VNU Journal of Science: Comp. Science & Com. Eng., Vol. 30, No. 4 (2014) 44-56

A Hybrid Multi-objective PSO-SA Algorithm for the Fuzzy Rule Based Classifier Design Problem with the Order Based Semantics of Linguistic Terms*

Phong Pham Dinh[1], Thuy Nguyen Thanh[2], Thanh Tran Xuan[3]

[1,2]*Faculty of Information Technology, VNU University of Engineering and Technology, Vietnam*
[3]*Faculty of Information Technology, Thanh Do University, Vietnam*

Abstract

A number of studies [26, 28, 33] have shown that the method of designing fuzzy rule based classifiers (FRBCs) using multi-objective optimization evolutionary algorithms (MOEAs) clearly depends on the evolutionary quality. Each evolutionary algorithm has the advantages and the disadvantages. There are some hybrid mechanisms proposed to tackle the disadvantages of a specific algorithm by making use of the advantages of the others. To improve the application of the multi-objective particle swarm optimization with fitness sharing (MO-PSO) for the FRBC design method proposed in [33], this paper represents an application of a hybrid multi-objective particle swarm optimization algorithm with simulated annealing behavior (MOPSO-SA) to optimize the semantic parameters of the linguistic variables and fuzzy rule selection in designing FRBCs based on hedge algebras proposed in [7] which uses the genetic simulated annealing algorithm (GSA). By simulation, the MOPSO-SA has shown to be more efficient and produced better results than both the GSA algorithm in [7] and the MO-PSO algorithm in [33]. That is, to show a method of the FRBC design is better than another one using MOEA, the same MOEA must be used.

Manuscript communication: received 11 January 2014, revised 28 July 2014, accepted 18 September 2014
Corresponding author: Phong Pham Dinh, dinhphong_pham@yahoo.com

Keywords: Fuzzy Classification System, Hedge Algebras, Particle Swarm Optimization, PSO, Simulated Annealing.

1. Introduction

In recent years, the fuzzy rule based system (FRBS) which is composed of fuzzy rules in the form of if-then sentences has had many successful applications in some different fields. The fuzzy rule based classification system (FRBCS) is the simplest model of the FRBS. One of the concerned study trends in this field is the fuzzy rule based classifier (FRBC) design and has achieved many successful results. In several works in the fuzzy set theory approach [1-4], the fuzzy partitions and the linguistic labels of their fuzzy sets are fixed and pre-specified and, when it is necessary, only the fuzzy set parameters are adjusted using MOEAs.

Hedge algebras (HAs) [5-10] are mathematical formalism that allows to model and design the linguistic terms along with their fuzzy set for the FRBCs. By utilizing this formalism, the concepts of the fuzzy model [10], fuzziness measure, fuzziness intervals of terms and semantically quantifying mappings (SQMs) of

* This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2013.34.

hedge algebras have been introduced and examined [7, 9]. The fuzzy measures of the hedges and a primary term are called the fuzziness parameters and when they combine with a positive integer to limit the term lengths commonly called the semantic parameters, denoted by ***Л***. The SQM-values of the terms, which can be computed based on the given values of the fuzziness parameters, can be regarded as the cores of the fuzzy sets that represent the semantics of the respective terms. Utilizing these values, the triangular fuzzy sets of terms can be generated. Based on this, a method for designing linguistic terms along with their fuzzy sets for FRBCs can be developed [11] and it determines a method to design FRBCs using MOEAs, the GSA algorithm is used in [11]. For more specific, this method comprised two phases: the first phase is to generate linguistic term along with their triangular fuzzy set based semantics for each dataset feature. The GSA algorithm is used to find the optimal semantic parameter values. The second phase is to generate an optimal FRBCS from a given dataset with the semantic parameter values provided by the first phase. The MOEA used in this phase is also a GSA algorithm.

There are also many other MOEAs that can be used instead of those based on the GSA algorithm, the particle swarm optimization algorithm (PSO), for instance. They are examined intensively, e.g. in [12-20] and applied in the field of classification [21-25]. An application of PSO-based MOEA instead of the GSA-based MOEA to develop a hedge algebra based methodology algorithm for designing FRBC [11] proposed in [26]. The MO-PSO is shown to be more efficient and produces better results than the GSA algorithm. But, the disadvantage of the PSO is that it depends on the random initial state, i.e., if the initial solutions take the search closer to a local optimal solution, the particles will converge towards that solution and do not have ability to jump out to search for a global optimal solution. To overcome this shortcoming of the MO-PSO, the simulated annealing (SA) algorithm [27, 28] can be utilized to help the particles jump out of the local optimums to do further searching.

The purpose of the paper is to represent an application of a hybrid multi-objective PSO with fitness sharing proposed in [12] and the simulated annealing algorithm [27, 28], abbreviated as MOPSO-SA, to develop a hedge algebra based methodology algorithm for designing FRBC [11] in such a way that the MOPSO-SA is used instead of the GSA based MOEA. This ensures that two such methods are the same, except the MOEAs applied.

The experimental results have statistically shown that the MOPSO-SA based method is more effective than the GSA and the MO-PSO based methods under the condition that the number of the generations of the three methods is the same. That is, statistically, the FRBCS produced by the MOPSO-SA based method have higher classification accuracy, but the complexity is not higher than those obtained by both the GSA and the MO-PSO based methods. This shows that the role of the MOEAs should be taken into account in a comparative study of two FRBC design methods in question.

The rest of this paper is organized as follows: Section II is a brief description of fuzzy rule based classifier design based on hedge algebras. Section III represents the MOPSO-SA algorithm. Section IV discusses the application of MOPSO-SA algorithm for the fuzzy rule based classifier design based on hedge algebras. Section V shows the experimental results and discussion. Concluding remarks are included in Section VI.

