INTERACTIVE EVOLUTIONARY APPROACHES
TO
MULTI-OBJECTIVE FEATURE SELECTION

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

BY

MÜBERRA ÖZMEN

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

AUGUST 2016

Approval of the thesis:

AN INTERACTIVE EVOLUTIONARY APPROACH TO MULTI-OBJECTIVE FEATURE SELECTION

submitted by **MÜBERRA ÖZMEN** in partial fulfillment of the requirement for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Enver
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Murat Köksalan
Head of Department, **Industrial Engineering** _______________

Prof. Dr. Murat Köksalan _______________
Supervisor, **Industrial Engineering Dept., METU**

Assist. Prof. Dr. Gülşah Karakaya _______________
Co-Advisor, **Dept. of Business Administration, METU**

Examining Committee Members:

Prof. Dr. Nur Evin Özdemirel _______________
Industrial Engineering Dept., METU

Prof. Dr. Murat Köksalan
Industrial Engineering Dept., METU _______________

Assist. Prof. Dr. Diclehan Tezcaner Öztürk
Dept. of Industrial Engineering, TEDU _______________

Assoc. Prof. Dr. İsmail S. Bakal
Industrial Engineering Dept., METU _______________

Assist. Prof. Dr. Özgen Karaer
Industrial Engineering Dept., METU _______________

Date: 24.08.2016

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

Name, Last name : Müberra ÖZMEN

Signature :

ABSTRACT

INTERACTIVE EVOLUTIONARY APPROACHES TO MULTI-OBJECTIVE FEATURE SELECTION

Özmen, Müberra
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Murat Köksalan
Co-Advisor: Assist. Prof. Dr. Gülşah Karakaya

August 2016, 93 pages

In feature selection problems, the aim is to select a subset of features to characterize an output of interest. In characterizing an output, we may want to consider multiple objectives such as maximizing classification performance, minimizing number of selected features or cost, etc. We develop a preference-based approach for multi-objective feature selection problems. Finding all Pareto optimal subsets may turn out to be a computationally demanding problem and we still would need to select a solution eventually. Therefore, we develop interactive evolutionary approaches that aim to converge to a subset that is highly preferred by the decision maker. We test our approach on several instances simulating decision-maker preferences by underlying preference functions and demonstrate that it works well.

Keywords: feature selection, subset selection, interactive approach, evolutionary algorithm

ÖZ

ÇOK AMAÇLI DEĞİŞKEN SEÇİMİNE ETKİLEŞİMLİ EVRİMSEL YAKLAŞIMLAR

Özmen, Müberra
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Murat Köksalan
Ortak Tez Yöneticisi: Y. Doç. Dr. Gülşah Karakaya

August 2016, 93 Sayfa

Özellik seçme problemlerinde, amaç çıktı değişkenini karakterize etmek için girdi değişkenlerinin bir altkümesini seçmektir. Çıktı değişkenini karakterize ederken, sınıflandırma performansını maksimize etmek, seçilen girdi değişkeni sayısını minimize etmek gibi birden fazla amaç düşünülebilir. Çok amaçlı değişken seçimi problem için tercihe bağlı bir yaklaşım geliştirdik. Tüm domine edilemeyen çözümleri bulmak, işlemsel açıdan büyük çaba gerektiren bir problemdir ve yine de sonunda bir çözümün seçilmesi gerekir. Bu nedenle, karar verici tarafından tercih edilen bir çözüme yönelmeyi amaçlayan etkileşimli evrimsel yaklaşımlar geliştirdik. Karar vericinin tercihlerini bir tercih fonksiyonuyla simule ederek, yaklaşımımızı örnekler üzerinde test ettik ve iyi çalıştığını gösterdik.

Anahtar Kelimeler: özellik seçimi, altküme seçimi, etkileşimli yaklaşım, evrimsel algoritma

To My Family

ACKNOWLEDGMENTS

I would like to express deepest gratitude to my advisors Prof. Dr. Murat Köksalan and Assistant Prof. Dr. Gülşah Karakaya for supporting me with their comprehensive expertise and inspirational advices. Thanks to their structured approach and constructive attitude, this study broadened my perception of science and research, and encouraged me a lot to pursue a career of research.

I am thankful to my friends Sıdıka Tunç and Yücel Türegün for their inexhaustible support in every stage of this study and for their valuable friendship. I always feel so lucky to have them around me anytime I need.

Thanks a lot to my friends Burcu Gökgöz and Burak Atmaca for making every break to work an incredible source of energy. I am grateful to them for always keeping me full of life with their friendship.

I owe my motivation and energy to my family and their unconditional love in every stage of my life. Therefore, most importantly, I wish to sincerely thank to my parents and sister for their encouragement and trust.

Finally, I would like to thank all members of my department for contributing to this positive and productive working environment.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS ix

LIST OF TABLES xi

LIST OF FIGURES xiii

CHAPTERS

1. INTRODUCTION 1

2. MAIN CONCEPTS AND DEFINITIONS 5
   - 2.1 Feature Selection Problem 5
   - 2.2 Multi-objective Optimization 8

3. LITERATURE REVIEW 11
   - 3.1 Feature Selection Theory 11
   - 3.2 Multi-objective Approaches and Applications 13

4. THE FEATURE SELECTION PROBLEM ADDRESSED 17
   - 4.1 General Formulation of the Problem 17
   - 4.2 Objective Functions 18
     - 4.2.1 Accuracy 18
     - 4.2.2 Cardinality 19
     - 4.2.3 Cost 20
     - 4.2.4 Risk 21

5. ALGORITHMS 23
   - 5.1 Overview 23

5.2 Interactive Territory Defining Evolutionary Algorithm for the Feature Selection Problem (iTDEA-fs) 26
5.2.1 Definitions 27
5.2.2 iTDEA-fs Interaction Stages 27
5.2.3 iTDEA-fs Population Update 30
5.2.4 iTDEA-fs Archive Update 30
5.3 Improvement Issues of iTDEA-fs 31
5.4 Interactive Weight Reducing Evolutionary Algorithm for Feature Selection (iWREA-fs) 36
5.4.1 iWREA-fs Interaction Stages 37
5.4.2 iWREA-fs Population Update 39
5.4.3 iWREA-fs Archive Update 39
6. COMPUTATIONAL EXPERIMENTS 41
6.1 Datasets 41
6.2 Implementation 42
6.3 Experimental Setting 43
6.4 Results and Discussion 46
6.4.1 Performance Indicator 46
6.4.2 Evaluating Algorithms 47
6.4.3 Comparison of Computational Efforts 53
6.4.4 Features Selected 54
6.4.5 Some Results for Quadratic Underlying Preference Functions 55
7. CONCLUSION 57
REFERENCES 59
APPENDICIES 63
A. PERFORMANCE MEASUREMENTS 63
B. ARCHIVE PROGRESSES 68
C. COMPUTATIONAL TIMES 87
D. QUADRATIC PREFERENCE FUNCTION RESULTS 89

LIST OF TABLES

TABLES

Table 2.1 Confusion matrix 7

Table 5.1 Filtered archive in the example 33

Table 6.1 Datasets used in experiments 41

Table 6.2 Types of DM's preference function weights tested 43

Table 6.3 Parameter settings of Heart Disease (13) experiments 45

Table 6.4 Parameter settings of Vehicle (18) experiments 45

Table 6.5 Parameter settings of German (24) experiments 45

Table 6.6 Parameter settings of Breast Cancer (32) experiments 45

Table 6.7 Mean and standard deviation of the percentage deviations for Heart Disease (13) experiments 48

Table 6.8 Mean and standard deviation of the percentage deviations for Vehicle (18) experiments 48

Table 6.9 Mean and standard deviation of the percentage deviations for German (24) Experiments 48

Table 6.10 Mean and standard deviation of the percentage deviations for Breast Cancer (32) experiments 48

Table 6.11 95% confidence intervals on paired differences of percentage deviations for Heart Disease (13) experiments 49

Table 6.12 95% confidence intervals on paired differences of percentage deviations for Vehicle (18) experiments 49

Table 6.13 95% confidence intervals on paired differences of percentage deviations for German (24) experiments 49

Table 6.14 95% confidence intervals on paired differences of percentage deviations for Breast Cancer (32) experiments 49

Table 6.15 CPU times of algorithms (in seconds) 54

Table 6.16 Existence frequencies of features in the subsets ranked in top 5 percent 55

Table A.1 Heart Disease – Accuracy Favored 63

Table A.2 Heart Disease – Accuracy and Cost Tradeoff 63

Table A.3 Heart Disease – Equally Treated 64

Table A.4 Vehicle – Accuracy Favored 64

Table A.5 Vehicle – Accuracy and Cost Tradeoff 64

Table A.6 Vehicle – Equally Treated 65

Table A.7 German – Accuracy Favored..65

Table A.8 German – Accuracy and Cost Tradeoff..65

Table A.9 German – Equally Treated 66

Table A.10 Breast Cancer – Accuracy Favored 66

Table A.11 Breast Cancer – Accuracy and Cost Tradeoff 66

Table A.12 Breast Cancer – Equally Treated 67

Table C.1 CPU Times of Heart Disease experiments (in seconds)...........................87

Table C.2 CPU Times of Vehicle experiments (in seconds).....................................87

Table C.3 CPU Times of German experiments (in seconds) 87

Table C.4 CPU Times of Breast Cancer experiments (in seconds) 88

LIST OF FIGURES

FIGURES

Figure 3.1 Artificial Neural Networks (ANN) 12

Figure 5.1 An example of crossover and mutation operations................................. 26

Figure 5.2 Smaller territory levels are assigned to the solutions in the preferred region in iTDEA-fs. 28

Figure 5.3 Archive update in iTDEA-fs: the offspring is accepted into archive if d ≤ τ.. 31

Figure 5.4 Favorable weights and actual weights in the example............................. 33

Figure 5.5 *Model (Mid∞)* weights, favorable weights and actual weights in the example 36

Figure 6.1 Archive progress of iTDEA-fs on Breast Cancer dataset experiments with Accuracy Favored weight set 50

Figure 6.2 Archive progress of iWREA-fs on Breast Cancer dataset experiments with Accuracy Favored weight set 51

Figure 6.3 Archive progress of No Interaction on Breast Cancer dataset experiments with Accuracy Favored weight set t.. 51

Figure 6.4 5th replication on Breast Cancer dataset experiments with Accuracy Favored weight set 52

Figure 6.5 4th replication of quadratic preference function experiments with Accuracy Favored weight set 56

Figure B.1 Heart Disease – Accuracy Favored – iTDEA-fs 68

Figure B.2 Heart Disease – Accuracy Favored – iWREA-fs 69

Figure B.3 Heart Disease – Accuracy Favored – No Interaction.............................. 69

Figure B.4 Heart Disease – Accuracy and Cost Tradeoff – iTDEA-fs 70

Figure B.5 Heart Disease – Accuracy and Cost Tradeoff – iWREA-fs 70

Figure B.6 Heart Disease – Accuracy and Cost Tradeoff – No Interaction 71

Figure B.7 Heart Disease – Equally Treated – iTDEA-fs 71

Figure B.8 Heart Disease – Equally Treated – iWREA-fs 72

Figure B.9 Heart Disease – Equally Treated – No Interaction 72

Figure B.10 Vehicle – Accuracy Favored – iTDEA-fs 73

Figure B.11 Vehicle – Accuracy Favored – iWREA-fs 73

Figure B.12 Vehicle – Accuracy Favored – No Interaction 74

Figure B.13 Vehicle – Accuracy and Cost Tradeoff – iTDEA-fs 74

Figure B.14 Vehicle – Accuracy and Cost Tradeoff – iWREA-fs 75

Figure B.15 Vehicle – Accuracy and Cost Tradeoff – No Interaction 75

Figure B.16 Vehicle – Equally Treated – iTDEA-fs 76

Figure B.17 Vehicle – Equally Treated – iWREA-fs 76

Figure B.18 Vehicle – Equally Treated – No Interaction 77

Figure B.19 German – Accuracy Favored – iTDEA-fs 77

Figure B.20 German – Accuracy Favored – iWREA-fs 78

Figure B.21 German – Accuracy Favored – No Interaction 78

Figure B.22 German – Accuracy and Cost Tradeoff – iTDEA-fs 79

Figure B.23 German – Accuracy and Cost Tradeoff – iWREA-fs 79

Figure B.24 German – Accuracy and Cost Tradeoff – No Interaction 80

Figure B.25 German – Equally Treated – iTDEA-fs 80

Figure B.26 German – Equally Treated – iWREA-fs 81

Figure B.27 German – Equally Treated – No Interaction 81

Figure B.28 Breast Cancer – Accuracy Favored – iTDEA-fs 82

Figure B.29 Breast Cancer – Accuracy Favored – iWREA-fs 82

Figure B.30 Breast Cancer – Accuracy Favored – No Interaction 83

Figure B.31 Breast Cancer – Accuracy Cost Tradeoff – iTDEA-fs 83

Figure B.32 Breast Cancer – Accuracy Cost Tradeoff – iWREA-fs 84

Figure B.33 Breast Cancer – Accuracy Cost Tradeoff – No Interaction 84

Figure B.34 Breast Cancer – Equally Treated – iTDEA-fs 85

Figure B.35 Breast Cancer – Equally Treated – iWREA-fs 85

Figure B.36 Breast Cancer – Equally Treated – No Interaction 86

Figure D.1 Accuracy Favored – iTDEA-fs 89

Figure D.2 Accuracy Favored – iWREA-fs 90

Figure D.3 Accuracy Favored – No Interaction 90

Figure D.4 Accuracy and Cost Tradeoff – iTDEA-fs 91

Figure D.5 Accuracy and Cost Tradeoff – iWREA-fs 91

Figure D.6 Accuracy and Cost Tradeoff – No Interaction 92

Figure D.7 Equally Treated – iTDEA-fs 92

Figure D.8 Equally Treated – iWREA-fs 93

Figure D.9 Equally Treated – No Interaction 93

CHAPTER 1

INTRODUCTION

In classification problems, supervised learning algorithms, such as decision trees, support vector machines, neural networks etc. are used to predict the class (or output variable) of an instance by observing its features' (or input variables) values. Supervised learning algorithms train a prediction model over a dataset, in which different feature and class values of some past observations are provided, by understanding the relationship between the features and classes. Hence, the prediction model can be used to classify a new instance based on its features.

The classification performance of the learning algorithm depends on its ability to detect the relationship between input and output variables accurately. However, the presence of features that are irrelevant to the class, or the redundancy within the features may have a negative impact on the classification performance of the learning algorithm (Kohavi and John, 1997). Yu and Liu (2004) classify the features based on their relevance with respect to the output as *strongly relevant*, *weakly relevant*, and *irrelevant*. A feature is *strongly relevant* to class if its existence affects classification performance independently from the other features used, *weakly relevant* if it affects the classification performance depending on the other features used and *irrelevant* if the feature does not affect the classification performance at all. They argue that the optimal subset of features in terms of classification performance includes all strongly relevant, and weakly relevant and non-redundant features. Selecting a subset that comprises of strongly relevant, and weakly relevant and non-redundant features to be used in the prediction model of the learning algorithm (or classifier), instead of using them all, is called as *feature selection problem*.

Feature selection aims to improve the classification performance by eliminating irrelevant and redundant features. The decrease in the number of features to be used in the prediction model is also useful in terms of reducing storage requirements, improving the time efficiency, and simplifying the prediction model itself (Guyon, 2003). Therefore, feature selection methods are used in many areas, such as handwritten digit recognition, facial recognition, medical diagnosis, gene marker recognition etc.

Even though, reduction in the number of input variables seems to be a natural outcome of the feature selection procedure that aims at maximizing the classification performance, it is possible to consider minimizing the cardinality of subset as another objective. That is, one may be willing to reduce the number of variables beyond the number of variables in the subset that gives the best classification performance to enjoy the benefits of reducing cardinality. In that case, the problem is converted into a multi-objective problem. Depending on the scope of the problem other objectives can also be defined. For example, in a medical diagnosis application, minimizing the screening costs of medical tests that will give feature values or minimizing the health related risks involved in those tests for the patient could be set as objectives.

The algorithms developed for solving feature selection problem can be investigated in two dimensions. Firstly, since it is not straightforward to measure the impact of using a feature on classification performance, different strategies have been developed for subset selection; which are filter and wrapper approaches (Kohavi and John, 1997). Secondly, since the number of possible subsets grows exponentially with the number of available features, the feature selection problem is combinatorial in nature. Therefore, many optimization techniques are used to solve the feature selection problem, such as sequential backward selection, branch and bound, best-first search, and genetic algorithms (Kohavi and John, 1997).

In the literature, feature selection problem is usually treated as a bi-objective problem in which the objectives are maximizing the classification performance and minimizing the cardinality of the subset. Most of the studies aim to find all non-dominated solutions for these two objectives, which refers to finding the subset with best classification performance for each cardinality level. However, in the presence of more objectives, enumeration of all non-dominated solutions is not practical and useful because of the combinatorial nature of the problem. Instead of finding all non-dominated solutions, concentrating on solutions that are of more interest to the decision maker (DM) of the problem, is more practical. Therefore, in this study, interactive evolutionary algorithms are developed for multi-objective feature selection problems that aim to converge the most preferred solution by guiding the search towards the regions that consists of appealing solutions for the DM.

Measuring the classification performance is an important part of feature selection problems and a number of supervised learning algorithms have been developed in the literature. We leave this measurement problem out of the context of our research and we use one of the existing supervised algorithms for this purpose. The main contribution of this study is developing a multi-objective optimization approach that is compatible with the characteristics of the feature selection problem.

The rest of the thesis is organized as follows. In Chapter 2, main concepts and definitions regarding the problem are provided. In Chapter 3, a literature review of related studies is given. In Chapter 4, the feature selection problem addressed in this study is defined. In Chapter 5, the interactive algorithms to find a preferred solution of the DM are developed and in Chapter 6 these algorithms are tested on several instances. Concluding remarks and future research directions are outlined in Chapter 7.

CHAPTER 2

MAIN CONCEPTS AND DEFINITIONS

In this chapter, basic concepts and definitions regarding feature selection problem and multi-objective optimization will be provided and explained on a small example.

2.1 Feature Selection Problem

Let there be a medical doctor who would like to make diagnosis of her patients' disease. Assume there is a record of past patients on hand in which the patients' all test results and actual diseases are given. Using the past records, the doctor is to decide how the test results should be evaluated in order to make diagnosis on future patients accurately. The past records can be defined as the dataset in the feature selection problem in which each patient corresponds to an instance, and test results and actual disease of each patient correspond to feature values and class variable value of each instance, respectively. The doctor's expertise of constructing the relationship between the test results and diagnosis can be thought of as the learning algorithm. What we would like to decide in the feature selection problem is which tests should be performed so that the doctor's performance of making accurate diagnosis is maximized.

In this study, the classification problems where each instance is classified in only one of the non-overlapping classes are addressed. The classification problems with two non-overlapping classes and multiple non-overlapping classes are called as *binary class* and *multi-class* classification problems, respectively (Sokolova and Lapalme, 2009). In the medical diagnosis example, if the doctor has to decide whether or not her patient has cancer, the problem is binary class. On the other hand, if the doctor classifies the disease based on the existence/type of tumor as class 1: no tumor, class 2: benign tumor, and class 3: malignant tumor, then the problem is multi-class as there are more than two classes.

Let the *dataset* consist of N instances. Assuming there exists M available features defined as a vector $X = \{x_1, \dots, x_M\}$ and a class variable y, the observed values of M features and class variable for each instance is provided in the dataset. Let S be a subset of X and $f(S)$ denote the classification performance of using the features in S to train the prediction model.

The feature selection problem with a single objective of maximizing the classification performance can be formulated as follows:

$$\max f(S)$$

s.to

$$S \in X$$

Once the learning algorithm is trained using past data, it can be used to classify the future observations based on their feature values. Actually, without knowing what will be the observations in future, the classification performance cannot be measured exactly. However, it can be estimated by using some observations on hand for testing the trained algorithm. For this aim, the dataset is divided into two sets: *training* and *testing sets*. The instances in the training set are used to train the learning algorithm and then the trained model is used to determine the classes of instances in the testing set. The classification performance of the algorithm can be estimated by using its performance on classifying the instances in the testing set.

The classification performance depends on the division of dataset into training and testing sets. In order to reduce this dependency, k-fold cross validation procedure can be applied (Kohavi and John, 1997). In this procedure the dataset is divided into training and testing sets k times, such that N/k instances are selected from the dataset randomly to form the testing set, and rest of the instances are used to form the training set. For each fold, the training set is used to train the algorithm and its classification performance on the testing set is calculated based on a predefined performance indicator (e.g. f). Let f_i be the classification performance at the i^{th}

fold. Then, the final classification performance, f^*, is calculated as the average of performances obtained in k folds, i.e. $f^* = \frac{\sum_{i=1}^{i=k} f_i}{k}$.

To calculate a certain classification performance measure, a confusion matrix is obtained by comparing the predicted and actual classes of the instances in the testing dataset. For a binary class classification problem where there are two classes (e.g. positive and negative), the confusion matrix would be as in Table 2.1.

Table 2.1 Confusion matrix

		Predicted class	
		Positive	**Negative**
Actual class	**Positive**	**True Positive (tp)** number of instances whose actual classes are positive and predicted as positive	**False Negative (fn)** number of instances whose actual classes are positive but predicted as negative
	Negative	**False Positive (fp)** number of instances whose actual classes are negative but predicted as positive	**True Negative (tn)** number of instances whose actual classes are negative and predicted as negative

The classification performance of a subset of features, namely $f(S)$, can be measured in terms of different indicators using the confusion matrix. There are several performance measures defined to be used in different areas (Sokolova and Lapalme, 2009). The formulations of three different indicators are given below.

$$\text{Accuracy} = \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}} \quad \text{Precision} = \frac{\text{tp}}{\text{tp+fp}} \quad \text{Sensitivity} = \frac{\text{tp}}{\text{tp+fn}}$$

Different indicators evaluate the performance of the learning algorithm in different senses. Accuracy is used as an indicator of overall effectiveness of the classifier, precision indicates what proportion of positively labeled instances are actually positive, and sensitivity stands for measuring the performance of identifying positively labeled instances. Back to the example of medical diagnosis, let us assume that the doctor is asked to make a diagnosis on 10 patients for having cancer or not, and it is actually known that 3 patients have cancer and 7 do not. It is observed that she classified 3 patients, who actually do not have cancer, as having cancer and classified all 3 patients with cancer correctly. Assuming the class of having cancer is the positive class, the accuracy, precision, and sensitivity of the doctor are 70%, 50%, and 100%, respectively.

2.2 Multi-objective Optimization

In multi-objective optimization problems there are two or more, generally conflicting, objectives to be optimized.

Let $\boldsymbol{x}$ and X represent the decision variable vector and feasible decision space, respectively. Let there be p objectives $z_1(\boldsymbol{x}), \dots, z_p(\boldsymbol{x})$ to be minimized and Z be the objective space defined by the feasible decision vectors. The general multi-objective optimization problem can be formulated as follows:

$$\text{"min"}\left\{z_1(\boldsymbol{x}), \dots, z_p(\boldsymbol{x})\right\}$$

s. to

$$\boldsymbol{x} \in X$$

The quotation marks are used to emphasize that the minimization of a vector is not a well-defined mathematical operation.

Definition 2.2.1: An objective vector $\boldsymbol{z}(\boldsymbol{x}') = \left(z_1(\boldsymbol{x}'), \dots, z_p(\boldsymbol{x}')\right)$ is said to *dominate* $\boldsymbol{z}(\boldsymbol{x}) = \left(z_1(\boldsymbol{x}), \dots, z_p(\boldsymbol{x})\right)$ if and only if $z_j(\boldsymbol{x}') \leq z_j(\boldsymbol{x})$ for all $j = 1, \dots, p$ and $z_j\left(\boldsymbol{x}^{'}\right) < z_j(\boldsymbol{x})$ for at least one j.

Definition 2.2.2: $\boldsymbol{z}(\boldsymbol{x})$ is *non-dominated* if and only if no $\boldsymbol{z}(\boldsymbol{x}')$ dominates it.

Definition 2.2.3: An objective vector $\boldsymbol{z}^* = \left(z_1^*, \dots, z_p^*\right)$ forms the *ideal point* in Z if and only if $z_j^* = \min_{\boldsymbol{x} \in X}\{z_j(\boldsymbol{x})\}$ for all $j = 1, \dots, p$.

Definition 2.2.4: An objective vector $\boldsymbol{z}^{nad} = \left(z_1^{nad}, \ldots, z_p^{nad}\right)$ forms the *nadir point* in Z if and only if $z_j^{nad} = \max_{x \in X} \{z_j(\boldsymbol{x})\}$ where $\boldsymbol{z}(\boldsymbol{x})$ is non-dominated.

In this study, interactive evolutionary algorithms that aim to find appealing solutions for a DM are developed for multi-objective feature selection problems. The DM of the problem is assumed to have an underlying monotone preference function, $U_{DM}(\boldsymbol{z})$, to be minimized. When the DM is presented with two solutions $\boldsymbol{z}(\boldsymbol{x})$ and $\boldsymbol{z}(\boldsymbol{x}')$, he/she prefers $\boldsymbol{z}(\boldsymbol{x})$ if $U_{DM}\left(\boldsymbol{z}(\boldsymbol{x})\right) < U_{DM}\left(\boldsymbol{z}(\boldsymbol{x}')\right)$. We assume that there are no indifference responses. Indifference can be handled by allowing a small range in the estimated preference function values of both $\boldsymbol{z}(\boldsymbol{x})$ and $\boldsymbol{z}(\boldsymbol{x}')$ as in Karakaya and Köksalan (2014). However, we do not address this case in this study. We also do not consider the case where the DM gives responses inconsistent with his underlying preference function.

CHAPTER 3

LITERATURE REVIEW

In this chapter, a background of theory developed regarding feature selection problem and a general literature review of different multi-objective applications are mentioned.

3.1 Feature Selection Theory

As mentioned before, the number of possible subsets of features grows exponentially with the number of available features, that is; for M features there are 2^M possible subsets. Therefore, different searching algorithms can be used to explore the solution space, such as sequential backward selection, branch and bound, best-first search and evolutionary algorithms (Kohavi and John, 1997). For a survey of evolutionary algorithms used for feature selection problem Xue et al. (2016) can be referred to.

There are two main approaches developed for subset selection: wrapper and filter approaches.

In the search phase of a subset selection algorithm, in order to estimate the classification performance of a learning algorithm for a subset of features, namely $f(S)$, the learning algorithm itself can be used directly. Applying this procedure to find the subset of features with best classification performance is called the *wrapper approach* (Kohavi and John, 1997).

