Improving Ant Swarm Optimization with Embedded Vaccination for Optimum Reducts Generation

1Lustiana Pratiwi, 2Yun-Huoy Choo, 3Azah Kamilah Muda, 4Noor Azilah Muda
Faculty of Information and Communication Technology,
Universiti Teknikal Malaysia Melaka (UTeM),
Melaka, Malaysia
1lustiana@gmail.com, 2huoy@utem.edu.my, 3azah@utem.edu.my, 4azilah@utem.edu.my

Abstract— Ant Swarm Optimization refers to the hybridization of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms to enhance optimization performance. It is used in rough reducts calculation for identifying optimally significant attributes set. This paper proposes a hybrid ant swarm optimization algorithm by using immunity to discover better fitness value in optimizing rough reducts set. Unlike a conventional PSO/ACO algorithm, this hybrid algorithm shows improvement of the classification accuracy in its generated rough reducts to solve NP-Hard problem. This paper has evaluated the immune algorithm in 12 common benchmark dataset to evaluate the performance of rough reducts-based on attribute reduction. The results show that immune ant swarm algorithm is very competitive in terms of fitness value, number of iterations, and classification accuracy to produce a better optimization technique and more accurate results in rough reducts generation. The results also show that immune ant swarm optimization provides a slight increase in accuracy when compared to the differential evolution variant.

Keywords-component; rough reducts; particle swarm optimization; ant colony optimization; immunity; ant swarm optimization

I. INTRODUCTION

In the concept of rough set theory, reducts is an important attribute set which can discern all discernible objects by the original of information system. Given an information system $IS = (U, A)$, let $B \subseteq A$, where $U$ is a non-empty finite set of objects called the universe and $A$ is a non-empty finite set of attributes, such that $a : U \rightarrow V_a$ for every $a \in A$. A reduct of $A$ is a minimal set of attributes $B \subseteq A$ such that all attributes $a \in A - B$ are dispensable and an associated equivalence of indiscernibility relation denoted by $IND(B)$

$$IND(B) = \{(x, x') \in U^2 | \forall a \in B \ a(x) = a(x')\},$$

and $IND(B)$ is called the $B$-indiscernibility relation. Then an attribute $a$, is said to be dispensable in $B \subseteq A$ if $IND(B) = IND(B \setminus \{a\})$. Otherwise, the attribute is indispensable in $B$ [1], [2], [3], [4], and [5].

Reducts are such subsets that are minimal, and do not contain any dispensable attributes. The set of all reducts of an information system $IS$ is denoted $RED(IS)$ or simply $RED$. Reducts calculation has great importance in features selection analysis. It enables the calculation of absolute reduction as well as relative reduction with core. However, the computational complexity of generating optimal reducts is very high. Since the search space increase exponentially with the number of attributes, finding the optimal reducts, a minimal reducts with minimal cardinality of attributes among all reducts is a NP-hard problem [2].

Formally, the minimum attribute reduction problem is a nonlinearly constrained combinatorial optimization problem. Hence, global optimization methods could be used to solve it. Therefore, many existing optimization technique have been proposed to solve the optimal reducts problem, such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Immunity. This paper has analyzed the results from the previous study and suggested an improved solution by proposing a rough reducts optimization using a hybrid approach of PSO/ACO. This solution is also enhanced with embedded vaccination process which can significantly reduce iteration complexity and increase the fitness value, when PSO does not possess the ability to improve upon the quality of the solutions as the number of generations is increased.

The rest of the paper is organized as follows. Section II presents some related works on the optimization approach in attribute reduction problem. Section III describes the improved ant swarm optimization technique to enhance the previous studies with higher possibility of finding a minimum reduction with immunity. The effectiveness of proposed algorithm is demonstrated, compared to the other variants, and the computation results are discussed in Section IV, and finally, in section V outlines the conclusions and the future work.

II. RELATED WORK

Various optimization techniques have been suggested to solve the crucial problem in rough reduct; gaining interest in metaheuristic solutions of NP-hard problem, such as genetic algorithm (GA), ant colony optimization (ACO), artificial immune system (AIS), and more recently particle swarm optimization (PSO). Some researchers have proposed stochastic methods for inducing rough reducts [6]. Some studies used genetic algorithms to find minimal reducts [7]. Combination of genetic algorithm and greedy algorithm to generate short reducts was also proposed. However, it used
highly time-consuming operations and cannot assure the optimality of the resulting reducts subset.

