اولین کنفرانس محاسبات تکاملی و هوش جمعی
19- 21 اسفندماه، مجتمع آموزش عالی بم
(CSIEC2016)
1st Conference on Swarm Intelligence and Evolutionary Computation

An optimal SVM with feature selection using multi- objective PSO

Iman Behravan
Department of Electrical Engineering University of Birjand
Birjand, Iran
i.behravan@gmail .com

Oveis Dehghantanha
Department of Electrical Engineering University of Birjand
Birjand, Iran
O_dehghantanha@yahoo.com

Seyed Hamid Zahiri
Department of Electrical Engineering, University of Birjand
Birjand, Iran
shzahiri@yahoo.com

***Abstract*-Support vector machine is a classifier, based on the structured risk minimization principle. The performance of the SVM, depends on different parameters such as: penalty factor, C, and the kernel factor, σ. Also choosing an appropriate kernel function can improve the Recognition Score and lower the amount of computation. Furthermore, selecting the useful features among several features in dataset not only increases the performance of the SVM, but also reduces the computation time and complexity. So this is an optimization problem which can be solved by a heuristic algorithm. In some cases besides the Recognition Score, the Reliability of the classifier's output, is important. So in such cases a multi-objective optimization algorithm is needed. In this paper we have got the MOPSO algorithm to optimize the parameters of the SVM, choose appropriate kernel function and select the best features simultaneously in order to optimize the Recognition Score and the Reliability of the SVM. Nine different datasets, from UCI machine learning repository, are used to evaluate the power and the effectiveness of the proposed method (MOPSO-SVM). The results of the proposed method are compared to those which are achieved by RBF and MLP neural networks.**

Keywords: Multi-objective optimization, Particle Swarm Optimization, Pattern Recognition, Support Vector Machines

I. INTRODUCTION

A pattern recognition system consists of different parts. One of the most important parts of such a system is classifying, which is done by different classifiers at the end of the process. Obviously having a powerful classifier with high accuracy is critical in a pattern recognition system, since the output accuracy of the system is highly affected by the accuracy of the classifier. So an accurate pattern recognition system which can be used in different applications, strongly needs a high performance classifier. One of the powerful classifiers which has been introduced recently is Support Vector Machine, briefly called SVM. SVM is a supervised learning method that constructs a classification model using training data [1]. SVM minimizes the generalization error and maximizes the geometric margin between two classes. This classifier uses a kernel function to map the input data into a high-dimensional feature space in order to find an optimal hyperplane to separate the two-class data. The performance of the SVM depends on the amount of kernel parameter, σ, and the amount of penalty factor, C. Also choosing an appropriate kernel function is important. Furthermore, selecting the useful features among several features in the training dataset to train SVM, plays an important role in improving the performance of the SVM. So before training the SVM, the user should select a suitable kernel function and also optimal amounts for kernel parameter and penalty factor. Besides that, as mentioned before, feature selection is important for improving the performance and reducing the complexity. To solve this problem different methods based on heuristic algorithms have been proposed. For example, Huang and Wang have used GA algorithm to optimize the SVM's parameters and also performing feature selection simultaneously in order to increase the classification accuracy [2]. They used RBF kernel in all experiments. A similar research has been done by M.Zhao et al. in 2013 applying Genetic algorithm to SVM using Gaussian kernel [3]. B.Samanta et al. have proposed a GA-SVM method for bearing fault detection in rotating machines [4]. They had Genetic algorithm, optimize the parameters of SVM and also perform feature selection to improve the SVM ability in recognizing the vibration signals. Wu et al. proposed a method, based on GA and SVM, for predicting bankruptcy [5]. They have used GA only to optimize the classifier's parameters without feature selection. Like GA, other optimization algorithms such as PSO and SA have been used to promote the SVM's performance in different practical fields like Biomedical ([7] to [9]) and Face Recognition [10]. Another important point that is not considered in the mentioned researches, is the reliability of the classifier. Which means the validation of the classifier's output. This is a very critical point that should be considered in selecting a classifier for different applications such as military and medical. In all mentioned researches, the researchers have used only one fitness function to evaluate their methods. But in addition to recognition score, calculating the reliability of the classifier's output is a good way to evaluate the performance of the classifier. Reliability means the validation of the classifier's output, for an unknown sample. In some problems, although the recognition Score of a class is high, the corresponding reliability of that class may be low and vice versa. Fig.1 shows this concept. According to Fig.1 the Recognition Score of the hollow circles is 100% but the corresponding Reliability is (5/6) 83%. These numbers for dark circles are 80% and 100% respectively. In this study we have used multi-objective form of PSO to find optimal hyperplanes for two objective functions: recognition Score and reliability as illustrated in Fig 2.