Using the wrapper approach can be computationally time consuming since it requires to call learning algorithm to evaluate each subset found during search. Moreover, the classification performance estimated for a subset of features is dependent on the learning algorithm used. Therefore, instead of using a learning algorithm to estimate the classification performance during search, subsets of features can be evaluated with respect to some statistical measures (e.g. correlation, information theoretic measures), which is called the *filter approach* (Kohavi and John, 1997). Although the filter approach is more efficient in terms of computational time, wrapper approach provides more reliable estimation of classification performance of feature subset as it uses learning algorithm during the search phase.

There are many techniques used in supervised learning algorithms. *Decision Trees* are examples of logic-based algorithms in which classification rules are developed based on feature values. Instance based learning algorithms, such as *k-Nearest Neighbor (kNN)*, classifies the instances bases on their nearest neighbors in the training instances in terms of a distance metric. *Support Vector Machines (SVM)*, aim to create hyperplanes that separate the training data classes by maximizing the distance between the hyperplanes and the instances on different sides of the hyperplanes. *Artificial Neural Networks* (*ANN*) use input and output neurons together with hidden neurons to form a map between features and class variable (Figure 3.1).

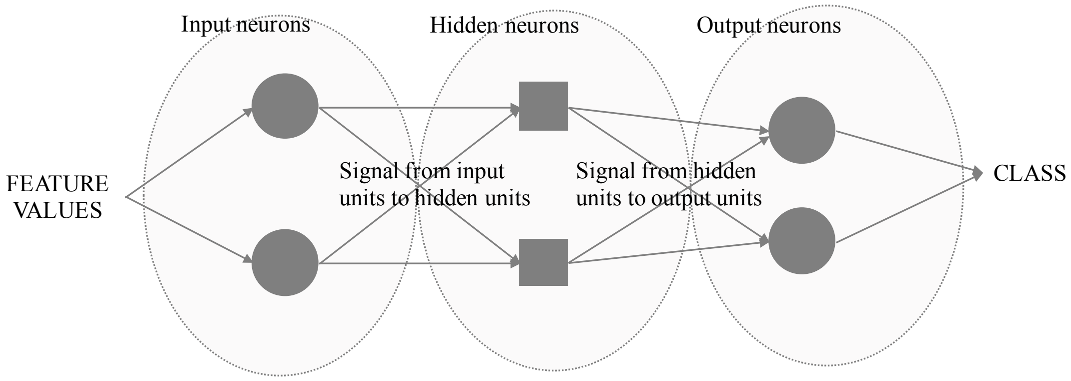
Input neurons
Hidden neurons
Output neurons
FEATURE VALUES
Signal from input units to hidden units
Signal from hidden units to output units
CLASS

Figure 3.1 Artificial Neural Networks (ANN)

The input neurons carry the activation function value of feature values. They send signals to hidden neurons and the collected signals are sent from hidden neurons to output neurons. The class values are determined based on the activation function values of output neurons.

For a review of supervised learning algorithms Kotsiantis (2007) can be referred to.

3.2 Multi-objective Approaches and Applications

Since feature selection is a combinatorial problem, it is popular to use evolutionary algorithms as search engines. In multi-objective feature selection problems it is typical to consider the number of selected features to be minimized and the classification performance of the corresponding subset to be maximized as two competing objectives.

We next provide the literature that aims to find Pareto optimal solutions for the objectives defined, by using different evolutionary algorithms and some of them focus on specific applications of feature selection problems.

Oliveira et al. (2002) use Non-dominated Sorting Genetic Algorithm (NSGA), which is suggested by Srinivas and Deb (1995), for feature selection in handwritten digit recognition in which the aim is to select the features that will contribute to characterize the expressions most. They define the objectives as minimizing number of features and maximizing accuracy. Once the non-dominated solutions are obtained, the one that has the minimum number of features with an accuracy level higher than a threshold value is chosen as the best solution.

Hamdani et al. (2007) also define the feature selection problem with two objectives minimizing number of features and maximizing accuracy. They suggest using Non-dominated Sorting Genetic Algorithm II (NSGA II), which is developed by Deb (2002), as search engine. Their results show that for simple problems, that is the problems with small number of features and small training sets, NSGA II is able to approximate the Pareto optimal solutions in a few iterations and even an exact convergence requires small number of iterations. When the training and test sets are relatively large compared to a simple problem, the computational time performance per iteration may drop but the number of iterations to convergence stays reasonable.

Huang et al. (2010) develop a modified version of NSGA II to feature selection for customer churn prediction, where the customers are classified as churn or non-churn. The authors state that in customer churn prediction, the classification performance should be evaluated in terms of three indicators: overall accuracy, sensitivity for churn class and sensitivity for non-churn class. They set their objectives as maximization of these three indicators and minimization of cardinality. They find non-dominated solution by NSGA II with a modification that population cannot include duplicated solutions. Then, they develop a method to find the best solution for each cardinality level among the non-dominated solutions.

Xue et el. (2013) investigate the performance of Particle Swarm Optimization (PSO), which is first developed by Kennedy and Eberhart (1995), for feature selection problem by comparing it with existing algorithms. They develop two algorithms using the framework of PSO; NSPSOFS and CMDPSOFS where the objectives are minimizing cardinality and maximizing accuracy. They compare these algorithms with two conventional methods, two single objective algorithms, and three multi-objective evolutionary algorithms in terms of non-dominated solutions obtained. They show that NSPSOFS and CMDPSOFS are able to find more and better solutions then conventional methods and single objective algorithms. In the final set of solutions obtained by NSPSOFS, some solutions dominate some of the solutions in the final solution set of multi-objective evolutionary algorithms, while the reverse is also possible. However, their results show that CMDPSOFS outperforms multi-objective algorithms in most of the experiments by achieving better feature subsets in terms of classification performance requiring less computational effort.

Most of the researchers work on the bi-objective (minimizing number of features and maximizing classification performance) version of the feature selection problem where typically the aim is to obtain Pareto optimal solutions. Karakaya et al. (2016) introduce the term “quasi equally informative subsets” into this problem. The idea is to find alternative subsets that have similar classification performances for each cardinality level. They propose two approaches, Wrapper for Quasi Equally Informative Subset Selection (WQEIS) and Filter for Quasi Equally Informative Subset Selection (FQEIS). They use Borg Multi-objective Evolutionary Algorithm, which is first proposed, by Hadka and Reed (2013) in both algorithms.

In recent years, cost-based feature selection methods have been developed, in which the subsets are evaluated in terms of costs associated with the features in the subset in addition to classification performance.

Bolón-Canedo et al. (2014) introduce a framework for cost-based feature selection. They suggest adding a term into evaluation function of a filter algorithm to represent the costs associated with the features. The tradeoff between the cost and classification performance is controlled by a parameter. Their approach simply converts the problem into a single objective problem, which is a linear combination of cost and performance indicator used in the filter approach with the tradeoff parameter.

Zhang et al. (2015) propose a multi-objective algorithm using PSO framework for cost-based feature selection problem where the objectives are minimizing cost and maximizing accuracy of feature subsets. While the earlier studies related with cost-based feature selection generally forms a single objective by combining cost and classification performance, this study aims to generate all non-dominated solutions for these two objectives.

CHAPTER 4

THE FEATURE SELECTION PROBLEM ADDRESSED

In this chapter, the feature selection problem addressed in this thesis is introduced. In Section 4.1 general formulation of the problem is given and in Section 4.2 the DM's objectives are explained in detail.

4.1 General Formulation of the Problem

We consider a DM who has four objectives: *accuracy*, *cardinality*, *cost,* and *risk* of the feature subset selected to use in the prediction model. The objectives will be discussed in detail in the following section.

The problem can be formulated in closed form as given below.

$$\text{"min"}\,\{z_1(\boldsymbol{x}),\ z_2(\boldsymbol{x}),\ z_3(\boldsymbol{x}),\ z_4(\boldsymbol{x})\}$$

s. to

$$\sum_{i=1}^{M} \mathrm{x_i} \geq 1 \tag{4.1}$$

$$\mathrm{x_i}\ \in \{0,1\}\ \forall\, i = 1, \dots, M$$

where x_i represents whether feature i is selected to construct the prediction model (value of 1) or not (value of 0), M is number of available features, $\mathrm{z_1}(\boldsymbol{x}), \mathrm{z_2}(\boldsymbol{x})$, $\mathrm{z_3}(\boldsymbol{x})$, and $\mathrm{z_4}(\boldsymbol{x})$ represent the accuracy, cardinality, cost, and risk objectives, respectively, of solution $\boldsymbol{x} = (x_1, \dots, x_M)$. Constraint (4.1) ensures the selected subset will include at least one feature.

Recall that the DM's preferences are assumed to be consistent with a monotone preference function denoted as $U_{DM}(\boldsymbol{z})$. Specifically, in the feature selection problem addressed, in our experiments we mostly consider a DM who would like to minimize weighted Chebyshev distance of a point from the ideal point in the objective space. We later briefly experiment with underlying quadratic preference functions as well. Since all objectives are scaled between 0 and 1, the ideal point can be defined as 0 for each objective. The underlying Chebyshev preference function can be formulated as in Equation (4.2).

$$U_{DM}(\boldsymbol{z}) = \max\{w_1 z_1, w_2 z_2, w_3 z_3, w_4 z_4\} \tag{4.2}$$

where $\boldsymbol{w} = (w_1, w_2, w_3, w_4)$ represents the objectives' weights of the DM in his/her preference function.

We note that the algorithms developed in this study are capable of handling any number of objectives. However, for the sake of completeness, in the rest of the paper we will address these four objectives only.

4.2 Objective Functions

Most of the studies that treat feature selection as a multi-objective problem focus on classification performance and cardinality. We use two more objectives; cost and risk.

Cost represents an objective that measures the difficulty of obtaining feature information. This may have a common aspect for a group of features (such as a fixed cost incurred that is necessary to obtain information on a group of features) and an individual part for each feature (such as a variable cost). Risk represents an attribute that is feature based. While each feature equally affects cardinality, the effect on risk could vary between features.

4.2.1 Accuracy

In feature selection problems it is typical to consider maximization of accuracy as one of the objectives. The accuracy of a certain subset is estimated by calling the learning algorithm. Recall the formula $Accuracy = \frac{tp+tn}{tp+fp+tn+fn}$ where tp, tn, fp, and fn stand for true positive, true negative, false positive, and false negative, respectively.

Accuracy is similar to rand index in statistics, which measures the similarity between two clusters of data. Generally, scaling of objective space is a critical issue in multi-objective optimization techniques. It is not possible to calculate the accuracy level at ideal and nadir points without considering all possible combinations of features. In the accuracy formula, the denominator is the sum of all testing observations and the numerator is sum of correctly classified instances. Therefore, theoretically the accuracy can take values between 0 and 1, so the other objectives are also scaled between 0 and 1. To be consistent with the other objectives, accuracy is also converted to a minimization type objective by simply subtracting its original value from 1.

The accuracy objective $z_1(\boldsymbol{x})$ is defined shown in Equation (4.3).

$$z_1(\boldsymbol{x}) = 1 - f(\boldsymbol{x}) \tag{4.3}$$

where $f(\boldsymbol{x})$ is the accuracy achieved for the selected features in solution $\boldsymbol{x}$.

4.2.2 Cardinality

As mentioned before, decreasing the cardinality of the subset used in the prediction model is favorable in terms of reducing storage requirements and improving the time efficiency. Therefore, minimizing the cardinality of selected subset is considered as another objective of the DM. In bi-objective feature selection problems this objective is widely used together with maximization of accuracy (Hamdani, 2007).

The maximum cardinality level is the number of features M, and without loss of generality it is assumed at least one feature is used in a solution. Linear scaling scheme is used to scale the real cardinality levels between 0 and 1, so the cardinality objective $z_2(\boldsymbol{x})$ is defined as shown in Equation (4.4).

$$z_2(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} x_i - 1}{M - 1} \tag{4.4}$$

4.2.3 Cost

In some classification problems, the features are grouped such that each group has a fixed investment cost and within the group each feature has a measuring cost. The medical diagnosis example mentioned before can be used to exemplify that structure. For example, it is possible to perform a blood test and MRI scan on the patient, and each of these two tests discloses different features regarding the patient. Both of the tests have fixed costs and each feature obtained from the tests has a variable cost. In that case, total screening cost of a subset is defined considering which test or tests are applied and which features are measured.

Let K be the total number of tests that can be applied, $k = 1, \dots, K$. To formulate the cost objective, following parameter definitions are made:

vc_i: variable cost of feature i

fc_k: fixed cost of test k

$t_{ik}: \begin{cases} 1, & \text{if feature } i \text{ belongs test } k \\ 0, & \text{otherwise} \end{cases}$

Note that each feature belongs to one test, that is $\sum_{k=1}^{K} t_{ik} = 1$ for $i = 1, \dots, M$.

In order to identify which tests should be applied to measure the selected features in solution $\boldsymbol{x}$, variable $v_k(\boldsymbol{x})$ is defined for each test $k = 1, \dots, K$ as in Equation (4.5).

$$v_k(\boldsymbol{x}) \geq t_{ik} x_i \quad \forall i = 1, \dots, M \tag{4.5}$$

Using those definitions, the total measuring cost, $TC(\boldsymbol{x})$, of a solution $\boldsymbol{x}$ is defined in Equation (4.6);

$$TC(\boldsymbol{x}) = \sum_{k=1}^{K} fc_k v_k + \sum_{i=1}^{M} vc_i x_i \tag{4.6}$$

In order to linearly scale total costs between 0 and 1, the feature subset with minimum total cost TC_{MIN} and maximum total cost TC_{MAX} should be identified as in Equations (4.7) and (4.8) respectively. The maximum total cost occurs simply when all features are used and the minimum total cost can be found by calculating the total costs of subsets with cardinality 1.

$$TC_{MIN} = \min_{i=1,\dots,M} \left\{ vc_i + \sum_{k=1}^{K} t_{ik} fc_k \right\} \tag{4.7}$$

$$TC_{MAX} = \sum_{k=1}^{K} fc_k + \sum_{i=1}^{M} vc_i \tag{4.8}$$

The resulting cost objective $z_3(\boldsymbol{x})$ is defined as shown in Equation (4.9).

$$z_3(\boldsymbol{x}) = \frac{TC(\boldsymbol{x}) - TC_{MIN}}{TC_{MAX} - TC_{MIN}} \tag{4.9}$$

4.2.4 Risk

In the risk objective, each feature is assigned a risk value and the risk of a subset is determined by the summation of the risk values of the selected features in the corresponding subset. Continuing with the medical diagnosis example, assume each feature has a health related risk for the patient and it is a concern for the DM to minimize the risks that the patient is exposed to.

To formulate the risk objective, the total risk involved in measuring the selected features in solution $\boldsymbol{x}$ is defined as in Equation (4.10):

$\mathrm{r_i}$: risks involved in measuring feature i

$$R(\boldsymbol{x}) = \sum_{i=1}^{M} r_i x_i \tag{4.10}$$

In order to linearly scale total risk between 0 and 1, the feature subset with minimum total risk R_{MIN} and maximum total risk R_{MAX} should be identified as in Equation (4.11) and (4.12). The maximum total risk occurs when all features are used, and the minimum total cost can be finding the feature with minimum risk.

$$R_{MIN} = \min_{i=1:M} \{\mathrm{r_i}\} \tag{4.11}$$

$$R_{MAX} = \sum_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{r}_{\mathrm{i}} \tag{4.12}$$

The risk objective $z_4(\boldsymbol{x})$ is defined as shown in Equation (4.13);

$$z_4(\boldsymbol{x}) = \frac{R(\boldsymbol{x}) - R_{MIN}}{R_{MAX} - R_{MIN}} \tag{4.13}$$

CHAPTER 5

ALGORITHMS

In this chapter, two different algorithms, iTDEA-fs (Interactive Territory Defining Evolutionary Algorithm for Feature Selection Problem) and iWREA-fs (Interactive Weight Reducing Evolutionary Algorithm for Feature Selection), are developed. The main framework used in both algorithms, the details of iTDEA-fs, the improvement issues related with iTDEA-fs, and the details of iWREA-fs are explained in the following sections.

5.1 Overview

The general framework of the algorithms is based on the framework of Interactive Territory Defining Evolutionary Algorithm (iTDEA) developed by Köksalan and Karahan (2010).

Being a preference-based multi-objective evolutionary algorithm, the main idea in iTDEA, is making a deeper search in the region that is estimated to be more appealing to the DM in order to approximate the most preferred solution better.

In order to identify the preferred region of the solution space, the algorithm is constructed in a way that the preference information is obtained progressively during the search process. That is, the algorithm allows the DM to indicate his/her preferences through interaction stages integrated into iterations and the search is guided towards preferred regions accordingly.

In iTDEA, to direct the search to the most preferred region, a territory defining approach is used. Generally speaking, non-dominated solutions are assigned with territory levels depending on their position in the solution space so that any other non-dominated solution cannot violate this territory. Eventually, this approach allows the algorithm to give a higher chance of surviving to the solutions that are more promising to be the most preferred solution.

In iTDEA, two populations are maintained through iterations; *regular population* and *archive*. In the initialization part, a regular population containing N random solutions are generated and the non-dominated solutions in that population are used to form the initial archive. Both the archive and regular population is updated, and used throughout the algorithm, however; the size of the regular population, N, is kept constant while there is no restriction on the size of archive.

Then, the number of iterations, T, and number of interaction stages, H, are determined. The interaction stages $h = 1, \dots, H$ are scheduled at iterations $G_1, \dots, G_H$, respectively; so that the DM is involved in the process after completing a certain number of regular iterations.

At each regular iteration, one offspring is generated by two selected parents. Then, it is decided whether the offspring will be accepted to the regular population and/or to the archive, and both the regular population and archive are updated accordingly. At each interaction stage, the DM is presented a set of solutions and asked to select the best solution among them. Based on the choice of the DM, the preference information is updated.

The algorithm stops when the maximum number of iterations, T, is reached and the final interaction with the DM is performed to find the most preferred solution. The general framework of the algorithm used in the algorithms iTDEA-fs and iWREA-fs, is given below.

1) Set iteration counter $t = 0$ and interaction counter $h = 0$. Schedule interaction stages at iterations $G_1, \dots, G_H$.
2) Generate initial regular population $P(0)$ of size N, and find the non-dominated solutions in the population to form initial archive $A(0)$.
3) Set $t \leftarrow t + 1$ and $h \leftarrow h + 1$. Set $P(t) = P(t-1)$ and $A(t) = A(t-1)$.

4) *Offspring Generation*: Select two parents, one from regular population and the other from archive, and apply crossover and mutation operators to create an offspring.
5) *Population Update*: Check whether the offspring satisfies the acceptance conditions to regular population. If it does not satisfy the conditions reject the offspring and go to step 7, otherwise insert it into $P(t)$.
6) *Archive Update*: Check whether the offspring satisfies the acceptance conditions to archive. If so insert it into $A(t)$, otherwise reject it.
7) If $t < G_h$, go to step 3.
8) *Interaction Stages*: Interact with the DM and update offspring acceptance conditions according to the preference information gathered.
9) If $t = T$, perform the final interaction and stop. Otherwise, go to step 3.

iTDEA-fs and iWREA-fs use the same framework defined above as well as the same *Offspring Generation* procedure. At each iteration t, two parents are selected to create an offspring. First parent is selected from regular population by tournament selection with tournament size of two and probability of 1. That is, two random solutions are chosen from the current population and it is checked whether one of the solutions dominates the other solution. If so, the solution dominating the other one is selected as the first parent. If there is no dominance between two solutions, one of them is selected randomly. The second parent is selected from current archive randomly.

In both evolutionary algorithms the chromosome representation is constructed such that each gene represents whether or not the corresponding feature is selected.

When the two parents are selected, uniform crossover is applied with a crossover probability of $p_c = 0.5$, and binary mutation is applied with a mutation probability of $p_m = 1/M$, where M is the total number of features on each gene to generate offspring $\boldsymbol{z}_{\text{off}}$ (Deb, 2001).

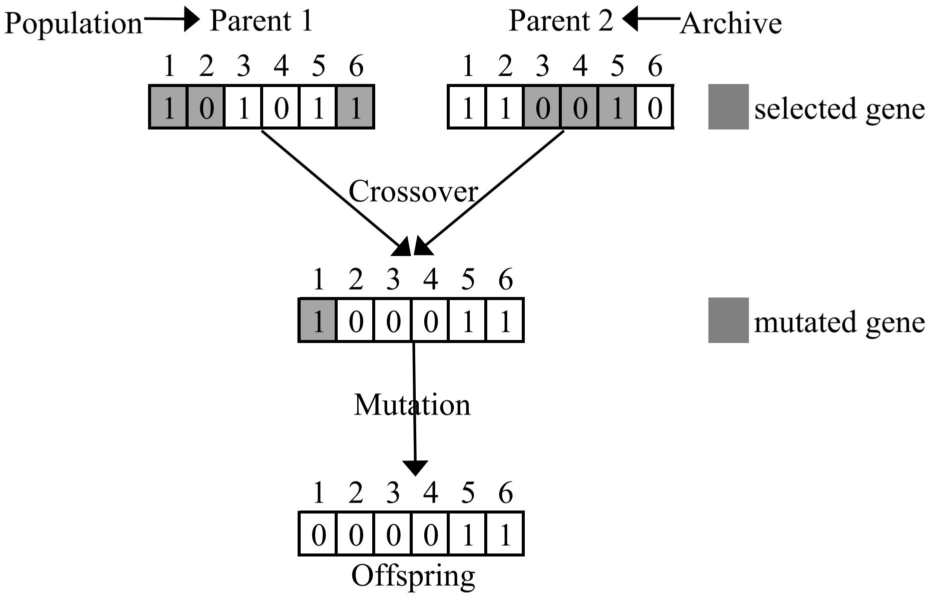

Figure 5.1 An example of crossover and mutation operations

Crossover and mutation procedure is exemplified in Figure 5.1. In this example there are 6 available features. Parent 1 consists of features 1, 3, 5, and 6 and Parent 2 includes features 1, 2, and 5. The generated offspring after applying crossover and mutation operations consists of features 5 and 6.

iTDEA-fs and iWREA-fs differ in the *Population Update*, *Archive Update*, and *Interaction Stages* that are explained for iTDEA-fs in Section 5.2 and for iWREA-fs in Section 5.4. Furthermore, the two algorithms include different parameter setting due to rules in their differing operations.

5.2 Interactive Territory Defining Evolutionary Algorithm for the Feature Selection Problem (iTDEA-fs)

In this study, iTDEA has been implemented on feature selection problem with some variations that are expected to be more compatible with the characteristics of the problem. The adopted version is called iTDEA-fs. In this section, the details of iTDEA-fs are explained.

5.2.1 Definitions

In archive update and interaction stage of iTDEA-fs, additional operations are required to calculate the objective weights that will minimize the Chebyshev distance of a solution from the ideal point, namely calculating *favorable weights*. Favorable weight vector $\widehat{\boldsymbol{w}}_i = (\widehat{w}_{i1}, \dots, \widehat{w}_{ip})$ of a solution $\boldsymbol{z_i}$, is calculated using the following formula;

Favorable weights formula

$$\widehat{w}_{ij} = \begin{cases} \dfrac{1}{z_{ij} - z_j^*}\left(\displaystyle\sum_{\mathrm{k}=1}^{\mathrm{p}} \frac{1}{z_{ik} - z_k^*}\right)^{-1} & \text{if } z_{ik} \neq z_k^* \text{ for all } k = 1, \dots p \\ 1 & \text{if } z_{ij} = z_j^* \\ 0 & \text{if } z_{ij} \neq z_j^* \text{ but } \exists k \\ & \text{such that all } z_{ik} = z_k^* \end{cases}$$

where $\boldsymbol{z}^*$ is the ideal point and p is the number of objectives.

Since all objectives are scaled between 0 and 1, as mentioned in Section 4.2, the ideal point can be defined as $\boldsymbol{z}^* = (0)_{1\times p}$.

At each interaction stage h, the preference information obtained from DM is used to estimate the *preferred weight region*, R^h. A preferred weight region is defined by a set of Chebyshev weight ranges $[\boldsymbol{l^h}, \boldsymbol{u^h}] = \{[l_1^h, u_1^h], \dots, [l_p^h, u_p^h]\}$ where l_i^h and u_i^h refers the lower and upper bound defined for preference function weight of objective i. Since there is no information regarding DM's preferences until first interaction stage, the initial preferred weight region R^0 includes all feasible weight ranges, $[l_j^0, u_j^0] = [0, 1]$ for all $j = 1, \dots p$.

5.2.2 iTDEA-fs Interaction Stages

At each interaction stage h of iTDEA-fs, P solutions are filtered from the current archive to present to the DM, and he/she is asked to choose the most preferred solution among them. The selected solution $\boldsymbol{z}_{fav}$, is used to estimate the preferred weight region R^h.

The preferred weight regions are used to direct the search by taking role in the archive update rules. As the algorithm progresses and more preference information is gathered, it is expected to converge to the preferred region. Therefore, the preferred weight region shrinks progressively with the help of reduction factor r around the favorable weights of the selected solution, $\mathbf{z}_{fav}$ in each interaction stage.

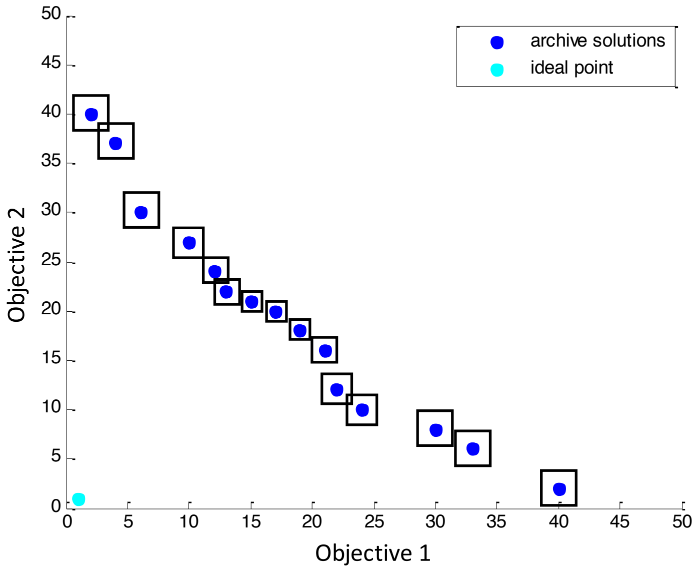

Figure 5.2 Smaller territory levels are assigned to the solutions in the preferred region in iTDEA-fs.

Each preferred weight region R^h has a territory level τ_h. The territory level assigned to the recently found region is smaller as exemplified in Figure 5.2 in order to allow more solutions in the preferred region. This is accomplished with the usage of territories in archive update. Generally speaking, an offspring's favorable weights determine in which preferred weight region it falls into. If it violates the territory of a solution within that region, then it is not accepted into archive. The archive update rules will be discussed in Section 5.2.4 in more detail.

The steps of an interaction stage h, are explained below.

