International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4391–4406

A NEW SIMPLIFIED SWARM OPTIMIZATION (SSO) USING EXCHANGE LOCAL SEARCH SCHEME

Changseok Bae[1], Wei-Chang Yeh[2], Noorhaniza Wahid[3]
Yuk Ying Chung[4] and Yao Liu[4]

[1]Personal Computing Research Team
Electronics and Telecommunications Research Institute
218 Gajeongro, Yuseonggu, Daejeon 305-700, Korea
csbae@etri.re.kr

[2]Advanced Analytics Institute
Faculty of Engineering and Information Technology
University of Technology Sydney
PO Box 123 Broadway, NSW 2007, Australia
profwcyeh@gmail.com

[3]Faculty of Science Computer and Information Technology
University Tun Hussein Onn Malaysia
86400 Parit Raja, Batu Pahat, Johor, Malaysia
nhaniza@uthm.edu.my

[4]School of Information Technologies
University of Sydney
Sydney, NSW 2006, Australia
vera.chung@sydney.edu.au; yliu5215@uni.sydney.edu.au

Received March 2011; revised October 2011

Abstract. *Swarm-based optimization algorithms have demonstrated to have effective ability to solve the classification problem in multiclass databases. However, these algorithms tend to suffer from premature convergence in the high dimensional problem space. This paper proposes a novel simplified swarm optimization (SSO) algorithm to overcome the above convergence problem by incorporating it with the new local search strategy. The proposed algorithm can find a better solution from the neighbourhood of the current solution produced by SSO. The performance of the proposed algorithm has been evaluated by using 13 different widely used databases and compared with the standard PSO and three other well-known classification algorithms. In addition, the practicability of the approach is studied by applying it in analysing golf swing from weight shift data. Empirical results illustrate that the proposed algorithm can achieve the highest classification accuracy.*
Keywords: Particle swarm optimization, Discrete particle swarm optimization, Simplified swarm optimization, Local search, Data classification, Data mining

1. **Introduction.** Data mining is the process of analysing data from different perspectives and summarizing it into useful information. It blends the traditional data analysis methods with sophisticated algorithms for processing large volumes of data [1]. It has been widely used and unifies research in fields such as statistics, databases, machine learning and artificial intelligence (AI). Regarding that, data mining has been seen as an explosion of interest from both academia and industry to alleviate the process of visualizing and understanding the pattern of the data. Data mining applies specific algorithm to extracting meaningful knowledge so that the discovered knowledge can be applied in the related areas to increase the working efficiency and also improve the quality of decision making [2].

The most commonly used data mining techniques include classification, data clustering, association rule discovery, and outlier detection. Data classification is one of the most common tasks in data mining that generates a set of rules from a set of training examples to classify future testing data. The classification system usually starts by generating a model from data instances (or learning examples) of labelled class, and finally classifies the new instances from the target variable with the use of a mapping from instances to classes.

The previous literature introduces the most commonly used conventional classification algorithms such as Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbour (KNN) and Neural Networks. SVM [3] is a well-known learning algorithm based on statistical learning theory [4]. SVM has shown promising empirical good performance and has been successfully used in many fields such as bioinformatics, text categorization, speaker verification, handwritten digit recognition, face detection, financial market evaluation [5], pattern recognition [6,7] and image recognition [8,9]. The classification accuracy of SVM is good, but it is slow to classify new examples. There is a fast algorithm to reduce the processing time [10]. Meanwhile, the Decision Tree adopts a divide-and-conquer approach to build a predictive model that estimates the value of a target variable based on several input variables. There are a lot of modified versions of the Decision Tree methods such as Classification and Regression Trees (CART), PART, C4.5 and J4.8 [11]. The Decision Tree algorithms have been successfully applied in a broad range of tasks from medical diagnosis to credit risk assessment for loan application [12]. In addition, KNN, Naïve Bayes, and Neural Networks have also been widely used in various applications as reported in [13-17].

Recently, biology inspired algorithms have been implemented and tried out as a new method for data classification problems. In the previous studies, the stochastic population-based algorithms including Genetic Algorithm [18,19], Ant Colony [20], Immune Algorithm [21], Artificial Bee Colony [22], Particle Swarm Optimization [23,24] were the most commonly used algorithms in the context of optimization [22,25]. The hybrid approach of PSO with simulated annealing and k-means [26], ant colony [27], SVM [28], Neural Networks [29], and the fuzzy set theory with rough set theory [30] are several attempts which have been made to accomplish a classification task in data mining. The experimental results of the above studies show that these methods can outperform the conventional approaches in terms of classification accuracy. Recent studies have shown that PSO is one of the popular heuristic techniques which have emerged as promising technique to discover the useful and interesting knowledge from databases [23]. It has been successfully applied in many different application areas due to its robustness and simplicity [31,32]. However, PSO suffers from premature convergence especially in high dimension multimodal problems. The convergence speed decreases as the number of iteration is increased. These facts lead to the difficulties for the particles to achieve the best fitness values [33]. To improve the performance and overcome the drawback of PSO, this paper proposes a novel Simplified Particle Swarm Optimization (SSO) algorithm with a new Exchange Local Search (ELS) strategy. We shall refer this new algorithm as the SSO-ELS algorithm. In this paper, the ELS strategy is introduced to find a better solution from the neighbourhood of the current solution which is produced by SSO. It allows the particles to better explore the search space, and preserves swarm diversity which is important in preventing premature convergence of the particles. SSO is originally named DPSO (Discrete Particle Swarm Optimization). In order to emphasize its simpleness, we shall refer it as SSO instead of DPSO.

The performance of the proposed algorithm is first measured using 13 popular datasets from UCI repository. Moreover, in order to evaluate the practicability of the new approach, it is used to analyse golf swing from weight shift signal. Weight shift in golf refers to the change of weight between the feet during the swing. In typical golf swing, bodyweight shifts from evenly distributed between the feet at address towards the back foot during backswing, and moves towards the front foot in downswing till follow-through [34,35]. A correct weight shift is an important factor in developing momentum in the golf swing, and is crucial to a shot's range and accuracy [35]. In this study, we applied our SSO-ELS algorithm to classifying weight shift data and identifying patterns of actual golf swing. The information can then be used for correcting and improving swing pose of golfer.

The rest of the paper is organized as follows. Section 2 briefly introduces the previous PSO and SSO algorithm. Section 3 describes in detail about the proposed SSO-ELS algorithm in the context of data mining. Section 4 reports the experimental results of SSO-ELS compared with SSO, PSO, PSO-ELS and three other benchmark classifiers on selected datasets. Finally, conclusions and future works are provided towards the end.