Bjorvand applied genetic algorithms to compute approximate reducts [8]. He followed Wroblem’s work as a foundation, and has suggested several variations and practical improvements in both speed and the quality of approximation. To obtain a good initial population for the GA, Bjorvand included the attribute core in all candidates. Zhai also proposed an integrated feature extraction approach based on rough set theory and genetic algorithms [9]. Based on this approach, a prototype feature extraction system has been established and illustrated in an application for the simplification of product quality evaluation. The results show that it can remarkably reduce the cost and time consumed on product quality evaluation without compromising the overall specifications of the acceptance tests [9] and [10].

Another commonly used optimization technique in data mining strategy is particle swarm optimization (PSO) technique. PSO was proposed by Kennedy and Eberhart [11] to be applied to difficult combinatorial optimization problem which aims to find reducts of minimal cardinality for attribute subsets [12], [13], and [14]. In [12] and [14] PSO is applied to find reducts of minimal cardinality. Like classical genetic algorithms, the particle’s position is a binary representation of the attribute subsets. The fitness function and position-updating strategy are also the key factors in PSO for feature selection, which needed to be improved.

On the other hand, B. Yue et al has discovered the best feature combinations in an efficient way to observe the change of the positive region as the particle proceed through the search space in PSO [13]. A comparison between PSO and GA was presented, where PSO does not have genetic operators like crossover and mutation. The search is influenced by the speed of the particles and the processing time has reduced as compared to other approaches [15]. Subsequently, better results were obtained by using PSO algorithm, as reported in [13] and [15].

Chen et al. [16] and Zeng et al. [17] verified and demonstrated some experiments to provide efficient solution to find the minimal features subset using ant colony optimization (ACO). ACO has been verified again by [18] to provide competitive solutions efficiently to deal with attribute reduction in rough set theory. This algorithm has the features to update the pheromone trails of the edges connecting every two different attributes of the best-so-far solution. The pheromone values were limited between the upper and lower trail limits and a rapid procedure was used to construct candidate solutions. Due to its pheromone update rule and solution construction process, the proposed algorithm has the ability to identify solutions with very small cardinality rapidly [18].

Dongyi Ye et al [19] presented a new attribute reduction algorithm by using a combination of binary particle swarm optimization and the vaccination approach. The experimental result has remarkably improved local search from the immune selection mechanism and outperformed some recent global optimization techniques.

III. IMMUNE ANT SWARM OPTIMIZATION ALGORITHM FOR ROUGH REDUCTS (IASORR)

The term “reducts” corresponds to a wide class of concepts. What typifies all of them is that they are used to reduce information (decision) systems by removing redundant attributes. Given an information system $IS = (\mathcal{U}, A)$, a reducts is a minimal set of attributes $B \subseteq A$ such that $\text{IND}(B) = \text{IND}(A)$, where $\text{IND}(B) = \text{IND}(A)$ are the indiscernibility relations defined by $B$ and $A$, respectively by Pawlak and Skowron in [20].

The intersection of all reducts is called a core. Intuitively, reducts is a minimal set of attributes from $A$ that preserves the original classification defined by $A$. The improvement in rough reduces optimization using PSO/ACO with vaccination is based on the common characteristics of both PSO and ACO algorithm, like, survival as a colony by sharing information locally and globally in the swarm between particles (ants).

A. Ant Swarm Optimization for Rough Reducts (ASORR)

Both PSO and ACO adapt swarm intelligence metaheuristics which is based on population global search and co-operative biologically inspired algorithm motivated by social analogy [21]. PSO was inspired by real life social behavior of bird flocking or fish schooling, while ACO imitates foraging behavior of real life ants. PSO still has the problems of dependency on initial point and parameters, difficulty in finding their optimal design parameters, and the stochastic characteristic of the final outputs for local searching [23].

