INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 0 (2017) 1–26
DOI: 10.1111/itor.12428

Interactive evolutionary approaches to multiobjective feature selection

Müberra Özmen[a], Gülşah Karakaya[b] and Murat Köksalan[a]

[a]*Department of Industrial Engineering, Middle East Technical University, 06800 Ankara, Turkey*
[b]*Department of Business Administration, Middle East Technical University, 06800 Ankara, Turkey*
E-mail: muberra.ozmen@gmail.com [Özmen]; kgulsah@metu.edu.tr [Karakaya]; koksalan@metu.edu.tr [Köksalan]

Received 30 September 2016; received in revised form 20 April 2017; accepted 24 April 2017

Abstract

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 multiobjective 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. Therefore, we develop interactive evolutionary approaches that aim to converge to a subset that is highly preferred by the decision maker (DM). We test our approaches on several instances simulating DM preferences by underlying preference functions and demonstrate that they work well.

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

1. Introduction

In classification problems, supervised learning algorithms, such as decision trees, support vector machines (SVMs), neural networks, etc. are used to predict the class (or output variable) of an instance by observing its feature (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 a 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 nonredundant features. Selecting a subset that comprises strongly relevant, and weakly relevant and nonredundant 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 and Elisseeff, 2003). Therefore, feature selection methods are used in many areas, such as handwritten digit recognition (Oliveria et al., 2003), medical diagnosis (Chyzhyk et al., 2014), gene marker recognition (Banerjee et al., 2007), 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 multiobjective 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 provide 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. First, 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). Second, 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 biobjective 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 nondominated 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 nondominated solutions is not practical and useful because of the combinatorial nature of the problem. Instead of finding all nondominated 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 multiobjective feature selection problems that aim to converge the most preferred solution by guiding the search toward 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 use an existing supervised learning algorithm for this purpose. Our contribution is rather in developing

a multiobjective optimization approach that is compatible with the characteristics of the feature selection problem.

The remainder of the paper is organized as follows. In Section 2, main concepts and definitions regarding the feature selection problem are provided and a literature review of related studies is given. In Section 3, interactive algorithms to find a preferred solution of the DM are developed. In Section 4, the algorithms are tested computationally on several datasets. Concluding remarks and future research directions are presented Section 5.

2. Main concept and definitions

In this section, first basic concepts and definitions regarding the feature selection problem and multi-objective optimization are provided. Then the relevant literature is discussed and the problem formulation is presented.

2.1. Feature selection problem

In this study, the classification problems in which each instance is classified in only one of the nonoverlapping classes are addressed. The classification problems with two and multiple nonoverlapping classes are called as *binary class* and *multiclass* classification problems, respectively (Sokolova and Lapalme, 2009).

Let the *dataset* consist of N instances. Assuming there exist M available features defined as a vector $X = \{x_1, \ldots, x_M\}$ and a class variable y, the dataset consists of the value of y and the corresponding X vector for each of the N instances. Let S be a subset of X and $f(S)$ denote the classification performance of using the features in S.

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

$$\max f(S)$$

s.t.

$$S \in X.$$

That is, we try to select the subset of features that maximizes the classification performance. Although classification performance cannot be measured exactly, it can be estimated. For estimation purposes, the dataset is divided into *training* and *testing sets*. Once the learning algorithm is trained on the training set, it is used to determine the classes of instances in the testing set. In this paper, we use the k-fold cross-validation procedure of Kohavi and John (1997) in order to reduce the effect of the specific training and testing sets chosen.

The classification performance of a subset of features, namely $f(S)$, can be measured in terms of different indicators comparing the predicted and actual classes of the instances in the testing set.

Several performance measures have been defined (see Sokolova and Lapalme, 2009). Of these, we use the *accuracy* indicator:

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn},$$

where *tp*, *tn*, *fp*, and *fn* stand for true positive, true negative, false positive, and false negative, respectively.

There are two main approaches developed for subset selection: wrapper and filter approaches. In the search phase of a subset selection algorithm, the learning algorithm itself can be directly used to evaluate a subset S and estimate $f(S)$. This approach is called the *wrapper approach* (Kohavi and John, 1997).

Using the wrapper approach can be computationally time-consuming since it employs the learning algorithm to evaluate each subset found during search. Using statistical measures (such as correlation and information theoretic measures) instead of a learning algorithm to estimate the classification performance during the search phase is called the *filter approach* (Kohavi and John, 1997). Although the filter approach is computationally more efficient, the wrapper approach provides more reliable estimation of classification performance.

2.2. Multiobjective optimization

In multiobjective 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 $z_1(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x})$ to be minimized and Z be the objective space defined by the feasible decision vectors. The general multiobjective optimization problem can be formulated as follows:

$$\min \left\{ z_1(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x}) \right\}$$

s.t.

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

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* $\boldsymbol{z}(\boldsymbol{x}') = (z_1(\boldsymbol{x}'), \ldots, z_p(\boldsymbol{x}'))$ *is said to dominate* $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x}))$, *if and only if* $z_j(\boldsymbol{x}') \leq z_j(\boldsymbol{x})$ *for all* $j = 1, \ldots, p$ *and* $z_j(\boldsymbol{x}') < z_j(\boldsymbol{x})$ *for at least one.*

Definition 2.2.2. $\boldsymbol{z}(\boldsymbol{x})$ *is nondominated, if and only if no* $\boldsymbol{z}(\boldsymbol{x}')$ *dominates it.*

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

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

In this study, interactive evolutionary algorithms that aim to converge the DM's preferred solutions are developed for multiobjective 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 $U_{DM}(\boldsymbol{z}(\boldsymbol{x})) < U_{DM}(\boldsymbol{z}(\boldsymbol{x}'))$.

In the feature selection problem, the number of possible subsets of features grows exponentially with the number of available features; 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 for feature selection problem, see Xue et al. (2016). We next discuss the literature that aims to find Pareto-optimal solutions utilizing different evolutionary algorithms in the context of feature selection problems.

Oliveira et al. (2002) use nondominated sorting genetic algorithm (NSGA) developed by Srinivas and Deb (1995), for feature selection in handwritten digit recognition. They approximate the nondominated solutions minimizing cardinality and maximizing accuracy. Among the solutions that satisfy a minimum acceptable accuracy level, they choose the solution having minimum cardinality. Hamdani et al. (2007) also use the same two objectives and employ NSGA II, developed by Deb et al. (2002), as the search engine. Xue et al. (2013) employ particle swarm optimization for minimizing cardinality and maximizing accuracy and show that it works well.

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 nonchurn. They evaluate the classification performance maximizing overall accuracy, sensitivity, specificity, and minimizing cardinality. Many researchers try to approximate the Pareto-optimal set minimizing cardinality and maximizing classification performance in addressing the feature selection problem. Karakaya et al. (2016a) introduce the term "quasi equally informative subsets" into this problem to find alternative subsets that have similar classification performances for each cardinality level.

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 (see, e.g., Bolón-Canedo et al., 2014; Zhang et al., 2015).

2.3. Problem formulation

Let the DM have p objectives to be minimized in the feature selection problem. Then the general problem can be formulated as

$$\min \left\{ z_1(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x}) \right\}$$

s.t.

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

$$x_i \in \{0, 1\} \quad \forall\, i = 1, \ldots, M,$$

where x_i represents whether feature i is selected or not, M is the number of available features, $z_j(\boldsymbol{x})$ represents the value of the jth objective corresponding to solution $\boldsymbol{x} = (x_1, \ldots, x_M)$. Constraint (1) ensures the selected subset will include at least one feature.