2. **Particle Swarm Optimization and Simplified Swarm Optimization.** The proposed approach is based on the idea of the original PSO [36] and DPSO [37]. This section briefly describes these two algorithms.

2.1. **PSO (particle swarm optimization).** Particle Swarm Optimization has been known as an emerging population-based meta-heuristic algorithm that performs searching using a population (called swarm) of individuals (called particles) that are updated from iteration to iteration [28,36]. In the standard PSO [36], each particle has its own fitness value which is calculated by a fitness function at its current position in order to be optimized [38], and a velocity with social and cognitive components guiding flying towards the optimum [39,40]. PSO starts with initial population of random particles, random positions and velocities which are updated iteration-by-iteration inside the problem space. In each iteration, the particles move around in a multidimensional search space with their velocities constantly updated by the particle's own experience and the best experience of the whole swarm. The former is called the particle's best position (*pbest*) or local best position (*lbest*), while the latter is called the particle's global best position (*gbest*) in the literature [23,36,37].

In PSO, a swarm consists of N particles moving around in a D-dimensional searching space. The i-th particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The best previous position *pbest* of any particle is $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and the velocity for particle i is $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The global best particle which represents the fittest particle found so far in the entire swarm is denoted by P_g. During each iteration, each particle updates its velocity according to the following equation:

$$v_{id}^t = w \cdot v_{id}^{t-1} + c_1 \cdot rand_1 \cdot \left(p_{id} - x_{id}^{t-1}\right) + c_2 \cdot rand_2 \cdot \left(p_{gd} - x_{id}^{t-1}\right), \tag{1}$$

where c_1 and c_2 denote the acceleration coefficients, $d = 1, 2, \ldots, D$, and $rand_1$ and $rand_2$ are random numbers uniformly distributed within $[0, 1]$. Acceleration coefficients c_1 and c_2 control the exploration of the particle movement in a single iteration. Typically, both coefficients are equal to 2.0 in general cases. The inertia weight w in (1) is also used to control the convergence behaviour of the PSO [37]. Each particle then moves to a new potential position as follows:

$$x_{id}^t = x_{id}^{t-1} + v_{id}^t, \qquad d = 1, 2, \ldots, D. \tag{2}$$

The algorithm of the standard PSO is presented as follows:

1. Initialize a population of particles with random positions and velocities.
2. Evaluate the fitness value of each particle in the population.
3. Get the *pbest* value. If the fitness value of the particle i is better than its *pbest* fitness value, and then set the fitness value as a new *pbest* of particle i.
4. Get the *gbest* value. If any *pbest* is updated and it is better than the current *gbest*, and then set *gbest* to the current *pbest* value of particle i.
5. Update particle's velocity and position according to (1) and (2).
6. Stop iteration if the best fitness value or the maximum generation is met;

otherwise go back to step 2.

2.2. **SSO (simplified swarm optimization).** This section introduces SSO algorithm. Initially, the number of swarm population size, the number of maximum generation, and three prespecified parameters are determined. In every generation, the particle's position value in each dimension will be kept or be updated by its *pbest* value or by the *gbest* value or be replaced by new random value according to this procedure.

$$x_{id}^t = \begin{cases} x_{id}^{t-1}, & \text{if rand}() \in [0, C_w), \\ p_{id}^{t-1}, & \text{if rand}() \in [C_w, C_p), \\ g_{id}^{t-1}, & \text{if rand}() \in [C_p, C_g), \\ x, & \text{if rand}() \in [C_g, 1). \end{cases} \tag{3}$$

In this equation, $i = 1, 2, \ldots, m$, where m is the swarm population. $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where x_{iD} is the position value of the i-th particle with respect to the D-th dimension of the feature space. C_w, C_p and C_g are three predetermined positive constants with $C_w < C_p < C_g$. $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ denotes the best solution achieved so far by itself (*pbest*), and the best solution achieved so far by the whole swarm (*gbest*) is represented by $G_i = (g_{i1}, g_{i2}, \ldots, g_{iD})$. The x represents the new value for the particle in every dimension which are randomly generated from random function rand(), where the random number is between 0 and 1. The SSO algorithm is illustrated in Figure 1.

3. **The Proposed SSO-ELS Data Mining Algorithm.** In this paper, we propose a new data mining approach based on the idea of the original PSO [36] and DPSO [37] and call it Simplified Swarm Optimization (SSO) with Exchange Local Search scheme. The SSO-ELS has developed in which each particle is coded as a positive integer number with a new rule encoding scheme and local search strategy. In this paper, the proposed SSO-ELS algorithm is used to solve the classification problem and can cope with dataset containing both discrete and continuous variables.

This approach is significantly different from other previous research works which had only combined data mining and PSO. We found that most of their efforts were dealing with the development of PSO as optimization techniques to solve data mining problems, such as classification algorithm in [41] and clustering algorithm in [42]. To improve the performance of SSO, we proposed to incorporate it with the new local search strategy to perform on the global best solution obtained in each generation. Figure 2 shows the flowchart of how the proposed SSO-ELS algorithm incorporates with the novel local search for data mining classification task.

3.1. **The rule mining encoding.** In the context of the data mining task, knowledge discovery is represented by the form of IF-THEN prediction rules that have the advantage of being high-level symbolic knowledge representation which contribute to the ability to find small number of rules with high fitness value [43]. Figure 3 shows the form of rule mining encoding for the particles' position. The position of each particle contains N

