So, we ask to decision maker that which point s/he chooses. This comparison is done by score calculation at equation (29) and (30). The corresponding flow chart of the developed algorithm developed for the case of 2 classes can be seen at figure 2.

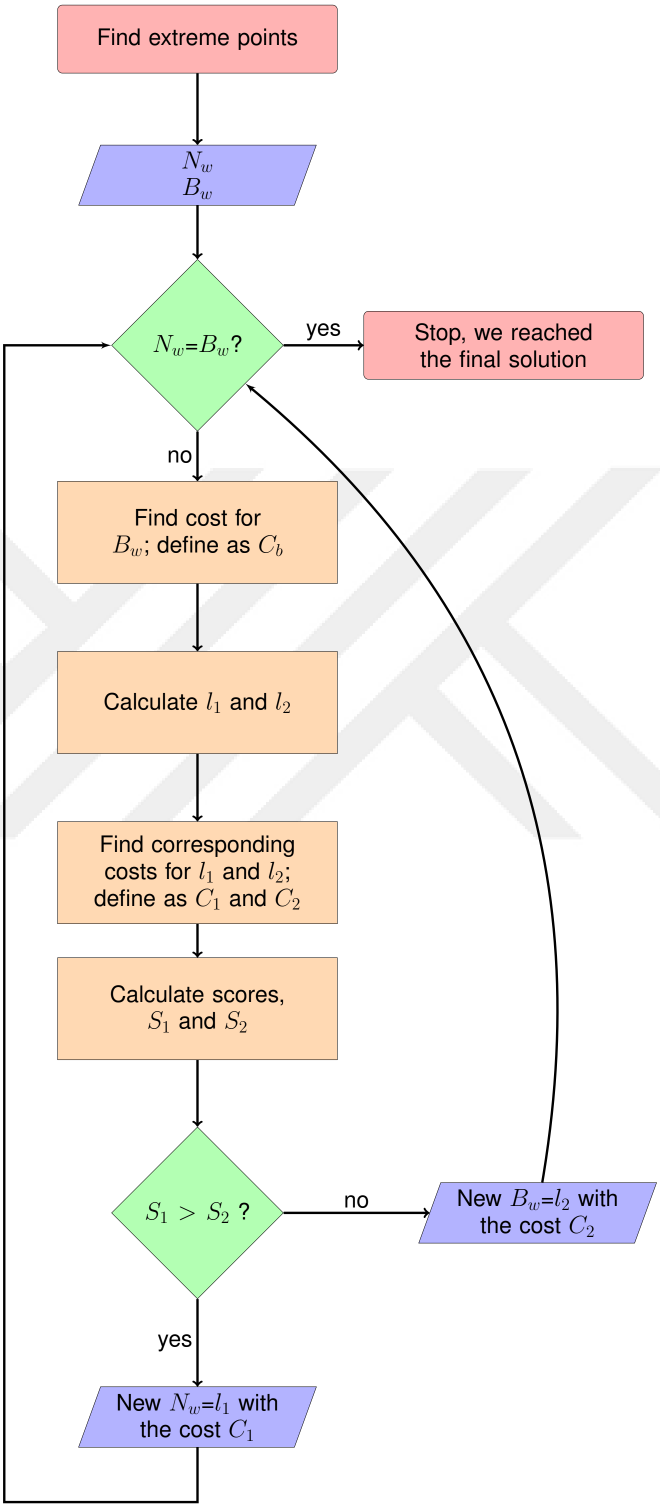

Figure 2: Flowchart of The Case of 2 Classes

3.2.2 The Case of 3 Classes

A similar approach with quite simple alterations is applied to the case for 3 classes. In this context, there are 3 classes which could be defined as worst, middle and best class. As in the case of 2 classes, the algorithm starts with finding the extreme points (31). The constraints are the same with the model M1. Model 1.3 is a version of M1, by just aiming to maximize the number of object in the best class while minimizing the cost and the number of objects in the worst class. The assignments of alternatives are controlled by the equation (34). The summation of the binary variable of y_{ij} such that $j = 1$ gives the number of objects in the worst class (37). The same summation with the index j equals to 2 gives the number of objects in the middle class. Lastly, the total number of objects in the best class is found by summing y_{ij} such that j equals to 3 (36). The equation (35) makes sure all the alternatives are assigned one of the three classes.

The number of objects in the worst class will be referred to as “worst", the number of objects in the best class will be referred to as “best" and corresponding expense needed to achieve corresponding number of objects in the corresponding class will be referred to as “cost" hereafter.

Model 1.3:

$$Maximize \quad B_b - \varepsilon_1 B_w - \varepsilon_2 \sum_i \sum_k x_{ik} c_k \tag{31}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{32}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{33}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{34}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{35}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{36}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{37}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{38}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{39}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{40}$$

The parameters of ε_1 and ε_2 are relatively small numbers. The best case values, B_b and B_w are the objects number of objects in the best and worst classes, respectively.

Since in this case we work on three planes, we cannot follow the same middle point procedure in the case of 2 classes. To find middle point in the case of 3 classes, we perform middle-most algorithm Koksalan, Karwan, and Zionts (1984) which is detailed in the model below.

Model 1.4:

$$Maximize \quad S_i \tag{41}$$

$$\text{s.t.} \quad S_i = \alpha_1 \frac{C_b - C_i}{C_b - C_w} + \alpha_2 \frac{N_w - w_i}{N_w - B_w} + \alpha_3 \frac{b_i - N_b}{B_b - N_b} \tag{42}$$

$$o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{43}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{44}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{45}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{46}$$

$$C_i = \sum_{k} x_{ik} c_k \quad \forall i \tag{47}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{48}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{49}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{50}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{51}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{52}$$

The parameters of α values declare the weight factors for cost, worst class and best class. If α_1, α_2 and α_3 are 1/3, then we would have the middle-most point. S_i is the score of point o_i. The logic is the same with score calculation of the 2 classes. However, normalization of the best class is added.

At the first iteration, the middle-most point is defined as I_0. As it is seen in the flow chart given in Figure 3, after the middle-most point is found out, score comparison with initial values are calculated. In case of having higher cost weight, decision maker may intend to stay on the initial case. If the score of the initial point is higher than the score of middle-most point, decision maker chooses the initial point and the algorithm sets it as new incumbent. Else, the new incumbent value would be the middle-most point. To decrease the bias, i.e. starting from middle-most point even the initial point is closer to the final result, it is necessary to compare these two point of values.

If middle-most point is chosen over the initial point, it is defined as the incumbent point of the first iteration i.e.I_1. Else, the candidate value, initial point at the first iteration, is defined as the incumbent point of the first iteration.

The algorithm is finalised if an incumbent point is chosen 3 times successively. Hence, the algorithm checks every time an incumbent value occurs.

The algorithm follows one of two different procedures at each iteration. In general, the following step after finding an incumbent value is generating new weight set for cost, worst and best classes or, generating adjacencies of the incumbent value. Let us detail this issue as follows.

The very beginning of the algorithm is finding the extreme points which are named as "best case" and "initial case". One of these extreme points is found by maximizing the number of objects in the best class. The other point is the initial point that we already know. The concept of our algorithm is set up basically starting from the middle-most point, finding 3 adjacencies of the middle-most, choosing the closest one with respect to the cost, calculating the scores and choosing the point with highest score. This new point is set as the new incumbent. The process continues with finding adjacencies of new incumbent point and so on. However, while the cost weight factor is much higher than the other two (worst and best class' weights), the initial point could be more realistic since this is a "do-nothing" case, i.e. the cost value is already 0 and our decision maker is not willing to spend more money. To consider this situation, we should ask the decision maker to choose the point s/he chooses. This pairwise comparison can only be possible by calculating the corresponding score of each point. The highest score is chosen and the algorithm follows by finding the new weight factors. The algorithm always begins with the weight of 3 factors (cost, worst and best) equal to each other i.e. 1/3, 1/3 and 1/3 respectively.

The new α values are found by following model.

Model 1.5:

$$Maximize \quad \varepsilon \tag{53}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} I_{ij} \times \alpha_j \geq A_{ij} \times \alpha_j + \varepsilon \quad ;\forall i, j \tag{54}$$

$$\sum_{j=1}^{n} \alpha_j = 1 \tag{55}$$

$$\alpha_j \geq \varepsilon \quad ;\forall j \tag{56}$$

Since the new chosen incumbent value at current iteration has higher score than the candidate point of value, the score of incumbent point should also be higher with new set of weights. By this logic, one can simply equate this interaction as the equation (54). The same score calculation at (42) is still valid however, for this time, set of weights i.e. α_j is missing. The value of I_{ij} includes number of objects in the worst class, number of objects in the best class, and corresponding amount of money should spend to achieve these numbers of objects. Similarly, A_{ij} stands for values of the adjacency point under criteria index j for the object i. To specifically ensure that the score of incumbent value, I_{ij}, is higher than the score of adjacency, A_{ij}, we use a relatively small number which is referred as ε. The objective function is maximizing this difference as much as possible at (53). The equation (55) ensures that the weight factors of each criterion (cost, worst and best) is totaled to 1. Lastly, the new weight factors must be greater than or equal to the epsilon for all criterion, as it is referred at the equation (56).

The next step, after calculating the new weight factors, is finding the new point. The below model maximizes the scores by using new weights so that we can reach the new incumbent point.

The model follows basic MR-Sort mathematical model.

Model 1.6:

$$Maximize \quad S_i \tag{57}$$

s.t. $$S_i = \alpha_1 \frac{C_b - C_i}{C_b - C_w} + \alpha_2 \frac{N_w - w_i}{N_w - B_w} + \alpha_3 \frac{b_i - N_b}{B_b - N_b} \tag{58}$$

$$o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{59}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{60}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{61}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{62}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{63}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{64}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{65}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{66}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{67}$$

From this point on, the algorithm starts to find adjacencies of the incumbent point. These adjacencies are constituted from the number of objects in the best class, the number of objects in the worst class and the cost aspect.

The first adjacency is occurred by running following model.

Model 1.7:

$$Minimize \quad C - \varepsilon \times (B_w - B_b) \tag{68}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{69}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{70}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{71}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{72}$$

$$C \leq \varphi - 1 \tag{73}$$

$$B_w \leq \omega - 1 \tag{74}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{75}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{76}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{77}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{78}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{79}$$

This model aims to find closest adjacency point based on the worst-class factor. The decision variable of C stands for the cost of adjacency point at the equation (68). Other 2 criteria, worst and best, are referred to as B_w and B_b of the corresponding adjacency point.

If the objective function is just minimizing the cost, most probably the model turns out 0. To prevent this situation, a small number of ε is added to the objective function as weight factor of worst and best.

The model intends to find closest number of object in the worst class of incumbent variable which is ω. This constraint is formed at the equation (74) by subtracting 1 from the the number of objects in the worst class of incumbent point. Likewise, not to diverge from the closest point, we also should add the constraint (73) in which the cost-value of adjacency point must be smaller than the incumbent's cost-value i.e. φ.

The equations (69), (70) and (71) are general MR-Sort equations. Binary variables are declared at constraints (77) and (79).

Following model 1.8 is constituted for the second point of adjacency.

Model 1.8:

$$Minimize \quad \varepsilon^- + \varepsilon^+ \tag{80}$$

$$\text{s.t.} \quad o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{81}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{82}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{83}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{84}$$

$$C + \varepsilon^- - \varepsilon^+ \leq \varphi \tag{85}$$

$$\varepsilon^- \leq M \times r \tag{86}$$

$$\varepsilon^+ \leq M \times (1 - r) \tag{87}$$

$$\varepsilon^- + \varepsilon^+ \geq 1 \tag{88}$$

$$B_b \geq \beta + 1, \tag{89}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{90}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{91}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{92}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{93}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{94}$$

$$r \in \{0, 1\} \tag{95}$$

This model aims to find closest adjacency point based on the cost factor. The objective function (80) minimizes the difference between the cost value of the incumbent point,φ and the cost of the adjacency point C by using ε. This difference composes of two sub-factor as ε^- and ε^+. The plus-factor turns 0 while the cost of the adjacency point is smaller than the incumbent point's cost. If the outcome cost value is higher than the φ, then the minus-factor becomes 0. These cases are symbolised at the equations (85), (86) and (87). The summation of the sub-factors should be greater than or equal to 1 (88).

So far, we've seen the calculation of adjacencies according to the number of objects in the worst class and the cost. The first one finds the adjacency by minimizing the cost and the number of objects in the worst class. Logically, as we try to minimize the cost, the number of objects in the best class tends to decrease. So, the first mathematical model gives always a worse point than the incumbent.

The second model tries to find closest cost adjacency, however, we haven't added any worst-class and/or best-class constraint. At the next adjacency model 1.9, we will see that this mathematical model is constructed for minimizing the number of objects in best best class. So, the second adjacency should stand for finding a point which has greater number of objects in the best class with smaller cost value. The corresponding equation for declaring best-class value is (89). This outcome point might be referred as better than the incumbent.

Lastly, following model is finding for the third adjacency whose outcome is closest but worst number of object at the best-class with smaller cost value.

Model 1.9:

$$Minimize \quad \varepsilon \times C + B_w - B_b \tag{96}$$

$$\text{s.t.} o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{97}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{98}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{99}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{100}$$

$$C \leq \varphi - 1 \tag{101}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{102}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{103}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{104}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{105}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{106}$$

At Chapter 4, we focus on the design of experiments with their results. We also need to cross check how effective our results are. The basic control is considering the solution in the case knowing the weights. To achieve this, below Model 1.10 is developed.

Model 1.10:

$$Maximize \quad S_i \tag{107}$$

$$\text{s.t.} \quad S_i = \alpha_1 \frac{C_b - C_i}{C_b - C_w} + \alpha_2 \frac{N_w - w_i}{N_w - B_w} + \alpha_3 \frac{b_i - N_b}{B_b - N_b} \tag{108}$$

$$o'_{ij} = o_{ij} + \sum_{k=1}^{m} \delta_{ijk} x_{ik}, \quad \forall i, j \tag{109}$$

$$o'_{ij} \geq b_j^h - M(1 - z_{ij}), \quad \forall i, j \tag{110}$$

$$\sum_{j=1}^{n} w_j z_{ij} \geq \lambda - M(1 - y_{ij}), \quad \forall i \tag{111}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad \forall i \tag{112}$$

$$B_b = \sum_{i|j=3} y_{ij} \tag{113}$$

$$B_w = \sum_{i|j=1} y_{ij} \tag{114}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i, k \tag{115}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j \tag{116}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \tag{117}$$

The aim is finding the maximum score as it is seen from the objective function equation (107). The weights which are mentioned as α_1,α_2 and α_3, are changed according to the test environment. Since only 3 data sets are developed, the α-values could equal to each other; could be 50% for cost, 40% for worst and 10% for best class; and lastly could be 70% for cost, 20% for worst and 10% for best class. After the optimal results are reached in the case if the decision maker known his/her importance factor on the each criteria, the comparison between solutions of the algorithm and the decision maker. This comparison is a gap factor. The smaller the gap is, the closer solution that we found.

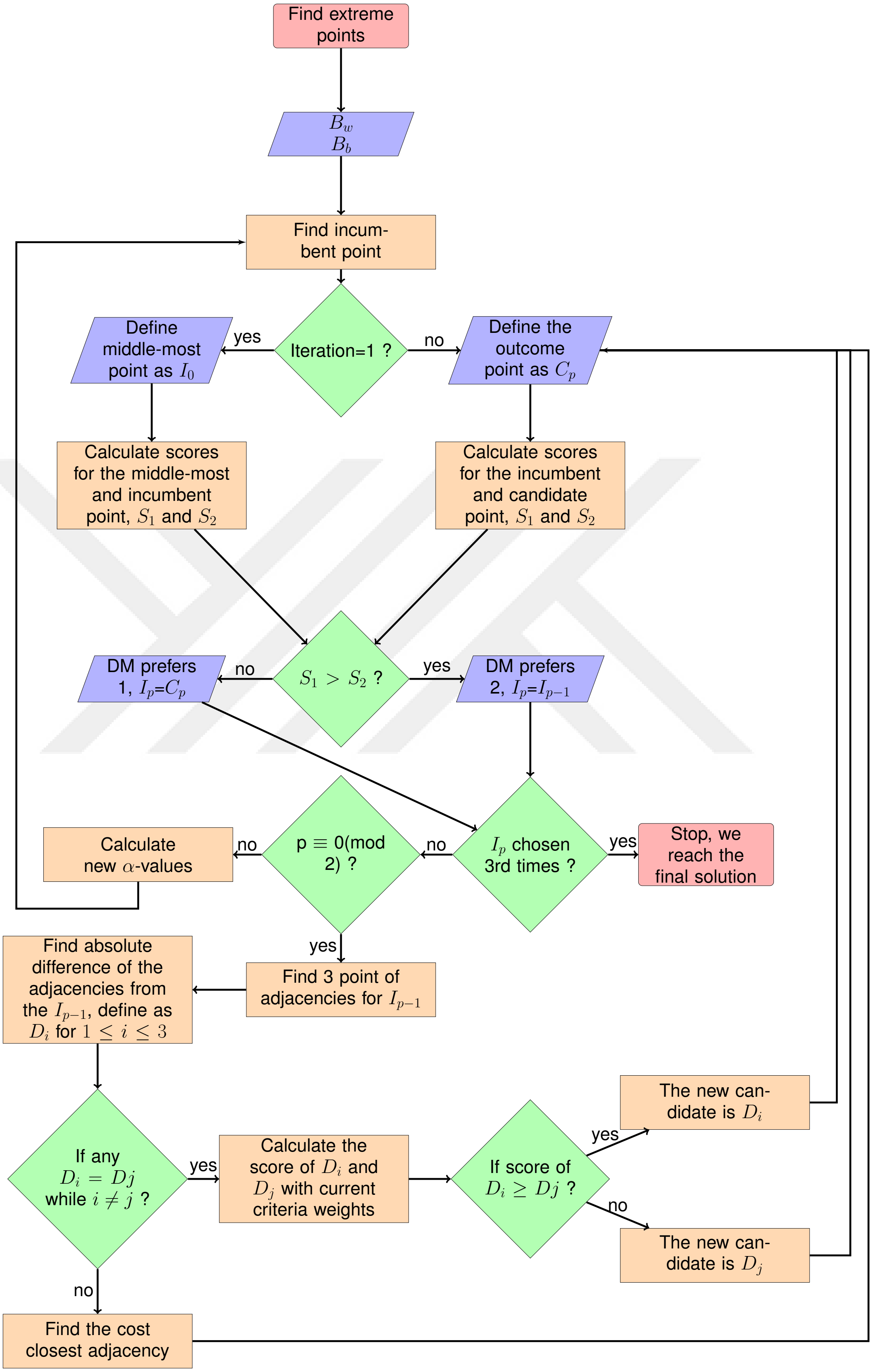

Figure 3: Flowchart of The Case of 3 Classes

4 INTERACTIVE ALGORITHM

In this section, we discuss the proposed algorithms in detail. Since the problem itself has 2 different contents, we will analyse them individually.

4.1 The Case of 2 Classes

The data consists of the pre-defined classification information that, the class of each object is already identified. The aim is assigning the objects while their original class degrees are upgraded. This alteration on between the classes are consider according to the necessary actions. The delta factor of each action for each object and criterion is also mentioned at the initial data.

The algorithms start by finding the extreme points as it is seen Algorithm 1.

Algorithm 1: Mathematical algorithm for the 2-classes case

1 Find extreme points by using Model 1.1 ;
2 Define the worst and best values of the extreme point are B_w and N_w ;
3 **if** $B_w = N_w$ **then**
4   Stop, we reached the final solution ;
5 **else**
6   Find the cost for B_w by Model 1.2 and define as C_b ;
7 **end**
8 Calculate l_1 and l_2 by equations (12) and (13) ;
9 Find the costs for l_1 and l_2 by using Model 1.2 using the constraints equations (14) and (15) ;
10 Ask to the decision maker to choose between l_1 and l_2 by score calculation ;
11 Define the score of l_1 as S_1 and the score of l_2 as S_2 ;
12 **if** $S_1 > S_2$ **then**
13   The decision maker chooses l_2 ;
14   Assign new N_w as follows, $N_w = l_1$;
15 **else**
16   The decision maker chooses l_1. ;
17   Assign new B_w as follows, $B_w = l_2$;
18 **end**
19 Increase the iteration number by 1 ;
20 Go to Step 3;

These extreme points are defined as B_w, the number of objects in the worst class at the best case and; N_w, the number of object in the worst class at the worst case i.e. the initial point. These two points stand for the numeric value of the best case which indicates the case of having minimum assignment to the worst class.

