Global Best Artificial Bee Colony for Minimal Test Cost Attribute Reduction

Anjing Fan, Hong Zhao*, and William Zhu

Lab of Granular Computing,
Minnan Normal University, Zhangzhou 363000, China
hongzhaocn@163.com

Abstract. The minimal test cost attribute reduction is an important component in data mining applications, and plays a key role in cost-sensitive learning. Recently, several algorithms are proposed to address this problem, and can get acceptable results in most cases. However, the effectiveness of the algorithms for large datasets are often unacceptable. In this paper, we propose a global best artificial bee colony algorithm with an improved solution search equation for minimizing the test cost of attribute reduction. The solution search equation introduces a parameter associated with the current global optimal solution to enhance the local search ability. We apply our algorithm to four UCI datasets. The result reveals that the improvement of our algorithm tends to be obvious on most datasets tested. Specifically, the algorithm is effective on large dataset Mushroom. In addition, compared to the information gain-based reduction algorithm and the ant colony optimization algorithm, the results demonstrate that our algorithm has more effectiveness, and is thus more practical.

Keywords: Cost-sensitive learning, Minimal test cost, Attribute reduction, Granular computing, Biologically-inspired algorithm.

1 Introduction

Cost-sensitive learning is one of the most active and important research areas in machine learning and data mining. In conventional data mining, attribute reduct tries to maximize the accuracy or minimize the error rate in general. In real-world applications, one should pay cost for obtaining a data item of an attribute. It is important to take the test cost account into attribute reduct [1,2]. The minimal test cost attribute reduction [9] is an important problem in cost-sensitive learning. This problem is not a simple extension of existing attribute reduction problems, it is a mandatory stage in dealing with the test cost issue. The problem is a task to select an attribute subset with minimal test cost. The performance of the minimal test cost attribute reduction is the test cost, which is independent of the performance of attribute reduction.

In the recent years, some algorithms are proposed to deal with the minimal test cost attribute reduction problem, such as ant colony optimization algorithm (ACO) [3] and information gain-based λ-weighted reduction (λ-weighted) algorithm [4]. However, the

* Corresponding author.

D. Miao et al. (Eds.): RSKT 2014, LNAI 8818, pp. 101–110, 2014.
DOI: 10.1007/978-3-319-11740-9_10

effectiveness of these algorithms for large datasets is often needed to improve. To deal with this problem, artificial bee colony (ABC) algorithm be considered. The ABC algorithm is a biologically-inspired optimization algorithm, it is able to produce high quality solutions with fast convergence. Due to its simplicity and easy implementation, the ABC algorithm has captured much attention and has been applied to solve many practical optimization problems [14].

In this paper, we propose a global best artificial bee colony (GABC) algorithm for the minimal test cost attribute reduction problem. The GABC algorithm is inspired by the ABC algorithm. The ABC algorithm [5] is proposed to optimize continuous functions. Although it has fewer control parameters, it shows competitive performance compared with other population-based algorithms. However the algorithm cannot be effective using the individual information to optimize search method, so the traditional artificial bee colony algorithm is good at exploration but poor at exploitation. As we know, the exploitation is determined by the solution search equation. In this paper, the GABC algorithm improves the solution to balance the exploration and exploitation ability of the ABC algorithm. The GABC algorithm induces a parameter Lb into the improved solution search equation. The parameter Lb value is mainly composed of the fitness of the global optimal solution.

We evaluate the performance of our algorithm on four UCI (University of California Irvine) datasets [6,7], which serve the machine learning community. Since there is no cost settings for attribute on the four datasets, we use Normal distribution to generate test cost for datasets. The viability and effectiveness of the GABC algorithm are tested on four datasets. The results demonstrate the good performance of the GABC algorithm in solving the minimal test cost attribute reduction problem when compared with the λ-weighted algorithm, ACO algorithm and ABC algorithms. Experiments are undertaken by an open source software called Coser (cost-sensitive rough sets) [8].

The rest of the paper is organized as follows. Section 2 presents attribute reduction in cost-sensitive learning and discusses the problem of the minimal test cost attribute reduction. Section 3 analyzes the parameters of the GABC algorithm for getting optimal. Section 4 presents the experimental results and the comparison results. Finally, conclusions and recommendations for future studies are drawn in Section 5.