2.4. Objective functions

Many studies in feature selection either consider a single objective (maximizing classification performance) or two objectives (maximizing classification performance and minimizing cardinality). There could be other relevant objectives such as minimizing cost and risk. In the remainder of this paper, we address these four objectives. The methodology, however, is applicable to any number of objectives. We linearly scale all four objectives between 0 and 1 in order to avoid problems that may arise from different scales.

Accuracy
Classification performance is an important objective in the feature selection problem and a commonly used measure is accuracy. A common way of estimating accuracy is employing a learning algorithm. Since we treat all objectives as minimization type, we transform accuracy by

$$z_1(\boldsymbol{x}) = 1 - f(\boldsymbol{x}), \tag{2}$$

where $f(\boldsymbol{x})$ is the accuracy achieved for the selected features in solution $\boldsymbol{x}$ and $z_1(\boldsymbol{x})$ is an accuracy measure to be minimized.

Cardinality
As mentioned before, decreasing the cardinality of the subset used in the prediction model is desirable in terms of reducing storage requirements and improving the time efficiency. We assume that at least one feature is used in a solution (see Equation (1)) in order to have a meaningful problem.

Cost
In some classification problems, the features may be grouped such that each group has a fixed investment cost and each feature has an additional measuring/monitoring cost. For example, it is possible to perform a blood test (comprising a group of features) and an MRI scan (comprising another group of features) on a patient. Both tests have fixed costs and each additional feature measured from each test produces an additional variable cost. The total screening cost of a subset is incurred based on the specific tests that are performed (fixed costs) and the specific features that are measured (variable costs).

Risk
In this study, we consider risk as a feature-based attribute. While each feature equally affects cardinality, their effects on risk could vary. For example, in a medical diagnosis problem, features

(such as different medical tests) may have associated health-related risks for the patient. Minimizing the risks the patient is exposed to is a relevant concern. We measure risk by summing the risk values of the features in the selected subset. Alternatively, in a different context, each combination of the features may provide a different level of uncertainty and a measure of risk may be defined to account for this uncertainty.

3. Algorithms

In this section, we develop two algorithms: iTDEA-fs (interactive territory defining evolutionary algorithm for feature selection problem) and iWREA-fs (interactive weight-reducing evolutionary algorithm for feature selection).

3.1. Overview

Our algorithms are built upon the idea of iTDEA developed by Köksalan and Karahan (2010). The algorithm obtains preference information and converges toward preferred regions of the DM progressively.

In iTDEA, territories are used to direct the search toward the most preferred region. A territory is defined for each solution within which no other solution is allowed. As such, the territories prevent congestion and in turn facilitate diversity. Defining smaller territories in preferred regions, it is possible to increase the relative density of solutions in those regions.

In iTDEA, two populations are maintained throughout the iterations; *the regular population* and *the archive*. Initially, a regular population of N random solutions are generated. The nondominated members of that population form the initial archive. Both archive and regular population are updated throughout the algorithm. The size of the regular population, N, is kept constant, whereas the archive size is flexible.

The number of iterations, T, and the number of interaction stages, H, are set at the beginning. The interaction stages $h = 1, \ldots, H$ are scheduled at iterations $G_1, \ldots, G_H$. At each regular iteration, one offspring is generated from two parents. Then, the offspring may be accepted to the regular population and/or to the archive and the corresponding sets are updated accordingly. At each interaction stage, the DM is presented a set of solutions and is asked to select the best among them. The preference information is updated accordingly. The algorithm stops when the maximum number of iterations, T, is reached. In order to select a specific solution, there is a need to make a final search within the archive.

The general framework of iTDEA is as follows.

1. Set iteration counter $t = 0$ and interaction counter $h = 0$. Schedule interaction stages at iterations $G_1, \ldots, G_H$.
2. Generate the initial regular population $P(0)$ of size N, and find the nondominated solutions in the population to form the initial archive $A(0)$.
3. Set $t \leftarrow t + 1$, $h \leftarrow h + 1$, $P(t) = P(t-1)$, and $A(t) = A(t-1)$.
4. *Offspring generation*: Select two parents, one from regular population and the other from archive; apply crossover and mutation to create offspring.

5. *Population update*: If the offspring satisfies the acceptance conditions to the regular population insert it into $P(t)$. Otherwise, go to step 7.
6. *Archive update*: If the offspring satisfies the acceptance conditions to the archive, insert it into $A(t)$.
7. If $t < G_h$, go to step 3.
8. *Interaction stage*: Interact with the DM and update offspring acceptance conditions according to the updated preference information.
9. If $t = T$, perform the final interaction and stop. Otherwise, go to step 3.

iTDEA-fs and iWREA-fs use the above framework and the *offspring generation* procedure of iTDEA. In both evolutionary algorithms, the chromosome representation is constructed such that each gene represents whether or not the corresponding feature is selected. After the two parents are selected, offspring is generated using uniform crossover with a crossover probability of $p_c = 0.5$, and binary mutation on each gene with a mutation probability of $p_m = 1/M$, where M is the total number of features. iTDEA-fs and iWREA-fs differ in the *population update*, *archive update*, and *interaction stages.* We discuss these in Sections 3.2 and 3.4 for iTDEA-fs and iWREA-fs, respectively.

3.2. Interactive territory defining evolutionary algorithm for the feature selection problem (iTDEA-fs)

We first provide relevant definitions in iTDEA and then discuss the details of iTDEA-fs.

Some definitions of iTDEA

In the archive update and interaction stage of iTDEA-fs, additional operations are required to calculate the objective weights that minimize the Chebychev distance of a solution from the ideal point, that is, for calculating *favorable weights*. Favorable weight vector $\hat{\boldsymbol{w}}_i = (\hat{w}_{i1}, \ldots, \hat{w}_{ip})$ of a solution $\boldsymbol{z}_i$, is calculated as follows:

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

where $\boldsymbol{z}^*$ is the ideal point and p is the number of objectives (see Steuer, 1986, p. 425). Since all objectives are scaled between 0 and 1, the ideal point can be defined as $\boldsymbol{z}^* = 0$, that is, $z_j^* = 0$, $j = 1, \ldots, p$.

At each interaction stage h, the preference information obtained from the DM is used to estimate the *preferred weight region*, R^h. A preferred weight region is defined by a set of Chebychev weight ranges $[\boldsymbol{l}^h, \boldsymbol{u}^h] = \{[l_1^h, u_1^h], \ldots, [l_p^h, u_p^h]\}$, where l_j^h and u_j^h refer to the lower and upper bounds, respectively, defined for the preference function weight of objective j (for the calculation of the bounds, see Köksalan and Karahan, 2010). Since there is no information regarding the DM's preferences

until the first interaction stage, the initial preferred weight region R^0 includes all feasible weight ranges, $[l_j^0, u_j^0] = [0,\ 1]\,,\ j\ =\ 1, \ldots,\ p$.

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, $\boldsymbol{z}_{fav}$, of them. The resulting preference information is used to update 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, $\boldsymbol{z}_{fav}$ in each interaction stage.

Each preferred weight region R^h has a territory level τ_h. As h gets larger τ_h gets smaller, as the algorithm is expected to converge to more preferred regions with every interaction stage. The smaller τ_h is, the denser the solutions are in the corresponding regions.