2. Fuzzy rule based classifier design based on the hedge algebra methodology

The knowledge of the fuzzy rule based classification system used in this paper is the weighted fuzzy rules in the following form [2, 11]:

Rule R_q: IF X_1 is $A_{q,1}$ AND ... AND X_n is $A_{q,n}$

THEN C_q with CF_q, for $q=1,\ldots,N$ (1)

where X = $\{X_j, j = 1, .., n\}$ is a set of n linguistic variables corresponding to n features of the dataset, $A_{q,j}$ is the linguistic terms of the j^{th} feature F_j, C_q is a class label, each dataset includes M class labels, and CF_q is the weight of the rule R_q. In short, the rule R_q can be written as:

$$A_q \Rightarrow C_q \text{ with } CF_q, \text{ for } q=1,\dots, N \tag{2}$$

where $\boldsymbol{A}_q$ is the antecedent part of the q^{th}-rule.

A fuzzy rule based classification problem P is defined as: a set $\boldsymbol{P} = \{(\boldsymbol{d}_p, C_p) \mid \boldsymbol{d}_p \in \boldsymbol{D}, C_p \in \boldsymbol{C}, p = 1, \dots, m;\}$ of m patterns, where $\boldsymbol{d}_p = [d_{p,1}, d_{p,2}, \dots, d_{p,n}]$ is a row of m data patterns, $\boldsymbol{C} = \{C_s \mid s = 1, \dots, M\}$ is the set of M class labels.

Solving the FRBC design problem is to extract from P a set S of fuzzy rules in the form (1) such that the FRBCS based on S comes with high performance, interpretability and comprehensibility. The FRBC design method based on the HA comprises two phases:

1. Design automatically the optimal linguistic terms and their fuzzy-set-based semantics for each dataset feature. An evolutionary multi-objective optimization algorithm is constructed to find a set of linguistic terms together with their respective fuzzy-set-based semantics for the problem P in such a way that its outputs are the consequences of the interaction between the semantics of terms and the data.

2. Extract fuzzy rule bases from a specific dataset to achieve their suitable interpretability–accuracy tradeoff. Based on the variety and suitability of the fuzzy linguistic terms provided in the first phase, the aim of this phase is to generate a fuzzy rule base having suitable interpretability-accuracy tradeoff to solve P.

In the first step of the first phase [11], each j^{th} feature of the specific dataset P is associated with a hedge algebra AX_j. Based on the given values of the semantic parameters $\boldsymbol{\Pi}$ comprising the fuzziness measure $fm_j(c^-)$ of the primary term c^-, the fuzziness measure $\mu(h_{j,i})$ of the hedges and a positive integer k_j for limiting the designed term lengths of j^{th} feature, the fuzziness intervals $\mathrm{I}_k(x_{j,i})$, $x_{j,i} \in X_{j,k}$ for all $k \le k_j$ and the SQM values $v(x_{j,i})$ are computed. Then, the triangular-fuzzy-set-based semantics of the terms in $X_{j,(kj)}$ will computationally be constructed by utilizing the SQM-values of the terms. The $X_{j,(kj)}$ is the union of the sets $X_{j,k}$, $k = 1$ to k_j, and the fuzziness intervals of the terms in each $X_{j,k}$ constitute a binary partition of the feature reference space. For example, the fuzzy sets of terms with $k_j = 2$ is denoted in Fig. 1.

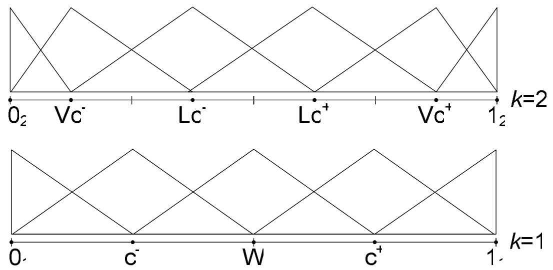

Fig. 1. The fuzzy sets of terms in case of $k_j = 2$.

After the fuzzy-set-based semantics of terms are constructed, the next step is to generate fuzzy rules from the dataset P. Then, a screening criterion is used to select NR_0 fuzzy rules, so-called the initial fuzzy rule set, denoted by S_0. All these steps form a so-called initial fuzzy rule set generation procedure and named as IFRG($\boldsymbol{\Pi}$, P, NR_0, λ) [11], where $\boldsymbol{\Pi}$ is a set of the semantic parameters obtained from the first step and λ is the maximum of rule length.

For a specific dataset, the different pre-specified semantic parameter values give us the different classification results (performance, the number of rules and the average rule length of the fuzzy rule bases). Therefore, in order to obtain the classification results as best as possible, an MOEA is used to find the optimal semantic parameter values for generating S_0. The number of the initial fuzzy rules NR_0 is large enough so that the applied evolutionary algorithm can produce an expected optimal solution.

In the second phase, the obtained optimal semantic parameter values are taken to be the input of the initial fuzzy rule set generation procedure to generate an NR_0 fuzzy rule set S_0. In this procedure, a screening criterion can be used to select S_0. Then, a MOEA is applied to select an optimal fuzzy rule base from S_0 having suitable interpretability-accuracy tradeoff for the desired FRBC.

3. Hybrid multi-objective pso-sa algorithm

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995 [13, 14]. Since then it has had many applications to the optimization problems [21-26, 31, 32]. The main idea of this technique is based on the way that birds travel when trying to find sources of food, or similarly, the way that a fish school will behave. The model of this algorithm is that the particles (or individuals) are treated as solutions inside the swarm (or population). The particles will move or travel through the solution space of the problem to search for the best solutions. PSO is very efficient for global search and just needs very few algorithm parameters. It is the fact that, similar to the genetic algorithm, it is easy to be trapped into local optimums during the search process and becomes premature convergence. Because of the velocity update equation, it is difficult for particles to jump out of the local optimums and continues the searching process. On the contrary, by using the "Metropolis law" during the search process, the simulated annealing (SA) algorithm [27, 28] has probability to jump out of the local optimums to do further searching. However, the disadvantage of SA compared to PSO is that the slow temperature variations are required leading to calculate time increasing. Therefore, this paper presents a hybrid multi-objective particle swarm optimization algorithm with simulated annealing behavior to solve the problem of FRBC design based on hedge algebras methodology. The proposed hybrid algorithm combines the advantages of both the SA and the PSO algorithms.