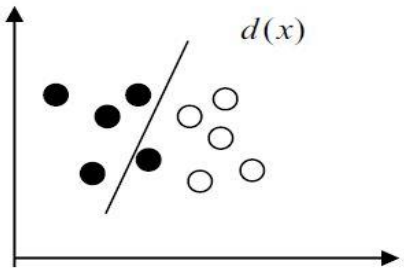

Figure 1. The Recognition Score for the hollow circles and dark circles are 100% and 80% respectively. The corresponding Reliabilities are 83% and 100% respectively.

The remainder of this paper is organized as follows. In section II, SVM is briefly introduced. In section III, PSO and MOPSO algorithms are reviewed. In section IV, we have introduced the proposed method. Section V shows the experimental results and the final section is devoted to conclusion.

II. SUPPORT VECTOR MACHINE

SVM is a two-class classifier described as follows [11]. Let (x_i , y_i), 1< i <N, indicates a set of data containing N training samples. Each sample must conform the criteria $x_i \in R^d$. y_i demonstrates the class of corresponding sample, x_i. So $y_i \in$ {-1,1} and *d* indicates the number of dimensions of input data. The separating hyperplane can be derived as in (1):

$$w.\ x_i + b = 0 \ , \ 1 \le i \le N \qquad (1)$$

If such a hyperplane exists, then linear separation is obtained. The samples which are nearest ones to the separating hyperplane are called support vectors. In boundaries (support vectors), (1) is reformed as (2):

$$w.\ x_i + b = \pm 1 \qquad (2)$$

According to (2) for each sample (3) is true:

$$y_i.(w.\ x_i + b) \ge 1 \qquad (3)$$

So the problem is, finding w and b. there are numerous hyperplanes which can separate the two-class data but SVM produces the optimal hyperplane as indicated in Fig. 3. This hyperplane has the maximum distance to support vectors. The margin of a separating hyperplane is $\frac{2}{||w||}$. So if we want to find the optimal hyperplane, we should minimize $||w||$. For simplicity we can substitute $\frac{1}{2}||w||^2$ with $||w||$. So we are dealing with an optimization problem. It means that we have to minimize $\frac{1}{2}||w||^2$ subjected to (3). In Fig. 3 the samples are linearly separable, but in most cases they can't be separated as easy as indicated in Fig.3. For nonlinear problems positive slack variables ζ_i are introduced. So the problem changed into (4):

$$\text{Min } \frac{1}{2}||w||^2 + C.\sum_{i=1}^{n} \zeta_i \qquad (4)$$

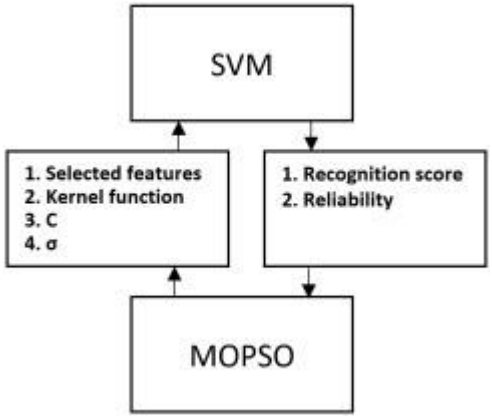

Figure 2. Block diagram of proposed method

$$\text{s.t } y_i.(w.\ x_i + b) \ge 1 - \zeta_i \ , \ \zeta_i \ge 0 \ , \ 1 \le i \le N$$

