INTRODUCTION

Classification is a supervised learning technique that using labeled data samples generates a model (classifier). The resulted model classifies new data samples into different predefined groups or classes. In other word, in classification problem objects assigned to one of several predefined categories. From the mathematical point of view, classification can be defined as a mapping from the input feature space into a set of labels. In recent years, researchers have developed many classification techniques including intelligent particle swarm (PS)-classifier,[1] binary classifiers,[2-4] decision tree classifiers,[5,6] artificial neural network classifiers,[7-10] Bayesian classifiers,[11] support vector machine classifiers,[12,13] and instance (prototype) based classifiers.[14]

This paper proposes a chaotic particle swarm optimization (PSO) with mutation-based classifier (MCPSO) based approach for classifier design. PSO algorithm is a powerful evolutionary algorithm inspired by the social behavior of bird flocks and fish schools.[15] PSO is one of the most promising optimization algorithms that used for a wide range of complex engineering optimization problems. Algorithmic simplicity and fast convergence of PSO are the most attractive features of this metaheuristic algorithm. However, when PSO applied to strongly multimodal optimization problems, it tends to suffer from premature convergence.[16,17] To overcome the premature convergence and enhances the optimization performance, chaotic PSO with mutation is proposed here. The presence of mutation operator helps to sharpen the convergence and tunes to the best solution.

Since MCPS is a simple and effective search technique in high dimensional spaces, with a little prior information a MCPSO-classifier has the potential of classifying different high dimensional feature spaces classes, successfully. With searches in solution space, the MCPS-classifier moves toward optima hyperplanes in such manner that the misclassified points are minimized.
In general, classification problems involve a number of features. All of these features have not equally important for a specific task. Better performance may be achieved by discarding redundant or irrelevant features. Therefore, using the smallest number of features, the classification process can be fast and accurate. Using feature selection this objective can be achieved. Feature selection strategies used to explore the effect of irrelevant attributes on the performance of classifier systems.\textsuperscript{[16,19]}

The goal of this study is to increase the classification accuracy rate by employing an approach based on proposed algorithm. To do this, we use two types of MCPSO: The continuous-valued version and binary version. The continuous-valued version is used to optimize the best model parameters, while the binary version is used to search the optimal feature subset. The developed MCPSO approach not only tunes the parameter values of model but also identifies a subset of features that maximize the classification accuracy rate.

For comparing experimental results three common benchmark problems in medical database classification were considered. The Wisconsin diagnostic breast cancer (WDBC), Wisconsin breast cancer (WBC) and heart-statlog data classifications are common problems in pattern recognition researches.

The performance of MCPS-classifier has been compared with k-nearest neighbor (k-NN), PS based-classifiers, genetic algorithm based-classifier (GA-classifier) and imperialist competitive algorithm based-classifier (ICA-classifier), to show that the average of recognition rates of designed MCPS-classifier are better than to those of the traditional and new classifiers. Some illustrative figures have been included for comparing convergence speed MCPS-classifier and other mentioned metaheuristic algorithms based-classifier.

In this paper, section two explains standard and improved real-binary PSO algorithm. MCPS-classifier is described in the next section. Section four considers implementation of the classifier and experimental results on three aforesaid pattern recognition problems. Finally, conclusion and discussion is presented in section five.

**PARTICLE SWARM OPTIMIZATION ALGORITHM**

**Standard Real-Binary Particle Swarm Optimization**

Dr. Eberhart and Dr. Kennedy in 1995 developed an evolutionary computation technique, named PSO, which inspired by social behavior of bird flocking or fish schooling.\textsuperscript{[15]} With a population of random solutions, the algorithm is initialized and searches for optima by updating generations. Unlike GA, there are not evolution operators such as crossover and mutation in PSO. A PS can consider as a population of individuals which each individual contain the appropriate amount of features to place it in a swarm problem space. The individuals are arranged in neighborhoods so that they can share information. In PSO, each single solution is called as “particle” and all the particles save fitness values, which these values evaluated by the objective function to be optimized. Particles have velocities, which direct the flying of the particles. With following the current optimum particles, the particles are flown through the problem space. Each particle is updated by two “best” values in each iteration. The first one is the best position (fitness) it has achieved so far which called Pbest. Another one, named Gbest, is the overall best value obtained so far by any particle in the population. After finding the two best values, with Eq. (1) and (2) the particle updates its velocity and position, respectively.