Multi-objective PSO algorithm with fitness sharing

The original PSO has been implemented to solve the single-objective problems (SOO) and it did not use crossover and mutation operators. There are many multi-objective optimization (MOO) problems need to be solved in the real-life. This type of problem becomes challenging because of the inherent conflicting among the optimized objectives. The PSO is one of the competing heuristic algorithms to solve the MOO problems. Some improved PSO algorithms have been developed to support this type of problem [12, 15-20] since 2002. One of them is the algorithm introduced in [12] that integrates the fitness sharing concept into the original PSO to improve the PSO technique to solve the MOO problems. The concept of fitness sharing can be found in [29].

The formula of the *fitness sharing* of a particle *i* is calculated as:

$$fshare_i \frac{f_i}{\sum_{j=0}^{n} sharing_i^j} \tag{3}$$

where *n* is the number of particles in the swarm,

$$sharing_i^j = \begin{cases} 1-(d_i^j/\sigma_{share})^2 & \text{If } d_i^j < \sigma_{share} \\ 0 & \text{Otherwise} \end{cases} \quad 4)$$

σ_{share} is calculated based on the farthest distance between particles in the repository, d_i^j is the distance between particle i and j.

$$d_i^j = \sqrt{(particle_i - particle_j)^2} \tag{5}$$

With the multi-objective problems, we can get more than one solution. So, the authors in [12] use the concept of Pareto dominance to collect the set of best solutions. The *Pareto dominance* and the *non-dominated set* concepts can be found in [12].

The main idea in [12] is use the fitness sharing concept to share the fitness functions of the MOO problems. This technique integrated with the dominance concepts improves the search of the particles. To do so, in each algorithm loop, the best particles found so far called non-dominated particles are stored in an external repository and the fitness sharing of each particle is calculated based on them. So in the next iterations, a set of non-dominated solutions are maintained. After the run, the set of particles in the external repository is the best found solutions which form the Pareto front.

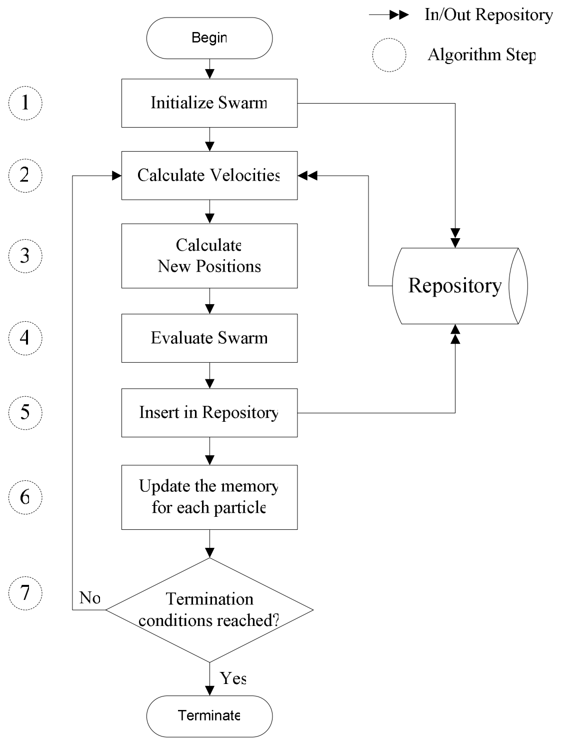

Fig. 2. An adapted diagram of the MOO algorithm [12]

The flow chart of the MO-PSO algorithm with fitness sharing proposed in [12] is shown in Fig. 2. Hereafter is a brief explanation of the algorithm step by step (for more details, see [12]):

1. All variables (pop_i, $pbest_i$, $gbest_i$, $fShare_i$) are initialized. The fitness value of each particle is evaluated. The value of fitness sharing of each particle $fShare_i$ is calculated as:

$$fShare_i = \frac{x}{nCount_i} \tag{6}$$

where $x = 10$. The $nCount_i$ value is calculated as:

$$nCount_i = \sum_{j=0}^{n} sharing_i^j \tag{7}$$

where n is the number of particles in the external repository and $sharing_i^j$ value is calculated by the formula (4).

2. Calculate the particle's velocity as:

$$vel_i = \omega \times vel_i + c_1 \times r_1 \times (pbest_i - pop_i) + c_2 \times r_2 \times (gbest_h - pop_i) \tag{8}$$

where ω is an inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random numbers between 0 and 1, vel_i is the previous velocity value, $pbest_i$ is the local best position, $gbest_h$ is the global best position and pop_i is the current particle's position.

3. Calculate the new particle position as:

$$pop_i = pop_i + vel_i \tag{9}$$

4. Evaluate the fitness value of each particle.

5. Update the external repository based on the dominance and fitness sharing concepts (see [12]).

6. Update the particle memory based on the dominance criteria (see [12]).

7. If the termination condition is reached, the algorithm will terminate. Otherwise, go to step 2.

Simulated Annealing Algorithm

The simulated annealing (SA) algorithm [27, 28] is a probabilistic hill-climbing technique. It is based on the freezing of liquids or the cooling process of metals in the process of annealing. The cooling process starts at a high temperature (T_{max}) which the metal is in the molten state. After the heat source is removed, the metal temperature commences to decrease gradually to the surrounding ambient temperature (T_{min}) at which the metal energy reaches the lowest value and the metal is perfectly solid. Hereafter is the brief explanation of the SA algorithm in case the energy of the system is minimized:

Step 1: Initialize the initial configuration with the energy E, the cooling rate $\alpha \in [0, 1]$ and the initial temperature $T = T_0$ which is high enough to avoid local convergence, but not too high to prevent the searching time from increasing too much.

Step 2: Calculate the change of energy ΔE of the configuration.

Step 3: If ΔE is negative, the new configuration is accepted. If ΔE is positive, the new configuration is accepted with a probability $P = e^{-(\Delta E / k_B T)}$, where k_B is the Boltzman constant. This mechanism is called the metropolis acceptance rule.

