Ant Feature Selection Using Fuzzy Decision Functions

Susana M. Vieira, João M.C. Sousa, and Uzay Kaymak

Abstract. One of the most important stages in data preprocessing for data mining is feature selection. Real-world data analysis, data mining, classification and modeling problems usually involve a large number of candidate inputs or features. Less relevant or highly correlated features decrease in general the classification accuracy, and enlarge the complexity of the classifier. Feature selection is a multi-criteria optimization problem with contradictory objectives, which are difficult to properly describe by conventional cost functions. This chapter proposes the use of fuzzy optimization to improve the performance of this type of system, since it allows for an easier and more transparent description of the criteria used in the feature selection process. In our previous work, an ant colony optimization algorithm for feature selection was proposed, which minimized two objectives: number of features and classification error. In this chapter, a fuzzy objective function is proposed to cope with the difficulty of weighting the different criteria involved in the optimization algorithm. The application of fuzzy feature selection to two benchmark problems show the usefulness of the proposed approach.

Keywords: Feature selection, fuzzy optimization, ant feature selection, ant colony optimization, fuzzy modeling.

Susana M. Vieira · João M.C. Sousa
Technical University of Lisbon, Instituto Superior Técnico
Dept. of Mechanical Engineering, Center of Intelligent Systems/IDMEC
Av. Rovisco Pais, 1049-001 Lisbon, Portugal
e-mail: susana@dem.ist.utl.pt, jmsousa@ist.utl.pt

Uzay Kaymak
Econometric Institute, Erasmus School of Economics
Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
e-mail: u.kaymak@ieee.org

W.A. Lodwick & J. Kacprzyk (Eds.): Fuzzy Optimization, STUDFUZZ 254, pp. 343–364.
springerlink.com

1 Introduction

Feature selection has been an active research area in data mining, pattern recognition and statistics communities. The main idea of feature selection is to choose a subset of available features, by eliminating features with little or no predictive information, and also redundant features that are strongly correlated. Many practical pattern classification tasks (as e.g., medical diagnosis) require learning of an appropriate classification function that assigns a given input pattern (typically represented by using a vector of feature values) to one of a set of classes. The choice of features used for classification has an impact on the accuracy of the classifier and on the time required for classification. The challenge is selecting the minimum subset of features with little or no loss of classification accuracy. The feature subset selection problem consists of identifying and selecting a useful subset of features from a larger set of often mutually redundant, possibly irrelevant, features with different associated importance [10].

Like many design problems, the feature selection problem is characterized by multiple goals, where a trade-off amongst various objectives must be made. Further, some of the objectives may be known only approximately. Fuzzy set theory provides ways of representing and dealing with flexible or soft goals and constraints. This flexibility can be exploited to obtain better solutions of the optimization problem. Various fuzzy optimization methods have been proposed in the literature to deal with different aspects of soft goals and constraints. In one formulation of fuzzy optimization due to Zimmermann [33], concepts from the Bellman and Zadeh [2] model of fuzzy decision making are used for formulating the fuzzy optimization problem. Recently, a method was proposed for satisfying the problem goals, where preference for different goals can be specified by the decision maker [13]. Fuzzy optimization admits the introduction of weight factors that represent the importance of the objectives for the optimization problem. Fuzzy optimization is used to handle feature selection problems in this chapter, where a weighted fuzzy objective function is proposed.

In this chapter, we present a feature selection algorithm based on an ant colony optimization algorithm, as proposed in [27, 28]. This algorithm uses two cooperative ant colonies, which cope with two different objectives. The two objectives we consider are minimizing the number of features and minimizing the error classification. Two pheromone matrices and two different heuristics are used for each objective.

The outline of this chapter is as follows. Section 2 gives a brief overview of the feature selection problem and its inherent difficulties. Model based feature selection is briefly described, as well as fuzzy modeling, which is used to evaluate the performance of the selected subsets. Further, feature selection is formulated as an optimization problem. A fuzzy optimization approach using fuzzy criteria is proposed for the feature selection problem in Section 3. The problem is defined, and membership functions to define the fuzzy goals are proposed. Additionally, fuzzy weighted optimization is presented, where the aggregation of fuzzy criteria for feature selection is also discussed. Section 4 presents the ant colony optimization algorithm used to solve the feature selection problem. Section 5 presents some applications of the proposed approach, and finally, Section 6 presents some concluding remarks.

2 Feature Selection

Feature selection, or variable subset selection, is the technique commonly used in selecting a subset of relevant features for building robust learning models. By removing most irrelevant and redundant features from data, feature selection helps improve the performance of learning models, by alleviating to some extent the effect of the curse of dimensionality, enhancing generalization capability, speeding up the learning process and even improving model interpretability. Feature selection also helps to better understand the data by discovering which are the important features and how they are related with each other.

From a theoretical perspective, it can be shown that optimal feature selection for supervised learning problems requires an exhaustive search of all possible subsets of features of the chosen cardinality. If large numbers of features are available, this is impractical. For practical supervised learning algorithms, the search is made for a satisfactory set of features, instead of for an optimal set [9].

Feature selection algorithms typically fall into two categories; feature ranking and subset selection. Feature ranking typically ranks the features by a metric and eliminates all features that do not achieve an adequate score. Subset selection usually entails the search for the optimal subset of possible features [11].

Subset selection evaluates a subset of features as a group, and the group of features that produces the most accurate model is selected. Subset selection algorithms can be broken into *wrappers* and *filters*. Wrappers use a search algorithm to search through the space of possible features and evaluate each subset by running a model on the subset. Wrappers can be computationally expensive and have a risk of overfitting the model. Filters are similar to wrappers in the search approach, but instead of evaluating against a model, the features are selected by evaluating a performance measure that does not require building a model.

Many popular search approaches use greedy hill climbing, which iteratively evaluates a candidate subset of features, then modifies the subset and determines whether or not if the new subset is an improvement over the old. Evaluation of the subsets requires a scoring metric that grades a subset of features. In the case of wrapper methods, the feature subsets are evaluated by the accuracy of the produced model. Exhaustive search is generally impractical, so at some defined stopping point, the subset of features with the highest score discovered up to that point is selected as the satisfactory feature subset. The stopping criterion varies by algorithm. Possible criteria include: a subset score exceeds a threshold, a program's maximum allowed run time has been surpassed, etc.

