J. Basic. Appl. Sci. Res., 3(12)210-217, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Comparative Analysis of Multi-Objective Feature Subset Selection using Meta-Heuristic Techniques

Ayesha Khan, Abdul Rauf Baig*, Kashif Zafar

National University of Computer and Emerging Sciences, Islamabad
*Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

Received: June 22 2013
Accepted: July 27 2013

ABSTRACT

This paper presents a comparison of evolutionary algorithm based technique and swarm based technique to solve multi-objective feature subset selection problem. The data used for classification contains large number of features called attributes. Some of these attributes are not significant and need to be removed. In the process of classification, a feature effects accuracy, cost and learning time of the classifier. So, before building a classifier there is a strong need to choose a subset of the attributes (features). This research treats feature subset selection as multi-objective optimization problem. The latest multi-objective techniques have been used for the comparison of evolutionary and swarm based algorithms. These techniques are Non-dominated Sorting Genetic Algorithms (NSGA – II) and Multi-objective Particle Swarm Optimization (MOPSO).MOPSO has also been converted into Binary MOPSO (BMOPSO) in order to deal with feature subset selection. The fitness value of a particular feature subset is measured by using ID3. The testing accuracy acquired is then assigned to the fitness value. The techniques are tested on several datasets taken from the UCI machine repository. The experiments demonstrate the feasibility of treating feature subset selection as multi-objective problem. NSGA-II has proved to be a better option for solving feature subset selection problem than BMOPSO.
KEYWORDS: Optimization, genetic algorithm, swarm optimization, classification, Feature subset selection

1. INTRODUCTION

The feature subset selection is developing into a challenging research area during the past decades, as data sets used for classification purposes in data mining are becoming huge in terms of number of features as well as number of instances. The datasets used for classification mostly have large number of features that are not all relevant. But all these features are used as input to the classification algorithm due to lack of sufficient domain knowledge. Some features in the dataset when used for the classification may just increase the cost and complexity of the classification algorithm. On the other hand they may be reducing the generalization ability and accuracy of the classification algorithm as well. So there is a huge need for a technique that can find smallest possible feature subset that has high classification accuracy.

The multi-objective problems include two or more objectives to be optimized at simultaneously[20], [23]. The real world consists of numerous multi-objective problems. The feature subset selection problem may also be considered as one of them. The accuracies of the different classes in a dataset are the multiple objectives to be optimized simultaneously. Hard work to increase the accuracy of one class may lessen the accuracy of another class. This research treats feature subset selection problem as multi-objective problem. It also compares the performance of genetic based multi-objective algorithm [21] with swarm based multi-objective algorithm [22] to resolve the feature subset selection problem.

The main features of the proposed method are:

- This research treats feature subset selection as a multi-objective optimization problem.
- The accuracy of each class is considered as a separate objective to be optimized
- Solving feature subset selection problem as multi-objective problem makes feature subset selection non-rigid.
- The comparison of NSGA-II (Non-dominated Sorting Genetic Algorithm II)[13]and MOPSO (Multi-objective Particle Swarm Optimization) [2]is carried out in order to solve multi-objective feature subset selection problem.

*Corresponding Author: Abdul Rauf Baig, National University of Computer and Emerging Sciences, Islamabad
Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia.
Email: raufbaig@ccis.imamu.edu.sa,

- MOPSO is converted to Binary MOPSO (BMOPSO) as feature subset selection is a binary problem in nature.
- The results have clearly shown that treating feature subset selection as multi-objective problem gives better accuracy and helps to produce less complex classifier.
- The comparison of NSGA-II and BMOPSO shows that NSGA-II is a better algorithm for multi-objective feature subset selection.

TABLE 1: Example of feature subsets and their accuracies

Feature Subset (X_i)	D1	D2	D3	D4	D5	acc(A)	acc(B)
1	1	0	1	1	0	0.9	0.8
2	0	0	1	1	1	0.8	0.9
3	1	0	1	1	1	0.2	1.0
4	0	1	0	0	1	1.0	0.1
5	0	1	1	1	0	0.6	0.6
	….	….	….	….	….	….	….

2. Multi-objective Feature Subset Selection. Feature subset selection is the problem of finding a subset of features from a larger set of features based on some optimization criteria. Multi-objectives comes in naturally while solving feature subset selection problem. Considering TABLE1, that provides the accuracies of the feature subsets for each of the class i.e. class A and class B. This table shows that feature subset X_4 gives better accuracy according to class A, while doesn't perform well as far as class B is concerned. On the other hand feature subset X_3 gives better result for class B instead of class A. If the objective is to maximize the accuracy of both the classes then they both are non-dominating to each other.