Step 4: If the termination condition is reached, the process is terminated. Otherwise, decrease the temperature $T = \alpha \times T$ and go to Step 2.

The implementation difficulties of this algorithm are how to choose the initial temperature, how many iterations are performed at each temperature and how slowly the temperature is decreased. E.g., if the initial temperature is too low, it can be trapped in a local optimum state. Whereas, if the initial temperature is too high, the searching time is inevitably increased.

The Proposed Hybrid Multi-objective PSO-SA Algorithm

The proposed hybrid multi-objective PSO-SA is an integration of the MO-PSO and the SA algorithms, so-called the MOPSO-SA algorithm. This hybrid algorithm makes use of the global search provided by the PSO and local search provided by the SA. A brief explanation of this algorithm is as below:

Step 1: According to the MO-PSO structure, let t = 0, and n particles of the swarm are randomly created. All variables are initialized including the initial temperature $T_0 = T_{\max}$ and cooling rate α, the number of generations or cycles G_{max}. The fitness value of each particle is evaluated. The fitness sharing value of each particle is calculated as formula (6).

Step 2: For each particle i in the swarm.

Step 2.1: Calculate the particle's velocity vel_i^{t+1} as formula (8).

Step 2.2: Calculate the new particle position pop_i^{t+1} as formula (9).

Step 2.3: Evaluate the objective values of the particle i.

Step 2.4: Check the dominance criteria between the new position pop_i^{t+1} and the previous one pop_i^t. If the position pop_i^{t+1} dominates pop_i^t, meaning that the new position is better, then pop_i^{t+1} is accepted as the new position of particle i. Otherwise, calculate the root mean squared residual of the current position and the previous one:

$$RMSR = \frac{1}{D}\sqrt{\sum_{j=1}^{D}(fitness_{i,j}^{t+1} - fitness_{i,j}^{t})^2} \qquad (10)$$

where D is the number of objectives. Generate a random number $\delta \in [0, 1]$. The new position is accepted if $\delta > e^{-(RMSR/T_t)}$ or the number of failures is greater than 100. If the new position is accepted, go to Step 2. Otherwise, go to Step 2.1.

Step 3: Update the external repository based on the dominance and fitness sharing concepts.

Step 4: Update the particle memory based on the dominance criteria.

Step 5: If the termination condition is reached, the algorithm will terminate and output the set of the best solutions stored in the external repository. Otherwise, modify the annealing temperature $T_{t+1} = \alpha \times T_t$, let $t = t + 1$, and go to Step 2.

The proposed hybrid algorithm explores the entire searching space by the multi-objective PSO technique to approach the global optimal area. Whereas, the SA technique helps to do the gradient search within a localized region for improving the ability of finding the global optimal solution. In the Step 2.4 of the multi-objective PSO, the metropolis acceptance rule is applied by utilizing the so-called root mean squared residual (RMSR) measure calculated as (10), i.e., the new position of particle i is accepted if it dominates the one in the previous generation. Otherwise, it is accepted if the probability $\delta > e^{-(RMSR/T_t)}$, where RMSR is the root mean squared residual of the current position and the previous one, or continues the search with the failing accepted particle with the same evaluation process. If many failures occur with the same particle, in this study is 100, the last position is accepted to avoid an endless loop. The annealing temperature is decreased gradually by the cooling rate α after each iteration, where t is the iteration step number.

4. Hybrid multi-objective pso-sa algorithm for designing optimal linguistic terms and fuzzy rule selection

In the fuzzy rule based classifier design method based on HAs examined in [11], the semantic parameters of linguistic variables (features) that originate from the inherent qualitative semantics of terms are used instead of the fuzzy set parameters. They have essential advantages, e.g., they permit designing linguistic terms integrated with their fuzzy set based semantics; they depend only on their own linguistic variables, not on individual terms; in comparison with the number of fuzzy set parameters, the number of semantic parameters is very small; and so on. In that paper, the GSA algorithm with weighted fitness function is applied to find the optimal semantic parameter values for each dataset feature. When having the optimal semantic parameter values, they are used as the inputs of the fuzzy rule genetic selection algorithm to achieve a fuzzy rule base having suitable interpretability–accuracy tradeoff. In [26], the MO-PSO is used instead of the GSA algorithm and has better results of both the classification accuracy and the complexity of FRBCSs. This section represents the application of the MOPSO-SA for the semantic parameter optimization and the optimal fuzzy rule selection processes.

Having a set of given semantic parameter values of the j^{th} feature, a finite set of terms and their fuzzy sets is completely determined. So, the search for the set of the optimal semantic parameter values of all features of a given dataset means that the term-sets of those features are optimally designed for that dataset.

In [11], a problem of designing optimal linguistic terms for any given classification problem P is formulated by utilizing the GSA-MOEA, named as GSA-SPO [11], which is generally described as follows:

(i) The aim of the algorithm is to find a set $\boldsymbol{\Pi}$ of the semantic parameter values of every j^{th} feature obeying the following constraints:

- On the fuzziness parameters:

$a \le fm_j(c^-) \le b, fm_j(c^-) + fm_j(c^+) = 1, a \le \mu(h_{j,i}) \le b, \sum_{h_{j,i} \in H_j} \mu(h_{j,i}) = 1, j = 1, \ldots, n.$ (11)

- On the integer k_j: $0 < k_j \le K, j = 1, \ldots, n$, where K is a given positive integer indicating an upper bound of the term lengths of all features.

That make

$perf(Cl(\boldsymbol{S}_0(\boldsymbol{\Pi}))) \to$ Max and $avg(Cl(\boldsymbol{S}_0(\boldsymbol{\Pi}))) \to$ Min (12)

where $Cl(\boldsymbol{S}_0(\boldsymbol{\Pi}))$ is the classifier whose fuzzy rule base is the initial fuzzy rule set generated by IFRG($\boldsymbol{\Pi}$, $\boldsymbol{P}$, NR_0, λ) procedure examined in [11]. *perf* denotes the accurate classification of the training set, *avg* denotes the average length of the antecedent of fuzzy rule based system.

