A Comparative Performance Analysis of Gaussian Distribution Functions in Ant Swarm Optimized Rough Reducts

Lustiana Pratiwi, Yun-Huoy Choo, Azah Kamilah Muda
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Melaka, Malaysia
lustiana@gmail.com, huoy@utem.edu.my, azah@utem.edu.my

ABSTRACT

This paper proposed to generate solution for Particle Swarm Optimization (PSO) algorithms using Ant Colony Optimization approach, which will satisfy the Gaussian distributions to enhance PSO performance. Coexistence, cooperation, and individual contribution to food searching by a particle (ant) as a swarm (ant) survival behavior, depict the common characteristics of both algorithms. Solution vector of ACO is presented by implementing density and distribution function to search for a better solution and to specify a probability functions for every particle (ant). Applying a simple pheromone-guided mechanism of ACO as local search is to handle P ants equal to the number of particles in PSO and generate components of solution vector, which satisfy Gaussian distributions. To describe relative probability of different random variables, PDF and CDF are capable to specify its own characterization of Gaussian distributions. The comparison is based on the experimental result to increase higher fitness value and gain better reducts, which has shown that PDF is better than CDF in terms of generating smaller number of reducts, improved fitness value, lower number of iterations, and higher classification accuracy.

KEYWORDS

probability density function, cumulative distribution function, particle swarm optimization, ant colony optimization, rough reducts

1 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.
If an attribute subset $B \subseteq A$preserves the indiscernibility relation R_A, then the attributes $A - B$ are said to be dispensable. Reducts are such subsets that are minimal, and do not contain any dispensable attributes. The set of all reducts of an information system S is denoted $RED(S)$ [1].

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 reduct is very high. Since the search space increase exponentially with the number of attributes, finding the optimal reduct, a minimal reduct with minimal cardinality of attributes among all reducts is a NP-hard problem [1].

Formally, the minimum attribute reduction problem is a nonlinearly constrained combinatorial optimization problem. Hence, global optimization methods could be used to solve reducts problem and gain a better result. Previous study suggested PSO/ACO for optimization problem. PSO/ACO is an enhanced process which can significantly improve the time and quality of PSO algorithm to perform the global exploration which can effectively reach the optimal or near optimal solution.

There are also studies to enhance the optimization ability by defining a suitable fitness function with ACO mechanism process, as function associated with the best solution found so far is used for generating further solutions by the ants and to increase the competency in attribute reduction.

This paper proposed to generate solution components for PSO algorithms using ant colony approach, which satisfy Gaussian distributions to improve the PSO performance. Solution vector of ACO is presented by implementing density and distribution function to search a better solution and to specify a probability functions for every particle (ant).

The rest of the paper is organized as follows. Section 2 illustrates the general uncertainty concept of rough set theory. In Section 3, the descriptions of rough sets-based attribute reduction are presented. Section 4 discusses the fundamental theory of PDF and CDF. Section 5 explains the algorithms of ant swarm optimization technique. Section 6 briefly describes an improved rough reducts optimization framework by using PSO/ACO hybridized algorithms. The performance of both Gaussian density and distribution functions are demonstrated, compared, and the computation results are discussed in Section 7. Finally, Section 8 outlines the conclusions followed by the future work.

2 ROUGH SETS THEORY

Rough set theory [2] is an approach to define vagueness. Similarly to fuzzy set theory it is not an alternative to classical set theory but it is embedded in it. Rough set theory can be viewed as a specific implementation of Frege’s idea of vagueness, i.e., imprecision in this approach is expressed by a boundary region of a set, and not by a partial membership, like in fuzzy set theory [1-6].

Rough set concept can be defined quite generally by means of topological operations, *interior* and *closure*, called *approximations*.To describe this problem more precisely, a set of objects U called the *universe* and an indiscernibility relation $R \subseteq U \times U$ are given, representing the lack of knowledge about elements of U. To be simple, R is assumed as an equivalence relation. Let X to be a subset of U. The objective is to characterize set X with respect to R. The basic concepts of rough set theory are given as below.