2 Preliminaries

In this section, we present some basic notions for the minimal test cost attribute reduction problem. The one conveyed is the test cost independent decision system, the other we proposed is the minimal test cost attribute reduction problem.

2.1 Test Cost Independent Decision System

In this paper, datasets are fundamental for the minimal test cost attribute reduction problem. We consider the datasets with a test cost independent decision system [4]. A test-cost-sensitive decision system is defined as follows.

Definition 1. *[4] A test cost independent decision system (TCI-DS) S is the 6-tuple:*

$$S = (U, C, d, \{V_a | a \in C \cup \{d\}\}, \{I_a | a \in C \cup \{d\}\}, tc), \tag{1}$$

where U is a finite set of objects called the universe, C is the set of attributes, d is the decision class, V_a is the set of values for each $a \in C \cup \{d\}$, $I_a : U \to V_a$ is an information function for each $a \in C \cup \{d\}$, and $tc : C \to R^+ \cup \{0\}$ is the test cost function for each $a \in C$.

Here test costs are independent of one another. A test cost function can be represented by a vector $tc = [tc(a_1), tc(a_2), ..., tc(a_{|C|})]$. It is easy to calculate the test cost for an attribute subset B (any $B \subseteq C$), which is counted as follows: $tc(B) = \sum_{a \in B} tc(a)$.

Table 1. A clinical decision system

Patient	Headache	Temperature	Lymphocyte	Leukocyte	Eosinophil	Heartbeat	Flu
x_1	yes	high	high	high	high	normal	yes
x_2	yes	high	normal	high	high	abnormal	yes
x_3	yes	high	high	high	normal	abnormal	no
x_4	no	high	normal	normal	high	normal	no

An exemplary decision system is given by Table 1. The attributes of this decision system are symbolic. Here $C = \{$Headache, Temperature, Lymphocyte, Leukocyte, Eosinophil, Heartbeat $\}$, $\{d\} = \{$Flu$\}$, $U = \{x_1, x_2, x_3, x_4\}$, and the corresponding test cost of attributes is represented by a vector $tc = [12, 5, 15, 20, 15, 10]$.

2.2 The Minimal Test Cost Attribute Reduction Problem

Attribute reduction plays an important role in rough sets [11]. We review the reduction based on positive region [12].

Definition 2. *[13] Any $B \subseteq C$ is called a decision relative reduction (or a reduction for brevity) of S if and only if:*
1.$POS_B(\{d\}) = POS_C(\{d\})$;
2.$\forall a \in B, POS_{B-\{a\}}(\{d\}) \neq POS_C(\{d\})$.

In applications, a number of reductions sometimes are needed. However, in most applications, only one reduction is needed. Since there may exist many reductions, an optimization metric is needed. In this paper, the test cost is taken into account in attribute reduction problem. Naturally, the test cost of the attribute reduction is employed as a metric in our work. In other words, we are interested in the attribute reduction with minimal test cost. We define reductions of this type as follows.

Definition 3. *[13] Let S be a TCI-DS and Red (S) be the set of all reductions of S. Any $R \in Red(S)$ where $tc(R) = min\{tc(R^{'}) \mid R \in Red(S)\}$ is called a minimal test cost attribute reduction.*

As indicated in Definition 3, the set of all minimal test cost attribute reductions is denoted by $MTR(S)$. The optimal objective of our paper is MTR problem.

3 Algorithm

This section introduces the global best artificial bee colony (GABC) algorithm in detail. Similar to the artificial bee colony (ABC) [14] algorithm, our algorithm consists food sources and three groups of bees: employed bees, onlookers and scouts. The ABC algorithm [15] is composed of two main steps: recruit an optimal good source and abandon a bad source. The process of artificial bees seeking good food sources equal the process of finding the minimal test cost attribute reduction.

In GABC algorithm, let one employed bee is on one food source and the number of employed bees or onlookers equal the number of food sources. The position of a food source represents an attribute subset and it is exploited by one employed bee or one onlookers. The number of food sources is set to 1.5 times number of attributes. Employed bees search new foods and remember the food source in their memory, and then pass the food information to onlookers. The onlookers tend to select good food sources from those foods founded by the employed bees, then further search the foods around the selected food source. The scouts are translated from a few employed bees, which abandon their food sources and search new ones.