In (4) C is called Penalty factor. It is introduced to control the tradeoff between margin maximization and error minimization. This problem can be solved by means of Lagrange multipliers. Thus the classification decision function becomes:

$$F(x) = \text{sign} (\sum_{i=1}^{N} \alpha_i . y_i . K(x_i, x_j) + b) \qquad (5)$$

Where α_i are the Lagrange multipliers. $K(x_i, x_j)$=φ(x_i).φ(x_j) is kernel function through some
another mapping function, φ(x). a QP solver is used to find α_i. After that w and b can be achieved by:

$$W = \sum_{i=0}^{N} \alpha_i . y_i . \varphi(x_i) \qquad (6)$$

$$b = \frac{1}{N_{SV}} \sum_i (y_i - \sum_j \alpha_j . y_j . K(x_j, x)) \qquad (7)$$

In (7) N_{SV} is the number of support vectors and x is the input, unknown sample. Some common kernel functions are:

Linear : k(x,y) = x.y + 1

Polynomial: k(x,y) = $(x.y + 1)^{\sigma}$

RBF: k(x,y) = exp($\frac{-||x-y||}{2.\sigma^2}$)

Quadratic: 1 - $\frac{||x-y||^2}{||x-y||+\sigma}$

In all these functions σ should be optimally tuned with C.

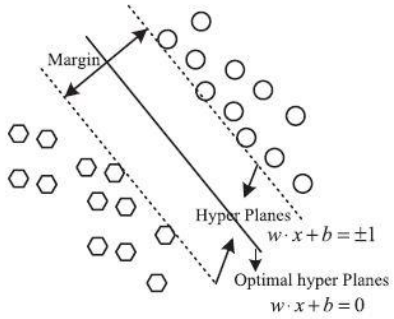

Figure 3. Optimal Hyperplane.

III. PARTICLE SWARM OPTIMIZATION METHOD

A. Single Objective PSO

Particle swarm optimization algorithm first suggested by Kennedy and Eberhard in 1995 [12]. This algorithm is produced by inspiration of birds flocking and fishes grouping. In fact they used the mechanism of birds flocking to solve optimization problems. It means that a group of particles search the solution space for the best solution. Each particle has a position, velocity and a memory to save its best position from the beginning of the process. In each iteration the particle which has the best position is regarded as the leader and the other particles tend to reach to its position. So their movement is affected by two factors: their best position from the first iteration to current iteration and the leader's position. (8) and (9) describe how particles move through iterations.

$$v_{id}^{t+1}=\text{w}.v_{id}^{t}+c_1.\text{rand}.(p_{best}^{d}-x_{id}^{t})+c_2.\text{rand}(p_{gbest}^{d}-x_{id}^{t}) \quad (8)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t} \quad (9)$$

In the above equations, v_{id} is the *d*th dimension of the velocity of the *i*th particle, *x* denotes the position of the particle, *t* is the number of iteration, c_1 and c_2 are learning factors, rand is a positive random number between 0 and 1 under normal distribution, *w* is the inertia weight coefficient, p_{best} is the best position of the particle from the beginning to current iteration and p_{gbest} shows the position of the leader in each iteration.

B. Multi Objective PSO

In a multi-objective optimization problem obviously, there are more than one objective function, so a multi-objective optimization problem can be defined as follows [13]

$$\text{Minimize F(x)} = [\, f_1(x), f_2(\text{x}), \ldots, f_k(x)] \quad (10)$$

$$\text{s.t } g_j(x) < 0 \quad \text{and } h_j(x) = 0$$

Where X = (x_1, x_2, …, x_n) is a solution, f_i, i=1, …, k, are objective functions and g_j, h_j are constraints of the problem. On the contrary to single-objective case, here we can't find a single solution which is the best for all objective functions. Instead we are looking for a set of solutions. Actually there is a trade-off between different objective functions. So the definition of the optimality is different in this case. We call X, an optimal solution if another solution, like Y, can't be found which has better fitness in all objective functions. Such a solution is called a Pareto optimal [14]. We say x_1 is dominated by x_2, if x_2 is better than x_1 in all objective functions. But if x_1 is better just in one objective function than x_2, it is non-dominated. So in multi-objective form we have a set of solutions that contains non-dominated particles. It means that the members of this set can't dominate each other. Fig.4 shows Pareto optimal front for a two-objective function problem. According to this picture the solutions in the Pareto front dominate the other solutions but can't dominate each other. In MOPSO each particle has a set of leaders and has to select one of them through a mechanism. Usually this set is called External Archive ([15],[16]). External Archive contains non-dominated particles from the first iteration.