(1) *Filtering*: Find the solutions $\mathbf{z}_i \in A(t)$ whose favorable weights fall into most currently estimated preferred weight region and form the set F. That is for all $\mathbf{z}_i \in A(t)$,

i. Calculate favorable weights $\widehat{\boldsymbol{w}}_{\mathrm{i}} = \{\widehat{w}_{i1}, \dots, \widehat{w}_{ip}\}$ of $\boldsymbol{z}_i$.

ii. Check if all $l_j^{h-1} \leq w_{ij} \leq u_j^{h-1}$ where $[l_j^{h-1}, u_j^{h-1}] \in R^{h-1}$ for all $j = 1, \dots, p$. If so, insert $\boldsymbol{z}_i$ into F.

If the number of solutions in F is more than P, select P of them randomly to present the DM. Otherwise, fill the remaining slots by the solutions in the $A(t)$ that are not presented to the DM before.

(2) Ask the DM to select $\boldsymbol{z}_{\mathrm{fav}}$, i.e, the most preferred solution of him/her in F.

(3) Estimate the weights of the DM's preference function, $\widehat{\boldsymbol{w}}_{\boldsymbol{DM}} = (\widehat{w}_1, \dots, \widehat{w}_p)$, with the favorable weights of the selected solution $\boldsymbol{z}_{\mathrm{fav}}$.

(4) Set the preferred weight region R^h defined by a set of Chebyshev weight ranges $[\boldsymbol{l^h}, \boldsymbol{u^h}] = \{[l_1^h, u_1^h], \dots, [l_p^h, u_p^h]\}$ as follows;

$$[l_j^h, u_j^h] = \begin{cases} [0, r^h] & \text{if } w_j^* - \dfrac{r^h}{2} \leq 0 \\ [1 - r^h, 1] & \text{if } w_j^* + \dfrac{r^h}{2} \geq 1 \\ \left[\widehat{w}_j - \dfrac{r^h}{2}, \widehat{w}_j + \dfrac{r^h}{2}\right] & \text{otherwise} \end{cases}$$

where $\widehat{w}_j \in \widehat{\boldsymbol{w}}_{\boldsymbol{DM}}$ and r is the reduction factor.

(5) Compute the territory level τ_h, and assign it to R^h using following formula.

$$\tau_h = \tau_H \left(\frac{\tau_0}{\tau_H}\right)^{\frac{H-h}{h}}$$

where τ_0 and τ_H represents initial and final territory level parameters, respectively and H is the total number of interaction stages.

The filtering procedure of iTDEA-fs differs from iTDEA. Since iTDEA originally implemented on the problems with continuous objective space, the number of non-dominated solutions found is usually enough to fill P slots, in fact they also check ε dominance relations and select non-dominated ones. However, since the objective space in the feature selection problem is discrete, sometimes P slots are not filled with the solutions whose favorable weights fall into currently estimated weight region, therefore in that case the remaining slots are filled with solutions that are not presented before.

5.2.3 iTDEA-fs Population Update

Each time an offspring $\boldsymbol{z}_{\text{off}}$ is generated, it is determined whether it will be accepted into the regular population. The steps of population update procedure at iteration t, are explained below:

(1) Check whether $\boldsymbol{z}_{\text{off}} \in P(t)$, if so do not accept the offspring into population, otherwise go to step 2.

(2) Set a counter $i = 1$.

(3) Test $\boldsymbol{z}_{\text{off}}$ against $\boldsymbol{z}_i \in P(t)$. If $\boldsymbol{z}_i$ is dominated by $\boldsymbol{z}_{\text{off}}$, discard $\boldsymbol{z}_i$, insert $\boldsymbol{z}_{\text{off}}$ into $P(t)$ and stop, otherwise set $i \leftarrow i + 1$ and go step 4.

(4) If $i \leq N$ go to step 3, otherwise choose a random solution $\boldsymbol{z}_k \in P(t)$ to discard and insert $\boldsymbol{z}_{\text{off}}$ into $P(t)$.

In iTDEA, a solution dominated by the population is not accepted, however, in the feature selection problem generating a non-dominated solution is more challenging. Thus, for the sake of divergence, the offspring that are not included in regular population are accepted even if they are dominated.

5.2.4 iTDEA-fs Archive Update

iTDEA-fs slightly differs in archive update rules from iTDEA. The steps followed to decide whether the offspring $\boldsymbol{z}_{\text{off}}$, will be accepted into archive is given below.

(1) Test $\boldsymbol{z}_{\text{off}}$ against each solution in the achieve, $\boldsymbol{z}_i \in A(t)$. If there exists $\boldsymbol{z}_i \in A(t)$ that dominates $\boldsymbol{z}_{\text{off}}$, reject it and stop, current archive remains same. Otherwise, go to step 2.

(2) Check whether there exists $\boldsymbol{z}_i \in A(t)$ that is dominated by $\boldsymbol{z}_{\text{off}}$. If there is, discard the dominated solutions from $A(t)$, insert $\boldsymbol{z}_{\text{off}}$ into $A(t)$ and stop. Otherwise, go to step 3.

(3) Calculate the favorable weights of the $\boldsymbol{z}_{\text{off}}$ as $\widehat{\boldsymbol{w}}_{\text{off}} = \left(\widehat{w}_1, \dots, \widehat{w}_p\right)$ to find the most recently estimated preferred weight region lower and upper bounds of which covers $\widehat{\boldsymbol{w}}_{\text{off}}$. That is,

    i. Set $q = h$.

ii. Check if all $l_j^q \leq \hat{w}_j \leq u_j^q$ where $[l_j^q, u_j^q] \in R^q$ for all $j = 1, \ldots, p$. If so, assign the territory level $\tau = \tau_q$ to the offspring and go to step 4. Otherwise, set $q = q - 1$, and repeat this step.

(4) Calculate the Chebyschev distances of $\mathbf{z}_i \in A(t)$ to $\mathbf{z}_{\text{off}}$ as d_i. Set $d = \min_i\{d_i\}$. If $d \leq \tau$ insert the $\mathbf{z}_{\text{off}}$ into $A(t)$, otherwise, reject it.

Unlike iTDEA, the offspring is accepted into archive if it is non-dominated and there exist at least one solution dominated by the offspring regardless of territories in iTDEA-fs.

The archive update procedure of this algorithm allows to keep more non-dominated solutions in the estimated preferred regions by comparing the territory level of the region that the offspring belongs to with the distance between the offspring and the closest solution in the archive as shown in Figure 5.3.

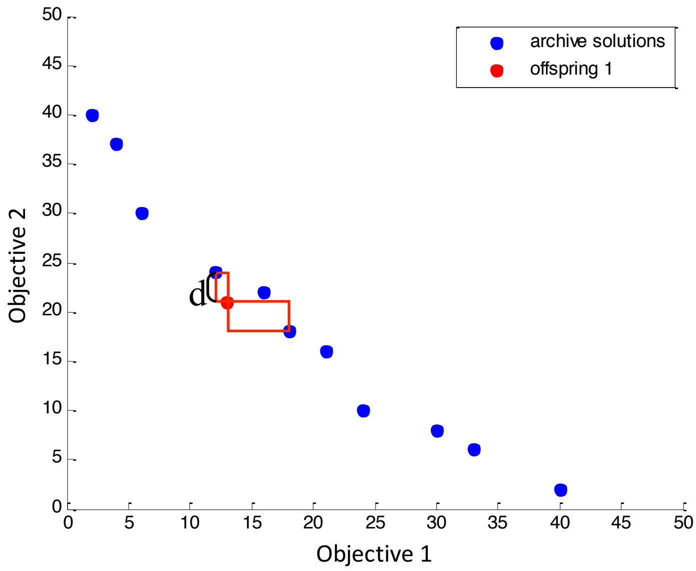

Figure 5.3 Archive update in iTDEA-fs: the offspring is accepted into archive if $d \leq \tau$.

5.3 Improvement Issues of iTDEA-fs

Feature selection problem has two characteristics originating from the nature of the problem that require a special treatment: *scaling* and *imbalanced solution space*.

As mentioned in Section 4.2, theoretically, accuracy can take any value between 0 and 1. However, in practice, depending on the dataset the minimum and maximum accuracy levels can form a much narrower interval. For example, consider a classification problem with two classes where 99% of the instances in the dataset belong to the first class, while only 1% of the instances are from the second class. In that case, even without constructing any relation between the features and the classes of the instance, estimating the class of all instances as the first class would result in a high accuracy level. Since the true interval depends on the dataset, applying a scaling procedure that will fit to every dataset can be challenging. Not scaling the objectives properly have a negative impact on the convergence of iTDEA-fs because the favorable weight calculation method highly relies on the assumption that the objectives are scaled consistently with each other.

In addition to the scaling issue, in the feature selection problem, discretized and imbalanced objective space may have a negative impact on estimating the objective weights of the DM. For example, for a feature selection problem with the number of features equal to 100, there are 100 distinct solutions with cardinality level of one while there is only one solution with cardinality level of 100. Moreover, all objectives have discrete values since accuracy is estimated on a certain size of testing set, and cardinality, cost, and risk depend on the features included in a subset.

Favorable weight calculation procedure in iTDEA-fs is performed for both estimating the preferred weight region from preferred solution in an interaction stage and identifying in which estimated weight region a solution is included. Therefore, this procedure is an important part of the algorithm but it may not perform well in case of imprecise scaling and imbalanced solution spaces. The reason of this claim is explained through an example.

Example: Consider a 3-objective minimization problem where the objectives are Z_1, Z_2, and Z_3. Suppose that during an interaction stage, when the archive is filtered to present to the DM, five solutions as given in Table 5.1 are obtained.

Let us assume that the DM has a preference function that minimizes the Chebyshev distance of a solution from the ideal point with objective weights w_1, w_2, and w_3.

Assume without loss of generality that the ideal point is zero at each objective. For a weight set of w_1, w_2, and w_3 equals to 0.1, 0.2, and 0.7 respectively, the DM would chose the solution which will minimize following function,

$$\max\{0.1\,Z_1\,,0.2\,Z_2\,,0.7\,Z_2\}$$

Table 5.1 Filtered archive in the example

Solution	Objective Value		
	Z_1	Z_2	Z_3
Y_1	0.10	0.40	0.40
Y_2	0.10	0.30	0.50
Y_3	0.20	0.30	0.45
Y_4	0.20	0.20	0.70
Y_5	0.30	0.10	0.70

For this filtered archive and preference function, the DM would select Y_1 as the most preferred solution. Using the favorable weight calculation method of iTDEA-fs, the DM's objective weights would be estimated as 0.66, 0.17, and 0.17, for w_1, w_2, and w_3, respectively.

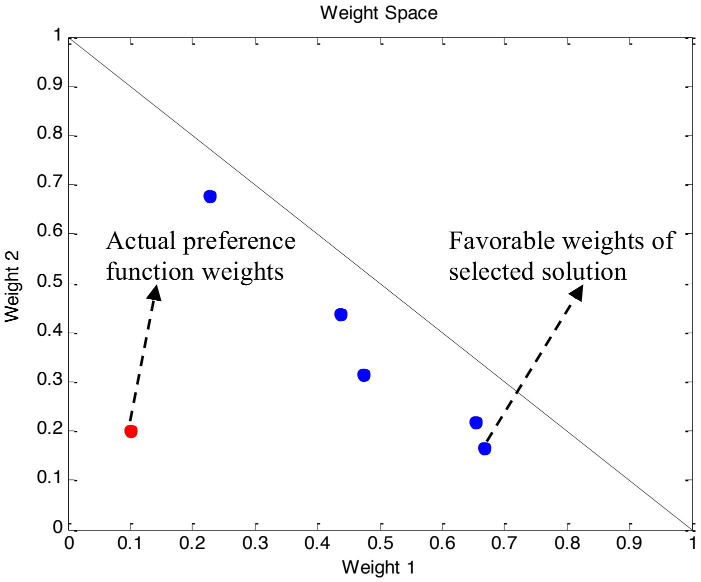

Figure 5.4 Favorable weights and actual weights in the example

Figure 5.4 represents the feasible weight space for the 3-objective problem mentioned in the example above. Favorable weights of the presented solutions to the DM and actual preference function weights are shown in the figure. It can be observed from Figure 5.4 that the favorable weights of the solutions presented to the DM tend to favor either first or second objectives, while the DM actually put more emphasis on the third objective. In iTDEA-fs, there can be two reasons for the filtered set presented to the DM consists of solutions whose favorable weights show a tendency of not to put emphasis on the third objective.

First, the preferred weight region is converged to a certain region of the weight space and the solutions in the filtered set are the ones whose favorable weights fall into this specific region. In that case, if the region shrinks on weights that represent the DM's actual weights well, the favorable weights of the solutions in the filtered set will be also representative and the algorithm will shrink the region around those weights even more, which is desirable. If the preferred weight region shrinks on weights that do not represent the DM's actual objective weights, the solutions in the archive that are potentially appealing for the DM may not be included in the filtered set as their favorable weights does not belong to estimated preferred weight region.

Second, the objective function values of non-dominated solutions in the archive may get squeezed within a narrow interval because of the imprecise scaling and/or imbalanced solution space. Especially, at early iterations of the algorithm it is possible that the solutions in the archive do not have balance in the objective space. In that case, the selected solution from the filtered set will have favorable weights that do not represent the actual weights well and the preferred weight region will shrink on those weights to be used in further iterations and interaction stages.

As in the example above, it may be difficult for iTDEA-fs to interpret the behavior of DM good enough for several reasons. To avoid such problems a mixed integer mathematical model, *Model (Mid$_\infty$),* suggested by Karakaya et al. (2016), is used to evaluate the DM's preferences in iWREA-fs. This model aims to find a weight set that will have a central location in the Chebyshev weight region that is feasible with respect to preferences of the DM. The details of the model are given below.

Assuming the ideal point is zero for each objective, let z_{ij} represents j^{th} objective value of solution $\boldsymbol{z}_i$ and L be the set of pairwise comparisons of the DM where $L = \{(z_m, z_l): z_m \text{ is preferred to } z_l\}$. Forming the sets $I^-_{z_m, z_l} = \{t: z_{mt} < z_{lt}\}$ and

$I^{+}_{z_m,z_l} = \{s: z_{ms} > z_{ls}\}$, and assigning a large positive value to M, the weight estimation model can be constructed as follows:

Model (Mid$_\infty$)

$$\max \varepsilon$$

s.to

$$\widehat{w}_t z_{lt} \geq \widehat{w}_s z_{ms} + \varepsilon - M\big(1 - y_t(z_m, z_l)\big),$$

$$\forall t \in I^{-}_{z_m,z_l}, \forall s \in I^{+}_{z_m,z_l}, \forall (z_m, z_l) \in L, \quad (5.1)$$

$$\sum_{t \in I^{-}_{z_m,z_l}} y_t(z_m, z_l) \geq 1, \qquad \forall (z_m, z_l) \in L, \quad (5.2)$$

$$\sum_{j=1}^{p} \widehat{w}_j = 1, \quad (5.3)$$

$$\widehat{w}_j \geq \varepsilon, \qquad \forall j, \quad (5.4)$$

$$y_t(z_m, z_l) \in \{0, 1\}, \qquad \forall t \in I^{-}_{z_m,z_l}, \qquad \forall (z_m, z_l) \in L.$$

where $\widehat{w}_j$ represents the estimated weight of j^{th} objective and $y_t(z_m, z_l)$ is a binary decision variable which equals to one if t^{th} weighted objective of z_l has the minimum difference with the maximum weighted objective of z_m among the objectives for which z_m is better than z_l.

In this model, constraint (5.1) ensures the weights will be feasible with respect to DM's preferences, however; together with objective function it also ensures that the minimum of the differences between the maximum of weighted objectives will be maximized. Constraint (5.2) stands for identifying the minimum of those differences for each preference. Constraint (5.3) normalizes the weights and constraint (5.4) helps to keep weights away from extremes on the feasible space.

Each preference of the DM restricts the feasible weight space and the optimal solution of the model $\widehat{\boldsymbol{w}} = \{\widehat{w}_1, \dots, \widehat{w}_p\}$ is the central weight vector in the feasible weight space.

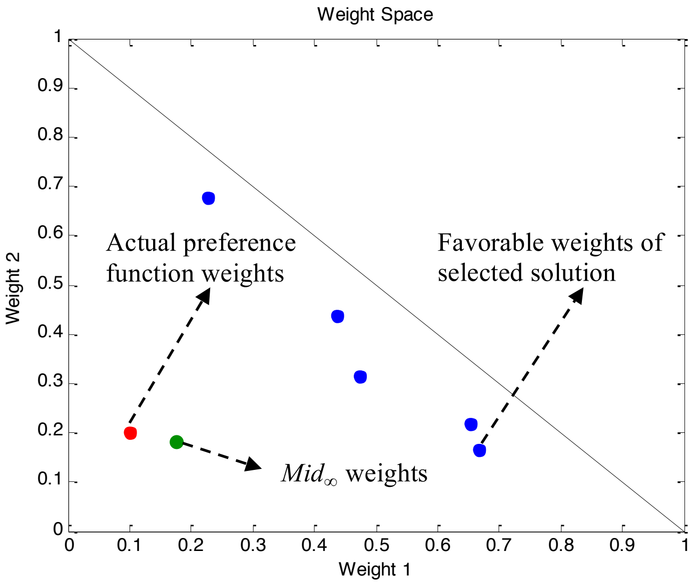

Figure 5.5 *Model (Mid$_\infty$)* weights, favorable weights and actual weights in the example

Consider the example mentioned above. If the DM selects Y_1, it refers to a preference list in which Y_1 is preferred to solutions Y_2, Y_3, Y_4 and Y_5. Using this preference list *Model (Mid$_\infty$)* estimates the DM's objective weights as 0.19, 0.19, and 0.62, w_1, w_2, and w_3, respectively. That is, with the same information *Model (Mid$_\infty$)* is able to estimate the true weights of the DM better by setting a relatively high estimated weight value to the 3rd objective as shown in Figure 5.5.

5.4 Interactive Weight Reducing Evolutionary Algorithm for Feature Selection (iWREA-fs)

In this section, a new interactive evolutionary algorithm that uses the general framework of iTDEA and improves iTDEA-fs in several dimensions is developed for feature selection problem. As mentioned in Section 5.2, in this algorithm an approach that is estimated to be more compatible with the characteristics of the problem is used.

5.4.1 iWREA-fs Interaction Stages

In iWREA-fs, at each interaction stage, the DM is asked to make Q pairwise comparisons, where each comparison made between the incumbent solution, $\boldsymbol{z}_{inc}$, and a selected solution to be compared with the incumbent, $\boldsymbol{z}_s$. The preferences of the DM are used as constraint to restrict the weight space in *Model (Mid*$_\infty$*)* introduced in Section 5.2. Before making any interactions with the DM the preference list is initialized as $L = \emptyset$. The steps of an interaction stage h, are explained below.

(1) Set the question counter $q = 0$.

(2) If $L = \emptyset$, select two random solutions, $\boldsymbol{z}_i, \boldsymbol{z}_k \in A(t)$ and $\boldsymbol{z}_i \neq \boldsymbol{z}_k$. Then, ask the DM to compare $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, and set $q \leftarrow q + 1$. Without loss of generality, assume that $\boldsymbol{z_i}$ is preferred to $\boldsymbol{z_j}$. Then, set $L = \{(\boldsymbol{z_i}, \boldsymbol{z_j})\}$ and the incumbent solution $\boldsymbol{z}_{inc} = \boldsymbol{z_i}$. Otherwise, go to step 3.

(3) Estimate the DM's preference function weights $\boldsymbol{\hat{w}_{DM}} = (\hat{w}_1, \ldots, \hat{w}_p)$, by solving *Model*(*Mid*$_\infty$) with the current preference list, L.

(4) Calculate the Chebyshev distances of the solutions in $A(t)$ to the ideal point, as follows:

$$\hat{u}(\boldsymbol{z}_i) = \max_j\{\hat{w}_j z_{ij} : \hat{w}_j \in \boldsymbol{\hat{w}_{DM}}\}$$

where z_{ij} represents j^{th} objective value of solution $\boldsymbol{z}_i \in A(t)$ and $\hat{w}_j$ represents the estimated weight of j^{th} objective.

Rank the solutions in increasing order of $\hat{u}(\boldsymbol{z_i})$ as $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{|A(t)|}\}$, where $|A(t)|$ represents the number of solutions in the current archive. Initialize rank counter $r = 1$.

(5) Check whether the comparison $(\boldsymbol{z}_{inc}, \boldsymbol{z}_r)$ is included in L, if it is not included set $\boldsymbol{z}_s = \boldsymbol{z}_r$ and go to step 8. Otherwise, set $r \leftarrow r + 1$. If $r \leq |A(t)|$ repeat step 5, if $r > |A(t)|$ go to step 6.

(6) Calculate the Chebyshev distances of the solutions in $P(t)$ to the ideal point, as follows:

$$\hat{u}(\boldsymbol{z}_i) = \max_j\{\hat{w}_j z_{ij} : \hat{w}_j \in \boldsymbol{\hat{w}_{DM}}\}$$

where z_{ij} represents j^{th} objective value of solution $\boldsymbol{z}_i \in P(t)$ and $\widehat{w}_j$ represents the estimated weight of j^{th} objective.

Rank the solutions in increasing order of $\hat{u}(\boldsymbol{z}_i)$ as $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_{|P(t)|}\}$, where $|P(t)|$ represents the number of solutions in the current regular population. Initialize rank counter $r = 1$.

(7) Check whether the comparison $(\boldsymbol{z}_{inc}, \boldsymbol{z}_r)$ is included in L, if it is not included set $\boldsymbol{z}_S = \boldsymbol{z}_r$ and go to step 8. Otherwise, set $r \leftarrow r + 1$ and repeat step 7.

(8) Ask the DM to make a pairwise comparison between the current incumbent solution $\boldsymbol{z}_{inc}$ and selected solution $\boldsymbol{z}_S$. Let $\boldsymbol{z_m}$ and $\boldsymbol{z_l}$ represent the preferred and non-preferred solutions, respectively. Set $\boldsymbol{z}_{inc} = \boldsymbol{z_m}$, update the preference list as $L = L \cup \{(\boldsymbol{z_m}, \boldsymbol{z_l})\}$. Set $q \leftarrow q + 1$. If $q = Q$ estimate the DM's preference function weights $\boldsymbol{\widehat{w}_{DM}}$ and stop by solving *Model (Mid*$_\infty$*)* with the current preference list, L, otherwise, go to step 3.

In the interaction stages of iWREA-fs, each pairwise comparison that the DM maker will make is aimed to be informative, in terms of reducing the feasible weight space as much as possible so that the objective weight of the DM can be estimated faster and more accurately and the most preferred solution can converge better. To do that after each comparison the incumbent solution is kept and the estimated weights are updated. The next solution to be compared with the incumbent solution is selected based on the updated weights. If the DM's objective weight can be estimated accurately, it is expected that the selected solution for the comparison and the incumbent solution have close preference function values and the preference of DM between these two solutions will be useful in terms of reducing the feasible weight space.

As mentioned before, in iTDEA-fs the DM is presented P solutions and asked to choose one of them, which actually refers asking the DM to make $P - 1$ pairwise comparisons. Therefore, to be consistent while comparing two algorithms, the number of questions Q, in each interaction stage of iWREA-fs is set to $P - 1$ in our computational experiments.

5.4.2 iWREA-fs Population Update

iWREA-fs uses regular population updating rules to direct the search towards appealing region of the solution space. When an offspring $\boldsymbol{z}_{\text{off}}$ is generated at iteration t, the regular population is updated with the procedure defined below.

(1) Calculate the Chebyshev distances of the offspring to the ideal point, as $\hat{u}(\boldsymbol{z}_{\text{off}}) = \max_j\{\hat{w}_j z_j : \hat{w}_j \in \hat{\boldsymbol{w}}_{\boldsymbol{DM}}\}$ where z_j represents j^{th} objective value of $\boldsymbol{z}_{\text{off}}$.

(2) Calculate the Chebyshev distances of the solutions in $P(t)$ to the ideal point, as follows:

$$\hat{u}(z_i) = \max_j\{\hat{w}_j z_{ij} : \hat{w}_j \in \hat{\boldsymbol{w}}_{\boldsymbol{DM}}\}$$

where z_{ij} represents j^{th} objective value of solution $\boldsymbol{z}_i \in P(t)$.

(3) Rank the solutions in increasing order of $\hat{u}(\boldsymbol{z}_i)$ as $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_{|P(t)|}\}$, where $|P(t)|$ represents the number of solutions in the current regular population.

(4) Compare $\hat{u}(\boldsymbol{z}_{\text{off}})$ and $\hat{u}(z_{|P(t)|})$, if $\hat{u}(\boldsymbol{z}_{\text{off}}) > \hat{u}(\boldsymbol{z}_{|P(t)|})$ do not accept the offspring into regular population. Otherwise, discard $\boldsymbol{z}_{|P(t)|}$ and accept the offspring into the regular population.

As in iTDEA, in WREA-fs it is allowed to keep dominated solutions in the regular population and its size is kept constant. However, while in iTDEA the search is directed by the archive with the acceptance rules, in iWREA-fs the search is directed by regular population. The offspring generated is compared with the worst solution in the regular population in terms of estimated preference function values and accepted if it's preference function value is lower. As a result, better solutions evolve in the population in terms of estimated preference function values throughout the iterations and if the weights are estimated well, this procedure allows to direct the search accurately.

5.4.3 iWREA-fs Archive Update

If the offspring $\boldsymbol{z}_{\text{off}}$, is accepted to the regular population at iteration t, in order to decide whether it will be accepted into the archive, $\boldsymbol{z}_{\text{off}}$ is tested against each

$\boldsymbol{z}_i \in A(t)$. If the offspring is dominated at least by one solution, it is rejected and current archive remains same. If the offspring is non-dominated, the solutions dominated by the offspring are discarded and the offspring is accepted into archive.

Having a discretized solution space, the number of non-dominated solutions in feature selection problem is small compared to continuous objective problems. Therefore, in iWREA-fs it is preferred to keep non-dominated solutions and use regular population to direct the search according to preferences of the DM.

CHAPTER 6

COMPUTATIONAL EXPERIMENTS

In this chapter, first, the datasets used to test the performances of the algorithms are introduced. Then, the parameter setting in the experiments is explained and lastly, computational results and their analysis are provided.

6.1 Datasets

The algorithms are implemented on four datasets from University of California (UCI) machine learning depository. Each dataset is designed for classification problems and none of them includes missing value in their observations. The number of features, number of classes and number of observations in each dataset are given in Table 6.1.

Heart Disease and *Breast Cancer* datasets are examples of classification problems in medical diagnosis area where the purpose is to identify the presence of diseases in the patients. In *Vehicle* dataset, the features extracted by processing vehicle's image are used to categorize the vehicle as Opel, Saab, Bus or Van. *German* dataset includes the data regarding the customers of a bank and the aim is to classify them as bad or good.