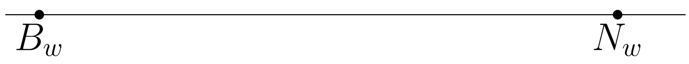

Figure 4: The case of 2 classes representation for the extreme points

At the first iteration, the incumbent point is B_w. The algorithm checks this number of objects in the worst class at the best case (B_w) equals to the initial number of objects in the worst class (N_w) or not. If they are equal to each other, we already found the solution and the model stops. If not, the corresponding cost value to reach B_w number of objects in the worst class and this output is assigned as C_b.

The middle point of N_w (current number of objects in the worst class) and B_w (as the best case) is calculated. This value l_1 is used for pairwise comparison by equation (12). The pairwise comparison is an action by asking to the decision maker to compare two point of values. This event is occurred by score calculation of compared points.

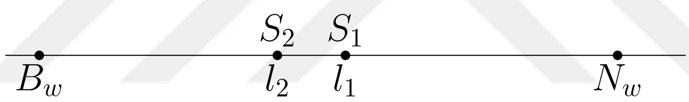

Figure 5: The case of 2 classes representation for the middle point and its adjacency

At the next step, we ask to the decision maker to choose the middle point of current and best cases (l_2) and its adjacency point ($l_2 = l_1 - 1$) by equation (13). Corresponding scores are evaluated (equations (16) and (17)), and the point who has the minimum score represents the decision maker choice. The algorithm posts new incumbent point of the current iteration according to the choice. This calculation is done at Step 11, and Steps 12 from 17 are followed based on the decision. If the score of l_1 is higher than the score of l_2, the decision maker prefers to the point of l_2. The new N_w takes the value of l_1.

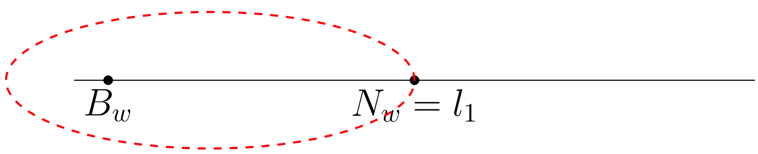

Figure 6: The case of 2 classes representation for the case where $S_1 \geq S_2$

Else, the new B_w takes the value of l_2 and the algorithm proceeds between the points of l_2 and N_w. This side of the line represents that the decision maker prefers spend less.

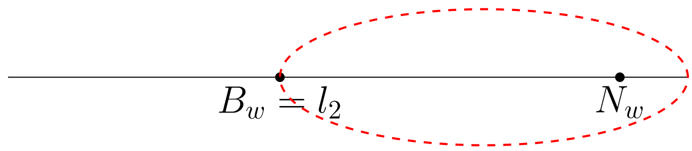

Figure 7: The case of 2 classes representation for the case where $S_1 < S_2$

After the comparison, we proceed to the next iteration (Step 19), and again check the number of objects in worst class at the best case is equal to the incumbent point (l_1 or l_2). The algorithm follows the same procedures until an equivalency is found.

4.2 The Case of 3 Classes

Regardless of the number of classes, the algorithms start by finding the extreme points as it is seen below Algorithm 2. There are two extreme points. One stands for the initial case. Since there would not be any worst scenario then the available one, it is simply considered the initial case with 0 cost-value is the first point (do-nothing case).

The second extreme point is known as best-case, i.e. possible maximum number of objects will be in the best class. The idea is finding necessary cost and actions to be taken for assigning all the objects to the best class. Model 1.3 is formed to find the best case. So, the first step is done.

The second step is actually a definition part. The outputs of Model 1.3 are noted for the number of the objects in the worst class (B_w), the number of the object in the best class (B_b) and the corresponding cost (C_b).

The next step is finding the incumbent point. At the first iteration of each instances, the first incumbent point is calculated as the middle-most point by the Model 1.4. This decision process is occurred at Step 4 and 7. If the current iteration is the first one, then the algorithm follows the middle-most model. If not, the algorithm also uses Model 1.4 with current set of weight (instead of 1/3, 1/3, and 1/3 for each criterion) and find an incumbent point. Since the pairwise comparison is not done yet, this output is noted as a candidate value, C_p which p states the iteration number.

Let us continue with the first iteration ($p = 1$). The incumbent point which is defined as I_0 for the first iteration, is calculated by the middle-most model according to the equal weights. The candidate point which is used for the pairwise comparison, stands for the initial point.

Since this concept does not include any person who acts as a decision maker, corresponding scores are calculated for comparison. At step 10, the scores of the candidate (initial) point and the incumbent (middle-most) point are calculated. At the beginning of the algorithm, the weight set is initially 1/3 for each criterion. So, for the first iteration, the score calculation is done by using equal weights.

After calculating the scores, a pairwise comparison should be done. It is assumed that any point with higher score value is chosen by the decision maker, and the algorithm defines this point as the incumbent point of current iteration, I_1. Steps 11 and 13 lead to which point should be chosen. The point who has the higher score is defined as the incumbent value.

Since we are at the first iteration, we skip the decision at Step 17.

The algorithm follows 2 different paths during the run. The one is finding adjacencies and the other is calculating new set of weights. To find which path that we should follow, the algorithm is designed by checking the current iteration is multiples of 2 or not except for the first iteration. At the first iteration, after the model finds the incumbent value, it follows by the adjacencies.

The adjacencies of the incumbent point I_1 are calculated by using the Model 1.7, 1.8 and 1.9. In parallel, the algorithm increases the current iteration as 2. After the closest adjacency points based on the number of objects in the worst class, number of objects in the best class and the cost value of the incumbent point, each differences from the cost of the incumbent are calculated. These absolute differences are sorted in ascending order to see which adjacency has the closest cost to the incumbent. If there are any two points who have the same difference, then the sub scores need to be calculated. The current α-values are used for these sub scores. Since this weight does not stand for the decision maker's set of weight, the algorithm does not count it as a question. With the same logic, the adjacency with the higher score will become the candidate point, C_2. If all 3 differences are different from each other, the minimum difference will be assigned as the candidate point, C_2.

We also need to check if the candidate point comes from the adjacencies, was already asked to decision maker to compare itself and the incumbent value. Otherwise, we always ask the same questions to the decision maker and stick on the same results. If this adjacency is already used, then we ask the decision maker to choose between the incumbent value and the second closest adjacency. If the second one is used before also, then the last closest adjacency should be used for comparison.

Since we are still at the second iteration, there is not any chance the incumbent point is chosen three consecutive times. So, the algorithm continues with Step 19. Again, it follows finding adjacencies, checking the closest adjacency was used before or not, according to the control result defining the candidate point to the adjacency as it should be, score calculation (meaning asking to the decision maker to do pairwise comparison), and finding the second iteration's incumbent point.

Hereby, the second iteration is done. Now we are at the third iteration and we need to follow by generating new set of weights. New weights are calculated by the model 1.5. The algorithm goes back finding the optimal point for new weight set of criteria. The same procedure is valid as it is done at the middle-most, but now, instead of using equal weights, the algorithm uses newly-introduced weights. By following the else case of Step 4, the outcome point is defined as C_3. Now, the decision maker has to make a selection between the candidate point of the current (third) iteration and the incumbent point of the previous (second) iteration. Again, according to the scores, the incumbent point of value would has the higher score.

This procedure continues in the same for the case of 3 classes until an incumbent point is chosen in three consecutive iterations. At that case, the model stops and brings out current incumbent point as the best decision.

Algorithm 2: Mathematical algorithm for the case of 3 classes

1 Find extreme points by using Model 1.3 ;
2 Define the worst, best and cost values of the extreme point are B_w, B_b and C_b ;
3 Find the incumbent/candidate point by using Model 1.4 ;
4 **if** *iteration=1* **then**
5   Define the initial point as the candidate point, C_1 ;
6   The incumbent point is calculated by middle-most model, I_0 ;
7 **else**
8   Define the candidate point as C_p ;
9 **end**
10 Ask to the decision maker to choose between C_p and I_{p-1} by score calculation ;
11 Define the scores as S_1 and S_2 ;
12 **if** $S_1 > S_2$ **then**
13   The decision maker chooses I_{p-1}. Assign new incumbent point of the iteration p, $I_p = I_{p-1}$;
14 **else**
15   The decision maker chooses C_p. Assign new incumbent point of the iteration p, $I_p = C_p$;
16 **end**
17 **if** I_p *is chosen as the third time* **then**
18   Stop, we reach the final solution which is I_p ;
19 **else**
20   **if** *p* $\equiv$ *0(mod 2) or p = 1* **then**
21   Find adjacencies ; Increase the iteration number by 1 ; Find the adjacency point which has closest cost ;
22   **if** *There are two points who have the same closest cost* **then**
23   Calculate sub-score by using current α-values and define the higher score one as C_p ;
24   **end**
25   **if** C_p *is used before for the comparison* **then**
26   Define the second/third closest adjacency as the candidate ;
27   **end**
28   Define the closest cost adjacency point C_p ; Go to Step 10 ;
29   **else**
30   Calculate new α-values ; Increase the iteration number by 1 ; Go to Step 3 ;
31   **end**
32 **end**

5 COMPUTATIONAL EXPERIMENTS

In this chapter, the data structure is explained, the experiment results are shared and the cross check of the results are analyzed.

The two mentioned algorithms are modelled at IBM ILOG CPLEX Optimization Studio 12.6.3. Corresponding tests are performed by a computer with Intel Core i5-2410M CPU at 2.30GHz.

The values occurred for design of experiment are summarized in Table 1 and Table 2. The sets are varied based on the parameters. One set of test consists of 50 objects, 3 criteria, 2 classes and 10 actions. Each set has 10 different instances in itself. All data sets are complied 3 times under the different set of weights. These sets stand for the decision maker's weights used while pairwise comparison by score calculation. In total, there are 1,620 instances.

Table 1: Values of Design Parameters

Parameters	Levels		
Number of objects	50	100	200
Number of criteria	3	5	7
Number of classes	2	3	
Number of actions	10	30	50
Number of instance	10		

Table 2: Values of Set of Weights

Parameters	Cost	Worst
	50%	50%
The case of 2 classes	40%	60%
	30%	70%

Parameters	Cost	Worst	Best
	1/3	1/3	1/3
The case of 3 classes	50%	40%	10%
	70%	20%	10%

After the data are compiled by the solver, we need to prove the accuracy of our outputs. This consistency can only be checked by the decision maker him/herself. To ensure this situation, new mathematical model is developed. This model uses the same data sets and gives the decision maker solution with optimality. The corresponding model is named after Model 1.10 which is described at Chapter 3.

This chapter is divided into two subsections. The first one includes all the experimental results of the case of 2 classes, and the second subsection includes the results of the case of 3 classes. Each part has summarized tables of the outputs and detail results. These tables includes the number of objects, criterion, class, action; the instance; the number of questions asked to the DM; CPU time; algorithm score; DM score; difference and the gap.

At these tables, it is seen the number of object as “OBJ", the number of criteria as “CRI", the number of actions as “ACT", the proper instance as “INS", the number of questions asked as “QUEST.", the cpu time as “CPU", the score of the developed algorithm as “ALG", the score of the case with known weights as “DM", the difference between these two scores as “DIF" and the percentage value of the difference as “GAP%".

At the score column, there are two different scores: “ALG" and “DM". As it’s predictable by its name, “ALG" stands for the algorithm score which is the result of referred model at this thesis. “DM" stands for optimal case. If we know the weight parameter, we just build a mathematical model 1.10 and find the optimal solution.

The “DIF" is the abbreviation of difference. This column is a simple subtraction from the optimal solution and the algorithm solution. Lastly, “GAP" shows the percentage of this difference. This value means how far we close to the optimal case.

All the results of both coming from the algorithm that we developed and from the decision maker, and the comparison (gap values) are shared.

The evaluation of the results are done by checking the gap values. Closer the gap values to zero, the better results we have.

5.1 The Case of 2 Classes

In this chapter, the results of the case for 2 classes will be represented. The experiments were done under three different weight sets. These weights are composes of two parameters which are cost-value and the number of alternatives at worst class. The dual of 40%-60%, the equal set and 30%-70% are hereafter referred to as “Weight Set 1", “Weight Set 2" and “Weight Set 3", respectively.

The detailed results of the first set weight are shown at the table 3, the results of the second weight set can be seen at 4 and lastly, the results of the third weight set are detailed at the table 5.

For the first set weight, the average gap value is 4.25% which means, we found a solution almost 4.25% close to the optimal solution. During the experiments of the case of 2 classes, 5.55 questions on average are asked to the decision maker and it takes 114.31 seconds on average takes to solve a problem instance.

Table 3: The Case of 2 Classes Experimental Results for Set Weight No.1

OBJ	CRI	CLASS	ACT	INS	QUEST.	CPU	ALG	DM	DIF	GAP %
50	3	2	10	1	3	30	0.400	0.385	0.02	3.85
50	3	2	10	2	5	52	0.400	0.365	0.03	8.75
50	3	2	10	3	5	48	0.400	0.360	0.04	9.95
50	3	2	10	4	3	31	0.400	0.400	0.00	0.00
50	3	2	10	5	4	42	0.400	0.380	0.02	4.93
50	3	2	10	6	5	57	0.400	0.358	0.04	10.55
50	3	2	10	7	5	52	0.400	0.356	0.04	10.99

50	3	2	10	8	5	61	0.400	0.377	0.02	5.82
50	3	2	10	9	5	63	0.400	0.380	0.02	5.00
50	3	2	10	10	5	60	0.400	0.387	0.01	3.24
50	3	2	30	1	5	78	0.400	0.388	0.01	2.95
50	3	2	30	2	5	85	0.400	0.380	0.02	4.99
50	3	2	30	3	5	80	0.400	0.382	0.02	4.57
50	3	2	30	4	4	50	0.400	0.394	0.01	1.46
50	3	2	30	5	5	78	0.400	0.386	0.01	3.45
50	3	2	30	6	5	74	0.400	0.371	0.03	7.31
50	3	2	30	7	3	35	0.400	0.400	0.00	0.00
50	3	2	30	8	4	47	0.400	0.386	0.01	3.53
50	3	2	30	9	4	50	0.400	0.381	0.02	4.81
50	3	2	30	10	5	74	0.400	0.388	0.01	3.09
50	3	2	50	1	5	87	0.400	0.386	0.01	3.60
50	3	2	50	2	5	68	0.400	0.383	0.02	4.34
50	3	2	50	3	5	71	0.400	0.390	0.01	2.52
50	3	2	50	4	5	74	0.400	0.392	0.01	2.08
50	3	2	50	5	4	50	0.400	0.395	0.00	1.20
50	3	2	50	6	3	36	0.400	0.400	0.00	0.00
50	3	2	50	7	2	28	0.400	0.400	0.00	0.00
50	3	2	50	8	5	73	0.400	0.382	0.02	4.48
50	3	2	50	9	2	28	0.400	0.400	0.00	0.00
50	3	2	50	10	3	42	0.400	0.400	0.00	0.00
50	5	2	10	1	4	39	0.400	0.365	0.04	8.79
50	5	2	10	2	5	55	0.400	0.387	0.01	3.27
50	5	2	10	3	5	59	0.400	0.372	0.03	7.09
50	5	2	10	4	4	45	0.400	0.385	0.01	3.69
50	5	2	10	5	5	57	0.400	0.359	0.04	10.13
50	5	2	10	6	5	48	0.400	0.382	0.02	4.59
50	5	2	10	7	5	62	0.400	0.372	0.03	6.99
50	5	2	10	8	3	33	0.400	0.400	0.00	0.00
50	5	2	10	9	4	40	0.400	0.376	0.02	6.07
50	5	2	10	10	5	50	0.400	0.384	0.02	4.04
50	5	2	30	1	5	58	0.400	0.381	0.02	4.76
50	5	2	30	2	4	45	0.400	0.388	0.01	3.05
50	5	2	30	3	5	83	0.400	0.382	0.02	4.58
50	5	2	30	4	5	138	0.400	0.395	0.00	1.19

50	5	2	30	5	5	77	0.400	0.379	0.02	5.23
50	5	2	30	6	5	66	0.400	0.368	0.03	8.02
50	5	2	30	7	3	40	0.400	0.400	0.00	0.00
50	5	2	30	8	5	75	0.400	0.372	0.03	6.95
50	5	2	30	9	5	101	0.400	0.382	0.02	4.51
50	5	2	30	10	5	113	0.400	0.392	0.01	1.95
50	5	2	50	1	4	50	0.400	0.378	0.02	5.57
50	5	2	50	2	4	65	0.400	0.399	0.00	0.18
50	5	2	50	3	5	77	0.400	0.379	0.02	5.18
50	5	2	50	4	5	80	0.400	0.375	0.02	6.17
50	5	2	50	5	4	52	0.400	0.392	0.01	2.00
50	5	2	50	6	5	90	0.400	0.378	0.02	5.49
50	5	2	50	7	5	93	0.400	0.383	0.02	4.35
50	5	2	50	8	5	97	0.400	0.381	0.02	4.66
50	5	2	50	9	5	96	0.400	0.390	0.01	2.60
50	5	2	50	10	5	74	0.400	0.372	0.03	7.01
50	7	2	10	1	4	33	0.400	0.350	0.05	12.39
50	7	2	10	2	5	53	0.400	0.369	0.03	7.67
50	7	2	10	3	5	44	0.400	0.384	0.02	4.05
50	7	2	10	4	5	55	0.400	0.374	0.03	6.38
50	7	2	10	5	5	56	0.400	0.382	0.02	4.42
50	7	2	10	6	4	44	0.400	0.380	0.02	5.11
50	7	2	10	7	4	37	0.400	0.400	0.00	0.00
50	7	2	10	8	4	45	0.400	0.380	0.02	5.09
50	7	2	10	9	4	40	0.400	0.371	0.03	7.18
50	7	2	10	10	5	61	0.400	0.378	0.02	5.61
50	7	2	30	1	3	37	0.400	0.400	0.00	0.00
50	7	2	30	2	5	159	0.400	0.392	0.01	1.99
50	7	2	30	3	5	63	0.400	0.396	0.00	0.89
50	7	2	30	4	5	79	0.400	0.385	0.01	3.69
50	7	2	30	5	5	99	0.400	0.377	0.02	5.77
50	7	2	30	6	5	110	0.400	0.387	0.01	3.34
50	7	2	30	7	4	50	0.400	0.377	0.02	5.82
50	7	2	30	8	4	54	0.400	0.365	0.03	8.71
50	7	2	30	9	5	68	0.400	0.382	0.02	4.40
50	7	2	30	10	5	97	0.400	0.386	0.01	3.55
50	7	2	50	1	5	70	0.400	0.377	0.02	5.69

50	7	2	50	2	4	58	0.400	0.400	0.00	0.00
50	7	2	50	3	5	76	0.400	0.397	0.00	0.65
50	7	2	50	4	5	87	0.400	0.381	0.02	4.69
50	7	2	50	5	5	174	0.400	0.396	0.00	0.92
50	7	2	50	6	5	103	0.400	0.389	0.01	2.82
50	7	2	50	7	5	118	0.400	0.382	0.02	4.48
50	7	2	50	8	5	77	0.400	0.381	0.02	4.79
50	7	2	50	9	5	96	0.400	0.380	0.02	5.11
50	7	2	50	10	4	51	0.400	0.384	0.02	4.08
100	3	2	10	1	6	73	0.400	0.357	0.04	10.76
100	3	2	10	2	5	59	0.400	0.390	0.01	2.44
100	3	2	10	3	6	97	0.400	0.383	0.02	4.35
100	3	2	10	4	6	74	0.400	0.340	0.06	15.04
100	3	2	10	5	6	77	0.400	0.388	0.01	2.96
100	3	2	10	6	6	106	0.400	0.358	0.04	10.62
100	3	2	10	7	6	70	0.400	0.351	0.05	12.23
100	3	2	10	8	6	78	0.400	0.377	0.02	5.75
100	3	2	10	9	4	46	0.400	0.394	0.01	1.38
100	3	2	10	10	6	110	0.400	0.374	0.03	6.62
100	3	2	30	1	5	70	0.400	0.400	0.00	0.00
100	3	2	30	2	6	122	0.400	0.388	0.01	3.01
100	3	2	30	3	6	105	0.400	0.384	0.02	3.96
100	3	2	30	4	6	120	0.400	0.372	0.03	7.00
100	3	2	30	5	6	105	0.400	0.392	0.01	1.97
100	3	2	30	6	6	110	0.400	0.384	0.02	3.94
100	3	2	30	7	6	121	0.400	0.392	0.01	1.98
100	3	2	30	8	6	139	0.400	0.389	0.01	2.66
100	3	2	30	9	5	82	0.400	0.392	0.01	2.11
100	3	2	30	10	3	38	0.400	0.400	0.00	0.00
100	3	2	50	1	6	115	0.400	0.380	0.02	5.10
100	3	2	50	2	6	175	0.400	0.389	0.01	2.66
100	3	2	50	3	5	82	0.400	0.398	0.00	0.52
100	3	2	50	4	6	156	0.400	0.388	0.01	2.96
100	3	2	50	5	6	146	0.400	0.380	0.02	4.94
100	3	2	50	6	6	130	0.400	0.382	0.02	4.57
100	3	2	50	7	5	82	0.400	0.396	0.00	0.95
100	3	2	50	8	6	178	0.400	0.389	0.01	2.71