In this work, the feature selection problem is approached in the subset selection wrapper perspective. Thus, the output of a feature selection optimization problem is the reduced feature subset chosen to model the process with sufficient or necessary accuracy. Therefore, one of the main issues in model based feature selection

(MBFS) is the optimization technique applied to derive the reduced feature subset. The decision criteria of the optimization problem are the translation of the complexity (or cardinality) and accuracy of the final model. The number of selected features (cardinality) is often used as a measure of complexity in feature selection.

One of the main issues in MBFS is the type of model used to describe the process under study. Fuzzy models are rule-based systems that can be interpreted by human experts. For these reasons, fuzzy modeling is often called a "gray-box" modeling approach. This chapter uses fuzzy modeling, which is briefly presented in the following section.

2.1 Fuzzy Modeling

Rule-based expert systems are often applied to classification problems in fault detection, biology, medicine, etc. Fuzzy logic improves classification and decision support systems by allowing the use of overlapping antecedents definitions and improves the interpretability of the results by providing more insight into the classifier structure and the decision making process [18, 24].

In general, fuzzy models can provide a more transparent model and can also give a linguistic interpretation in the form of rules, which is appealing when dealing with classification systems. Fuzzy models use rules and logical connectives to establish relations between the features defined to derive the model.

The automatic determination of fuzzy classification rules from data has been approached by several different techniques: neuro–fuzzy methods, genetic–algorithm based rule selection and fuzzy clustering in combination with GA–optimization [20]. An approach that addresses simplicity and accuracy issues is used. Interpretable fuzzy rule-based classifiers are obtained from observation data following the steps described below.

In this work, we use Takagi-Sugeno (TS) fuzzy models [26], which consist of fuzzy rules where each rule describes a local input-output relation, typically in an affine form. The rules in the affine TS model are given by:

$$R_i : \textbf{If } x_1 \text{ is } A_{i1}\textbf{and } \ldots \textbf{and } x_n \text{ is } A_{in}\textbf{then } y_{C_i} = a_{i1}x_1 + \ldots + a_{in}x_n + b_i \,, \quad (1)$$

where $i = 1, \ldots, K$, K denotes the number of rules in the rule base, R_i is the i^{th} rule, $\mathbf{x} = [x_1, \ldots, x_n]^T$ is the antecedent vector, n is the number of features, $A_{i1}, \ldots, A_{in}$ are fuzzy sets defined in the antecedent space, y_{C_i} is the output variable for rule i, $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset for rule i. The consequents of the affine TS model are hyperplanes in the product space of the inputs and the output. The model output, y_C, is computed by aggregating the individual rules contribution:

$$y_C = \frac{\sum_{i=1}^{K} \beta_i y_{C_i}}{\sum_{i=1}^{K} \beta_i}, \quad (2)$$

where β_i is the degree of activation of the ith rule:

$$\beta_i = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j), \tag{3}$$

and $\mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0,1]$ is the membership function of the fuzzy set A_{ij} in the antecedent of R_i. Each class is considered an output of the model. A model has C classes. The output of the classifier is given by the following classification decision:

$$\max_{C} y_C \tag{4}$$

Given N available input–output data pairs $(\mathbf{x}_k, \mathbf{y}_k)$, the n–dimensional pattern matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$, and the corresponding C–dimensional class vector $\mathbf{y} = [y_1, \ldots, y_N]^T$ are constructed.

The number of rules K, the antecedent fuzzy sets A_{ij}, and the consequent parameters b_i are determined by means of fuzzy clustering in the product space of the input and output variables [24]. Hence, the data set $\mathbf{Z}$ to be clustered is composed by $\mathbf{X}$ and $\mathbf{y}$:

$$\mathbf{Z} = [\mathbf{X}, \mathbf{y}]^T . \tag{5}$$

Given the data $\mathbf{Z}$ and the number of clusters K, several fuzzy clustering algorithms can be used. This paper uses the Gustafson-Kessel (GK) [8] clustering algorithm to compute the fuzzy partition matrix $\mathbf{U}$. The matrix $\mathbf{Z}$ provides a description of the system in terms of its local characteristic behavior in regions of the data identified by the clustering algorithm, and each cluster defines a rule. The GK algorithm applies an adaptive distance measure, finding hyper-ellipsoid regions in the data that can be efficiently approximated by the hyper-planes described by the consequents in the TS model.

The fuzzy sets in the antecedent of the rules are obtained from the partition matrix $\mathbf{U}$, whose ikth element $\mu_{ik} \in [0,1]$ is the membership degree of the data object $\mathbf{z}_k$ in cluster i. One-dimensional fuzzy sets A_{ij} are obtained from the multidimensional fuzzy sets defined point-wise in the ith row of the partition matrix by projections onto the space of the input variables x_j:

$$\mu_{A_{ij}}(x_{jk}) = \text{proj}_j^{\mathbb{N}^{n+1}}(\mu_{ik}), \tag{6}$$

where proj is the point-wise projection operator [15]. The point-wise defined fuzzy sets A_{ij} are approximated by suitable parametric functions in order to compute $\mu_{A_{ij}}(x_j)$ for any value of x_j. This is schematically represented in Fig. 1.

The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let $\theta_i^T = [\mathbf{a}_i, b_i]$, let $\mathbf{X}_e$ denote the matrix $[\mathbf{X}; \mathbf{1}]$ and let $\mathbf{W}_i$ denote a diagonal matrix in having the degree of activation, $\beta_i(\mathbf{x}_k)$, as its kth diagonal element. Assuming that the columns of $\mathbf{X}_e$ are linearly independent and $\beta_i(\mathbf{x}_k) > 0$ for $1 \leq k \leq N$, the weighted least-squares solution of $\mathbf{y} = \mathbf{X}_e\theta + \varepsilon$ becomes

$$\theta_i = \left[\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e\right]^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y} . \tag{7}$$

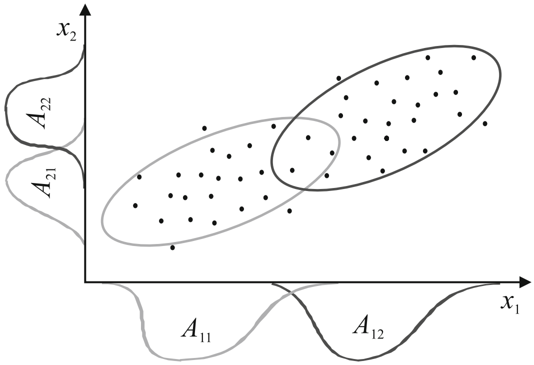


Fig. 1 Projection of multidimensional fuzzy sets onto the space of the input variables x_j

Rule bases constructed from clusters can be redundant due to the fact that the rules defined using the multidimensional antecedents are overlapping in one or more dimensions. A possible approach to solve this problem is to reduce the number of features n of the model, as addressed in this chapter.