On the other hand, ACO has positive feedbacks for rapid discovery of good solutions and a simple implementation of pheromone-guided will improve the performance of PSO. Thus in this study, a simple pheromone-guided mechanism is explored to improve the performance of PSO method for optimization of rough reducts [24].

1) Particle Swarm Optimization

In PSO, particles as candidate solutions of a population, simultaneously coexist and evolve based on knowledge sharing with neighboring particles. Each particle generates a solution using directed velocity vector, while flying through the problem search space. Each particle modifies its velocity to find a better solution (position) by applying its own flying experience for the best position memory found in the earlier flights and experience of neighboring particles as the best-found solution of the population [21].

Each particle’s movement is the composition of an initial random velocity and two randomly weighted influence; individuality, the tendency to return to the particle’s best position $p_{\text{best}}$, and sociality, the tendency to move forwards the best previous position of the neighborhood’s $g_{\text{best}}$. Particles update their positions and velocities as shown below

$$v_{i+1} = w_{i}v_{i} + c_{1}r_{1}(p_{i} - x_{i}) + c_{2}r_{2}(p_{b} - x_{i})$$

$$x_{i+1} = \begin{cases} 1, \text{rand}() < \text{sig}(v_{i+1}) \\ 0, \text{rand}() \geq \text{sig}(v_{i+1}) \end{cases}$$

2011 11th International Conference on Hybrid Intelligent Systems (HIS) 449
\[
sig(x) = \frac{1}{1 + e^{-x}}
\]

where \(x^t_i\) represents the current position of particle \(i\) in solution space and subscript \(t\) indicates an iteration count; \(p^t_i\) is the best-found position of particle \(i\) up to iteration count \(t\) and represents the cognitive contribution to the search velocity \(v^t_i\). Each component of \(v^t_i\) can be clamped to the range \([-v_{\text{max}}, v_{\text{max}}]\) to control excessive roaming of particles outside the search space; \(p^t_i\) is the global best-found position among all particles in the swarm up to iteration count \(t\) and forms the social contribution to the search velocity \(v^t_i\). In this way, all particles are updated in a gradual fashion by Shi et al in \([25]\). The variable \(w_t\) is updated as

\[
w_t = (w_{\text{max}} - w_{\text{min}}) \times \frac{t_{\text{max}} - t}{t_{\text{max}}} + w_{\text{min}},
\]

where, \(w_{\text{max}}\) and \(w_{\text{min}}\) denote the maximum and minimum of \(w_t\) respectively; \(t_{\text{max}}\) is a given number of maximum iterations. Particle \(i\) flies toward a new position according to \((1)\) and \((2)\). In this way, all particles \(P\) of the swarm find their new positions and apply these new positions to update their individual best \(p^t_i\) points and global best \(p^t_i\) of the swarm. This process is repeated until iteration count \(t = t_{\text{max}}\) (a user-defined stopping criterion is reached).

Given an information system \(IS = (U, A), A = (C \cup D)\), where \(C\) is a non-empty finite set of condition attributes and \(D\) is a non-empty finite set of decision attributes, such that \(RED \subseteq C\). In the algorithm of PSO, \(P\) denotes the number of particles in the population; \(f(x^t_i)\) represents the objective function value of particle \(i\) at position \(x^t\) \([20]\) and calculated as

\[
f(x^t_i) = \alpha \cdot y_{x^t_i}(D) + \beta \cdot \frac{|C| - |x^t_i|}{|C|},
\]

where \(y_{x^t_i}(D)\) is the classification quality of particle condition attribute set \(x^t_i\), which contains the reducts \(RED\), and relative to decision table \(D\), defined as follows

\[
y_{x^t_i}(D) = \frac{r_{\text{RED}}}{r_C} \quad [22],
\]

where \(r_{\text{RED}}\) represents a degree of dependency of \(RED\) on \(D\) and \(r_C\) represents a degree of dependency of \(C\) on \(D\). \(|x^t_i|\) is the ‘1’ number of the length of selected feature subset or the number of attributes for particle \(x^t_i\), while population of solutions \(P\) is at iteration count \(t\). \(|C|\) is the total number of condition attributes. \(\alpha\) and \(\beta\) parameters correspond to importance of classification quality and subset length, \(\alpha \in [0,1]\) and \(\beta = 1 - \alpha\) \([21]\).
mechanism of ant colony was implemented which acted locally to synchronize positions of the particles in PSO to attain the feasible domain of the objective function [20] faster.