100	3	2	50	9	5	79	0.400	0.388	0.01	3.04
100	3	2	50	10	5	89	0.400	0.398	0.00	0.55
100	5	2	10	1	6	64	0.400	0.361	0.04	9.79
100	5	2	10	2	6	66	0.400	0.396	0.00	1.12
100	5	2	10	3	6	64	0.400	0.392	0.01	2.05
100	5	2	10	4	6	69	0.400	0.354	0.05	11.49
100	5	2	10	5	6	67	0.400	0.369	0.03	7.76
100	5	2	10	6	6	66	0.400	0.381	0.02	4.87
100	5	2	10	7	5	46	0.400	0.400	0.00	0.00
100	5	2	10	8	5	68	0.400	0.394	0.01	1.55
100	5	2	10	9	6	71	0.400	0.393	0.01	1.83
100	5	2	10	10	5	74	0.400	0.373	0.03	6.65
100	5	2	30	1	6	171	0.400	0.386	0.01	3.59
100	5	2	30	2	6	193	0.400	0.390	0.01	2.47
100	5	2	30	3	6	131	0.400	0.384	0.02	3.90
100	5	2	30	4	6	160	0.400	0.381	0.02	4.84
100	5	2	30	5	6	107	0.400	0.382	0.02	4.46
100	5	2	30	6	4	63	0.400	0.394	0.01	1.47
100	5	2	30	7	5	91	0.400	0.385	0.01	3.70
100	5	2	30	8	6	185	0.400	0.392	0.01	2.12
100	5	2	30	9	5	83	0.400	0.368	0.03	7.90
100	5	2	30	10	5	65	0.400	0.391	0.01	2.14
100	5	2	50	1	6	154	0.400	0.379	0.02	5.31
100	5	2	50	2	6	286	0.400	0.395	0.01	1.35
100	5	2	50	3	6	212	0.400	0.391	0.01	2.30
100	5	2	50	4	6	128	0.400	0.375	0.03	6.34
100	5	2	50	5	6	150	0.400	0.372	0.03	7.01
100	5	2	50	6	6	140	0.400	0.384	0.02	3.89
100	5	2	50	7	5	88	0.400	0.376	0.02	5.94
100	5	2	50	8	6	143	0.400	0.376	0.02	5.98
100	5	2	50	9	6	184	0.400	0.380	0.02	4.96
100	5	2	50	10	6	127	0.400	0.378	0.02	5.52
100	7	2	10	1	6	68	0.400	0.368	0.03	7.96
100	7	2	10	2	5	53	0.400	0.390	0.01	2.44
100	7	2	10	3	5	54	0.400	0.370	0.03	7.44
100	7	2	10	4	5	58	0.400	0.374	0.03	6.39
100	7	2	10	5	5	40	0.400	0.375	0.02	6.19

100	7	2	10	6	6	59	0.400	0.380	0.02	4.94
100	7	2	10	7	4	38	0.400	0.400	0.00	0.00
100	7	2	10	8	6	50	0.400	0.368	0.03	7.93
100	7	2	10	9	4	29	0.400	0.400	0.00	0.00
100	7	2	10	10	5	41	0.400	0.398	0.00	0.51
100	7	2	30	1	6	123	0.400	0.372	0.03	7.10
100	7	2	30	2	6	120	0.400	0.379	0.02	5.37
100	7	2	30	3	6	67	0.400	0.368	0.03	8.02
100	7	2	30	4	6	62	0.400	0.378	0.02	5.54
100	7	2	30	5	6	116	0.400	0.361	0.04	9.82
100	7	2	30	6	6	183	0.400	0.390	0.01	2.58
100	7	2	30	7	6	115	0.400	0.377	0.02	5.73
100	7	2	30	8	5	76	0.400	0.395	0.00	1.25
100	7	2	30	9	6	128	0.400	0.372	0.03	7.10
100	7	2	30	10	6	155	0.400	0.385	0.01	3.65
100	7	2	50	1	6	155	0.400	0.377	0.02	5.78
100	7	2	50	2	6	172	0.400	0.389	0.01	2.63
100	7	2	50	3	6	108	0.400	0.385	0.02	3.80
100	7	2	50	4	6	292	0.400	0.391	0.01	2.24
100	7	2	50	5	6	199	0.400	0.380	0.02	5.10
100	7	2	50	6	6	221	0.400	0.384	0.02	4.00
100	7	2	50	7	5	121	0.400	0.398	0.00	0.57
100	7	2	50	8	6	197	0.400	0.392	0.01	1.90
100	7	2	50	9	6	260	0.400	0.390	0.01	2.38
100	7	2	50	10	6	125	0.400	0.378	0.02	5.52
200	3	2	10	1	5	64	0.400	0.397	0.00	0.86
200	3	2	10	2	7	107	0.400	0.390	0.01	2.50
200	3	2	10	3	6	82	0.400	0.373	0.03	6.63
200	3	2	10	4	6	82	0.400	0.381	0.02	4.71
200	3	2	10	5	5	78	0.400	0.400	0.00	0.00
200	3	2	10	6	7	134	0.400	0.381	0.02	4.63
200	3	2	10	7	6	105	0.400	0.390	0.01	2.62
200	3	2	10	8	7	154	0.400	0.364	0.04	9.09
200	3	2	10	9	5	44	0.400	0.400	0.00	0.08
200	3	2	10	10	6	89	0.400	0.399	0.00	0.24
200	3	2	30	1	7	220	0.400	0.380	0.02	4.94
200	3	2	30	2	7	200	0.400	0.381	0.02	4.76

200	3	2	30	3	7	136	0.400	0.382	0.02	4.44
200	3	2	30	4	7	188	0.400	0.380	0.02	4.99
200	3	2	30	5	7	176	0.400	0.378	0.02	5.41
200	3	2	30	6	5	99	0.400	0.400	0.00	0.11
200	3	2	30	7	6	147	0.400	0.383	0.02	4.28
200	3	2	30	8	7	199	0.400	0.381	0.02	4.85
200	3	2	30	9	5	86	0.400	0.400	0.00	0.00
200	3	2	30	10	4	65	0.400	0.400	0.00	0.00
200	3	2	50	1	6	145	0.400	0.399	0.00	0.31
200	3	2	50	2	7	204	0.400	0.387	0.01	3.32
200	3	2	50	3	7	143	0.400	0.384	0.02	4.07
200	3	2	50	4	7	216	0.400	0.386	0.01	3.43
200	3	2	50	5	6	133	0.400	0.385	0.01	3.67
200	3	2	50	6	7	219	0.400	0.383	0.02	4.23
200	3	2	50	7	7	273	0.400	0.388	0.01	2.88
200	3	2	50	8	7	179	0.400	0.390	0.01	2.50
200	3	2	50	9	4	77	0.400	0.400	0.00	0.00
200	3	2	50	10	7	113	0.400	0.384	0.02	4.10
200	5	2	10	1	7	86	0.400	0.362	0.04	9.46
200	5	2	10	2	7	158	0.400	0.369	0.03	7.72
200	5	2	10	3	7	112	0.400	0.391	0.01	2.32
200	5	2	10	4	6	128	0.400	0.385	0.01	3.67
200	5	2	10	5	6	63	0.400	0.396	0.00	0.95
200	5	2	10	6	6	92	0.400	0.379	0.02	5.16
200	5	2	10	7	6	75	0.400	0.390	0.01	2.61
200	5	2	10	8	7	126	0.400	0.378	0.02	5.38
200	5	2	10	9	6	69	0.400	0.391	0.01	2.29
200	5	2	10	10	7	140	0.400	0.379	0.02	5.20
200	5	2	30	1	7	188	0.400	0.383	0.02	4.15
200	5	2	30	2	7	347	0.400	0.379	0.02	5.36
200	5	2	30	3	7	228	0.400	0.382	0.02	4.39
200	5	2	30	4	6	119	0.400	0.393	0.01	1.64
200	5	2	30	5	5	81	0.400	0.395	0.00	1.19
200	5	2	30	6	7	222	0.400	0.372	0.03	6.92
200	5	2	30	7	6	154	0.400	0.387	0.01	3.15
200	5	2	30	8	6	127	0.400	0.391	0.01	2.33
200	5	2	30	9	5	79	0.400	0.389	0.01	2.70

200	5	2	30	10	6	134	0.400	0.380	0.02	4.90
200	5	2	50	1	7	343	0.400	0.381	0.02	4.63
200	5	2	50	2	6	183	0.400	0.381	0.02	4.81
200	5	2	50	3	7	333	0.400	0.387	0.01	3.29
200	5	2	50	4	7	301	0.400	0.373	0.03	6.63
200	5	2	50	5	7	302	0.400	0.390	0.01	2.38
200	5	2	50	6	7	227	0.400	0.386	0.01	3.46
200	5	2	50	7	7	220	0.400	0.379	0.02	5.21
200	5	2	50	8	7	325	0.400	0.379	0.02	5.27
200	5	2	50	9	7	326	0.400	0.381	0.02	4.68
200	5	2	50	10	4	97	0.400	0.400	0.00	0.00
200	7	2	10	1	6	107	0.400	0.383	0.02	4.15
200	7	2	10	2	7	128	0.400	0.382	0.02	4.48
200	7	2	10	3	7	111	0.400	0.372	0.03	7.02
200	7	2	10	4	6	99	0.400	0.387	0.01	3.37
200	7	2	10	5	7	106	0.400	0.383	0.02	4.15
200	7	2	10	6	7	131	0.400	0.371	0.03	7.34
200	7	2	10	7	7	108	0.400	0.371	0.03	7.36
200	7	2	10	8	6	86	0.400	0.375	0.02	6.21
200	7	2	10	9	6	77	0.400	0.380	0.02	5.11
200	7	2	10	10	7	149	0.400	0.377	0.02	5.64
200	7	2	30	1	7	253	0.400	0.377	0.02	5.68
200	7	2	30	2	6	145	0.400	0.383	0.02	4.21
200	7	2	30	3	7	176	0.400	0.383	0.02	4.22
200	7	2	30	4	7	223	0.400	0.376	0.02	5.94
200	7	2	30	5	7	207	0.400	0.382	0.02	4.40
200	7	2	30	6	7	158	0.400	0.377	0.02	5.67
200	7	2	30	7	7	281	0.400	0.388	0.01	2.97
200	7	2	30	8	7	248	0.400	0.394	0.01	1.48
200	7	2	30	9	7	202	0.400	0.389	0.01	2.66
200	7	2	30	10	6	130	0.400	0.369	0.03	7.71
200	7	2	50	1	7	225	0.400	0.329	0.07	17.64
200	7	2	50	2	7	222	0.400	0.374	0.03	6.60
200	7	2	50	3	6	139	0.400	0.394	0.01	1.56
200	7	2	50	4	7	225	0.400	0.380	0.02	5.03
200	7	2	50	5	7	265	0.400	0.375	0.02	6.23
200	7	2	50	6	7	262	0.400	0.395	0.00	1.22

200	7	2	50	7	7	172	0.400	0.376	0.02	5.92
200	7	2	50	8	7	258	0.400	0.381	0.02	4.73
200	7	2	50	9	7	179	0.400	0.387	0.01	3.22
200	7	2	50	10	7	252	0.400	0.392	0.01	2.10

For the second set weight, the average gap value is 7.53% which means, we found a solution almost 7% close to the optimal solution. During the experiments of the case of 2 classes under equal weights, 6.11 questions on average are asked to the decision maker and it takes 98.87 seconds on average takes to solve a problem instance.

Table 4: The Case of 2 Classes Experimental Results for Set Weight No.2

OBJ	**CRI**	**CLASS**	**ACT**	**INS**	**QUEST.**	**CPU**	**ALG**	**DM**	**DIF**	**GAP %**
50	3	2	10	1	3	45	0.500	0.456	0.044	8.85
50	3	2	10	2	5	63	0.440	0.393	0.048	10.80
50	3	2	10	3	5	63	0.372	0.372	0.000	0.00
50	3	2	10	4	4	45	0.497	0.497	0.000	0.00
50	3	2	10	5	5	52	0.436	0.426	0.010	2.19
50	3	2	10	6	6	66	0.387	0.385	0.002	0.52
50	3	2	10	7	5	54	0.369	0.369	0.000	0.00
50	3	2	10	8	6	89	0.482	0.397	0.085	17.68
50	3	2	10	9	6	79	0.488	0.395	0.093	19.05
50	3	2	10	10	6	58	0.408	0.406	0.002	0.42
50	3	2	30	1	6	74	0.435	0.408	0.027	6.17
50	3	2	30	2	6	61	0.399	0.398	0.001	0.25
50	3	2	30	3	6	85	0.486	0.403	0.084	17.24
50	3	2	30	4	5	42	0.428	0.428	0.000	0.06
50	3	2	30	5	6	77	0.448	0.414	0.033	7.46
50	3	2	30	6	5	69	0.500	0.382	0.118	23.50
50	3	2	30	7	3	29	0.500	0.500	0.000	0.00
50	3	2	30	8	5	46	0.436	0.433	0.003	0.61
50	3	2	30	9	5	45	0.408	0.408	0.000	0.00
50	3	2	30	10	6	76	0.431	0.418	0.013	3.05
50	3	2	50	1	5	73	0.414	0.396	0.018	4.34
50	3	2	50	2	6	67	0.402	0.402	0.000	0.00
50	3	2	50	3	5	56	0.414	0.412	0.002	0.53
50	3	2	50	4	5	51	0.413	0.413	0.000	0.00
50	3	2	50	5	5	60	0.483	0.468	0.015	3.19
50	3	2	50	6	3	35	0.500	0.477	0.023	4.63

50	3	2	50	7	2	28	0.500	0.500	0.000	0.00
50	3	2	50	8	5	53	0.403	0.401	0.002	0.43
50	3	2	50	9	2	24	0.500	0.500	0.000	0.00
50	3	2	50	10	4	40	0.489	0.489	0.000	0.00
50	5	2	10	1	5	46	0.413	0.382	0.03	7.485
50	5	2	10	2	5	56	0.473	0.406	0.07	14.029
50	5	2	10	3	5	64	0.433	0.379	0.05	12.488
50	5	2	10	4	4	54	0.450	0.442	0.01	1.957
50	5	2	10	5	6	83	0.377	0.374	0.00	0.610
50	5	2	10	6	5	70	0.500	0.402	0.10	19.658
50	5	2	10	7	6	78	0.414	0.381	0.03	8.032
50	5	2	10	8	4	57	0.482	0.481	0.00	0.259
50	5	2	10	9	4	49	0.500	0.412	0.09	17.595
50	5	2	10	10	5	65	0.438	0.416	0.02	5.078
50	5	2	30	1	6	45	0.412	0.403	0.01	2.271
50	5	2	30	2	5	39	0.451	0.445	0.01	1.167
50	5	2	30	3	5	55	0.473	0.396	0.08	16.320
50	5	2	30	4	6	87	0.474	0.423	0.05	10.690
50	5	2	30	5	6	55	0.466	0.384	0.08	17.586
50	5	2	30	6	5	47	0.382	0.381	0.00	0.409
50	5	2	30	7	3	32	0.500	0.500	0.00	0.000
50	5	2	30	8	5	51	0.437	0.400	0.04	8.357
50	5	2	30	9	5	64	0.434	0.410	0.02	5.472
50	5	2	30	10	6	102	0.491	0.420	0.07	14.461
50	5	2	50	1	5	45	0.403	0.403	0.00	0.000
50	5	2	50	2	5	51	0.466	0.418	0.05	10.277
50	5	2	50	3	5	58	0.396	0.389	0.01	1.756
50	5	2	50	4	5	55	0.394	0.392	0.00	0.457
50	5	2	50	5	5	49	0.412	0.412	0.00	0.000
50	5	2	50	6	6	67	0.453	0.396	0.06	12.45
50	5	2	50	7	6	67	0.446	0.408	0.04	8.72
50	5	2	50	8	6	66	0.407	0.401	0.01	1.39
50	5	2	50	9	5	64	0.434	0.409	0.02	5.66
50	5	2	50	10	6	57	0.458	0.386	0.07	15.68
50	7	2	10	1	4	30	0.5	0.389	0.11	22.17
50	7	2	10	2	6	48	0.484	0.389	0.1	19.69
50	7	2	10	3	5	36	0.418	0.401	0.02	4.11

50	7	2	10	4	5	41	0.5	0.4	0.1	20.06
50	7	2	10	5	6	50	0.488	0.404	0.08	17.21
50	7	2	10	6	5	38	0.399	0.396	0.00	0.75
50	7	2	10	7	5	32	0.445	0.445	0.00	0.00
50	7	2	10	8	5	35	0.433	0.422	0.01	2.43
50	7	2	10	9	4	31	0.5	0.39	0.11	22.06
50	7	2	10	10	6	48	0.456	0.397	0.06	12.86
50	7	2	30	1	3	26	0.5	0.482	0.02	3.57
50	7	2	30	2	6	101	0.471	0.423	0.05	10.16
50	7	2	30	3	5	45	0.412	0.411	0.00	0.17
50	7	2	30	4	6	56	0.449	0.403	0.05	10.13
50	7	2	30	5	6	90	0.484	0.406	0.08	16.14
50	7	2	30	6	6	91	0.489	0.425	0.06	12.99
50	7	2	30	7	4	36	0.442	0.4	0.04	9.37
50	7	2	30	8	5	42	0.44	0.404	0.04	8.06
50	7	2	30	9	5	50	0.5	0.399	0.1	20.21
50	7	2	30	10	6	71	0.465	0.418	0.05	9.91
50	7	2	50	1	5	55	0.5	0.408	0.09	18.45
50	7	2	50	2	4	46	0.5	0.5	0.00	0.00
50	7	2	50	3	5	53	0.425	0.425	0.00	0.02
50	7	2	50	4	6	67	0.406	0.406	0.00	0.00
50	7	2	50	5	6	107	0.481	0.432	0.05	10.02
50	7	2	50	6	6	76	0.477	0.427	0.05	10.37
50	7	2	50	7	6	91	0.414	0.403	0.01	2.81
50	7	2	50	8	6	64	0.424	0.411	0.01	3.18
50	7	2	50	9	6	71	0.395	0.387	0.01	1.99
50	7	2	50	10	5	50	0.41	0.41	0.00	0.00
100	3	2	10	1	7	93	0.36	0.356	0.00	1.13
100	3	2	10	2	5	63	0.472	0.468	0.00	0.95
100	3	2	10	3	7	76	0.438	0.403	0.04	8.00
100	3	2	10	4	6	81	0.42	0.368	0.05	12.42
100	3	2	10	5	6	86	0.457	0.402	0.05	11.97
100	3	2	10	6	7	79	0.401	0.375	0.03	6.34
100	3	2	10	7	6	49	0.373	0.365	0.01	1.97
100	3	2	10	8	7	65	0.425	0.376	0.05	11.6
100	3	2	10	9	5	49	0.484	0.47	0.01	2.91
100	3	2	10	10	7	107	0.489	0.396	0.09	19.11