\[
V_{id}^{k+1} = w \cdot V_{id}^{k} + c_1 \cdot rand \cdot (P_{best}^{k} - X_{id}^{k}) + c_2 \cdot rand \cdot (G_{best}^{k} - X_{id}^{k}) \tag{1}
\]

\[
X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k} \tag{2}
\]

Where \(w\) is the inertia weight, \(V_{id}\) is the particle velocity, \(X_{id}\) is the current particle position, rand is a random number between (0, 1) and \(c_1, c_2\) are learning or acceleration factors. The velocities of particles on each dimension are clamped to a maximum velocity \(V_{max}\).

In PSO, the key factors affecting the convergence behavior are: The parameters \(w, c_1\) and \(c_2\).\textsuperscript{[20,21]} The balance between the global exploration and the local search ability control by the inertia weight, in which a large inertia weight favors the global search and a small inertia weight favors the local search. Hence, usually an inertia weight that linearly decreases from 0.9 to 0.4 throughout the search process is used.\textsuperscript{[22]}

In order to extend PSO algorithm to tackle binary problems effectively, Kennedy and Eberhart adapted the continuous PSO algorithm to binary spaces.\textsuperscript{[23]} In binary version of PSO, the position of the particle has two values 0 or 1, and the velocity of the particle represents the probability that a bit (position) takes on 0 or 1. In based on particle swarm optimization (PSO) the Eq. (1) remains unchanged, but the Eq. (2) is redefined by Eq. (3):

\[
X(t + 1) = \begin{cases} 
0 & \text{if } rand() < S(v(t + 1)) \\
1 & \text{if } rand() \geq S(v(t + 1)) 
\end{cases} \tag{3}
\]

where \(S(\cdot)\) is the sigmoid function, which is used to transform the velocity to a probability and defined as follow:

\[
S(v(t + 1)) = \frac{1}{1 + e^{-v(t+1)}} \tag{4}
\]
and \( \text{rand()} \) is a random number selected from the uniform distribution over \((0, 1)\).

**Chaotic Particle Swarm Optimization**

Simulation dynamic behavior of nonlinear systems called chaos. It has raised enormous interest in different fields of sciences such as synchronization, chaos control, optimization theory, pattern recognition and so on.\(^{[24]}\) In optimization algorithms based on the chaos theory, the methods using chaotic variables instead of random variables are called chaotic optimization algorithm (COA). COA is a stochastic search methodology that differs from any of the existing swarm intelligence methods and evolutionary computation. COA can carry out overall searches at higher speeds than stochastic searches that depend on probabilities.\(^{[25]}\)

There are several different chaotic sequences which the most commonly used such chaotic sequences are logistic maps that are considered in this paper. Logistic maps are frequently used chaotic behavior maps and chaotic sequences can be quickly generated and easily stored. For this reason, there is no need for storage of long sequences.\(^{[26]}\) In this study, we substitute the random parameters in PSO with sequences generated by the logistic map. The parameters random are modified by the logistic map based on the following equation:

\[
C_{r(t+1)} = k \times C_{r(t)} \times (1 - C_{r(t)})
\]  

In Eq. (5), \( k = 4 \) and for each independent run, \( C_{r(0)} \) is generated randomly, which \( C_{r(0)} \) not being equal to \( \{0, 0.25, 0.5, 0.75, 1\} \). Behavior of \( C_{r(t)} \) is controlled by the driving parameter \( k \) of the logistic map (as \( t \) goes to infinity).\(^{[27]}\)

Considering to the above descriptions the velocity update equation for chaotic particle swarm optimization can be formulated as:

\[
V_{id}^{k+1} = w \times V_{id}^k + c_1 C_{r}(P_{best}^k - X_{id}^k) \\
+ c_2 (1 - C_{r})(G_{best}^k - X_{id}^k)
\]  

In Eq. (6), \( C_{r} \) is a function based on the results of the logistic map with values between 0.0 and 1.0.