In the same manner another subset of feature X_5gives 60% accuracy for class A and 60% accuracy for class B. These three feature subsets are all non-dominated to each other. But if another feature subset provides 20% accuracy to class A and 40% accuracy to class B, this feature subset is inferior to the last three points. So the accuracies of multiple classes may be considered as multiple objectives to be optimized simultaneously. Therefore improved results may be acquired by applying multi-objective optimization techniques. For example, consider a dataset with 5 dimensions (features) and two classes (A and B) as shown in Table 1.

The user has the option to select the feature subset (from the feature subsets on the non-dominated front) which fulfills his needs. The accuracy of one class could be more important than accuracy of another class for the user. If, for a problem, all the class accuracies are important then the user can select a subset which gives high, but balanced accuracies. It may also transpire, for a given dataset, that one of the subsets may dominate all other subsets.

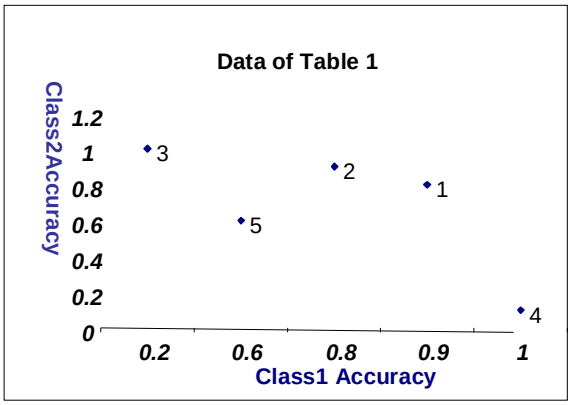


FIGURE 1: Decision Space for class A and class B

3. Methodology. This research uses NSGA II and MOPSO for the solving multi-objective feature subset selection. The selection of non-dominated points in search space takes into concern the accuracies of all the classes. ID3 classifier is used for the calculation of fitness of each candidate feature subset.

3.1 Input Parameters. The parameters used as input to both the algorithms (NSGA_II and BMOPSO) are

- Population size
- Number of cycles
- Data set
- Number of features
- Number of classes

All the values in the dataset should be of nominal type as ID3 works for nominal data only.

3.2 Chromosome Encoding. The number of solutions (chromosomes in case of NSGA-II and particles in case of BMOPSO) initialized for both the algorithms is equal to the population size given by the user. A single chromosome (solution)has bits equal to total number of features in the data set. The value of the bit is 0 is the feature is not present in that particular solution and 1 if the feature is present as shown in Table 2.These binary strings are initialized randomly in the beginning.

TABLE 2: An example of a single chromosome

f1	f2	f 3	f 4	f 5	f 6	f 7	f 8	f 9	f10
0	1	0	1	1	0	1	0	1	1

3.3 Conversion of Multiple to Two Class Problem. As mentioned before each class's accuracy is an objective so there is a need of converting a multiple class problem into two class problem for each class in the dataset. This conversion is necessary because the fitness of each class needs to be evaluated separately. For the conversion the concerned class is labeled as Y while all the other classes are labeled N.

3.4 Trimming Data According to Feature Subset. Before any solution is evaluated, the data is trimmed according to the feature subset represented by the solution. The data is trimmed by deleting the columns of those attributes that are not present in that particular feature subset. After that this trimmed data is evaluated.

3.5 Evaluation Function. The entire data set is separated into training and testing data in the start of the algorithm. In the evaluation function each solution is evaluated by first applying ID3 algorithm on training data for each converted class (see section 3.3). The decision tree is the output of ID3 algorithm. The testing data is then used to calculate the accuracy of the solution. The percentage of the instances of the testing data correctly classified by the decision tree is the accuracy of the solution. For example, testing data has 20 instances and 15 out of them are correctly classified. Then the accuracy of the solution will be 75% which is basically the fitness of the solution. This evaluation function creates tree for each solution in a population and for each class. If there are 20 solutions and 3 classes, then 3 trees are built for each class (a total of 60 trees).

The fitness value for each class is considered as a separate objective to be optimized in a multi-objective space. The selection of a particular chromosome depends on the fitness values of all the class and the distance of that chromosome from other chromosomes.

