

## Research Article

# A Novel Strategy for Minimum Attribute Reduction Based on Rough Set Theory and Fish Swarm Algorithm

Yuebin Su<sup>1,2</sup> and Jin Guo<sup>1</sup>

<sup>1</sup>*School of Information Science and Technology, Southwest Jiao Tong University, Chengdu 610031, China*

<sup>2</sup>*School of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong 643000, China*

Correspondence should be addressed to Yuebin Su; [mathsyb@163.com](mailto:mathsyb@163.com)

Received 28 March 2017; Revised 25 June 2017; Accepted 5 July 2017; Published 15 August 2017

Academic Editor: Naveed Ejaz

Copyright © 2017 Yuebin Su and Jin Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For data mining, reducing the unnecessary redundant attributes which was known as attribute reduction (AR), in particular, reduces with minimal cardinality, is an important preprocessing step. In the paper, by a coding method of combination subset of attributes set, a novel search strategy for minimal attribute reduction based on rough set theory (RST) and fish swarm algorithm (FSA) is proposed. The method identifies the core attributes by discernibility matrix firstly and all the subsets of noncore attribute sets with the same cardinality were encoded into integers as the individuals of FSA. Then, the evolutionary direction of the individual is limited to a certain extent by the coding method. The fitness function of an individual is defined based on the attribute dependency of RST, and FSA was used to find the optimal set of reducts. In each loop, if the maximum attribute dependency and the attribute dependency of condition attribute set are equal, then the algorithm terminates, otherwise adding a single attribute to the next loop. Some well-known datasets from UCI were selected to verify this method. The experimental results show that the proposed method searches the minimal attribute reduction set effectively and it has the excellent global search ability.

## 1. Introduction

Data mining, which was known as knowledge discovery in database, includes extracting knowledge, discovering new patterns, and predicting the future trends from the amounts of data. Nowadays, with an increasing number of applications in different fields, massive volumes of very high-dimensional data were produced; the data mining faces the great challenge. As known to all, much of datasets contain unnecessary redundant attributes, which not only occupy extensive computing resources but also seriously impact the decision-making process. Reducing the unnecessary redundant attributes becomes very necessary for data mining [1]. Attribute reduction (AR) in the rough set theory (RST) removes redundant or insignificant knowledge with keeping the classification ability of the information system the same as before. It was proposed by Pawlak and Slowinski [2]. Now, RST is widely used in many fields such as machine learning, data mining, and knowledge discovery [3–6].

AR is one of the core problems in RST. In particular, minimal reduction problem is an important part of AR in RST, in which the cardinality of attribute subset is the smallest among all possible reductions. It has been paid much attention by many researchers. One basic solution to find the minimal reducts is to construct a discernibility function and simplify it from the dataset by discernibility matrix [7–9]. Unfortunately, it has been shown that the problem of minimal reduct generation is NP-hard and the run time of generating all reducts is exponential [10]. Recently, because many kinds of NP-hard problems can be solved by heuristic algorithms with increasing computational cost, heuristic attribute reduction algorithm is the main research direction in the field of AR [11].

In general, swarm intelligence algorithm is one kind of heuristic approaches which were used widely for solving attribute reduction problem, including genetic algorithm (GA) [12–14], particle swarm optimization (PSO) [15–18], ant colony optimization (ACO) [19, 20], and fish swarm algorithm (FSA) [11, 21, 22]. FSA is a kind of evolutionary algorithm

which was inspired by the natural schooling behaviors of fish to generate candidate solutions for optimization problems, such as random, swarming, following, and preying behaviors. It has a strong ability to avoid local minimums in order to achieve a global optimization [23]. Due to its abilities to perform, FSA has received much attention in recent years.

In this paper, a new coding method about the subset of attribute sets is proposed. By the coding method, a novel strategy for minimal attribute reduction algorithm based on FSA and RST is proposed. It firstly identifies the core attributes by discernibility matrix. Based on the core attributes, all subsets without containing the core attribute are encoded into an integer by the proposed coding method and an initial population is generated for FSA used to find the optimal set of reducts. The fitness function of a subset is defined based on the attribute dependency of the formed rough set. In each loop, the evolutionary direction of the individual is limited to a certain extent by the coding method. If the maximum attribute dependency and the attribute dependency of condition attribute set are equal, then the algorithm terminates, otherwise, adding a single attribute to the next loop. Different benchmark datasets are used to compare the numerical results; our proposed method is a robust and cheap method for calling the fitness function.

