

# A Comparative Study of Evolutionary Algorithms with a New Penalty Based Fitness Function for Feature Subset Selection

Atsushi Kawamura and Basabi Chakraborty<sup>(✉)</sup>

Faculty of Software and Information Science, Iwate Prefectural University,  
152-52 Sugo, Takizawa 020-0693, Japan  
basabi@iwate-pu.ac.jp

**Abstract.** Feature subset selection is an important task for knowledge extraction from high dimensional huge data which reduces dimension of data, accelerates processing of data and improves classification accuracy. For mining of knowledge from huge data, feature subset selection acts as extraction of the context for classification process. Feature subset selection is basically an optimization problem in which a search technique is used to find out the best possible feature subset from all possible subsets of a large feature set with the use of a feature evaluation function. Evolutionary techniques are well known for their efficiency as search algorithms and are used in feature subset selection problem. In this work, a comparative study of well known EC algorithms (GA, PSO) and not so known algorithms (CS, GSA, FireFly and BAT) used for feature subset selection has been done with classification accuracy of a SVM classifier as the wrapper fitness function. A new fitness function with an added penalty term based on two objectives of improving classification accuracy while reducing dimension is proposed and its efficiency over classification accuracy alone is examined by simulation experiments with bench mark data sets. The simulation results show that the new fitness function is more effective than classification accuracy based fitness function. It also produces better results in reducing dimension and improving classification accuracy than using popular multi-objective search algorithm NSGA II for feature selection.

## 1 Introduction

With the increasing generation of information and data due to advancement of internet and communication technologies, the analysis and summarization of data for knowledge extraction from huge data becoming a challenge day by day. For efficient categorization, mining or classification, the data need to be preprocessed to contain the characteristic or discriminatory information while being free from redundant and irrelevant information. Feature selection aims to solving this problem and it is an important preprocessing task in the area of pattern recognition or data mining [1,2] prior to classification or clustering. A sample data or pattern in the paradigm of pattern recognition or machine

learning is represented by a  $n$ -dimensional vector or a point in a  $n$ -dimensional space where individual dimension represents individual feature. Feature subset selection refers to the processing of selecting a subset of  $d$  features from the set of  $n$  features by discarding irrelevant features and retaining discriminatory informative features. Reduction of features facilitates speedy processing of data and improves classification accuracy. Feature extraction also reduces dimensionality by projecting original high dimensional feature set to a lower dimensional set in which the new features are created instead of retaining a subset of original features. This paper focuses on feature selection paradigm.

Basically feature subset selection process needs to define two things: an evaluation function to evaluate the goodness of a feature or a feature subset and a search algorithm to find the best feature subset from all possible feature subsets according to the evaluation function. Depending on the nature of the evaluation function, the algorithms of feature selection are of two types, filter and wrapper. Filter algorithms evaluate the data set without reference to a particular classifier while wrapper algorithms use classifier accuracy as the evaluation function. The history of pattern recognition is long and early reasearches on feature selection evolved from statistical community. A lot of statistical feature selection algorithms have been proposed so far [3]. However, real world problems are often characterised by vagueness rather than randomness and are difficult to be modelled by rigid framework of mathematics or statistics. Soft computing technologies emerged to bridge this gap and lots of algorithms based on neural computation, fuzzy logic, rough set theory, evolutionary algorithms have been proposed for feature selection and classification in the area of pattern recognition and data mining [4–6]. Evolutionary computational(EC) algorithms are well known tools for solving optimization problems and have been efficiently used for search stage in feature subset selection problem. Among EC based algorithms, Genetic Algorithms (GA) [7], Particle Swarm Optimization (PSO) [8] and Ant Colony Optimization (ACO) [9] are widely used for feature selection. Other less commonly used algorithms are Cuckoo Search (CS) [10], Gravitational Search Algorithm (GSA) [11], Firefly Algorithm (FA) [12], Bat Algorithm(BA) [13] etc.

In this work, a comparative study of evolutionary computation (EC) based feature subset selection algorithms with classification accuracy as a wrapper fitness function (default) has been done. A new penalty based fitness function has been proposed for EC based feature selection. The efficiency of the new fitness function over the default function has been studied by simulation experiments with a number of bench mark data sets from UCI repository. The next section represents a brief overview of the algorithms used for study in this work. The following section describes comparative study of different EC algorithms and the proposal of a new penalty based fitness function followed by the simulation experiments and results in the next section. The final section contains discussion and conclusion.