3.6 Applying NSGA-II. The solutions (population) are randomly initialized. The solutions are then sorted with the help of preprocessed data based on non-domination. The population is divided into different fronts. The sorting is done after evaluating each candidate subset. As stated before according to the candidate feature subset, the data is trimmed and is then passed on to the evaluation module. The feature subsets in the same front are given the same fitness value known as rank. Along with the fitness value the crowding distance is also calculated for each solution. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population [13].

The method used for selecting parents for crossover is binary tournament selection. This selection is based on the rank and crowding distance. A solution is preferred over the other solution if its rank is higher than the other. But if the ranks of both the individuals are the same, then the decision is made on the basis of crowding distance. A solution with higher crowding distance is given preference. After the selection of parents, these parent solutions are used to produce the next generation through crossover and mutation. The old and the new generations are combined. The combined solutions are then sorted again based on non-domination. And only the best N solutions are selected, where *N* is the population size. The selection is based on rank and then on crowding distance on the last front.

The last population comprises of the feature subsets in the form of chromosomes, along with the fitness according to each class and the rank of the chromosome. The user can choose any of the feature subset that has rank equal to 1.

3.7 Applying BMOPSO. The Multi-objective Particle Swarm Optimization deals with datasets having real values. The feature subset selection problem is binary in nature, as each solution i.e. feature subset is represented in binary form. So, MOPSO is altered to Binary MOPSO (BMOPSO) as feature subset selection problem is binary in nature.

For binary PSO, particles represent different feature subsets in binary space. Each element of the particle's position vector can take only binary value 0 or 1. The change in particle's position is basically a mutation of bits, by flipping a bit from one value to the other. The velocities are defined in terms of probabilities that a bit will be in one state or the other. For example, velocity = 0.3 implies a 30% chance to be bit 1, and a 70% chance to be bit 0. This means that velocities are restricted to be in the range [0,1] to be interpreted as a probability.

The BMOPSO algorithm for feature subset selection works in the following manner. The population is initialized and the velocity of each particle is set to zero. Each particle is evaluated and the non-dominated particles are stored in the external repository EP. The hypercube is generated for the search space explored so far and the particles are located where each particle's coordinates are defined according to the value of accuracy for each class. In the beginning the personal bestPbest for each solution (particle) is initialized to itself.

While last count of cycles is not reached following steps are repeated.

(a) The velocity of each particle is calculated using following expression

$$v[i] = W\ x\ v[i] + r1\ x\ (Pbest[i] - Pop[i]) + r2\ x\ (EP[h] - Pop[i])$$

where$W$ is the inertia weight; $r1$ and $r2$ are random numbers in the range [0..1]. The procedure for determining the index h is as follows. The hyper cubes containing more than one particle are assigned a fitness by dividing 10 by number of particles that it contains. The hypercubes with one particle will be assigned fitness equal to 10. Then a roulette wheel selection is used to select the hypercube from which one particle is chosen randomly that will be EP[h].

(b) Then new positions of the particles are computed by using velocity as the probability by which the particular bit of the particle is flipped or not.

(c) Each particle in the population is evaluated as mentioned in Section 3.5.

(d) The contents of EPare updated along with the graphical representation of the particles within the hypercubes. The update includes the insertion of all the current non-dominated particles. The size of the REP is limited so if it is full, less crowded particles are preferred by examining the current hypercube.

(e) The personal bests Pbests are updated.

4. Experimentation. The experiments have been carried out using 3.2 GHz Intel processor with 2RAM. The tool used for development is MATLAB 7.0.

The experiments reported here use real-world data sets to authenticate the practicability of the proposed technique for feature subset selection. These datasets are taken from the data repository of machine learning at the University of California at Irvine [19]. The experiments were performed on four datasets that are Salary Data, Pittsburgh Bridges Data, DNA Sequences and House Votes

The reason for choosing these data sets out of many other possibilities is that these data sets required minimum preprocessing.

The objective was to experiment with data sets having variations in records size and dimensions. The datasets vary in sizes such as *salary data* is relatively large with 2270 rows while *house votes* is medium sized data and the other two datasets, *Pittsburgh bridges* and *DNA sequences* are of small sizes. In the same manner the datasets vary in terms of dimensions. *DNA sequences* have 57 attributes while other datasets have smaller number of dimensions.