The interaction stage of iTDEA-fs differs from that of iTDEA only in the *filtering* procedure. iTDEA was originally implemented on problems with continuous objective space and the archive size has been big enough to find P solutions whose favorable weights fall in the preferred weight region. In our case, the objective space of the feature selection problem is discrete and P distinct solutions having favorable weights in the preferred weight region may not be available. In such cases, we keep selecting new solutions randomly among those in the archive that have not been presented to the DM, until P solutions are obtained.

iTDEA-fs population update

Each time an offspring, $\boldsymbol{z}_{\text{off}}$, is generated, it is checked for acceptance into the regular population. The steps of the population update procedure at iteration t, are as follows:

1. If $\boldsymbol{z}_{\text{off}} \in P(t)$, do not accept it into population and go to step 4.
2. Compare $\boldsymbol{z}_{\text{off}}$ against each solution in the regular population, $\boldsymbol{z}_i \in P(t)$. If $\boldsymbol{z}_{\text{off}}$ dominates any $\boldsymbol{z}_i \in P(t)$, discard $\boldsymbol{z}_i$, insert $\boldsymbol{z}_{\text{off}}$ into $P(t)$ and go to step 4.
3. Replace a randomly selected solution $\boldsymbol{z}_k \in P(t)$ with $\boldsymbol{z}_{\text{off}}$.
4. Stop.

In iTDEA, an offspring that is dominated by any regular population member is not accepted into the population. In the feature selection problem generating a nondominated solution is challenging. Therefore, we include an offspring into the population even if it is dominated so long as it is distinct from the population members, for the sake of diversity (see step 3 above).

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 the archive is given below.

1. Compare $\boldsymbol{z}_{\text{off}}$ against each $\boldsymbol{z}_i \in A(t)$. If $\boldsymbol{z}_{\text{off}}$ is dominated by any $\boldsymbol{z}_i \in A(t)$, go to step 5.
2. If there exists any $\boldsymbol{z}_i \in A(t)$ that is dominated by $\boldsymbol{z}_{\text{off}}$, discard all such solutions from $A(t)$, insert $\boldsymbol{z}_{\text{off}}$ into $A(t)$, and go to step 5.
3. Calculate the favorable weights of the $\boldsymbol{z}_{\text{off}}$, $\hat{\boldsymbol{w}}_{\text{off}} = (\hat{w}_1, \ldots, \hat{w}_p)$.
   i. Set $q = h$.
   ii. If $l_j^q \leq \hat{w}_j \leq u_j^q$ where $[l_j^q, u_j^q] \in R^q$, $j = 1, \ldots, p$, 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 distance of $\boldsymbol{z}_i \in A(t)$ to $\boldsymbol{z}_{\text{off}}$ and denote it as d_i. Set $d = \min_{\{i:\boldsymbol{z}_i \in A(t)\}} \{d_i\}$. If $d \geq \tau$, insert $\boldsymbol{z}_{\text{off}}$ into $A(t)$.
5. Stop.

Unlike iTDEA, the offspring is accepted into the archive if it is nondominated and it dominates at least one solution in the archive even if it falls into the territory of an existing solution.

A nondominated offspring that does not dominate any solution in the archive, on the other hand, is accepted only if it does not violate the territories of existing solutions. By keeping the territory levels in the estimated preferred regions small, the archive update procedure of this algorithm allows increasing the chance of survival in the population for the nondominated solutions in those regions.

3.3. Improvement issues of iTDEA-fs

Feature selection problem has two characteristics originating from the nature of the problem that requires special treatment: *scaling* and *imbalanced solution space*. In terms of scaling, it is important that all objectives have approximately the same ranges. We achieve this by linearly transforming all objectives such that they are scaled in the range [0,1].

In terms of the imbalance in the solution space, the feature selection problem poses difficulties due to the discrete nature of the solution space. For a problem having 100 features, there are 100 distinct solutions with cardinality of one and a single solution with cardinality of 100. At the cardinality value of 1, the accuracy can vary substantially depending on which single feature is selected. Consequently, the favorable weights could be misleading for a solution having cardinality of 1 and a poor accuracy level. On the other hand, there are solutions with the same cardinality and better accuracy level (other objectives being the same) that would lead to different favorable weights. Such a situation is demonstrated in the next example.

Example 1. *Consider a 3-objective minimization problem in which the objectives are Z_1, Z_2, and so on. Suppose that the archive is filtered during an interaction stage and the DM is presented as the five solutions given in Table 1.*

Assume that the DM has a preference function that minimizes the weighted Chebychev distance of a solution from the ideal point with weights $w_1 = 0.1$, $w_2 = 0.2$, and $w_3 = 0.7$. Assume, without loss of generality, that the ideal point is 0 for each objective. That is, the DM minimizes

$$\max\left\{0.1Z_1,\ 0.2Z_2,\ 0.7Z_3\right\}.$$

Table 1
Filtered archive of Example 1

Solution	Objective value			
	Z_1	Z_2	Z_3	U_{DM}
Y_1	0.10	0.40	0.40	0.280
Y_2	0.10	0.30	0.50	0.350
Y_3	0.20	0.30	0.45	0.315
Y_4	0.20	0.20	0.70	0.490
Y_5	0.30	0.10	0.70	0.490

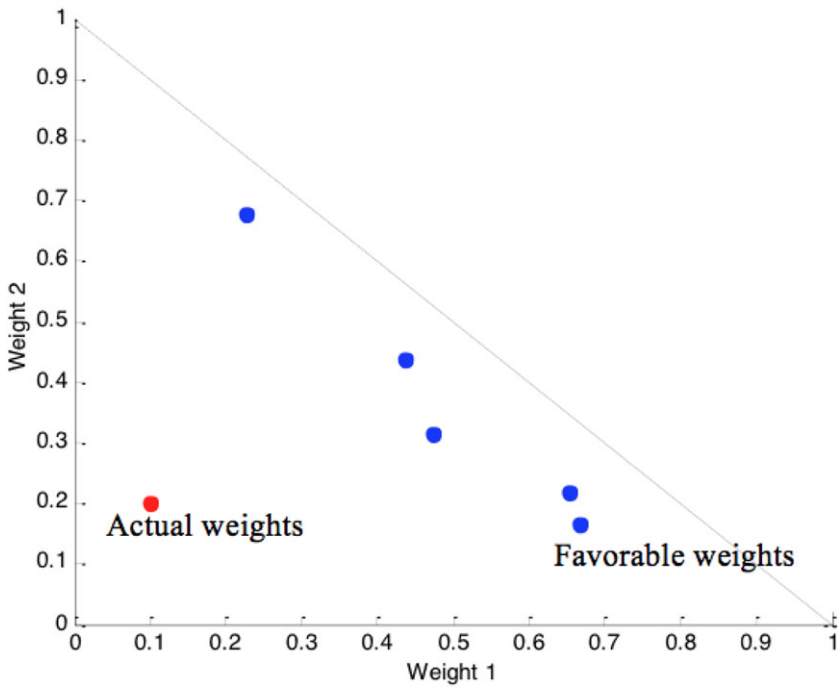

Fig. 1. Favorable weights and actual weights in the example. [Colour figure can be viewed at wileyonlinelibrary.com]

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, which are far from the actual weights as shown in Fig. 1. Once the favorable weights differ from the actual weights, the algorithm reduces the weight space in the wrong region and it may not be possible to recover in later stages.

To avoid a poor estimation of the weights, Karakaya et al. (2016b) developed a mixed integer mathematical model, Model (Mid_∞), to evaluate the DM's preferences. We utilize this model (given below) in iWREA-fs. The model aims to find a weight set that has a central location in the feasible Chebychev weight region constructed using the past preferences of the DM.