- The *lower approximation* of a set X with respect to R is the set of all objects, which can be for *certain* classified as X with respect to R (are *certainly* X with respect to R).

- The *upper approximation* of a set X with respect to R is the set of all objects which can be *possibly* classified as X with respect to R (are *possibly* X in view of R).

- The *boundary region* of a set *X* with respect to *R* is the set of all objects, which can be classified neither as *X* nor as not-*X* with respect to *R*.

A rough sets can be defined as:

- Set *X* is *crisp* (exact with respect to *R*), if the boundary region of *X* is empty.
- Set *X* is *rough* (inexact with respect to *R*), if the boundary region of *X* is nonempty.

Thus a set is *rough* (imprecise) if it has nonempty boundary region; otherwise the set is *crisp* (precise). This is exactly the idea of vagueness proposed by Frege. The approximations and the boundary region can be defined more precisely.

The equivalence class of *R* determined by element *x* will be denoted by *R*(*x*). The indiscernibility relation in certain sense describes the lack of knowledge about the universe. Equivalence classes of the indiscernibility relation, called *granules* generated by *R*, represent elementary portion of knowledge able to perceive due to *R*.

Thus in view of the indiscernibility relation, in general, it is possible to observe individual objects but are forced to reason only about the accessible granules of knowledge. Formal definitions of approximations and the boundary region are as follows:

- *R-lower approximation* of *X*

$$R_*(x) = \bigcup_{x\in U}\{R(x) : R(x) \subseteq X\}. \qquad (1)$$

- *R-upper approximation* of *X*

$$R^*(x) = \bigcup_{x\in U}\{R(x) : R(x) \cap X \neq \varnothing\} \qquad (2)$$

- *R-boundary region* of *X*

$$RN_R(X) = R^*(X) - R_*(X). \qquad (3)$$

The definition of approximations is expressed in terms of granules of knowledge. The lower approximation of a set is union of all granules which are entirely included in the set; the upper approximation – is union of all granules which have non-empty intersection with the set; the boundary region of set is the difference between the upper and the lower approximation. This definition is clearly depicted in Fig. 1.

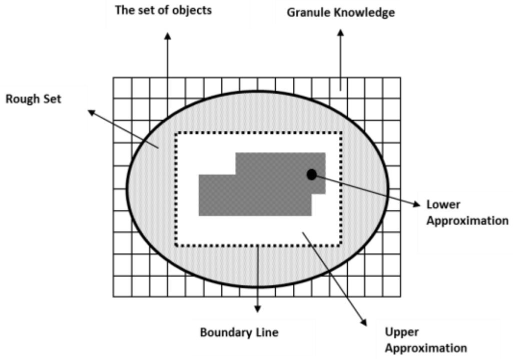


Fig. 1. Rough Set Theory

It is interesting to compare definitions of classical sets, fuzzy sets and rough sets. Classical set is a primitive notion and is defined intuitively or axiomatically. Fuzzy sets are defined by employing the fuzzy membership function, which involves advanced mathematical structures, numbers and functions. Rough sets are defined by approximations. Thus this definition also requires advanced mathematical concepts [2].

It is easily seen that approximations are in fact interior and closure operations in a topology generated by data. Thus fuzzy set theory and rough set theory require completely different in mathematical setting.

Rough sets can be also defined employing, instead of approximation, rough membership function [6]

$$\mu_X^R : U \to < 0,1 >, \tag{4}$$

where

$$\mu_X^R(x) = \frac{|\, X \cap R(x) \,|}{|\, R(x) \,|}, \tag{5}$$

and $|X|$ denotes the cardinality of X.

The rough membership function expresses conditional probability that x belongs to X given R and can be interpreted as a degree that x belongs to X in view of information about x expressed by R [7] and [8]. The meaning of rough membership function can be depicted as shown in Fig.2.