In fact External Archive preserves outputs of the algorithm. Up to now different versions of MOPSO are introduced. In this study we have used the one, introduced in [17] because of its speed and rapid convergence. In this form to select a leader for each particle, the solution space is divided into numerous hypercubes and different solutions from the external archive exist in these hypercubes.

They are placed in hypercubes according to their coordination calculated by objective functions. Each hypercube is evaluated through dividing the number of its solutions to a constant. After evaluating each hypercube, Roulette wheel mechanism will select one of these hypercubes. And finally a solution, placed in the selected hypercube, will be selected randomly as the leader for the particle. MOPSO process is described as following:

1) Initializing the position and the velocity of each particle.
2) Evaluating the particles.
3) Saving non-dominated particles in a Repository.
4) Producing hypercubes to cover the solution space.
5) Initializing the memory of each particle
p_{best}[i]=position[i] (11)
6) Main loop
   a) Calculating the velocity of each particle by (8).
   In this form p_{gbest}^{d}, should be replaced by *rep[h]*.
   b) Updating the position of the particles through (9).
   c) Evaluating the particles.
   e) Updating the repository.
   f) Updating the p_{best} for each particle.
7) End of the main loop.

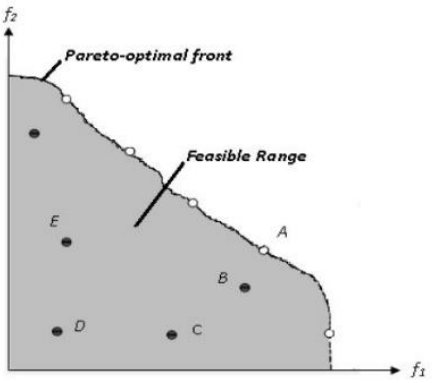

Figure 4. Pareto Optimal Front

IV. PROPOSED METHOD

In this paper we have used MOPSO to optimize penalty factor, choose adequate kernel function, tune the selected kernel's parameter and feature selection for two objective functions: Recognition Score and Reliability and we compared its performance with RBF and MLP neural networks. The construction of particles is indicated in Fig. 5.

The first variable, C, is for tuning penalty factor. KN is for selecting kernel functions. The amount of this variable can be 1, 2, 3 or 4 to choose one kernel among the four kernels introduced in section II. σ is for selecting the selected kernel's parameter (except linear). The rest of the particle are for feature selection. For a dataset with n number of features, F_1, F_2, …, F_n are between 0 and 1. If they are less than or equal to 0.5, the corresponding feature is not selected. Conversely if they are bigger than 0.5, the corresponding feature is selected.

If we consider the two classes as "positive" and "negative", then the predicted test samples can be divided into four groups.

1) Samples which are "positive" and correctly predicted as "positive" (TP)
2) Samples which are "positive" but classified as "negative" (FN).
3) Samples which are "negative" and correctly classified as "negative" (TN).
4) Samples which are "negative" but predicted as "positive" (FP).

According to these categorization, Recognition Score is calculated by (14):

$$\text{Recognition Score} = \frac{TP+TN}{TP+TN+FN+FP} \quad (14)$$

And the reliability for each class equals to (15) and (16):

$$\text{Pos-reliability} = \frac{TP}{TP+FN} \quad (15)$$

$$\text{Neg-reliability} = \frac{TN}{TN+FP} \quad (16)$$

The termination criteria are that the iteration number reaches 1000. To calculate the fitness functions, for each particle, SVM should be trained by the determined parameters, kernel function and selected features and then recognition Score and Reliability for each class can be achieved by (14) to (16). For multi-class classification we have used one-vs-all method. In this method for each class of the dataset we found the optimal hyperplane, which separates the corresponding class from the others. Thus the input sample is labeled according to the opinion of the obtained hyperplanes about that sample. Fig. 6 shows this method for a 3- class dataset. Hyperplanes about that sample. Fig. 6 shows this method for a 3- class dataset.