The data preprocessing is the first step in the experimentation. The classification algorithm is ID3 in our algorithm that accepts only nominal values. For this restriction, all the continuous values are converted into nominal values. Apart

from that missing values are handled before giving the data as input to the algorithm. All these nominal data values are then encoded in digits. This encoding is done as it simplifies the implementation of our algorithm.

The accuracy of the selected feature subset is calculated by applying ID3 algorithm. The data set is trimmed according to the selected feature subset. Then this trimmed data is given as input to the ID3 algorithm. The application of ID3 outputs a decision tree. This decision tree is then tested on the testing data. The testing accuracy of the tree is the accuracy (fitness) of the selected feature subset for the class under.

For the authentication of the results, 30 runs have been carried out for each experiment and the average of all the 30 runs have been reported in the results. Along with this, testing is based on 10-fold cross-validation.

4.5 Parameter Setting and Comparisons. While applying NSGA-II for feature subset selection, the population size is set to 70 as before applying 10-fold cross validation some experimentation was carried out with different population sizes ranging from 20 to 100.This experimentation showed that 70 is the appropriate population size. The binary tournament is used as selection procedure and the mating pool size is set as half of the population. The probabilities for crossover and mutation are 20% because the authors of NSGA-II have found that increasing or decreasing these probabilities results in degradation of the accuracy. While applying BMOPSO for feature subset selection, the population size is set to 90 as before applying 10-fold cross validation some experimentation was carried out with different population sizes ranging from 20 to 100.

5. Results. This section shows the results obtained by applying our feature subset selection technique on the four datasets. The results are validated by 10-fold cross validation. The data is divided in ten equal parts for the validation purpose. In the evaluation function, ID3 is applied for each solution ten times. Each time nine parts are used for training while the remaining tenth part is used as testing data. The average of all the ten testing accuracies from all the ten parts is taken in order to declare it as the fitness of the particular feature subset.

The comparison of accuracy produced by using simple ID3, NSGA-II and BMOPSO is shown in the table 7.Along with this four different graphs for the four datasets showing the pareto-front obtained by applying NSGA-II and BMOPSO are also shown in figure 2-5.

The results have been produced using multiple datasets that are taken from the data repository of machine learning at the University of California at Irvine [19].

By treating feature subset selection as multi-objective problem, pareto-front is obtained that has more than one trade-off optimal solution. The direct comparison of ID3 with our proposed techniques is not possible. So for the accuracy of class 1, the feature subset considered is the one giving highest accuracy considering class 1 only. In the same manner, for the accuracy of class 2, the feature subset considered is the one giving highest accuracy considering class 2 only. If the user needs a balanced accuracy for both classes, than he can choose among other solutions in between these two.

The results clearly show that NSGA-II provides better results when applied for feature subset selection. NSGA-II provides a superiorpareto-front for all the datasets.

TABLE 7: Comparison of ID3, NSGA-II and BMOPSO

Datasets	**ID3**			**NSGA II Selected Subset**			**BMOPSO Selected subset**		
	Features	**Accuracy**		**Features**	**Accuracy**		**Features**	**Accuracy**	
		Class1	Class2		Class1	Class2		Class1	Class2
Salary data	11	0.7504	0.7519	6-8	0.8156	0.80887	7-8	0.796	0.795
Pittsburgh bridges	11	0.8000	0.7368	5-8	0.8901	0.84563	8-9	0.875	0.83
DNA sequences	57	0.7619	0.7500	35-42	0.8761	0.8989	40-43	0.86	0.865
House votes	16	0.9302	0.9302	8-10	0.9866	0.97332	7-9	0.98	0.985