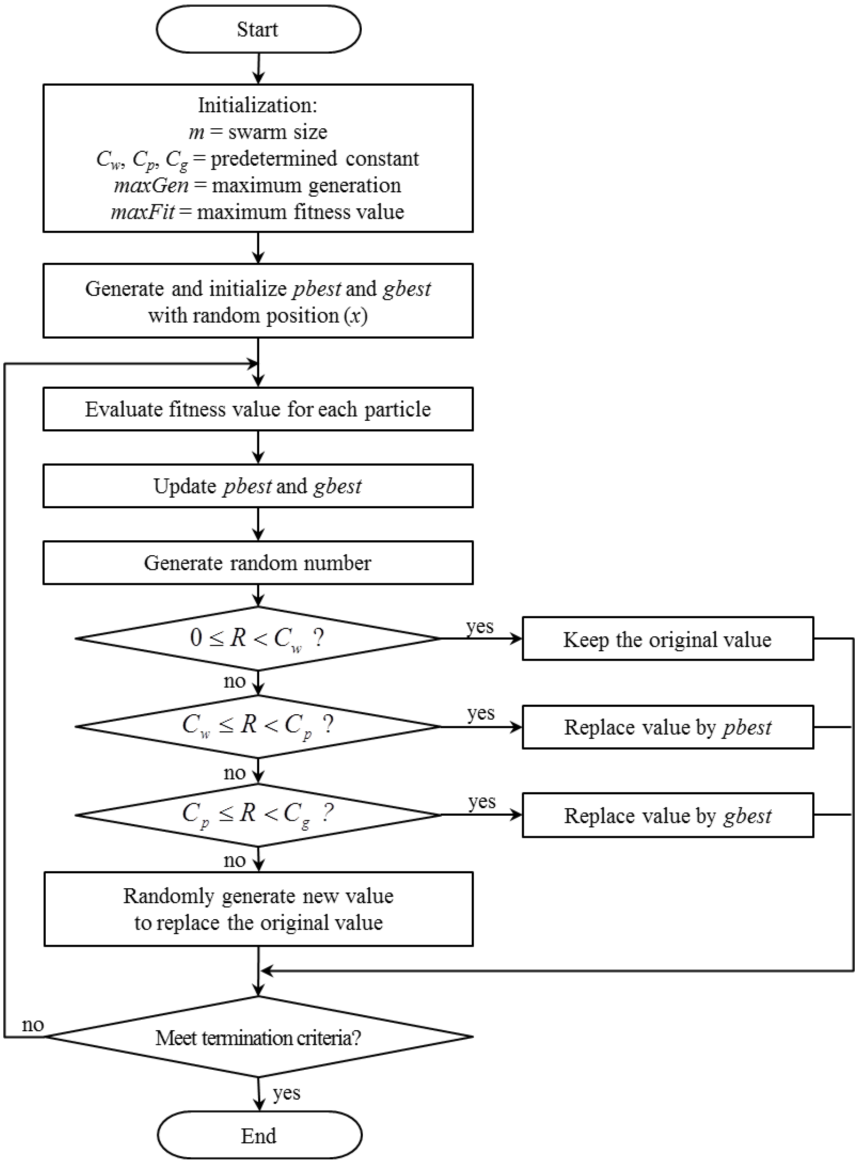

Figure 1. Flowchart of SSO algorithm

dimensions (attributes) and the predictive class, namely Class X. Here we introduce a threshold for each attribute which comes from the lowest data range value and the highest data range value of the given dataset. The former is called *LowerBound* and the latter is called *UpperBound*. The *LowerBound* and *UpperBound* values are obtained by using (4) and (5), respectively.

$$LowerBound = x - \text{rand}() * (\max(X_i) - \min(X_i)), \tag{4}$$

$$UpperBound = x + \text{rand}() * (\max(X_i) - \min(X_i)), \tag{5}$$

where $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ denotes the i-th seed value of the N-th corresponding attribute in each D-dimension, rand() is a random number in a range between 0 and 1, and $(\max(X_i) - \min(X_i))$ is the range value of the data source in each attribute. The general form of the IF-THEN rules generated by PSO and SSO will be performed in all dimensions:

IF $LowerBound \leq x_{ij} \leq UpperBound$ is true, THEN prediction is Class X.

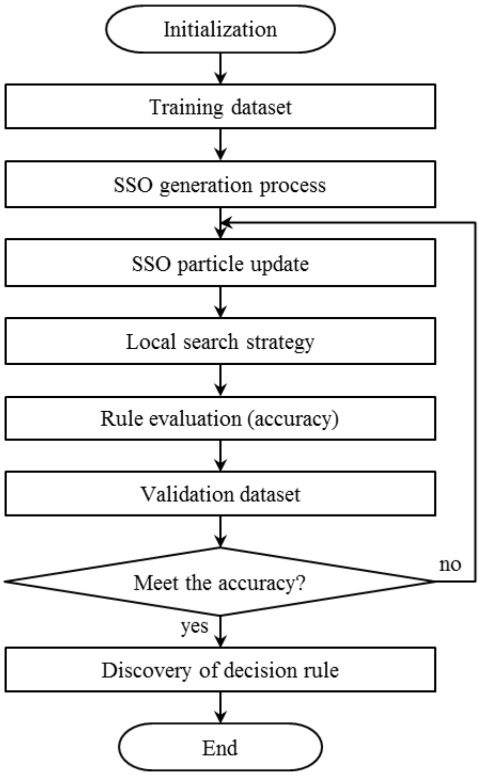

FIGURE 2. The SSO based mining approach with local search strategy

Attribute 1	*Lower Bound*	*Upper Bound*	…	Attribute *N*	*Lower Bound*	*Upper Bound*	Class *X*

FIGURE 3. Rule mining encoding

This approach is likely to produce some seeding position outside the range of the values seen within the dataset. The most likely place a particle will be seeded is around the lowest and the highest values from all of their examples. However, the seeding examples are from the class being predicted by the rule that the particle is encoding. Hence, if the distribution of data from these examples is different from all other examples, then hopefully the search can go in other useful way. In the case of PSO and SSO rule mining, the value of *LowerBound* and *UpperBound* will be updated during and after the generation process. (2) has been employed to update the new current value of the corresponding position.

3.2. **Rule evaluation.** In previous literature, several different evaluation metrics for rule evaluation have been presented and the most commonly used metric is the classification accuracy which has a more direct relation to generalization accuracy. It is necessary to estimate the quality of every candidate rule. To evaluate the goodness of the rules, the rule's quality (fitness function) is computed in the solution space. The rule evaluation method returns a single number that representing the value of the related position. In data mining, typically, data will be divided into two parts, such as training data and testing data. Training data is used to generate a model according to the given rules in the target problem, and later the model will be used on the testing data to obtain the validation accuracy. Usually, we use classification accuracy to measure how well the rule can perform in the testing phase. In general, the standard classification accuracy rate can be written as:

$$\text{The standard classification accuracy rate} = \frac{TP + TN}{TP + FP + FN + TN}, \tag{6}$$

where TP, FP, TN and FN are the number of true positive, false positive, true negative and false negative associated with the rule respectively [11].

- True Positive (TP): the number of examples that covered by the rule that have the class predicted by the rule.
- False Positive (FP): the number of examples covered by the rule that have a class different from the class predicted by the rule.
- True Negative (TN): the number of examples that are not covered by the rule that have a class different from the class predicted by the rule.
- False Negative (FN): the number of examples that are not covered by the rule that have the class predicted by the rule.

On the other hand, the class distribution is highly unbalanced in most nonlinear classification problems. Therefore, (6) is ineffective to measure the accuracy rate of the model [44]. For that reason, a more comprehensive metric for rule evaluation is adopted in this paper and its quality is expressed as follows:

$$\text{The rule quality} = \text{sensitivity} \times \text{specifigity} = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP}. \tag{7}$$

According to [33], the highest fitness value of the individual within the range will be searched in every optimization process which is significantly important to obtain the best quality of the rule to solve the classification problems.