The rest of the paper is organized as follows. In Section 2, we introduced some basic concepts in rough sets and fish swarm algorithm. In Section 3, we focus the coding method of combination set. In Section 4, a novel attribute reduction algorithm based on fish swarm algorithm and rough set is proposed. In Section 5, some well-known datasets are used to test the performance of the proposed method. Finally, Section 6 concludes the paper and the areas of further research.

## 2. Background

*2.1. Base Notions of Rough Set Theory.* In this section, some basic notions and its proposition will be reviewed in the theory of rough set.

A decision table can be represented as  $S = \{U, A, V, f\}$ , where  $U = \{x_1, x_2, \dots, x_n\}$  is a nonempty finite set of objects,  $A = C \cup D$ , where  $C$  is a set of condition attribute and  $D$  is a decision attribute set,  $V$  is the domains of attributes belonging to  $A$ , and  $f : U \times A \rightarrow V$  is a function assigning attribute values to objects in  $U$ .

For any  $R \subseteq C \cup D$ , there is an associated indiscernibility relation  $\text{IND}(R)$ :

$$\begin{aligned} \text{IND}(R) &= \{(x, y) \mid f(x, a) = f(y, a), \forall a \in P, x, y \in U\}. \end{aligned} \quad (1)$$

Let  $X \subseteq U$ ; the  $R$ -lower approximation of  $X$  is defined as  $\underline{R}X = \{x \in U \mid [x]_R \subseteq X\}$  where  $[x]_R$  denotes an equivalence class of  $\text{IND}(R)$  determined by object  $x$ . The notation  $\text{POS}_R(D)$  refers to the  $R$ -positive region is given by  $\text{POS}_R(D) = \bigcup_{X \in U/\text{IND}(D)} \underline{R}X$ . The  $R$ -approximation quality

with respect to decisions attribute set  $D$  is defined as follows:

$$\gamma_R = \frac{|\text{POS}_R(D)|}{|U|} \quad (2)$$

and the core attribute set is defined as

$$\text{Core} = \{a : \gamma_{R \setminus \{a\}} \neq \gamma_R\}. \quad (3)$$

*2.2. The Principle of FSA.* FSA is a new bionic optimization algorithm which simulates the fish swarm behaviors such as preying, swarming, and following behaviors and updates the maximum fitness value on the bulletin board. In FSA, let  $N$  be the population size, the Artificial Fishes (AF) are generated by random function which is represented by a  $D$ -dimensional position  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , and  $X_{i|\text{next}}$  is the updated value of  $X_i$ . Food satisfaction of  $X_i$  is represented as fitness function value  $Y_i = F(X_i)$ . The Euclidean distance  $d_{ij} = \|X_i - X_j\|$  is denoted as the relationship between  $X_i$  and  $X_j$ . Other parameters include step (representing maximum step length), visual (the visual distances of fish), rand being a random number in  $[0, 1]$ , and  $\sigma$  (a crowd factor).

Preying behavior is a basic behavior of FSA. As shown in (4), for  $X_i$ , we randomly select a random  $X_j$  within the current visual scope. If  $Y_i < Y_j$ , then move a step from  $X_i$  to  $X_j$ . Otherwise,  $X_i$  move a step to another random  $X_j$  that  $Y_i < Y_j$ . After a number of trials, if the random  $X_j$  that meets  $Y_i < Y_j$  is not satisfied,  $X_i$  will be replaced with a random position within the visual scope directly. It makes the FSA escape from the local optimal solution. Define the prey( $X_i$ ) function as (4).

Swarming behavior is described as (5). It shows the attraction of the swarm center to the individual. Let  $n_f$  be the number of AFs within the current visual scope of  $X_i$ , and  $X_i^c$  is the center position of those neighbors. For the swarm center  $X_i^c$ , if the food satisfaction is greater and not too crowded (i.e.,  $\sigma Y_i < Y_c/n_f$ ), then move a step from  $X_i$  to  $X_i^c$ . Otherwise, preying behavior is to identify a next position for the current.