100	3	2	30	1	5	48	0.49	0.485	0.00	0.95
100	3	2	30	2	7	80	0.413	0.41	0.00	0.74
100	3	2	30	3	6	75	0.436	0.401	0.04	8.08
100	3	2	30	4	7	96	0.408	0.378	0.03	7.48
100	3	2	30	5	6	71	0.406	0.405	0.00	0.3
100	3	2	30	6	7	68	0.467	0.411	0.06	11.98
100	3	2	30	7	7	82	0.414	0.412	0.00	0.48
100	3	2	30	8	7	90	0.442	0.4	0.04	9.65
100	3	2	30	9	6	68	0.406	0.403	0.00	0.66
100	3	2	30	10	3	29	0.481	0.481	0.00	0.00
100	3	2	50	1	6	74	0.405	0.395	0.01	2.5
100	3	2	50	2	7	109	0.454	0.407	0.05	10.25
100	3	2	50	3	5	73	0.5	0.478	0.02	4.40
100	3	2	50	4	7	102	0.444	0.408	0.04	8.22
100	3	2	50	5	7	89	0.479	0.404	0.07	15.6
100	3	2	50	6	6	75	0.405	0.405	0.00	0.00
100	3	2	50	7	6	70	0.49	0.474	0.02	3.33
100	3	2	50	8	7	99	0.441	0.41	0.03	7.04
100	3	2	50	9	5	52	0.456	0.445	0.01	2.36
100	3	2	50	10	5	64	0.5	0.466	0.03	6.82
100	5	2	10	1	6	60	0.453	0.384	0.07	15.24
100	5	2	10	2	6	63	0.421	0.405	0.02	3.86
100	5	2	10	3	6	81	0.435	0.435	0.00	0.04
100	5	2	10	4	7	100	0.464	0.379	0.09	18.38
100	5	2	10	5	6	85	0.5	0.393	0.11	21.47
100	5	2	10	6	6	83	0.482	0.402	0.08	16.51
100	5	2	10	7	5	64	0.5	0.5	0.00	0.00
100	5	2	10	8	6	93	0.487	0.4	0.09	17.92
100	5	2	10	9	7	83	0.408	0.408	0.00	0.00
100	5	2	10	10	6	69	0.468	0.381	0.09	18.47
100	5	2	30	1	7	117	0.479	0.41	0.07	14.58
100	5	2	30	2	6	123	0.475	0.421	0.05	11.39
100	5	2	30	3	6	78	0.418	0.401	0.02	3.98
100	5	2	30	4	7	139	0.493	0.398	0.09	19.13
100	5	2	30	5	6	71	0.416	0.394	0.02	5.33
100	5	2	30	6	5	55	0.484	0.479	0.00	0.90
100	5	2	30	7	6	66	0.411	0.411	0.00	0.00

100	5	2	30	8	7	126	0.422	0.418	0.00	1.09
100	5	2	30	9	5	57	0.416	0.416	0.00	0.00
100	5	2	30	10	5	45	0.47	0.47	0.00	0.00
100	5	2	50	1	7	98	0.403	0.402	0.00	0.30
100	5	2	50	2	7	265	0.495	0.431	0.06	13.08
100	5	2	50	3	7	145	0.459	0.421	0.04	8.24
100	5	2	50	4	6	85	0.407	0.405	0.00	0.60
100	5	2	50	5	6	134	0.5	0.393	0.11	21.37
100	5	2	50	6	7	97	0.404	0.402	0.00	0.62
100	5	2	50	7	5	67	0.479	0.412	0.07	13.90
100	5	2	50	8	7	112	0.477	0.405	0.07	15.04
100	5	2	50	9	7	125	0.437	0.393	0.04	10.16
100	5	2	50	10	6	95	0.442	0.403	0.04	8.85
100	7	2	10	1	6	59	0.5	0.389	0.11	22.21
100	7	2	10	2	6	49	0.453	0.449	0.00	0.75
100	7	2	10	3	6	52	0.455	0.394	0.06	13.49
100	7	2	10	4	6	54	0.4	0.4	0.00	0.00
100	7	2	10	5	6	57	0.419	0.397	0.02	5.40
100	7	2	10	6	6	61	0.414	0.41	0.00	0.75
100	7	2	10	7	5	48	0.486	0.485	0.00	0.38
100	7	2	10	8	6	60	0.479	0.391	0.09	18.38
100	7	2	10	9	4	36	0.5	0.486	0.01	2.88
100	7	2	10	10	6	51	0.436	0.423	0.01	2.89
100	7	2	30	1	7	110	0.443	0.389	0.05	12.04
100	7	2	30	2	7	107	0.479	0.405	0.07	15.44
100	7	2	30	3	6	77	0.5	0.378	0.12	24.49
100	7	2	30	4	6	78	0.5	0.411	0.09	17.79
100	7	2	30	5	7	106	0.454	0.378	0.08	16.65
100	7	2	30	6	7	134	0.481	0.421	0.06	12.33
100	7	2	30	7	7	105	0.474	0.416	0.06	12.23
100	7	2	30	8	5	69	0.5	0.479	0.02	4.27
100	7	2	30	9	7	113	0.378	0.373	0.01	1.38
100	7	2	30	10	7	126	0.435	0.41	0.03	5.84
100	7	2	50	1	6	114	0.413	0.4	0.01	3.15
100	7	2	50	2	7	146	0.436	0.409	0.03	6.22
100	7	2	50	3	6	102	0.403	0.402	0.00	0.12
100	7	2	50	4	7	267	0.494	0.421	0.07	14.78

100	7	2	50	5	7	119	0.469	0.395	0.07	15.74
100	7	2	50	6	7	176	0.478	0.413	0.07	13.59
100	7	2	50	7	6	106	0.423	0.418	0.00	1.03
100	7	2	50	8	7	180	0.494	0.425	0.07	14.01
100	7	2	50	9	7	193	0.475	0.419	0.06	11.71
100	7	2	50	10	6	108	0.442	0.403	0.04	8.85
200	3	2	10	1	6	55	0.474	0.47	0.00	0.84
200	3	2	10	2	8	92	0.414	0.401	0.01	2.93
200	3	2	10	3	7	83	0.41	0.41	0.00	0.00
200	3	2	10	4	6	130	0.433	0.433	0.00	0.00
200	3	2	10	5	5	115	0.499	0.499	0.00	0.00
200	3	2	10	6	7	96	0.39	0.386	0.00	1.03
200	3	2	10	7	7	80	0.406	0.405	0.00	0.21
200	3	2	10	8	7	115	0.487	0.379	0.11	22.17
200	3	2	10	9	5	46	0.5	0.486	0.01	2.88
200	3	2	10	10	6	79	0.441	0.412	0.03	6.58
200	3	2	30	1	8	150	0.407	0.401	0.01	1.50
200	3	2	30	2	7	130	0.482	0.405	0.08	15.95
200	3	2	30	3	7	96	0.422	0.398	0.02	5.68
200	3	2	30	4	7	126	0.405	0.405	0.00	0.00
200	3	2	30	5	7	157	0.5	0.399	0.1	20.22
200	3	2	30	6	6	88	0.495	0.495	0.00	0.00
200	3	2	30	7	7	99	0.483	0.421	0.06	12.87
200	3	2	30	8	7	120	0.405	0.404	0.00	0.31
200	3	2	30	9	6	73	0.484	0.479	0.00	0.93
200	3	2	30	10	5	60	0.499	0.488	0.01	2.19
200	3	2	50	1	7	100	0.419	0.411	0.01	1.94
200	3	2	50	2	8	151	0.464	0.405	0.06	12.73
200	3	2	50	3	7	127	0.456	0.407	0.05	10.83
200	3	2	50	4	8	181	0.407	0.407	0.00	0.00
200	3	2	50	5	6	128	0.5	0.454	0.05	9.16
200	3	2	50	6	7	140	0.491	0.408	0.08	16.79
200	3	2	50	7	7	245	0.455	0.412	0.04	9.48
200	3	2	50	8	8	193	0.409	0.409	0.00	0.11
200	3	2	50	9	4	89	0.468	0.468	0.00	0.00
200	3	2	50	10	7	175	0.413	0.413	0.00	0.15
200	5	2	10	1	7	80	0.5	0.388	0.11	22.38

200	5	2	10	2	8	108	0.467	0.387	0.08	17.17
200	5	2	10	3	7	70	0.466	0.389	0.08	16.71
200	5	2	10	4	7	87	0.399	0.395	0.00	0.89
200	5	2	10	5	6	49	0.489	0.489	0.00	0.00
200	5	2	10	6	6	62	0.407	0.404	0.00	0.8
200	5	2	10	7	7	66	0.492	0.441	0.05	10.36
200	5	2	10	8	7	84	0.414	0.403	0.01	2.65
200	5	2	10	9	7	59	0.463	0.463	0.00	0.00
200	5	2	10	10	7	93	0.407	0.399	0.01	1.83
200	5	2	30	1	8	128	0.405	0.405	0.00	0.02
200	5	2	30	2	8	168	0.398	0.387	0.01	2.78
200	5	2	30	3	8	140	0.415	0.395	0.02	4.65
200	5	2	30	4	6	83	0.44	0.432	0.01	1.65
200	5	2	30	5	5	65	0.5	0.486	0.01	2.71
200	5	2	30	6	8	151	0.395	0.387	0.01	2.08
200	5	2	30	7	7	110	0.408	0.408	0.00	0.00
200	5	2	30	8	6	89	0.459	0.454	0.01	1.18
200	5	2	30	9	5	65	0.465	0.465	0.00	0.00
200	5	2	30	10	7	102	0.454	0.403	0.05	11.24
200	5	2	50	1	8	249	0.488	0.407	0.08	16.63
200	5	2	50	2	6	138	0.449	0.432	0.02	3.87
200	5	2	50	3	8	255	0.461	0.416	0.05	9.86
200	5	2	50	4	8	238	0.389	0.388	0.00	0.15
200	5	2	50	5	8	236	0.455	0.413	0.04	9.04
200	5	2	50	6	7	202	0.472	0.405	0.07	14.32
200	5	2	50	7	7	219	0.5	0.403	0.1	19.44
200	5	2	50	8	8	249	0.489	0.396	0.09	19.11
200	5	2	50	9	8	283	0.467	0.399	0.07	14.55
200	5	2	50	10	4	148	0.494	0.494	0.00	0.1
200	7	2	10	1	7	105	0.399	0.399	0.00	0.00
200	7	2	10	2	7	106	0.407	0.407	0.00	0.04
200	7	2	10	3	8	106	0.399	0.395	0.00	0.88
200	7	2	10	4	7	97	0.406	0.394	0.01	3.00
200	7	2	10	5	8	104	0.41	0.41	0.00	0.08
200	7	2	10	6	7	105	0.484	0.404	0.08	16.48
200	7	2	10	7	7	100	0.421	0.395	0.03	6.21
200	7	2	10	8	7	90	0.405	0.405	0.00	0.01

200	7	2	10	9	6	76	0.5	0.437	0.06	12.5
200	7	2	10	10	8	156	0.496	0.401	0.1	19.17
200	7	2	30	1	8	195	0.463	0.395	0.07	14.52
200	7	2	30	2	7	138	0.414	0.413	0.00	0.16
200	7	2	30	3	7	153	0.408	0.405	0.00	0.68
200	7	2	30	4	7	221	0.5	0.394	0.11	21.21
200	7	2	30	5	7	205	0.5	0.402	0.1	19.62
200	7	2	30	6	7	151	0.454	0.396	0.06	12.83
200	7	2	30	7	8	251	0.483	0.41	0.07	15.12
200	7	2	30	8	8	198	0.428	0.426	0.00	0.41
200	7	2	30	9	8	247	0.497	0.42	0.08	15.49
200	7	2	30	10	7	147	0.436	0.394	0.04	9.72
200	7	2	50	1	7	265	0.5	0.326	0.17	34.85
200	7	2	50	2	8	286	0.485	0.399	0.09	17.71
200	7	2	50	3	6	180	0.488	0.465	0.02	4.66
200	7	2	50	4	8	254	0.49	0.399	0.09	18.48
200	7	2	50	5	7	327	0.5	0.397	0.1	20.67
200	7	2	50	6	8	370	0.498	0.43	0.07	13.56
200	7	2	50	7	8	238	0.457	0.4	0.06	12.31
200	7	2	50	8	8	411	0.449	0.395	0.05	11.98
200	7	2	50	9	8	191	0.493	0.408	0.08	17.19
200	7	2	50	10	8	264	0.497	0.426	0.07	14.45

For the third set weight, the average gap value is 0.24%. During the experiments of the case of 2 classes under the weights of 70%-30%, 5.54 questions on average are asked to the decision maker and it takes 106.55 seconds on average takes to solve a problem instance.

Table 5: The Case of 2 Classes Experimental Results for Set Weight No.3

OBJ	CRI	CLASS	ACT	INS	QUEST.	CPU	ALG	DM	DIF	GAP %
50	3	2	10	1	3	24	0.300	0.300	0.00	0.00
50	3	2	10	2	5	44	0.300	0.290	0.01	3.37
50	3	2	10	3	5	45	0.300	0.296	0.00	1.26
50	3	2	10	4	3	29	0.300	0.300	0.00	0.00
50	3	2	10	5	4	42	0.300	0.300	0.00	0.00
50	3	2	10	6	5	48	0.300	0.298	0.00	0.53
50	3	2	10	7	5	47	0.300	0.299	0.00	0.20
50	3	2	10	8	5	54	0.300	0.297	0.00	0.85
50	3	2	10	9	5	59	0.300	0.300	0.00	0.00

50	3	2	10	10	5	58	0.300	0.300	0.00	0.00
50	3	2	30	1	5	77	0.300	0.300	0.00	0.00
50	3	2	30	2	5	75	0.300	0.300	0.00	0.00
50	3	2	30	3	5	72	0.300	0.300	0.00	0.00
50	3	2	30	4	4	41	0.300	0.300	0.00	0.00
50	3	2	30	5	5	68	0.300	0.300	0.00	0.00
50	3	2	30	6	5	65	0.300	0.299	0.00	0.42
50	3	2	30	7	3	29	0.300	0.300	0.00	0.00
50	3	2	30	8	4	41	0.300	0.300	0.00	0.00
50	3	2	30	9	4	43	0.300	0.300	0.00	0.00
50	3	2	30	10	5	68	0.300	0.300	0.00	0.00
50	3	2	50	1	5	77	0.300	0.300	0.00	0.00
50	3	2	50	2	5	61	0.300	0.300	0.00	0.00
50	3	2	50	3	5	63	0.300	0.300	0.00	0.00
50	3	2	50	4	5	67	0.300	0.300	0.00	0.00
50	3	2	50	5	4	42	0.300	0.300	0.00	0.00
50	3	2	50	6	3	33	0.300	0.300	0.00	0.00
50	3	2	50	7	2	25	0.300	0.300	0.00	0.00
50	3	2	50	8	5	67	0.300	0.300	0.00	0.00
50	3	2	50	9	2	26	0.300	0.300	0.00	0.00
50	3	2	50	10	3	36	0.300	0.300	0.00	0.00
50	5	2	10	1	4	37	0.300	0.300	0.00	0.00
50	5	2	10	2	5	43	0.300	0.300	0.00	0.00
50	5	2	10	3	5	47	0.300	0.300	0.00	0.00
50	5	2	10	4	4	40	0.300	0.300	0.00	0.00
50	5	2	10	5	5	53	0.300	0.299	0.00	0.37
50	5	2	10	6	5	43	0.300	0.299	0.00	0.29
50	5	2	10	7	5	52	0.300	0.299	0.00	0.40
50	5	2	10	8	3	32	0.300	0.300	0.00	0.00
50	5	2	10	9	4	35	0.300	0.300	0.00	0.00
50	5	2	10	10	5	44	0.300	0.300	0.00	0.00
50	5	2	30	1	5	49	0.300	0.300	0.00	0.00
50	5	2	30	2	4	41	0.300	0.300	0.00	0.00
50	5	2	30	3	5	65	0.300	0.300	0.00	0.00
50	5	2	30	4	5	109	0.300	0.300	0.00	0.00
50	5	2	30	5	5	62	0.300	0.300	0.00	0.00
50	5	2	30	6	5	56	0.300	0.300	0.00	0.00

50	5	2	30	7	3	34	0.300	0.300	0.00	0.00
50	5	2	30	8	5	61	0.300	0.300	0.00	0.00
50	5	2	30	9	5	76	0.300	0.300	0.00	0.00
50	5	2	30	10	5	89	0.300	0.300	0.00	0.00
50	5	2	50	1	4	43	0.300	0.300	0.00	0.02
50	5	2	50	2	4	52	0.300	0.300	0.00	0.00
50	5	2	50	3	5	65	0.300	0.300	0.00	0.00
50	5	2	50	4	5	64	0.300	0.300	0.00	0.00
50	5	2	50	5	4	44	0.300	0.300	0.00	0.00
50	5	2	50	6	5	73	0.300	0.299	0.00	0.29
50	5	2	50	7	5	78	0.300	0.300	0.00	0.00
50	5	2	50	8	5	79	0.300	0.300	0.00	0.00
50	5	2	50	9	5	87	0.300	0.300	0.00	0.00
50	5	2	50	10	5	64	0.300	0.300	0.00	0.00
50	7	2	10	1	4	31	0.300	0.282	0.02	6.02
50	7	2	10	2	5	45	0.300	0.299	0.00	0.30
50	7	2	10	3	5	40	0.300	0.300	0.00	0.00
50	7	2	10	4	5	50	0.300	0.294	0.01	2.04
50	7	2	10	5	5	47	0.300	0.300	0.00	0.00
50	7	2	10	6	4	41	0.300	0.300	0.00	0.01
50	7	2	10	7	4	35	0.300	0.300	0.00	0.00
50	7	2	10	8	4	34	0.300	0.300	0.00	0.00
50	7	2	10	9	4	38	0.300	0.300	0.00	0.00
50	7	2	10	10	5	49	0.300	0.300	0.00	0.00
50	7	2	30	1	3	41	0.300	0.300	0.00	0.00
50	7	2	30	2	5	123	0.300	0.300	0.00	0.00
50	7	2	30	3	5	50	0.300	0.300	0.00	0.00
50	7	2	30	4	5	64	0.300	0.300	0.00	0.00
50	7	2	30	5	5	124	0.300	0.299	0.00	0.49
50	7	2	30	6	5	116	0.300	0.300	0.00	0.00
50	7	2	30	7	4	62	0.300	0.299	0.00	0.49
50	7	2	30	8	4	63	0.300	0.300	0.00	0.00
50	7	2	30	9	5	83	0.300	0.300	0.00	0.00
50	7	2	30	10	5	106	0.300	0.300	0.00	0.00
50	7	2	50	1	5	73	0.300	0.300	0.00	0.00
50	7	2	50	2	4	64	0.300	0.300	0.00	0.00
50	7	2	50	3	5	77	0.300	0.300	0.00	0.00

50	7	2	50	4	5	83	0.300	0.300	0.00	0.00
50	7	2	50	5	5	162	0.300	0.300	0.00	0.00
50	7	2	50	6	5	126	0.300	0.299	0.00	0.21
50	7	2	50	7	5	154	0.300	0.300	0.00	0.00
50	7	2	50	8	5	123	0.300	0.300	0.00	0.00
50	7	2	50	9	5	134	0.300	0.300	0.00	0.00
50	7	2	50	10	4	41	0.300	0.300	0.00	0.00
100	3	2	10	1	6	60	0.300	0.294	0.01	2.12
100	3	2	10	2	5	49	0.300	0.300	0.00	0.00
100	3	2	10	3	6	78	0.300	0.300	0.00	0.00
100	3	2	10	4	6	100	0.300	0.276	0.02	7.92
100	3	2	10	5	6	73	0.300	0.300	0.00	0.00
100	3	2	10	6	6	95	0.300	0.298	0.00	0.74
100	3	2	10	7	6	65	0.300	0.292	0.01	2.55
100	3	2	10	8	6	76	0.300	0.298	0.00	0.56
100	3	2	10	9	4	44	0.300	0.300	0.00	0.00
100	3	2	10	10	6	110	0.300	0.297	0.00	1.15
100	3	2	30	1	5	68	0.300	0.300	0.00	0.00
100	3	2	30	2	6	107	0.300	0.300	0.00	0.00
100	3	2	30	3	6	113	0.300	0.300	0.00	0.00
100	3	2	30	4	6	133	0.300	0.300	0.00	0.10
100	3	2	30	5	6	107	0.300	0.300	0.00	0.00
100	3	2	30	6	6	110	0.300	0.299	0.00	0.17
100	3	2	30	7	6	121	0.300	0.300	0.00	0.00
100	3	2	30	8	6	136	0.300	0.300	0.00	0.00
100	3	2	30	9	5	85	0.300	0.300	0.00	0.00
100	3	2	30	10	3	42	0.300	0.300	0.00	0.00
100	3	2	50	1	6	120	0.300	0.299	0.00	0.25
100	3	2	50	2	6	175	0.300	0.300	0.00	0.00
100	3	2	50	3	5	89	0.300	0.300	0.00	0.00
100	3	2	50	4	6	172	0.300	0.300	0.00	0.00
100	3	2	50	5	6	150	0.300	0.300	0.00	0.02
100	3	2	50	6	6	138	0.300	0.299	0.00	0.18
100	3	2	50	7	5	87	0.300	0.300	0.00	0.00
100	3	2	50	8	6	171	0.300	0.300	0.00	0.00
100	3	2	50	9	5	78	0.300	0.300	0.00	0.00
100	3	2	50	10	5	86	0.300	0.300	0.00	0.00