Table 6.1 Datasets used in experiments

Dataset	Number of features	Number of classes	Number of observations
Heart Disease	13	2	270
Vehicle	18	4	846
German	24	2	1000
Breast Cancer	32	2	569

In order to address the problem defined in Chapter 4, it is required to generate the parameters regarding cost and risk objectives for each dataset. In this study we consider the case where the objectives of the DM conflict with each other. This is generally the case in real life situation where a solution that performs well in one objective performs worse in another objective. To reflect such a conflict in our objectives, we assume that the cost and risk objectives are inversely proportional to accuracy in different ways. We measure the accuracy level of each feature individually and use these values to generate inversely proportional cost and risk parameters for the corresponding feature.

To exemplify the procedure of cost and risk parameters' generation, consider a dataset with 5 features. Recall that the DM aims to maximize accuracy whereas to minimize the risk and the cost of the selected subset. Suppose that in terms of accuracy features 1 and 2 perform well, feature 3 perform moderately, and features 4 and 5 perform poorly. The features that have close performances are grouped together and fixed costs of the group are generated directly proportional to their accuracy levels. That is, fixed costs of the groups comprising features 1 and 2, feature 3, and features 4 and 5 are high, moderate, and low, respectively. The variable costs in a group are generated randomly between an interval proportional to accuracy level, e.g. random between 80% and 120% of accuracy level. The individual risk levels are generated randomly from an interval that is proportional to the variable costs of the features.

Heart Disease dataset in UCI repository includes the fixed and variable cost information of the features and these original values are used as cost parameters in our experiments.

6.2 Implementation

To estimate the accuracy level of a subset of features, a single hidden layer feedforward neural network Extreme Learning Machine (ELM), which is suggested by Huang et al. (2012), is used as the learning algorithm. ELM achieves comparably high classification performances with a high training speed when compared with gradient-based methods, traditional SVM, and least square SVM. In our experiments, we use the suggested parameter setting in Huang et al. (2006).

We use 10-fold cross validation to determine the training and test sets, and repeat this procedure 5 times in order to reduce variation in accuracy level estimation caused by the random nature of ELM.

To be able to observe the effect of employing the DM's preferences, a version, namely *No Interaction*, in which the number of interaction stages is set to zero, is also tested on each dataset in addition to iTDEA-fs and iWREA-fs.

Recall that the Chebyshev preference function of DM, $U_{DM}(\boldsymbol{z})$, is formulated in section 4.1 as follows:

$$U_{DM}(\boldsymbol{z}) = \max\{w_1 z_1, w_2 z_2, w_3 z_3, w_4 z_4\}$$

where z_1, z_2, z_3 and z_4 refer to accuracy, cardinality, cost and risk objectives of solution $\boldsymbol{z}$, respectively, and $\boldsymbol{w} = (w_1, w_2, w_3, w_4)$ represents the objective weight vector of the DM.

We use different weight vectors to simulate the preferences of the DM so that different sets of solutions are favored by the DM for different weight sets. We refer to these weights sets as: *Accuracy Favored*, *Accuracy and Cost Tradeoff*, *Equally Treated*. In Table 6.2, the objective weights in $U_{DM}(\boldsymbol{z})$ for each type of DM are given as $\boldsymbol{w} = (w_1, w_2, w_3, w_4)$ where w_1, w_2, w_3, and w_4 refer to the weights of accuracy, cardinality, cost, and risk objectives, respectively.

Table 6.2 Types of DM's preference function weights tested

Test Name	Weight set
Accuracy Favored	(0.97, 0.01, 0.01, 0.01)
Accuracy and Cost Tradeoff	(0.40, 0.10, 0.40, 0.10)
Equally Treated	(0.25, 0.25, 0.25, 0.25)

6.3 Experimental Setting

The algorithms are tested on each dataset for different types of underlying preference functions of the DM. In Tables 6.3-6.6, the evolutionary algorithms' parameter settings are given for each experiment.

Within an experimental setting, iTDEA-fs, iWREA-fs and No Interaction shares the same parameter setting for the population size, N, and number of iterations, T, and number of interaction stages, H, is also set same for iTDEA-fs and iWREA-fs. The interactions with DM scheduled in equal intervals for iTDEA-fs and iWREA-fs in each experiment. That is, $G(h) = \left(\frac{T}{H}\right)h$ for $h = 1, \dots, H$. It is also possible to set an adaptive scheduling procedure for interactions such that the DM is interacted whenever solutions that are estimated to be favorable for the DM are obtained. We do not apply an adaptive scheduling procedure to be able to make a fair comparison between the algorithms. The number of comparisons, Q, given in Tables 6.3-6.6 refers the number of questions asked to the DM in iWREA-fs at each interaction stage. The number of solutions presented to the DM in the interactions stages of iTDEA-fs, P, is set to $P = Q + 1$. Together with this setting, in an experiment iTDEA-fs and iWREA-fs employ same amount of DM interaction.

The population size and the number of questions in the experiments of a dataset are kept constant, however, the number of iterations and number of interaction stages are determined such that the search is stopped at a point in which the algorithms can be compared. Accuracy Favored weight set is generally more challenging in terms of convergence, thus the number of iterations and interaction stages are set higher for its experiments.

In addition to those parameters, iTDEA-fs requires to set initial and final territory levels, τ_0 and τ_H, and reduction factor, r. Based on our preliminary experiments, we used $\tau_0 = 0.1$, $\tau_H = 0.0001$ and $r = (1/p)^H$ where p is number of objectives in the experiments.

In general, as the solution space enlarges, that is as the number of features increases, to converge the most preferred solution of the DM the number of iterations, number of interactions and number of questions asked to the DM are increased, as we did in our experimental settings. Different settings can be employed such as an adaptive interaction schedule based on the progress of non-dominated solutions. However, to make a fair comparison of the algorithms we do not consider such procedures and keep our original settings.

Table 6.3 Parameter settings of Heart Disease (13) experiments

	Weight Set		
Parameter	Accuracy Favored	Accuracy and Cost Tradeoff	Equally Treated
Population size, *N*	50	50	50
Number of iterations, *T*	600	200	200
Number of interactions, *H*	6	4	4
Number of comparisons, *Q*	3	3	3

Table 6.4 Parameter settings of Vehicle (18) experiments

	Weight Set		
Parameter	Accuracy Favored	Accuracy and Cost Tradeoff	Equally Treated
Population size, *N*	200	200	200
Number of iterations, *T*	10,000	10,000	10,000
Number of interactions, *H*	10	10	10
Number of comparisons, *Q*	3	3	3

Table 6.5 Parameter settings of German (24) experiments

	Weight Set		
Parameter	Accuracy Favored	Accuracy and Cost Tradeoff	Equally Treated
Population size, *N*	500	500	500
Number of iterations, *T*	20,000	6,000	6,000
Number of interactions, *H*	10	3	3
Number of comparisons, *Q*	5	5	5

Table 6.6 Parameter settings of Breast Cancer (32) experiments

	Weight Set		
Parameter	Accuracy Favored	Accuracy and Cost Tradeoff	Equally Treated
Population size, *N*	1,000	1,000	1,000
Number of iterations, *T*	20,000	10,000	10,000
Number of interactions, *H*	20	10	10
Number of comparisons, *Q*	5	5	5

6.4 Results and Discussion

Three algorithms are tested on each experimental setting with 10 replications. The algorithms are compared based on a performance indicator (defined in the next section) and their computational efficiency.

6.4.1 Performance Indicator

The performance of algorithms on finding an appealing solution for DM in an experiment can be evaluated based on the best solution in the final archive $U^*(T) = \min_{z_i \in A(T)}\{U_{DM}(z_i)\}$. Although during the search process the underlying preference function of the DM is unknown to us, we use this simulated underlying preference function to calculate the performance indicator. Let $U^r_{\text{iTDEA-fs}}$, $U^r_{\text{iWREA-fs}}$, and $U^r_{\text{No Interaction}}$ represent $U^*(T)$ values obtained in replication r of an experimental setting by iTDEA-fs, iWREA-fs, and No Interaction, respectively. The values of $U^r_{\text{iTDEA-fs}}$, $U^r_{\text{iWREA-fs}}$, and $U^r_{\text{No Interaction}}$ are given in Appendix A for each experimental setting.

Since in the feature selection problem addressed it is not possible to find the nadir and ideal point without total enumeration of possible subsets, in order to define a normalized performance indicator, for each experimental setting, the best and worst performance obtained by algorithms through 10 replications are found as shown in Equations (6.1) and (6.2).

$$U_{MAX} = \max_{r=1,\dots,10}\{\max\{U^r_{\text{iTDEA-fs}}, U^r_{\text{iWREA-fs}}, U^r_{\text{No Interaction}}\}\} \tag{6.1}$$

$$U_{MIN} = \min_{r=1,\dots,10}\{\min\{U^r_{\text{iTDEA-fs}}, U^r_{\text{iWREA-fs}}, U^r_{\text{No Interaction}}\}\} \tag{6.2}$$

By using U_{MAX} and U_{MIN}, the performance of algorithms in a replication is evaluated as percentage deviations, which are defined as given in Equations (6.3), (6.4) and (6.5).

$$\Delta^r_{\text{iTDEA-fs}} = \frac{U^r_{\text{iTDEA-fs}} - U_{MIN}}{U_{MAX} - U_{MIN}} \tag{6.3}$$

$$\Delta^{r}_{\text{iWREA-fs}} = \frac{U^{r}_{\text{iWREA-fs}} - U_{MIN}}{U_{MAX} - U_{MIN}} \quad (6.4)$$

$$\Delta^{r}_{\text{No Interaction}} = \frac{U^{r}_{\text{No Interaction}} - U_{MIN}}{U_{MAX} - U_{MIN}} \quad (6.5)$$

6.4.2 Evaluating Algorithms

The mean and standard deviation of the percentage deviations of each algorithm from the minimum value on each experimental setting are given in Tables 6.7-6.10. The mean of percentage deviations is zero in some experimental settings, which indicates that the algorithm found the best solution of the three algorithms in 10 runs, U_{MIN}, in all replications. Those types of results are bold-faced in Tables 6.7-6.10, and it is observed that in some experiments iWREA-fs is able to converge the best solution found in all of the replications.

One sample t test is applied on the paired differences of percentage deviations: $(\Delta^{r}_{\text{iWREA-fs}} - \Delta^{r}_{\text{iTDEA-fs}})$, $(\Delta^{r}_{\text{iWREA-fs}} - \Delta^{r}_{\text{No Interaction}})$ and $(\Delta^{r}_{\text{iTDEA-fs}} - \Delta^{r}_{\text{No Interaction}})$ in order to identify whether there exists statistically significant difference between means and 95% confidence intervals are computed as given in Tables 6.11-6.14.

In Tables 6.11-6.14, the results in which there is statistically significant difference between the algorithms are bold-faced. The results indicate that iWREA-fs performs better than iTDEA-fs and No Interaction in many cases. Based on our preliminary experiments, it is known that Vehicle and Breast Cancer datasets and Accuracy Favored weight set are comparably more challenging in terms of convergence than other settings since the relevance and redundancy relations between the features are more complicated. iWREA-fs's performance is more apparent in those cases. On the other hand, according to Tables 6.11-6.14, there is no statistical difference between iTDEA-fs and No Interaction in none of the experimental settings, which will be discussed later in this section in detail.

Table 6.7 Mean and standard deviation of the percentage deviations for Heart Disease (13) experiments

Weight Set		Algorithm iTDEA-fs	 iWREA-fs	 No Interaction
Accuracy Favored	Mean	0.3887	0.0402	0.0919
	Std. Dev.	1.2051	0.3810	0.4437
Accuracy and Cost Tradeoff	Mean	0.2086	**0.0000**	0.1756
	Std. Dev.	1.0185	0.0000	1.1241
Equally Treated	Mean	0.1543	0.0241	0.1497
	Std. Dev.	1.0279	0.2289	0.7951

Table 6.8 Mean and standard deviation of the percentage deviations for Vehicle (18) experiments

Weight Set		Algorithm iTDEA-fs	 iWREA-fs	 No Interaction
Accuracy Favored	Mean	0.6670	0.1523	0.7522
	Std. Dev.	1.0437	0.7501	1.0092
Accuracy and Cost Tradeoff	Mean	0.5263	**0.0000**	0.5861
	Std. Dev.	1.2447	0.0000	0.8261
Equally Treated	Mean	0.8000	0.4000	0.8000
	Std. Dev.	1.2649	1.5492	1.2649

Table 6.9 Mean and standard deviation of the percentage deviations for German (24) Experiments

Weight Set		Algorithm iTDEA-fs	 iWREA-fs	 No Interaction
Accuracy Favored	Mean	0.7166	0.2811	0.6716
	Std. Dev.	0.6888	0.7127	0.9189
Accuracy and Cost Tradeoff	Mean	0.1176	0.0117	0.0176
	Std. Dev.	0.9339	0.0740	0.0848
Equally Treated	Mean	0.2106	0.0409	0.2162
	Std. Dev.	1.2521	0.3263	1.0046

Table 6.10 Mean and standard deviation of the percentage deviations for Breast Cancer (32) experiments

Weight Set		Algorithm iTDEA-fs	 iWREA-fs	 No Interaction
Accuracy Favored	Mean	0.4436	0.0449	0.6640
	Std. Dev.	0.9839	0.4258	0.9090
Accuracy and Cost Tradeoff	Mean	0.3688	**0.0000**	0.4233
	Std. Dev.	1.2026	0.0000	1.0990
Equally Treated	Mean	0.1996	**0.0000**	0.0195
	Std. Dev.	1.0356	0.0000	0.0942

Table 6.11 95% confidence intervals on paired differences of percentage deviations for Heart Disease (13) experiments

Weight Set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Accuracy Favored	**(-0.61, -0.09)**	(-0.21, 0.10)	(-0.04, 0.63)
Accuracy and Cost Tradeoff	(-0.45, 0.03)	(-0.44, 0.09)	(-0.38, 0.45)
Equally Treated	(-0.33, 0.07)	(-0.29, 0.04)	(-0.20, 0.21)

Table 6.12 95% confidence intervals on paired differences of percentage deviations for Vehicle (18) experiments

Weight Set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Accuracy Favored	**(-0.88, -0.15)**	**(-0.84, -0.36)**	(-0.36, 0.19)
Accuracy and Cost Tradeoff	**(-0.82, -0.23)**	**(-0.78, -0.39)**	(-0.33, 0.21)
Equally Treated	(-0.90, 0.10)	**(-0.77, -0.03)**	(-0.34, 0.34)

Table 6.13 95% confidence intervals on paired differences of percentage deviations for German (24) experiments

Weight Set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Accuracy Favored	**(-0.62, -0.25)**	**(-0.73, -0.05)**	(-0.24, 0.33)
Accuracy and Cost Tradeoff	(-0.33, 0.12)	(-0.02, 0.01)	(-0.13, 0.33)
Equally Treated	(-0.41, 0.07)	(-0.38, 0.03)	(-0.13, 0.12)

Table 6.14 95% confidence intervals on paired differences of percentage deviations for Breast Cancer (32) experiments

Weight Set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Accuracy Favored	**(-0.64, -0.15)**	**(-0.92, -0.32)**	(-0.55, 0.11)
Accuracy and Cost Tradeoff	**(-0.66, -0.08)**	**(-0.69, -0.16)**	(-0.36, 0.25)
Equally Treated	(-0.45, 0.05)	**(-0.04, -0.01)**	(-0.07, 0.43)

The deviations are used to compare the algorithms in their performance to converge to the best solution found in all replications for a given number of iterations. In order to evaluate the convergence speed of algorithms in detail, the deviations in each replication can be investigated. The progress of the best solution in the archive through iterations, $U^*(t) = \min_{z_i \in A(t)}\{U_{DM}(z_i)\}$, for 10 replications of experiments on Breast Cancer dataset with Accuracy Cost Tradeoff weight set are shown in Figures 6.1, 6.2, and 6.3 for iTDEA-fs, iWREA-fs, and No Interaction, respectively. As it can be inferred from those figures that iWREA-fs is converging better and faster to the best solution found by three algorithms in 10 replications. The progress of the best solution in the archive through iterations, namely archive progress, is given for each experimental setting in Appendix B as figures.

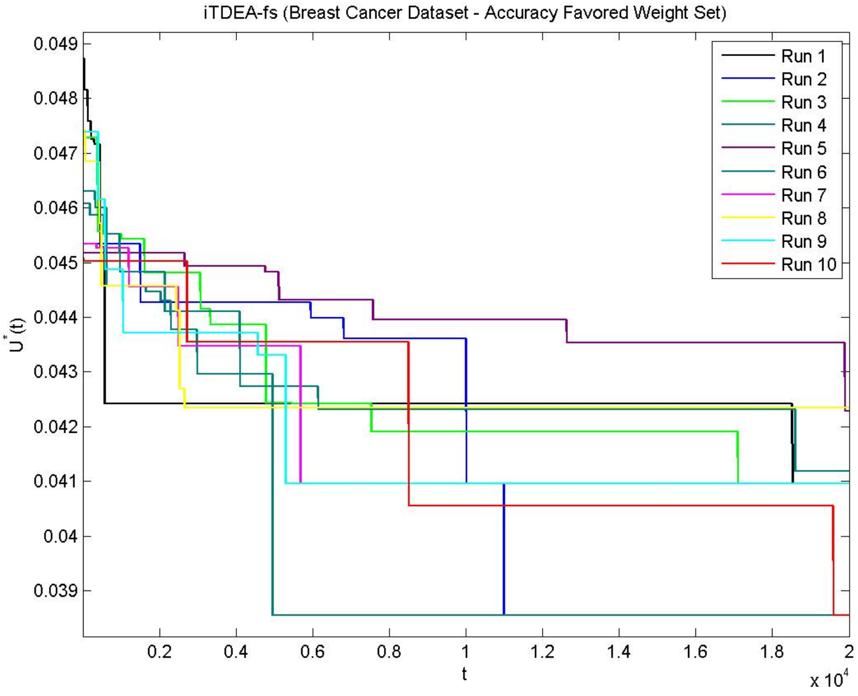

Figure 6.1 Archive progress of iTDEA-fs on Breast Cancer dataset experiments with Accuracy Favored weight set

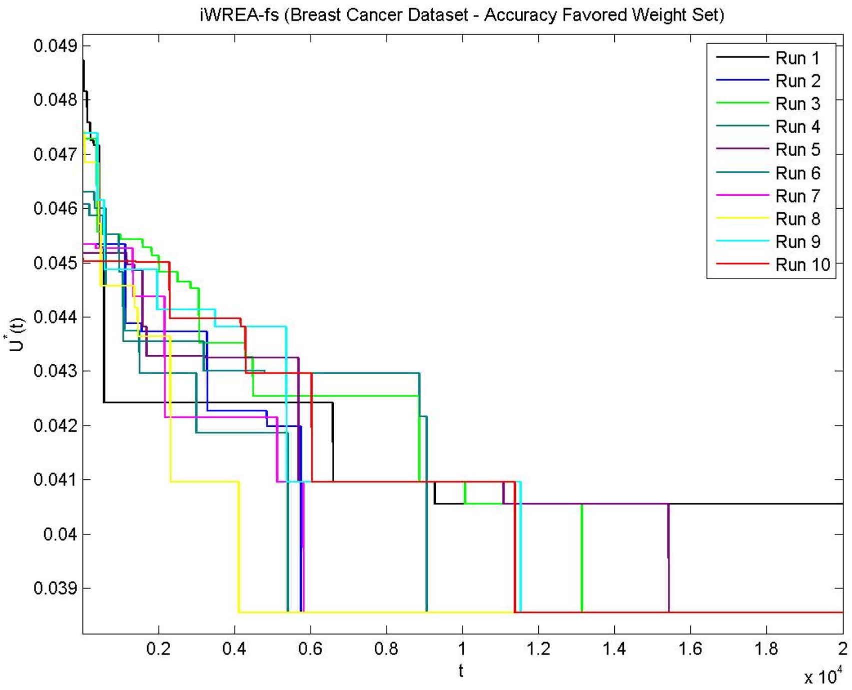

Figure 6.2 Archive progress of iWREA-fs on Breast Cancer dataset experiments with Accuracy Favored weight set

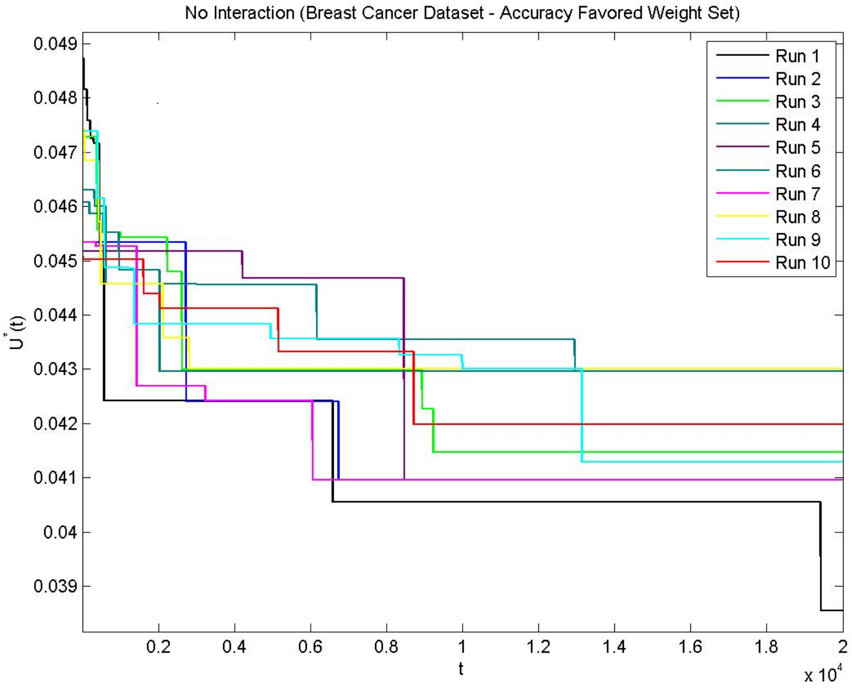

Figure 6.3 Archive progress of No Interaction on Breast Cancer dataset experiments with Accuracy Favored weight set t

Although it is expected that the information gathered from the DM will be useful to find appealing solutions for the DM, one of the observation that can be inferred from the confidence intervals given in Tables 6.11-6.14 is that there is no statistical difference between the performances of iTDEA-fs and No Interaction. In order to explain the reason, one of the replications in which iTDEA-fs does not perform as well as No Interaction is investigated.

In Figure 6.4, the progress of best solution for DM in the archive through iterations, $U^*(t)$, is shown for the 5th replication of experiments on Breast Cancer dataset with Accuracy Cost Tradeoff weight set. Additionally, the preference function value of the selected solutions of iTDEA-fs's and incumbent solutions of iWREA-fs's at interaction stages are represented in the same figure.

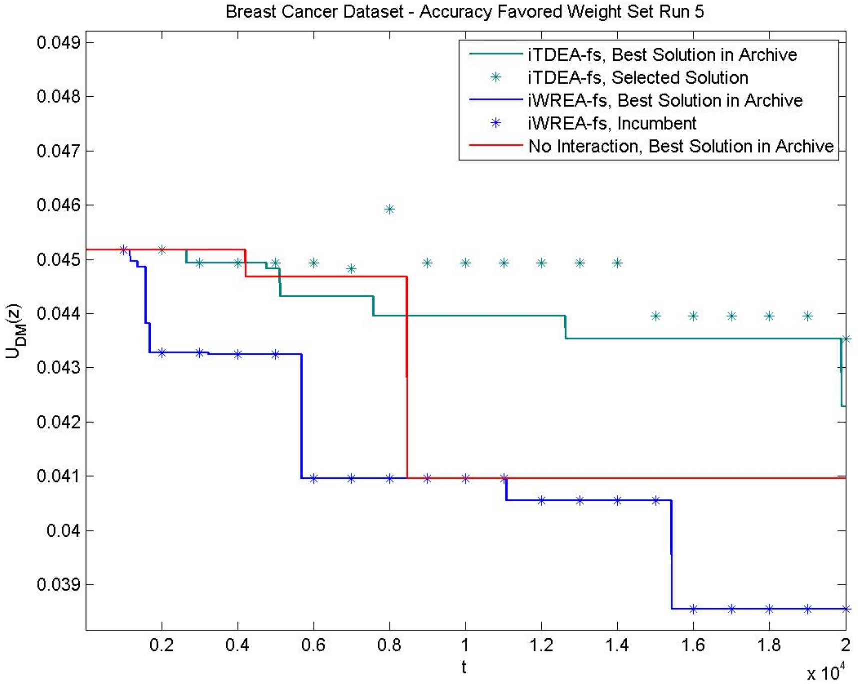

Figure 6.4 5th replication on Breast Cancer dataset experiments with Accuracy Favored weight set

As it can be observed from Figure 6.4, the selected solution is not the same with the best solution of the archive after the fourth interaction stage of iTDEA-fs. This is only possible if the best solution is not included in the set of solutions presented to the DM. Recall that the filtered set in iTDEA-fs includes the solutions whose favorable weights fall into the most recently estimated preferred weight region. Even though in the first three interaction stages the best solution in the archive is presented to the DM, the preferred weight region is not shrunk on objective weights that represent the DM's preference function well. Hence, in the later interaction stages the favorable weights of the best solution in the archive do not belong to the estimated preferred weight region and the search is not directed towards the appealing region of solution space for the DM.

On the other hand, the incumbent solution in iWREA-fs is same with the best solution in the archive in most of the interaction stages, which indicates that the DM's objective weights are represented well with the estimated weights throughout the algorithm. In addition to its benefit in directing search accurately, this property of iWREA-fs enables to identify best solution found without an additional interaction. By investigating plots provided in supplementary material that has the same form with Figure 6.4 for all individual runs, it can be observed that the conflict with the selected solution and best solution in iTDEA-fs is valid in most of the replications, while in iWREA-fs generally the best solution and incumbent overlaps.

6.4.3 Comparison of Computational Efforts

As mentioned before, ELM has randomness in its nature. Therefore, in order to compare the algorithms in terms of convergence performance precisely, accuracy level of one feature subset found in a replication is used in other replications without calling ELM again. As a result, it would not be fair to compare the algorithms in terms of computational effort with the original experiments.

In order to compare the computational efforts, accuracy objective is defined as a simple function and experiments regarding Accuracy Favored weight set are conducted with that modification for 10 replications.

The algorithms are coded on MATLAB R2014b, and implemented on a computer with Intel(R)Core(TM)i7-4770S CPU @ 3.10 GHz, 16 GB RAM and Windows 7.

The average CPU times for the implementation of algorithms on each dataset in Table 6.15. The individual CPU time of runs are provided in Appendix C.