**Proposed Mut Particle Swarm Optimization Algorithm**

Velocity and position updating are the two major operations in standard PSO. These operators use to update the search space repeatedly and may cause difficulties in certain situation that leads to get stuck in the local optima. The proposed PSO incorporates the some mutation operators from GA to overcome this difficulty. Mutation operator with generating new material into the population, thereby allows faster convergence and prevents trapping to a local optimal value.

One of the most widely used mutation operators in real coded GAs is Michalewicz’s nonuniform mutation.\(^{[28]}\) The muted point \( x_{id}^{k+1} = \{x_{i1}^{k+1}, x_{i2}^{k+1}, \ldots, x_{in}^{k+1}\} \) from a point \( x_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\} \) is created as follows:

\[
x_{id}^{k+1} = \begin{cases} 
    x_{id}^k + \Delta \cdot (t, x_{id}^k - x_{id}^0), & \text{if} \ r \leq 0.5 \\
    x_{id}^k - \Delta \cdot (t, 1 - x_{id}^0), & \text{otherwise}
\end{cases}
\]

where \( t \) is current generation number and \( r \) is a random number between 0 and 1 with uniform distribution. \( x_{id}^0 \) and \( x_{id}^k \) show lower and upper bounds of the \( i \)th component of the decision vector, respectively. The function \( \Delta(t, r) \) given below takes value in the interval \((0, 1)\).

\[
\Delta(t, r) = y \left(1 - \left(\frac{t}{T}\right)^b\right)
\]

where \( T \) is the maximum number of generations, \( u \) is a random number in the interval \((0, 1)\) with uniform distribution and \( b \) is a parameter, determining the strength of the mutation operator. In the initial generations with emphasize to exploration nonuniform mutation tends to search the space uniformly and for tuning the solution in the later generations it tends to search the space locally, that is, closer to its descendants.\(^{[29]}\)

Mutation operator just described, provide the ability to “fly” to the new search area and as that in GA information can be changed in the individuals. In other word, the presence mutation operator makes the proposed MutPSO more exploitative search mechanism than standard PSO and consequently MutPSO can finds better optima more consistently.

In BPSO to explore untried areas of the search space we used the following mutation operator as suggested in:

\[
\begin{cases} 
    x_{id}^{k+1} = x_{id}^k, & \text{if} \ r \leq r_{\text{mut}}, i = 1 \ldots N, j = 1 \ldots N_t \\
    x_{id}^{k+1} = x_{id}^k + \Delta \cdot (t, x_{id}^k - x_{id}^0), & \text{otherwise}
\end{cases}
\]

where \( r_{\text{mut}} \) is the probability of random mutation, \( N \) is the total number of particles and \( N_t \) is the total (initial) number of features of the dataset. After updating the particle position as in (1) and (3), each of the bits of the position vector is mutated with a probability \( r_{\text{mut}} \).

**CHAOTIC PARTICLE SWARM OPTIMIZATION WITH MUTATION BASED CLASSIFIER**

In this paper, the mentioned mutation operator and chaotic sequences considered simultaneously to improving performance of PSO. A chaotic PSO with mutation based-classifier (MCPS-classifier) has three major parts including decision hyperplanes, fitness function definition, and its structure.
Decision Hyperplanes

A general hyperplane is in the following form:
\[ d(X) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + w_{n+1} \] (10)
where \( W = (w_1, w_2, \ldots, w_n, w_{n+1}) \) and \( X = (x_1, x_2, \ldots, x_n, 1) \) are called weight vector and the augmented feature respectively and \( n \) is the feature space dimension.