B. Immunity System

The biological immune system, consisting of the innate and adaptive immune systems, is an effective and efficient defense mechanism against infections [23]. The important characters of immunity system are diversity of antibodies and immune memory. To ensure the diversity of antibodies during the evolvement, high affinity and low concentration antibodies are stimulated, but low affinity and high concentrations antibodies are restrained [23]. Immunity memory will be a part of memory cells in immunity system, which is retained during the reaction between the invasion of antigen and antibody. It is the certainty mapping of selected some antibodies in a given probability of $\alpha$, where $0 < \alpha < 1$, and only the high affinity antibodies are breed and mutated, but the low-affinity antibodies still exist in the immune system, and gradually are expelled [23].

According to the different important degrees of attribute, the particles are vaccinated by the abstracted bacteria. Theoretically, the more the weight of an attribute is greater, the more the attribute is more important and it is greater that the probability of attribute is selected [23]. Otherwise, it is smaller which the probability of attribute is selected. So we can vaccinate each particle by the prior knowledge and the more optimization particles will be generated. For mutation probability of a bit in a particle can be defined as [23]:

$$\eta = k_1(w_i - p_i)^2 - k_2,$$

(9)

$$p_i = \{ p_i, \text{rand}() \succeq \eta \} \cup \{ p_i, \text{rand}() < \eta \},$$

(10)

where $w_i$ is the weight of the $i^{th}$ attribute which is mapped to $[0,1]$ in particle $P$, and $p_i$ is the new value of condition attribute. Where $k_1$ and $k_2$ are the adjustable parameters, $\text{rand}()$ is the random number of uniform distribution and $k_1 = 0.4$, $k_2 = 0.005$, $\text{rand}() \in [-1,1]$. If the difference between $w_i$ and $p_i$ is greater, the value $\eta$ is greater, and the aberrance probability of $p_i$ will be greater [23]. Otherwise, the probability of mutation is smaller and even never mutate as the algorithm progresses as shown in the following algorithms:

Step 1: Initialize Optimization

1.1 Initialize algorithm constants $t_{\text{max}}$, $P$, and $[0,1]^m$ is the m-dimensional Boolean particle space.

1.2 Initialize the positive acceleration constants $c_1, c_2$ and MaxFit as the maximum fitness value.

1.3 Calculate the inertia weight of each particle space in (5).

1.4 Initialize randomly all particles’ positions $x_i$ and velocities $v_i$.

1.5 Evaluate all particles’ objective function values $f(x_i)$ in (6).

1.6 Find $f_i^{\text{best}}(p_i^{\text{best}}) = \max\{f(x_1), f(x_2), \ldots, f(x_P)\}$ and initialize $p_i = p_i^{\text{best}}$ and $G_{\text{best}} = f_i^{\text{best}}(p_i^{\text{best}})$.

Step 2: Perform Optimization

While ($t < t_{\text{max}}$)

2.1 Calculate all particles’ velocities $v_i$ in (2) and update particles’ positions in (3).

2.2 Vaccinate each particle in (9) and (10).

2.3 Evaluate all particles objective function value $f(x_i)$ in (6).

2.4 Generate $P$ solutions $x_i$ using (8).

2.5 Evaluate objective function value $f(x_i)$ in (6) and if $f(x_i) > f(x_j)$ then $f(x_i) = f(x_j)$ and $x_i = x_j$.

2.6 Update particle best position if $f_{\text{best}} < f(x_i)$ then update $f_{\text{best}} = f(x_i)$ and $p_i = x_i$.

2.7 Find $f_i^{\text{best}}(p_i^{\text{best}}) = \max\{f_{\text{best}}(P_i), f_{\text{best}}(p_i^{\text{best}})\}$ and if $f_{\text{best}} > f_{\text{best}}(p_i^{\text{best}})$ in history then $G_{\text{best}} = f_{\text{best}}(p_i^{\text{best}})$ and $p_i = p_i^{\text{best}}$.