3.3. **Rule pruning.** In data mining, the main goal of rule pruning is to eliminate the irrelevant attributes that might have been unnecessarily included in the rule. Rule pruning can potentially increase the predictive power of the rule, and also can avoid overfitting to the training set [11]. In addition, it also contributes to the minimalism of the rule's length as the shorter rule can be easier to understand by the user [20]. In the rule discovery process, once we found the highest quality rule for the main class in the training set, the best rule is then added to the rule set after being pruned using a pruning procedure.

The principal idea of pruning process is to iteratively remove each attribute (term) at one time from the rule, and at the same time keep improving the quality of the rule. Typically, the pruning process will start with the full rule in the initial iteration. Then the rule quality is computed according to (7). After that, the attribute pairs are checked in reverse order in which they were selected to see if a pair can be removed without decreasing the rule quality. This involves tentatively removing terms from each rule and seeing if each terms' removal affects the accuracy of the entire rule set. If the individual terms' removal does not affect the accuracy, it will be removed permanently. If it does affect the accuracy, it will be replaced and the algorithm will move to the next term, and eventually to the next rule.

After the pruning procedure, the examples which are covered by the rule are removed from the training set. An example is said to be covered by the rule if that example satisfies all the terms (attribute value pairs) in the rule antecedent (IF part). A WHILE loop is performed as long as the number of uncovered examples of the main class in the training set is greater than zero. Once this threshold has been reached, the training set will be reset by adding the previously covered examples. After completing the pruning process, all the redundant rules which do not contribute to the classification accuracy are removed. This is achieved by classifying the training set using the rule list. If any rules do not classify any examples correctly, they will be removed. Later, a series of testing data are used to measure its classification accuracy according to the rule set obtained. For each instance, a prediction value is computed by examining every element in the rule set for the corresponding class if it is covered by the rule. The prediction value is calculated according to the prediction function as in (8).

$$\text{Prediction value}{+}{=}\alpha * \text{rule quality} + \beta * \text{percentage of the rule covered}, \tag{8}$$

where α and β are two parameters corresponding to the importance of the rule quality and the percentage of the rule covered, respectively. The former is known as Quality Weight and the latter is known as Coverage Weight, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$. The prediction value for each class is accumulated and the final result is calculated from the class with the highest prediction value.

3.4. **The proposed exchange local search strategy.** Apparently, SSO can only conduct a rough search that produces premature results which may not be able to offer the satisfactory solutions. For this reason, we have embedded a local search strategy to SSO for producing more satisfactory solutions. Local search is an algorithm that moves from one solution to another solution that allows us to explore the solution space until an optimal solution is found [45]. It starts from a current solution and then tries to improve the searching result by looking for a better solution from its neighbour solution. The neighbourhood search is repeated until a local optimal solution is found.

In this paper, we propose a novel exchange local search (ELS) method to incorporate with the SSO algorithm, namely SSO-ELS. The aim of ELS is to find a new *pbest* of the particle or a new *gbest* from the current particles themselves without doing any new generation. The ELS scheme can also be applied to the original PSO algorithm. We have considered PSO-ELS as well as SSO-ELS.

Figure 4 shows the process of the ELS in flow diagram. The principle of the ELS applied in SSO and PSO rule mining is to exchange the lower bound and upper bound value for one selected attribute from the neighbour particle, re-evaluate the fitness value of the target particle, and then try to find out the new *pbest* of the target particle or new *gbest* in the swarm. This idea is supported by an improvement in performance observed in our initial experiments.

Figure 5 has illustrated the performance of ELS in each generation. By assuming that the rule set of the corresponding dataset contains four attributes, the following eight steps present the sequence involved in ELS algorithm. Figure 5 has illustrated Step 2 to Step 6 in detail.

The steps of ELS strategy:

Step 1 Pre-determine local search time (T) which will only be used for *gbest*.

Step 2 Choose a target particle (P_t). In this phase, *gbest* will be the first target particle to be run in T times' of local search. Later, the other *pbest* will be sequentially selected as target particles and they will only be run once in local search.

Step 3 Randomly select one attribute from the rule set in the dataset. This process is called *exchangeAttribute*.

Step 4 Randomly select two different neighbour particles, P_x and P_y from the population.

Step 5 Get a $LowerBound(x)$ of the selected attribute (selected from Step 3) of P_x, and get an $UpperBound(y)$ of the selected attribute of P_y.

Step 6 Temporarily replace the corresponding *LowerBound* and *UpperBound* of the target particle with x and y.

Step 7 Re-evaluate the fitness value of the target particle.

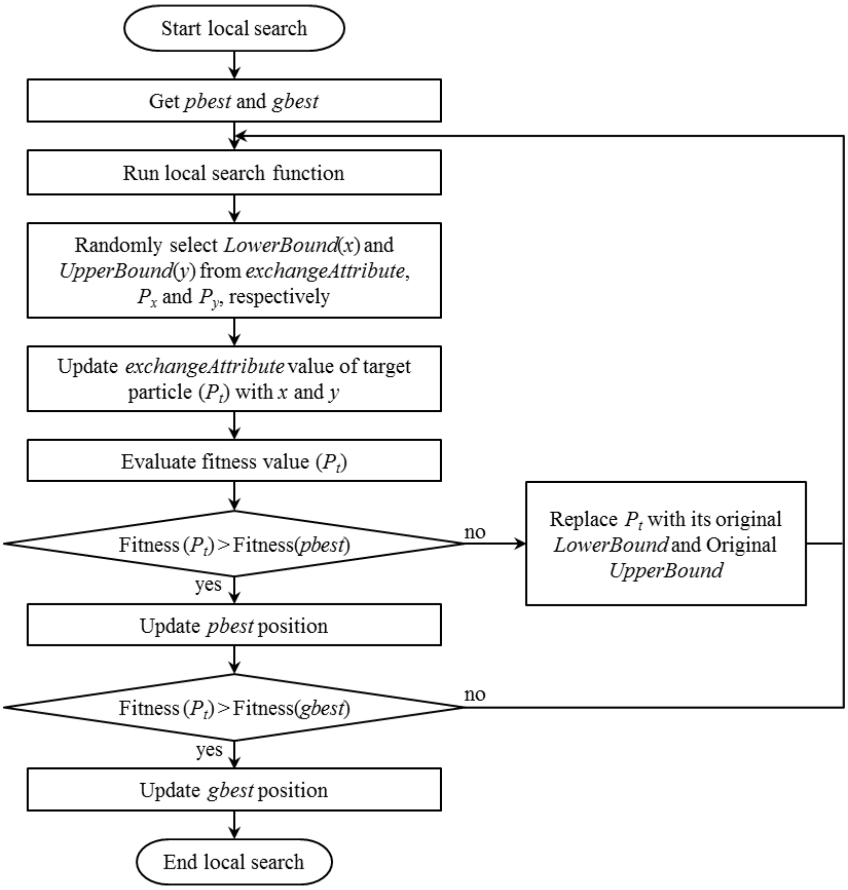

FIGURE 4. The process of the exchange local search (ELS)