100	5	2	10	1	6	69	0.300	0.295	0.01	1.83
100	5	2	10	2	6	65	0.300	0.300	0.00	0.00
100	5	2	10	3	6	64	0.300	0.300	0.00	0.00
100	5	2	10	4	6	69	0.300	0.290	0.01	3.22
100	5	2	10	5	6	68	0.300	0.298	0.00	0.59
100	5	2	10	6	6	67	0.300	0.300	0.00	0.00
100	5	2	10	7	5	48	0.300	0.300	0.00	0.00
100	5	2	10	8	5	68	0.300	0.300	0.00	0.00
100	5	2	10	9	6	71	0.300	0.300	0.00	0.00
100	5	2	10	10	5	76	0.300	0.299	0.00	0.36
100	5	2	30	1	6	177	0.300	0.300	0.00	0.00
100	5	2	30	2	6	179	0.300	0.300	0.00	0.00
100	5	2	30	3	6	121	0.300	0.300	0.00	0.00
100	5	2	30	4	6	158	0.300	0.300	0.00	0.12
100	5	2	30	5	6	104	0.300	0.300	0.00	0.00
100	5	2	30	6	4	62	0.300	0.300	0.00	0.00
100	5	2	30	7	5	97	0.300	0.300	0.00	0.15
100	5	2	30	8	6	213	0.300	0.300	0.00	0.00
100	5	2	30	9	5	79	0.300	0.298	0.00	0.53
100	5	2	30	10	5	64	0.300	0.300	0.00	0.00
100	5	2	50	1	6	144	0.300	0.300	0.00	0.00
100	5	2	50	2	6	265	0.300	0.300	0.00	0.00
100	5	2	50	3	6	184	0.300	0.300	0.00	0.00
100	5	2	50	4	6	143	0.300	0.300	0.00	0.00
100	5	2	50	5	6	144	0.300	0.300	0.00	0.06
100	5	2	50	6	6	136	0.300	0.300	0.00	0.00
100	5	2	50	7	5	81	0.300	0.300	0.00	0.00
100	5	2	50	8	6	151	0.300	0.299	0.00	0.19
100	5	2	50	9	6	202	0.300	0.300	0.00	0.00
100	5	2	50	10	6	131	0.300	0.300	0.00	0.07
100	7	2	10	1	6	70	0.300	0.298	0.00	0.58
100	7	2	10	2	5	55	0.300	0.300	0.00	0.00
100	7	2	10	3	5	95	0.300	0.300	0.00	0.00
100	7	2	10	4	5	96	0.300	0.298	0.00	0.50
100	7	2	10	5	5	45	0.300	0.300	0.00	0.00
100	7	2	10	6	6	67	0.300	0.300	0.00	0.00
100	7	2	10	7	4	43	0.300	0.300	0.00	0.00

100	7	2	10	8	6	60	0.300	0.298	0.00	0.75
100	7	2	10	9	4	34	0.300	0.300	0.00	0.00
100	7	2	10	10	5	41	0.300	0.300	0.00	0.00
100	7	2	30	1	6	133	0.300	0.299	0.00	0.37
100	7	2	30	2	6	105	0.300	0.300	0.00	0.09
100	7	2	30	3	6	75	0.300	0.300	0.00	0.00
100	7	2	30	4	6	63	0.300	0.293	0.01	2.30
100	7	2	30	5	6	92	0.300	0.297	0.00	0.97
100	7	2	30	6	6	150	0.300	0.300	0.00	0.00
100	7	2	30	7	6	71	0.300	0.298	0.00	0.67
100	7	2	30	8	5	55	0.300	0.300	0.00	0.00
100	7	2	30	9	6	118	0.300	0.299	0.00	0.29
100	7	2	30	10	6	139	0.300	0.300	0.00	0.00
100	7	2	50	1	6	122	0.300	0.299	0.00	0.43
100	7	2	50	2	6	137	0.300	0.300	0.00	0.00
100	7	2	50	3	6	86	0.300	0.300	0.00	0.01
100	7	2	50	4	6	236	0.300	0.300	0.00	0.00
100	7	2	50	5	6	160	0.300	0.300	0.00	0.00
100	7	2	50	6	6	188	0.300	0.300	0.00	0.01
100	7	2	50	7	5	89	0.300	0.300	0.00	0.00
100	7	2	50	8	6	161	0.300	0.300	0.00	0.00
100	7	2	50	9	6	225	0.300	0.300	0.00	0.00
100	7	2	50	10	6	101	0.300	0.300	0.00	0.07
200	3	2	10	1	5	59	0.300	0.300	0.00	0.00
200	3	2	10	2	7	89	0.300	0.300	0.00	0.00
200	3	2	10	3	6	76	0.300	0.297	0.00	1.12
200	3	2	10	4	6	69	0.300	0.300	0.00	0.00
200	3	2	10	5	5	66	0.300	0.300	0.00	0.00
200	3	2	10	6	7	112	0.300	0.299	0.00	0.19
200	3	2	10	7	6	95	0.300	0.300	0.00	0.00
200	3	2	10	8	7	150	0.300	0.296	0.00	1.41
200	3	2	10	9	5	55	0.300	0.300	0.00	0.00
200	3	2	10	10	6	89	0.300	0.300	0.00	0.00
200	3	2	30	1	7	193	0.300	0.299	0.00	0.35
200	3	2	30	2	7	175	0.300	0.300	0.00	0.00
200	3	2	30	3	7	109	0.300	0.300	0.00	0.07
200	3	2	30	4	7	187	0.300	0.300	0.00	0.00

200	3	2	30	5	7	153	0.300	0.300	0.00	0.08
200	3	2	30	6	5	82	0.300	0.300	0.00	0.00
200	3	2	30	7	6	123	0.300	0.300	0.00	0.06
200	3	2	30	8	7	186	0.300	0.300	0.00	0.00
200	3	2	30	9	5	73	0.300	0.300	0.00	0.00
200	3	2	30	10	4	59	0.300	0.300	0.00	0.00
200	3	2	50	1	6	128	0.300	0.300	0.00	0.00
200	3	2	50	2	7	175	0.300	0.300	0.00	0.00
200	3	2	50	3	7	117	0.300	0.300	0.00	0.00
200	3	2	50	4	7	195	0.300	0.300	0.00	0.00
200	3	2	50	5	6	118	0.300	0.300	0.00	0.00
200	3	2	50	6	7	193	0.300	0.299	0.00	0.22
200	3	2	50	7	7	240	0.300	0.300	0.00	0.00
200	3	2	50	8	7	163	0.300	0.300	0.00	0.00
200	3	2	50	9	4	66	0.300	0.300	0.00	0.00
200	3	2	50	10	7	149	0.300	0.300	0.00	0.00
200	5	2	10	1	7	97	0.300	0.295	0.00	1.60
200	5	2	10	2	7	148	0.300	0.299	0.00	0.35
200	5	2	10	3	7	84	0.300	0.300	0.00	0.00
200	5	2	10	4	6	119	0.300	0.300	0.00	0.00
200	5	2	10	5	6	53	0.300	0.300	0.00	0.00
200	5	2	10	6	6	81	0.300	0.300	0.00	0.07
200	5	2	10	7	6	66	0.300	0.299	0.00	0.31
200	5	2	10	8	7	116	0.300	0.296	0.00	1.21
200	5	2	10	9	6	55	0.300	0.300	0.00	0.00
200	5	2	10	10	7	127	0.300	0.300	0.00	0.09
200	5	2	30	1	7	171	0.300	0.300	0.00	0.06
200	5	2	30	2	7	320	0.300	0.300	0.00	0.00
200	5	2	30	3	7	198	0.300	0.300	0.00	0.00
200	5	2	30	4	6	100	0.300	0.300	0.00	0.00
200	5	2	30	5	5	65	0.300	0.300	0.00	0.00
200	5	2	30	6	7	179	0.300	0.300	0.00	0.00
200	5	2	30	7	6	125	0.300	0.300	0.00	0.00
200	5	2	30	8	6	120	0.300	0.300	0.00	0.00
200	5	2	30	9	5	85	0.300	0.300	0.00	0.00
200	5	2	30	10	6	123	0.300	0.300	0.00	0.00
200	5	2	50	1	7	354	0.300	0.299	0.00	0.31

200	5	2	50	2	6	211	0.300	0.299	0.00	0.29
200	5	2	50	3	7	372	0.300	0.300	0.00	0.00
200	5	2	50	4	7	290	0.300	0.299	0.00	0.21
200	5	2	50	5	7	298	0.300	0.300	0.00	0.00
200	5	2	50	6	7	228	0.300	0.300	0.00	0.00
200	5	2	50	7	7	183	0.300	0.299	0.00	0.20
200	5	2	50	8	7	241	0.300	0.300	0.00	0.03
200	5	2	50	9	7	241	0.300	0.300	0.00	0.00
200	5	2	50	10	4	57	0.300	0.300	0.00	0.00
200	7	2	10	1	6	87	0.300	0.300	0.00	0.10
200	7	2	10	2	7	105	0.300	0.299	0.00	0.22
200	7	2	10	3	7	75	0.300	0.297	0.00	0.96
200	7	2	10	4	6	76	0.300	0.300	0.00	0.12
200	7	2	10	5	7	74	0.300	0.300	0.00	0.04
200	7	2	10	6	7	100	0.300	0.296	0.00	1.34
200	7	2	10	7	7	79	0.300	0.299	0.00	0.30
200	7	2	10	8	6	65	0.300	0.300	0.00	0.00
200	7	2	10	9	6	55	0.300	0.300	0.00	0.00
200	7	2	10	10	7	122	0.300	0.300	0.00	0.16
200	7	2	30	1	7	275	0.300	0.300	0.00	0.09
200	7	2	30	2	6	105	0.300	0.300	0.00	0.00
200	7	2	30	3	7	128	0.300	0.300	0.00	0.00
200	7	2	30	4	7	188	0.300	0.299	0.00	0.47
200	7	2	30	5	7	177	0.300	0.300	0.00	0.03
200	7	2	30	6	7	240	0.300	0.299	0.00	0.36
200	7	2	30	7	7	258	0.300	0.300	0.00	0.00
200	7	2	30	8	7	301	0.300	0.300	0.00	0.00
200	7	2	30	9	7	173	0.300	0.300	0.00	0.00
200	7	2	30	10	6	113	0.300	0.295	0.00	1.51
200	7	2	50	1	7	235	0.300	0.289	0.01	3.65
200	7	2	50	2	7	229	0.300	0.298	0.00	0.68
200	7	2	50	3	6	151	0.300	0.300	0.00	0.00
200	7	2	50	4	7	231	0.300	0.300	0.00	0.00
200	7	2	50	5	7	338	0.300	0.299	0.00	0.18
200	7	2	50	6	7	239	0.300	0.300	0.00	0.00
200	7	2	50	7	7	152	0.300	0.300	0.00	0.00
200	7	2	50	8	7	238	0.300	0.300	0.00	0.00

200	7	2	50	9	7	155	0.300	0.300	0.00	0.00
200	7	2	50	10	7	187	0.300	0.300	0.00	0.00

5.2 The Case of 3 Classes

In this chapter, the results of the case for 3 classes will be represented. These results will be presented at three sub-chapters. These sub-chapters are named after the weight sets. The experiments were done under three different weight sets. These weights are composes of three parameters which are cost-value, the number of alternatives at best class, and the number of alternatives at worst class. The trio of 50%-40%-10%, the equal set and 70%-20%-10% are hereafter referred to as “Weight Set 1", “Weight Set 2" and “Weight Set 3", respectively.

The number of alternatives, criterion and action based of outcomes are shown at the following tables. Having 10 number of instances for each set of experiments are mentioned before. It could be seen how many times we turn to the decision maker also. Asking minimum number of questions would be appreciated. In anyhow, since we don’t imply pure pairwise comparison within all alternatives, it is expected the satisfying number of questions that we have.

The detail results are shown for the 3 criteria case in Table 6, Table 7 and Table 8 for the first weight set, second weight set and third weight set respectively. These tables include the number of questions asked to the decision maker for each instance. Also, it can be seen the scores of the developed algorithm and the scores of decision maker i.e. algorithm with known set of weights. Lastly, the percentage values of the differences, the gaps, are shown.

The first result that we are going to examine covers the weights which are 50%, 40% and 10%. 3 different size of alternatives, actions and criteria were computed. In general, these results show that in average, 6.7 questions are asked. It takes 740 seconds in average to solve a problem instance. In average, 3.33% of gap from the exact solution occurs while considering the weight set is known.

Table 6: The Case of 3 Classes Experimental Results for Set Weight No.1

OBJ	CRI	CLASS	ACT	INS	QUEST.	CPU	ALG	DM	DIF	GAP %
50	3	3	10	1	3	52	0.597	0.717	0.12	16.72
50	3	3	10	2	3	64	0.524	0.692	0.17	24.28
50	3	3	10	3	7	261	0.814	0.814	0.00	0.00
50	3	3	10	4	2	48	0.608	0.730	0.12	16.66
50	3	3	10	5	10	245	0.689	0.743	0.05	7.23
50	3	3	10	6	7	179	0.804	0.804	0.00	0.06
50	3	3	10	7	3	53	0.666	0.780	0.11	14.62
50	3	3	10	8	7	1,949	0.760	0.760	0.00	0.03
50	3	3	10	9	7	191	0.762	0.763	0.00	0.11
50	3	3	10	10	7	186	0.646	0.653	0.01	1.16

50	3	3	30	1	3	50	0.511	0.722	0.21	29.19
50	3	3	30	2	7	229	0.627	0.637	0.01	1.62
50	3	3	30	3	7	213	0.575	0.588	0.01	2.31
50	3	3	30	4	7	172	0.777	0.777	0.00	0.00
50	3	3	30	5	3	52	0.721	0.825	0.10	12.63
50	3	3	30	6	7	225	0.731	0.731	0.00	0.00
50	3	3	30	7	7	248	0.656	0.658	0.00	0.34
50	3	3	30	8	7	235	0.651	0.655	0.00	0.68
50	3	3	30	9	7	187	0.609	0.611	0.00	0.40
50	3	3	30	10	7	767	0.789	0.789	0.00	0.00
50	3	3	50	1	7	196	0.765	0.768	0.00	0.34
50	3	3	50	2	7	181	0.711	0.713	0.00	0.29
50	3	3	50	3	3	79	0.577	0.734	0.16	21.35
50	3	3	50	4	7	434	0.691	0.691	0.00	0.00
50	3	3	50	5	7	198	0.692	0.694	0.00	0.23
50	3	3	50	6	7	178	0.816	0.816	0.00	0.00
50	3	3	50	7	7	185	0.794	0.794	0.00	0.00
50	3	3	50	8	3	65	0.633	0.757	0.12	16.35
50	3	3	50	9	3	55	0.513	0.649	0.14	21.02
50	3	3	50	10	7	1,401	0.689	0.693	0.00	0.59
50	5	3	10	1	7	128	0.726	0.727	0.00	0.05
50	5	3	10	2	9	198	0.643	0.657	0.01	2.18
50	5	3	10	3	11	919	0.652	0.652	0.00	0.00
50	5	3	10	4	7	722	0.837	0.837	0.00	0.00
50	5	3	10	5	7	159	0.674	0.675	0.00	0.23
50	5	3	10	6	2	36	0.709	0.820	0.11	13.47
50	5	3	10	7	7	142	0.635	0.635	0.00	0.08
50	5	3	10	8	7	118	0.858	0.858	0.00	0.00
50	5	3	10	9	7	127	0.750	0.750	0.00	0.00
50	5	3	10	10	7	1,003	0.788	0.788	0.00	0.00
50	5	3	30	1	7	691	0.837	0.837	0.00	0.00
50	5	3	30	2	7	143	0.767	0.772	0.01	0.69
50	5	3	30	3	7	263	0.719	0.720	0.00	0.24
50	5	3	30	4	7	1,797	0.724	0.724	0.00	0.00
50	5	3	30	5	7	771	0.814	0.814	0.00	0.00
50	5	3	30	6	7	823	0.830	0.830	0.00	0.02
50	5	3	30	7	7	1,836	0.774	0.774	0.00	0.00

50	5	3	30	8	7	167	0.842	0.889	0.05	5.26
50	5	3	30	9	3	51	0.542	0.671	0.13	19.18
50	5	3	30	10	7	749	0.876	0.876	0.00	0.00
50	5	3	50	1	3	266	0.547	0.719	0.17	23.98
50	5	3	50	2	11	368	0.612	0.613	0.00	0.04
50	5	3	50	3	7	309	0.710	0.712	0.00	0.33
50	5	3	50	4	7	229	0.835	0.835	0.00	0.00
50	5	3	50	5	7	2,010	0.780	0.845	0.06	7.64
50	5	3	50	6	7	250	0.736	0.737	0.00	0.13
50	5	3	50	7	7	2,022	0.735	0.735	0.00	0.00
50	5	3	50	8	7	227	0.658	0.661	0.00	0.46
50	5	3	50	9	3	59	0.538	0.651	0.11	17.28
50	5	3	50	10	7	205	0.670	0.670	0.00	0.00
50	7	3	10	1	9	148	0.683	0.685	0.00	0.28
50	7	3	10	2	12	253	0.612	0.616	0.00	0.67
50	7	3	10	3	3	77	0.636	0.783	0.15	18.86
50	7	3	10	4	3	80	0.736	0.838	0.10	12.15
50	7	3	10	5	15	677	0.655	0.655	0.00	0.00
50	7	3	10	6	7	511	0.601	0.608	0.01	1.19
50	7	3	10	7	7	176	0.701	0.705	0.00	0.52
50	7	3	10	8	7	419	0.818	0.818	0.00	0.01
50	7	3	10	9	7	150	0.680	0.683	0.00	0.45
50	7	3	10	10	3	60	0.500	0.628	0.13	20.42
50	7	3	30	1	7	108	0.874	0.874	0.00	0.08
50	7	3	30	2	3	60	0.520	0.701	0.18	25.81
50	7	3	30	3	3	68	0.552	0.693	0.14	20.32
50	7	3	30	4	7	348	0.849	0.849	0.00	0.00
50	7	3	30	5	7	736	0.751	0.751	0.00	0.06
50	7	3	30	6	7	2,529	0.806	0.808	0.00	0.29
50	7	3	30	7	7	142	0.879	0.879	0.00	0.00
50	7	3	30	8	7	1,346	0.761	0.762	0.00	0.19
50	7	3	30	9	7	2,353	0.817	0.817	0.00	0.00
50	7	3	30	10	7	196	0.733	0.734	0.00	0.04
50	7	3	50	1	7	986	0.739	0.746	0.01	1.04
50	7	3	50	2	7	146	0.684	0.684	0.00	0.05
50	7	3	50	3	7	200	0.728	0.731	0.00	0.47
50	7	3	50	4	7	1,421	0.664	0.667	0.00	0.44