2.2 Formulation of the Feature Selection Problem

When a classification system is designed, performance criteria must be specified. These criteria are usually defined in terms of a desired minimum error between the real classification and the model output, model complexity, number of used features, etc., representing the goals of the classification system. In MBFS, these goals must be translated as criteria into an objective function. This function normally is minimized (or maximized) over a given number of iterations.

Conventional MBFS mainly uses classification accuracy as the objective function [12]. In model based feature selection, besides classification accuracy, model complexity reduction is also desired. The model complexitiy is directly related to the number of features used for modeling.

Let the overall MBFS goals be stated as achieving good performance while reducing the features subset size, and implicitly the model complexity. The performance criterion used to evaluate the fuzzy model is the classification accuracy γ, given by:

$$\gamma = \frac{(N_n - N_e)}{N_n}. \tag{8}$$

where N_e is the number of errors in test samples and N_n is the number of used samples.

Let $\mathcal{F}$ be the n-dimensional set of features. The vector $\mathbf{z} = (z_1, z_2, \ldots, z_{N_f})$, is a subset of $\mathcal{F}$, and N_f is the features cardinality (number of used features). It is desirable that $N_f << n$. The goals can be represented by the following objective function

$$J = w_1(1-\gamma) + w_2\frac{N_f}{n}, \tag{9}$$

where γ is the percentage of correct classifications in test samples and $\{w_1, w_2\}$ represent weight factors to help setting priorities of the optimization problem. The first term of (9) represents the score of the features subset in terms of model performance. The objective function of this optimization problem aggregate both criteria: the minimization of the classification error and the minimization of the features cardinality. These are contradictory objectives, and are difficult to properly describe by this conventional cost function, once the weights are difficult to settle although the terms are normalized.

The use of fuzzy objective functions can improve the performance of this type of optimization problem, since it allows for an easier and transparent description of the different criteria used in the feature selection process. In next section, a fuzzy optimization problem is presented for MBFS to cope with the difficulty of weighting the different criteria involved in the optimization algorithm.

3 Fuzzy Optimization in Feature Selection

The objective function for feature selection can be seen as the simultaneous satisfaction of different criteria. These fuzzy criteria must be defined for different objectives inherent to the feature selection problem. When fuzzy criteria is used in the objective function, fuzzy optimization is the most obvious technique to deal with the optimization problem in MBFS.

3.1 Formulation of the Feature Selection Problem Using Fuzzy Goals

Feature selection using fuzzy goals can be defined as follows. Let G_ℓ, with $\ell = 1, \ldots, q$, be a fuzzy goal (or criterion) characterized by its membership function μ_{G_ℓ}, which is a mapping from the space of the goal G_ℓ to the interval $[0,1]$. The goals are defined on relevant optimization criteria. Each goal is defined in the domain Φ_j, $j = 1, \ldots, q$, which can be any of the various domains used in feature selection. This optimization problem has a discrete search space with a finite and countable set Ω of subset solutions. The fuzzy criteria must be aggregated in the subset selection environment. The membership value μ_z for the subset solution z is obtained using the aggregation operator $\circledast$ to combine the fuzzy goals (criteria),

$$\mu_z = \mu_{G_1} \circledast \ldots \circledast \mu_{G_q}\,. \tag{10}$$

Various types of aggregation operations can be used as decision functions for expressing different decision strategies using the well-known properties of these operators [24]. Parametric triangular norms can generalize a large number of t-norms, and can control the degree of compensation between the different criteria. The decision criteria in (10) (e.g. small number of features and high accuracy) should be satisfied as much as possible, which corresponds to the maximal value of the overall decision. The optimal subset z^* is found by the maximization of μ_z:

$$z^* = \arg \max_{\mathbf{z}} \mu_z. \tag{11}$$

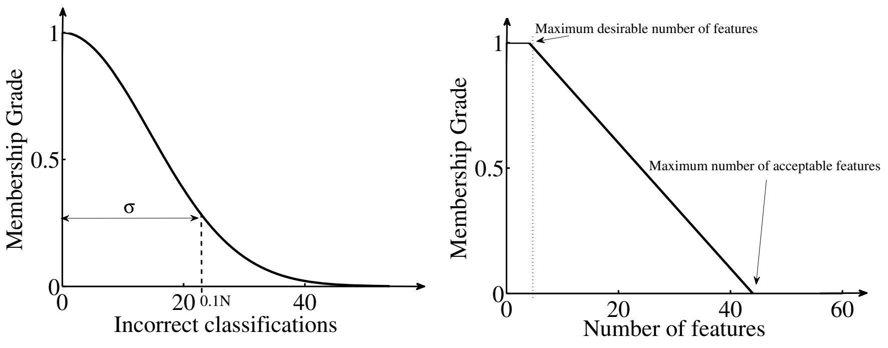
Because the membership functions for the fuzzy criteria can have an arbitrary shape, and because of the nonlinearity of the decision function, the optimization problem (11) is usually non-convex. This problem is discussed in Section 4. However, we just say at this point that given that (10) is a non-convex nonlinear optimization problem, a heuristic method such as ant colony is useful to use.

3.2 Fuzzy Criteria in Feature Selection

Fuzzy goals must be a translation of the (fuzzy) performance criteria defined for the system. The definition of these criteria in terms of model performance has shown to be quite powerful in the model based feature selection framework [9].

Additional flexibility can be introduced in MBFS, when fuzzy multicriteria is applied to determine the objective function. This flexibility provides additional control to the model builder to have control over the optimization problem. Each goal G_ℓ is described by a fuzzy set. Fuzzy criteria can be described in different ways. The most straightforward and easy way is just to adapt the classical criteria in MBFS, as defined in (9).

Figure 2 shows examples of general membership functions that can be used for the error N_e and for the features cardinality N_f. In this example, the minimization of the classification error is represented by an exponential membership function, given by (see Fig. 2a):



(a) Membership function for the number of incorrect classifications.