As well known that both exploration and exploitation are necessary for the ABC algorithm. In the algorithm, the exploration refers to the ability to investigate the various unknown regions in the solution space to discover the global optimum. While the exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions. In practice, to achieve good optimization performance, the two abilities should be well balanced. As we know, a new candidate solution is given by the following solution search equation in the artificial bee colony algorithm:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}). \tag{2}$$

In Equation (2), we can know that the coefficient ϕ_{ij} is an uniform random number in [0, 1] and x_{kj} is a random individual in the population, therefore, the solution search dominated by Equation (2) is random enough for exploration. However, alternatively the new candidate solution is generated by moving the old solution towards another solution selected randomly from the population. That is to say, the probability that the randomly selected solution is a good solution is the same as that the randomly selected solution is a bad one, so the new candidate solution is not promising to be a solution better than the previous one. To sum up, the solution search equation described by Equation (2) is good at exploration but poor at exploitation.

By taking advantage of the information of the global best solution to guide the search of candidate solutions, we rebuild the ABC algorithm to improve the exploitation. The GABC algorithm as follows.

Step 1. Create an initial food source position, and calculate the fitness value of the food source.

Food sources initialization is a crucial task in the ABC algorithm because it can affect the convergence speed and the quality of finding optimal solution. We replace the random select attribute subset with an attribute subset containing core attribute and satisfying the position region constraint [16].

The fitness value of the food source is defined as the reciprocal of the corresponding test cost. The fitness equation as follows:

$$fitness = \frac{1}{1+tc}, \tag{3}$$

where tc is the test cost of an attribute subset selected.

After initialization, the GABC algorithm enters a loop of operations: updating feasible solutions by employed bees, selecting feasible solutions by onlooker bees, and avoiding suboptimal solutions by scout bees.

Step 2. Produce new solution v_{ij} for the employed bees by Equation (5) and evaluate it by Equation (3).

The best solution in the current population is a very useful source which can be used to improve the convergence speed. We introduce a parameter Lb that associates with the current global optimal solution. The equation of Lb is conveyed as follows:

$$Lb = fitness_i / global fitness, \tag{4}$$

where fitness $_i$ is the i-th iteration fitness of food source, and global fitness stand for the fitness of current global optimal food source. As can be seen from Equation (4), Lb is a positive real number, typically less than 1.0.

Through the analysis, we propose a new solution search equation as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + Lb(g_i - x_{ij}), \tag{5}$$

where $k \in \{1, 2, 3, ..., SN\}$ and $j \in \{1, 2, 3, ..., D\}$ are randomly chosen indexes. k is different from i. SN is the number of the attribute, D is the number of the food source. ϕ_{ij} is a random value in [0, 1]. v_{ij} and x_{kj} is a new feasible solution that is modified from its previous solution x_{ij}, g_i is the best solution that explored in the history used to direct the movement of the current population.

When Lb takes 0, Equation (5) is identical to Equation (2). We can get a new solution better than the old one, then turn the new solution to be an old one in the next iteration. Apply the greedy selection process for the employed bees.

Step 3. Calculate the probability values P_i for the solution v_{ij} by Equation (6).

Produce the new solution u_{ij} for onlooker bee by Equation (5), and evaluate it by Equation (3). Where u_{ij} is produced from the solutions v_{ij} depending on P_i. An onlooker bee chooses a food source depending on the probability values P_i associated with that food source.

The equation of calculating the probability values P_i is shown as follows:

$$P_i = \frac{fitness_i}{\sum_{n=1}^{SN} fitness_n}, \tag{6}$$

where fitness$_i$ is the fitness value of the i-th solution, SN is the food number.

Apply the greedy selection process for the onlookers.

Step 4. When a food source can not improve further through limit cycles, the food source is abandoned for a scout bee. The food source is replaced with a new randomly solution produced by Equation (5).

The limit is an important control parameter of the GABC algorithm for abandonment. This step avoid the algorithm falling into suboptimal solutions. The Steps 2, 3 and 4 are repeated until the running generation reaches the maximal number of iteration.