50	7	3	50	5	7	150	0.604	0.608	0.00	0.80
50	7	3	50	6	3	107	0.673	0.818	0.15	17.76
50	7	3	50	7	7	1,415	0.741	0.742	0.00	0.04
50	7	3	50	8	7	215	0.699	0.700	0.00	0.08
50	7	3	50	9	7	181	0.839	0.839	0.00	0.00
50	7	3	50	10	7	1,334	0.774	0.775	0.00	0.10
100	3	3	10	1	7	164	0.880	0.880	0.00	0.00
100	3	3	10	2	7	172	0.854	0.854	0.00	0.00
100	3	3	10	3	7	179	0.769	0.769	0.00	0.00
100	3	3	10	4	7	207	0.727	0.728	0.00	0.11
100	3	3	10	5	7	1,398	0.734	0.737	0.00	0.37
100	3	3	10	6	3	78	0.575	0.731	0.16	21.34
100	3	3	10	7	3	54	0.601	0.706	0.11	14.91
100	3	3	10	8	7	1,988	0.764	0.765	0.00	0.10
100	3	3	10	9	7	249	0.618	0.626	0.01	1.16
100	3	3	10	10	7	1,516	0.612	0.616	0.00	0.79
100	3	3	30	1	7	179	0.835	0.835	0.00	0.00
100	3	3	30	2	7	179	0.678	0.681	0.00	0.54
100	3	3	30	3	7	519	0.835	0.837	0.00	0.32
100	3	3	30	4	11	774	0.712	0.712	0.00	0.00
100	3	3	30	5	7	1,120	0.785	0.786	0.00	0.08
100	3	3	30	6	7	1,978	0.879	0.879	0.00	0.00
100	3	3	30	7	7	2,194	0.775	0.775	0.00	0.00
100	3	3	30	8	7	831	0.825	0.825	0.00	0.00
100	3	3	30	9	7	244	0.847	0.848	0.00	0.13
100	3	3	30	10	7	1,499	0.626	0.633	0.01	1.08
100	3	3	50	1	3	93	0.641	0.797	0.16	19.53
100	3	3	50	2	9	248	0.789	0.798	0.01	1.09
100	3	3	50	3	3	41	0.957	0.957	0.00	0.00
100	3	3	50	4	11	2,734	0.638	0.650	0.01	1.88
100	3	3	50	5	3	82	0.559	0.691	0.13	19.05
100	3	3	50	6	7	570	0.696	0.697	0.00	0.08
100	3	3	50	7	7	226	0.761	0.762	0.00	0.09
100	3	3	50	8	3	483	0.605	0.774	0.17	21.80
100	3	3	50	9	7	158	0.748	0.752	0.00	0.49
100	3	3	50	10	7	187	0.741	0.741	0.00	0.00
100	5	3	10	1	7	1,149	0.778	0.778	0.00	0.00

100	5	3	10	2	3	46	0.628	0.731	0.10	14.07
100	5	3	10	3	7	186	0.661	0.663	0.00	0.40
100	5	3	10	4	7	765	0.869	0.870	0.00	0.05
100	5	3	10	5	7	394	0.769	0.802	0.03	4.10
100	5	3	10	6	3	41	0.693	0.864	0.17	19.80
100	5	3	10	7	7	156	0.818	0.818	0.00	0.06
100	5	3	10	8	19	4,652	0.695	0.696	0.00	0.14
100	5	3	10	9	7	166	0.885	0.887	0.00	0.19
100	5	3	10	10	7	290	0.781	0.782	0.00	0.12
100	5	3	30	1	7	1,424	0.727	0.727	0.00	0.00
100	5	3	30	2	7	155	0.894	0.894	0.00	0.00
100	5	3	30	3	7	1,968	0.727	0.727	0.00	0.03
100	5	3	30	4	7	183	0.843	0.843	0.00	0.04
100	5	3	30	5	7	162	0.854	0.854	0.00	0.00
100	5	3	30	6	7	157	0.846	0.846	0.00	0.00
100	5	3	30	7	7	780	0.828	0.828	0.00	0.00
100	5	3	30	8	7	2,042	0.759	0.759	0.00	0.00
100	5	3	30	9	7	262	0.719	0.724	0.00	0.58
100	5	3	30	10	7	195	0.797	0.797	0.00	0.00
100	5	3	50	1	7	367	0.667	0.671	0.00	0.56
100	5	3	50	2	7	944	0.843	0.843	0.00	0.02
100	5	3	50	3	7	1,994	0.802	0.802	0.00	0.00
100	5	3	50	4	3	703	0.555	0.717	0.16	22.61
100	5	3	50	5	3	716	0.603	0.752	0.15	19.85
100	5	3	50	6	7	193	0.821	0.821	0.00	0.05
100	5	3	50	7	7	210	0.716	0.720	0.00	0.60
100	5	3	50	8	7	212	0.753	0.753	0.00	0.00
100	5	3	50	9	7	1,678	0.718	0.718	0.00	0.04
100	5	3	50	10	7	328	0.823	0.831	0.01	0.93
100	7	3	10	1	12	403	0.602	0.602	0.00	0.06
100	7	3	10	2	7	221	0.890	0.890	0.00	0.00
100	7	3	10	3	7	154	0.662	0.663	0.00	0.22
100	7	3	10	4	2	53	0.641	0.781	0.14	17.97
100	7	3	10	5	9	210	0.582	0.582	0.00	0.04
100	7	3	10	6	7	755	0.857	0.857	0.00	0.00
100	7	3	10	7	7	1,409	0.625	0.627	0.00	0.37
100	7	3	10	8	7	176	0.728	0.730	0.00	0.26

100	7	3	10	9	7	763	0.830	0.830	0.00	0.00
100	7	3	10	10	13	2,760	0.621	0.630	0.01	1.41
100	7	3	30	1	7	316	0.796	0.798	0.00	0.29
100	7	3	30	2	7	145	0.836	0.836	0.00	0.00
100	7	3	30	3	7	211	0.857	0.860	0.00	0.32
100	7	3	30	4	7	1,398	0.789	0.789	0.00	0.01
100	7	3	30	5	7	1,441	0.839	0.841	0.00	0.17
100	7	3	30	6	7	301	0.807	0.808	0.00	0.12
100	7	3	30	7	7	1,988	0.691	0.692	0.00	0.23
100	7	3	30	8	7	1,049	0.815	0.817	0.00	0.26
100	7	3	30	9	3	76	0.607	0.755	0.15	19.61
100	7	3	30	10	7	1,440	0.815	0.820	0.01	0.66
100	7	3	50	1	7	2,033	0.815	0.815	0.00	0.08
100	7	3	50	2	7	1,500	0.724	0.724	0.00	0.12
100	7	3	50	3	7	1,629	0.644	0.650	0.01	0.92
100	7	3	50	4	7	857	0.827	0.827	0.00	0.00
100	7	3	50	5	7	2,046	0.785	0.785	0.00	0.00
100	7	3	50	6	7	2,082	0.841	0.842	0.00	0.11
100	7	3	50	7	7	1,107	0.683	0.686	0.00	0.32
100	7	3	50	8	7	1,553	0.725	0.726	0.00	0.15
100	7	3	50	9	7	2,126	0.854	0.854	0.00	0.00
100	7	3	50	10	7	999	0.684	0.690	0.01	0.81
200	3	3	10	1	7	134	0.757	0.757	0.00	0.05
200	3	3	10	2	7	726	0.817	0.818	0.00	0.08
200	3	3	10	3	3	122	0.606	0.749	0.14	19.05
200	3	3	10	4	8	1,516	0.725	0.727	0.00	0.27
200	3	3	10	5	7	733	0.712	0.713	0.00	0.07
200	3	3	10	6	3	102	0.571	0.729	0.16	21.55
200	3	3	10	7	7	151	0.744	0.745	0.00	0.13
200	3	3	10	8	7	754	0.603	0.604	0.00	0.09
200	3	3	10	9	7	136	0.646	0.646	0.00	0.05
200	3	3	10	10	3	42	0.553	0.715	0.16	22.65
200	3	3	30	1	9	1,008	0.634	0.635	0.00	0.09
200	3	3	30	2	7	949	0.639	0.643	0.00	0.58
200	3	3	30	3	3	88	0.553	0.734	0.18	24.60
200	3	3	30	4	7	811	0.703	0.704	0.00	0.10
200	3	3	30	5	7	155	0.854	0.854	0.00	0.00

200	3	3	30	6	7	392	0.896	0.896	0.00	0.00
200	3	3	30	7	7	781	0.825	0.825	0.00	0.00
200	3	3	30	8	7	778	0.638	0.648	0.01	1.62
200	3	3	30	9	7	203	0.594	0.597	0.00	0.63
200	3	3	30	10	7	174	0.853	0.857	0.00	0.46
200	3	3	50	1	7	857	0.775	0.776	0.00	0.13
200	3	3	50	2	7	1,521	0.716	0.717	0.00	0.04
200	3	3	50	3	7	1,749	0.647	0.652	0.01	0.82
200	3	3	50	4	7	284	0.697	0.698	0.00	0.15
200	3	3	50	5	9	1,052	0.624	0.626	0.00	0.25
200	3	3	50	6	7	298	0.617	0.631	0.01	2.16
200	3	3	50	7	7	219	0.751	0.752	0.00	0.14
200	3	3	50	8	7	240	0.781	0.781	0.00	0.00
200	3	3	50	9	7	1,429	0.798	0.798	0.00	0.00
200	3	3	50	10	7	761	0.886	0.886	0.00	0.00
200	5	3	10	1	15	1,106	0.681	0.685	0.00	0.46
200	5	3	10	2	7	743	0.867	0.867	0.00	0.00
200	5	3	10	3	7	737	0.784	0.784	0.00	0.01
200	5	3	10	4	7	169	0.821	0.821	0.00	0.00
200	5	3	10	5	9	189	0.677	0.686	0.01	1.19
200	5	3	10	6	2	75	0.624	0.753	0.13	17.17
200	5	3	10	7	7	141	0.728	0.728	0.00	0.00
200	5	3	10	8	2	53	0.773	0.860	0.09	10.05
200	5	3	10	9	7	149	0.775	0.775	0.00	0.00
200	5	3	10	10	3	119	0.660	0.803	0.14	17.75
200	5	3	30	1	7	278	0.746	0.749	0.00	0.46
200	5	3	30	2	3	671	0.726	0.730	0.00	0.56
200	5	3	30	3	7	288	0.763	0.764	0.00	0.10
200	5	3	30	4	7	774	0.788	0.788	0.00	0.05
200	5	3	30	5	7	1,404	0.769	0.769	0.00	0.00
200	5	3	30	6	3	86	0.551	0.708	0.16	22.17
200	5	3	30	7	7	335	0.745	0.748	0.00	0.48
200	5	3	30	8	7	199	0.797	0.797	0.00	0.00
200	5	3	30	9	7	225	0.793	0.793	0.00	0.00
200	5	3	30	10	3	668	0.722	0.727	0.00	0.67
200	5	3	50	1	7	1,436	0.836	0.836	0.00	0.00
200	5	3	50	2	7	296	0.863	0.863	0.00	0.00

200	5	3	50	3	7	969	0.839	0.839	0.00	0.00
200	5	3	50	4	7	3,501	0.796	0.796	0.00	0.00
200	5	3	50	5	7	1,111	0.751	0.752	0.00	0.13
200	5	3	50	6	7	278	0.836	0.836	0.00	0.08
200	5	3	50	7	7	14,821	0.738	0.740	0.00	0.27
200	5	3	50	8	9	1,189	0.749	0.752	0.00	0.41
200	5	3	50	9	3	216	0.548	0.727	0.18	24.66
200	5	3	50	10	4	250	0.725	0.725	0.00	0.00
200	7	3	10	1	2	69	0.550	0.742	0.19	25.90
200	7	3	10	2	9	1,987	0.672	0.676	0.00	0.60
200	7	3	10	3	7	150	0.851	0.851	0.00	0.00
200	7	3	10	4	12	522	0.589	0.597	0.01	1.35
200	7	3	10	5	7	778	0.681	0.685	0.00	0.60
200	7	3	10	6	7	196	0.826	0.827	0.00	0.09
200	7	3	10	7	17	1,690	0.599	0.601	0.00	0.30
200	7	3	10	8	3	104	0.606	0.615	0.01	1.42
200	7	3	10	9	7	155	0.691	0.691	0.00	0.02
200	7	3	10	10	3	1,242	0.649	0.777	0.13	16.42
200	7	3	30	1	7	1,208	0.760	0.761	0.00	0.07
200	7	3	30	2	9	1,562	0.655	0.663	0.01	1.30
200	7	3	30	3	7	595	0.812	0.813	0.00	0.20
200	7	3	30	4	7	1,483	0.796	0.796	0.00	0.06
200	7	3	30	5	7	1,551	0.815	0.815	0.00	0.00
200	7	3	30	6	7	1,490	0.793	0.793	0.00	0.03
200	7	3	30	7	3	132	0.590	0.725	0.14	18.65
200	7	3	30	8	7	360	0.788	0.788	0.00	0.07
200	7	3	30	9	13	1,997	0.640	0.654	0.01	2.12
200	7	3	30	10	7	843	0.843	0.843	0.00	0.10
200	7	3	50	1	9	2,223	0.684	0.691	0.01	1.01
200	7	3	50	2	7	554	0.715	0.717	0.00	0.23
200	7	3	50	3	7	253	0.791	0.791	0.00	0.00
200	7	3	50	4	7	5,197	0.782	0.783	0.00	0.06
200	7	3	50	5	7	1,110	0.663	0.669	0.01	0.91
200	7	3	50	6	7	1,566	0.816	0.817	0.00	0.16
200	7	3	50	7	7	1,565	0.846	0.846	0.00	0.00
200	7	3	50	8	7	977	0.822	0.822	0.00	0.00
200	7	3	50	9	7	960	0.808	0.808	0.00	0.00

200	7	3	50	10	3	664	0.675	0.677	0.00	0.38

The experimental study results of the second weight set are detailed at the Table 7. It is shown that the gap values of almost all the instance are zero. This value is expected of course, since the algorithm is initiated by following the middlemost point. When examining the average values, the average number of questions asked to the decision maker is 3, the average solution time of one problem is 344 and the average gap value is 0.03%.

Table 7: The Case of 3 Classes Experimental Results for Set Weight No.2

OBJ	CRI	CLASS	ACT	INS	QUEST.	CPU	ALG	DM	DIF	GAP %
50	3	3	10	1	3	44	0.696	0.696	0.00	0.00
50	3	3	10	2	3	52	0.670	0.670	0.00	0.00
50	3	3	10	3	3	53	0.722	0.722	0.00	0.00
50	3	3	10	4	2	48	0.691	0.691	0.00	0.00
50	3	3	10	5	3	53	0.686	0.686	0.00	0.00
50	3	3	10	6	3	45	0.752	0.752	0.00	0.00
50	3	3	10	7	3	52	0.694	0.694	0.00	0.00
50	3	3	10	8	3	648	0.678	0.678	0.00	0.00
50	3	3	10	9	3	38	0.723	0.723	0.00	0.00
50	3	3	10	10	3	49	0.701	0.701	0.00	0.00
50	3	3	30	1	3	52	0.667	0.667	0.00	0.00
50	3	3	30	2	3	53	0.667	0.667	0.00	0.00
50	3	3	30	3	3	49	0.670	0.670	0.00	0.00
50	3	3	30	4	3	67	0.705	0.705	0.00	0.00
50	3	3	30	5	3	53	0.710	0.710	0.00	0.00
50	3	3	30	6	3	68	0.712	0.712	0.00	0.00
50	3	3	30	7	3	63	0.667	0.667	0.00	0.00
50	3	3	30	8	3	72	0.670	0.670	0.00	0.00
50	3	3	30	9	3	42	0.670	0.670	0.00	0.00
50	3	3	30	10	3	648	0.679	0.679	0.00	0.00
50	3	3	50	1	3	62	0.681	0.681	0.00	0.00
50	3	3	50	2	3	56	0.676	0.676	0.00	0.00
50	3	3	50	3	3	70	0.674	0.674	0.00	0.00
50	3	3	50	4	3	49	0.670	0.670	0.00	0.00
50	3	3	50	5	3	47	0.674	0.674	0.00	0.00
50	3	3	50	6	3	54	0.689	0.689	0.00	0.00
50	3	3	50	7	3	59	0.674	0.674	0.00	0.00
50	3	3	50	8	3	60	0.674	0.674	0.00	0.00

50	3	3	50	9	3	55	0.669	0.669	0.00	0.00
50	3	3	50	10	3	56	0.668	0.668	0.00	0.00
50	5	3	10	1	3	47	0.680	0.680	0.00	0.00
50	5	3	10	2	3	43	0.671	0.671	0.00	0.00
50	5	3	10	3	3	42	0.669	0.669	0.00	0.00
50	5	3	10	4	3	640	0.721	0.721	0.00	0.00
50	5	3	10	5	3	41	0.678	0.678	0.00	0.00
50	5	3	10	6	2	40	0.728	0.728	0.00	0.00
50	5	3	10	7	3	36	0.746	0.746	0.00	0.00
50	5	3	10	8	3	41	0.722	0.722	0.00	0.00
50	5	3	10	9	3	35	0.715	0.715	0.00	0.00
50	5	3	10	10	3	649	0.682	0.682	0.00	0.00
50	5	3	30	1	3	652	0.703	0.703	0.00	0.00
50	5	3	30	2	3	48	0.704	0.704	0.00	0.00
50	5	3	30	3	3	160	0.681	0.681	0.00	0.00
50	5	3	30	4	3	512	0.674	0.674	0.00	0.00
50	5	3	30	5	3	646	0.703	0.703	0.00	0.00
50	5	3	30	6	3	657	0.683	0.683	0.00	0.00
50	5	3	30	7	3	394	0.701	0.701	0.00	0.00
50	5	3	30	8	3	46	0.718	0.718	0.00	0.00
50	5	3	30	9	3	49	0.670	0.670	0.00	0.00
50	5	3	30	10	3	654	0.716	0.716	0.00	0.00
50	5	3	50	1	3	243	0.671	0.671	0.00	0.00
50	5	3	50	2	3	49	0.686	0.686	0.00	0.00
50	5	3	50	3	3	70	0.667	0.667	0.00	0.00
50	5	3	50	4	3	88	0.709	0.709	0.00	0.00
50	5	3	50	5	3	671	0.681	0.681	0.00	0.00
50	5	3	50	6	3	54	0.681	0.681	0.00	0.00
50	5	3	50	7	3	623	0.674	0.674	0.00	0.00
50	5	3	50	8	3	56	0.674	0.674	0.00	0.00
50	5	3	50	9	3	54	0.674	0.674	0.00	0.00
50	5	3	50	10	3	43	0.668	0.668	0.00	0.00
50	7	3	10	1	3	41	0.672	0.672	0.00	0.00
50	7	3	10	2	3	41	0.681	0.681	0.00	0.00
50	7	3	10	3	3	47	0.732	0.732	0.00	0.00
50	7	3	10	4	3	45	0.740	0.740	0.00	0.00
50	7	3	10	5	3	39	0.671	0.671	0.00	0.00

50	7	3	10	6	3	198	0.683	0.683	0.00	0.00
50	7	3	10	7	3	41	0.669	0.669	0.00	0.00
50	7	3	10	8	3	230	0.698	0.698	0.00	0.00
50	7	3	10	9	3	39	0.696	0.696	0.00	0.00
50	7	3	10	10	3	37	0.667	0.667	0.00	0.00
50	7	3	30	1	3	51	0.708	0.708	0.00	0.00
50	7	3	30	2	3	59	0.668	0.668	0.00	0.00
50	7	3	30	3	3	61	0.680	0.680	0.00	0.00
50	7	3	30	4	3	199	0.714	0.714	0.00	0.00
50	7	3	30	5	3	60	0.678	0.678	0.00	0.00
50	7	3	30	6	3	658	0.689	0.689	0.00	0.00
50	7	3	30	7	3	57	0.745	0.745	0.00	0.00
50	7	3	30	8	3	659	0.683	0.683	0.00	0.00
50	7	3	30	9	3	648	0.693	0.693	0.00	0.00
50	7	3	30	10	3	101	0.668	0.668	0.00	0.00
50	7	3	50	1	3	376	0.720	0.720	0.00	0.00
50	7	3	50	2	3	48	0.679	0.679	0.00	0.00
50	7	3	50	3	3	80	0.692	0.692	0.00	0.00
50	7	3	50	4	3	103	0.679	0.679	0.00	0.00
50	7	3	50	5	3	45	0.674	0.674	0.00	0.00
50	7	3	50	6	3	89	0.712	0.712	0.00	0.00
50	7	3	50	7	3	675	0.674	0.674	0.00	0.00
50	7	3	50	8	3	76	0.682	0.682	0.00	0.00
50	7	3	50	9	3	114	0.712	0.712	0.00	0.00
50	7	3	50	10	3	663	0.681	0.681	0.00	0.00
100	3	3	10	1	3	51	0.730	0.730	0.00	0.00
100	3	3	10	2	3	55	0.714	0.714	0.00	0.00
100	3	3	10	3	3	53	0.704	0.704	0.00	0.00
100	3	3	10	4	3	60	0.693	0.693	0.00	0.00
100	3	3	10	5	3	74	0.688	0.688	0.00	0.00
100	3	3	10	6	3	69	0.692	0.692	0.00	0.00
100	3	3	10	7	3	49	0.691	0.691	0.00	0.00
100	3	3	10	8	3	654	0.699	0.699	0.00	0.00
100	3	3	10	9	3	50	0.695	0.695	0.00	0.00
100	3	3	10	10	3	71	0.672	0.672	0.00	0.00
100	3	3	30	1	3	73	0.749	0.749	0.00	0.00
100	3	3	30	2	3	81	0.690	0.690	0.00	0.00