(b) Membership function for the number of selected features

Fig. 2 Membership functions for the feature selection goals

$$\mu_e = \exp^{-\left(\frac{N_e}{2\sigma}\right)^2} \tag{12}$$

This well-known function has the nice property of never reaching the zero value, and the membership value is still quite considerable for an error of 10%. This property is very useful, once the desirable solution should have the smaller error possible. The σ parameter can be defined as a small percentage of the number of data samples or as a percentage of error that is admissible for the problem.

The features cardinality N_f can be represented, for example, by a trapezoidal membership function, as shown in Fig. 2b. A reduced number of features is considered to be a desired outcome of the optimization algorithm. The membership function is defined so that for a low number of features the membership degree is one and linearly decreases to zero. The membership degree should be zero outside the maximum number of available features. This is because we are trying to get the number of features to be as close as possible to the minimum desired. The parameters defining the range of the trapezoidal membership function are application dependent. Sometimes it is convenient to make the upper limit of the membership function significantly lower than the maximum number of allowed features, especially if a very large number of features is being tested.

In general, the parameters of the different membership functions are application dependent. However, it is possible to derive some tuning guidelines, as described in the following. The membership functions quantify how much the system satisfies the criteria given a particular feature subset, bringing various quantities into a unified domain.

The use of the membership functions introduces additional flexibility in the goals, and it leads to increased transparency as it becomes possible to specify explicitly what kind of solution is preferred. For instance, it becomes easier to penalize more severely a subset of features that have bigger classification errors. Alternatively, if we prefer a solution with less features, a higher number of features can be penalized instead.

After the membership functions have been defined, they are combined by using a decision function, such as a parametric aggregation operator from the fuzzy sets theory, as e.g. the Yager t-norm [31]. The formulation in (10) does not use weights for the criteria. However, a general formulation that takes into account different importance of different goals must consider weights, as in the conventional objective function (9). The generalization of the fuzzy objective function (10) in order to use weighted criteria is used here, and is presented in the next section.

3.3 Weighted Fuzzy Optimization

Fuzzy objective functions using weighted criteria have been used quite extensively, especially in fuzzy decision making, where the weights are used to represent the relative importance that the decision maker attaches to different decision criteria. An averaging operator is normally used for the weighted aggregation, such as generalized means [6], fuzzy integrals [7] or the ordered weighted average (OWA) operators [32]. Consequently, weighted aggregation of fuzzy sets has been studied with averaging type of operators. The generalized means extend naturally to weighted equivalents. The weighted generalized mean operator has been used in many fields, and it has been studied in the context of fuzzy set aggregation in [6, 14]. The OWA operators and the fuzzy integrals are inherently weighted operators, which do not need a separate extension to the weighted case. Applications of these operators have also been reported in the literature, see e.g. [7, 19].

In the fuzzy feature selection approach, the goal is to satisfy simultaneously the model accuracy and the feature subset reduction. The averaging operators are suitable for modeling compensatory aggregation. In compensatory aggregation, one criterion can compensate for another one. However, they are not suitable for modeling simultaneous satisfaction of aggregated contradictory criteria, where the aggregated value should never be larger then the least satisfied criterion. In this case, t-norms must be used to model the conjunctive aggregation [25]. Therefore, weighted aggregation using t-norms must be considered. Note that the most common axiomatic definition of t-norms does not allow for weighted aggregation. Hence, the commutativity and the associativity properties must be dropped, since weighted operators are by definition not commutative.

Weighted aggregation of fuzzy sets by using t-norms has been considered first by Yager in [29], where the membership functions are modified by associated weight factors before the fuzzy aggregation. The application of a generalized form of this idea, introduced in [30], is given by

$$\mu_z(\mathbf{w}) = t[I(\mu_{G_1}, w_1), \ldots, I(\mu_{G_i}, w_q)], \tag{13}$$

where $\mathbf{w}$ is a vector of weight factors $w_l \in [0,1]$, t is a t-norm and I is a function that transforms the membership functions. Note that the fuzzy objective function for feature selection in this chapter has two goals ($q = 2$), where G_1 translates the classification accuracy and G_2 the features cardinality reduction. The most common fuzzy aggregation operator uses the power-raising method for the transformation and the minimum operator for the t-norm.

$$\mu_z(\mathbf{w}) = \bigwedge_{l=1}^{q} [\mu_{G_l}]^{w_l} . \tag{14}$$

This aggregation function has been used in many publications regarding the application of fuzzy weighted aggregation, especially in multicriteria decision making. Weighted aggregation of fuzzy sets has been investigated in more detail in a generalized framework [13, 14], where weighted counterparts of fuzzy t-norms have also been proposed based on a sensitivity analysis of weighted fuzzy aggregation. This analysis provides a general mechanism for introducing weight factors into Archimedean t-norms and t-conorms by considering several requirements that can be imposed on a weighted aggregation operator. In this work, weighted counterparts of Archimedean t-norms are used, namely, the weighted extension of the product t-norm

$$\mu_z(\mathbf{w}) = \prod_{l=1}^{q} [\mu_{G_l}]^{w_l}, \tag{15}$$

the extension of the Hamacher t-norm

$$\mu_z(\mathbf{w}) = \begin{cases} \dfrac{1}{1+\sum\limits_{l=1}^{q} w_l \dfrac{1-\mu_{G_l}}{\mu_{G_l}}} & \text{if } \forall l, \mu_{G_l} > 0 \\ 0 & \text{if } \exists l, \mu_{G_l} = 0 \end{cases} \tag{16}$$

and the extension of the Yager t-norm

$$\mu_z(\mathbf{w}) = \max\left(0, 1 - \sqrt[s]{\sum_{l=1}^{q} w_l (1-\mu_{G_l})^s}\right), \qquad s > 0\,. \tag{17}$$

Note that the extension of the product t-norm in (15), according to the sensitivity based analysis, is the same as the application of (13) using the product operator as t, with power raising method as I. However, the extensions (16) and (17) cannot be obtained from (13). Applications of weighted fuzzy optimization can be found in [13, 17].

After combining the objectives, the resulting optimization is non-convex. Furthermore, gradient descent methods may not be suitable for the maximization due to possible and likely discontinuity in the first derivative of the final aggregated function. Derivative-free search and optimization algorithms such as simulated annealing, evolutionary algorithms or other bio-inspired algorithms, such as ant colony optimization, can be used to solve this type of optimization problems. In this work, an ant colony optimization approach is used to solve the MBFS optimization problem, as proposed in [27, 28]. This approach is described in the next section.