C	KN	σ	F_1	....	F_n

Figure 5. Construction of particles

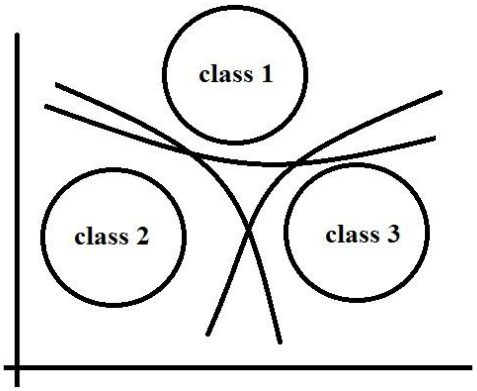

Figure. 6. Classifying of a 3-class dataset with one-vs-all method

V. EXPERIMENTAL RESULTS

According to Table 2 it can be seen that SVM gives comparable with and also better results than MLP and RBF neural networks such as Glass, Iris, Wine, Ionosphere, Hepatitis and Vowel. The important point demonstrated in the Table 2, is the rates of reliabilities given for different datasets. As indicated in Table 2, the proposed method, gives high rates of reliabilities for most of the datasets, meaning that the output of the promoted classifier is strongly reliable. In fact we have transformed a normal classifier into an expert one using a heuristic multi-objective algorithm which has not only high accuracy but also high rates of reliabilities. As shown in Table 2, MOPSO-SVM has given 100% average reliability for Wine samples. It means that all samples classified are labeled correctly. But the recognition score is 97.75% while it is expected to be 100% because there are some samples which can't be assigned to each of the classes, in other words they can't be classified. In fact since the hyperplanes obtained by MOPSO have an amount of errors in classifying of the test samples (unknown samples), some samples exist that more than one hyperplane assign them to their corresponding classes. Also there may be some samples that none of hyperplanes assign them to their corresponding classes. Such samples are considered as error samples, which their classes can't be distinguished. Fig. 7 illustrates this concept. Analyzing the numbers seen in Table 2, we can conclude that MOPSO-SVM is a powerful and effective classifier, due to rates of reliability and recognition score achieved by this method for different datasets. These numbers show that MOPSO-SVM is a reliable classifier which means that, this promoted classifier can acts perfectly in special applications such as Military and Medical which strongly require a high-reliable classifier.

The suggested method applied to nine different datasets from UCI machine learning repository [18]. In Table 1 the characteristics of these datasets are shown. Table 2 shows the experimental results on these datasets.

Table 1- Characteristics of used datasets

Dataset	No.of Classes	No.of Samples	No.of Features
Glass	6	214	9
Iris	3	150	4
wine	3	175	13
German	2	1000	20
Ionosphere	2	351	33
Sonar	2	208	60
Hepatitis	2	80	19
Bupa	2	345	6
vowel	11	990	13

Table 2- Percentage of Recognition Score and Reliability

	GLASS	IRIS	WINE	GERMAN	IONOSPHERE	SONAR	HEPATITIS	BUPA	VOWEL
MOPSO-SVM									
RECOGNITION SCORE	81.31	94.67	97.75	85.90	92.31	90.87	96.25	82.32	97.78
RELIABILITY	92.94	97.93	100	83.89	93.99	90.85	92.095	82.06	99.89
MLP									
RECOGNITION SCORE	82.78	98.54	98.42	89.86	96.44	93.76	92.76	87.94	77.6
RELIABILITY	73.088	98.68	98.438	88.35	96.35	93.97	86.51	87.59	78.30
RBF									
RECOGNITION SCORE	81.76	96.92	81.58	91.5	90.02	94.72	94.78	88.12	99.12
RELIABILITY	75.822	96.96	88.87	94.94	93.11	94.86	97.07	91.49	99.3