Following behavior is described as (6). Let  $X_i^{\max}$  be a AF with the greatest food consistence among AFs in the current visual scope. If the food satisfaction is greater and not too crowded (i.e.,  $\sigma Y_i < Y_i^{\max}/n_f$ ), then move a step from  $X_i$  to  $X_i^{\max}$ . Otherwise, preying behavior is to identify a next position for the current.

$$X_{i|\text{next}} = \begin{cases} X_i + \text{rand} \times \text{step} \times \frac{X_j - X_i}{\|X_j - X_i\|} & Y_i < Y_j \\ X_i + \text{rand} \times \text{step} & \text{else,} \end{cases} \quad (4)$$

$$X_{i|\text{next}} = \begin{cases} X_i + \text{rand} \times \text{step} \times \frac{X_i^c - X_i}{\|X_i^c - X_i\|} & \sigma Y_i < \frac{Y_c}{n_f} \\ \text{prey}(X_i) & \text{else,} \end{cases} \quad (5)$$

$$X_{i|\text{next}} = \begin{cases} X_i + \text{rand} \times \text{step} \times \frac{X_i^{\max} - X_i}{\|X_i^{\max} - X_i\|} & \sigma Y_i < \frac{Y_{\max}}{n_f} \\ \text{prey}(X_i) & \text{else.} \end{cases} \quad (6)$$

TABLE I:  $C(7, 4)$ .

1	(1 2 3 4)	8	(1 2 5 6)	15	(1 3 5 7)	22	(2 3 4 6)	29	(2 4 6 7)
2	(1 2 3 5)	9	(1 2 5 7)	16	(1 3 6 7)	23	(2 3 4 7)	30	(2 5 6 7)
3	(1 2 3 6)	10	(1 2 6 7)	17	(1 4 5 6)	24	(2 3 5 6)	31	(3 4 5 6)
4	(1 2 3 7)	11	(1 3 4 5)	18	(1 4 5 7)	25	(2 3 5 7)	32	(3 4 5 7)
5	(1 2 4 5)	12	(1 3 4 6)	19	(1 4 6 7)	26	(2 3 6 7)	33	(3 4 6 7)
6	(1 2 4 6)	13	(1 3 4 7)	20	(1 5 6 7)	27	(2 4 5 6)	34	(3 5 6 7)
7	(1 2 4 7)	14	(1 3 5 6)	21	(2 3 4 5)	28	(2 4 5 7)	35	(4 5 6 7)

In addition, step and visual parameters play an important role in FSA. They determine the convergence speed of FSA and make it escape from the local optimal solution. They are described as follows [11]:

$$\begin{aligned} \text{visual} &= \text{visual} \times f\left(4 \times \frac{\text{gen}}{\text{GEN}}, 0, 2\right), \\ \text{step}_{i+1} &= \text{step}_i \times g\left(\frac{\text{gen}}{\text{GEN}}\right), \end{aligned} \quad (7)$$

where  $f()$  is the Lorentzian function and  $g()$  is the normal distribution function.

### 3. A Coding Method for Combination

Let  $C = \{1, 2, \dots, n\}$  be an integers set which contain  $n$  elements. The permutations number of  $C$  is  $n!$ . Sort them from small to large by lexicographic order. The Cantor expansion and inverse Cantor expansion indicate that there is a one-to-one correspondence between the full permutation set of  $C$  and  $\{1, 2, \dots, n!\}$ . Converting the full permutation into decimal number can be used to solve the TSP problem by the heuristic algorithm. Different from the TSP problem, rough set attribute reduction focuses on the combination of attribute set; it is necessary to discuss the ranking of a combination in the combinations sequence.

Let  $C(n, m) = \{\alpha \mid \alpha \subseteq C, |\alpha| = m\}$ ; then the cardinal number of  $C(n, m)$  is  $C_n^m$ . For  $\alpha \in C(n, m)$  and  $\alpha = (a_1, a_2, \dots, a_m)$ , then  $\alpha \subseteq C$ . Sort all the elements that, in  $\alpha$  from small to large, that is,

$$\begin{aligned} &\text{if } 1 \leq i < j \leq m, \\ &\text{then } a_i < a_j. \end{aligned} \quad (8)$$

By lexicographic order,  $C(n, m)$  can be regarded as a sequence.

*Example 1.* Let  $C = \{1, 2, 3, 4, 5, 6, 7\}$  and  $m = 4$ . All the elements of  $C(7, 4)$  were shown in Table 1.

From Table 1, we can find that the NO 22 is (2 3 4 6). By lexicographic order, the following proposition is apparent.