Table 6.15 CPU times of algorithms (in seconds)

	Algorithm		
Dataset	iTDEA-fs	iWREA-fs	No Interaction
Heart Disease	0.38	1.55	0.38
Vehicle	16.57	7.40	16.52
German	71.83	18.06	72.65
Breast Cancer	144.30	33.97	146.90

In the interaction stages of iWREA-fs after each question asked to the DM, *Model* (Mid_{∞}) is solved which is a mixed integer programming, while the favorable weight calculation procedure in iTDEA-fs is a simple algebraic function. However, at each iteration in order to update the regular population, the dominance relation between the offspring and population members is checked in iTDEA-fs and No Interaction, while in iWREA-fs after the first interaction estimated preference function value of the offspring is compared with the maximum of estimated preference function values of population members only. In Heart Disease dataset experiments for which the interaction stages are set more frequently, iWREA-fs requires higher computational effort. However, as the frequency of interaction stages decreases the efficiency of population update rules in iWREA-fs shows its effect, therefore the average CPU time of iWREA-fs is smaller than iTDEA-fs and No Interaction for the experiments of Vehicle, German and Breast Cancer datasets.

6.4.4 Features Selected

In feature selection problems it is a concern to identify the features that contribute to classification performance most. Therefore, we looked at the similarity of the features of the subsets that have high preference function values for the accuracy-favored preference function casein this section. We selected the accuracy-favored case since it proved to be a difficult case in the experiments. Heart Disease dataset is used for that purpose since total enumeration of solutions is possible for this dataset.

When the DM's objective weights are Accuracy Favored, it is expected that the subsets of features with high accuracy levels will be favored according to the underlying preference function. Among all possible feature subsets of Heart Disease dataset, the best solution in terms of preference function value with Accuracy Favored weight set includes only features 3, 12 and 13. In Table 6.16, the existence frequencies of features in the subsets ranked in top 5 percent by their respective preference function values with Accuracy Favored weight set are given. As it can be inferred from these results, most frequent features are 3rd, 12th and 13th, which are consistent with the features included in the best solution, as expected.

Table 6.16 Existence frequencies of features in the subsets ranked in top 5 percent

Feature	Frequency
1	0.3634
2	0.4829
3	**0.7634**
4	0.3878
5	0.3512
6	0.3195
7	0.3585
8	0.4415
9	0.4366
10	0.5195
11	0.4000
12	**0.8415**
13	**0.9171**

6.4.5 Some Results for Quadratic Underlying Preference Functions

In order to demonstrate the performance of algorithms for a different form of DM's underlying preference function, we repeated the experiments on Vehicle dataset for a quadratic preference function to be minimized as formulated in Equation 6.6.

$$U_{DM}(\boldsymbol{z}) = (w_1 z_1)^2 + (w_2 z_2)^2 + (w_3 z_3)^2 + (w_4 z_4)^2 \qquad (6.6)$$

where z_1, z_2, z_3 and z_4 refer to (modified) accuracy, cardinality, cost and risk objectives of solution $\boldsymbol{z}$, respectively, and $\boldsymbol{w} = (w_1, w_2, w_3, w_4)$ represents the objective weight vector of the DM as discussed in Section 4.2. We used Accuracy Favored, Accuracy and Cost Tradeoff, Equally Treated weight vectors as defined in Section 6.2 and same evolutionary algorithms' parameter settings with the previous experiments on Vehicle dataset as given in Table 6.2 for the experiments of quadratic preference function.

The progress of the best solution in the archive through iterations, $U^*(t) = \min_{z_i \in A(t)}\{U_{DM}(z_i)\}$, for 10 replications of iTDEA-fs, iWREA-fs and No Interaction are given in Appendix D for each experimental setting. As it can be inferred from the figures, in Accuracy Favored weight set experiments, iWREA-fs has a better and faster convergence to the best solution found by three algorithms in 10 replications, namely U_{MIN}. On the other hand, for Accuracy and Cost Tradeoff and Equally Treated weight vectors, all three algorithms are successful in converging to U_{MIN}, while iWREA-fs seems to converge faster in several replications.

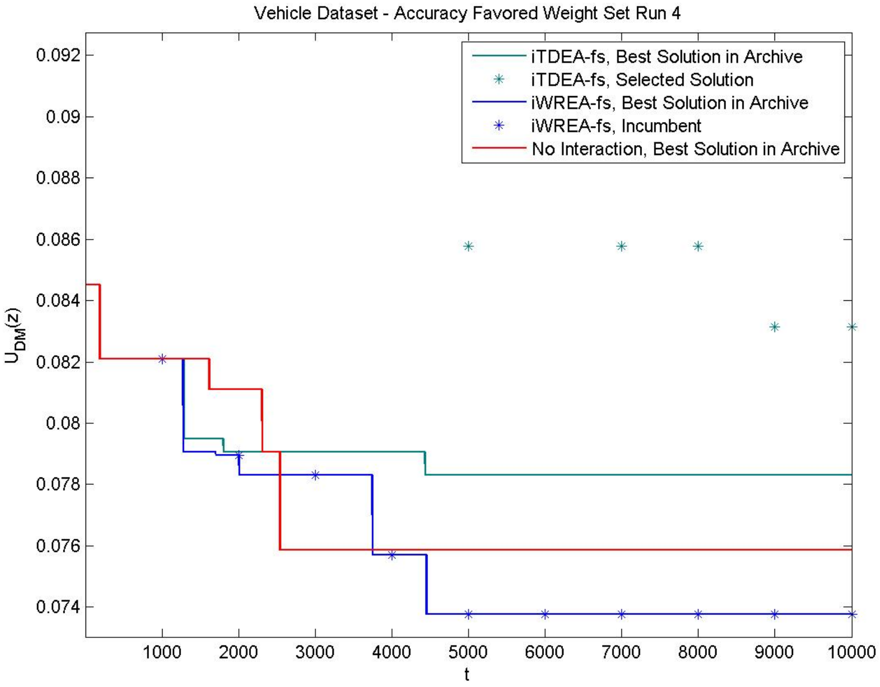

Figure 6.5 4th replication of quadratic preference function experiments with Accuracy Favored weight set

In Figure 6.5, the progress of the best solution for DM in the archive through iterations, $U^*(t)$, is shown for the 4th replication of quadratic preference function experiments on Vehicle dataset with Accuracy Favored weight set. The preference function value of the selected solutions of iTDEA-fs's and incumbent solutions of iWREA-fs's at interaction stages are represented in the same figure. Figure 6.5 demonstrates that iWREA-fs is able to identify best solution for DM as in previous experiments.

CHAPTER 7

CONCLUSION

Feature selection is an important problem as the result has a major impact on the performance, storage requirements, and computational efforts of learning algorithms.

In this study, we have implemented several variations of a preference-based evolutionary algorithm, iTDEA-fs, on the feature selection problem. As the results revealed special characteristics of the problem, we developed a new preference based evolutionary algorithm, iWREA-fs, that is compatible with those characteristics.

In addition to the traditional objectives defined for the feature selection problem in the literature, we set generic objectives that can be useful within different contexts of the problem. We defined the problem with representative objectives; however, in the presence of more objectives the algorithms can be used for more than four objectives.

The feature selection is used for many applications of classification problems. The DM of the problem can be different agencies or customers depending on the scope of the application area. For example, in health care, association of medical doctors, governmental agencies or patients can be the DM of the problem whose concerns are selecting a set of tests that provides accurate diagnosis while being cost-efficient and/or while minimizing health related risks involved in the tests. It may also be possible to select several meaningful subsets and then involve the patient in the final decision of which subset to use.

The results show that the interactions with the DM provide a higher convergence speed while finding solutions appealing to the DM in iWREA-fs. We believe employing the DM preferences is beneficial for solving feature selection problem in terms of bringing both flexibility on implementing the algorithm without dataset-specific parameters and ability that the final best solution for the DM is known at the end of algorithm. To the best of our knowledge, this is the first study that uses a preference-based approach and considers additional objectives together with the traditional ones for the feature selection problem.

It may be useful to try different underlying preference functions to further demonstrate the performance of algorithm. It may also be worthwhile to study DM inconsistencies. Köksalan and Karahan (2010) demonstrated that such inconsistencies did not deteriorate the performance of iTDEA much.

Being a parameter-free and computationally efficient algorithm, iWREA-fs can be tested for other combinatorial problems as a preference-based evolutionary algorithm as a future work. We also intend to compare the performance of the algorithms with commonly used multi-objective evolutionary algorithms in feature selection problem.

In some classification problems, there can be more than one class variable to be determined. That is, the features' values of an instance are used to classify it in more than one class. To illustrate, in a medical diagnosis case the patients can be classified in terms of two diseases: flu and cold, and it is possible that the patient has both, only one, or none of those illnesses. As a future work, the algorithms developed in this study can be applied on feature selection for this type of classification problems.

REFERENCES

Bolón-Canedo, V., Porto-Díaz, I., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2014). A framework for cost-based feature selection. *Pattern Recognition*, *47*(7), 2481-2489.

Deb, K. (2001). *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research,* 3, 1157-1182.

Hadka, D., & Reed, P. (2013). Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework. *Evolutionary Computation*, *21*(2), 231-259.

Hamdani T. M., Won J.-M., Alimi A. M., & Karray F. (2007). Multi-objective feature selection with NSGA II. *Proceedings of 8th ICANNGA Part I*, 4431, 240-247.

Huang, B., Buckley, B., & Kechadi, T. (2010). Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. *Expert Systems with Applications*, *37*(5), 3638-3646.

Huang, G., Zhu, Q., & Siew, C. (2006). Extreme learning machine: theory and applications. *Neurocomputing, 70*(1), 489-501.

Huang, G., Zhu, Q., & Siew, C. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 42*(2), 513-529.

Karakaya, G., Galelli, S., Ahipasaoglu, S., & Taormina, R. (2016). Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach. *IEEE Transactions on Cybernetics*, 46(6), 1424-1437.

Karakaya, G. & Köksalan, M. (2014). An interactive approach for Bi-attribute multi-item auctions. *Annals of Operations Research*, doi: 10.1007/s10479-014-1669-4.

Karakaya G., Köksalan M., & Ahipaşaoğlu S.D. (2016). Interactive Algorithms for a broad underlying family of preference functions, Industrial Engineering Department, METU, Technical Report, 16(1).

Kennedy, J. & Eberhart, R. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence,* 97(1), 273-324.

Koksalan, M., & Karahan, I. (2010). An interactive territory defining evolutionary algorithm: ITDEA. *IEEE Transactions on Evolutionary Computation, 14*(5), 702-722.

Kotsiantis, S.B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica,* 31, 249-268.

Liu, Y., Tang, F., & Zeng, Z. (2015). Feature selection based on dependency margin. *IEEE Transactions on Cybernetics, 45*(6), 1209-1221.

Oliveira, L., Sabourin, R., Bortolozzi, F., & Suen, C. (2003). A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(06), 903-929.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427-437.

Srinivas, N. & Deb, K. (1995). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, *2*(3), 221-248.

Tan, M. (1993). Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, *13*(1), 7-33.

Xue, B., Zhang, M., & Browne, W. (2013). Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Transactions on Cybernetics*, *43*(6), 1656-1671.

Xue, B., Zhang, M., Browne, W., & Yao, X. (2016). A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606-626.

Yu, L., & Liu, H., (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*,1205-1224.

Zhang, Y., Gong, D., & Cheng, J. (2015). Multi-objective Particle Swarm Optimization Approach for Cost-based Feature Selection in Classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1-1.

APPENDICIES

APPENDIX A

PERFORMANCE MEASUREMENTS

Table A.1 Heart Disease – Accuracy Favored

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	1	0	0
2	0	0	0
3	0	0	0.3062
4	0.3062	0	0.3062
5	0.9632	0	0
6	0.3062	0	0
7	0	0	0
8	0.3062	0	0.3062
9	0.1498	0	0
10	0.8554	0.4016	0

Table A.2 Heart Disease – Accuracy and Cost Tradeoff

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0	0	0
2	0.7563	0	0
3	0.573	0	0
4	0	0	0
5	0.7563	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0.7563
10	0	0	1

Table A.3 Heart Disease – Equally Treated

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0	0	0
2	0	0	0
3	0.5427	0	0.2413
4	0	0	0.6967
5	1	0.2413	0.5593
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Table A.4 Vehicle – Accuracy Favored

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0	0	0
2	0.4681	0.625	0.9925
3	0.43	0.4681	0.8432
4	1	0	0.4681
5	1	0	1
6	0.4681	0.43	1
7	1	0	0.802
8	0.4681	0	0.9483
9	0.9925	0	1
10	0.8432	0	0.4681

Table A.5 Vehicle – Accuracy and Cost Tradeoff

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	1	0	0.6206
2	0	0	0.2525
3	0.2525	0	0.6206
4	0.2525	0	0.6206
5	1	0	0.6206
6	0.2525	0	0.2525
7	0.2525	0	1
8	1	0	1
9	1	0	0.6206
10	0.2525	0	0.2525

Table A.6 Vehicle – Equally Treated

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	1	1	1
2	1	0	1
3	1	1	1
4	1	0	1
5	1	1	1
6	1	0	1
7	0	1	1
8	1	0	1
9	0	0	0
10	1	0	0

Table A.7 German – Accuracy Favored

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0.338	0.338	0.6068
2	0.6656	0.338	0.6068
3	0.7826	0.338	0.8166
4	0.811	0	0.7826
5	0.7826	0.338	0.7826
6	1	0.338	1
7	0.7826	0	0.7826
8	0.338	0	1
9	0.6656	0.338	0
10	1	0.7826	0.338

Table A.8 German – Accuracy and Cost Tradeoff

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0	0	0
2	0.0585	0	0
3	0	0	0
4	0	0	0
5	0.0585	0	0.0585
6	0	0.0585	0.0585
7	1	0	0
8	0	0	0
9	0	0	0
10	0.0585	0.0585	0.0585

Table A.9 German – Equally Treated

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0.1063	0	0.1063
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0.3451
6	1	0.3451	0.6463
7	0	0	0.1063
8	0	0	0
9	0	0	0
10	1	0.064	0.9576

Table A.10 Breast Cancer – Accuracy Favored

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0.54	0.4488	0
2	0	0	0.54
3	0.54	0	0.6548
4	0	0	0.9896
5	0.8363	0	0.54
6	0.5896	0	0.9907
7	0.54	0	0.54
8	0.8497	0	1
9	0.54	0	0.6152
10	0	0	0.7692

Table A.11 Breast Cancer – Accuracy and Cost Tradeoff

Run	$U^r_{\text{iTDEA-fs}}$	$U^r_{\text{iWREA-fs}}$	$U^r_{\text{No Interaction}}$
1	0.672	0	0.672
2	0.672	0	0.672
3	0	0	0.7724
4	1	0	0.672
5	0	0	0
6	0	0	0
7	0	0	0
8	0.672	0	0
9	0.672	0	0.7724
10	0	0	0.672

Table A.12 Breast Cancer – Equally Treated

Run	$U^{r}_{\text{iTDEA-fs}}$	$U^{r}_{\text{iWREA-fs}}$	$U^{r}_{\text{No Interaction}}$
1	0	0	0
2	0.065	0	0
3	1	0	0
4	0	0	0
5	0.065	0	0
6	0.065	0	0.065
7	0	0	0
8	0.065	0	0
9	0.6705	0	0.065
10	0.065	0	0.065