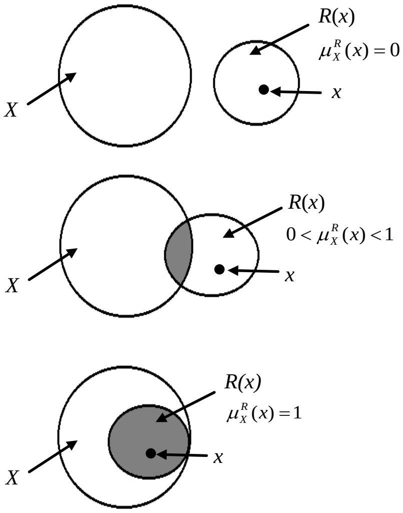


Fig. 2. Areas under a probability density function f on the interval [a,b] [8].

The rough membership function can be used to define approximations and the boundary region of a set, as shown below [9],

$$R_*(X) = \{x \in U : \mu_X^R(x) = 1\}, \tag{6}$$

$$R^*(X) = \{x \in U : \mu_X^R(x) > 0\}, \tag{7}$$

$$RN_R(X) = \{x \in U : 0 < \mu_X^R(x) < 1\} \tag{8}$$

There are two definitions of rough sets, which are as follows:

Definition 1: Set X is *rough* with respect to R if $R_*(X) \neq R^*(X)$ and
Definition 2: Set X *rough* with respect to R if for some x, $0 < \mu_X^R(x) < 1$.

The definition 1 and definition 2 are not equivalent [7] and rough set theory clearly distinguishes two very important concepts, vagueness and uncertainty, very often confused in the Artificial Intelligence literature. Vagueness is the property of sets and can be described by approximations, whereas uncertainty is the property of elements of a set and can be expressed by the rough membership function [10] and [11].

3 ROUGH SETS-BASED ATTRIBUTE REDUCTION

Reducts is an important notion of knowledge discovery in rough set theory. It is a minimal set of attributes having the equivalent discernible capability as the original information system. Given an information system *A* = (*U*, *A*), let $B \subseteq A$. Where U is the universe, a non-empty finite set of objects, *A* reduct of *B* is a set of attributes $B' \subseteq B$ such that all attributes $a \in B - B'$ are dispensable and an associated equivalence of indiscernibilty relation denoted by *IND* where $IND(B) = IND(B')$. An attribute *a*, is said to be dispensable in $B \subseteq A$ if $IND(B) = IND(B - \{a\})$; otherwise, the attribute is indispensable in *B* [1-6].

The rough reducts calculation is a NP-hard problem. Before calculating reducts, the rough reducts method will discretize the quantitative attributes into the appropriate interval range. The rough set theory recommends the Boolean reasoning approach as the discretization method since its performance has been proven on discretizing quantitative attributes solely based on data [3]. Searching for the prime implicant in the Boolean reasoning approach is not a trivial step. It involves the optimization theory which is highly complex in computation. In addition, optimization is also implied in approximation algorithm after the discretization step.

Various optimization techniques have been suggested to solve the crucial problem in rough reducts including the genetic algorithm (GA), Johnson's algorithm, Holte's 1R, and RESS-based reducers [3], ant colony optimization (ACO), artificial immune system, and more recently particle swarm optimization (PSO) and stochastic methods for inducing rough reducts [12].

PSO has been used to solve the difficult combinatorial optimization problem in finding minimal reducts. Similar to chromosomes in GA, the particle position in PSO is used to represent the attribute subsets. The PSO algorithm showed convincing results in opposes to GA algorithms in term of speed and optimization results in the comparison of both techniques [12].

Cheng et al., Zeng et al. and Ke et al. have verified and demonstrated ACO algorithm to provide efficient solution to find the minimal features subset in rough reducts calculation [12]. ACO 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.

4 PROBABILITY DENSITY AND CUMULATIVE DISTRIBUTION FUNCTIONS

The ACO stage of the algorithm is based on a continuous ant colony optimization (ACOR) which was introduced by Socha and Dorigo [13]. They have considered a weighted sum of several one-dimensional Gaussian functions. Here, since ACO works as an auxiliary tool to guide the exploration and to increase the control in Exploitation, to utilize a simple Gaussian functions as explained below.