## 2 EC Based Feature Subset Selection

Evolutionary algorithms are now becoming popular for solving feature subset selection problem. A brief presentation of the algorithms used for feature subset selection in this paper are done in the following subsections.

### 2.1 Genetic Algorithm

Genetic algorithm(GA), a randomized heuristic and adaptive search technique based on the principal of natural selection and the most popular evolutionary approach is a good candidate for solving optimization problems where the search space is large [14, 15]. Several research works for solving feature subset selection problem with GA have been reported. In genetic algorithm, a population of possible solutions i.e the possible candidate feature subsets from a feature set of  $n$  features, encoded as a binary string of  $n$  bits, are maintained through several generations. In each generation, genetic operators such as crossover and mutation are used to generate new population from the most elite pairs of the current generation and the good ones are retained after evaluation by a fitness function. Through the generations, the population is led to the better solution space and finally produces the near optimal solution in the final generation. GA requires no domain knowledge and quite robust than other random or local search methods.

### 2.2 Particle Swarm Optimization

Recently particle swarm optimization (PSO), specially binary particle swarm optimization (BPSO) [16] have been also become popular for feature subset selection [17–19]. Particle Swarm Optimization [8] is a population based evolutionary algorithm. The conventional PSO algorithm begins by initializing a random swarm of  $m$  particles in  $d$  dimensional space characterizing candidate solution like genetic algorithm. However PSO is motivated by simulation of social behaviour instead of survival of fittest and each particle is associated with a velocity. The particles fly through the search space, constantly adjusting their velocity according to corresponding particle's experience and the particle's neighbours' experience. Each particle  $X_i$  makes use of its individual memory and knowledge gained by the swarm as a whole to find the best solution. At each iteration, the fitness of each particle is evaluated by an appropriate fitness function and the algorithm progressively stores and replaces two best values, called *pbest* and *gbest*.  $pbest_i$ , ( $i = 1, 2, \dots, m$ ) denotes the best position associated with the best fitness value achieved so far for each individual and *gbest* denotes the position corresponding to global best value. Binary Particle Swarm optimization (BPSO) algorithm also proposed by Kennedy and Eberhart [16] is an extension of PSO to solve optimization problems with discrete valued parameters. Here each particle (candidate solution) represents a position in a binary multidimensional space, i.e components of  $X_i$  can take only binary values instead of continuous values. The velocity vector associated with each particle is real valued.

### 2.3 Gravitational Search Algorithm

Gravitational Search algorithm (GSA) is a nature inspired heuristic optimization algorithm based on the law of gravity and mass interactions. The algorithm is comprised of collection of agents which interact with each other through the gravity force. The agents are considered as objects and their performances are measured in terms of their masses. The gravity force causes a global movement where all objects move toward other objects with heavier masses. In GSA, the agent has four parameters which are position, inertial mass, active gravitational mass and passive gravitational mass. The position represent the solution of the problem. The gravitational and inertial masses are determined by fitness function. The algorithm is navigated by adjusting gravitational and inertial mass. Finally the position of the heaviest mass presents the optimum solution. The details are found in [11]. A binary version of GSA, known as BGSA is found in [20]

### 2.4 Cuckoo Search

Cuckoo search is an optimization algorithm belonging to the class of swarm intelligence (SI) based algorithms like particle swarm optimization. It is inspired by the obligate interspecific brood parasitism of some cuckoo species that lay their eggs in the nests of other host birds. In this behavior of reproduction, there are two possible cases for a cuckoo egg dumped into a host bird nest including: the host bird does not recognize the cuckoo egg and the cuckoo egg will hatch and carry over to the next generation or the host bird identifies the cuckoo egg and either throw it away or abandon its nest to build a new one. The two mentioned phenomena have been inspired in the CSA method for two phases of new solution generation including the exploration phase via Levy flights (the first phenomenon) and the exploitation phase via replacement of a fraction of eggs (the second phenomenon). The detail algorithm is presented in [10]. A binary version of the algorithm is presented in [21].