APPENDIX B

ARCHIVE PROGRESSES

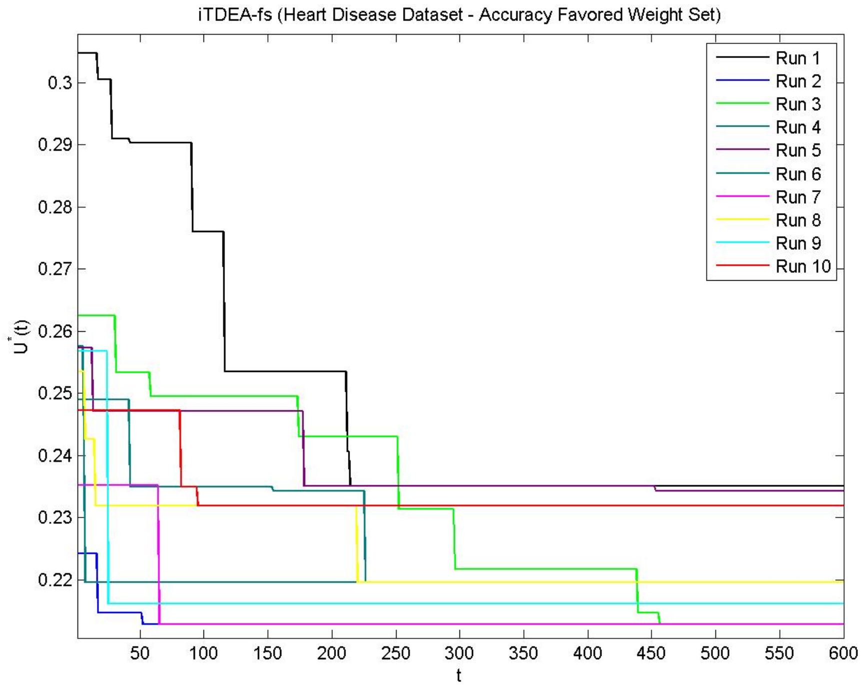

Figure B.1 Heart Disease – Accuracy Favored – iTDEA-fs

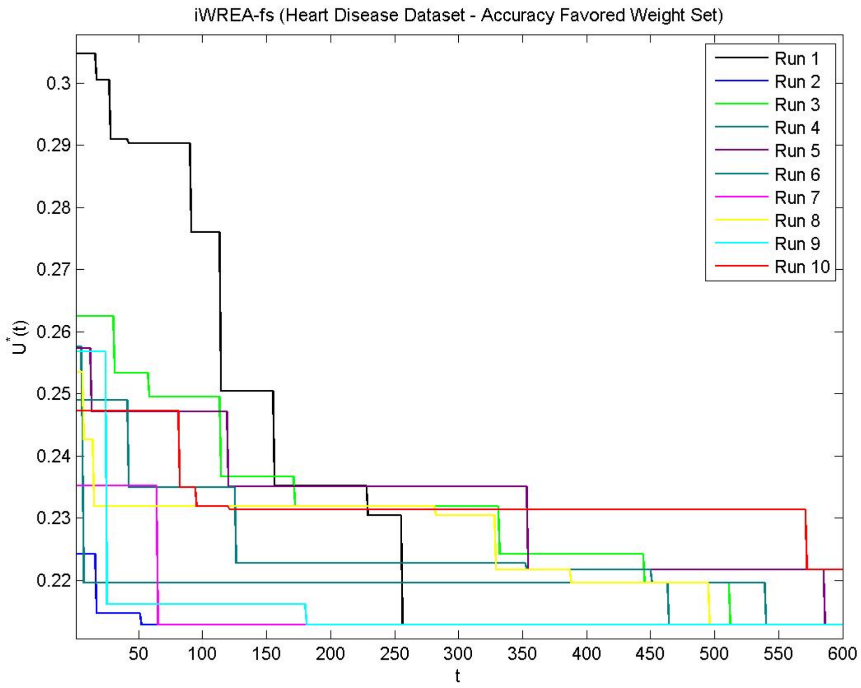

Figure B.2 Heart Disease – Accuracy Favored – iWREA-fs

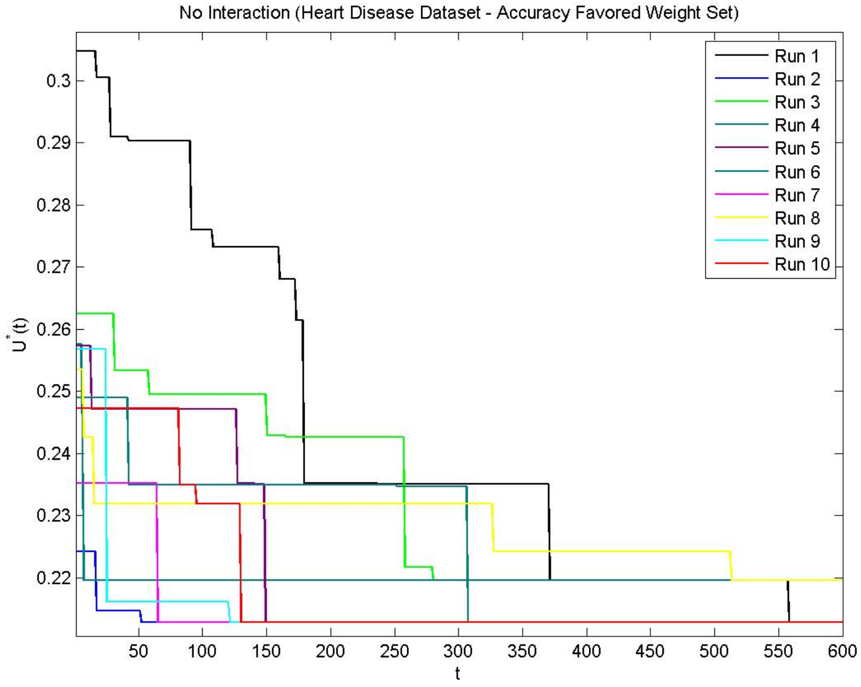

Figure B.3 Heart Disease – Accuracy Favored – No Interaction

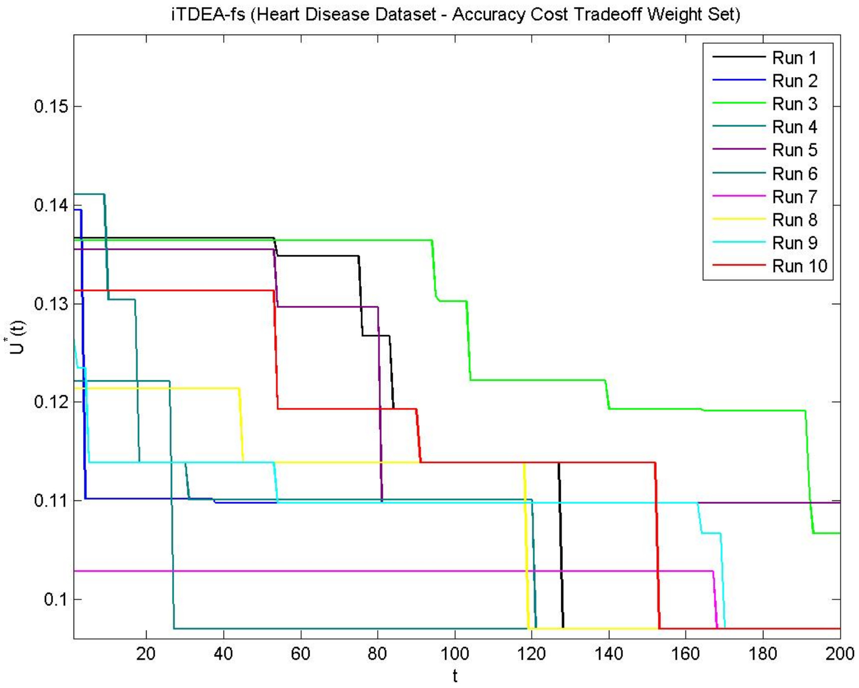

Figure B.4 Heart Disease – Accuracy and Cost Tradeoff – iTDEA-fs

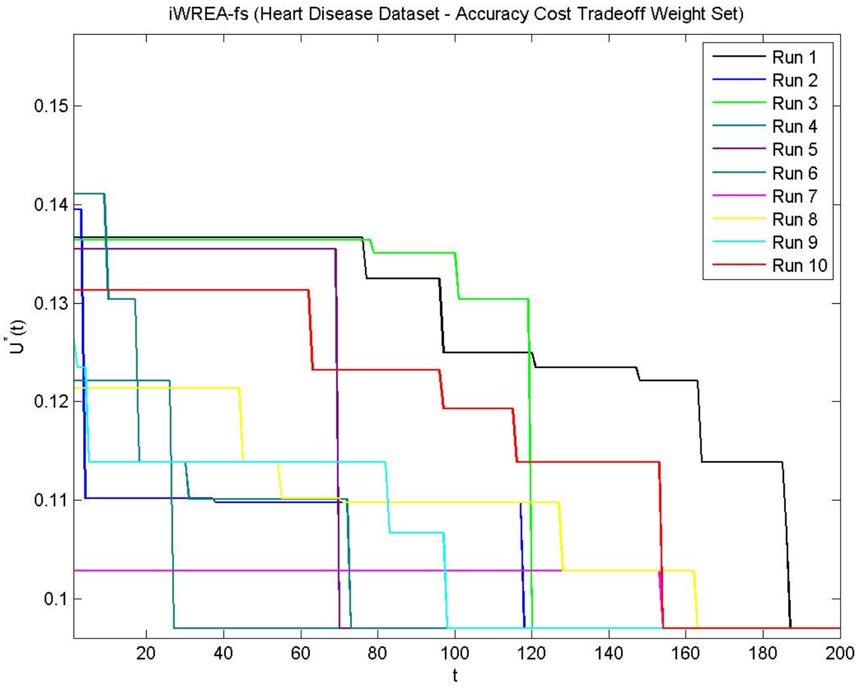

Figure B.5 Heart Disease – Accuracy and Cost Tradeoff – iWREA-fs

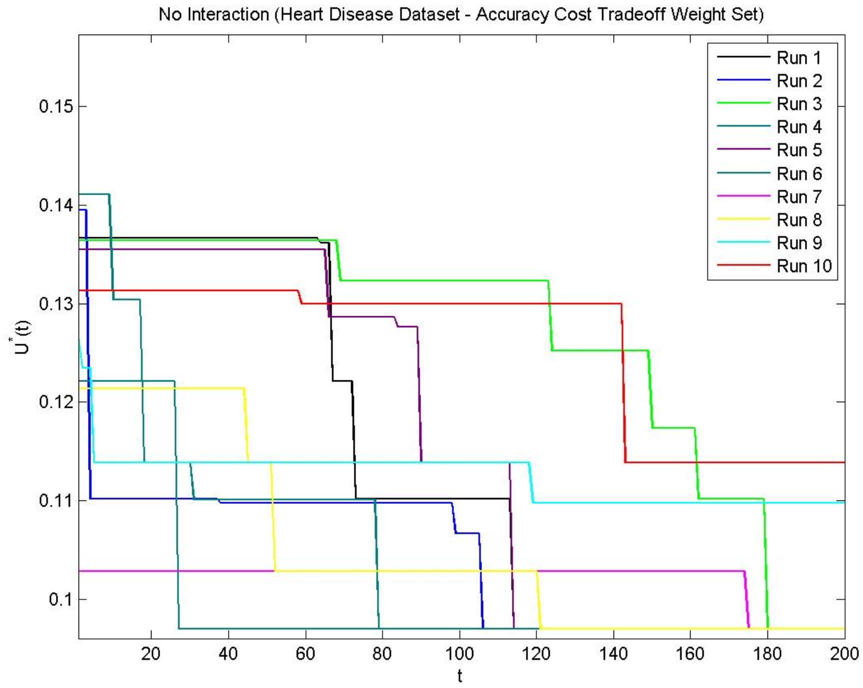

Figure B.6 Heart Disease – Accuracy and Cost Tradeoff – No Interaction

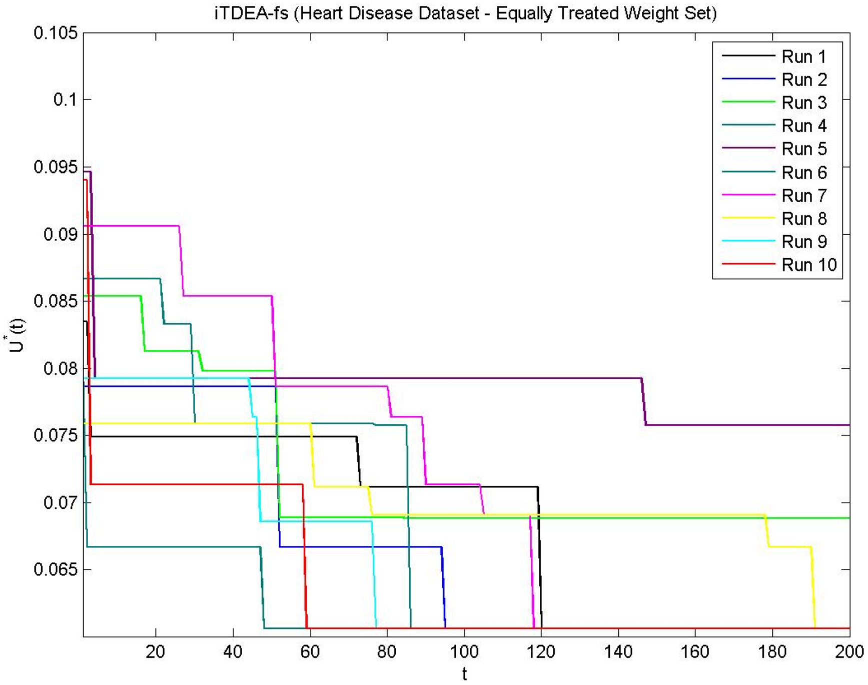

Figure B.7 Heart Disease – Equally Treated – iTDEA-fs

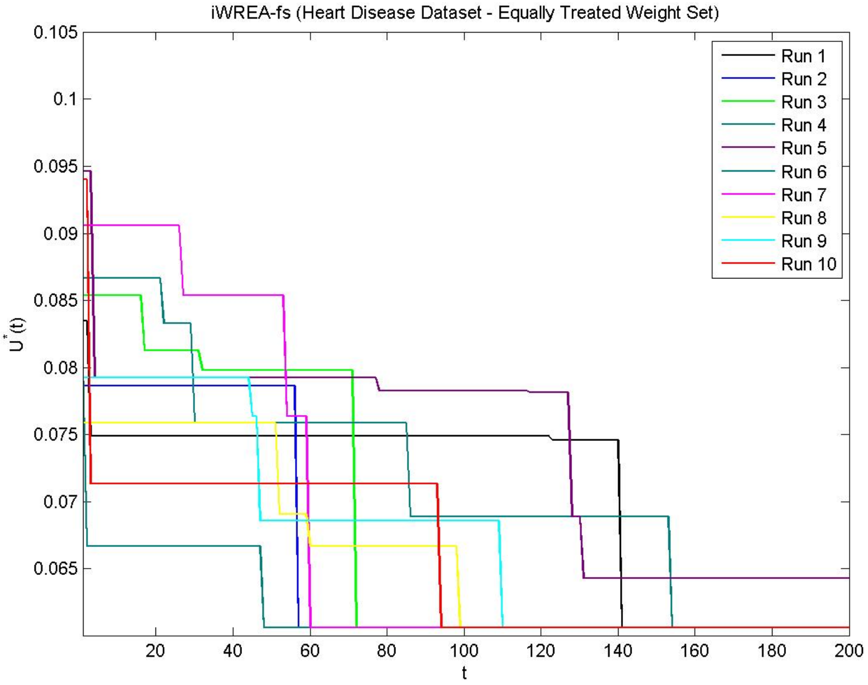

Figure B.8 Heart Disease – Equally Treated – iWREA-fs

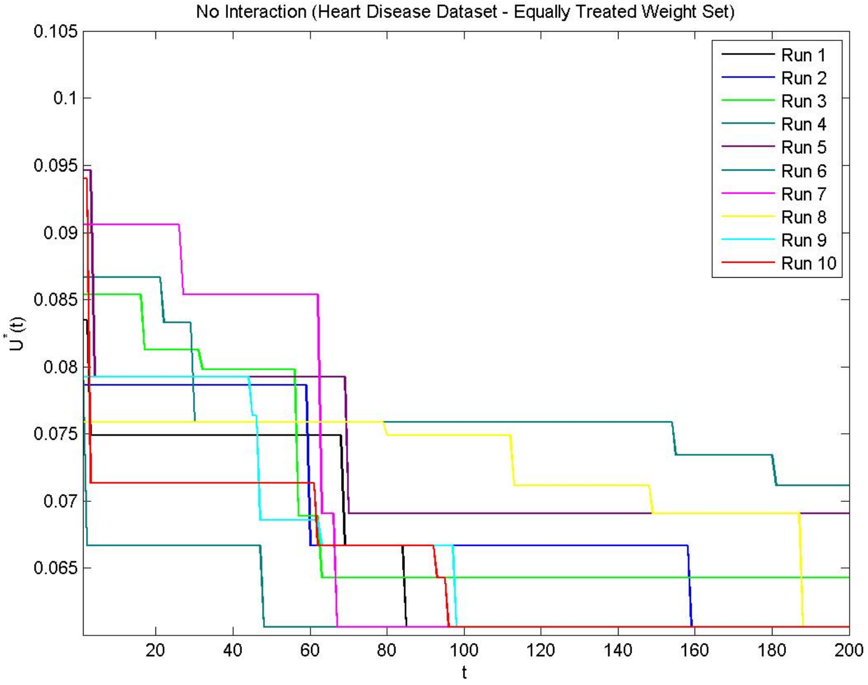

Figure B.9 Heart Disease – Equally Treated – No Interaction

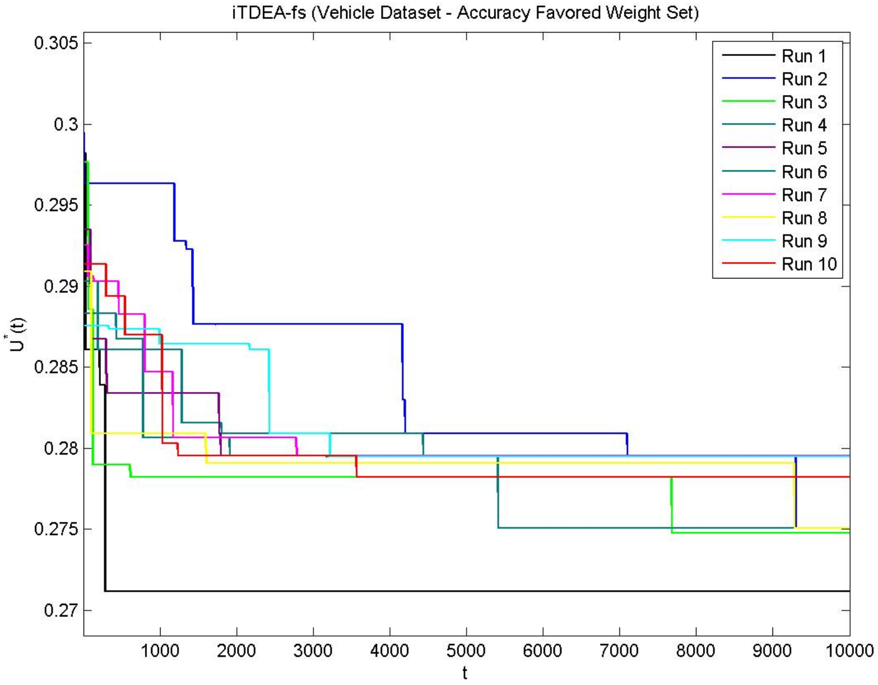

Figure B.10 Vehicle – Accuracy Favored – iTDEA-fs

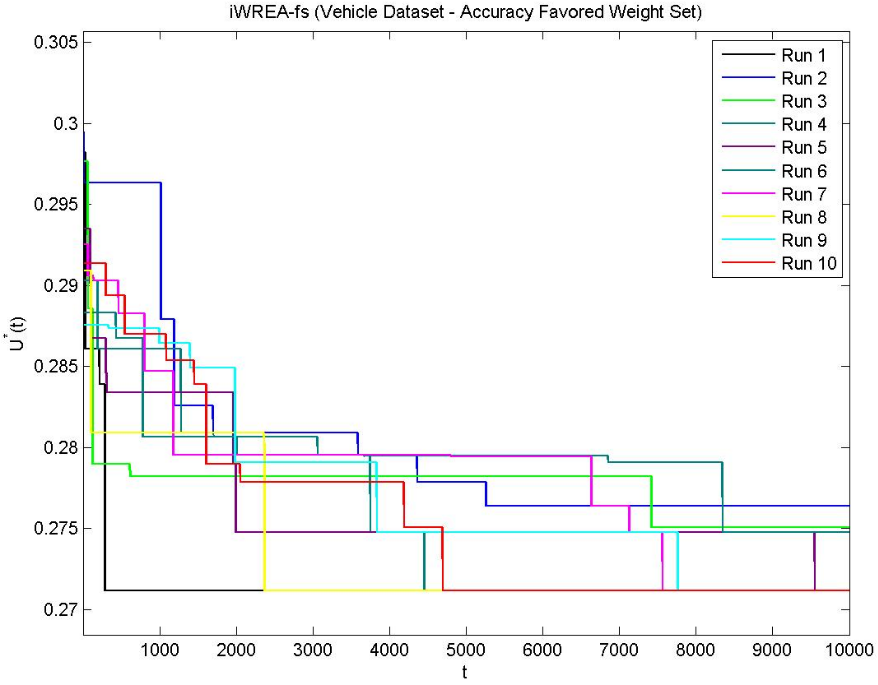

Figure B.11 Vehicle – Accuracy Favored – iWREA-fs

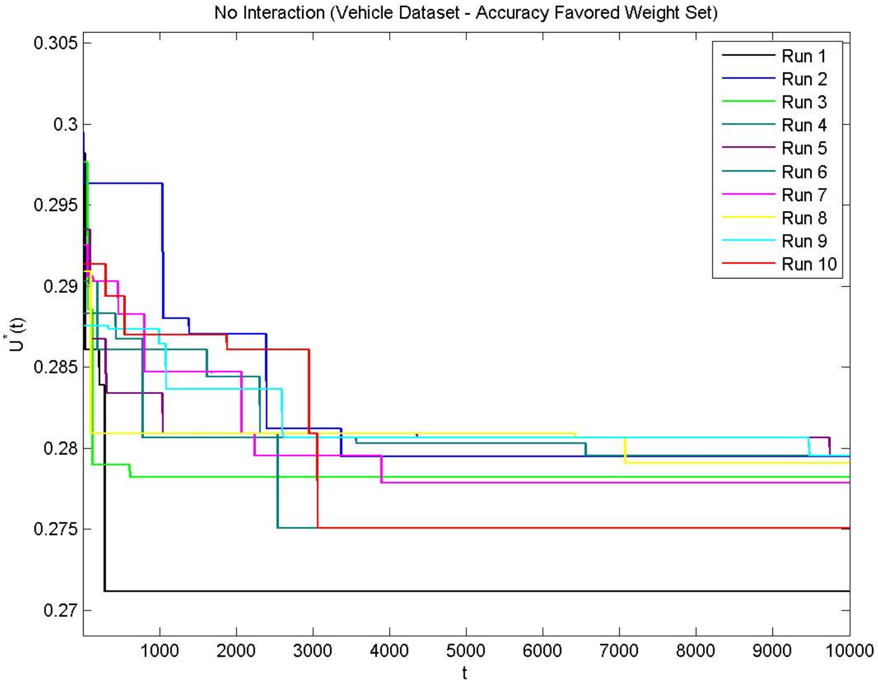

Figure B.12 Vehicle – Accuracy Favored – No Interaction

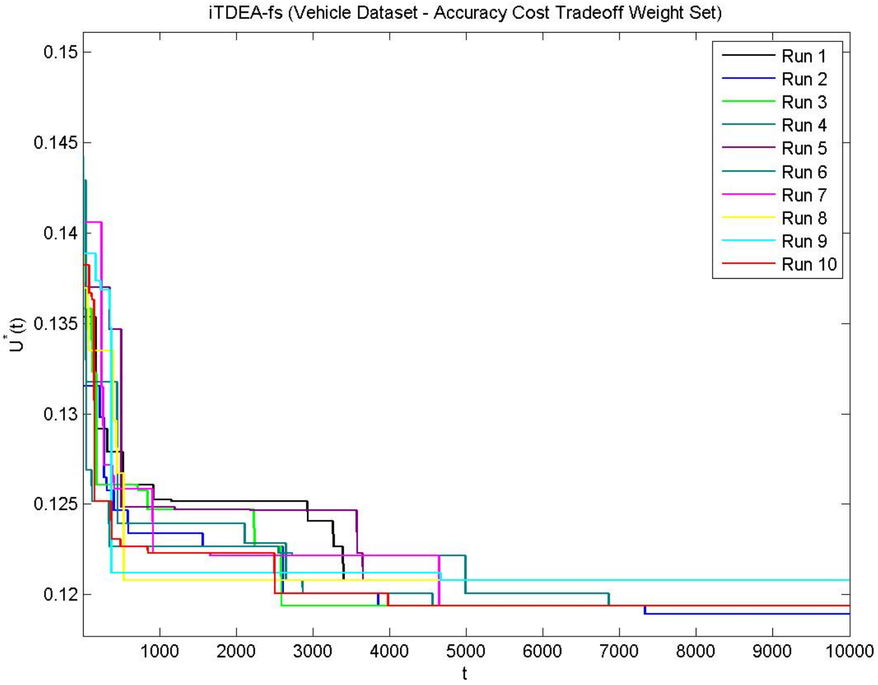

Figure B.13 Vehicle – Accuracy and Cost Tradeoff – iTDEA-fs

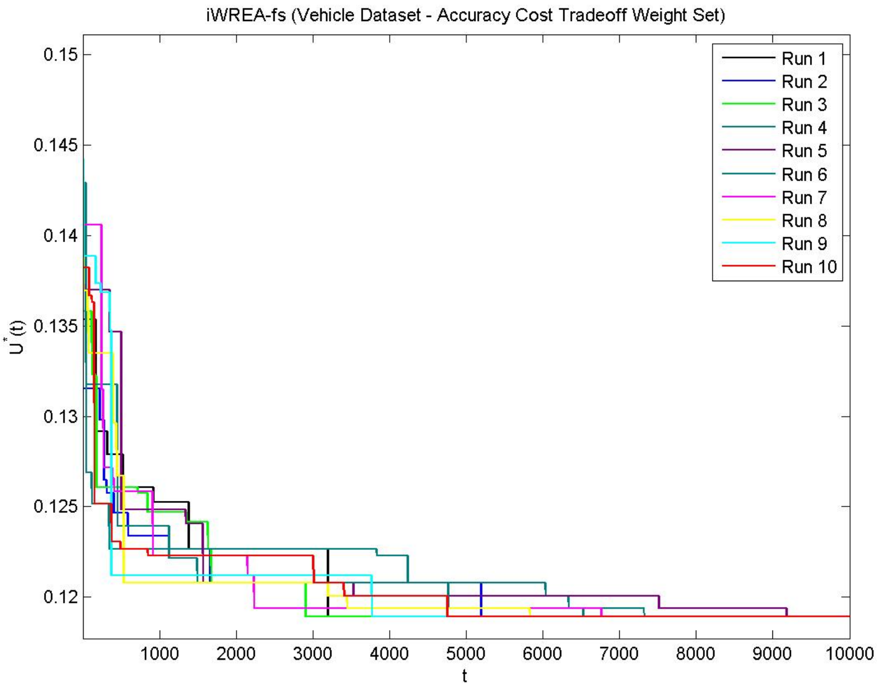

Figure B.14 Vehicle – Accuracy and Cost Tradeoff – iWREA-fs

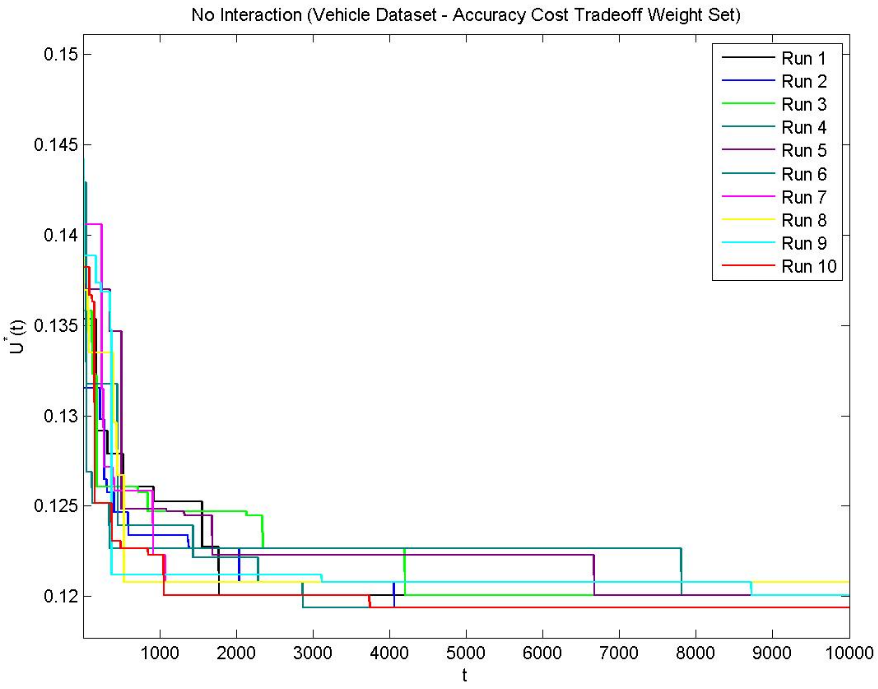

Figure B.15 Vehicle – Accuracy and Cost Tradeoff – No Interaction

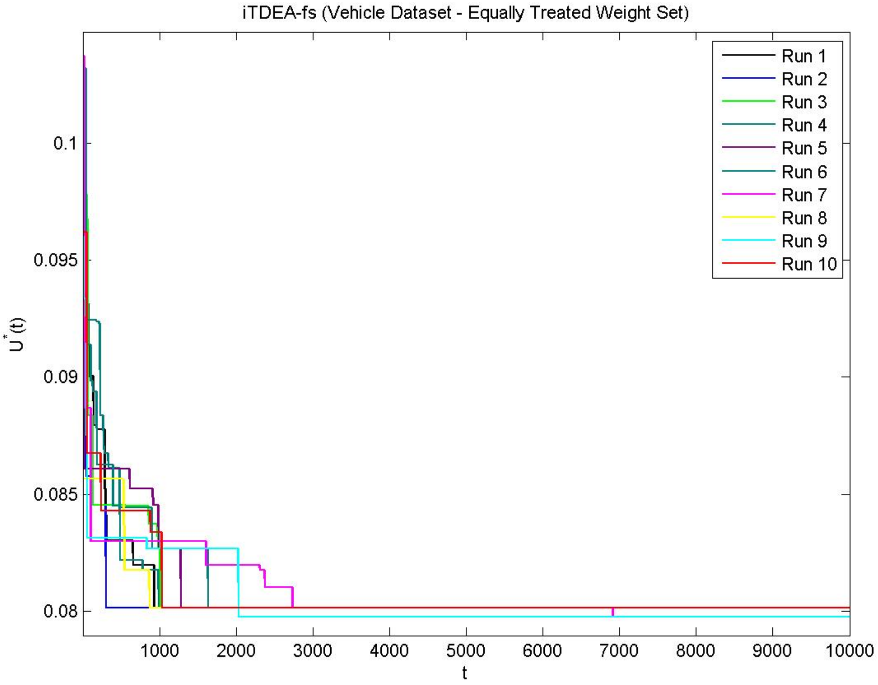

Figure B.16 Vehicle – Equally Treated – iTDEA-fs

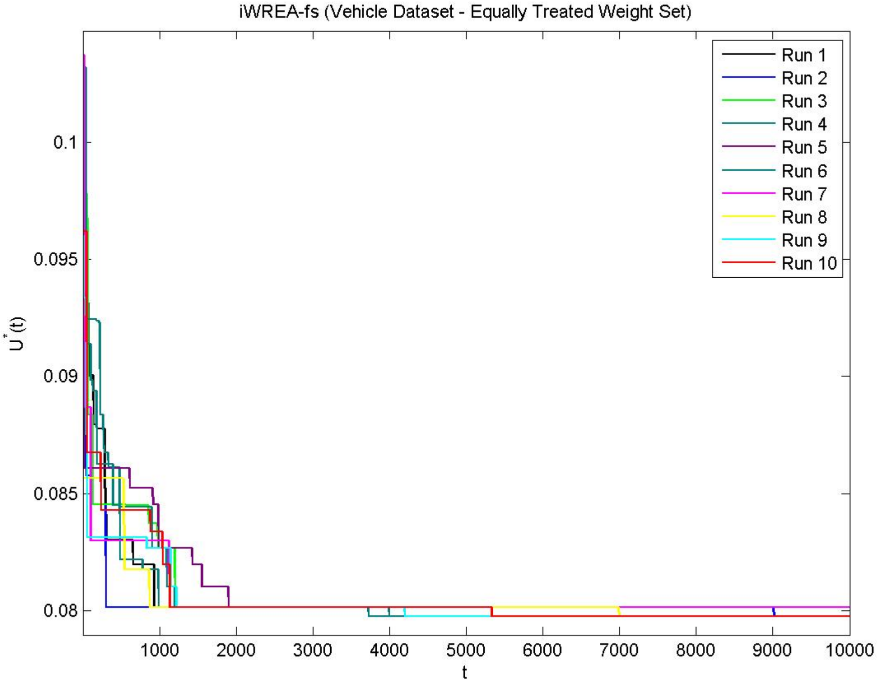

Figure B.17 Vehicle – Equally Treated – iWREA-fs

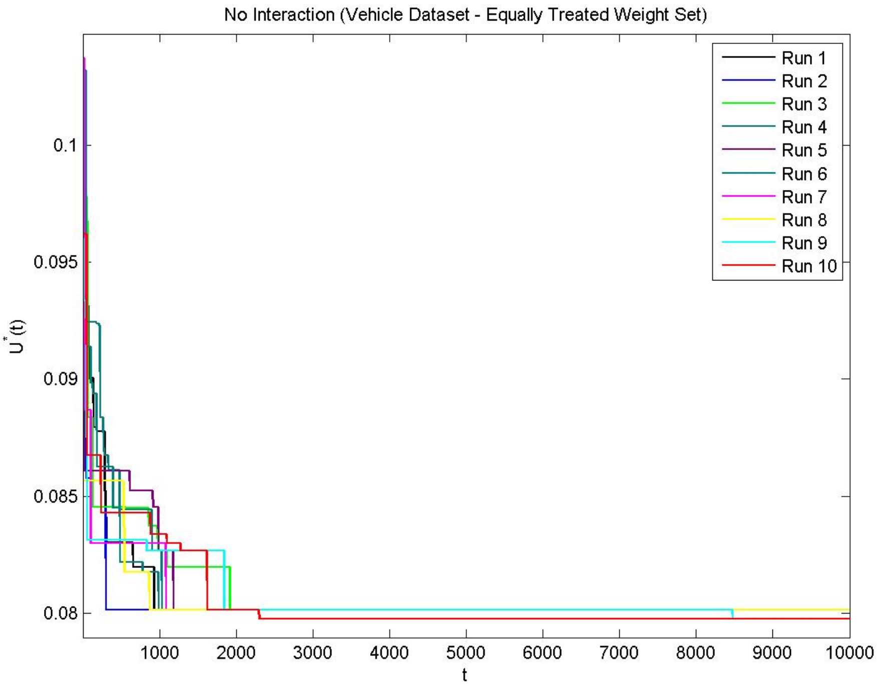

Figure B.18 Vehicle – Equally Treated – No Interaction

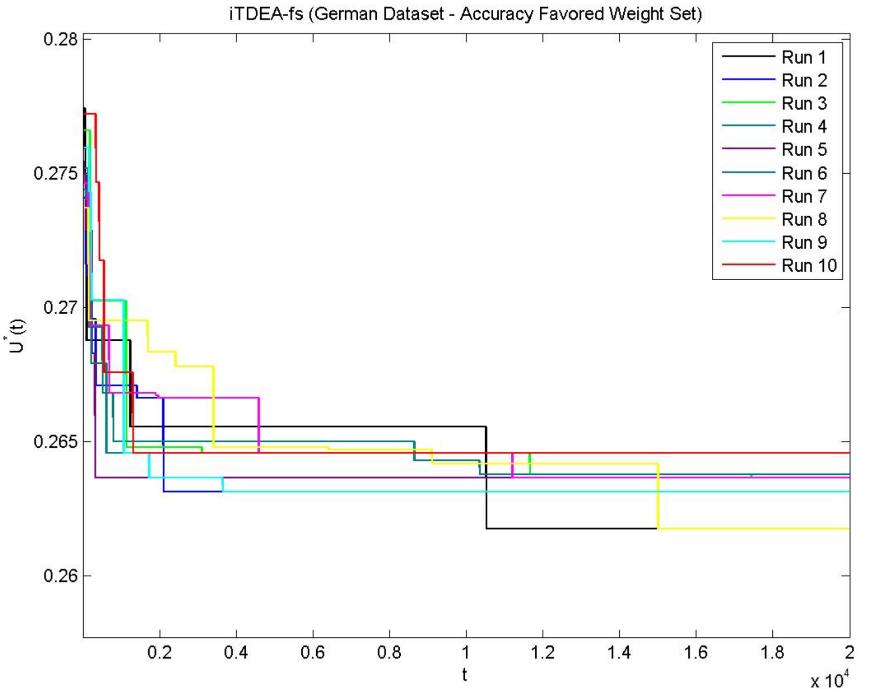