4 Ant Feature Selection

Ant algorithms were first proposed by Dorigo [3] as a multi-agent approach to difficult combinatorial optimization problems, such as traveling salesman problem, quadratic assignment problem or supply chain management [21, 22, 23]. The ant colony optimization (ACO) methodology is an optimization method suited to find minimum cost paths in optimization problems described by graphs [4].

This chapter presents an implementation of ACO applied to feature selection, where the best number of features is determined automatically [27, 28]. In this approach, two objectives are considered: minimizing the number of features and minimizing the classification error. Two cooperative ant colonies are considered, one for each objective. The first colony determines the number (cardinality) of features and the second selects the features based on the cardinality given by the first colony. Thus, two pheromone matrices and two different heuristics are used. The heuristic value is computed using the Fisher discriminant criterion for feature selection [5], which ranks the features by giving them a relative importance.

The determination of the *features cardinality* N_f is addressed in the first colony sharing the same minimization cost function with the second colony, which in this case aggregates both the maximization of the classification accuracy and the

Table 1 Variables definition

Variable	**Description**
General	
n	Number of features
N	Number of samples
N_n	Number of samples used for validation
I	Number of iterations
K	Number of rules/clusters of the fuzzy model
C	Number of existing classes in database
g	Number of ants
$\mathbf{x}$	Set with all the features
$\mathbf{w}$	Subset of features selected to build classifiers
J^k	Cost of the solution for each ant k
J^q	Cost of the winner ant q
Ant colony for cardinality of features	
N_f	Features cardinality (number of selected features)
$N_f(k)$	Features cardinality of ant k
I_n	Number of iterations with same feature cardinality
α_n	Pheromone weight of features cardinality
β_n	Heuristic weight of features cardinality
τ_n	Pheromone trails for features cardinality
η_n	Heuristic of features cardinality
ρ_n	Evaporation of features cardinality
$\varGamma_n^k$	Feasible neighborhood of ant k (features cardinality availability)
$\mathcal{Q}_i$	Amount of pheromone laid in the features cardinality of the best solution
Ant colony for selecting subset of features	
$L_f^k(t)$	Feature subset for ant k at tour t
α_f	Pheromone weight of features
β_f	Heuristic weight of features
τ_f	Pheromone trails for feature selection
η_f	Heuristic of features
ρ_f	Evaporation of features
$\varGamma_f^k$	Feasible neighborhood of ant k (features availability)
$\mathcal{Q}_j$	Amount of pheromone laid in the features of the best solution

minimization of the features cardinality. Hence, the first colony determines the size of the subsets of the ants in the second colony, and the second colony selects the features that will be part of the subsets.

The algorithm used in this study deals with the feature selection problem as a multi–criteria problem with a single objective function. Therefore, a pheromone matrix is computed for each criterion, and different heuristics are used. Table 1 describes the variables used in the algorithm. To evaluate the classification error, a fuzzy classifier is built for each solution using the procedure described in Section 2.

4.1 Probabilistic Rule

Consider a problem with N_f nodes and two colonies of g ants. First, g ants of the first colony randomly select the number of nodes N_f to be used by the g ants of the second colony. Following the original ACO [3], the probability that an ant k chooses the features cardinality $N_f(k)$ is given by

$$p_i^k(t) = \frac{[\tau_{n_i}]^{\alpha_n} \cdot [\eta_{n_i}]^{\beta_n}}{\sum_{l \in \Gamma_n^k} [\tau_{n_l}]^{\alpha_n} \cdot [\eta_{n_l}]^{\beta_n}} \tag{18}$$

where τ_{n_i} is the pheromone concentration matrix and η_{n_i} is the heuristic function matrix, for path (i). The values of the pheromone matrix are limited to $[\tau_{n_{\min}}, \tau_{n_{\max}}]$, with $\tau_{n_{\min}} = 0$ and $\tau_{n_{\max}} = 1$. Γ_n^k is the feasible neighborhood of ant k (available number of features to be selected), which acts as the memory of the ants, and contains all the trails that the ants have not passed and can be chosen, here the trails represent the features. The parameters α_n and β_n measure the relative importance of trail pheromone and heuristic knowledge, respectively.

After all the g ants from the first colony have chosen the features cardinality $N_f(k)$, each ant k from the second colony selects $N_f(k)$ features (nodes). The probability that an ant k chooses feature j as the next feature to visit is given by

$$p_j^k(t) = \frac{[\tau_{f_j}(t)]^{\alpha_f} \cdot [\eta_{f_j}]^{\beta_f}}{\sum_{l \in \Gamma_f^k} [\tau_{f_l}(t)]^{\alpha_f} \cdot [\eta_{f_l}]^{\beta_f}} \tag{19}$$

where τ_{f_j} is the pheromone concentration matrix and η_{f_j} is the heuristic function matrix for the path (j). Again, the pheromone matrix values are limited to $[\tau_{f_{\min}}, \tau_{f_{\max}}]$, with $\tau_{f_{\min}} = 0$ and $\tau_{f_{\max}} = 1$. Γ_f is the feasible neighborhood of ant k (available features), which contains all the features that the ants have not selected and can be chosen. Again, the parameters α_f and β_f measure the relative importance of trail pheromone and heuristic knowledge, respectively.

4.2 Updating Rule

After a complete tour, when all the g ants have visited all the $N_f(k)$ nodes, both pheromone concentration in the trails are updated by

$$\tau_{n_i}(t+1) = \tau_{n_i}(t) \times (1 - \rho_n) + \Delta\tau_{n_i}(t) \tag{20}$$

$$\tau_{f_j}(t+1) = \tau_{f_j}(t) \times (1 - \rho_f) + \Delta\tau_{f_j}(t) \tag{21}$$

where $\rho_n \in [0,1]$ is the pheromone evaporation of the features cardinality , $\rho_f \in [0,1]$ is the pheromone evaporation of the features and $\Delta\tau_{n_i}$ and $\Delta\tau_{f_j}$ are the pheromone deposited on the trails (i) and (j), respectively, by the ant q that found the best solution J^q for this tour:

$$\Delta\tau^q_{n_i} = \begin{cases} \mathcal{Q}_i & \text{if node } (i) \text{ is used by the ant } q \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

$$\Delta\tau^q_{f_j} = \begin{cases} \mathcal{Q}_j & \text{if node } (j) \text{ is used by the ant } q \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

The number of nodes $N_f(k)$ that each ant k has to visit on each tour t is only updated every I_n tours (iterations), in order to allow the search for the best features for each N_f. The algorithm runs I times. Both colonies share the same cost functions. Classical and fuzzy cost functions, given respectively by (9) and (10) are both tested in this chapter.