The MCPS-classifier must find \( W_j, (j = 1, 2, \ldots, H) \) in solution space in such manner that the misclassified points are minimized, where \( H \) is the necessary number of decision hyperplanes.

Fitness Function Definition

This study developed an improved PSO approach for parameter determination and feature selection in an evolutionary classifier. For each hyperplane, \( n + 1 \) decision variables are required. For feature selection, \( n \) decision variables must be adopted. The feature selection is Boolean that “1” represents the feature is selected, and “0” indicates feature is not selected.

In this study, classification accuracy and the number of selected features are two measures used to design a fitness function. We defined fitness function for an individual such that, a high fitness value achieved with high classification accuracy and small number of features. Thus, fitness function is defined as follow:
\[
\text{Fitness}_i = \text{Accuracy}_i + w_F \times \left[ 1 - \frac{\left( \sum_{j=1}^{n_f} f_j \right)}{n_f} \right] 
\] (11)
where, \( w_f \) is the weight for the number of selected features (0<\( w_f <1 \)), \( f_j \) is the value of feature mask “1” represents that feature \( j \) is selected and “0” represents that feature \( j \) is not selected, and \( n_f \) is the total number of features. The classification accuracy is defined as follow:
\[
\text{Accuracy} = \frac{\text{Number of correctly classified objects}}{\text{Number of objects in datasets}} \times 100\% 
\] (12)

To fully characterize the classifier performance, additional information from the confusion matrix is considered too. This information is necessary in the classification of data with imbalanced class distribution, where even a total error in predicting a rare class, would have only a small impact on the total accuracy%. Therefore, following measures is also considered:[28]

\[
\text{Specificity} = \frac{TN}{TN + FP} 
\] (13)
\[
\text{Sensitivity} = \frac{TP}{TP + FN} 
\] (14)

Matthews’s correlation coefficient (MCC) =
\[
\frac{TP \times TN - P \times FN}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FN) \times (TN + FP)}} 
\] (15)
where TP is the number of the true positives, TN is the number of true negatives, FP is the number of the false positives and FN is the number of false negatives. Sensitivity measures the proportion of actual positives which are correctly identified. Specificity measures the proportion of negatives, which are correctly identified. MCC is the MCC,[28] which reflects both the sensitivity and specificity of the prediction algorithm.

The Structure of Mutation Based-Classifier

Particle Swarm-Classifier

According to the above descriptions, designing a MCPS-classifier has the pseudo-code in Figure 1. In a MCPS-classifier each particle is selected randomly from the solutionspace and has the form of \( P = [W_1, W_2, \ldots, W_i, \ldots, W_H] \) where \( W'_i = [w'_1, w'_2, \ldots, w'_j, \ldots, w'_{n+1}] \) is the weight vector of \( i \)th hyperplane, and \( H \) is the predefined number of hyperplanes. Fitness function can be defined as Eq. (11). Default maximum number of iterations or the best fitness value can be considered as termination condition. After enough iteration the particles converges to a solution and the decision hyperplanes with minimum misclassified training points is achieved.

IMPLEMENTATION AND RESULTS

Datasets

Three pattern recognition problems with different augmented feature vectors dimensions (10, 14 and 31) were used to show the performance of the MCPS-classifier. These datasets obtained from University of California at Irvine machine learning repository (http://mlearn.ics.edu/MLRepository.html). A description of the data sets is given here:

Wisconsin diagnostic breast cancer

Breast cancer is the first current cancer and is the second largest cause of cancer deaths among women. WDBC dataset is arrived from Dr. Woldberg’s clinical cases reports and contains 569 instances. WDBC has 30 inputs that are continuous and classify a tumor as either benign or malignant.