(ii) Initialize a population Pop_0. For each individual of the population Pop_0 consisting of a set of values $\boldsymbol{\Pi}_{0,i}$ of the semantic parameters, calculate its fitness based on the objectives given in (12). Repeat the step of calculating the next generation Pop_{t+1}, for every t, using genetic operators. The loop is terminated when the termination condition is met.

During the evolutionary optimization, the linguistic terms of the designated feature are generated with the term lengths limited by k_j. Then, the values of the fuzziness parameters $\boldsymbol{\Pi}$ of the designated feature are immediately generated. In turn, they determine the fuzzy sets of the linguistic terms which create the multiple with granularities of the feature. To evaluate the learning process, the values of all objectives are computed. The learning process is repeated in order to produce better linguistic terms integrated their fuzzy sets.

To serve the purpose of the study as discussed previously, the new algorithm ***MPSOSA_SPO*** structured hereafter is essentially the same as the above GSA-SPO except its evolutionary procedure:

Algorithm MOPSOSA_SPO (semantic parameter optimization)

Input: The dataset $\boldsymbol{P} = \{(\boldsymbol{d}_p, C_p) \mid p = 1, \ldots, m\}$;

Parameters: $a, b, NR_0, N_{pop}, G_{max}, K, \lambda, T_{max}, \alpha$;

//N_{pop} is the swarm size, G_{max} is the number of generations.

Output: the set of the optimal semantic parameter values $\boldsymbol{\Pi}_{opt}$.

Begin

Randomly initialize a swarm $pop_0 = \{\boldsymbol{\Pi}_{0,i} \mid i = 1, \ldots, N_{pop}\}$;

$T_0 = T_{max}$;

For i =1 **to** N_{pop} **do begin**

Generate the fuzzy rule set $\boldsymbol{S}_0(\boldsymbol{\Pi}_{0,i})$ from $\boldsymbol{\Pi}_{0,i}$ by applying the algorithm IFRG($\boldsymbol{\Pi}_{0,i}$, $\boldsymbol{P}$, NR_0, λ);

Compute the value of all objectives for particle i using the given semantic parameter values $\boldsymbol{\Pi}_{0,i}$;

Set the particle memory $pbest_i$ to the current location;

End;

Fill the external repository *gbest* with all the non-dominated particles;

Calculate the value of Fitness sharing *fShare* for all particles in the repository;

$t = 0$;

Repeat

Assign a leader from the repository to particles;

For i =1 **to** N_{pop} **do begin**

Repeat

Update the velocity vel_i^{t+1} of particle i using (8);

Calculate the new position pop_i^{t+1} of particle i using formula (9);

Generate the fuzzy rule set $\boldsymbol{S}_0(\boldsymbol{\Pi}_{t,i})$ from $\boldsymbol{\Pi}_{t,i}$ by applying the algorithm IFRG($\boldsymbol{\Pi}_{t,i}$, $\boldsymbol{P}$, NR_0, λ);

Evaluate the value of all objectives for particle i;

If the new position pop_i^{t+1} dominates pop_i^t **then**

Accept pop_i^{t+1} as the new position of particle i;

Else

Calculate the root mean squared residual (*RMSR)* of the current position and the previous one as formula (10);

Generate a random number $\delta \in [0, 1]$;

If $\delta > e^{-(RMSR \times 1000/T_t)}$ **or** the number of failures is greater than 100 **then**

The new position pop_i^{t+1} is accepted;

End If;

End If;

Until the new position is accepted or the number of failures is greater than 100;

End;

Update the repository *gbest* with current best solutions found by the particles;

Update Fitness sharing of all particles if the repository is changed;

Update the memory *pbest* of all particles with the criteria of dominance;

$T_{t+1} = \alpha \times T_t$;

$t = t + 1$;

Until $t = G_{max}$;

Return the set of the best semantic parameter values $\boldsymbol{\Pi}_{opt}$ from the set of the best solutions in the repository;

End.

The ***MOPSOSA_SPO*** algorithm is implemented by utilizing the hybrid algorithm MOPSO-SA described in the previous section to find the optimal semantic parameter values for each dataset feature of the fuzzy rule based classifier design problem. In this application, the value of the root mean squared residual is quite small ($0 < RMSR < 1$) leading to the value of the expression $e^{-(RMSR/T_i)}$ is contiguous to 1. Thus, the ability of jumping out a local optimal search is reduced, so the searching time is increased accordingly. To overcome this shortcoming, the *RMSR* value is multiplied by 1000.

After the learning process, a set of the best semantic parameter values $\boldsymbol{\Pi}_{opt}$ is produced. We take any one of them, $\boldsymbol{\Pi}_{opt,i^*}$, to generate the initial fuzzy rule set $\boldsymbol{S}_0(\boldsymbol{\Pi}_{opt,i^*})$ using IFRG($\boldsymbol{\Pi}_{opt,i^*}$, $\boldsymbol{P}$, NR_0, λ) containing NR_0 fuzzy rules. The problem now is to select a subset of fuzzy rules $\boldsymbol{S}$ from $\boldsymbol{S}_0$ satisfying the following objectives:

maximize $perf(\boldsymbol{S})$,

maximize $NR(\boldsymbol{S})^{-1}$ and,

maximize $avg(\boldsymbol{S})^{-1}$, obey to the constraints

$$\boldsymbol{S} \subset \boldsymbol{S}_0,\ NR(\boldsymbol{S}) \leq N_{max}, \qquad (13)$$

where $NR(\boldsymbol{S})^{-1}$ and $avg(\boldsymbol{S})^{-1}$ are the inverses of $NR(\boldsymbol{S})$ and $avg(\boldsymbol{S})$ respectively. N_{max} is the pre-specified positive integer limiting the number of the fuzzy rules in $\boldsymbol{S}$ in the learning process of the algorithm. The MOPSO-SA algorithm is utilized again for the optimal fuzzy rule set selection and it is named as ***MOPSOSA_RBO***.