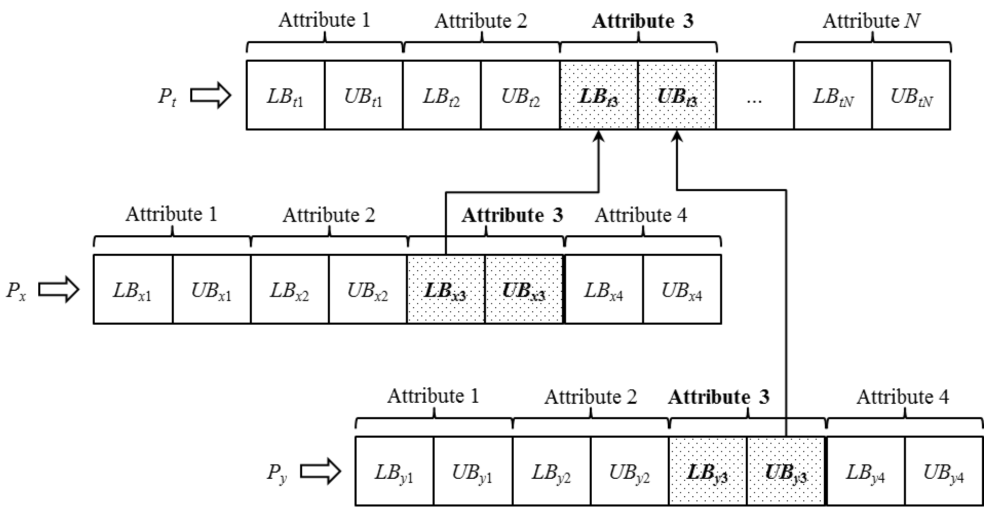

FIGURE 5. The *UpperBound* and *LowerBound* exchange strategy for one selected attribute in each generation (note: $UB = UpperBound$, $LB = LowerBound$)

Step 8 Check whether the fitness value is better than the current *pbest* of the target particle or better than *gbest*. If it is better, *pbest* and *gbest* will be updated, and the exchange value for the target particle will be kept; otherwise, the original *LowerBound* and *UpperBound* values will be positioned back to the target particle.

This process will be repeated until all particles have completed their local search strategy.

4. **Experimental Results.** We have compared our scheme with PSO and three other well-known data mining classification algorithms which include Support Vector Machine (SVM), and two Decision Tree algorithms such as PART and J48.

All benchmark classifiers have been implemented from Weka 3.6.1 [11]. The classification performance of all classifiers was evaluated by using 10-fold cross validation procedure. The dataset was divided into 10 partitions where each procedure is run ten times by using a different partition of testing data set each time, while the other nine data sets will be used for training set. The average of the classification accuracies of the ten runs are reported as the classification accuracy of the discovered rule set. In order to verify the performance of the proposed SSO-ELS method, we have conducted the experiments in two parts. First, we tested the algorithms on chosen UCI datasets. The description of test data and set up are described in Section 4.1, and results are discussed in Section 4.2. In second part, weight shift data of golf swing is used for evaluation, and the details are explained in Section 4.3, followed by discussion of the results in Section 4.4.

4.1. **Dataset description and testing conditions for experiments on UCI data.** In the first part our testing experiments, we have selected 13 datasets taken from UCI repository database [46] to test the performance of the proposed SSO-ELS. These datasets contain various types of attributes that are discrete, continuous and nominal. The nominal attributes in the Credit dataset were mapped into discrete values to suit the SSO or SSO-ELS program.

As mentioned earlier in Section 3, SSO can cope with discrete and continuous variables; hence, data discretization is completely discarded in the pre-processing phase. Details of the properties of 13 datasets used in the experiments have been summarized in Table 1. Some of these datasets contain missing values such as Breast Cancer, Lung, Credit and Heart-Cleveland (from now on called Heart) dataset. The number of missing values of the corresponding datasets is shown in brackets in the last column. In this experiment,

Table 1. Details about the 13 UCI repository datasets used in the experiments

Dataset	Attributes	Instances	Classes	Data type	Missing value
Breast Cancer	9	683	2	Discrete	Yes (16)
Lung	56	27	3	Discrete	Yes (5)
Iris	4	150	3	Continuous	No
Zoo	16	101	7	Discrete	No
Monk	6	432	2	Discrete	No
Thyroid	5	215	3	Discrete, Continuous	No
Credit	15	653	2	Discrete, Continuous	Yes (37)
Glass	9	214	6	Continuous	No
Wine	13	178	3	Continuous	No
Ecoli	7	336	8	Continuous	No
Balance	4	625	3	Continuous	No
Dermatology	34	358	6	Discrete	Yes (8)
Heart	13	297	6	Discrete, Continuous	Yes (6)

TABLE 2. Parameters setting for SSO and PSO algorithms

Parameter Setting	SSO/SSO-ELS	PSO/PSO-ELS
Number of Particles	30	30
Maximum Generation	20	20
Maximum Fitness	1.0	1.0
C_w, C_p, C_g	0.1, 0.4, 0.9	–
Quality Weight (α)	0.5	–
Coverage Weight (β)	0.5	–
c_1, c_2	–	2.0, 2.0
Maximum Weight	–	0.9
Minimum Weight	–	0.4

those missing values were discarded from our consideration as their deleted numbers are relatively small [47].

Meanwhile, Table 2 shows the parameter settings for the algorithm of SSO, SSO-ELS, PSO and PSO-ELS which have been used in all datasets for the experiments. The number of particles and maximum generation were chosen based on the best results obtained from trails. It is worth mentioning that the best setting of the parameters is case dependent and requires further study, and the rest of the parameter values used in this experiment were adopted from [48]. It is assumed based on our preliminary testing that this setup provides a good chance of finding the global optimal solution and ensures convergence of particles in a satisfactory amount of time.

4.2. **Results and discussion on UCI data experiments.** The classification accuracies and the average rule set size results are presented in Table 3. The CA columns list the highest classification accuracies from each run while the ARS columns show the shortest average rule set size generated from the highest classification accuracy of the particular problems. The rankings of the techniques in each problem are also given in the parenthesis. In Table 3, we can see that SSO-ELS could achieve 7 highest score from all datasets when compared with SSO, PSO and PSO-ELS. According to Table 3, the classification accuracy of SSO-ELS can outperform the standard PSO and PSO-ELS in 5 datasets (Breast Cancer, Lung, Monk, Wine and Ecoli) and competitive with PSO in Iris and PSO-ELS in Iris and Zoo datasets. According to the testing results, the SSO-ELS algorithm manages to achieve higher than 94% of classification accuracy in Breast Cancer, Iris, Monk, Zoo and Wine datasets. Furthermore, for Lung dataset the accuracy for SSO-ELS is higher than SSO and PSO by about 6.1% and 15% respectively, while being only slightly better than PSO-ELS by 1.6%.