Wisconsin breast cancer dataset

This breast cancer data set was created by Wolberg from the University of Wisconsin. It contains 699 instances characterized by nine features: (1) Clump thickness, (2) uniformity of cell size, (3) uniformity of cell

Journal of Medical Signals & Sensors
Vol 5  |  Issue 1  |  Jan-Mar 2015
shape, (4) marginal adhesion, (5) single epithelial cell size, (6) bare nuclei, (7) bland chromatin, (8) normal nucleoli, and (9) mitoses, which are used to predict benign or malignant growths. In this data set, 241 (34.5%) instances are malignant and 458 (65.5%) instances are benign.

Heart-statlog
The data set is based on data from the Cleveland Clinic Foundation and it contains 270 instances belonging to two classes: The presence or absence of heart disease. It is described by 13 features (age, sex, chest, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic, maximum heart rate, exercise induced angina, old peak, slope, number of major vessels and thal).

Partition of Datasets
A well-known ten-fold cross validation procedure is used to supply the dataset. Each dataset is partitioned into ten data subsets and MCPS-classifier and other classifier are executed once for each partition. In each run a different partition is used as testing set and the remaining 9 are grouped together to build training set. The training set is used to train the model for good learning capability, in which the generalization capability of the proposed classifier is evaluated by the testing set.

Comparison with Promising Methods
The performance of proposed classifier is compared with the performance of k-NN classifier, PS-classifier, GA-classifier and ICA-classifier to show that the average recognition rates of the designed classifiers is better than k-NN as a conventional classifier and PS-classifier, GA based-classifiers and ICA-classifier as new classifiers. In k-NN classifier k is considered equal to, where T is the number of training samples.

In PS-classifier, GA-classifier, ICA-classifier and proposed classifiers for each problem the initial population size is set to 20 and the termination condition is considered as a maximum value of number of function evaluation, which is set to an experimentally obtained value of 10000. In PS-classifier to effectively balance the local and global search abilities of the swarm, the inertia weight is decreased linearly from 0.9 to 0.4 throughout the search process. The learning factors $c_1$ and $c_2$ are set equal to 2. Roulette-wheel selection, uniform crossover with crossover probability $p_c (p_c = 0.5)$ and uniform mutation with mutation probability $p_m (p_m = 0.3)$ for GA-classifier is considered. In simulation of ICA-classifier, revolution rate, damp ratio and unifying threshold respectively are set to 0.2, 0.99 and 0.02. Furthermore, the number of imperialists and the colonies are considered 4 and 16.

Performance Comparisons
Performance of proposed MCPS-classifier, compared with k-NN classifier, PS-classifier, GA-classifier and ICA-classifier and all of them are tested on the data sets described earlier.

All algorithms are coded and executed on the same computer in MATLAB 7.12. Tables 1-3 present the results corresponding to WDBC data, WBC data and heart-statlog data classifications, respectively. These tables show ten-fold cross validation results of the all studied classifiers for each of the three data sets. In all the datasets, the performance metrics of the 10 runs are averaged and report. Testing accuracies, standard deviations of testing accuracy, sensitivity, specificity, MCC are shown in these tables. These values demonstrate the ability of proposed classifier, in comparison with other mentioned classifiers. As these tables show testing accuracy and MCC of MCPS-classifier is better than other classifiers in every three datasets and it can be seen MCPS-classifier give reasonably good results in these dataset. These experiments have been done using 2 hyperplanes for all the datasets ($H = 2$).
Statistical paired t-test using accuracy is also conducted for all data sets. Specifically, paired t-test between MCPS-classifier and each one of the other methods is conducted. The results of t-test at the confidence level of 5% between the MCPS-classifier and each of the other algorithms is shown in Tables 1-3. “+” indicate that the proposed algorithm is significantly better than the compared algorithm. “≈” indicates that the difference is not statistically significant.