100	3	3	30	3	3	64	0.775	0.775	0.00	0.00
100	3	3	30	4	3	45	0.787	0.787	0.00	0.00
100	3	3	30	5	3	673	0.710	0.710	0.00	0.00
100	3	3	30	6	3	69	0.760	0.760	0.00	0.00
100	3	3	30	7	3	669	0.690	0.690	0.00	0.00
100	3	3	30	8	3	673	0.748	0.748	0.00	0.00
100	3	3	30	9	3	49	0.799	0.799	0.00	0.00
100	3	3	30	10	3	98	0.703	0.703	0.00	0.00
100	3	3	50	1	3	108	0.695	0.695	0.00	0.00
100	3	3	50	2	3	55	0.855	0.855	0.00	0.00
100	3	3	50	3	3	44	0.972	0.972	0.00	0.00
100	3	3	50	4	3	99	0.672	0.672	0.00	0.00
100	3	3	50	5	3	77	0.677	0.677	0.00	0.00
100	3	3	50	6	3	96	0.682	0.682	0.00	0.00
100	3	3	50	7	3	68	0.772	0.772	0.00	0.00
100	3	3	50	8	3	507	0.687	0.687	0.00	0.00
100	3	3	50	9	3	66	0.701	0.701	0.00	0.00
100	3	3	50	10	3	49	0.689	0.689	0.00	0.00
100	5	3	10	1	3	652	0.683	0.683	0.00	0.00
100	5	3	10	2	3	44	0.707	0.707	0.00	0.00
100	5	3	10	3	3	43	0.677	0.677	0.00	0.00
100	5	3	10	4	3	659	0.733	0.733	0.00	0.00
100	5	3	10	5	3	243	0.749	0.749	0.00	0.00
100	5	3	10	6	3	43	0.757	0.757	0.00	0.00
100	5	3	10	7	3	46	0.746	0.746	0.00	0.00
100	5	3	10	8	3	526	0.693	0.693	0.00	0.00
100	5	3	10	9	3	43	0.726	0.726	0.00	0.00
100	5	3	10	10	3	57	0.705	0.705	0.00	0.00
100	5	3	30	1	3	87	0.680	0.680	0.00	0.00
100	5	3	30	2	3	90	0.765	0.765	0.00	0.00
100	5	3	30	3	3	708	0.681	0.681	0.00	0.00
100	5	3	30	4	3	85	0.758	0.758	0.00	0.00
100	5	3	30	5	3	80	0.744	0.744	0.00	0.00
100	5	3	30	6	3	70	0.782	0.782	0.00	0.00
100	5	3	30	7	3	667	0.700	0.700	0.00	0.00
100	5	3	30	8	3	751	0.695	0.695	0.00	0.00
100	5	3	30	9	3	89	0.759	0.759	0.00	0.00

100	5	3	30	10	3	79	0.713	0.713	0.00	0.00
100	5	3	50	1	3	157	0.670	0.670	0.00	0.00
100	5	3	50	2	3	723	0.697	0.697	0.00	0.00
100	5	3	50	3	3	697	0.695	0.695	0.00	0.00
100	5	3	50	4	3	702	0.670	0.670	0.00	0.00
100	5	3	50	5	3	719	0.679	0.679	0.00	0.00
100	5	3	50	6	3	82	0.704	0.704	0.00	0.00
100	5	3	50	7	3	86	0.679	0.679	0.00	0.00
100	5	3	50	8	3	89	0.669	0.669	0.00	0.00
100	5	3	50	9	3	160	0.689	0.689	0.00	0.00
100	5	3	50	10	3	62	0.850	0.850	0.00	0.00
100	7	3	10	1	3	49	0.693	0.693	0.00	0.00
100	7	3	10	2	3	116	0.747	0.747	0.00	0.00
100	7	3	10	3	3	41	0.676	0.676	0.00	0.00
100	7	3	10	4	2	48	0.714	0.714	0.00	0.00
100	7	3	10	5	3	41	0.684	0.684	0.00	0.00
100	7	3	10	6	3	652	0.710	0.710	0.00	0.00
100	7	3	10	7	3	53	0.676	0.676	0.00	0.00
100	7	3	10	8	3	64	0.720	0.720	0.00	0.00
100	7	3	10	9	3	648	0.707	0.707	0.00	0.00
100	7	3	10	10	3	36	0.679	0.679	0.00	0.00
100	7	3	30	1	3	82	0.805	0.805	0.00	0.00
100	7	3	30	2	3	52	0.712	0.712	0.00	0.00
100	7	3	30	3	3	101	0.836	0.836	0.00	0.00
100	7	3	30	4	3	61	0.739	0.739	0.00	0.00
100	7	3	30	5	3	85	0.838	0.838	0.00	0.00
100	7	3	30	6	3	96	0.805	0.805	0.00	0.00
100	7	3	30	7	3	676	0.683	0.683	0.00	0.00
100	7	3	30	8	3	127	0.795	0.795	0.00	0.00
100	7	3	30	9	3	71	0.722	0.722	0.00	0.00
100	7	3	30	10	3	89	0.841	0.841	0.00	0.00
100	7	3	50	1	3	702	0.709	0.709	0.00	0.00
100	7	3	50	2	3	111	0.700	0.700	0.00	0.00
100	7	3	50	3	3	126	0.687	0.687	0.00	0.00
100	7	3	50	4	3	159	0.738	0.738	0.00	0.00
100	7	3	50	5	3	697	0.699	0.699	0.00	0.00
100	7	3	50	6	3	718	0.719	0.719	0.00	0.00

100	7	3	50	7	3	105	0.705	0.705	0.00	0.00
100	7	3	50	8	3	55	0.697	0.697	0.00	0.00
100	7	3	50	9	5	756	0.712	0.712	0.00	0.00
100	7	3	50	10	3	728	0.701	0.701	0.00	0.00
200	3	3	10	1	3	40	0.740	0.740	0.00	0.00
200	3	3	10	2	3	48	0.690	0.690	0.00	0.00
200	3	3	10	3	3	98	0.696	0.696	0.00	0.00
200	3	3	10	4	3	658	0.714	0.714	0.00	0.00
200	3	3	10	5	3	665	0.677	0.677	0.00	0.00
200	3	3	10	6	3	104	0.707	0.707	0.00	0.00
200	3	3	10	7	3	53	0.706	0.706	0.00	0.00
200	3	3	10	8	3	56	0.673	0.673	0.00	0.00
200	3	3	10	9	3	70	0.675	0.675	0.00	0.00
200	3	3	10	10	3	46	0.686	0.686	0.00	0.00
200	3	3	30	1	3	147	0.690	0.690	0.00	0.00
200	3	3	30	2	3	125	0.696	0.696	0.00	0.00
200	3	3	30	3	3	94	0.687	0.687	0.00	0.00
200	3	3	30	4	3	81	0.684	0.684	0.00	0.00
200	3	3	30	5	3	78	0.709	0.709	0.00	0.00
200	3	3	30	6	3	84	0.743	0.743	0.00	0.00
200	3	3	30	7	3	698	0.696	0.696	0.00	0.00
200	3	3	30	8	3	72	0.706	0.706	0.00	0.00
200	3	3	30	9	3	63	0.684	0.684	0.00	0.00
200	3	3	30	10	3	107	0.748	0.748	0.00	0.00
200	3	3	50	1	3	103	0.681	0.681	0.00	0.00
200	3	3	50	2	3	105	0.704	0.704	0.00	0.00
200	3	3	50	3	3	402	0.696	0.696	0.00	0.00
200	3	3	50	4	3	160	0.698	0.698	0.00	0.00
200	3	3	50	5	3	181	0.671	0.671	0.00	0.00
200	3	3	50	6	3	73	0.682	0.682	0.00	0.00
200	3	3	50	7	3	64	0.830	0.830	0.00	0.00
200	3	3	50	8	3	112	0.692	0.692	0.00	0.00
200	3	3	50	9	3	717	0.706	0.706	0.00	0.00
200	3	3	50	10	3	682	0.735	0.735	0.00	0.00
200	5	3	10	1	3	68	0.712	0.712	0.00	0.00
200	5	3	10	2	3	664	0.730	0.730	0.00	0.00
200	5	3	10	3	3	661	0.696	0.696	0.00	0.00

200	5	3	10	4	3	77	0.700	0.700	0.00	0.00
200	5	3	10	5	3	56	0.672	0.672	0.00	0.00
200	5	3	10	6	2	72	0.732	0.732	0.00	0.00
200	5	3	10	7	3	42	0.686	0.686	0.00	0.00
200	5	3	10	8	2	52	0.713	0.713	0.00	0.00
200	5	3	10	9	3	77	0.695	0.695	0.00	0.00
200	5	3	10	10	3	118	0.717	0.717	0.00	0.00
200	5	3	30	1	3	147	0.736	0.736	0.00	0.00
200	5	3	30	2	5	480	0.669	0.734	0.07	8.85
200	5	3	30	3	3	145	0.693	0.693	0.00	0.00
200	5	3	30	4	3	100	0.700	0.700	0.00	0.00
200	5	3	30	5	3	720	0.698	0.698	0.00	0.00
200	5	3	30	6	3	87	0.685	0.685	0.00	0.00
200	5	3	30	7	3	131	0.722	0.722	0.00	0.00
200	5	3	30	8	3	112	0.705	0.705	0.00	0.00
200	5	3	30	9	3	131	0.708	0.708	0.00	0.00
200	5	3	30	10	5	203	0.727	0.727	0.00	0.00
200	5	3	50	1	3	753	0.702	0.702	0.00	0.00
200	5	3	50	2	3	193	0.734	0.734	0.00	0.00
200	5	3	50	3	3	822	0.722	0.722	0.00	0.00
200	5	3	50	4	3	2,669	0.711	0.711	0.00	0.00
200	5	3	50	5	3	372	0.713	0.713	0.00	0.00
200	5	3	50	6	3	144	0.711	0.711	0.00	0.00
200	5	3	50	7	3	15,811	0.705	0.705	0.00	0.00
200	5	3	50	8	3	151	0.810	0.810	0.00	0.00
200	5	3	50	9	3	304	0.683	0.683	0.00	0.01
200	5	3	50	10		14,211	0.707	0.707	0.00	0.00
200	7	3	10	1	2	66	0.685	0.685	0.00	0.00
200	7	3	10	2	3	656	0.669	0.669	0.00	0.00
200	7	3	10	3	3	69	0.715	0.715	0.00	0.00
200	7	3	10	4	3	51	0.688	0.688	0.00	0.00
200	7	3	10	5	3	653	0.682	0.682	0.00	0.00
200	7	3	10	6	3	85	0.740	0.740	0.00	0.00
200	7	3	10	7	3	87	0.683	0.683	0.00	0.00
200	7	3	10	8	3	81	0.710	0.710	0.00	0.00
200	7	3	10	9	3	54	0.713	0.713	0.00	0.00
200	7	3	10	10	3	56	0.699	0.699	0.00	0.00

200	7	3	30	1	3	741	0.696	0.696	0.00	0.00
200	7	3	30	2	3	123	0.681	0.681	0.00	0.00
200	7	3	30	3	3	290	0.742	0.742	0.00	0.00
200	7	3	30	4	3	741	0.718	0.718	0.00	0.00
200	7	3	30	5	3	823	0.718	0.718	0.00	0.00
200	7	3	30	6	3	801	0.685	0.685	0.00	0.00
200	7	3	30	7	3	124	0.724	0.724	0.00	0.00
200	7	3	30	8	3	204	0.732	0.732	0.00	0.00
200	7	3	30	9	3	257	0.682	0.682	0.00	0.00
200	7	3	30	10	3	683	0.739	0.739	0.00	0.00
200	7	3	50	1	3	779	0.679	0.679	0.00	0.00
200	7	3	50	2	3	162	0.716	0.716	0.00	0.00
200	7	3	50	3	3	165	0.699	0.699	0.00	0.00
200	7	3	50	4	3	185	0.697	0.697	0.00	0.00
200	7	3	50	5	3	170	0.707	0.707	0.00	0.00
200	7	3	50	6	3	828	0.697	0.697	0.00	0.00
200	7	3	50	7	3	814	0.735	0.735	0.00	0.00
200	7	3	50	8	3	393	0.733	0.733	0.00	0.00
200	7	3	50	9	3	334	0.719	0.719	0.00	0.00
200	7	3	50	10	3	213	0.697	0.697	0.00	0.00

The results of the last weight set are listed at the Table 8. The average questions asked to the decision maker is 3.90 with 4.68% average gap value. A problem instance is solved 209 seconds in average.

Table 8: The Case of 3 Classes Experimental Results for Set Weight No.3

OBJ	CRI	CLASS	ACT	INS	QUEST.	CPU	ALG	DM	DIF	GAP %
50	3	3	10	1	3	48	0.700	0.725	0.03	3.45
50	3	3	10	2	3	53	0.700	0.721	0.02	2.88
50	3	3	10	3	3	48	0.700	0.783	0.08	10.63
50	3	3	10	4	2	49	0.700	0.739	0.04	5.28
50	3	3	10	5	3	52	0.700	0.730	0.03	4.16
50	3	3	10	6	3	44	0.700	0.764	0.06	8.37
50	3	3	10	7	3	53	0.700	0.748	0.05	6.47
50	3	3	10	8	3	50	0.700	0.742	0.04	5.70
50	3	3	10	9	11	277	0.713	0.713	0.00	0.00
50	3	3	10	10	3	54	0.700	0.701	0.00	0.19
50	3	3	30	1	3	61	0.700	0.700	0.00	0.00

50	3	3	30	2	3	72	0.700	0.701	0.00	0.20
50	3	3	30	3	3	61	0.700	0.700	0.00	0.00
50	3	3	30	4	3	66	0.700	0.744	0.04	5.94
50	3	3	30	5	3	64	0.700	0.795	0.09	11.90
50	3	3	30	6	3	81	0.700	0.726	0.03	3.58
50	3	3	30	7	3	67	0.700	0.705	0.00	0.64
50	3	3	30	8	3	67	0.700	0.706	0.01	0.80
50	3	3	30	9	3	59	0.700	0.700	0.00	0.00
50	3	3	30	10	3	66	0.700	0.750	0.05	6.67
50	3	3	50	1	3	85	0.700	0.718	0.02	2.54
50	3	3	50	2	3	70	0.700	0.705	0.01	0.73
50	3	3	50	3	3	77	0.700	0.727	0.03	3.72
50	3	3	50	4	3	65	0.700	0.703	0.00	0.44
50	3	3	50	5	3	63	0.700	0.700	0.00	0.00
50	3	3	50	6	3	65	0.700	0.783	0.08	10.60
50	3	3	50	7	3	68	0.700	0.751	0.05	6.78
50	3	3	50	8	3	74	0.700	0.729	0.03	4.04
50	3	3	50	9	5	139	0.701	0.705	0.00	0.62
50	3	3	50	10	3	70	0.700	0.712	0.01	1.68
50	5	3	10	1	9	195	0.715	0.716	0.00	0.13
50	5	3	10	2	3	45	0.700	0.702	0.00	0.36
50	5	3	10	3	3	43	0.700	0.710	0.01	1.43
50	5	3	10	4	3	42	0.700	0.812	0.11	13.75
50	5	3	10	5	3	49	0.700	0.714	0.01	1.96
50	5	3	10	6	2	46	0.700	0.788	0.09	11.22
50	5	3	10	7	3	39	0.700	0.700	0.00	0.00
50	5	3	10	8	3	41	0.700	0.842	0.14	16.84
50	5	3	10	9	3	39	0.700	0.728	0.03	3.79
50	5	3	10	10	7	273	0.743	0.755	0.01	1.61
50	5	3	30	1	7	256	0.811	0.811	0.00	0.00
50	5	3	30	2	3	59	0.700	0.736	0.04	4.94
50	5	3	30	3	3	64	0.700	0.724	0.02	3.36
50	5	3	30	4	3	65	0.700	0.717	0.02	2.40
50	5	3	30	5	3	56	0.700	0.781	0.08	10.33
50	5	3	30	6	7	227	0.802	0.802	0.00	0.00
50	5	3	30	7	3	61	0.700	0.739	0.04	5.29
50	5	3	30	8	3	56	0.900	0.900	0.00	0.00

50	5	3	30	9	3	59	0.700	0.708	0.01	1.11
50	5	3	30	10	3	63	0.700	0.867	0.17	19.22
50	5	3	50	1	3	69	0.700	0.708	0.01	1.19
50	5	3	50	2	3	66	0.700	0.700	0.00	0.00
50	5	3	50	3	3	68	0.700	0.724	0.02	3.27
50	5	3	50	4	3	85	0.700	0.809	0.11	13.43
50	5	3	50	5	3	81	0.700	0.728	0.03	3.87
50	5	3	50	6	3	63	0.700	0.706	0.01	0.79
50	5	3	50	7	3	92	0.700	0.724	0.02	3.30
50	5	3	50	8	3	67	0.700	0.700	0.00	0.05
50	5	3	50	9	3	63	0.700	0.707	0.01	1.03
50	5	3	50	10	3	57	0.700	0.700	0.00	0.00
50	7	3	10	1	3	70	0.700	0.712	0.01	1.71
50	7	3	10	2	3	61	0.700	0.700	0.00	0.02
50	7	3	10	3	3	52	0.700	0.748	0.05	6.43
50	7	3	10	4	3	54	0.700	0.811	0.11	13.70
50	7	3	10	5	3	54	0.700	0.720	0.02	2.83
50	7	3	10	6	13	413	0.706	0.707	0.00	0.14
50	7	3	10	7	3	53	0.700	0.717	0.02	2.35
50	7	3	10	8	3	54	0.700	0.785	0.08	10.80
50	7	3	10	9	3	51	0.700	0.703	0.00	0.39
50	7	3	10	10	3	51	0.700	0.702	0.00	0.26
50	7	3	30	1	7	170	0.862	0.862	0.00	0.00
50	7	3	30	2	3	55	0.700	0.706	0.01	0.84
50	7	3	30	3	3	57	0.700	0.711	0.01	1.61
50	7	3	30	4	7	180	0.829	0.829	0.00	0.00
50	7	3	30	5	7	189	0.708	0.733	0.02	3.38
50	7	3	30	6	3	58	0.700	0.769	0.07	8.98
50	7	3	30	7	7	163	0.871	0.871	0.00	0.00
50	7	3	30	8	7	218	0.752	0.753	0.00	0.10
50	7	3	30	9	3	64	0.700	0.787	0.09	11.03
50	7	3	30	10	9	564	0.719	0.722	0.00	0.49
50	7	3	50	1	3	72	0.700	0.713	0.01	1.76
50	7	3	50	2	3	52	0.700	0.704	0.00	0.60
50	7	3	50	3	3	60	0.700	0.706	0.01	0.88
50	7	3	50	4	3	78	0.700	0.708	0.01	1.17
50	7	3	50	5	3	61	0.700	0.700	0.00	0.00

50	7	3	50	6	3	100	0.700	0.792	0.09	11.63
50	7	3	50	7	9	303	0.736	0.737	0.00	0.11
50	7	3	50	8	9	280	0.717	0.723	0.01	0.88
50	7	3	50	9	7	226	0.814	0.814	0.00	0.00
50	7	3	50	10	3	60	0.700	0.746	0.05	6.20
100	3	3	10	1	7	130	0.871	0.871	0.00	0.00
100	3	3	10	2	3	53	0.700	0.836	0.14	16.25
100	3	3	10	3	3	48	0.700	0.732	0.03	4.34
100	3	3	10	4	3	58	0.700	0.730	0.03	4.14
100	3	3	10	5	3	64	0.700	0.725	0.03	3.50
100	3	3	10	6	3	53	0.700	0.731	0.03	4.30
100	3	3	10	7	3	43	0.700	0.718	0.02	2.44
100	3	3	10	8	3	52	0.700	0.741	0.04	5.52
100	3	3	10	9	3	48	0.700	0.708	0.01	1.16
100	3	3	10	10	13	907	0.709	0.709	0.00	0.02
100	3	3	30	1	3	59	0.700	0.810	0.11	13.54
100	3	3	30	2	3	51	0.700	0.714	0.01	1.98
100	3	3	30	3	3	52	0.700	0.809	0.11	13.45
100	3	3	30	4	3	45	0.700	0.700	0.00	0.00
100	3	3	30	5	3	67	0.700	0.757	0.06	7.53
100	3	3	30	6	3	66	0.700	0.870	0.17	19.58
100	3	3	30	7	3	58	0.700	0.744	0.04	5.93
100	3	3	30	8	3	68	0.700	0.795	0.10	11.98
100	3	3	30	9	3	48	0.700	0.826	0.13	15.27
100	3	3	30	10	3	60	0.700	0.705	0.00	0.71
100	3	3	50	1	3	74	0.700	0.757	0.06	7.55
100	3	3	50	2	3	50	0.700	0.754	0.05	7.14
100	3	3	50	3	3	39	0.940	0.940	0.00	0.00
100	3	3	50	4	3	58	0.700	0.700	0.00	0.00
100	3	3	50	5	3	59	0.700	0.708	0.01	1.13
100	3	3	50	6	3	73	0.700	0.713	0.01	1.88
100	3	3	50	7	3	60	0.700	0.735	0.04	4.78
100	3	3	50	8	3	381	0.700	0.742	0.04	5.63
100	3	3	50	9	3	51	0.700	0.701	0.00	0.08
100	3	3	50	10	3	46	0.700	0.701	0.00	0.16
100	5	3	10	1	3	50	0.700	0.749	0.05	6.60
100	5	3	10	2	3	39	0.700	0.733	0.03	4.44