Assuming that the ideal point is 0 for each objective, let z_{ij} be the jth objective value of solution $\boldsymbol{z}_i$ and L be the set of past pairwise comparisons of the DM, where $L = \{(\boldsymbol{z}_m, \boldsymbol{z}_l) : \boldsymbol{z}_m \text{ is preferred to } \boldsymbol{z}_l\}$. 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_∞)

$$\max \varepsilon$$

s.t.

$$\hat{w}_t z_{lt} \geq \hat{w}_s z_{ms} + \varepsilon - M\left(1 - y_t\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right)\right),$$

$$\forall t \in I^-_{z_m,z_l},\ \forall s \in I^+_{z_m,z_l}, \quad \forall\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right) \in L, \tag{3}$$

$$\sum_{t \in I^-_{z_m,z_l}} y_t\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right) \geq 1, \quad \forall\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right) \in L, \tag{4}$$

$$\sum_{j=1}^{p} \hat{w}_j = 1, \tag{5}$$

$$\hat{w}_j \geq \varepsilon, \quad \forall j, \tag{6}$$

$$y_t\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right) \in \{0,\ 1\},\ \forall t \in I^-_{z_m,z_l}, \quad \forall\left(\boldsymbol{z}_m, \boldsymbol{z}_l\right) \in L,$$

where $\hat{w}_j$ represents the estimated weight of the jth objective and $y_t(\boldsymbol{z}_m, \boldsymbol{z}_l)$ is a binary decision variable (for a detailed explanation of the model, see Karakaya et al., 2016b). It is important to note that the constraint set of Model (Mid_∞) guarantees to contain the actual weights of the DM, in contrast with the reduced weight region obtained from the favorable weights of iTDEA-fs. Corresponding to each new preference information obtained from the DM, a new constraint set (3) is enforced, which restrict the feasible weight space further. As the amount of preference information increases, the weight space reduces and the optimal solution of the model $\hat{\boldsymbol{w}} = \{\hat{w}_1, \ldots, \hat{w}_p\}$ converges toward the actual weights of the DM.

Recall that the DM selects Y_1 over Y_2, Y_3, Y_4 , and Y_5 in Example 1. Using this preference list, Model (Mid_∞) estimates the DM's objective weights as $\hat{w}_1 = 0.19$, $\hat{w}_2 = 0.19$, and $\hat{w}_3 = 0.62$. With the same information, Model (Mid_∞) is able to estimate the actual weights of the DM much better than the favorable weights, as demonstrated in Fig. 2.

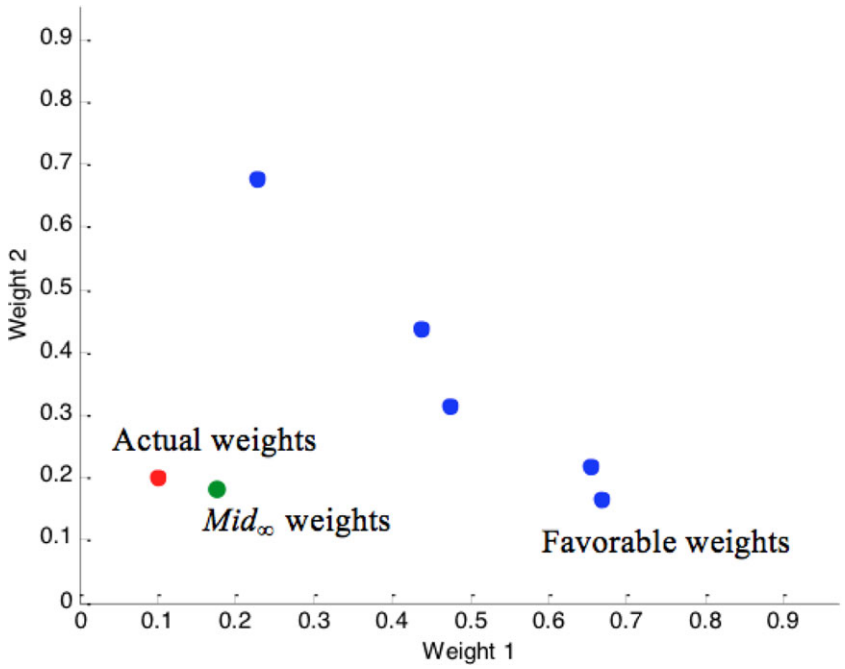

Fig. 2. Model (Mid_∞) weights, favorable weights, and actual weights of Example 1. [Colour figure can be viewed at wileyonlinelibrary.com]

3.4. *Interactive weight-reducing evolutionary algorithm for feature selection (iWREA-fs)*

In this section, we develop a new interactive evolutionary algorithm, iWREA-fs, which improves iTDEA-fs in several aspects.

iWREA-fs interaction stages

In iWREA-fs, at each interaction stage, the DM is asked to compare the best-known solution so far (the incumbent solution), $\boldsymbol{z}_{inc}$, and a selected solution, $\boldsymbol{z}_s$. Q constraints are constructed based on the preferences of the DM to further restrict the weight space in Model (Mid_∞). In the first interaction stage, $h = 1$, the preference list is initialized to $L = \emptyset$. L is updated with new preference pairs throughout the interaction stages. The estimated weights are updated with every new preference information and are carried between interaction stages. The steps of an interaction stage, h, are as follows:

1. Set the question counter $q = 0$.
2. If $L = \emptyset$, select two distinct random solutions, $\boldsymbol{z}_i,\ \boldsymbol{z}_k \in A(t)$. Ask the DM to compare $\boldsymbol{z}_i$ with $\boldsymbol{z}_k$, and set $q \leftarrow q + 1$. Assume $\boldsymbol{z}_i$ is preferred to $\boldsymbol{z}_k$. Let $L = \{(\boldsymbol{z}_i, \boldsymbol{z}_k)\}$ and $\boldsymbol{z}_{inc} = \boldsymbol{z}_i$. If $L \neq \emptyset$, go to step 4.
3. Estimate the DM's preference function weights, $\hat{\boldsymbol{w}}_{DM} = (\hat{w}_1, \ldots, \hat{w}_p)$, solving Model($\text{Mid}_\infty$) with the current preference list, L.

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

$$\hat{u}(\boldsymbol{z}_i) = \max_j \left\{ \hat{w}_j z_{ij} \right\}$$

where z_{ij} represents the j^{th} objective value of solution $\boldsymbol{z}_i \in A(t)$ and $\hat{w}_j$ represents the estimated weight of the jth objective. Rank the solutions in increasing order of $\hat{u}(\boldsymbol{z}_i)$, $\{\boldsymbol{z}_{(1)}, \ldots, \boldsymbol{z}_{(|A(t)|)}\}$, where $|A(t)|$ represents the number of solutions in the current archive. Initialize rank counter $r = 1$.

5. If $(\boldsymbol{z}_{inc}, \boldsymbol{z}_{(r)}) \notin L$ and $\boldsymbol{z}_{inc} \neq \boldsymbol{z}_{(r)}$, 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 Chebychev distances of the solutions in $P(t)$ to the ideal point, as follows:

$$\hat{u}(\boldsymbol{z}_i) = \max_j \left\{ \hat{w}_j z_{ij} \right\}$$

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

7. If $(\boldsymbol{z}_{inc}, \boldsymbol{z}_{(r)}) \notin L$ and $\boldsymbol{z}_{inc} \neq \boldsymbol{z}_{(r)}$, 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 $\boldsymbol{z}_{inc}$ and $\boldsymbol{z}_S$. Let $\boldsymbol{z}_m$ denote the preferred solution and $\boldsymbol{z}_l$ the inferior solution of the pair. Set $\boldsymbol{z}_{inc} = \boldsymbol{z}_m$, update $L = L \cup \{(\boldsymbol{z}_m, \boldsymbol{z}_l)\}$. Set $q \leftarrow q + 1$. If $q = Q$ estimate the DM's preference function weights $\hat{\boldsymbol{w}}_{DM}$ solving Model (Mid_∞) with the current preference list, L, and stop. Otherwise, go to step 3.