SSO-ELS also shows good performance in wine dataset where its classification accuracies are 3.5%, 5.5% and 6.6% higher than SSO, PSO and PSO-ELS respectively. Interestingly, after applying ELS in PSO, the classification accuracies for Thyroid and Dermatology datasets have increased to 94.8% and 86.0% respectively. We believe that this is due to the ability of ELS which can perform a refine searching to find the best solution within the problem space. On the other hand, when we compared the average rule set size, PSO and PSO-ELS tend to produce a minimal rule set size with 9.1 respectively. However, their classification performances are still not as good as SSO-ELS even though the average rule set size of SSO-ELS is greater than PSO and PSO-ELS. Overall, SSO-ELS (84.8%) could outperform SSO, PSO and PSO-ELS with 84.0%, 83.0% and 84.1% respectively over the 13 problems. Among swarm-based optimization algorithms involved

Table 3. Summarization of classification accuracies (%) and average rule set size for SSO and PSO with and without exchange local search

	SSO		PSO		SSO-ELS		PSO-ELS	
Dataset	CA	ARS	CA	ARS	CA	ARS	CA	ARS
Breast Cancer	97.1(3)	6.4(4)	96.8(4)	**6.1(1)**	**97.4(1)**	6.2(2)	97.2(2)	6.3(3)
Lung	68.3(3)	9.5(4)	59.4(4)	**7.9(1)**	**74.4(1)**	8.3(3)	72.8(2)	8.1(2)
Iris	95.3(2)	4.7(2)	**96.0(1)**	4.8(3)	**96.0(1)**	**4.5(1)**	**96.0(1)**	4.8(3)
Zoo	90.2(3)	12.6(3)	93.1(2)	**7.4(1)**	**94.1(1)**	9.9(2)	**94.1(1)**	**7.4(1)**
Monk	**100.0(1)**	4.0(2)	99.3(3)	**3.9(1)**	**100.0(1)**	4.0(2)	99.5(2)	4.1(3)
Thyroid	94.1(3)	6.3(3)	94.4(2)	**5.1(1)**	93.5(4)	6.3(3)	**94.8(1)**	5.5(2)
Credit	**85.8(1)**	11.5(3)	85.2(4)	**9.0(1)**	85.7(2)	14.1(4)	85.3(3)	9.1(2)
Glass	**65.8(1)**	14.3(4)	57.9(4)	**6.2(1)**	62.3(2)	6.4(3)	60.8(3)	6.3(2)
Wine	94.1(2)	**6.8(1)**	92.1(3)	14.8(3)	**97.6(1)**	14.1(2)	91.0(4)	14.1(2)
Ecoli	82.2(2)	14.3(3)	81.8(3)	**12.2(1)**	**83.1(1)**	12.7(3)	80.5(4)	**12.2(1)**
Balance	79.7(2)	12.8(4)	**80.4(1)**	11.0(2)	78.6(3)	11.1(3)	78.1(4)	**10.9(1)**
Dermatology	83.0(4)	**11.5(1)**	85.6(2)	15.1(2)	83.6(3)	27.1(3)	**86.0(1)**	14.8(2)
Heart	56.2(3)	31.6(4)	**56.9(1)**	14.9(3)	56.4(2)	**11.3(1)**	**56.9(1)**	14.8(2)
Average	84.0	11.3	83.0	9.1	84.8	10.5	84.1	9.1
Rank	3	3	4	1	1	2	2	1

Table 4. Comparison of classification accuracies (%) and ranking of the techniques for SSO-ELS and other classifiers

Data	SSO	SSO-ELS	PSO	PSO-ELS	PART	SVM	J48
Breast Cancer	97.1(3)	97.4(1)	96.8(4)	97.2(2)	95.5(7)	96.3(5)	96.1(6)
Lung	68.3(3)	74.4(1)	59.4(4)	72.8(2)	48.2(5)	40.7(6)	48.2(5)
Iris	95.3(2)	96.0(1)	96.0(1)	96.0(1)	94.0(3)	92.7(4)	96.0(1)
Zoo	90.2(4)	94.1(1)	93.1(2)	94.1(1)	92.1(3)	73.3(5)	92.1(3)
Monk	100.0(1)	100.0(1)	99.3(3)	99.5(2)	100.0(1)	81.7(4)	100.0(1)
Thyroid	94.1(3)	93.5(5)	94.4(2)	94.8(1)	94.0(4)	69.8(7)	92.0(6)
Credit	85.3(3)	85.7(2)	85.2(4)	85.3(3)	83.8(6)	86.4(1)	84.8(5)
Glass	65.8(3)	62.3(4)	57.9(6)	60.8(5)	67.8(1)	35.5(7)	65.9(2)
Wine	94.1(2)	97.6(1)	92.1(5)	91.0(6)	93.3(4)	41.6(7)	93.8(3)
Ecoli	82.2(4)	83.1(3)	81.8(5)	80.5(6)	83.6(2)	42.6(7)	84.2(1)
Balance	79.7(4)	78.6(5)	80.4(3)	78.1(6)	83.5(2)	88.3(1)	76.6(7)
Dermatology	83.0(6)	83.6(5)	85.6(4)	86.0(3)	93.3(2)	81.6(7)	95.3(1)
Heart	56.2(3)	56.4(2)	56.9(1)	56.9(1)	50.2(6)	53.9(4)	52.2(5)
Average	83.9(3)	84.8(1)	83.0(4)	84.1(2)	83.0(4)	68.0(6)	82.9(5)

in this study, SSO-ELS shows the best performance, while PSO shows the lowest results, as shown in Table 3.

In order to validate the competitiveness of SSO-ELS in various datasets, we have compared its classification accuracy performance with three other benchmark classifiers. All three benchmark classifiers were employed with their default parameters as set in WEKA. In this paper, sequential minimal optimization (SMO) [49] algorithm was implemented for training an SVM classifier.