The GABC algorithm deletes redundant attributes of each food source in inverted order with the positive region constraint. Through the above steps, an attribute reduction with minimal test cost has been produced, it is the final solution.

4 Experiments

To test the performance of the GABC algorithm, an extensive experimental evaluation and comparison with the ABC, the λ-weighted [4] and the ACO [3] algorithms are provided based on four datasets as follows. The four datasets are shown in Table 2. The finding optimal factor (FOF) [4] is used as comparison criteria in this paper.

4.1 Data Settings

In our experiments, there are four UCI datasets used to test. These are Zoo, Voting, Tic-tac-toe and Mushroom. The information of the four datasets is summarized in Table 2. On the four datasets, attributes are no test cost settings, so we apply Normal distribution to generate random test cost in [1, 10].

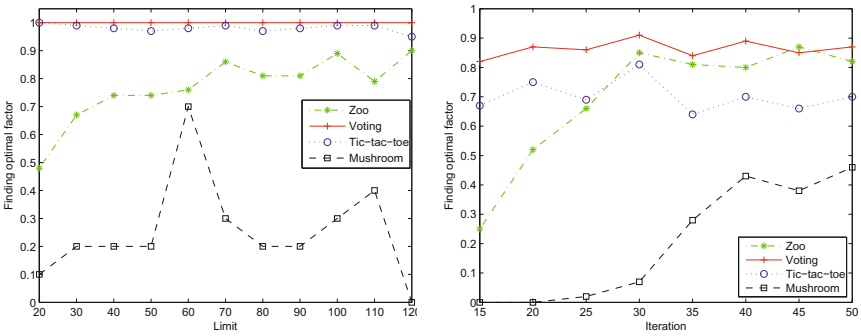
Table 2. Database information

Name	Domain	$\|U\|$	$\|C\|$	$D = \{d\}$
Zoo	Zoology	101	16	Type
Voting	Society	435	16	Vote
Tic-tac-toe	Game	958	9	Class
Mushroom	Botany	8124	22	Class

4.2 Experiment Results

In experiment, each algorithm is undertaken with 100 different test cost settings on four datasets. The experiments reveal the performance of the GABC algorithm through analyzing parameters: limit, iteration and Lb. Nextly, we investigate the impact of the three parameters on the GABC algorithm.

Figure 1 presents solutions along iterations and limits for the four datasets. It can be observed that the evolution curves of the GABC algorithm reach higher FOF much faster. Thus, it can be concluded that overall the GABC algorithm outperforms well. It can be found from Figure 1, the FOF of a large limit(i.e., 60 or 80) is superior to the FOF of a small limit (i.e., 20 or 30) on most datasets, this rule also applies to the iteration.



(a) (b)

Fig. 1. Finding optimal factor on four datasets: (a) Limit, (b) Iteration

In Figure 1, we investigate the impact of parameter of limit and iterations on the GABC algorithm.

1) When the maximal iteration is set to 300, we let the parameter Lb be 0.8. As can be seen, we obtain better value of limit on the Mushroom dataset when limit is 60. For the other three test datasets, better results are obtained when limit is 110.

2) When limit is set to 110 and the parameter Lb is kept in 0.8. We can obtain that better value of iteration on the Mushroom dataset is 40. For the other three test datasets, better results are obtained when iteration is 30. The performance on parameter iteration is likely sensitive to the number of attributes.

3) As can be seen, when the value of limit is increased, the FOF is also improved. This trend also applies to the parameter of iteration. Figure 1(b) shows the parameter of iteration is needed to converge towards the optimal solution for the GABC algorithm, which same to the parameter of limit. We observe that the values of limit and iteration can greatly influence the experimental results.

In order to reveal the impact of control parameter Lb, we conduct experiments for our algorithm, where Lb in [0, 1] with 0.2 stepsize and use the competition approach [4] to improve the results. In Figure 2, when the value of Lb is set to 1, we can obtain a good result on Mushroom dataset. For the other three test datasets, better results are obtained when Lb is around 0.8. As can be seen, when the values of Lb are increased, the values of FOF are also improved. Therefore, the selective Lb is set at 0.8 for all the datasets tested. We can observe that the values of Lb also have effect on the results.