In the interaction stages of iWREA-fs, we try to reduce the weight space substantially by comparing solutions that have close estimated preference values as measured by the estimated weighted Chebychev function. In iTDEA-fs, in each interaction stage, the DM is required to choose the best of P solutions and this corresponds to making $P - 1$ pairwise comparisons among them. In our approach, we also require $P - 1$ pairwise comparisons in each interaction stage. However, we decide on the pairs to be compared sequentially, utilizing the recent preference information.

iWREA-fs population update

iWREA-fs uses regular population updating rules to direct the search toward appealing regions of the solution space. When an offspring $\boldsymbol{z}_{\text{off}}$ is generated at iteration t, the regular population is updated using the following procedure:

1. Calculate the Chebychev distance of the offspring to the ideal point:
$\hat{u}(\boldsymbol{z}_{\text{off}}) = \max_j \{\hat{w}_j z_j\}$ where z_j represents the jth objective value of $\boldsymbol{z}_{\text{off}}$.
2. Calculate the Chebychev distances of the solutions in $P(t)$ to the ideal point:
$\hat{u}(z_i) = \max_j \{\hat{w}_j z_{ij}\}$ where z_{ij} represents the jth objective value of solution $\boldsymbol{z}_i \in P(t)$.
3. Rank the solutions in increasing order of $\hat{u}(\boldsymbol{z}_i)$ as $\{\boldsymbol{z}_{(1)}, \ldots, \boldsymbol{z}_{(|P(t)|)}\}$, where $|P(t)|$ represents the number of solutions in the current regular population.
4. If $\hat{u}(\boldsymbol{z}_{\text{off}}) > \hat{u}(\boldsymbol{z}_{(|P(t)|)})$ do not accept the offspring into $P(t)$. Otherwise, discard $\boldsymbol{z}_{(|P(t)|)}$ and accept the offspring into $P(t)$.

As in iTDEA, iWREA-fs may keep dominated solutions in $P(t)$. However, while in iTDEA the search is directed by $A(t)$ with the acceptance rules, in iWREA-fs the search is directed by $P(t)$.

Table 2
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

The generated offspring replaces the worst solution in $P(t)$ (in terms of the estimated preference function values) if its estimated preference function value is better.

iWREA-fs archive update

After an offspring, $\boldsymbol{z}_{\text{off}}$, is accepted into $P(t)$, it is tested against each $\boldsymbol{z}_i \in A(t)$ in order to decide whether it will be accepted into $A(t)$. If $\boldsymbol{z}_{\text{off}}$ is dominated by at least one $\boldsymbol{z}_i \in A(t)$, it does not enter $A(t)$. If $\boldsymbol{z}_{\text{off}}$ is not dominated by any $\boldsymbol{z}_i \in A(t)$, $\boldsymbol{z}_{\text{off}}$ is placed into $A(t)$ and all solutions dominated by $\boldsymbol{z}_{\text{off}}$, if any, are eliminated from $A(t)$.

Different from iTDEA-fs, in iWREA-fs we keep all nondominated solutions in the archive, regardless of their proximity to existing solutions, in order to favor all nondominated solutions.

4. Computational experiments

In this section, 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.

4.1. Datasets

The algorithms are implemented on four datasets from University of California (UCI) machine learning repository (http://archive.ics.uci.edu/ml/). The number of features, number of classes, and the number of observations of each dataset are given in Table 2.

Heart Disease and *Breast Cancer* datasets are examples of classification problems in the medical diagnosis area whose purposes are to diagnose the diseases of patients. In the *Vehicle* dataset, the features extracted by processing a vehicle's image are used to categorize the vehicle. *German* dataset aims to classify the customers of a bank using past data about customers.

In addition to the available objectives of the datasets, we use two additional objectives: cost and risk. We need to generate the values of these objectives for each data entry. It is natural to have cost and risk objectives to be in conflict with the accuracy objective. We generate the values of cost and risk in relation with the accuracy of a given feature. That is, when a feature has a positive impact on the accuracy, then we generate its cost and risk values to make sure that they are on the high side.

To demonstrate the generation of cost and risk values, we use an example dataset with five features. The DM aims to maximize accuracy and minimize the risk and the cost of the selected subset. Suppose that features 1 and 2 perform well, feature 3 performs moderately, and features 4 and 5 perform poorly in terms of accuracy. The features that have close performances in accuracy are grouped together and fixed costs of the groups are generated in proportion 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 cost of a feature in a group is generated randomly within an interval defined by the accuracy level of the feature. We generate the individual risk levels uniformly from in an interval that is defined by the variable costs of the features. Heart Disease dataset in the UCI repository includes the fixed and variable cost information of the features and these original values are used as cost parameters in our experiments.

4.2. Implementation

There are many techniques used in supervised learning algorithms such as *extreme learning machines* (*ELMs*), *decision trees*, *k-nearest neighbor*, *SVMs*, and *artificial neural networks* (for a review of supervised learning algorithms, see Kotsiantis, 2007). ELMs estimate the accuracy level of a subset of features using a single hidden layer feedforward neural network (see Huang et al., 2006). We use ELMs as the learning algorithm in our study as they have been argued to work well in the literature (Huang et al., 2012). We use 10-fold cross-validation to determine the training and test sets, and repeat this procedure five times in order to reduce variation in accuracy level estimation caused by the random nature of ELMs.

To be able to observe the effect of employing the DM's preferences, a version, *No Interaction*, in which the number of interaction stages is set to 0, is also tested on each dataset in addition to iTDEA-fs and iWREA-fs.

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, we consider a DM who would like to minimize weighted distance of a point from the ideal point in the objective space (where the function is unknown to us). Since all objectives are scaled between 0 and 1, the ideal point can be defined as 0 for each objective. In our experiments, we use two different underlying preference functions: Chebychev (Equation (7)) and quadratic (Equation (8)):

$$U_{DM}(\boldsymbol{z}) = \max\left\{w_1 z_1,\ w_2 z_2,\ w_3 z_3,\ w_4 z_4\right\} \tag{7}$$

$$U_{DM}(\boldsymbol{z}) = \left(w_1 z_1\right)^2 + \left(w_2 z_2\right)^2 + \left(w_3 z_3\right)^2 + \left(w_4 z_4\right)^2, \tag{8}$$

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.

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* (*AF*), *Accuracy and Cost Tradeoff* (*ACT*), *Equal Weights* (*EW*). In Table 3, the objective weights in $U_{DM}(\boldsymbol{z})$ for each underlying preference function are given as $\boldsymbol{w} = (w_1,\ w_2,\ w_3,\ w_4)$,

Table 3
Types of DM's preference function weights tested

Test name	Weight set
Accuracy Favored (AF)	(0.97, 0.01, 0.01, 0.01)
Accuracy and Cost Tradeoff (ACT)	(0.40, 0.10, 0.40, 0.10)
Equal Weights (EW)	(0.25, 0.25, 0.25, 0.25)

Table 4
Evolutionary algorithms' parameter settings

	Heart Disease (13)			Vehicle (18)			German (24)			Breast Cancer (32)		
Parameter	AF	ACT	EW	AF	ACT	EW	AF	ACT	EW	AF	ACT	EW
Population size, N	50	50	50	200	200	200	500	500	500	1K	1K	1K
Iterations, T	600	200	200	10K	10K	10K	20K	6K	6K	20K	10K	10K
Interactions, H	6	4	4	10	10	10	10	3	3	20	10	10
Comparisons, Q	3	3	3	3	3	3	5	5	5	5	5	5

where w_1, w_2, w_3, and w_4 refer to the weights of accuracy, cardinality, cost, and risk objectives, respectively.