Figure B.19 German – Accuracy Favored – iTDEA-fs

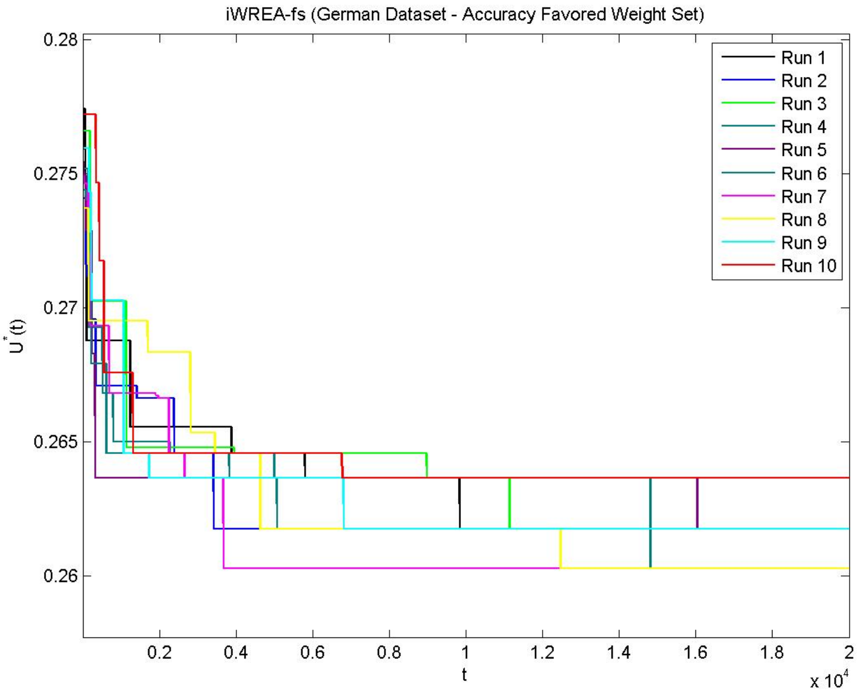

Figure B.20 German – Accuracy Favored – iWREA-fs

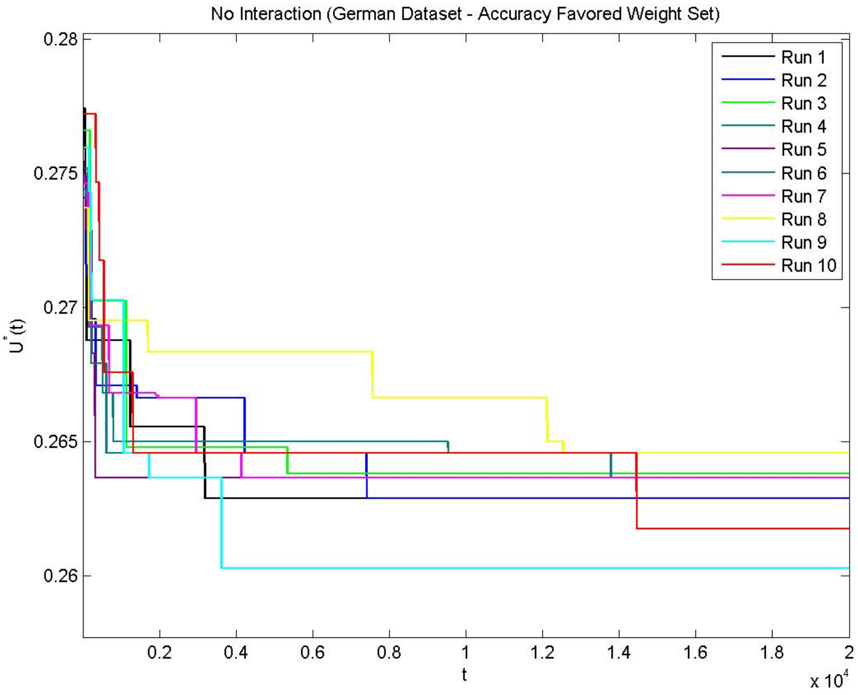

Figure B.21 German – Accuracy Favored – No Interaction

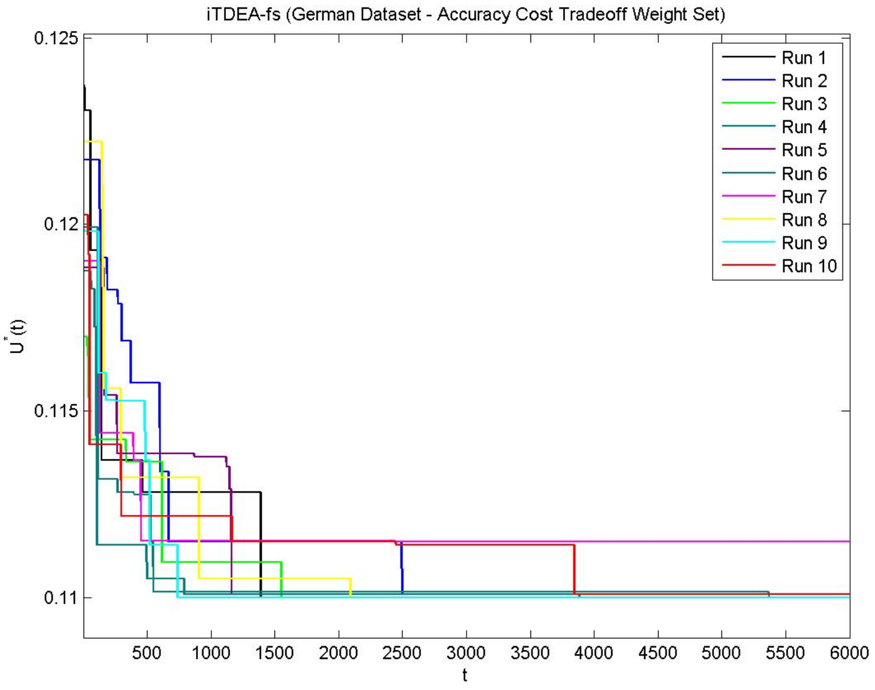

Figure B.22 German – Accuracy and Cost Tradeoff – iTDEA-fs

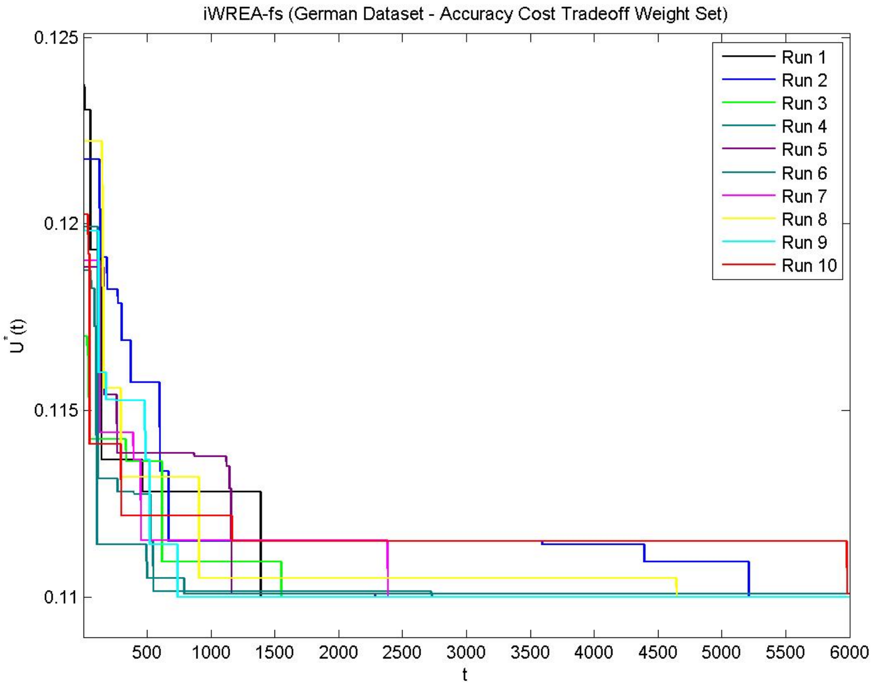

Figure B.23 German – Accuracy and Cost Tradeoff – iWREA-fs

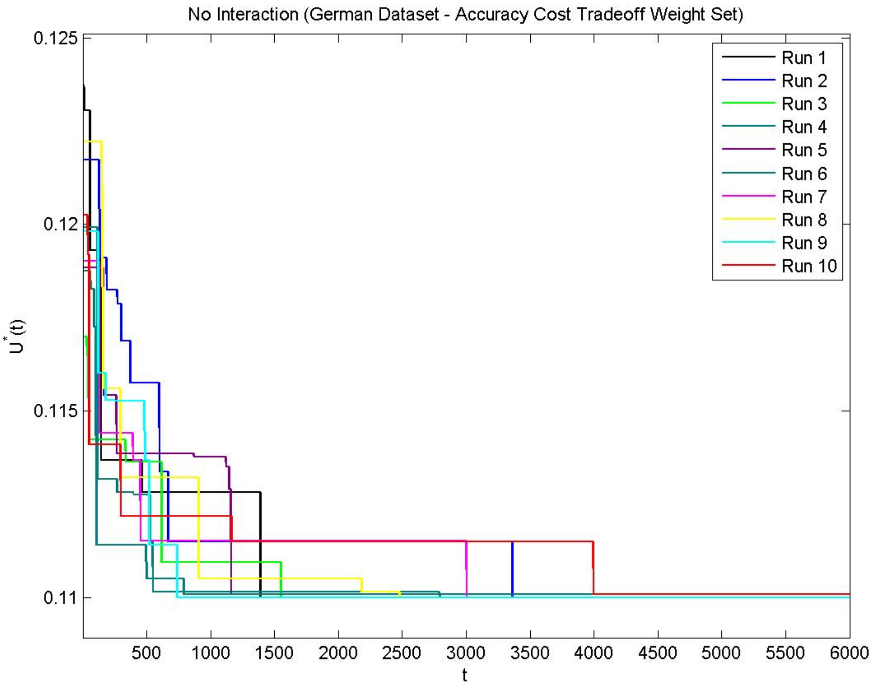

Figure B.24 German – Accuracy and Cost Tradeoff – No Interaction

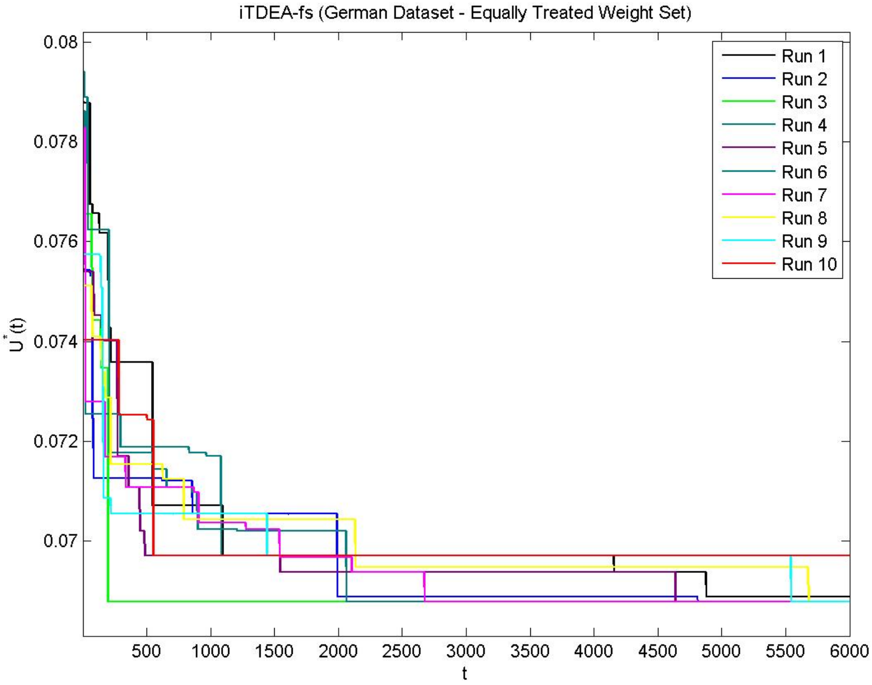

Figure B.25 German – Equally Treated – iTDEA-fs

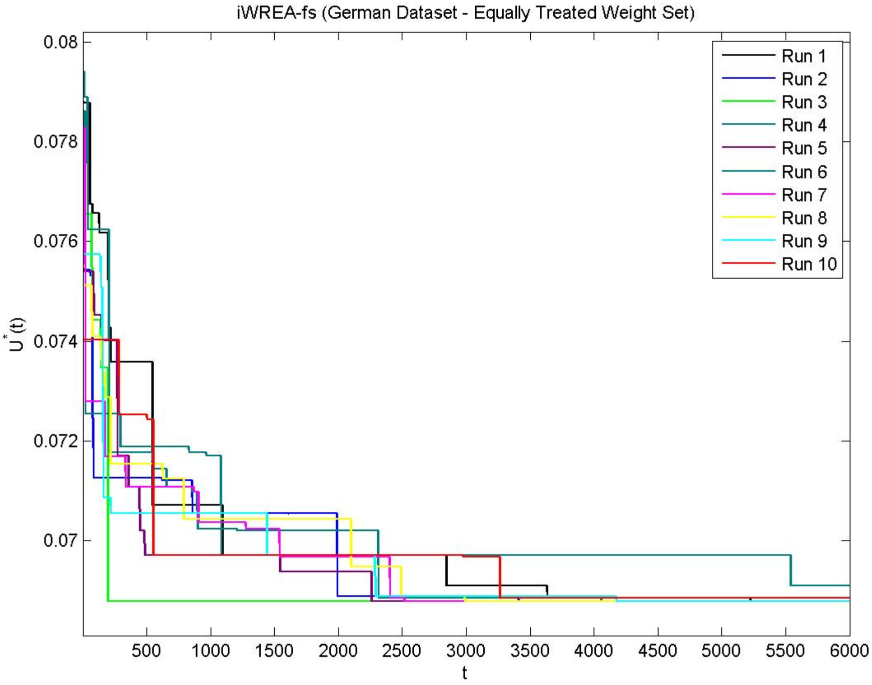

Figure B.26 German – Equally Treated – iWREA-fs

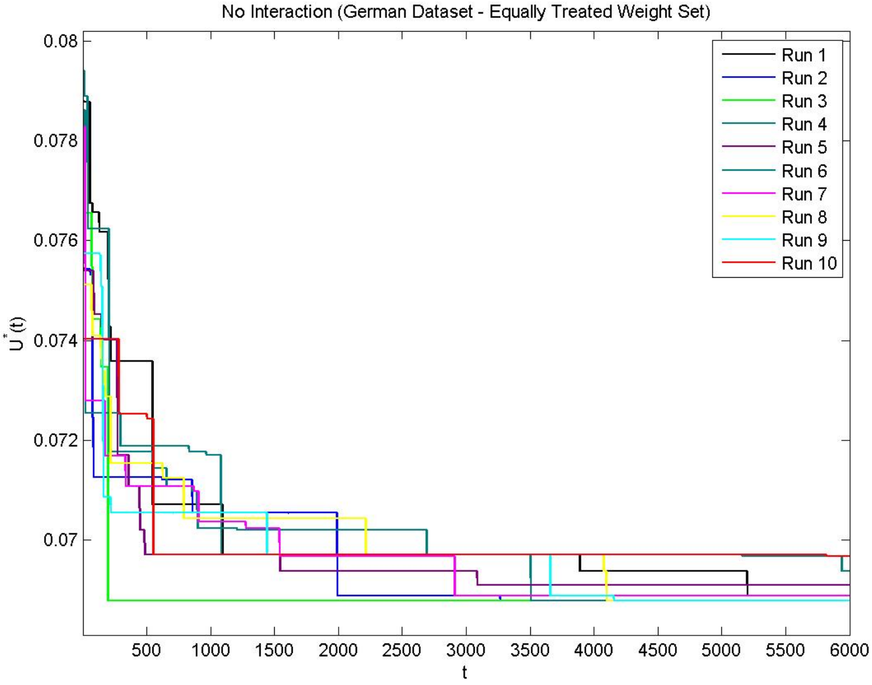

Figure B.27 German – Equally Treated – No Interaction

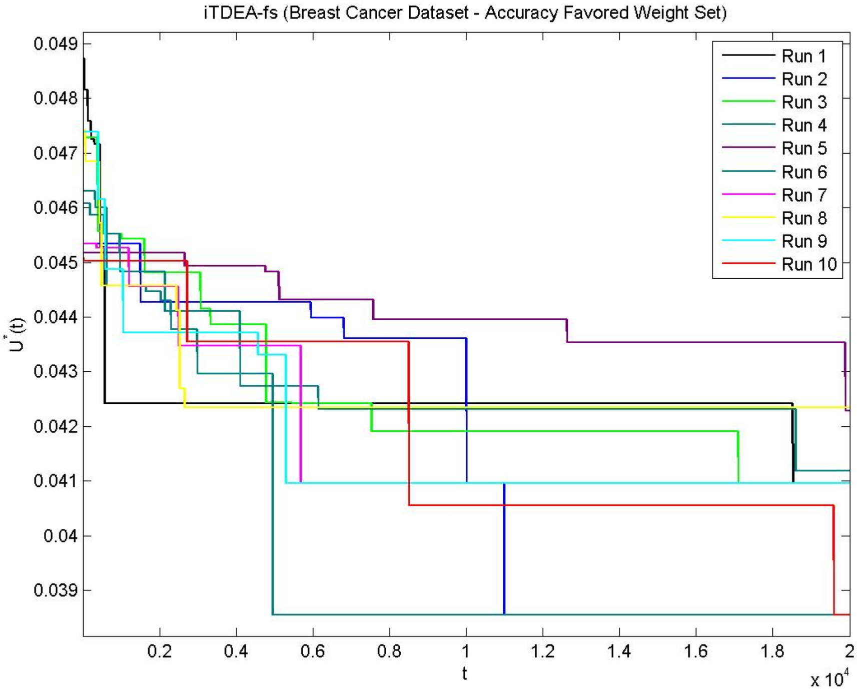

Figure B.28 Breast Cancer – Accuracy Favored – iTDEA-fs

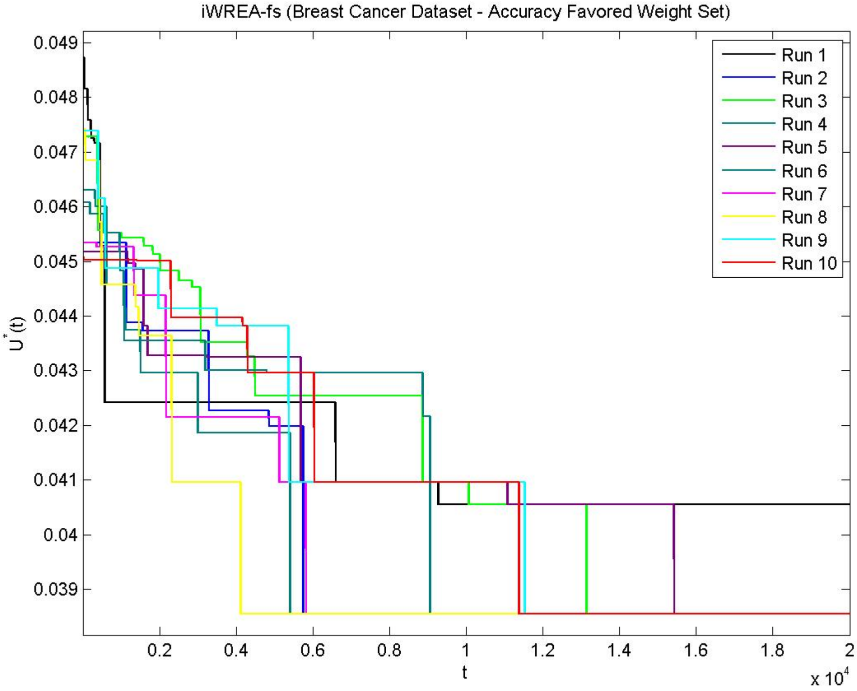

Figure B.29 Breast Cancer – Accuracy Favored – iWREA-fs

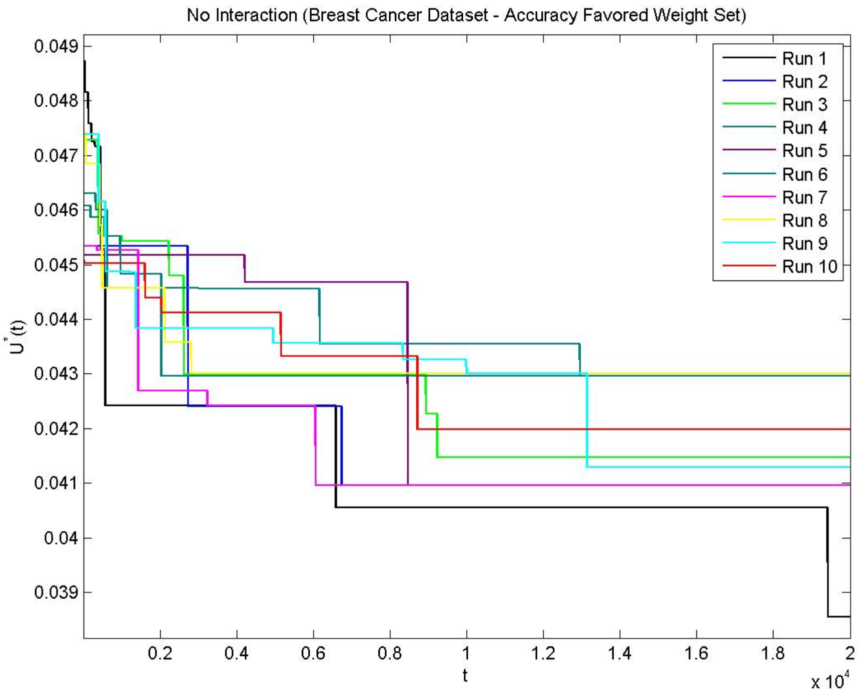

Figure B.30 Breast Cancer – Accuracy Favored – No Interaction

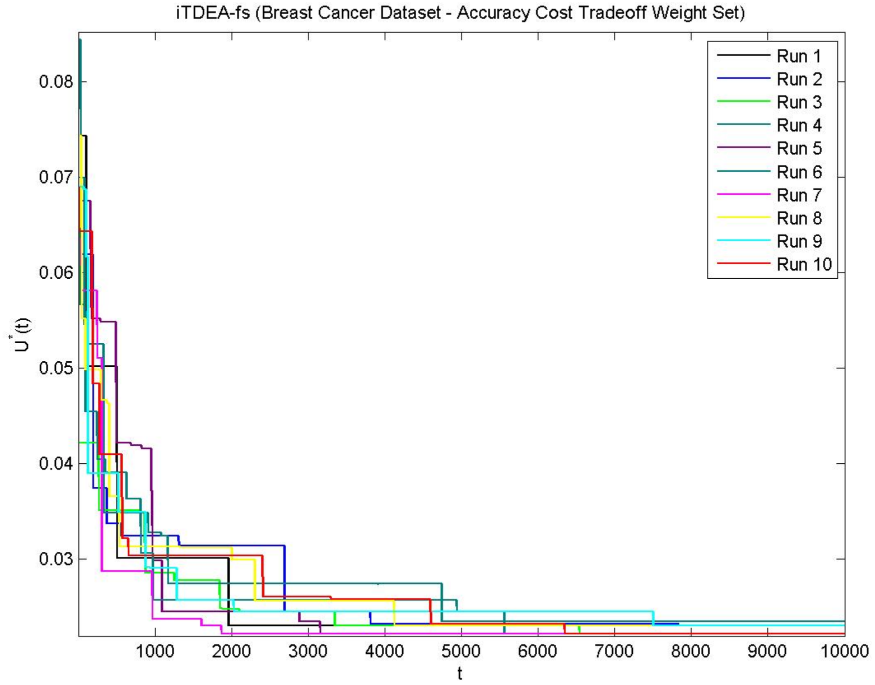

Figure B.31 Breast Cancer – Accuracy Cost Tradeoff – iTDEA-fs

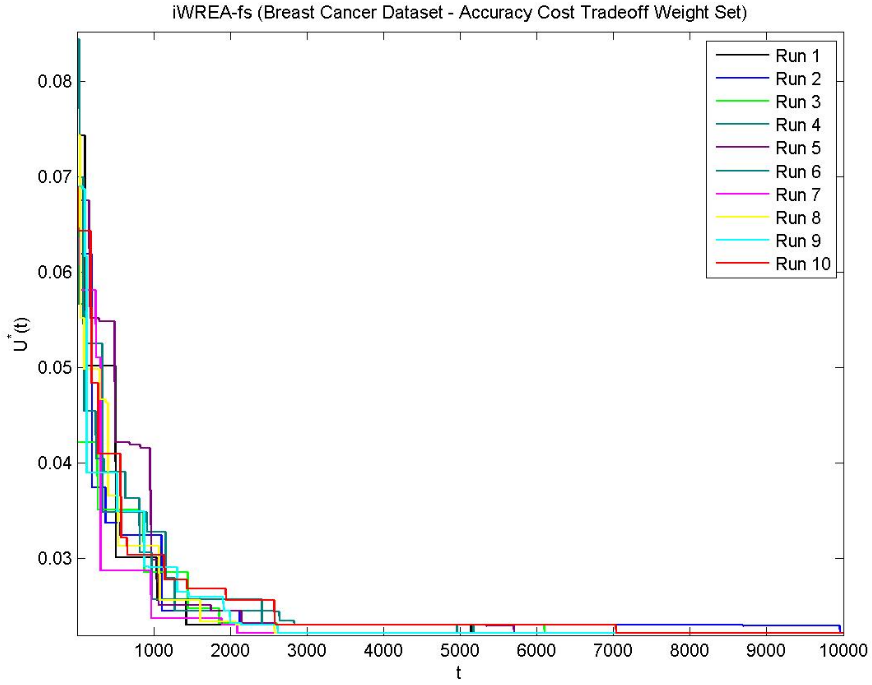

Figure B.32 Breast Cancer – Accuracy Cost Tradeoff – iWREA-fs

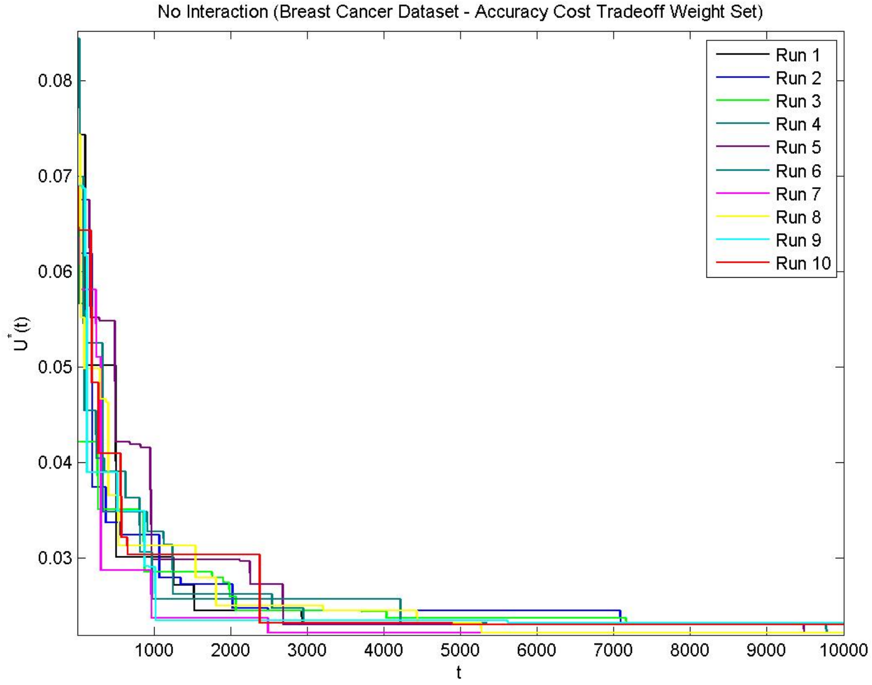

Figure B.33 Breast Cancer – Accuracy Cost Tradeoff – No Interaction

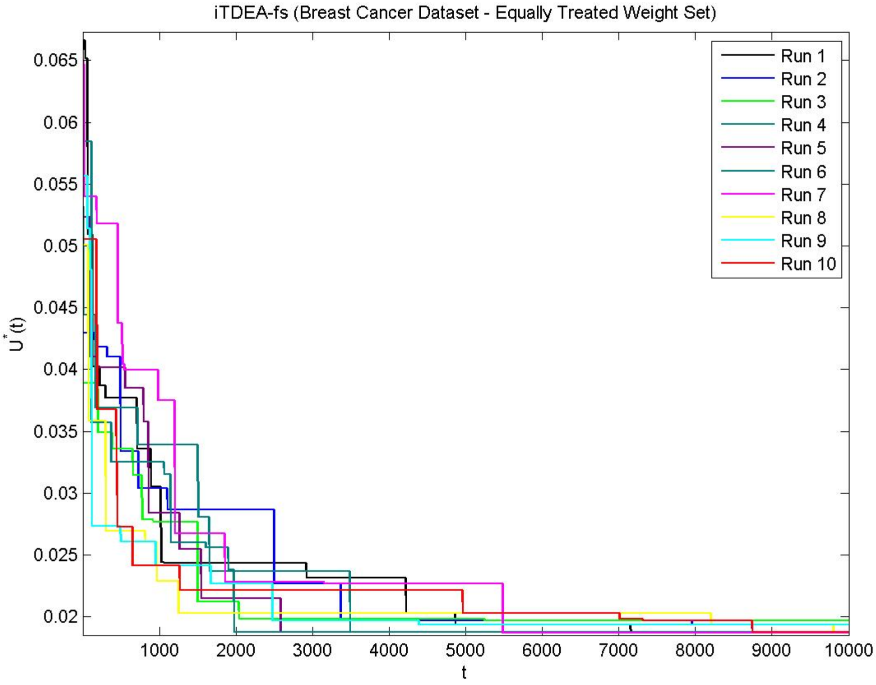

Figure B.34 Breast Cancer – Equally Treated – iTDEA-fs

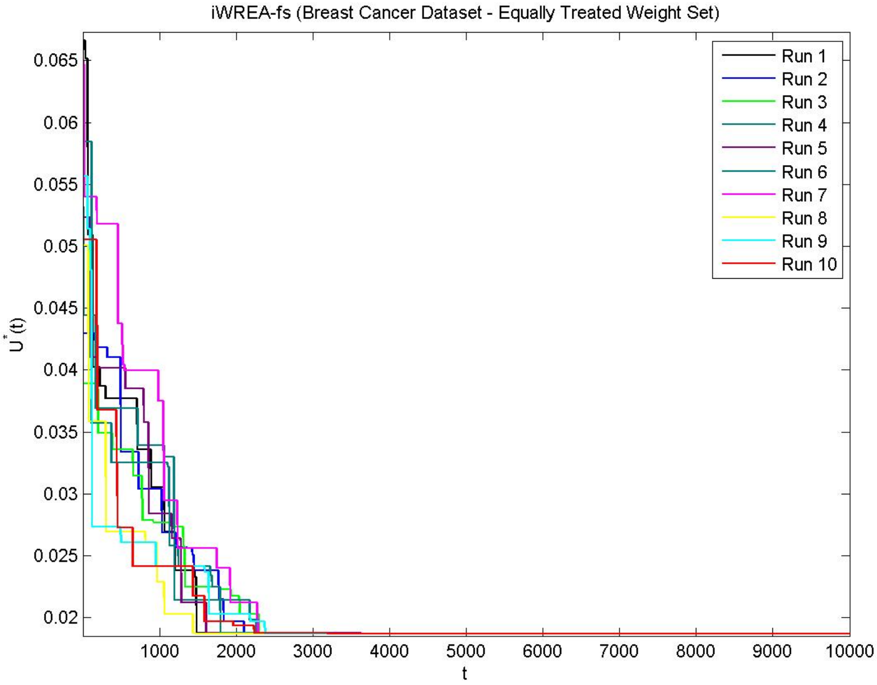

Figure B.35 Breast Cancer – Equally Treated – iWREA-fs

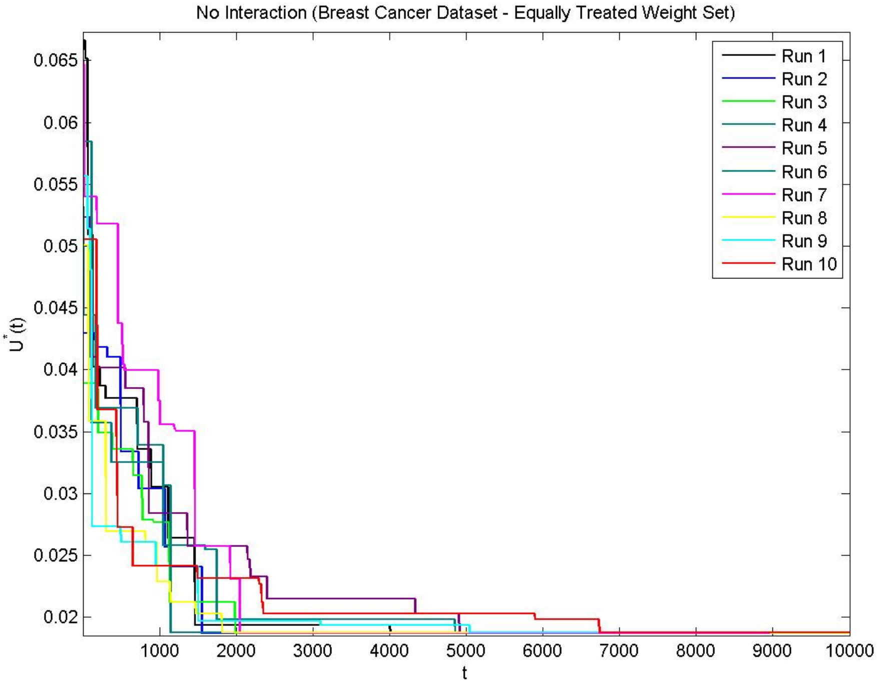

Figure B.36 Breast Cancer – Equally Treated – No Interaction

APPENDIX C

COMPUTATIONAL TIMES

Table C.1 CPU Times of Heart Disease experiments (in seconds)

Run	iTDEA-fs	iWREA-fs	No Interaction
1	0.38	1.47	0.39