4.3 Heuristics

The heuristic value used for the second ant colony is computed as

$$\eta_{f_j} = 1/Ne_j \tag{24}$$

for $j = 1,\ldots,n$. For the features cardinality (first colony), the heuristic value is computed using the Fisher discriminant criterion for feature selection [5]. Considering a classification problem with two possible classes, class 1 and class 2, the Fisher discriminant criterion is described as

$$F(i) = \frac{|\mu_1(i) - \mu_2(i)|^2}{\sigma_1^2 + \sigma_2^2} \tag{25}$$

where $\mu_1(i)$ and $\mu_2(i)$ are the mean values of feature i for the samples in class 1 and class 2, respectively, and σ_1^2 and σ_2^2 are the variances of feature i for the samples in classes 1 and 2. The score aims to maximize the between-class difference and minimize the within-class spread. Other currently proposed rank-based criteria generally come from similar considerations and show similar performance [5]. Since our goal is to work with several classification problems, which can contain two or more possible classes, a one versus-all strategy is used to rank the features. Thus, for a C-class prediction problem, a particular class is compared with the other $C-1$ classes that are considered together. The features are weighted according to the total score summed over all C comparisons:

$$\sum_{j=1}^{C} F_j(i), \tag{26}$$

Algorithm 1. Ant Feature Selection

```
/*Initialization*/
set the parameters $\rho_f, \rho_n, \alpha_f, \alpha_n, \beta_f, \beta_n, I, I_n, g$.
for $t = 1$ to $I$ do
    for $k = 1$ to $g$ do
        Choose the subset size $N_f(k)$ of each ant $k$ using (18)
    end for
    for $l = 1$ to $I_n$ do
        for $k = 1$ to $g$ do
            Build feature set $L_f^k(t)$ by choosing $N_f(k)$ features using (19)
            Compute the fuzzy model using the $L_f^k(t)$ path selected by ant $k$
            Compute the cost function $J^k(t)$
            Update $J^q$
        end for
        Update pheromone trails $\tau_{n_i}(t+1)$ and $\tau_{f_j}(t+1)$, as defined in (20) and (21).
    end for
end for
```

where $F_j(i)$ denotes the Fisher discriminant score for the i^{th} feature at the j^{th} comparison. Algorithm 1 presents the description of the ant feature selection algorithm.

5 Application Examples

5.1 Data Sets

The effectiveness of the proposed approach is tested using data sets taken from some well known benchmarks in the UCI repository [1]. Two real data sets, Wine and Wisconsin Breast Cancer were used to test the presented approach. The characteristics of the data are presented in Table 2.

Table 2 Description of the used data sets

Data sets used	# features	# classes	# samples
Wine	13	3	178
Breast Cancer	9	2	699

Wine. The wine data set is widely used in the literature. The classification data is available online in the repository of the University of California [1], and contains the chemical analysis of 178 wines grown in the same region in Italy, derived from three different cultivars. Thirteen continuous attributes are available for classification: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, nonflavanoids phenols, proanthocyanism, color intensity, hue, the ratio OD280/OD315 of dilluted wines and proline.

Breast Cancer. The Wisconsin breast cancer data is also widely used to test the effectiveness of classification algorithms. The aim of the classification is to distinguish between benign and malignant cancers based on the available nine measurements (attributes): clump thickness, uniformity of cell size, uniformity of cell shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli and Mitoses. The attributes have integer value in the range [1,10]. The original database contains 699 instances, however 16 samples are omitted because they are incomplete. The class distribution is 65.5% benign and 34.5% malignant. The breast cancer data set is also available in the repository of the University of California [1] and it was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

5.2 Performance Evaluation

This chapter uses two types of performance evaluation. First, the data sets are divided in training and test instances (50% for training and 50% for test). As is common, the test data set contains data points different from the ones used to train the model. The models are constructed using the procedure described in Section 2.1. As these results are excellent for the considered data sets, more demanding validation tests were necessary. Thus, we used the well-known cross validation method, which is briefly described next.

Cross-validation. In cross validation, the data set with N_n samples is divided into N mutually exclusive sets of approximately equal size, with each subset consisting of approximately the same proportions of labels as the original data set, known as stratified cross validation [16]. The classifier is trained N times, with a different subset left out as the test set and the other samples used to train the classifier at each time. During the training phase, the classifier is trained on $N-1$ out of N folds in which classification accuracy is used, as defined in (8). The prediction performance of the classifier is estimated by considering the average classification accuracy of the 10 cross-validation experiments, described as

$$E_{CV} = \left(\frac{1}{N_n} \sum_{i=1}^{N} C_i \right) \times 100\% \tag{27}$$

where C_i is the number of correctly classified samples:

$$C_i = N_n - N_e.$$

The classification error rates of the final subset solutions are obtained by performing N-fold cross validation, with $N = 10$ in our case (CV10). The experimental results are presented as the best, the worst and the mean value of the classification accuracy γ, as defined in (8).

5.3 Results

5.3.1 Train-Test Results

First the data sets were divided into training and test data, and 10 trials are simulated. The fuzzy goals were aggregated using the Yager t-norm. The maximum, mean and minimum classification accuracies of these 10 trials are shown in Table 3. The results are compared with the ant feature selection (AFS) approach in [28]. Table 3 shows clearly that using a fuzzy objective function the algorithm always converges to the optimal solution. Further, the algorithm converges always to the same number of features, which was not the case with AFS. Clearly, more demanding tests were necessary to test the fuzzy approach. Therefore, 10-fold cross validation was applied to the data.

Table 3 Classification rates for train/test data sets

Data set	**Methods**	**Reduced Subset**	**Classification accuracy** (%)		
			Max.	**Mean**	**Min.**
Wine	AFS	4-8	100.0	99.8	98.9
	Fuzzy AFS	4	100.0	100.0	100.0
Breast Cancer	AFS	2-5	100.0	96.4	91.3
	Fuzzy AFS	3	100.0	100.0	100.0

5.3.2 Cross Validation Results

Ten–fold cross validation was applied to both data sets. Different aggregation operators were used to test the fuzzy optimization, namely: product, Yager and Hamacher t-norms. The results for the data sets are presented next.