4.3. Experimental setting

The algorithms are tested on each dataset for Chebychev and quadratic preference functions of the DM. In Table 4, the evolutionary algorithms' parameter settings are given for each experiment.

Within an experimental setting, iTDEA-fs, iWREA-fs, and No Interaction share the same parameter setting for the population size, N, and number of iterations, T. A common number of interaction stages, H, are used for iTDEA-fs and iWREA-fs. The interactions with the DM are scheduled in equal intervals for iTDEA-fs and iWREA-fs in each experiment. That is, $G(h) = (\frac{T}{H}) \cdot h$ for $h = 1, \ldots, H$. It is also possible to set an adaptive scheduling procedure for interactions such that the DM is consulted whenever new 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 Table 4 refers to 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 $P = Q + 1$. This setting guarantees iTDEA-fs and iWREA-fs to employ the same number of interactions in each experiment. In the AF weight set experiments of Heart Disease dataset, for example, the DM is consulted every 100 iterations. At each interaction stage, the DM is asked to make three pairwise comparisons with both iTDEA-fs and iWREA-fs.

We set the same population size and the number of pairwise comparisons for each algorithm for each dataset as shown in Table 4. We determine the number of iterations and the number of interaction stages so that the algorithms enjoy the same experimental settings. The parameter settings of different problems are tried to be chosen to account for the difficulties caused by the sizes of the corresponding problems.

In practice, the number of interactions made with the DM might be a concern and assuming the DM would be available for excessive interactions could be unrealistic. Although what is excessive could change from DM to DM, keeping the interactions below 20, for example, could be a good rule of thumb. Many interactive multiple criteria decision-making methods have considered keeping low the amount of preference information asked from DM. There may also be alternative ways of when to conduct the interactions. For example, instead of fixing the interaction schedule at the beginning, one may want to set rules based on the progress of the search process such as interacting when solutions close in estimated utility value are found. One may wish to terminate when the improvement of the algorithm becomes small for a number of consecutive iterations or when a predefined interaction limit is reached, whichever comes first. Alternatively, one may terminate when the DM is satisfied with a solution. Many of these procedures would depend on the complexity of the problem as well as the availability of the DM.

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 the number of questions asked to the DM are increased, as we did in our experimental settings.

4.4. 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.

Performance indicator

The performance of algorithms on finding an appealing solution for the DM in an experiment can be evaluated based on the best solution in the final archive $U^*(T) = \min_{z_i \in A(T)} \{U_{DM}(\boldsymbol{z}_i)\}$. 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.

In the feature selection problem, it is not possible to find the nadir and ideal points without total enumeration of possible subsets. In order to define a normalized performance indicator, for each experimental setting, the best (U_{MIN}) and worst (U_{MAX}) performance values obtained by the algorithms in 10 replications are recorded as shown in Equations (9) and (10).

$$U_{MAX} = \max_{r=1,\ldots,10} \left\{ \max \left\{ U^r_{\text{iTDEA-fs}}, U^r_{\text{iWREA-fs}}, U^r_{\text{No Interaction}} \right\} \right\} \tag{9}$$

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

Table 5
Mean and standard deviation of the percentage deviations in Chebychev preference function experiments

		iTDEA-fs		iWREA-fs		No Interaction	
	Weight set	Mean	SD	Mean	SD	Mean	SD
Heart Disease (13)	AF	0.3887	1.2051	0.0402	0.3810	0.0919	0.4437
	ACT	0.2086	1.0185	0.0000	0.0000	0.1756	1.1241
	EW	0.1543	1.0279	0.0241	0.2289	0.1497	0.7951
Vehicle (18)	AF	0.6670	1.0437	0.1523	0.7501	0.7522	1.0092
	ACT	0.5263	1.2447	0.0000	0.0000	0.5861	0.8261
	EW	0.8000	1.2649	0.4000	1.5492	0.8000	1.2649
German (24)	AF	0.7166	0.6888	0.2811	0.7127	0.6716	0.9189
	ACT	0.1176	0.9339	0.0117	0.0740	0.0176	0.0848
	EW	0.2106	1.2521	0.0409	0.3263	0.2162	1.0046
Breast Cancer (32)	AF	0.4436	0.9839	0.0449	0.4258	0.6640	0.9090
	ACT	0.3688	1.2026	0.0000	0.0000	0.4233	1.0990
	EW	0.1996	1.0356	0.0000	0.0000	0.0195	0.0942

Using U_{MAX} and U_{MIN}, the performance of algorithms in a replication is evaluated as percentage deviations, which are defined in Equations (11)–(13).

$$\Delta^r_{\text{iTDEA-fs}} = \frac{U^r_{\text{iTDEA-fs}} - U_{MIN}}{U_{MAX} - U_{MIN}} \tag{11}$$

$$\Delta^r_{\text{iWREA-fs}} = \frac{U^r_{\text{iWREA-fs}} - U_{MIN}}{U_{MAX} - U_{MIN}} \tag{12}$$

$$\Delta^r_{\text{No Interaction}} = \frac{U^r_{\text{No Interaction}} - U_{MIN}}{U_{MAX} - U_{MIN}}. \tag{13}$$

Results for Chebychev preference functions

The mean and standard deviation of the percentage deviations of each algorithm from the best value on each experimental setting are given in Table 5 for Chebychev preference function experiments. The mean of percentage deviations is 0 in some experimental settings, which indicates that the corresponding algorithm found the best solution of the three algorithms in 10 runs, U_{MIN}, in all replications. It is observed that in some experiments, iWREA-fs is able to converge the best solution found in all the replications.

Ninety-five percent confidence intervals are constructed for 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 exist statistically significant differences between means (see Table 6). The results in which the algorithms are statistically significantly different are bold-faced. The results indicate that iWREA-fs performs better than both iTDEA-fs and No Interaction in many cases. Based on our preliminary experiments, it is known that Vehicle and Breast Cancer

Table 6
Ninety-five percent confidence intervals on paired differences of percentage deviations in Chebychev preference function experiments

	Weight set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Heart Disease (13)	AF	**(−0.61, −0.09)**	(−0.21, 0.10)	(−0.04, 0.63)
	ACT	(−0.45, 0.03)	(−0.44, 0.09)	(−0.38, 0.45)
	EW	(−0.33, 0.07)	(−0.29, 0.04)	(−0.20, 0.21)
Vehicle (18)	AF	**(−0.88, −0.15)**	**(−0.84, −0.36)**	(−0.36, 0.19)
	ACT	**(−0.82, −0.23)**	**(−0.78, −0.39)**	(−0.33, 0.21)
	EW	(−0.90, 0.10)	**(−0.77, −0.03)**	(−0.34, 0.34)
German (24)	AF	**(−0.62, −0.25)**	**(−0.73, −0.05)**	(−0.24, 0.33)
	ACT	(−0.33, 0.12)	(−0.02, 0.01)	(−0.13, 0.33)
	EW	(−0.41, 0.07)	(−0.38, 0.03)	(−0.13, 0.12)
Breast Cancer (32)	AF	**(−0.64, −0.15)**	**(−0.92, −0.32)**	(−0.55, 0.11)
	ACT	**(−0.66, −0.08)**	**(−0.69, −0.16)**	(−0.36, 0.25)
	EW	(−0.45, 0.05)	**(−0.04, −0.01)**	(−0.07, 0.43)

datasets and AF weight set are relatively more challenging in terms of convergence than other settings since the relevance and redundancy relations between the features are more complicated. iWREA-fs' superiority is more apparent in those cases. On the other hand, according to Table 6, there is no statistical difference between iTDEA-fs and No Interaction in all experimental settings, which will be discussed later in this section in detail.