**Proposition 2.** Let  $\alpha = (a_1, a_2, \dots, a_m)$ ,  $\beta = (b_1, b_2, \dots, b_m) \in C(n, m)$ .  $\alpha$  precede  $\beta$  if and only if  $\exists k$  ( $1 \leq k \leq m$ ) such that,  $\forall i$  ( $1 \leq i < k$ ),  $a_i = b_i$  and  $a_k < b_k$ .

**Proposition 3.** Let  $\alpha = (a_1, a_2, \dots, a_m) \in C(n, m)$ , then,  $\forall i$  ( $1 \leq i \leq m$ ),  $i \leq a_i \leq n + 1 - m$ .

According to the property of combination, let  $\alpha = (a_1, a_2, \dots, a_m) \in C(n, m)$  and  $a_i = j$ . Then the number of combinations in  $C(n, m)$  as  $\alpha$  is  $C_{n-j}^{m-i}$  by Proposition 3. Thus we can get a matrix  $M(T(i, j))_{m \times n}$  about  $C(n, m)$ , where

$$T(i, j) = \begin{cases} C_{n-j}^{m-i} & 1 \leq i \leq m, i \leq j \leq n + 1 - m \\ 0 & \text{other.} \end{cases} \quad (9)$$

The follow equation is apparent: if  $T(i, j) \neq 0$ , then

$$T(i, j) = \sum_{k=j+1}^n T(i+1, k). \quad (10)$$

*Example 4.* For  $C(7, 4)$ , matrix  $M(T(i, j))_{7 \times 4}$  is as follows:

$$M = \begin{pmatrix} 20 & 10 & 4 & 1 & 0 & 0 & 0 \\ 0 & 10 & 6 & 3 & 1 & 0 & 0 \\ 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (11)$$

Since  $|\{1, 2, \dots, C_n^m\}| = |C(n, m)| = C_n^m$ , define a mapping  $h$  from  $\{1, 2, \dots, C_n^m\}$  to  $C(n, m)$  as follows:

$$\begin{aligned} h: \{1, 2, \dots, C_n^m\} &\longrightarrow C(n, m) \\ h(x) &= \alpha = (a_1, a_2, \dots, a_m), \end{aligned} \quad (12)$$

where  $x \in \{1, 2, \dots, C_n^m\}$  and  $\alpha = (a_1, a_2, \dots, a_m) \in C(n, m)$ . We will give Algorithm 1 for calculating the combination by the matrix  $M(T(i, j))_{m \times n}$ .

For all  $x \in \{1, 2, \dots, C_n^m\}$ , according to Algorithm 1 and (9) and (10), we can calculate  $h(x) = \alpha = \{a_1, a_2, \dots, a_m\}$ ; thus we can decode a integer within  $C_n^m$  into a combination in  $C(n, m)$ .

*Example 5.*  $C_7^4 = 35$ ; let  $x = 22 \in \{1, 2, \dots, 35\}$  and  $h(22) = (a_1, a_2, a_3, a_4)$ . By Algorithm 1, we know

$$\begin{aligned} 20 &= T(1, 1) < x = 22 < T(1, 1) + T(1, 2) = 30; \text{ then } \\ a_1 &= 2, \quad x = 22 - T(1, 1) = 2; \\ x &= 2 < T(2, 2 + 1) = 6; \text{ then } a_2 = 3, \quad x = 2 - 0 = 2; \\ x &= 2 < T(3, 3 + 1) = 3; \text{ then } a_3 = 4, \quad x = 2 - 0 = 2; \\ 1 &= T(4, 4 + 1) < x = 2 = T(4, 4 + 1) + T(4, 4 + 2) = 2; \\ \text{then } a_4 &= 6; \\ h(22) &= (2, 3, 4, 6); \end{aligned}$$

in Table 1, the 22th permutation being (2, 3, 4, 6), that is, the same result.

```

input:  $x, n, m.$  ( $x \in C_n^m$ )
output:  $\alpha = (a_1, a_2, \dots, a_m)$ 
begin
   $l = 1;$ 
  if  $\sum_{k=1}^{j_1-1} T(1, k) < x \leq \sum_{k=1}^{j_1} T(1, k);$ 
     $a_1 = j_1$  and  $x = x - \sum_{k=1}^{j_1-1} T(1, k);$ 
  end
   $l = l + 1;$ 
  for  $l = 2$  to  $m$  do
    if  $\sum_{k=j_{l-1}+1}^{j_l-1} T(l, k) < x \leq \sum_{k=j_{l-1}+1}^{j_l} T(l, k).$ 
       $a_l = j_l,$  and  $x = x - \sum_{k=j_{l-1}+1}^{j_l-1} T(l, k);$ 
    end
  end
end

```

ALGORITHM 1: DetoCo( $x, n, m$ ).