This can be explained by the basic principle of the ABC algorithm. The parameter Lb in Equation (5) plays an important role in balancing the exploration and exploitation of the candidate solution search. When Lb increases from zero to a certain value, the exploitation of Equation (5) will also increase correspondingly.

4.3 Comparison Results

In the following, we illustrate the advantage of the GABC algorithm compared with the λ-weighted algorithm, the ACO algorithm and the ABC algorithm. The limit of the GABC and ABC algorithms is set to 100, and the iteration is 40.

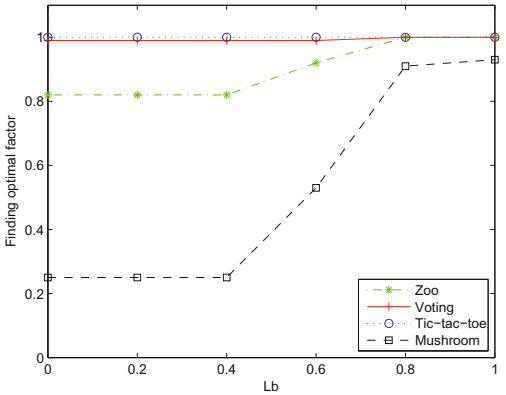


Fig. 2. Finding optimal factor for Lb value on four datasets

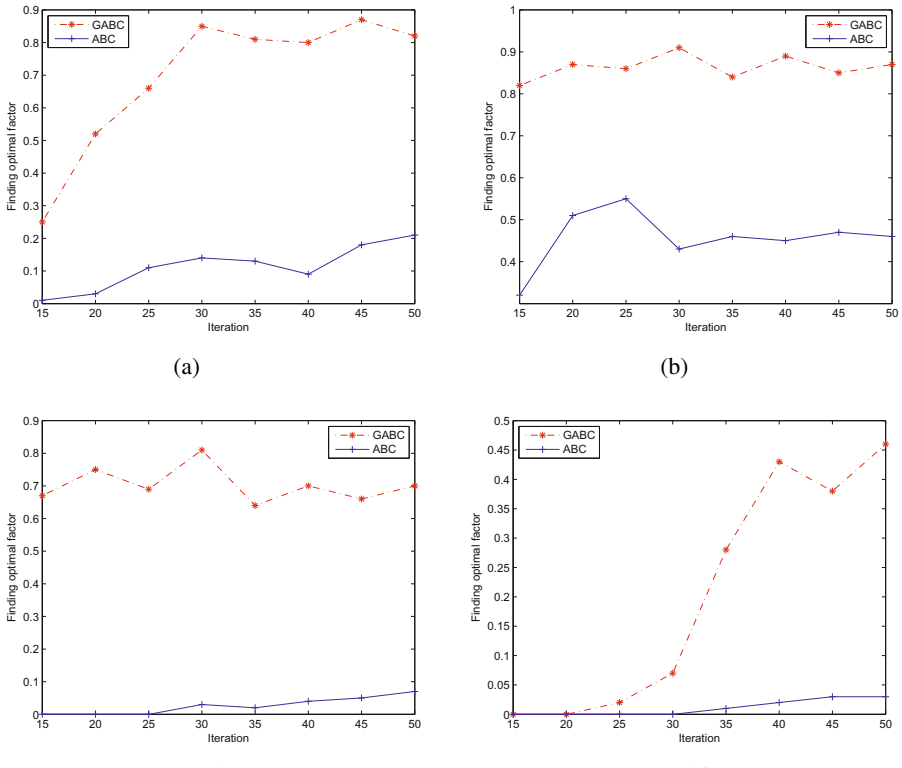


Fig. 3. Finding optimal factor for iteration on four datasets: (a) Zoo, (b) Voting, (c) Tic-tac-toe, (d) Mushroom

Table 3. Finding optimal factor of three algorithms with the competition approach

Datasets	λ-weighted	ACO	GABC
Zoo	0.833	0.987	**1.000**
Voting	**1.000**	**1.000**	**1.000**
Tic-tac-toe	0.408	**1.000**	**1.000**
Mushroom	0.176	0.958	**0.970**

Figure 3 presents the FOF for iteration on the four different datasets. Table 3 draws the FOF for three algorithms on the four different datasets by competition approach. The best results are marked in bold in table. The results in Table 3 and Figure 3 further demonstrate that the GABC algorithm is a great algorithm since it generates significantly better results than the λ-weighted algorithm, ACO algorithm and ABC algorithm for datasets.