The comparison results are presented in Table 4 which can clearly show that SSO-ELS can achieve higher classification accuracy for Breast Cancer, Lung, and Wine datasets when compared with three other benchmark classifiers. Moreover, the SSO-ELS algorithm is found competitive with J48 in Lung dataset as well as competitive with PART and J48

Table 5. Average classification accuracies and ranking of all techniques on 13 testing datasets

Data	SSO	SSO-ELS	PSO	PSO-ELS	PART	SVM	J48
Average (%)	83.9	84.8	83.0	84.1	83.0	68.0	82.9
Rank	3	1	4	2	4	6	5

Table 6. Sum of ranking of the techniques and ranking based on the total ranking

Data	SSO-ELS	PSO-ELS	SSO	PSO	J48	PART	SVM
Total Score	32	39	41	44	46	46	65
Rank	1	2	3	4	5	5	6

in Monk dataset. In addition, for Lung dataset SSO-ELS shows about 1.4%~33.7% better against 6 other benchmark classifiers which can be considered as a significant contribution in data mining problem. Besides, about 56% of difference in accuracy can be seen in Wine dataset when compared with SVM.

In order to make a good comparison of these 7 algorithms, Table 5 and Table 6 are reported. The former one presents the average classification accuracies of all datasets and the ranking based on the average values, and the latter one is the sum of the algorithm's ranking of each problem which have arranged from minimum value to maximum value. From Table 5, we can note that the best two techniques on this databases set are SSO-ELS and PSO-ELS, followed by SSO. Then, a group of three techniques (PSO, PART and J48) follows at some distance. Finally, SVM follows at more distant.

Since the difference of average classification accuracy in these 7 techniques is quite slim, Table 6 has calculated the sum of the ranks of each dataset from Table 4. From this ranking, SSO-ELS can be seen as the best approach among swarm intelligence algorithms and among all 7 data mining techniques. Therefore, we can conclude that SSO-ELS is the most effective technique in facing classification problem and also can compete with other most popular data mining techniques. According to the good performance of SSO-ELS, we can conclude that the proposed swarm-intelligence data mining algorithm can be used to solve classification problems.

4.3. **Dataset description and testing conditions for experiments on golf swing data.** In this study, we used data collected from our golf swing experiment. The dataset contains 516 instances of weight shift patterns, of which 150 of them represent golf swings and the rest were not actual swing motions. The attributes are shown in Table 7. The first three attributes are measures of time between various points of swing motion. These points are time when weight starts to transfer from one foot to another. *PeakRatio* is the maximum weight per second of the front foot in the second run, and the next three features are ratios of weights between the two feet. Lastly, *PeakLoc* is the ratio of length from first peak to cross point over length of *SecondRun*. The parameters set up in this experiment were the same as part 1.

4.4. **Results and discussion on golf experiments.** The classification accuracies of SSO-ELS and rest of the algorithms are listed in Table 8. SSO-ELS again achieves the best result with the accuracy rate of 98.8%, and outruns SVM and PART. In the meantime, the accuracy of PSO-ELS is also 0.7% higher than then standard PSO. The high accuracy obtained shows that our proposed algorithm is suitable for classifying real world data and can be applied in practical systems.

TABLE 7. Details about the golf swing weight shift data used in the experiments

Attribute	Description
PrevRun	time of a range from cross point before assumed address point to address point
FirstRun	time of a range from assumed address point to assumed impact point
SecondRun	time of a range from assumed impact point to next cross point
PeakRatio	ratio of width to height of the first peak point in the SecondRun
PrevRatio	ratio of the minimum value of the right weight to the maximum value of the left weight
FirstRatio	ratio of the minimum value of the left weight to the maximum value of the right weight
SecondRatio	ratio of the minimum value of the right weight to the maximum value of the left weight in the SecondRun
PeakLoc	ratio of PeakLoctime to the length of SecondRun

TABLE 8. Average classification accuracies and ranking of all techniques on golf swing weight shift data

Data	SSO	SSO-ELS	PSO	PSO-ELS	PART	SVM	J48
Accuracy (%)	98.1	98.8	97.5	98.2	98.3	98.4	97.7
Rank	5	1	7	4	3	2	6

5. **Conclusions.** In this paper, we have presented a novel way of incorporating local search strategy into SSO and PSO algorithm for rule mining classification algorithm. A local search strategy named ELS (Exchange Local Search) is proposed and the results have been analysed and discussed in detail. The main idea of exchange local search is to improve and refine the searching process by exchanging the *LowerBound* and *UpperBound* for one selected attribute into the *LowerBound* and *UpperBound* of the corresponding target particle. The fitness evaluation is performed to find out the new *pbest* or *gbest* from the target particle without doing any new generation. The performance of SSO-ELS has been compared with SSO, PSO, PSO-ELS and other three most popular data mining techniques on 13 UCI repository datasets and weight shift data from golf swing. Our experimental results show that the proposed method has comparable results for both SSO-ELS and PSO-ELS. The technique proposed in this paper managed to get the highest classification accuracy with more than 94% (except for Lung with 74.4%) and can outperform the other three benchmark classifiers in 13 datasets. It also achieved the best result in classifying golf swing from weight shift data and demonstrated it is suitable for practical systems. In the future, we plan to investigate the applicability of the proposed SSO algorithm with exchange local strategy to various domains of problems.

Acknowledgment. This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST) (NRF-M1AXA003-2010-0029793).

REFERENCES

[1] P. N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Addison Wesly, 2006.