## 4. RFSA-RST Algorithm

In this section, for finding minimal reducts of a dataset, an algorithm which is named RFSA-RST in this paper based on RST and FSA is proposed. It first uses the concepts of the core to find the core attributes, and FSA is employed in a restrained manner to find the minimal reducts to ensure that the result is the minimal length.

*4.1. Encoding Method.* Essentially, AR is a combinatorial optimization problem and the solution space is the power set of the attribute set. Let  $C = \{1, 2, \dots, n\}$  be conditional attribute set, the power set of  $C$  be  $2^C$ , and  $C(n, m) = \{\alpha \mid \alpha \subseteq C, |\alpha| = m\}$ ; then  $C(n, m) \subseteq 2^C$ . Clearly,  $\sum_{i=0}^n C(n, i) = 2^C$  and  $C(n, i) \cap C(n, j) = \emptyset$  ( $i \neq j$ ). By the cardinality of the subset of  $C$ ,  $2^C$  is split into  $n$  disjoint parts. According to Algorithm 1, every integer within  $C_n^m$  can be decoded into a combination in  $C(n, m)$ , so integer encoding method will be adopted. Thus, each integer represents a combination in  $C(n, m)$  and the solution space is integer type. Let  $\Delta$  be defined as the increment of  $X_i$ . By (4)–(6), it may be decimal. So, it is not adaptive to the solution space. In order to improve this problem, the new  $X_i$  increment is shown as follows [11]:

$$X_{i|next} = \begin{cases} X_i + \text{round}(\Delta) + 1 & \Delta > 0 \\ X_i + \text{round}(\Delta) - 1 & \text{else.} \end{cases} \quad (13)$$

*4.2. Fitness Function.* In FSA, the quality of a AF is determined by fitness function. For AR problem, the subset should retain the classification ability and has minimal attributes. Generally, the fitness function must meet those issues. In addition, it should be as simple as possible to improve the computation efficiency. In this paper, all subsets without containing the core attribute are encoded into an integer as a AF. Since the cardinality of each combination in  $C(n, m)$  is fixed, FSA is restricted to a fixed length subset of condition attribute set and judges a AF by its dependency value for the attribute subset represented by the integer, as explained below.

TABLE 2: Basic information about the datasets.

Data set	$n$	$m$
Lung-cancer	26	52
Soybean-small	47	36
Sponge	76	45
Zoo	101	17
Blance	625	5
Spect	187	23

For a AF and its position being  $X$ , calculate the corresponding combination  $h(x)$  in  $C(n, m)$  by Algorithm 1. Define the fitness value of  $X$  as follows:

$$F(X) = \gamma_{h(X) \cup \text{Core}}. \quad (14)$$

*4.3. Algorithm Description.* In FSA, the initial population of size  $N$  needs to be provided representing multiple points in the search space. If there is a fitness value of AF that is equal to  $\gamma_C$ , then the process terminates and outputs the corresponding attribute subsets of AF. If change in average fitness value of two successive generations is 0.001 or less, no further generations are created and terminate this loop. Then a new initial population is created for the next loop. These attribute subsets of the new initial population correspond to one more attribute. So, initial strings in  $i$ th loop of the algorithm have one more attribute than that in  $(i - 1)$ th loop. The whole process is repeated based on this new initial population until a minimal set of reduct(s) is found. The detailed process of RFSA-RST is shown in Algorithm 2.

## 5. Performance Comparison

To evaluate the effectiveness of the proposed algorithm, RFSA-RST, we carried out experiments in the following way. Another two different types of algorithms were used. One is proposed in [12], denoted here by RGA-RST, and the other is proposed in [17], denoted here by PSO-RST. A PC running Windows XP with 3.1 GHz CPU and 1 GB of main memory was employed to run these three algorithms. The test datasets were obtained from the UCI machine learning repository and 6 datasets were chosen. Basic information about the datasets is listed in Table 2, where  $n$  and  $m$  are the number of objects and (condition) attributes, respectively.