In order to evaluate the convergence of algorithms to good solutions, we observe the progress of the best solutions in the archives of the algorithms in each replication. 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 Chebychev preference function with ACT weight set experiments of Breast Cancer dataset are shown in Figs. 3–5 for iTDEA-fs, iWREA-fs, and No Interaction, respectively. As it can be seen from those figures, iWREA-fs converges better and faster to the best solution found by the three algorithms in 10 replications.

Although it is expected that the information gathered from the DM will be useful to find appealing solutions for the DM, an observation that can be inferred from the confidence intervals given in Table 6 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 Fig. 6, the progress of the best solution for the DM in the archive through iterations, $U^*(t)$, is shown for the fifth replication of the Chebychev preference function with Accuracy–Cost Tradeoff weight set experiments of the Breast Cancer dataset. Additionally, the preference function values of the selected solutions of iTDEA-fs and the incumbent solutions of iWREA-fs at interaction stages are shown in the same figure.

As it can be observed from Fig. 6, 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

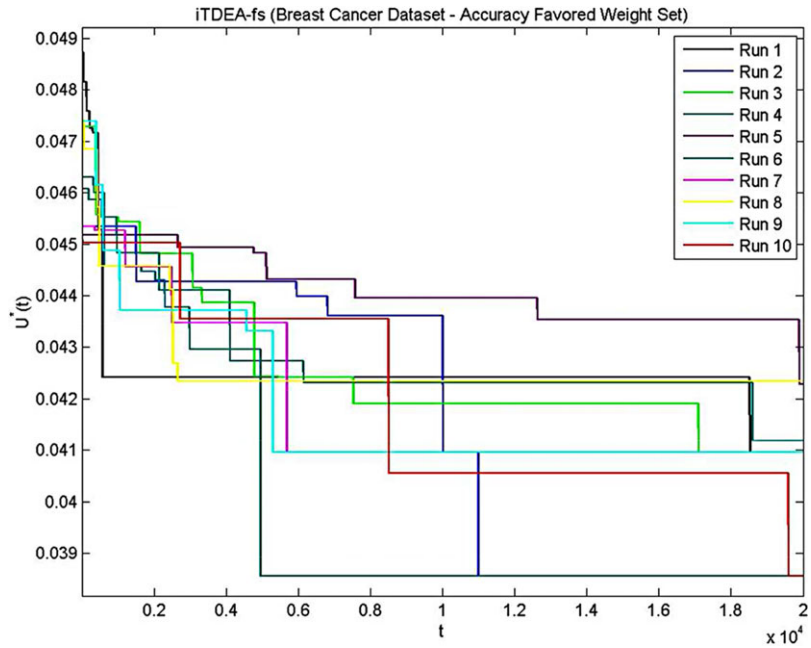

Fig. 3. Archive progress of iTDEA-fs on Chebychev preference function with ACT weight set experiments of Breast Cancer dataset. [Colour figure can be viewed at wileyonlinelibrary.com]

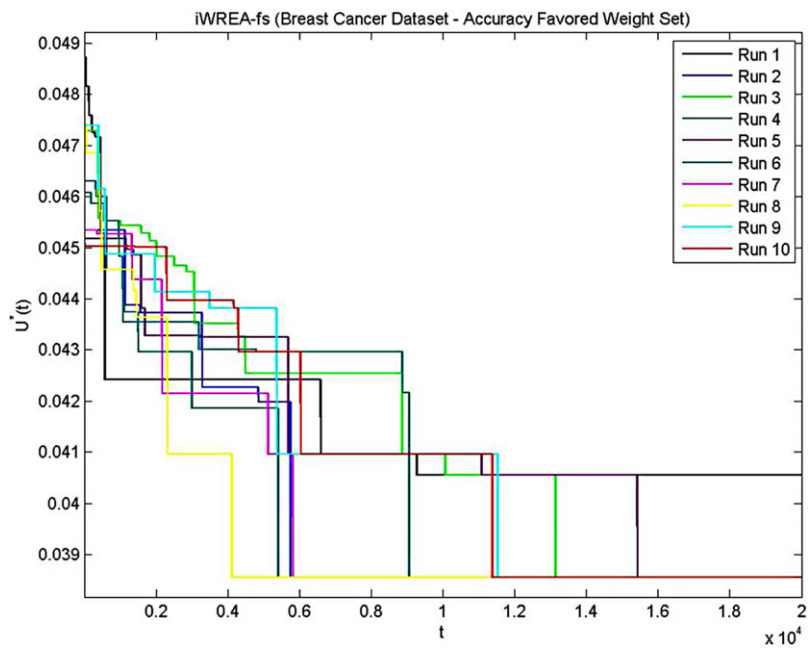

Fig. 4. Archive progress of iWREA-fs on Chebychev preference function with ACT weight set experiments of Breast Cancer dataset. [Colour figure can be viewed at wileyonlinelibrary.com]

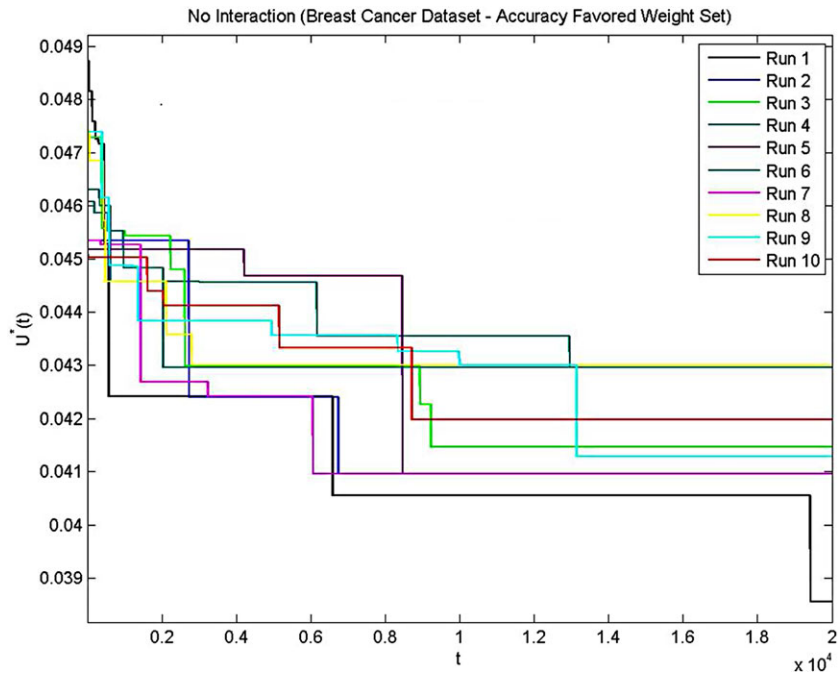

Fig. 5. Archive progress of No Interaction on Chebychev preference function with ACT weight set experiments of Breast Cancer dataset. [Colour figure can be viewed at wileyonlinelibrary.com]

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 toward the appealing region of the solution space for the DM.