[2] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, The KDD process for extracting useful knowledge from volumes of data, *Communication of the ACM*, vol.39, no.11, pp.27-34, 1996.
[3] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.
[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
[5] S. T. Li, W. Shiue and M. H. Huang, The evaluation of consumer loans using support vector machines, *Expert Systems with Applications*, vol.30, no.4, pp.772-782, 2006.
[6] Y. C. Wu, Y. S. Lee and J. C. Yang, Robust and efficient multiclass SVM models for phrase pattern recognition, *Pattern Recognition Letters*, vol.41, no.9, pp.2874-2889, 2008.
[7] A. Wang, W. Yuan, J. Liu, Z. Yu and H. Li, A novel pattern recognition algorithm: Combining ART network with SVM to reconstruct a multi-class classifier, *Computers & Mathematics with Applications*, vol.57, no.11-12, pp.1908-1914, 2009.
[8] H. A. Park and K. R. Park, Iris recognition based on score level fusion by using SVM, *Pattern Recognition Letters*, vol.28, no.15, pp.2019-2028, 2007.
[9] T. Youwen, L. Tianlai and N. Yan, The recognition of cucumber disease based on image processing and support vector machine, *Proc. of Congress on Image and Signal Processing*, Sanya, China, pp.262-267, 2008.
[10] S. Yoon, C. Lyuh, I. Chun, J. Suk and T. M. Roh, A new support vector compression method based on singular value decomposition, *ETRI Journal*, vol.33, no.4, pp.652-655, 2011.
[11] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, San Fransisco, 2005.
[12] C.-H. Cheng and Y.-S. Chen, A fuzzy-based rough sets classifier for forecasting quarterly PGR in the stock market (part II), *International Journal of Innovative Computing, Information and Control*, vol.7, no.3, pp.1209-1228, 2011.
[13] M. M. Mostafa and N. Billor, Recognition of western style musical genres using machine learning techniques, *Expert Systems with Applications*, vol.36, no.8, pp.11378-11389, 2009.
[14] Y. Li and L. Guo, An active learning based TCM-KNN algorithm for supervised network intrusion detection, *Computers and Security*, vol.26, no.7-8, pp.459-467, 2007.
[15] Y. Liao and V. R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, *Computers and Security*, vol.21, no.5, pp.439-448, 2002.
[16] T. Pao, Y. Chen and J. Yeh, Emotion recognition and evaluation from mandarin speech signals, *International Journal of Innovative Computing, Information and Control*, vol.4, no.7, pp.1695-1709, 2009.
[17] O. Addin, S. M. Sapuan, E. Mahdi and M. Othman, A Naïve-bayes classifier for damage detection in engineering materials, *Materials & Design*, vol.28, no.8, pp.2379-2386, 2007.
[18] M. H. Tseng and H. C. Liao, The genetic algorithm for breast tumor diagnosis – The case of DNA viruses, *Applied Soft Computing*, vol.9, no.2, pp.703-710, 2009.
[19] T. C. Chen and T. C. Hsu, A GAs based approach for mining breast cancer pattern, *Expert System with Applications*, vol.30, no.4, pp.674-681, 2006.
[20] R. S. Parpinelli, H. S. Lopes and A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transactions on Evolutionary Computation*, vol.6, no.4, pp.321-332, 2002.
[21] R. Z. Wang, Q. Zhang and D. Zhang, An IA-based classification rule mining algorithm, *Proc. of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, China, pp.309-313, 2007.
[22] P.-W. Tsai, J.-S. Pan, B.-Y. Liao and S.-C. Chu, Enhanced artificial bee colony optimization, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.5081-5092, 2009.
[23] S. H. Zahiri and S. A. Seyedin, Swarm intelligence based classifiers, *Journal of the Franklin Institute*, vol.344, no.5, pp.362-376, 2007.
[24] T. Sousa, A. Silva and A. Neves, Particle swarm based data mining algorithms for classification tasks, *Parallel Computing*, vol.30, no.5-6, pp.767-783, 2004.
[25] D. Karaboga and B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, vol.214, no.1, pp.108-132, 2009.
[26] B. B. Firouzi, M. S. Sadeghi and T. Niknam, A new hybrid algorithm based on PSO, SA, and K-means for cluster analysis, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.3177-3192, 2010.

[27] Y. Marinakis, M. Marinaki, M. Doumpos and C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, *Expert System with Applications*, vol.36, no.7, pp.10604-10611, 2009.
[28] S. W. Lin, K. C. Ying, S. C. Chen and Z. J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, *Expert System with Applications*, vol.35, no.4, pp.1817-1824, 2008.
[29] Z. Chen, B. Yang, Y. Chen, L. Wang, H. Wang, A. Abraham and C. Grosan, Improving neural nNetwork classification using further division of recognition space, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.301-310, 2009.
[30] A. Hassanien, Fuzzy rough sets hybrid scheme for breast cancer detection, *Image and Vision Computing*, vol.25, no.2, pp.172-183, 2007.
[31] F. Du, W. Shi, L. Chen, Y. Deng and Z. Zhu, Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimization (PSO), *Pattern Recognition Letters*, vol.26, no.5, pp.597-603, 2005.
[32] Z. Lian, X. Gu and B. Jiao, A similar particle swarm ooptimization algorithm for permutation flowshop scheduling to minimize makespan, *Applied Mathematics and Computation*, vol.175, no.1, pp.773-785, 2006.
[33] H. Liu, A. Abraham and W. Zhang, A fuzzy adaptive turbulent particle swarm optimisation, *International Journal of Innovative Computing and Applications*, vol.1, no.1, pp.39-47, 2007.
[34] K. A. Ball and R. J. Best, Different centre of pressure patterns within the golf stroke I: Cluster analysis, *Journal of Sports Sciences*, vol.25, pp.757-770, 2007.
[35] I. H. Witten and E. Frank, *Weight Transfer Styles in the Golf Swing: Individual and Group Analysis*, Ph.D. Thesis, Victoria University, Australia, 2006.
[36] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of the 1995 IEEE International Conference on Neural Networks. Part 4*, Perth, Australia, pp.1942-1948, 1995.
[37] R. Eberhart and Y. Shi, Particle swarm optimization: Developments, applications and resources, *Proc. of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, pp.81-86, 2001.
[38] R. Poli, J. Kennedy and T. Blackwell, Particle swarm optimization an overview, *Swarm Intelligence*, vol.1, pp.33-57, 2007.
[39] A. Allahverdi and F. S. Al-Anzi, A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application, *Computer & Operations Research*, vol.33, no.4, pp.1056-1080, 2006.
[40] X. Cai, Z. Cui, J. Zeng and Y. Tan, Particle swarm optimization with self-adjusting cognitive selection strategy, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.943-952, 2008.
[41] F. Melgani and Y. Bazi, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, *IEEE Transactions on Information Technology in Biomedicine*, vol.12, no.5, pp.667-677, 2008.
[42] F. Yang, T. Sun and C. Zhang, An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization, *Expert Systems with Applications*, vol.36, no.6, pp.9847-9852, 2009.
[43] J. H. Ang, K. C. Tan and A. A. Mamun, An evolutionary memetic algorithm for rule extraction, *Expert Systems with Applications*, vol.37, no.2, pp.1302-1315, 2010.
[44] E. S. Correa, A. A. Freitas and C. G. Johnson, A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set, *Proc. of GECCO*, Seattle, Washington, USA, pp.35-42, 2006.
[45] A. Schaerf, Local search technique for large high-school timetabling problems, *IEEE Transactions on Systems, Man, and Cybernatics*, vol.4, pp.368-377, 1999.
[46] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml/, 2007.
[47] A. P. White, Probabilistic induction by dynamic path generation in vertual trees, in *Research and Development in Expert Systems III*, M. A. Bramer (ed.), Cambridge University Press, 1987.
[48] W. C. Yeh, W. W. Chang and Y. Y. Chung, A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method, *Expert Systems with Applications*, vol.36, no.4, pp.8204-8211, 2009.
[49] J. Platt, Fast training of support vector machines using sequential mminimal optimization, in *Advances in Kernel Methods – Support Vector Learning*, B. Schoelkopf, C. Burges and A. Smola (eds.), Cambridge, MA, MIT Press, 1988.