To make the comparison fair and reasonable, the three algorithms were independently run 50 times on each of the datasets with the same setting of the parameters involved. For each run, three values needed to be recorded for each experiment, the length of the output, the run time, and whether the output is a reduction. If the result is a reduction, then the run is said to be normal; otherwise, the run is said to be unsuccessful. If the result corresponds to a minimum reduct, then the run is not only normal but also successful. Let STL, AVL, and AVT be denoted as the shortest length, average length, and average running time, respectively, during the 50 runs. The ratios of successful and normal runs are denoted, respectively, by  $s_1$  and  $s_2$ . The performances of the three

```

Input:  $DS = (U, C, D, V, f)$ .
Output:  $Red$ 
begin
   $Red = \{\}$ ; Calculate  $\gamma_C$  and  $\gamma_{Core}$ ;
  if  $\gamma_{Core} = \gamma_C$ ,
     $Red = Core$ , output  $Red$  and terminate process;
  else
     $L = C \setminus Core$ ;  $n = |L|$ ;  $m = 0$ ;
    repeat
       $m = m + 1$ ;  $avgo = 0$ ;
      Get  $\{r_1, r_2, \dots, r_N\} \subseteq \{1, 2, \dots, C_n^m\}$  randomly;
      repeat
         $sum = 0$ ;
        for  $i = 1$  to  $N$ 
          Calculate  $P_i = f(r_i)$  by Algorithm 1;  $P_i = P_i \cup CS$ ;
          Calculate  $F(P_i)$  for the permutation  $P_i$  by Eq (14).
          if  $F(n_i) = \gamma_C$ 
             $Red = Red \cup P_i$ ;
          else
             $sum = sum + F(P_i)$ ;
          end
        end
         $avgn = sum/N$ ;  $change = avgn - avgo$ ;  $avgo = avgn$ ;
        Implement swarming behavior, following behavior
        and preying behavior.
      until  $change \leq 0.001$ ;
      until  $Red \neq \{\}$ ;
      return  $Red$ ;
    end
  end

```

ALGORITHM 2: RFSR-RST(DS).

TABLE 3: Performance of three algorithms.

Dataset	PSO-RST		RGA		RFSR-RST	
	STL	AVL	STL	AVL	STL	AVL
Lung-cancer	3	4.14	3	4.02	3	<b>3.92</b>
Soybean-small	2	2.29	2	<b>2.32</b>	2	2.38
Sponge	8	8.7	8	8.4	8	<b>8.02</b>
Zoo	5	5.32	5	5.05	5	<b>5</b>
Blance	4	4.02	4	4	4	4
Spect	16	16	16	16	16	16

algorithms on the datasets are reported in Tables 3 and 4, respectively.

From Table 3, the proposed algorithm has the same performance as the other two algorithms for the shortest length of the outputs, but outperforms the other two algorithms in terms of the average length of output solutions except for the Soybean-small dataset. It means that the stability of the proposed algorithm is higher than the other two algorithms.

From Table 4, for the ratio of normal runs, all the outputs of the three algorithms are roughly the same, but the proposed algorithm outperforms the other two algorithms in terms of the ratio of successful runs except for the Soybean-small dataset. Therefore, if the minimum attribute reduction is required, the proposed algorithm is the best of the three

algorithms. It is also reflected to the average length in Table 3. As far as the average running time is concerned, the proposed algorithm is slightly worse than PSO-RST but better than RGA.

Since the operation modes of the proposed algorithm and RGA algorithm are the same, another experiment has been done to evaluate the time efficiency of the two algorithms. We modify some process of both algorithms where the core attribute set is no longer identified and  $n$  in RGA (for the parameter  $m$  in the proposed algorithm) is set from 1 to the cardinality of minimal attribute reduction for each dataset. Record the running time for each loop. In order to show the results more clearly and length limitations, we only show the test results about the datasets Spect and Sponge. As shown

TABLE 4: Performance of three algorithms.