From results of the studied classifiers (in without feature selection manner) following points can be seen:

For the WDBC dataset, MCPS-classifier is the best classifier with 92.6071% means testing accuracy and 1.2366% standard deviations. Other classifiers give lower testing accuracies: GA-classifier 72.3333%, PS-classifier 68.5556%, k-NN classifier 65.9259% and ICA-classifier 65.2963%.

In all dataset, the highest accuracy is reported when feature selection is employed.

From the results of the WDBC dataset classification, it can be seen that the best result is achieved using MCPS-classifier with feature selection with an accuracy of 92.7857%, sensitivity of 0.9472, specificity of 0.9179 and MCC of 0.8457. These results are achieved with about 15 features, compared with the 30 features of the original dataset.

In WBC dataset the best accuracy of 95.4179% is reached using MCPS-classifier with feature selection. In this dataset, the best accuracy is seen with <5 features, compared with the 9 features of the original dataset.

As can be seen in Table 3, when the dataset is classified using MCPS-classifier with the original features, classification
accuracy of 74.1111%, sensitivity of 0.7520, specificity of 0.7254 and a MCC of 0.4729 are obtained. All the results were improved using feature selection, the accuracy increased from 74.1111% to 75.8889%, the sensitivity increased from 0.7520 to 0.7678, the specificity increased from 0.7254 to 0.7461, and the MCC value increased from 0.4729 to 0.5093.

The results of classification mentioned medical dataset indicate that some redundancy features does exist in the whole feature set, and feature selection is an important and necessary block in model construction.

Figure 2 shows the average rate of recognition (%) with respect to the number of function evaluation for (a) WDBC data classification, (b) WBC data classification, and (c) heart data classification. In Figure 3 for fair comparison between proposed PSO and standard PSO we consider number of function evaluation instead of number of iteration.

Figure 3 demonstrates the fact that MCPS-classifier finds a proper trajectory for converging to the solutions with lower number of function evaluation and this is the result of using the mutation operator and chaotic sequences.

CONCLUSION

This paper presents mutation operators and chaotic sequences to overcome the premature convergence and enhance the optimization performance of PSO. Effectiveness and powerfulness of MCPSO as a global search metaheuristic algorithm, especially in high dimensional spaces, were motivated us to design swarm intelligence based-classifier. Due to this, the MCPSO is used to obtain the decision
hyperplanes in the feature space. The experimental show that the performance of the MCPS-classifier better than those of the k-NN classifier, PS-classifier, GA-classifier and ICA-classifier.

Our results also show that the propose classifier works well for medical dataset recognition. In these cases, feature selection help to reduce the amount of unnecessary, irrelevant and redundant features in datasets and improves the classification accuracy with less computational efforts.

REFERENCES

BIOGRAPHIES

Zahra Assarzadeh received B.Sc. degree in computer engineering from Mashhad University, Mashhad, Iran, in 2007, and she is currently a M.Sc. student at the Department of Artificial Intelligence Engineering, University of Isfahan, Isfahan, Iran. Her research interests include image processing, neural networks, pattern recognition, and its applications in medicine.

E-mail: assar3336@yahoo.com

Ahmad Reza Naghsh-Nilchi is an associate professor at the University of Isfahan, Iran. He received his B.S., M.S. and PhD, all in electrical engineering from the University of Utah, Salt Lake City, Utah, USA. His research interests include medical image and signal processing as well as intensive computing. He has been an author or co-author of several journal articles and conference papers and a couple of book sections. He is the editor-in-chief of the Journal of Computing and Security.

He has served as the chairman of the Computer Engineering department for three terms and now is the chairman of newly established department of Artificial Intelligent and Multimedia Engineering all at the University of Isfahan. He has collaboration with internationally known institutions and peers and served as research scholar at the National University of Ireland (summer 2011), and the University of California, Irvine (2012), He was listed as Who's Who in the World 2011®.

E-mail: nilchi@eng.ui.ac.ir