Wine Results. The results obtained for the wine data set are presented in Table 4. The t-norm that constantly achieves the smaller number of features is the Yager t-norm. The best results in terms of accuracy are obtained with the Hamacher t-norm, using however a much larger number of features. The product t-norm does not converge to the same number of features, achieving results similar to the ones without the fuzzy approach, i.e. using the AFS algorithm.

Table 4 Cross validation results for the Wine data set

t-**norm**	**Number of features**	**Classification accuracy** (%)		
		Max.	**Mean**	**Min.**
Product	4-8	100	91.1	72.2
Yager	4	100	92.0	75.0
Hamacher	11	100	93.9	88.9

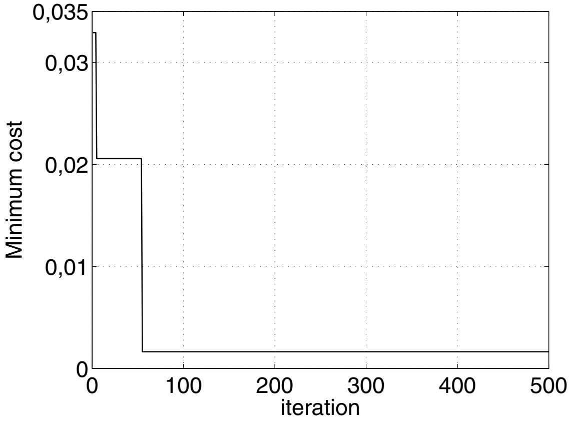


Fig. 3 Cost values for each iteration in the Wine data set

Figure 3 presents an example of the values obtained for the fuzzy objective function in (17), when the Yager t-norm is used. The convergence of the fuzzy optimization algorithm using again the Yager t-norm is depicted in Fig. 4, where the convergence to a constant number of features is clearly shown. Note that a 10-fold

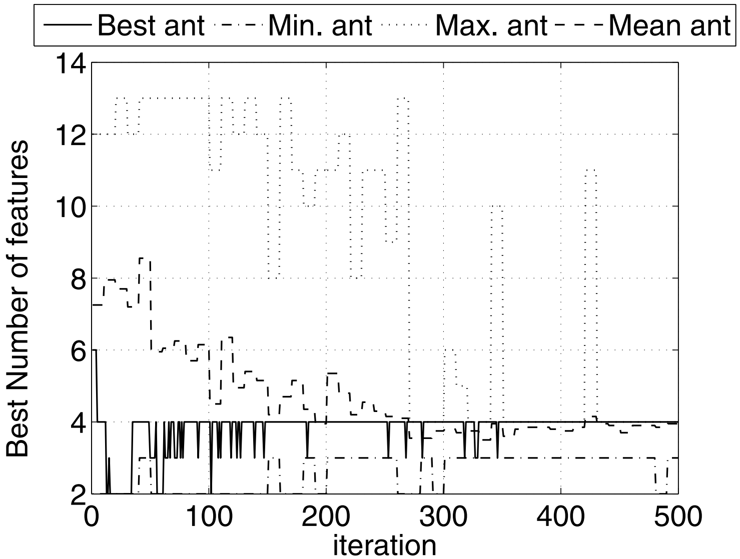


Fig. 4 Best number of features for each iteration in the Wine data set

Table 5 Cross validation results for the Wisconsin Breast Cancer data set

t-norm	Number of features	Classification accuracy (%)		
		Max.	Mean	Min.
Product	3-5	100	96.1	91.4
Yager	3-4	100	96.0	94.3
Hamacher	9	98.6	94.9	90.0

cross validation in the wine data set is clearly a very demanding test for the classification model, as the data set has very few instances.

Breast Cancer results. Table 5 presents the results obtained for this data set using several aggregation t-norms. Depending on the t-norm used, is not always possible to minimize the number of features. Again, the t-norm that constantly achieves the smaller number of features is the Yager t-norm. Further, the product t-norm does not converge to the same number of features, as happened with the Wine data set. In terms of accuracy, the best results are now obtained with the Yager t-norm (although the product has similar results), and the worst results are obtained with the Hamacher t-norm.

Figure 5 presents the convergence of the fuzzy optimization algorithm using the Yager t-norm. This t-norm converges to a small number of features (four in this case).

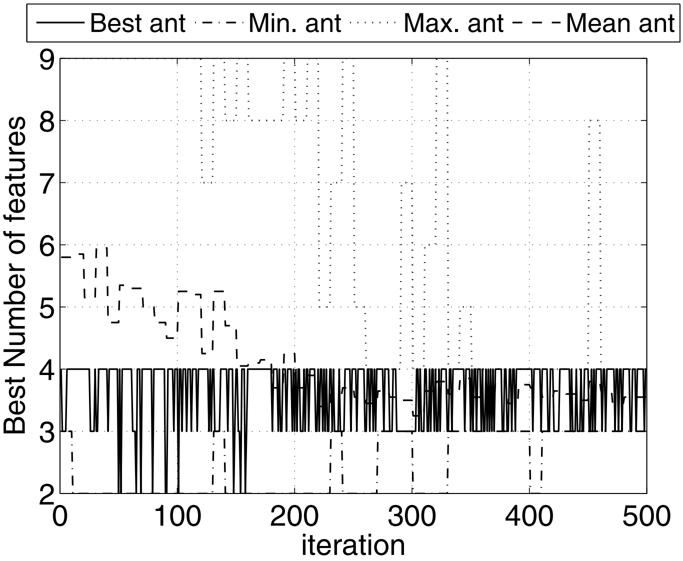


Fig. 5 Best number of features for each iteration in the breast cancer data set

5.4 Discussion

By observing Tables 4 and 5 it becomes clear that the Yager t-norm is the one obtaining the smaller number of features. The accuracy is also very good, but not always the best. In the Wine data set the best accuracy was obtained with the Hamacher t-norm.

Ant feature selection using the conventional objective function in (9) has some difficulties to satisfy both optimization criteria, even when weight factors are used. In general, the use of fuzzy optimization results in a better convergence, especially when the Yager t-norm is utilized. In summary, it is possible to conclude from the results that the performance of the optimization algorithm has improved using a fuzzy objective function.