Dataset	PSO-RST		RGA		RFSA-RST	
	$s_1/s_2$	AVT(s)	$s_1/s_2$	AVT(s)	$s_1/s_2$	AVT(s)
Lung-cancer	0.06/1	0.2073	0.08/1	0.8103	<b>0.22/1</b>	0.2255
Soybean-small	0.7/1	0.7117	<b>0.78/1</b>	1.0177	0.72/1	0.7163
Sponge	0.6/0.92	6.4159	0.62/0.98	16.1129	<b>0.78/0.98</b>	6.8089
Zoo	0.94/1	1.3892	0.96/1	2.1648	<b>1/1</b>	1.4584
Blance	0.98/1	0.5726	1/1	1.9935	1/1	0.6045
Spect	1/1	12.0372	1/1	36.0937	1/1	12.9012

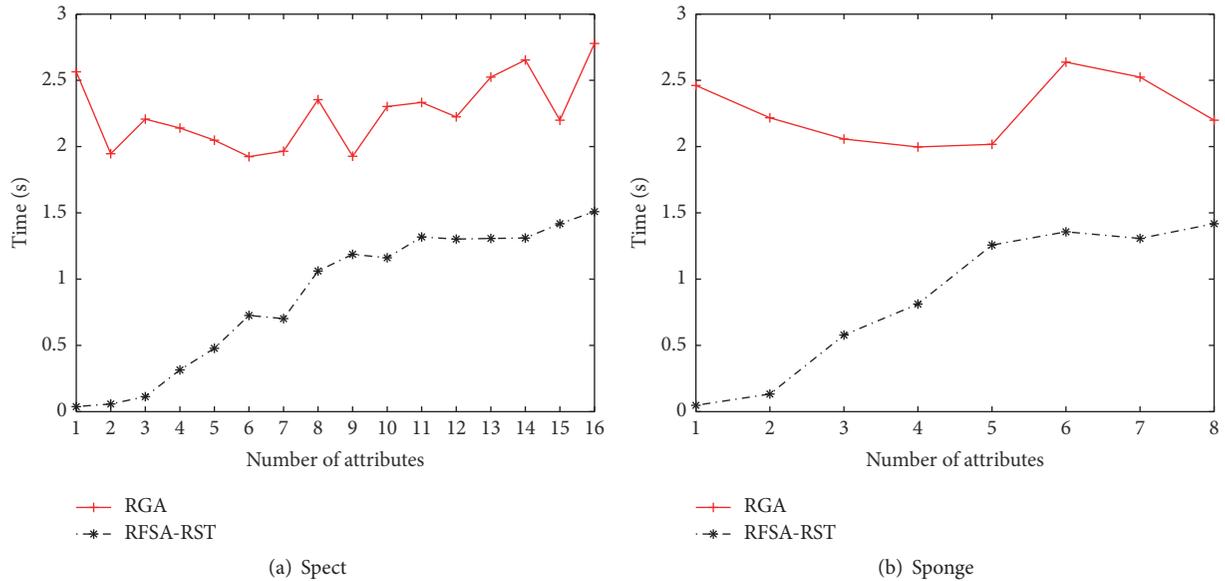


FIGURE 1: Average running time.

in Figure 1. By Figure 1, we can find that, for all datasets, the running time in each loop of the proposed algorithm was increased with the increase of  $m$ , while RGA were relatively stable. That is due to the fact that the search process of the proposed algorithm focuses on the attribute subset with fixed length. Therefore, the time efficiency of the algorithm is higher in each loop and the convergence rate is faster. It shows that the algorithm proposed in this paper is more efficient.

Summing up the experiment results, we see that the proposed algorithm is more efficient than the other two typical algorithms on the datasets for test, although the running time is slightly worse than PSO-RST.

## 6. Conclusions

In this paper, we derived a new method for minimum attribute reduction based on rough set theory and fish swarm algorithm. In this method, by a coding method of combination subset of attributes set, the FSA has been used to search minimum attribute reduction and attribute dependency has been applied to calculate the fitness values of the attribute subsets. The FSA has been restrained in which every integer corresponds to a specified length attribute subset in each search process. The cardinality of the attribute subset

represented by AF was starting from the length of the core and incremented by one after in each loop.

Numerous test results show that it can improve not only the accuracy of minimal attribute reduction but also the efficiency of convergence rate.

To improve search efficiency, future enhancements to this work are to confirm whether the starting length was reasonable and, to improve the search time efficiency, these works can reduce redundant search process.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the Opening Project of Sichuan Province University Key Laboratory of Bridge Non-Destruction Detecting and Engineering Computing (2014QYJ02).

## References

- [1] H. Liu and M. Hiroshi, *Feature Selection for Knowledge Discovery And Data Mining*, Springer Science & Business Media, 1998.