### 2.5 Firefly Algorithm

Firefly algorithm (FFA) is also another SI based optimization algorithm inspired by the flashing pattern of tropical fireflies. It is based on three rules: (1) the fireflies are unisex and one is attracted by other irrespective of sex (2) attractiveness is proportional to brightness, less brighter firefly moves to more brighter one, brightness decreases as their distance increases (3) brightness is determined by landscape of the objective function. The objective function of a given optimization problem is based on differences of light intensity. The fireflies are characterized by light intensity which helps to change their position iteratively to more attracting position in order to obtain optimal solution. The details are in [12]. A binary version of the algorithm BFFA is proposed in [22].

## 2.6 Bat Algorithm

Bat algorithm (BA) is a newly proposed SI based metaheuristic optimization algorithm based on echolocation behaviour of bats. Microbats, small bats, use extensive echolocation. They use a type of sonar, to detect prey and to avoid obstacles and locate their resting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Bat algorithm is a modification of particle swarm optimization in which the position and the velocity of virtual microbats are updated based on frequency of their emitted pulses and loudness. The pseudocode of the algorithm and the details can be found in [13]. A binary version of bat algorithm BBA is proposed in [23].

## 3 Comparative Study of Different Algorithms

In this paper a comparative study of different EC algorithms for feature subset problem has been done. The solution space is considered as a binary multidimensional space and represented by a binary string or a binary vector, a point in binary multidimensional space. Binary versions of the algorithms BGA, BPSO, BGSA, BCS, BFFA and BBA are used in this study. The fitness function of the EC algorithm is the classification accuracy of a SVM classifier as a default wrapper fitness function which is defined as:

Fitness function  $S_1 = \text{No. of test samples correctly classified } (T_c) / \text{Total no. of test samples } (T)$ .

### 3.1 Proposal of a New Fitness Function with Penalty

The objective of feature subset selection is two fold: to reduce the dimensionality to lower computational cost as well as to increase the classification accuracy to make the performance higher. But it seems that this two objectives are somewhat contradictory. Reduction of features leads to lower classification accuracy, so use of classification accuracy as the evaluation function of the optimization algorithm is not sufficient for obtaining optimally reduced feature set. So a new fitness function is proposed with the addition of a penalty term in  $S_1$ . The new fitness function  $S_2$  is given by

$$S_2 = S_1 - \alpha \times \frac{D}{N} \quad (1)$$

Where  $D$  and  $N$  represent the number of features in the selected feature subset and total number of features respectively, whereas  $\alpha$  is a control parameter used to adjust the weight of the penalty term in the fitness function. The above fitness function is used in conjunction with various evolutionary algorithms to find the feature subset. The performance of the new fitness function  $S_2$  is compared to the performance of the default fitness function  $S_1$  in terms of the final reduction of feature set and classification accuracy. The performance

of the new fitness function with various EC algorithms (single objective) is also compared with the performance of NSGA II, a popular multiobjective genetic algorithm [24] using classification accuracy and reduction of features in the feature set as two separate objectives.

## 4 Simulation Experiments and Results

Simulation experiments are done with several benchmark data sets from UCI machine learning repository [25]. The EC algorithms used in the simulation experiments are BGA, BPSO, BGSA, BCS, BFFA and BBA. The parameters of different algorithms are set by trial and error experiments so that maximum number of comparison in the search algorithms are 10,000. Table 1 represents the parameters used. For BGA, two point crossover and rank based selection is used.  $P_c$  and  $P_m$  represent probability of crossover and mutation respectively. For other algorithms, relevant parameter values (details are omitted here due to lack of space, can be found in the references) are noted in Table 1.

**Table 1.** Parameters of EC algorithms

	Population size	Maximum epoch	Parameter values
BGA	8	1250	$P_c = 0.1, P_m = 0.05$
BPSO	20	500	$c_1, c_2 = 1, w = 0.5$
BCS	20	500	$\alpha = 0.1, \beta = 1.5, p = 0.25$
BGSA	20	500	$R_p = E_c = 1, min - flag = 0$
BFFA	20	25	$\alpha = 0.25, \beta = 0.2, \gamma = 1$
BBA	20	500	$loudness = 0.25, r = 0.1$
NSGA	20	250	same as BGA

Wine, Cancer and Sonar data sets with number of features 13, 30 and 60, number of classes 3, 2 and 2 and number of data samples 178, 569 and 208 respectively are used for experiment with default fitness function and penalty based fitness function with control parameter  $\alpha = 0.05$  to 0.4 for feature subset selection by different EC algorithms. Finally SVM is used for measuring classification accuracy with the final reduced subset. Different training -test ratio of samples are used for experiments. The evaluation of the fitness function is done by the final number of features in the reduced feature subset and classification accuracy with the final selected subset.