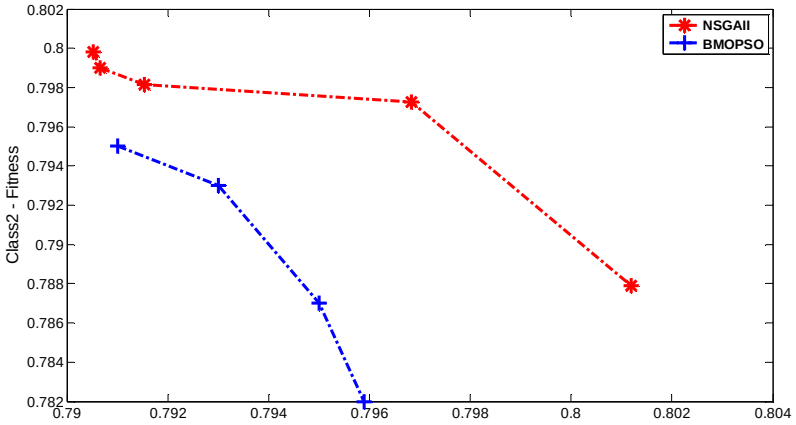
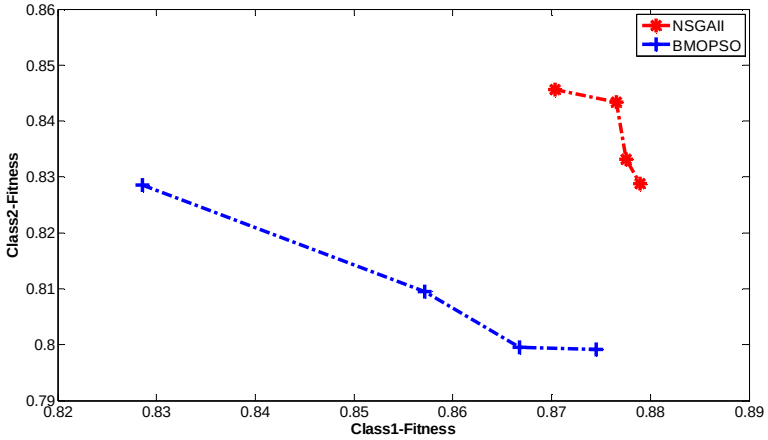


FIGURE 2: Results for Salary Dataset



-

FIGURE 3: Results for Bridges Dataset

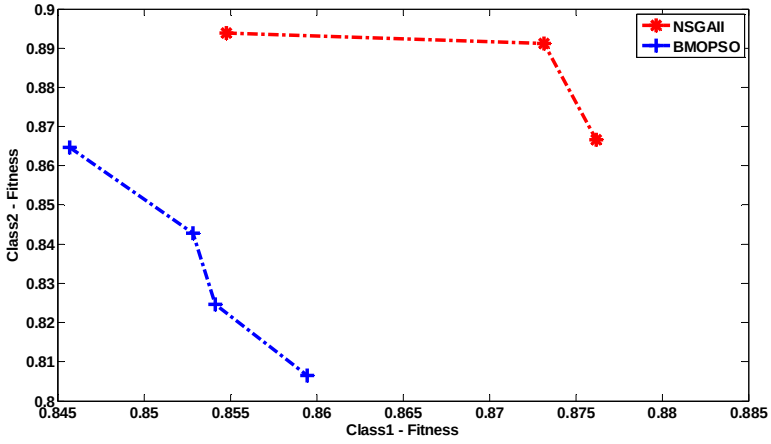


FIGURE 4: Results for DNA Dataset

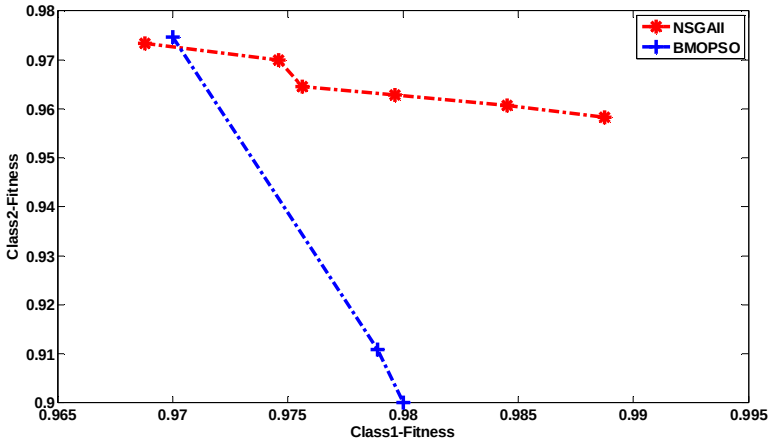


FIGURE 5: Results for Housevotes Dataset

6. Conclusion. This research has used multi-objective optimization for feature subset selection problem. The wrapper approach has been used to solve this problem. The fitness for each possible solution during the algorithm is measured by applying ID3 and the testing accuracy of the built tree is considered as the fitness value of that particular feature subset. The problem of feature subset selection is considered as multi-objective problem considering each class is an objective to be optimized. This was the basic motivation of using multi-objective genetic algorithm for solving feature subset selection problem.