- [2] Z. Pawlak and R. Sowiński, "Rough set approach to multi-attribute decision analysis," *European Journal of Operational Research*, vol. 72, no. 3, pp. 443–459, 1994.
- [3] Y. Yu, W. Pedrycz, and D. Miao, "Neighborhood rough sets based multi-label classification for automatic image annotation," *International Journal of Approximate Reasoning*, vol. 54, no. 9, pp. 1373–1387, 2013.
- [4] J. Zhou, W. Pedrycz, and D. Miao, "Shadowed sets in the characterization of rough-fuzzy clustering," *Pattern Recognition*, vol. 44, no. 8, pp. 1738–1749, 2011.
- [5] L. Polkowski and A. Skowron, "Rough sets: a perspective," in *Rough sets in knowledge discovery, 1*, vol. 18 of *Stud. Fuzziness Soft Comput.*, pp. 31–56, Physica, Heidelberg, 1998.
- [6] Q. Hu, D. Yu, Z. Xie, and J. Liu, "Fuzzy probabilistic approximation spaces and their information measures," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 191–201, 2006.
- [7] J. Starzyk, D. E. Nelson, and K. Sturtz, "Reduct generation in information systems," *Engineering Letters*, vol. 2, no. 2, pp. 36–41, 1999.
- [8] J. A. Starzyk, D. E. Nelson, and K. Sturtz, "A mathematical foundation for improved reduct generation in information systems," *Knowledge and Information Systems*, vol. 2, no. 2, pp. 131–146, 2000.
- [9] A. Skowron and C. Rauszer, *The Discernibility Matrices and Functions in Information Systems*, vol. 11 of *Theory & Decision Library*, 1992, 331–362.
- [10] R. Skowron, "Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory," in *Intelligent Decision Support*, R. Slowinski, Ed., pp. 311–362, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.
- [11] X.-Y. Luan, Z.-P. Li, and T.-Z. Liu, "A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm," *Neurocomputing*, vol. 174, pp. 522–529, 2016.
- [12] J. Wroblewski, "Finding minimal reducts using genetic algorithms," in *Proceedings of the Second Annual Joint Conf. on Information Sciences*, pp. 186–189, Wrightsville Beach, NC, USA, 1995.
- [13] A. K. Das, S. Chakrabarty, S. K. Pati, and A. H. Sahaji, "Applying restrained genetic algorithm for attribute reduction using attribute dependency and discernibility matrix," *Communications in Computer and Information Science*, vol. 292, pp. 299–308, 2012.
- [14] A.-R. Hedar, M. A. Omar, and A. A. Sewisy, "Rough sets attribute reduction using an accelerated genetic algorithm," in *Proceedings of the 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPDC 2015*, jpn, June 2015.
- [15] S. Abdul-Rahman, Z.-A. Mohamed-Hussein, and A. A. Bakar, "Integrating Rough Set Theory and Particle Swarm Optimisation in feature selection," in *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, pp. 1009–1014, egypt, December 2010.
- [16] L. Cervante, B. Xue, L. Shang, and M. Zhang, "Binary particle swarm optimisation and rough set theory for dimension reduction in classification," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pp. 2428–2435, mex, June 2013.
- [17] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.
- [18] J. Anuradha and B. K. Tripathy, "Improved intelligent dynamic swarm PSO algorithm and rough set for feature selection," *Communications in Computer and Information Science*, vol. 270, no. II, pp. 110–119, 2012.
- [19] Y. Chen, D. Miao, and R. Wang, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognition Letters*, vol. 31, no. 3, pp. 226–233, 2010.
- [20] R. Forsati, A. Moayedikia, R. Jensen, M. Shamsfard, and M. R. Meybodi, "Enriched ant colony optimization and its application in feature selection," *Neurocomputing*, vol. 142, pp. 354–371, 2014.
- [21] Y. Chen, Q. Zhu, and H. Xu, "Finding rough set reducts with fish swarm algorithm," *Knowledge-Based Systems*, vol. 81, pp. 22–29, 2015.
- [22] F. Wang, J. Xu, and L. Li, "A novel rough set reduct algorithm to feature selection based on artificial fish swarm algorithm," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8795, pp. 24–33, 2014.
- [23] X. Li, Z. Shao, and J. Qian, "An optimizing method based on autonomous animates: fish swarm algorithm," *Systems Engineering-Theory & Practice*, vol. 22, pp. 32–38, 2002.