The results show that the FOF of the other two algorithms produced are acceptable results on most datasets. However, the performances of the two algorithms are shortly on Mushroom dataset. The results of the λ-weighted algorithm are especially obvious. For example, the FOF is only 17.6% of the λ-weighted algorithm on the Mushroom dataset. However, it is 97% of our algorithm on the Mushroom dataset.

In summary, the GABC algorithm can produce an optimal reduction in general. The algorithm has the highest performance among the three algorithms for all four datasets.

5 Conclusions

In this paper, we have developed the global based artificial bee colony algorithm to cope with the minimal test cost attribute reduction problem. The algorithm has been improved by introducing the parameter Lb based on global optimal. We have demonstrated the effectiveness of the GABC algorithm and provided comparisons with two other algorithms. The results have shown that the GABC algorithm possesses superior performance in finding optimal solution as compared to the other algorithms. In the future, we will improve the stability and efficiency of the global based artificial bee colony algorithm.

Acknowledgments. This work is in part supported by the Zhangzhou Municipal Natural Science Foundation under Grant No. ZZ2013J03, the Key Project of Education Department of Fujian Province under Grant No. JA13192, the National Science Foundation of China under Grant Nos. 61379049 and 61379089, and the Postgraduate Research Innovation Project of Minnan Normal University under Grant No. YJS201438.

References

1. Fumera, G., Roli, F.: Cost-sensitive learning in support vector machines. In: Proceedings of VIII Convegno Associazione Italiana per L' Intelligenza Artificiale (2002)
2. Ling, C.X., Yang, Q., Wang, J.N., Zhang, S.C.: Decision trees with minimal costs. In: Proceedings of the 21st International Conference on Machine Learning, p. 69 (2004)

3. Xu, Z., Min, F., Liu, J., Zhu, W.: Ant colony optimization to minimal test cost reduction. In: 2012 IEEE International Conference on Granular Computing (GrC), pp. 585–590. IEEE (2012)
4. Min, F., He, H.P., Qian, Y.H., Zhu, W.: Test-cost-sensitive attribute reduction. Information Sciences 181, 4928–4942 (2011)
5. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214(1), 108–132 (2009)
6. Johnson, N., Kotz, S.: Continuous distributions. J. Wiley, New York (1970) ISBN: 0-471-44626-2
7. Johnson, R., Wichern, D.: Applied multivariate statistical analysis, vol. 4. Prentice Hall, Englewood Cliffs (1992)
8. Min, F., Zhu, W., Zhao, H., Xu, Z.L.: Coser: Cost-senstive rough sets (2012), http://grc.fjzs.edu.cn/~{}fmin/coser/
9. Min, F., Zhu, W.: Minimal cost attribute reduction through backtracking. In: Kim, T.-H., et al. (eds.) DTA/BSBT 2011. CCIS, vol. 258, pp. 100–107. Springer, Heidelberg (2011)
10. Susmaga, R.: Computation of minimal cost reducts. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 448–456. Springer, Heidelberg (1999)
11. Zhu, W.: A class of fuzzy rough sets based on coverings. In: Proceedings of Fuzzy Systems and Knowledge Discovery, vol. 5, pp. 7–11 (2007)
12. Pawlak, Z.: Rough sets and intelligent data analysis. Information Sciences 147(12), 1–12 (2002)
13. Min, F., Zhu, W.: Attribute reduction of data with error ranges and test costs. Information Sciences 211, 48–67 (2012)
14. Gao, W., Liu, S.: Improved artificial bee colony algorithm for global optimization. Information Processing Letters 111(17), 871–882 (2011)
15. Banharnsakun, A., Achalakul, T., Sirinaovakul, B.: The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing 11(2), 2888–2901 (2011)
16. Cai, J., Ding, H., Zhu, W., Zhu, X.: Artificial bee colony algorithm to minimal time cost reduction. J. Comput. Inf. Systems 9(21), 8725–8734 (2013)