On the other hand, the incumbent solution in iWREA-fs is the 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 the search accurately, this property of iWREA-fs enables to identify best solution found without additional interactions.

Results for quadratic preference functions

In order to demonstrate the performance of the algorithms for a different form of DM's underlying preference function, we repeated the experiments for a quadratic preference function to be minimized. We analyze this case with the same structure we analyzed the Chebychev preference function case. Table 7 shows that iWREA-fs outperforms the other algorithms, finding the best solution in all runs of many of the cases. Table 8 shows that the difference between iWREA-fs and other algorithms is statistically significant in many cases.

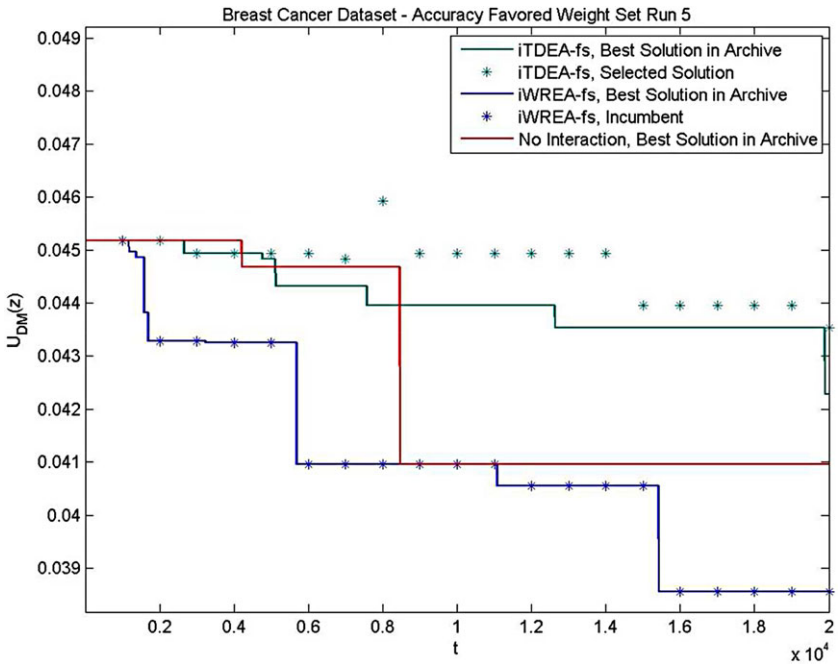

Fig. 6. Fifth replication of Chebychev preference function with ACT weight set experiments of Breast Cancer dataset. [Colour figure can be viewed at wileyonlinelibrary.com]

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 AF 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 are given in Table 9.

In the interaction stages of iWREA-fs after each question asked to the DM, Model (Mid_∞) is solved, which is a mixed integer program, 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 to the maximum of the 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

Table 7
Mean and standard deviation of the percentage deviations in quadratic preference function experiments

		iTDEA-fs		iWREA-fs		No Interaction	
	Weight set	Mean	SD	Mean	SD	Mean	SD
Heart Disease (13)	AF	0.7784	0.3944	0.1000	0.3000	0.8568	0.2980
	ACT	0.2000	0.4000	0.2000	0.4000	0.4000	0.4899
	EW	0.3000	0.4583	0.0000	0.0000	0.3000	0.4583
Vehicle (18)	AF	0.6736	0.3420	0.2381	0.2446	0.6733	0.3919
	ACT	–	–	–	–	–	–
	EW	0.1000	0.3000	0.0000	0.0000	0.0000	0.0000
German (24)	AF	0.5648	0.3936	0.1524	0.1425	0.4659	0.3963
	ACT	0.2909	0.3677	0.0000	0.0000	0.0941	0.0655
	EW	0.2320	0.2804	0.0000	0.0000	0.0890	0.0971
Breast Cancer (32)	AF	0.4160	0.3242	0.0361	0.1083	0.4661	0.3804
	ACT	0.3124	0.3994	0.0000	0.0000	0.0496	0.1487
	EW	–	–	–	–	–	–

"–" indicates that the best solution has been found in all replications by all algorithms.

Table 8
Ninety-five percent confidence intervals on paired differences of percentage deviations in quadratic preference function experiments

	Weight set	iWREA-fs vs. iTDEA-fs	iWREA-fs vs. No Interaction	iTDEA-fs vs. No Interaction
Heart Disease (13)	AF	**(−1.02, −0.34)**	**(−1.20, −0.31)**	(−0.49, 0.33)
	ACT	(−0.34, 0.34)	(−0.76, 0.36)	(−0.65, 0.25)
	EW	(−0.65, 0.05)	(−0.65, 0.05)	(−0.48, 0.48)
Vehicle (18)	AF	**(−0.72, −0.15)**	**(−0.78, −0.10)**	(−0.39, 0.39)
	ACT	–	–	–
	EW	(−0.33, 0.13)	(0.00, 0.00)	(−0.13, 0.33)
German (24)	AF	**(−0.69, −0.14)**	**(−0.62, −0.01)**	(−0.30, 0.50)
	ACT	**(−0.57, −0.01)**	**(−0.14, −0.04)**	(−0.06, 0.45)
	EW	**(−0.44, −0.02)**	**(−0.16, −0.02)**	(−0.09, 0.38)
Breast Cancer (32)	AF	**(−0.67, −0.09)**	**(−0.69, −0.17)**	(−0.44, 0.34)
	ACT	**(−0.61, −0.01)**	(−0.16, 0.06)	(0.01, 0.52)
	EW	–	–	–

"–" indicates that the best solution has been found in all replications by all algorithms.

Table 9
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

shows its effect. Therefore, the average CPU time of iWREA-fs is smaller than those of iTDEA-fs and No Interaction for the experiments of Vehicle, German, and Breast Cancer datasets.

5. Conclusions

Feature selection is an important problem as its results have major impacts 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. Noting the special characteristics of the problem, we developed a new preference-based evolutionary algorithm, iWREA-fs. In addition to the traditional objectives defined for the feature selection problem in the literature, we define additional objectives that can be useful within different contexts of the problem.

Feature selection is used in 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 could 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 preferred solutions of the DM with iWREA-fs. To the best of our knowledge, this is the first study that uses an interactive approach and considers additional objectives together with the traditional ones for the feature selection problem.

There may be many different variations in the actual implementation of our approach, especially in the interaction schedule and termination conditions. These may depend on both problem context and DM. The objectives to be used are also highly context dependent. We intend to apply our approach in different practical problems with real DMs and try different variations in these applications as future research.

References

Banerjee, M., Mitra, S., Banka, H., 2007. Evolutionary rough feature selection in gene expression data. *IEEE Transactions On Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37, 4, 622–632.

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.

Chyzhyk, D., Savio, A., Graña, M., 2014. Evolutionary ELM wrapper feature selection for Alzheimer's disease CAD on anatomical brain MRI. *Neurocomputing* 128, 73–80.

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.

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., 2016a. 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., Ahipaşaoğlu, S.D., 2016b. Interactive algorithms for a broad underlying family of preference functions. Technical Report, Industrial Engineering Department, METU, Vol. 16.

Kohavi, R., John, G., 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1, 273–324.

Köksalan, 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.

Oliveira, L., Sabourin, R., Bortolozzi, F., Suen, C., 2002. Feature selection using multi-objective genetic algorithms for handwritten digit recognition. *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)* 1, 568–571.

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, 6, 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. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2, 3, 221–248.

Steuer, R.E., 1986. *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley, New York.

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* 14, 64–75.