100	5	3	10	3	7	109	0.700	0.716	0.02	2.17
100	5	3	10	4	3	47	0.700	0.857	0.16	18.31
100	5	3	10	5	3	50	0.700	0.749	0.05	6.54
100	5	3	10	6	7	135	0.849	0.849	0.00	0.00
100	5	3	10	7	3	40	0.700	0.700	0.00	0.00
100	5	3	10	8	9	231	0.721	0.721	0.00	0.00
100	5	3	10	9	3	37	0.700	0.880	0.18	20.46
100	5	3	10	10	3	39	0.700	0.739	0.04	5.32
100	5	3	30	1	3	80	0.700	0.728	0.03	3.85
100	5	3	30	2	3	61	0.700	0.892	0.19	21.50
100	5	3	30	3	3	72	0.700	0.726	0.03	3.64
100	5	3	30	4	3	72	0.700	0.819	0.12	14.55
100	5	3	30	5	3	73	0.700	0.835	0.14	16.20
100	5	3	30	6	3	65	0.700	0.823	0.12	14.99
100	5	3	30	7	3	63	0.700	0.799	0.10	12.37
100	5	3	30	8	7	883	0.737	0.739	0.00	0.27
100	5	3	30	9	3	77	0.700	0.737	0.04	5.05
100	5	3	30	10	3	71	0.700	0.762	0.06	8.15
100	5	3	50	1	3	82	0.700	0.707	0.01	1.02
100	5	3	50	2	7	205	0.803	0.820	0.02	2.05
100	5	3	50	3	7	806	0.735	0.766	0.03	3.94
100	5	3	50	4	3	131	0.700	0.710	0.01	1.37
100	5	3	50	5	3	133	0.700	0.743	0.04	5.75
100	5	3	50	6	3	89	0.700	0.788	0.09	11.21
100	5	3	50	7	3	87	0.700	0.709	0.01	1.31
100	5	3	50	8	3	77	0.700	0.701	0.00	0.20
100	5	3	50	9	3	114	0.700	0.719	0.02	2.63
100	5	3	50	10	3	78	0.700	0.788	0.09	11.16
100	7	3	10	1	6	129	0.700	0.703	0.00	0.39
100	7	3	10	2	3	40	0.700	0.886	0.19	21.02
100	7	3	10	3	9	143	0.700	0.713	0.01	1.78
100	7	3	10	4	2	46	0.700	0.761	0.06	8.00
100	7	3	10	5	3	39	0.700	0.700	0.00	0.00
100	7	3	10	6	3	51	0.700	0.839	0.14	16.61
100	7	3	10	7	3	43	0.700	0.704	0.00	0.58
100	7	3	10	8	6	145	0.707	0.722	0.01	2.06
100	7	3	10	9	3	43	0.700	0.802	0.10	12.71

100	7	3	10	10		159	0.700	0.710	0.01	1.44
100	7	3	30	1	7	790	0.758	0.762	0.00	0.43
100	7	3	30	2	3	43	0.700	0.810	0.11	13.58
100	7	3	30	3	3	60	0.700	0.842	0.14	16.87
100	7	3	30	4	7	171	0.760	0.761	0.00	0.12
100	7	3	30	5	3	62	0.700	0.815	0.12	14.15
100	7	3	30	6	7	153	0.700	0.779	0.08	10.11
100	7	3	30	7	3	63	0.700	0.715	0.01	2.06
100	7	3	30	8	11	507	0.781	0.785	0.00	0.57
100	7	3	30	9	3	56	0.700	0.739	0.04	5.31
100	7	3	30	10	3	60	0.700	0.788	0.09	11.12
100	7	3	50	1	3	91	0.700	0.785	0.08	10.80
100	7	3	50	2	3	65	0.700	0.724	0.02	3.36
100	7	3	50	3	3	75	0.700	0.708	0.01	1.17
100	7	3	50	4	7	300	0.793	0.799	0.01	0.81
100	7	3	50	5	3	78	0.700	0.753	0.05	7.06
100	7	3	50	6	7	829	0.817	0.818	0.00	0.09
100	7	3	50	7	3	76	0.700	0.718	0.02	2.47
100	7	3	50	8	3	48	0.700	0.707	0.01	1.04
100	7	3	50	9	3	54	0.700	0.835	0.14	16.20
100	7	3	50	10	3	82	0.700	0.710	0.01	1.37
200	3	3	10	1	6	144	0.703	0.705	0.00	0.36
200	3	3	10	2	7	151	0.784	0.785	0.00	0.10
200	3	3	10	3	3	42	0.700	0.725	0.03	3.47
200	3	3	10	4	3	59	0.700	0.730	0.03	4.16
200	3	3	10	5	3	74	0.700	0.720	0.02	2.71
200	3	3	10	6	3	76	0.700	0.729	0.03	3.92
200	3	3	10	7	3	56	0.700	0.705	0.00	0.71
200	3	3	10	8	3	59	0.700	0.700	0.00	0.00
200	3	3	10	9	3	58	0.700	0.700	0.00	0.00
200	3	3	10	10	3	40	0.700	0.702	0.00	0.29
200	3	3	30	1	3	77	0.700	0.706	0.01	0.87
200	3	3	30	2	3	74	0.700	0.707	0.01	0.96
200	3	3	30	3	3	55	0.700	0.726	0.03	3.62
200	3	3	30	4	3	75	0.700	0.706	0.01	0.80
200	3	3	30	5	3	53	0.700	0.836	0.14	16.29
200	3	3	30	6	7	395	0.895	0.895	0.00	0.00

200	3	3	30	7	3	73	0.700	0.795	0.09	11.91
200	3	3	30	8	3	67	0.700	0.700	0.00	0.00
200	3	3	30	9	3	57	0.700	0.700	0.00	0.00
200	3	3	30	10	3	78	0.700	0.834	0.13	16.07
200	3	3	50	1	3	81	0.700	0.725	0.02	3.44
200	3	3	50	2	3	87	0.700	0.718	0.02	2.54
200	3	3	50	3	3	121	0.700	0.709	0.01	1.33
200	3	3	50	4	3	88	0.700	0.704	0.00	0.53
200	3	3	50	5	3	97	0.700	0.703	0.00	0.39
200	3	3	50	6	3	71	0.700	0.700	0.00	0.00
200	3	3	50	7	3	62	0.700	0.700	0.00	0.00
200	3	3	50	8	3	73	0.700	0.734	0.03	4.60
200	3	3	50	9	7	864	0.757	0.765	0.01	1.03
200	3	3	50	10	3	60	0.700	0.880	0.18	20.46
200	5	3	10	1	3	75	0.700	0.716	0.02	2.28
200	5	3	10	2	3	88	0.700	0.853	0.15	17.97
200	5	3	10	3	3	91	0.700	0.754	0.05	7.20
200	5	3	10	4	3	86	0.700	0.789	0.09	11.27
200	5	3	10	5	9	317	0.707	0.707	0.00	0.00
200	5	3	10	6	2	87	0.700	0.741	0.04	5.58
200	5	3	10	7	3	62	0.700	0.716	0.02	2.30
200	5	3	10	8	2	69	0.703	0.843	0.14	16.56
200	5	3	10	9	3	93	0.700	0.730	0.03	4.15
200	5	3	10	10	3	107	0.700	0.769	0.07	9.01
200	5	3	30	1	3	90	0.700	0.721	0.02	2.93
200	5	3	30	2	3	129	0.700	0.733	0.03	4.54
200	5	3	30	3	3	122	0.700	0.744	0.04	5.89
200	5	3	30	4	3	78	0.700	0.747	0.05	6.26
200	5	3	30	5	3	131	0.700	0.744	0.04	5.87
200	5	3	30	6	3	83	0.700	0.719	0.02	2.70
200	5	3	30	7	3	103	0.700	0.717	0.02	2.39
200	5	3	30	8	3	107	0.700	0.763	0.06	8.21
200	5	3	30	9	3	116	0.700	0.756	0.06	7.47
200	5	3	30	10	3	107	0.700	0.725	0.02	3.40
200	5	3	50	1	3	125	0.700	0.810	0.11	13.62
200	5	3	50	2	3	120	0.700	0.848	0.15	17.44
200	5	3	50	3	9	1,466	0.783	0.814	0.03	3.78

200	5	3	50	4	3	2,277	0.700	0.760	0.06	7.93
200	5	3	50	5	3	162	0.700	0.720	0.02	2.78
200	5	3	50	6	3	86	0.700	0.810	0.11	13.56
200	5	3	50	7	3	14,250	0.700	0.721	0.02	2.92
200	5	3	50	8	3	95	0.700	0.728	0.03	3.81
200	5	3	50	9	3	112	0.700	0.715	0.01	2.08
200	5	3	50	10	3	86	0.700	0.721	0.02	2.98
200	7	3	10	1	2	66	0.700	0.718	0.02	2.56
200	7	3	10	2	3	61	0.700	0.710	0.01	1.37
200	7	3	10	3	7	158	0.831	0.831	0.00	0.00
200	7	3	10	4	3	44	0.700	0.700	0.00	0.00
200	7	3	10	5	3	48	0.700	0.709	0.01	1.22
200	7	3	10	6	7	210	0.736	0.798	0.06	7.74
200	7	3	10	7	3	58	0.700	0.705	0.01	0.75
200	7	3	10	8	7	227	0.739	0.741	0.00	0.22
200	7	3	10	9	3	67	0.700	0.700	0.00	0.06
200	7	3	10	10	3	68	0.700	0.747	0.05	6.23
200	7	3	30	1	3	105	0.700	0.742	0.04	5.64
200	7	3	30	2	7	231	0.711	0.715	0.00	0.53
200	7	3	30	3	3	133	0.700	0.779	0.08	10.17
200	7	3	30	4	3	106	0.700	0.765	0.07	8.50
200	7	3	30	5	7	886	0.781	0.782	0.00	0.20
200	7	3	30	6	7	854	0.750	0.760	0.01	1.27
200	7	3	30	7	3	81	0.700	0.727	0.03	3.72
200	7	3	30	8	7	301	0.742	0.751	0.01	1.25
200	7	3	30	9	13	1,216	0.715	0.716	0.00	0.06
200	7	3	30	10	9	351	0.808	0.818	0.01	1.21
200	7	3	50	1	3	191	0.700	0.723	0.02	3.23
200	7	3	50	2	3	149	0.700	0.718	0.02	2.47
200	7	3	50	3	7	462	0.758	0.758	0.00	0.00
200	7	3	50	4	7	3,626	0.758	0.758	0.00	0.00
200	7	3	50	5	9	639	0.719	0.721	0.00	0.24
200	7	3	50	6	3	178	0.700	0.785	0.09	10.88
200	7	3	50	7	3	167	0.700	0.824	0.12	15.05
200	7	3	50	8	3	183	0.700	0.790	0.09	11.41
200	7	3	50	9	7	1,153	0.771	0.773	0.00	0.28
200	7	3	50	10	3	135	0.700	0.716	0.02	2.23

6 CONCLUSION AND FUTURE RESEARCH

This thesis is concerned with investigating Interactive Inverse Multiple Criteria Sorting Problem. This type of problem is very new in the literature. There are, of course, number of studies for the aspect of inverse problems and the interactive approaches. The major difference of this research is combining the inverse multiple criteria sorting case with the contribution of a decision making mechanism which is called the decision maker.

The developed model aims to re-assign alternatives to the better classes, if possible, by taking a subset of actions. The corresponding procedure involves a decision maker.

“Majority Rule Sorting" is considered as the underlying sorting method of this thesis. The developed solution approach is not unique for the MR-Sort so, can be easily modified for the preferred sorting model.

Within the scope of this thesis, we developed 2 algorithms for 2 cases. These cases differ from each other according to the number of classes in the problem. We studied the problems of having 2 classes and 3 classes. The solution methodology of each cases is basically the same with minor differences. Essentially, the developed algorithm starts with finding the middlemost point between the best and the worst point. After that, the decision maker chooses between the middlemost point and the initial point. The algorithm continues by finding out the adjacency point to the preferred point. For the case of 2 classes, the adjacency is a one step side point of the preferred point. For the case of 3 classes, 3 number of adjacencies are occurred and the cost closest point to the preferred point is chosen of them. When the adjacency points is detected, it is asked to the decision maker to choose one of them. By using the decision maker's choices the iterations are continued with the same logic. Finding of middlemost point is only done in the first iteration. At the rest of the algorithms, the preferred point by the decision maker is set as the incumbent point and we find the corresponding adjacency of the new incumbent point. By asking the decision maker, new incumbent point of the current iteration is found out.

These two algorithms are varied from each other also by calculating a new point for the case of 3 classes. Every 2 iterations, by using the data from the preferences of the decision maker, the new set of weights is calculated. While maximizing the score by considering with the mentioned weight set, the new point is calculated. In this iteration, the decision maker becomes a part and made a selection between new point and the incumbent point of the previous iteration. In general, the algorithm for the case of 3 classes follows calculating the middlemost point, asking the decision maker to choose the middlemost and the initial point, setting the preferred point as incumbent of the current iteration, finding the 3 adjacencies of the incumbent point, calculating the cost closest one, asking the decision maker to choose the incumbent and the adjacency point, setting the preferred point as incumbent of the current iteration, finding new set of weight, finding new point by using new set of weight, asking the decision maker to choose between the new point and the incumbent of the previous iteration, setting the preferred point as incumbent of the current iteration, finding the 3 adjacencies of the incumbent point and etc.

The definitions and the developed model are stated and explained in detail. Around 1,000 instances are solved by using this algorithm and their results are presented. It is observed from the results of experiments that in average the algorithms find very close to the optimal solution.

This thesis is open for new researches to study of different approaches and different cases. Some possible studies could be considering more than 3 class-case, a group of decision makers, the quadratic and tchebycheff cases of value function.

REFERENCES

Ahuja, Ravindra K. and James B. Orlin (2001). *Inverse Optimization*. Operations Research 49.

Aksoy, Yasemin, W. Timothy Butler, and D. Eliot Minor (1994). *Comperative Studies in Interactive Multiple Objective Mathematical Programming*.

Alves, Maria Joao and Joao Climaco (2006). *A Review of Interactive Methods for Multiobjective Integer and Mixed-integer Programming*. European Journal of Operational Research 180.

Belton, Valerie and Theodor J. Stewart (2002). *Multiple Criteria Decision Analysis An Integrated Approach*. Springer Science+Business Media.

Bernard, Roy and Denis Bouyssou (1993). *Aide Multicritere a la Decision: Methodes et Cas*.

— (1996). *Decision-Aid: An Elementary Introduction With Emphasis On Multiple Criteria*.

Chen, Yu (2006). *Multiple Criteria Decision Analysis: Classification Problems and Solutions*. A Thesis, University of Waterloo.

Cóndor, Rocío Dánica (2008). *A Multicriteria Decision Framework: Assessing synergies among the Rio Conventions at forestry project level*.

Dyer, James S. et al. (1992). *Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years*. Management Science 38.

Figueira, Jose, Salvatore Greco, and Matthias Ehrgott (2005). *Multiple Criteria Decision Analysis A State of Art Survey*. Springer Science+Business Media.

— (2010). *Trend in Multiple Criteria Decision Analysis*. Springer Science+Business Media.

Gardiner, Lorraine R. and Ralph E. Steuer (1994). *Unified interactive multiple objective programming*. European Journal of Operational Research 74.

Govindan, Kannan and Martin Brandt Jepsen (2015). *ELECTRE: A Comprehensive Literature Review on Methodologies and Applications*.

Grauer, M., A. Lewandowski, and A.P. Wierzbicki (1982). *Multiobjective and Stochastic Optimization*. International Institute for Applied Systems Analysis.

Jacquet-Lagreze, E. (1995). *An application of the UTA discriminant model for the evaluation of RD projects*.

Jain, Sanjay and Nitin Arya (2013). *An Inverse Capacitated Transportation Problem*. IOSR Journal of Mathematics 5.

Kaliszewski, Ignacy and Stanley Zionts (2004). *A Generalization of The Zionts-Wallenius Multiple Criteria Decision Making Algorithm*. Control and Cybernetics 33.

Koksalan, Murat, H. Mark Karwan, and Stanley Zionts (1984). *An Improved Method For Solving Multiple Criteria Problems Involving Discrete Alternatives*. IEEE Trans. on Systems, Man and Cybernetics 14.

Kosmidou, K. et al. (2008). *Country Risk Evaluation*. Springer Science+Business Media.

Leroy, Agnes, Vincent Mousseau, and Marc Pirlot (2011). *Learning the Parameters of a Multiple Criteria Sorting Method Based on a Majority Rule*.

MacCrimmon, K.R. (1973). *An Overview Of Multiple Objective Decision Making*. University of South Carolina Press, Columbia.

Mousseau, Vincent, Özgür Özpeynirci, and Selin Özpeynirci (2017). *Inverse Multiple Criteria Sorting Problem*. Annals of Operations Research, forthcoming.

Narula, Subhash C. (1992). *An Interactive Algorithm For Solving Multiple Objective Integer Linear Programming Problems*. European Journal of Operational Research 79.

Özlen, Melih and Meral Azizoğlu (2008). *Multi-Objective Integer Programming: A General Approach for Generating All Non-dominated Solutions*. European Journal of Operational Research 199.

Reeves, R. Gary and R. Kenneth MacLeod (1998). *Some Experiments in Tchebycheff-based Approaches for Interactive Multiple Objective Decision Making*.

Roy, Bernard (1971). *Problems and Methods With Multiple Objective Functions*. Mathematical Programming 1.

— (1985). *Methodologie Multicritère d'aide a la decision*. France: Economica, Paris.

— (1991). *The Outranking Approach and The Foundations of ELECTRE Methods*. Theory and Decision 31.

— (1996). *Multicriteria Methodology for Decision Analysis*. Germany: Kluwer Academic Publishers.

Saaty, T.L. (1990). *How to make a decision: The Analytic Hierarchy Process*. European Journal of Operational Research 48.

Serefini, P. (1985). *Mathematics of Multi-Objective Optimization*. Springer-Verlag Wien GMBH.

Sobrie, Olivier, Vincent Mousseau, and Marc Pirlot (2013). *Learning a Majority Rule Model from Large Set of Assignment Examples*.

Steuer, Ralph E. and Eng-Ung Choo (1983). *An Interactive Weighted Tchebycheff Procedure For Multiple Objective Programming*. Mathematical Programming 26.

Triantaphyllou, E. et al. (1998). *Multi-Criteria Decision Making: An Operations Research Approach*. Encyclopedia of Electrical and Electronics Engineering 15.

Villareal, Bernardo and H. Mark Karwan (1981). *An Interactive Dynamic Programming Approach to Multicriteria Discrete Programming*. Journal of Mathematical Analysis and Applications 81.

Wang, Tairan (2015). *Decision Making and Modelling Uncertainty for the Multi-criteria Analysis of Complex Energy Systems*. A PhD Thesis, Génie Industriel.

Zionts, Stanley and Jykri Wallenius (1976). *An Interactive Programming Method for Solving The Multiple Criteria Problem*. Management Science 22.

Zopounidis, Constantin and Michael Doumpos (1999). *A multicriteria decision aid methodology for sorting decision problems: The case of financial distress*. Computational Economics 14.

— (2002). *Inverse Optimization*. European Journal of Operational Research 138.