2.8 If the $G_{\text{best}}$ continuously can’t be changed and can’t meet the $10^{th}$ termination conditions by several evolutionary generations, the program will go to (12) refresh the particles.

Step 3: Report best solution $p^*$ as $P_{\text{best}}$ global best position of the swarm with objective function value $f(p^*)$.

According to the principle that the high affinity and low concentration antibodies are stimulated, but low affinity and high concentrations antibodies are restrained, the $i^{th}$ particle selected probability formulas based on concentration particle is as follows [23], [29], and [30]:

$$D(X_i) = \frac{1}{\sum_{j=1}^{N_0} |f(x_i) - f(x_j)|}, i = 1,2, \ldots, N_0, \quad (11)$$

$$p(x_j) = \frac{\frac{1}{D(X_i)}}{\sum_{j=1}^{N_0} \frac{1}{D(X_i)}}, \quad (12)$$

IV. EXPERIMENTAL RESULTS

The performance of the proposed enhanced Ant Swarm algorithm for global optimization function has been tested on several well-known benchmark multimodal problems [13]. All the test functions are multimodal in nature. Because of the characteristics from previous work [31], it is difficult to seek for the global minima. Ant swarm optimization algorithm parameter settings used in all the simulations is given as: number of particles, $P = 10$; cognitive and social scaling parameters, $c_1 = 2$, $c_2 = 2$; maximum and minimum values of inertia weights, $w_{\text{max}} = 0.7$, $w_{\text{min}} = 0.4$; maximum number of iterations, $t_{\text{max}} = 100$ * $n$, $n$ is the size of solution vector. Implementation of IASORR has been tested on 12 datasets.
The experimental results are presented based on the number of reducts and iterations, fitness values, and the classification accuracy for performance analysis are shown in Table I and II. Three algorithms, which are Particle Swarm Optimization for Rough Sets-based Feature Selection (PSORSFS), Ant Swarm Optimization for Rough Reducts (ASORR), and Immune Ant Swarm Optimization for Rough Reducts (IASORR); were implemented by using Naive Bayes to extract rules from the data for rule induction in classification. Ten-fold cross validation was applied to estimate the classification accuracy.

Results reported in Table I are in term of reducts calculation (also shown graphically in Figure 1) and iterations, which has shown that IASORR has more optimal results for number of iterations than another two algorithms in most of datasets, but in average of attribute reduction, all three algorithms yield the same number of reducts. However, the number of reducts is not sufficient to reflect the performance, in terms of finding the optimal fitness value which is able to gain higher classification accuracy.

The fitness values of each algorithm (shown graphically in Figure 2) reported in Table II has shown that IASORR can achieve better values while the fitness value is proven to be affected by number of reducts. As shown in Table I, IASORR obtained better results than PSORSFS and ASORR by reducing 2 to 3 iterations in average for 10 independent runs. Table II shows the results in terms of classification accuracy where IASORR has induced more significant optimal solution with the generated fitness values. IASORR has achieved best fitness values in 8 datasets as compared to PSORSFS and ASORR, which only gained best results in 2 and 4 datasets respectively.

In term of classification accuracy, IASORR has achieved best accuracy readings in 4 datasets while ASORR in only 1 datasets. On the other hand, PSORSFS lines on top of the three techniques which has gained best accuracy readings in 7 datasets. Thus, based on the experimental results, the IASORR technique has better performances, both in gaining higher fitness value and better quality of reducts as compared to its opponents.

![Figure 1: Performance comparison in term of no. rough reducts set](image1.png)

![Figure 2: Performance comparison in term of fitness values](image2.png)