VI. CONCLUSION

In this study Multi-objective PSO has been used to tune the parameters of SVM and also perform feature selection process for two objective functions and compared its performance with RBF and MLP neural networks. According to Table 2, it can be seen that the proposed method gives reliabilities and recognition scores, comparable with RBF which has shown its effectiveness in classifying overlapped datasets, and in some cases even gives better reliabilities and/or recognition scores than RBF such as Glass, Iris, Wine, Ionosphere, Hepatitis and Vowel. Actually the numbers shown in Table 2, say that using a heuristic algorithm to convert SVM from a normal classifier in to an expert one was successful. Furthermore optimizing SVM in order to increase its reliability besides its accuracy by using Multi-objective heuristic algorithm is a successful idea according to Table 2. This table also shows the power and effectiveness of MOPSO in searching the solution space. In other word, MOPSO is a powerful algorithm which can be used for solving Multi-objective optimization problems

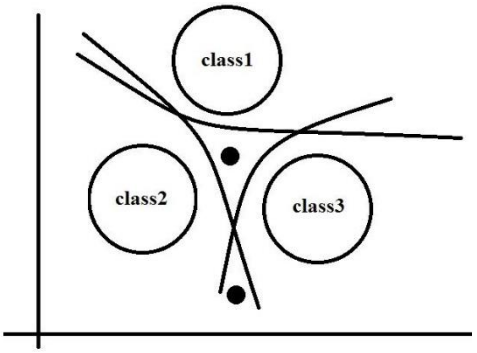

Figure 7. Samples which are considered as error samples.

VII. REFERENCE

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed.: Springer Verlag,, 1995.

[2] C.L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, pp. 231-240, 8// 2006.

[3] M. Zhao, C. Fu, L. Ji *et al.*, "Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5197-5204, 2011.

[4] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 16, pp. 657-665, 10// 2003.

[5] C.H. Wu, G.H. Tzeng, Y.-J. Goo, and W.-C. Fang, "A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy," *Expert Systems with Applications*, vol. 32, pp. 397-408, 2// 2007.

[6] F. Melgani and Y. Bazi, "Classification of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, pp. 667-677, 2008.

[7] J. S. Sartakhti, M. H. Zangooei, and K. Mozafari, "Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA)," *Computer Methods and Programs in Biomedicine*, vol. 108, pp. 570-579, 11// 2012.

[8] Q. Shen, W.-M. Shi, W. Kong, and B.X. Ye, "A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification," *Talanta*, vol. 71, pp. 1679-1683, 3/15/ 2007.

[9] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Computers in biology and medicine*, vol. 43, no. 5, pp. 576-586, 2013.

[10] J. Wei, Z. Jian-qi, and Z. Xiang, "Face recognition method based on support vector machine and particle swarm optimization," *Expert Systems with Applications*, vol. 38, pp. 4390-4393, 4// 2011.

[11] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.

[12] J. Kennedy, "Eberhart. RC, 1995, Particle swarm optimization {C}," in *International Conference on Neural Networks, IV (Perth, Australia), Piscataway, NJ,(IEEE Service Center)*, 1942.

[13] A. Abraham, and L. Jain, "Evolutionary Multiobjective Optimization," *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing A. Abraham, L. Jain and R. Goldberg, eds., pp. 1-6: Springer London, 2005.

[14] V. Pareto, CoursD'EconomiePolitique, volume I and II, F. Rouge, Lausanne, 1896.

[15] M. Reyes-Sierra and C. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International journal of computational intelligence research*, vol. 2, pp. 287-308, 2006.

[16] M. Bhuvaneswari, *Application of Evolutionary Algorithms for Multi-objective Optimization in VLSI and Embedded Systems*: Springer, 2015.

[17] C. A. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, 2002, pp. 1051-1056.

[18] Hettich, S., Blake, C., &Merz, C. (1998). UCI repository of machine information and computer sciences,available<http://www.ics.uci.edu/~mlearn/MLRepository.htm