The real encoding of individuals is used for the ***MOPSOSA_RBO*** algorithm. Each individual corresponds to a solution of the problem represented as a real number string $r_i = (p_1, ..., p_{Nmax})$, $p_j \in [0, 1]$. Each fuzzy rule R_i of the candidate fuzzy rule set $\boldsymbol{S}$ for the desired FRBC is selected from $\boldsymbol{S}_0(\boldsymbol{\Pi}_{opt,i^*})$. The zero based index of the fuzzy rule R_i in $\boldsymbol{S}_0$ is calculated as $p_j \times |\boldsymbol{S}_0|$ with $0 \leq p_j \times |\boldsymbol{S}_0| < |\boldsymbol{S}_0|$.

$$\boldsymbol{S} = \{R_i \in \boldsymbol{S}_0 \mid i = \lfloor p_j \times |\boldsymbol{S}_0| \rfloor,\ i \geq 0\} \qquad (14)$$

where $\lfloor \bullet \rfloor$ is the integer portion of a real number.

The ***MOPSO_RBO*** algorithm is structured similarly as the ***MOPSO_SPO*** algorithm with suitable changes. The output of the ***MOPSO_RBO*** procedure for a specific dataset is a set of near optimal solutions, from which we can choose the best one, that is the solution whose corresponding FRBCS has the best classification performance with respectively low complexity measured by the total number of the conditions of its rule base.

5. Experimental results and discussion

This section presents the experimental results of applying the proposed MOPSO-SA algorithm to the FRBC design based on hedge algebras methodology over some standard classification datasets that can be found on the KEEL-Dataset repository: *http://sci2s.ugr.es/keel/datasets.php* and the comparisons with the ones proposed in [11] and [26]. To make a comparative study, the same cross validation method should be applied with the same folds. Therefore, we apply the *ten-folds cross-validation* method to every dataset, i.e., each dataset is randomly partitioned into ten folds, nine folds for the training phase and one fold for the testing phase. Three trials of each algorithm are executed for each of the ten folds and hence it permits to design 30 (= 3 *times* × 10 *folds*) fuzzy rule based classification systems. The results of the classification performance and the complexity of the 30 designed fuzzy rule based classification systems of each dataset are averaged out respectively.

To limit the searching space in the learning process, the same constraints on the semantic parameter values are applied as examined in [11].

I.e., we have: the number of both negative hedge and positive hedge is 1, and assume that the negative hedge is L and the positive hedge is V; $0 \le k_j \le 3$; $0.2 \le fm_j(c^-) \le 0.8$; $fm_j(c^-) + fm_j(c^+) = 1$; $0.2 \le \mu_j(L) \le 0.8$ and $\mu_j(L) + \mu_j(V) = 1$.

The ***MOPSOSA_SPO*** algorithm has been run with the following parameters: the number of generations: 250, the same as examined in [11]; the number of particles of each generation: 300; Inertia coefficient: 0.4; the self cognitive factor: 0.2; the social cognitive factor: 0.2; the number of initial fuzzy rules is equal to the number of attributes; the maximum of rule length is 1.

The ***MOPSOSA_RBO*** algorithm has been run with the same parameters of the ***MOPSOSA_SPO***, except the number of generations: 1000; the number of particles of each generation: 600; the number of initial fuzzy rules $|S_0|$ = *300* × *number of classes*; the maximum of rule length is 3 if the number of attributes is less than 30, otherwise the maximum of rule length is 2.

The parameters of the SA for both the ***MOPSOSA_SPO*** and the ***MOPSOSA_RBO*** algorithms: the initial temperature: $T_0 = 90$; the cooling rate: $\alpha = 0.995$.

The real-world datasets considered in this study, which comprise the high dimensional datasets (the number of attributes is greater than and equal to 30) and the multi-class datasets (the number of classes is greater than 2) are listed in the Table I.

The experimental results of the application of the MOPSO-SA, the MO-PSO and the GSA algorithms for the FRBC design are shown in Table II and Table III, where note that $\#R$ is the number of fuzzy rules in the extracted fuzzy rule set; $\#C$ is the number of conditions of the fuzzy rule set; $\#R*\#C$ is the complexity; P_{tr} is the performance in the training phase and P_{te} is the performance in the testing phase.

TABLE I. THE LIST OF DATASETS CONSIDERED IN THE STUDY

No.	Dataset name	Number of attributes	Number of classes	Number of patterns
1	Australian	14	2	690
2	Bands	19	2	365
3	Bupa	6	2	345
4	Dermato.	34	6	358
5	Haberman	3	2	306
6	Ionosphere	34	2	351
7	Pima	8	2	768
8	Saheart	9	2	462
9	Vehicle	18	4	846
10	Wdbc	30	2	569
11	Wine	13	3	178
12	Wisconsin	9	2	683

TABLE II. EXPERIMENTAL RESULTS OF 10-*FOLDS* CROSS VALIDATION ON 12 DATASETS BY APPLYING THE MOPSO-SA AND THE GSA ALGORITHMS

No.	Dataset	MOPSO-SA algorithm			GSA algorithm [11]			≠Pte
		#R*#C	P_{tr}	P_{te}	#R*#C	P_{tr}	P_{te}	
1	Australian	46.86	88.27	86.47	43.00	87.83	86.18	0.29
2	Bands	63.00	77.79	73.50	83.40	75.57	70.63	2.87
3	Bupa	186.68	80.91	70.02	196.37	77.40	67.71	2.31
4	Dermato.	217.77	98.26	96.07	194.61	98.82	95.52	0.55
5	Haberman	9.79	76.98	76.72	11.30	76.78	75.11	1.61
6	Ionosph.	110.21	95.74	91.66	91.73	94.60	90.21	1.45
7	Pima	61.20	79.15	76.35	51.17	79.03	75.70	0.65
8	Saheart	96.37	77.03	71.15	107.57	74.91	68.99	2.16
9	Vehicle	237.47	71.66	68.01	324.98	70.59	67.46	0.55
10	Wdbc	39.67	97.79	96.32	45.86	96.51	94.90	1.42
11	Wine	37.40	99.54	98.30	65.17	99.79	98.30	0.00
12	Wisconsin	55.97	97.95	97.22	67.42	98.38	96.72	0.50

The *≠Pte* column represents the differences of the performances of the comparison methods. Specifically, the comparison results between the MOPSO-SA and the GSA-based methods in the Table II show that *all performance differences are positive*. The comparison results between the MOPSO-SA and the MO-PSO methods in the Table III show that there is only one negative performance difference. Intuitively, the MOPSO-SA is better than both the MO-PSO and the GSA-based methods. However, the final conclusion should rely upon the statistic studies given in the Table IV and V in which the Wilcoxon's signed-rank tests [30] have been applied to test the complexities and performances of the fuzzy rule bases extracted by three methods respectively. We assume that the two compared versions are statistically equivalent (null-hypothesis).