Tables 2, 3 and 4 represent the effectiveness of the proposed fitness function with penalty (for  $\alpha = 0.15$ ) for wine, cancer and sonar data set respectively in terms of average classification accuracy and average number of features in the final selected feature subset. For all the data sets we found that control parameter  $\alpha = 0.15$  produces the best result.

**Table 2.** Simulation results for Wine data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
BGA	0.95224	7.87	0.93374	3.84
BPSO	0.96009	8.72	0.93915	3.61
BCS	0.94514	7.89	0.9286	4.74
BFFA	0.94561	6.24	0.93444	3.01
BBA	0.95999	8.42	0.93813	3.95
BGSA	0.95421	8.43	0.92869	4.69

**Table 3.** Simulation results for Cancer data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
BGA	0.94933	14.77	0.94476	7.55
BPSO	0.96632	13.45	0.96059	4.16
BCS	0.94965	13.05	0.94004	8.82
BFFA	0.96061	8.37	0.95193	3.71
BBA	0.96488	13.27	0.95658	4.43
BGSA	0.95173	13.51	0.94412	8.64

**Table 4.** Simulation results for Sonar data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
BGA	0.85392	30.16	0.94476	24.65
BPSO	0.88224	30.51	0.96059	17.87
BCS	0.82344	30.48	0.94004	21.23
BFFA	0.83592	18.82	0.95193	10.56
BBA	0.88808	29.88	0.95658	16.11
BGSA	0.84312	31.62	0.94412	22.0

In all the cases, new fitness function produce final feature subset with lesser number of features without much degradation in classification accuracy. Also it is found that BFFA produced the highest reduction in feature set. For comparison with multiobjective algorithm, we used NSGA II with two objective functions classification accuracy and number of features in the selected feature subset. The classification accuracy and number of features in the final selected subset for wine, cancer and sonar data set came out to be 0.95, 0.95, 0.85 and 8.07, 14.65 and 29.5 respectively. It seems that our proposed penalty based single

objective fitness function is better in efficiency than NSGAI in terms of reducing feature number in the final feature subset without having much degradation in classification accuracy.

We repeated the experiment with three other high dimensional datasets, Hill, Gas and Madelon from UCI repository with number of features 100, 128 and 500 respectively. Tables 5, 6 and 7 represent the simulation results. Here we used only three EC algorithms GA, BFFA and BBA as those are found to be the effective algorithms for feature subset selection.

**Table 5.** Simulation results for Hill data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
GA	0.95987	54.18	0.95234	45.7
BFFA	0.93545	47.7	0.92666	34.66
BBA	0.97677	55.36	0.97059	39.5

**Table 6.** Simulation results for Gas data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
GA	0.97148	68.7	0.96869	58.16
BFFA	0.98916	18.8	0.98697	12.4
BBA	0.98243	68.06	0.97894	44.26

**Table 7.** Simulation results for Madelon data

Algorithm	Default fitness		Fitness with penalty	
	Accuracy	Av. no. of features	Accuracy	Av. no. of features
GA	0.65150	250.76	0.59433	246.3
BFFA	0.62129	125.32	0.59141	73.31
BBA	0.67824	249.72	0.62487	242.38

For high dimensional data also, it is found that the penalty based fitness function works better than the default fitness function in reducing the number of features in optimal feature subset, though the classification accuracy falls drastically for too high dimensional data after reducing the dimension to 50% (Madelon data set). Here also we found that BFFA is the most effective EC algorithm. NSGA II is also used for the high dimensional data sets and it is found that the result is similar to single objective GA and worse than using penalty based fitness function.



## 5 Conclusion

Optimal feature subset selection is an extremely important preprocessing step for any pattern recognition or machine learning problem. The successful elimination of redundant and irrelevant information increases the performance of the classifier while retaining important and informative feature is highly needed for improved performance. For high dimensional data, the judicious selection of feature from available features becomes more important as reduction of feature having relevance to the class reduces classification accuracy while retaining all the features heavily increases the computational cost. So feature evaluation function should be carefully designed. In search algorithm based optimal feature subset selection, feature evaluation function is used as the fitness function of the search algorithm. For wrapper method, classification accuracy itself is generally used as the default fitness function of the search algorithm.