The experimentation is carried out on four real-life data sets. The results are authenticated by using 10-fold cross validation. First part of the experimentation is to observe the effect of applying NSGA-II for feature subset selection. The second part includes the effect of applying BMOPSO for feature subset selection. The results have shown that treating feature subset selection as multi-objective problem is beneficial by providing non-rigidness, less complex classifiers and better accuracy. Apart from this conclusion, results have also shown that NSGA-II is a better candidate for multi-objective feature subset selection by providing better pareto-fronts for all the datasets

7. Future Work. The experimentation on more datasets with multiple classes will be done as a future enhancement to evaluate the feasibility of these techniques with other techniques that solve feature subset selection problem. The research in future will also comprise of other multi-objective algorithms for solving feature subset selection problem. The comparison among the application of all these algorithms for feature subset selection problem will be carried out. The hybrid approach combining neural network and evolutionary algorithm can be used to improve the results further and to make them less conservative.

Acknowledgment

The authors declare that they have no conflicts of interest in this research.

REFERENCES

[1] A. P. Engelbrecht, "Computational Intelligence – An Introduction", *John Wiley & Sons Inc.*, NJ, USA 2002

[2] C. A. Coello, "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, Jun 2004

[3] D. W. Aha, and R. L. Banket, "A comparative evaluation of sequential feature selection algorithms" in *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, pp 1–7, Menlo Park, CA, USA. 1994

[4] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem", in *Proceedings of the Eleventh International Conference on Machine learning*, pages 121–129, New Brunswick, NJ, 1994

[5] H. Vafaie, and K. D. Jong, "Genetic algorithms as a tool for feature selection in machine learning," in *Center for Artificial Intelligence*, George Mason University, 1992

[6] H. Vafaie, and K. D. Jong, "Genetic algorithms as a tool for restructuring feature space representations", *Computer Science Department*, George Mason University Fairfax, USA, 1995

[7] I. S Oh, J. S. Lee, and B. R. Moon, "Hybrid genetic algorithms for feature selection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, Nov 2004

[8] J. D. Shaffer, "Multiple objective optimization with vector evaluated genetic algorithms", in *Proceedings of an International Conference on Genetic Algorithms and their Applications*, Pittsburgh, PA, Jul 1985

[9] J. Han and M. Kamber, "Data Mining Concept and Techniques", *Morgan Kaufmann Publishers*, San Francisco, CA, USA, 2001

[10] J. Horn and N. Nafpliotis, "Multi-objective using the niched pareto genetic algorithm" *IlliGAL Report 93005*, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign, Jul 1993

[11] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multi-objective optimization", *in Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, Piscataway, NJ, Jun 1994

[12] J. Yang, and V. Honavar, "Feature subset selection using a genetic algorithm" *IEEE Intelligent Systems*, vol. 13, no. 2, 1998

[13]K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast elitist multi- objective genetic algorithm: NSGA-II", *Evolutionary Computation*, vol. 2, 1995

[14] K. Deb, "Multi-objective optimization using evolutionary algorithms", *Reading, John Wiley & Sons, Ltd*, Reprinted in April 2002

[15] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A.K. Jain, "Dimensionality reduction using genetic algorithms", in *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, Jul 2000

[16] N. Srinivas, and K. Deb, "Multi-objective optimization using non-dominated sorting in genetic algorithms", in *IEEE Transactions on Evolutionary Computation*, 1994

[17] P. Hajela, and C.Y.Lin, "Genetic search strategies in multi-criterion optimal design," *Structural Optimization*, vol. 4, June 1992

[18] S. M. Weiss, and C.A. Kulikowski, "*Computer Systems That Learn*", Morgan Kaufmann, 1991

[19] Datasets from the "University of Irvine" http://archive.ics.uci.edu/ml/

[20] M. Rashid, A. R. Baig and K. Zafar, "Niching Sub-swarm based Particle Swarm Optimization" *Proceedings of IEEE-ICITE*, Malaysia, 2009

[21] K. Zafar, A. R. Baig, and A. Khan, "Collaborative Evolutionary Planning Framework (EPF) for Route Planning", *International Journal of Computer Applications*, Vol. 4, No. 8, August 2010

[22] K. Zafar, and A R. Baig, "Optimization of Route Planning and Exploration Using Multi Agent System", *Multimedia Tools and Applications;* Springer, ISSN: 1380-7501, 2012

[23] K. Zafar, and A. R. Baig, "Multiple Route Generation Using Simulated Niche Based Particle Swarm Optimization", *Computing and Informatics*, ISSN: 1335-9150, 2013