TABLE III. EXPERIMENTAL RESULTS OF 10-*FOLDS* CROSS VALIDATION ON 12 DATASETS BY APPLYING THE MOPSO-SA AND THE MO-PSO ALGORITHMS

No.	Dataset	MOPSO-SA algorithm			MO-PSO algorithm[26]			≠Pte
		#R*#C	P_{tr}	P_{te}	#R*#C	P_{tr}	P_{te}	
1	Australian	46.86	88.27	86.47	36.20	88.06	86.38	0.09
2	Bands	63.00	77.79	73.50	52.20	76.17	72.80	0.70
3	Bupa	186.68	80.91	70.02	190.00	78.91	69.64	0.38
4	Dermato.	217.77	98.26	96.07	198.05	98.03	96.07	0.00
5	Haberman	9.79	76.98	76.72	10.20	76.91	75.76	0.96
6	Ionosph.	110.21	95.74	91.66	90.03	95.35	90.22	1.44
7	Pima	61.20	79.15	76.35	60.89	78.28	76.18	0.17
8	Saheart	96.37	77.03	71.15	86.70	76.35	69.33	1.82
9	Vehicle	237.47	71.66	68.01	240.93	70.54	67.30	0.71
10	Wdbc	39.67	97.79	96.32	37.30	97.62	96.96	-0.64
11	Wine	37.40	99.54	98.30	35.80	99.86	98.30	0.00
12	Wisconsin	55.97	97.95	97.22	74.40	97.81	96.74	0.48

TABLE IV. THE COMPARISON RESULT OF FUZZY RULE COMPLEXITIES OF THE MOPSO-SA, THE GSA AND THE MO-PSO (MPSO) ALGORITHMS USING THE WILCOXON SIGNED RANK TEST AT LEVEL *ALPHA* = 0.05.

VS	R^+	R^-	E. *P*-value	A. *P*-value	Conf. Inte.	Exact. Conf.
GSA	50.0	16.0	0.14746	0.119722	[-20.4 , 1.38]	0.917
MPSO	23.0	43.0	≥ 0.2	1	[-3.32 , 9.655]	0.917

VS column is the list of the methods which we want to compare with. The abbreviation terms used in the Table IV and V: E. is Exact; A. is Asymptotic; Inte. is Interval and Conf. is Confidence.

The comparison of the designed FRBCS complexities using Wilcoxon's signed-rank test at level α = 0.05 represented in the Table IV statistically states that the complexities of the FRBCSs designed by the MOPSO-SA method are similar to those of the respective FRBCSs designed by both the MO-PSO and the GSA-based methods because the null-hypothese cannot be rejected.

TABLE VI. THE COMPARISON RESULT OF THE FRBCS PERFORMANCES OF THE MOPSO-SA, THE GSA AND THE MO-PSO (MPSO) ALGORITHMS USING THE WILCOXON SIGNED RANK TEST AT LEVEL *ALPHA* = 0.05.

VS	R^+	R^-	E. *P*-value	A. *P*-value	Conf. Inte.	Exact. Conf.
GSA	66.0	0.0	9.766E-4	0.002897	[0.55 , 1.695]	0.90772
MPSO	69.5	8.5	0.014161	0.015022	[0.065 , 0.91]	0.90772

The comparison of the designed FRBCS performances using Wilcoxon's signed-rank test at level α = 0.05 is shown in the Table V. Since all R^- values which are the sum of the ranking results of the MOPSO-SA algorithm are less than the critical value of T Wilcoxon distribution [33] associated with the number of datasets N_{ds} = 10 (two equivalent results of Wine and Dermatology datasets are eliminated from the test) and p = 0.05, all the null-hypotheses are rejected. This critical value is 10 in this case and it can be found in the Table B.12 in [33]. Therefore, we can sate that the proposed MOPSO-SA based method for designing FRBCs outperforms both the MO-PSO and the GSA based methods, noting that the three methods are different from each other merely the applied evolutionary technique.

6. Conclusion

Fuzzy rule bases that deal with fuzzy information play an important role in designing FRBCs. HAs can be regarded as an algebraic model of the semantic-order-based structure of the term-domains of the linguistic variables so that it can be used to solve the FRBC design problem with the order based semantics of linguistic terms. This paper presents a method for improving the accuracy of the FRBC based on HA-methodology using the MOPSO-SA algorithm. In addition, the proposed method of designing FRBCs is developed mainly to be the same the one examined in [11], except the utilized evolutionary algorithm, where the MOPSO-SA based algorithm are utilized instead of the GSA based algorithm used in that paper. We also compare the MOPSO-SA algorithm with the MO-PSO algorithm proposed in [26]. Our experimental results with the same condition found that the MOPSO-SA-based algorithm to design FRBCs is better than both the MO-PSO and the GSA based algorithms. An important result is that to show a method of the FRBC design using MOEA is better than another, the same MOEA must be used.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2013.34.

References

[1] Chen Ji-lin, Hou Yuan-long, Xing Zong-y, Jia Li-min, Tong Zhong-zhi, A Multi-objective Genetic-based Method for Design Fuzzy Classification Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.8A, August 2006.

[2] H. Ishibuchi, T. Yamamoto, Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining, Fuzzy Sets and Systems, vol.141, no.1, 2004, pp. 59-88.