6 Conclusions

A fuzzy objective function for ant feature selection is proposed in this chapter. The problem is divided into two objectives: minimizing the features cardinality and selecting the most relevant features. The feature selection algorithm uses fuzzy classifiers to evaluate the selected subsets of features. The proposed algorithm was applied to two well known classification databases that are considered benchmarks. Different fuzzy aggregation t-norms were tested. The results show that the proposed approach leaded to better results than ant feature selection using a classical objective function.

A systematic procedure to choose the weighting factors was not used in this study. Thus, further work is necessary to evaluate the weighting effect on the aggregation methods. This study should also test changing the weights during the evolution of the optimization algorithm. Finally, data sets with a larger number of features must be tested to confirm the obtained results.

Acknowledgements. This work is supported by: the Portuguese Government and FEDER under the programs: Programa de financiamento Plurianual das Unidades de I&D da FCT (POCTI-SFA-10-46-IDMEC) and by the FCT grant SFRH/25381/2005, Fundação para a Ciência e a Tecnologia, Ministério do Ensino Superior, da Ciência e da Tecnologia, Portugal; and by the European Science Foundation, COST Action IC0702, "Combining Soft Computing Techniques and Statistical Methods to Improve Data Analysis Solutions".

References

[1] Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
[2] Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. Management Science 17(4), 141–164 (1970)
[3] Dorigo, M.: Optimization, Learning and Natural Algorithms (in Italian). PhD thesis (1992)

[4] Dorigo, M., Birattari, M., Stützle, T.: Ant colony optimization. IEEE Computational Intelligence Magazine 1(4), 28–39 (2006)

[5] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, vol. 2. Wiley–Interscience Publication, Chichester (2001)

[6] Dyckhoff, H., Pedrycz, W.: Generalized means as model of compensative connectives. Fuzzy Sets and Systems 14, 143–154 (1984)

[7] Grabisch, M., Nguyen, H.T., Walker, E.A.: Fundamentals of uncertainty calculi with applications to fuzzy inference. In: Mathematical and Statistical Methods, vol. 30. Kluwer Academic Publishers, Dordrecht (1995)

[8] Gustafson, D.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: Proceedings of the 18^{th} IEEE Conference on Decision and Control, San Diego, CA, USA, pp. 761–766 (1979)

[9] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

[10] Motoda, H., Liu, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Dordrecht (1998)

[11] Jensen, R., Shen, Q.: Finding rough set reducts with ant colony optimization. In: Proceedings of the 2003 UK Workshop on Computational Intelligence, pp. 15–22 (2003)

[12] Jensen, R., Shen, Q.: Fuzzy-rough data reduction with ant colony optimization. Fuzzy Sets and Systems 149, 5–20 (2005)

[13] Kaymak, U., Sousa, J.M.: Weighted constraint aggregation in fuzzy optimization. Constraints 8(1), 61–78 (2003)

[14] Kaymak, U., van Nauta Lemke, H.R.: A sensitivity analysis approach to introducing weight factors into decision functions in fuzzy multicriteria decision making. Fuzzy Sets and Systems 97(2), 169–182 (1998)

[15] Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: theory and applications. Prentice-Hall, Upper Saddle River (1995)

[16] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. International Joint Conf. Artificial Intelligence (1995)

[17] Mendonça, L.F., Sousa, J.M.C., Kaymak, U., Sá da Costa, J.M.G.: Weighting goals and comstraints in fuzzy predictive control. Journal of Intelligent and Fuzzy Systems 17(5), 517–532 (2006)

[18] Roubos, J.A., Setnes, M., Abonyi, J.: Learning fuzzy classification rules from labeled data. International Journal of Information Sciences 150(1), 77–93 (2003)

[19] Salido, J.M.F., Murakami, S.: Extending Yager's orness concept for the OWA aggregators to other mean operators. Fuzzy Sets and Systems 139(3), 515–542 (2003)

[20] Setnes, M., Roubos, J.A.: GA-fuzzy modeling and classification: complexity and performance. IEEE Transactions on Fuzzy Systems 8(5), 509–522 (2000)

[21] Silva, C.A., Sousa, J.M.C., Runkler, T.A.: Rescheduling and optimization of logistic processes using GA and ACO. Engineering Applications of Artificial Intelligence 21(3), 343–352 (2007)

[22] Silva, C.A., Sousa, J.M.C., Runkler, T.A., Sá da Costa, J.M.G.: Distributed optimization of a logistic system and its suppliers using ant colony optimization. International Journal of Systems Science 37(8), 503–512 (2006)

[23] Silva, C.A., Sousa, J.M.C., Runkler, T.A., Sá da Costa, J.M.G.: Distributed supply chain management using ant colony optimization. To appear in European Journal of Operational Research (2009), doi:10.1016/j.ejor.2008.11.021

[24] Sousa, J.M.C., Kaymak, U.: Fuzzy Decision Making in Modeling and Control. World Scientific/Imperial College, Singapore/UK (2002)

[25] Sousa, J.M.: Optimization issues in predictive control with fuzzy objective functions. International Journal of Intelligent Systems 15(9), 879–899 (2000)

[26] Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modelling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1), 116–132 (1985)

[27] Vieira, S.M., Sousa, J.M.C., Runkler, T.A.: Fuzzy classification in ant feature selection. In: Proc. of 2008 IEEE World Congress on Computational Intelligence, WCCI 2008, pp. 1763–1769, Hong Kong, China (June 2008)

[28] Vieira, S.M., Sousa, J.M.C., Runkler, T.A.: Two cooperative ant colonies for feature selection using fuzzy models. Submitted to Expert Systems with Applications (2009)

[29] Yager, R.R.: Fuzzy decision making including unequal objectives. Fuzzy Sets Systems 1, 87–95 (1978)

[30] Yager, R.R.: General multiple-objective decision functions and linguistically quantified statements. International Journal of Man-Machine Studies 21(5), 389–400 (1984)

[31] Yager, R.R.: On a general class of fuzzy connectives. Fuzzy Sets and Systems 4, 235–242 (1980)

[32] Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transaction Systems, Man and Cybernetics 18(1), 183–190 (1988)

[33] Zimmermann, H.J.: Description and optimization of fuzzy systems. International Journal of General Systems 2, 209–215 (1976)