2	0.41	1.50	0.39
3	0.38	1.98	0.38
4	0.39	1.41	0.39
5	0.41	1.61	0.39
6	0.39	1.41	0.36
7	0.34	1.70	0.38
8	0.36	1.58	0.38
9	0.38	1.27	0.36
10	0.38	1.63	0.41

Table C.2 CPU Times of Vehicle experiments (in seconds)

Run	iTDEA-fs	iWREA-fs	No Interaction
1	15.98	8.92	16.19
2	16.38	8.09	16.03
3	17.34	7.38	17.06
4	16.48	7.33	16.72
5	16.83	6.17	16.23
6	17.05	7.61	16.88
7	16.19	6.66	16.56
8	16.28	6.58	16.17
9	16.20	7.25	16.66
10	17.00	8.06	16.67

Table C.3 CPU Times of German experiments (in seconds)

Run	iTDEA-fs	iWREA-fs	No Interaction
1	71.45	17.47	71.92
2	74.30	17.64	74.05
3	73.70	19.81	75.13
4	71.72	18.41	72.56
5	71.14	17.17	71.08
6	71.55	18.97	72.06
7	69.89	18.72	71.45
8	70.66	18.14	71.86
9	73.39	18.48	73.98
10	70.50	15.80	72.45

Table C.4 CPU Times of Breast Cancer experiments (in seconds)

Run	iTDEA-fs	iWREA-fs	No Interaction
1	143.66	32.70	150.73
2	147.38	37.14	154.27
3	148.00	33.13	145.64
4	145.06	33.47	149.06
5	145.47	35.64	142.52
6	139.25	32.16	146.03
7	145.61	34.64	146.50
8	141.13	32.05	144.86
9	145.52	34.86	144.48
10	141.94	33.88	144.92

APPENDIX D

QUADRATIC PREFERENCE FUNCTION RESULTS

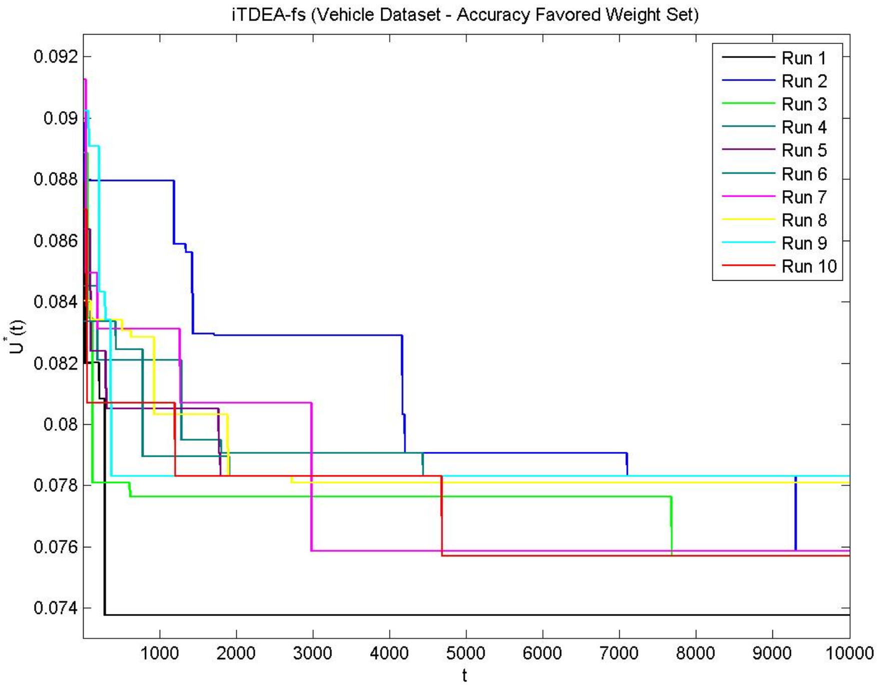

Figure D.1 Accuracy Favored – iTDEA-fs

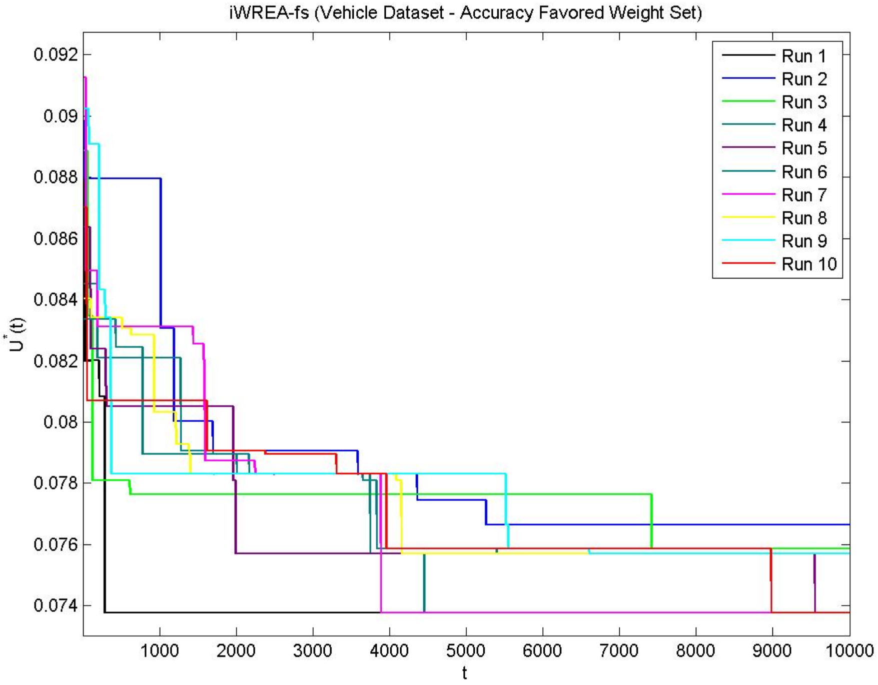

Figure D.2 Accuracy Favored – iWREA-fs

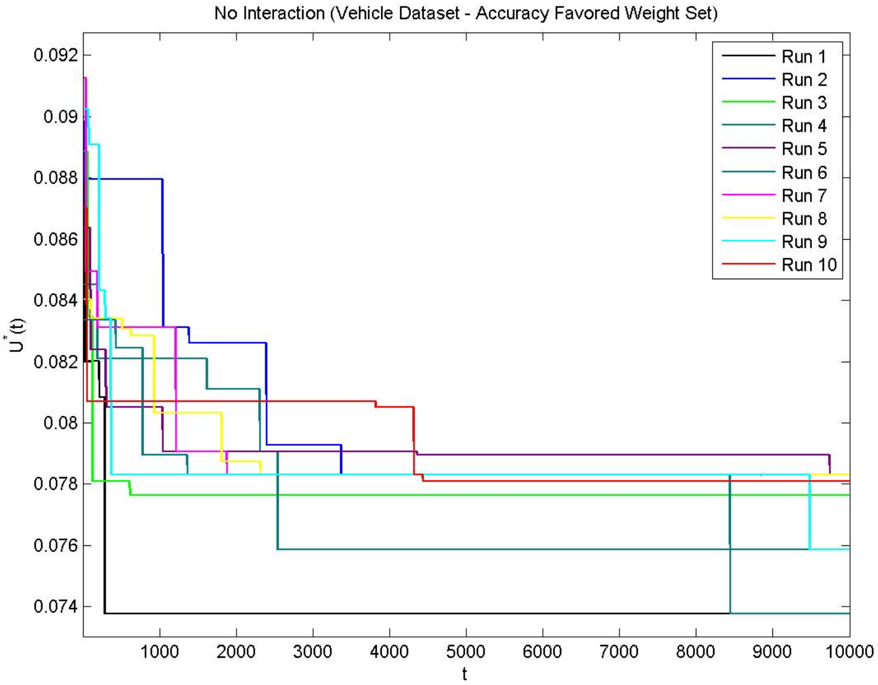

Figure D.3 Accuracy Favored – No Interaction

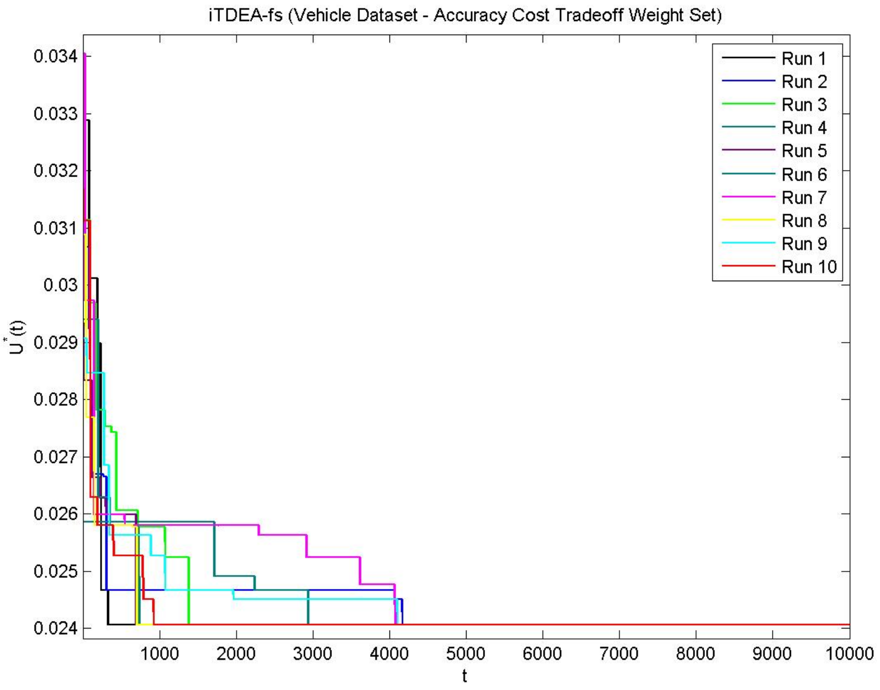

Figure D.4 Accuracy and Cost Tradeoff – iTDEA-fs

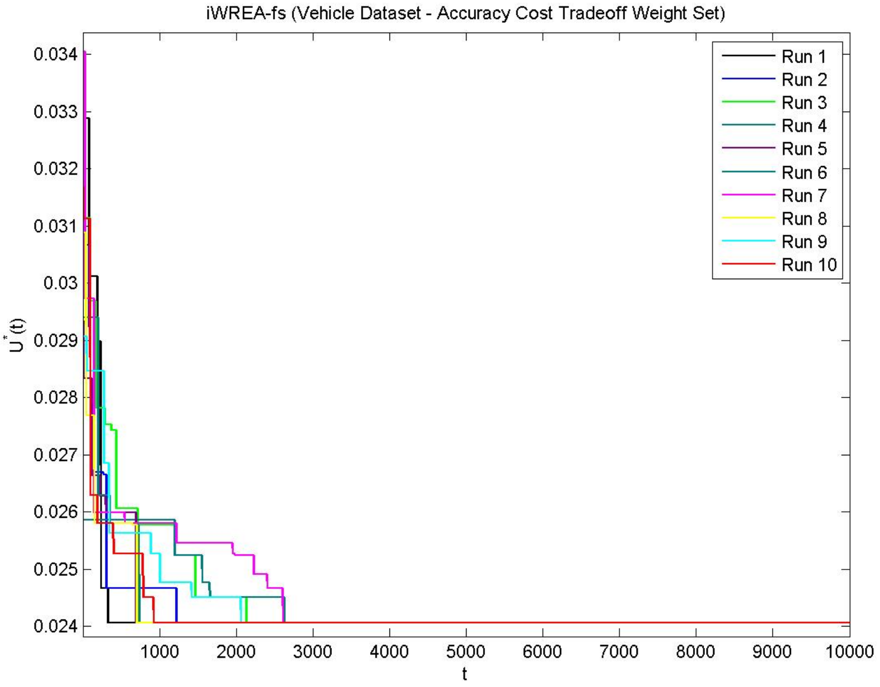

Figure D.5 Accuracy and Cost Tradeoff – iWREA-fs

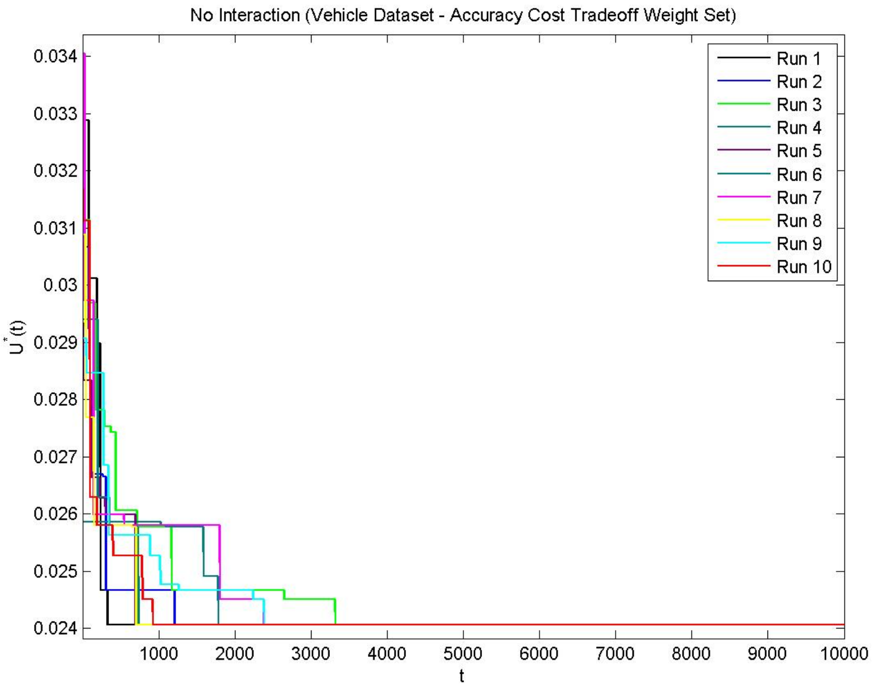

Figure D.6 Accuracy and Cost Tradeoff – No Interaction

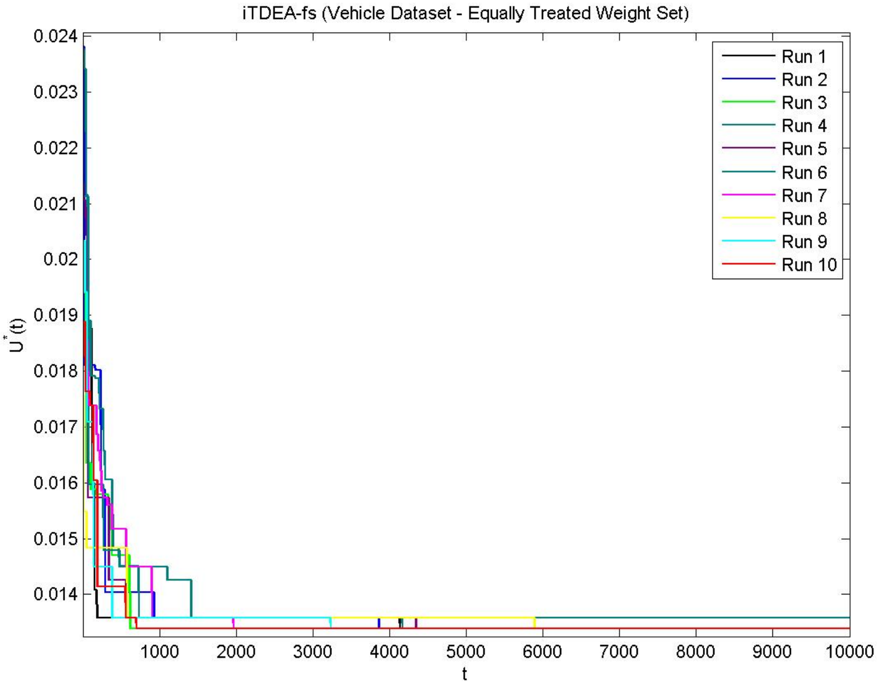

Figure D.7 Equally Treated – iTDEA-fs

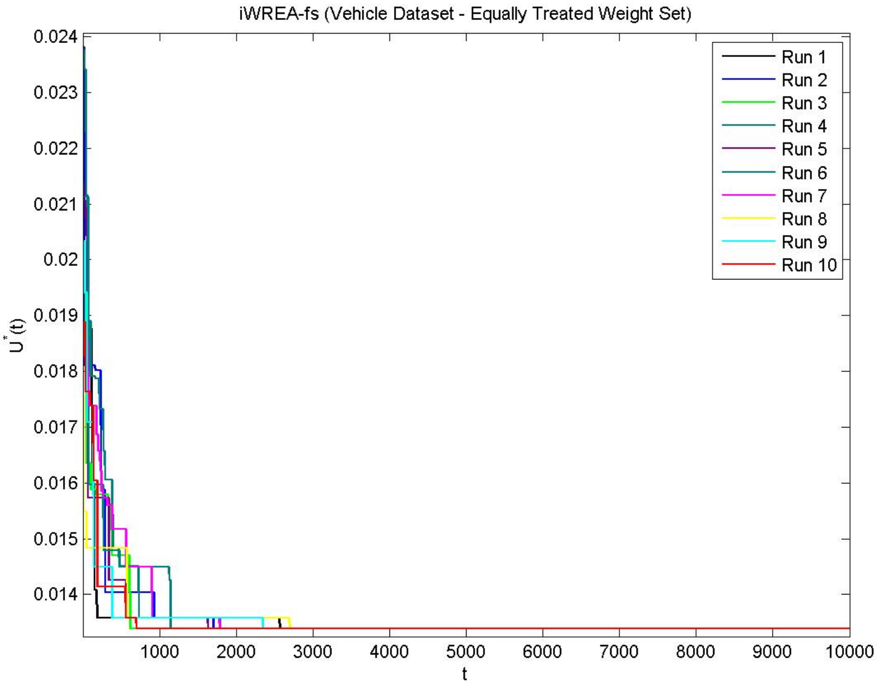

Figure D.8 Equally Treated – iWREA-fs

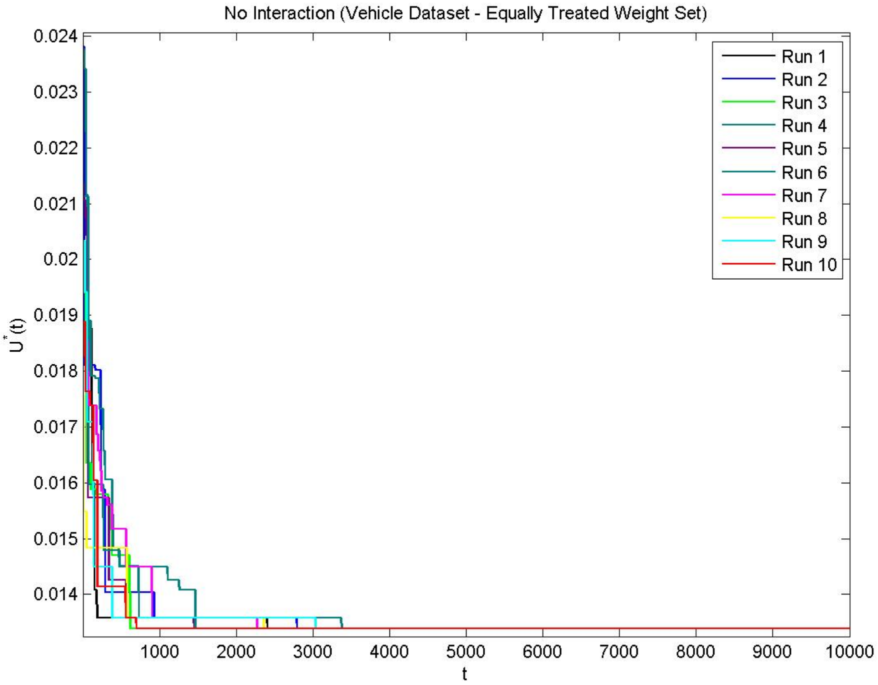

Figure D.9 Equally Treated – No Interaction