In this work various evolutionary algorithm has been used for searching the optimal feature subset from a set of features with classification accuracy as the default fitness function and a newly proposed fitness function in which a penalty term for high number of features in the selected subset is added to classification accuracy. The new penalty function seems to be very effective in reducing the number of features in the selected subset without much degradation in classification accuracy. The proposed fitness function also seems to be effective compared to multiobjective genetic algorithm. Among the EC algorithms, BFFA produced the best result in terms of reduction of number of features. Simulation experiments with high dimensional data also show that the algorithms are quite effective for data sets of dimension 100 to 200. Further experiments are now carried on for modifications of the fitness function for filter type to be used with high dimensional data having dimension in the range of 1000.

## References

1. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall International, London (1982)
2. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Springer, Heidelberg (1998)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience, New York (2001)
4. Verikas, A., Bacauskiene, M.: Feature selection with neural networks. Pattern Recogn. Lett. **23**(11), 1323–1335 (2002)
5. Oh, I., Lee, J., Moon, B.: Hybrid genetic algorithm for feature selection. IEEE Trans. PAMI **26**(11), 1424–1437 (2004)
6. Chakraborty, B.: Genetic algorithm with fuzzy operators for feature subset selection. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E85–A**(9), 2089–2092 (2002)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston (1989)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference On Neural Network, vol. 4, pp. 1942–1948 (1995)

9. Dorigo, M., Stuzle, T.: *Ant Colony Optimization*. The MIT Press, Cambridge (2004)
10. Yang, X.S., Deb, S.: Cuckoo search via levy flights. In: *Proceedings of NaBIC 2009*, pp. 210–214 (2009)
11. Reshedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)
12. Yang, X.S.: *Nature Inspired Metaheuristic Algorithms*. Luniver Press, Bristol (2008)
13. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Gonzalez, J.R. (ed.) *NISCO 2010. SCI*, vol. 284, pp. 65–74. Springer, Heidelberg (2010)
14. Chakraborty, B.: Evolutionary computational approaches to feature subset selection. *Int. J. Soft Comput. Bioinform.* **1**(2), 59–65 (2010)
15. Chakraborty, B.: Genetic algorithm with fuzzy fitness function. In: *Proceedings of IEEE International Symposium on Industrial Electronics (IEEE-ISIE 2002)*, pp. 315–319 (2002)
16. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: *Proceedings of IEEE International Conference on Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108 (1997)
17. Chakraborty, B.: Feature subset selection by particle swarm optimization with fuzzy fitness function. In: *Proceedings of ISKE 2008, 3rd International Conference on Intelligent Systems and Knowledge Engineering*, pp. 1038–1042 (2008)
18. Chakraborty, B.: Binary particle swarm optimization based algorithm for feature subset selection. In: *Proceedings of ICAPR 2009*, pp. 145–148 (2009)
19. Chakraborty, B.: Feature subset selection using hybrid particle swarm optimization. *ICIC Express Lett.* **6**(5), 1161–1168 (2012)
20. Reshedi, E., Nezamabadi-pour, H., Saryazdi, S.: BGSA: binary gravitational search algorithm. *Nat. Comput.* **9**(3), 727–745 (2009)
21. Rodrigues, D., Pereira, L.A.M., Souza, N., Ramos, C.C.O., Yang, X.S.: BCS: a Binary Cuckoo Search algorithm for feature selection. In: *2013 IEEE International Symposium on Circuits and Systems*, pp. 465–468 (2013)
22. Crawford, B., et al.: A binary coded firefly algorithm that solves the set covering problem. *Rom. J. Inform. Sci. Technol.* **17**(3), 252–264 (2014)
23. Nakamura, Y.M., Pereira, L.A.M., Costa, K.A., Rodrigues, D., Papa, J.P., Yang, X.S.: Binary bat algorithm. *Neural Comput. Appl.* **25**(3), 663–668 (2014)
24. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolu. Comput.* **2**(3), 182–197 (2002)
25. UCI Machine learning Repository. <http://archive.ics.uci.edu/ml/>