[3] R. Alcalá, Y. Nojima, F. Herrera, H. Ishibuchi, Generating single granularity-based fuzzy classification rules for multi-objective genetic fuzzy rule selection, in: Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju (Korea), 2009 pp. 1718-1723.

[4] Rafael Alcalá, Yusuke Nojima, Francisco Herrera, Hisao Ishibuchi, Multi-objective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions, Journal Soft Computing, V. 15, Issue 12, December 2011, 2303-2318.

[5] Ho N. C. and Wechler W. (1990), Hedge algebras: an algebraic approach to structures of sets of linguistic domains of linguistic truth variables, Fuzzy Sets and Systems, 35(3), pp. 281-293.

[6] Ho N. C. and Wechler W. (1992), Extended hedge algebras and their application to fuzzy logic, Fuzzy Sets and Systems, vol.52, pp. 259-281.

[7] C.H. Nguyen, V.N. Huynh, D.K. Tran, H.C. Le, Hedge Algebras, Linguistic-valued logic and their application to fuzzy reasoning, Internat. J.Uncertain. Fuzziness Knowledge-Based Systems, 7 (4) (1999) 347-361.

[8] Nguyen Cat Ho, Tran Thai Son, Tran Dinh Khang, Le Xuan Viet, Fuzziness Measure, Quantified Semantic Mapping And Interpolative Method of Approximate Reasoning in Medical Expert Systems, Journal of Computer Science and Cybernetics, V.18 (3) (2002), 237-252.

[9] N. C. Ho and N. V. Long (2007), Fuzziness measure on complete hedges algebras and quantifying semantics of terms in linear hedge algebras, Fuzzy Sets and Systems, vol.158, pp.452-471.

[10] C.H. Nguyen, A topological completion of refined hedge algebras and a model of fuzziness of linguistic terms and hedges, Fuzzy Sets and Systems, 158 (2007) 436-451.

[11] Cat Ho Nguyen, Witold Pedrycz, Thang Long Duong, Thai Son Tran, A genetic design of

linguistic terms for fuzzy rule based classifiers, International Journal of Approximate Reasoning, volumn 54, issue 1, January 2013, 1-21.

[12] Maximino Salazar Lechuga, Multi-Objective Optimisation using Sharing in Swarm Optimisation Algorithms, Doctor thesis, School of Computer Science, The University of Birmingham, 2006.

[13] J. Kennedy and R. C. Eberhart (1995). Particle Swarm Optimization, In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, New Jersey. IEEE Service Center, pp. 1942-1948.

[14] R. C. Eberhart and J. Kennedy (1995), A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43. Piscataway, NJ: IEEE Service Center.

[15] C.A.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, IEEE Press, Hawaii, 2002, pp. 1051-1056, May.

[16] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation, 8 (3) (2004) 256–279.

[17] X. Hu, R. Eberhart, Multiobjective Optimization Using Dynamic Neighbourhood Particle Swarm Optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, IEEE Press, Hawaii, 1677-1681.

[18] X. Li, A non-dominated sorting particle swarm optimizer for multi-objective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, vol. 2723, Springer, 2003, pp. 37-48.

[19] S. Mostaghim, J. Teich, Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (SIS'03), IEEE Service Center, Inidanapolis, Indiana, USA, 2003. April 26-33.

[20] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, Sankar Kumar Pal, Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients, Information Sciences, 177 (2007) 5033-5049.

[21] C. C. Chen, Design of PSO-based Fuzzy Classication Systems, Tamkang Journal of Science and Engineering, vol. 9, no. 1, pp. 63-70, 2006.

[22] Parag Puranik, Preeti Bajaj, Ajith Abraham, Prasanna Palsodkar, and Amol Deshmukh, Human Perception-based Color Image Segmentation Using Comprehensive Learning Particle Swarm Optimization, Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 3, (2011) 227-235.

[23] Hassan M. Elragal, Using Swarm Intelligence for Improving Accuracy of Fuzzy Classifiers, International Journal of Electrical and Computer Engineering, Vol. 5, Issue 2, 2010, pp. 105-112.

[24] Yudong Zhang, LenanWu, A Robust Hybrid Restarted Simulated Annealing Particle Swarm Optimization Technique, Advances in Computer Science and its Applications, Vol. 1, No. 1, 2012, pp. 5-8.

[25] S. G. Sanjeevi, A. Naga Nikhila, Thaseem Khan, G. Sumathi, Comparison of Hybrid PSO-SA Algorithm and Genetic Algorithm for Classification, Vol 3, No.2, 2012, pp. 37-45.

[26] P. D. Phong, N. C. Ho, N. T. Thuy, Multi-objective Particle Swarm Optimization Algorithm and its Application to the Fuzzy Rule Based Classifier Design Problem with the Order Based Semantics of Linguistic Terms, In proceeding of The 10th IEEE RIVF International Conference on Computing and Communication Technologies (RIVF-2013), November 2013, Hanoi, Vietnam, pp 12 - 17.

[27] Horng-Lin Shieh, Cheng-Chien Kuo, Chin-Ming Chiang, Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification, Applied Mathematics and Computation, Volume 218, Issue 8, December 2011, pp. 4365-4383.

[28] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing, Science, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.

[29] Goldberg D. E., Richardson, J. (1987). Genetic Algorithms with Sharing for Multimodal Function Optimization. In Grefenstette, J., editor, Proceedings of the 2nd International Conference on Genetic Algorithms, pages 41–49, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

[30] Janez Demˇsar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, 7 (2006) 1-30.

[31] Su-Fang An, Bo Liu, Shuang Zhao, Kun-Qi Liu, Improved Weighted Fuzzy Reasoning Algorithm Based on Particle Swarm Optimization, International Conference on Machine Learning and Cybernetics, Volume 3, 9-22 Aug. 2007.

[32] Rehab F. Abdel-Kader, Fuzzy Particle Swarm Optimization with Simulated Annealing and Neighborhood Information Communication for Solving TSP, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 5, 2011, pp. 15-21.

[33] J. Zar, Biostatistical Analysis, Prentice-Hall, Upper Saddle River, NJ, 1999, 758-760.