### Table I. IASORR Experimental Results on No. of Reducts and Iterations

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>Number of Condition Attributes</th>
<th>Instances</th>
<th>No. of Reducts</th>
<th>No. of Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PSORSFS</td>
<td>ASORR</td>
</tr>
<tr>
<td>1</td>
<td>Soybean-small</td>
<td>35</td>
<td>47</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Lung</td>
<td>56</td>
<td>32</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Zoo</td>
<td>16</td>
<td>101</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Lymphography</td>
<td>18</td>
<td>148</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>DNA</td>
<td>57</td>
<td>318</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>Breastcancer</td>
<td>9</td>
<td>699</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Corral</td>
<td>6</td>
<td>64</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>Vote</td>
<td>16</td>
<td>300</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>M-of-N</td>
<td>13</td>
<td>1000</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Tic-Tac-Toe</td>
<td>9</td>
<td>958</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>Exactly</td>
<td>13</td>
<td>1000</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>Mushroom</td>
<td>22</td>
<td>8124</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong></td>
<td></td>
<td><strong>5</strong></td>
<td><strong>5</strong></td>
<td><strong>5</strong></td>
</tr>
</tbody>
</table>
### Table II. IASORR Experimental Results on Fitness Value and Classification Accuracy (%)

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>Fitness Value</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSORSFS ASORR IASORR</td>
<td>PSORSFS ASORR IASORR</td>
</tr>
<tr>
<td>1</td>
<td>Soybean-small</td>
<td>0.8971 0.9917 0.9929</td>
<td>IASORR 100.0000</td>
</tr>
<tr>
<td>2</td>
<td>Lung</td>
<td>0.9152 0.9891 0.9914</td>
<td>IASORR 87.5000</td>
</tr>
<tr>
<td>3</td>
<td>Zoo</td>
<td>0.6891 0.9688 0.9684</td>
<td>ASORR 98.0198</td>
</tr>
<tr>
<td>4</td>
<td>Lymphography</td>
<td>0.6652 0.9631 0.9611</td>
<td>ASORR 83.7838</td>
</tr>
<tr>
<td>5</td>
<td>DNA</td>
<td>0.8927 0.9879 0.9896</td>
<td>IASORR 34.5912</td>
</tr>
<tr>
<td>6</td>
<td>Breastcancer</td>
<td>0.6607 0.9556 0.9556</td>
<td>ASORR 96.1373</td>
</tr>
<tr>
<td>7</td>
<td>Corral</td>
<td>0.9333 0.9333 0.9333</td>
<td>PSORSFS 84.3750</td>
</tr>
<tr>
<td>8</td>
<td>Vote</td>
<td>0.6319 0.6237 0.9500</td>
<td>IASORR 92.7833</td>
</tr>
<tr>
<td>9</td>
<td>M-of-N</td>
<td>0.5385 0.5385 0.9538</td>
<td>IASORR 95.7000</td>
</tr>
<tr>
<td>10</td>
<td>Tic-Tac-Toe</td>
<td>0.2575 0.9147 0.9145</td>
<td>ASORR 69.2119</td>
</tr>
<tr>
<td>11</td>
<td>Exactly</td>
<td>0.5385 0.5385 0.9538</td>
<td>IASORR 68.8000</td>
</tr>
<tr>
<td>12</td>
<td>Mushroom</td>
<td>0.8316 0.8343 0.9811</td>
<td>IASORR 79.7717</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.7043 0.8533 0.9621</td>
<td>IASORR 82.5562</td>
</tr>
</tbody>
</table>

### V. Conclusion

An improved approach of rough reducts optimization based on PSO/ACO with vaccination has been presented. Although the approach is quite simple in structure, it is feasible to solve NP-Hard problem and accelerate the search process. The main contribution of this research lies in the thoroughly an application of ant swarm optimization technique in depth as compared to the past research in attribute reduction. By combining PSO and ACO algorithms with embedded immunity process, the proposed approach is expected to be able to generate better optimal rough reducts, where PSO algorithm performs the global exploration which can effectively reach the optimal or near optimal solution.

This study is also expected to enhance the optimization ability by defining a suitable fitness function with vaccination process to increase the competency in attribute reduction. Thus, the experimental results have also provided further justification on previous studies which have implemented PSO in attribute reduction domain. The initial results of IASORR are promising in most of the tested datasets. Hence, future works are to test on the enhanced IASORR algorithm in various domains to validate its performance in yielding the optimal reducts set.

### Acknowledgment

This work is funded by the Ministry of Higher Education Malaysia and Universiti Teknikal Malaysia Melaka (UTeM) through the Fundamental Research Grant Scheme, FRGS/2009/FTMK(1)-F0072.

### References