Each continues random variable X has an associated Probability Density Function (PDF), $f(x)$. It "records" the probability associated with X as areas under its graph. More precisely, "the probability that a value X is between a and b" $= P(a \leq X \leq b) = \int_a^b f(x)dx$.

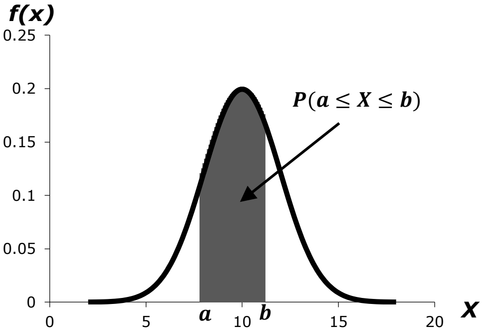


Fig. 3. Areas under a probability density function f on the interval $[a, b]$.

For example,

$$P(1 \leq X \leq 3) = \int_1^3 f(x)dx, \quad \textbf{(9)}$$

$$P(3 \leq X) = P(3 \leq X < \infty) = \int_3^\infty f(x)dx, \quad \textbf{(10)}$$

$$P(X \leq -1) = P(-\infty < X \leq -1) = \int_{-\infty}^{-1} f(x)dx. \quad \textbf{(11)}$$

i) Since probabilities are always between 0 and 1, it must be that $f(x) \geq 0$, so that $\int_a^b f(x)dx$ can never give a "negative probability", and

ii) Since a "certain" event has probability 1, $P(-\infty < X < \infty) = 1 = \int_{-\infty}^{\infty} f(x)dx =$ total area under the graph of $f(x)$

The properties i) and ii) are necessary for a function $f(x)$ to be the PDF for some random variable X [13].

In 1809, Gaussian distribution was applied to the first application of normal distribution as a central role in probability theory and statistics. Normal distribution is an important tool to approximate the probability distribution of the average of independent random variables. A continues random variable has a normal distribution with parameters μ and $\sigma^2 > 0$ if its probability density function f is given by,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad \textbf{(12)}$$

for $-\infty < x < \infty$.

This distribution function is denoted as $\mathcal{N}(p_t^g, \sigma)$.

The Cumulative Distribution Function (CDF) records the same probabilities associated with X, but in a different way. The CDF $F(x)$ is defined by

$$F(x) = P(X \le x). \quad \textbf{(13)}$$

$F(x)$ gives the "accumulated" probability "up to x" and be able to be seen immediately how PDF and CDF are related and defined by [14],

$$F(x) = P(X \le x) = \int_{\infty}^{x} f(t)dt. \quad \textbf{(14)}$$

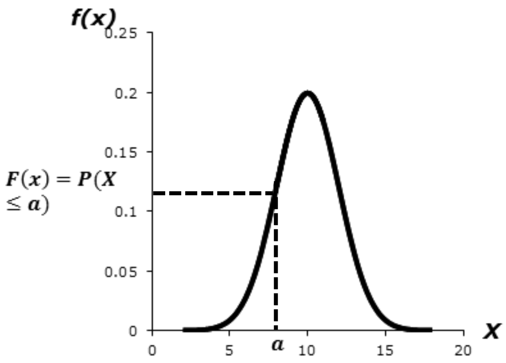


Fig. 4. A graphical representation of the relationship between PDF and CDF.

If X has an $\mathcal{N}(p_t^g, \sigma)$ distribution, then its distribution function is given by,

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \quad \textbf{(15)}$$

or

$$F(x) = \frac{1}{2}[1 + \mathrm{erf}(\frac{x-\mu}{\sigma\sqrt{2}})], \quad \textbf{(16)}$$

for $-\infty < x < \infty$.

5 PSO/ACO HYBRIDIZED ALGORITHMS

Both PSO and ACO adapt swarm intelligence metaheuristics which is based on population global search and co-operative biologically inspirited algorithm motivated by social analogy [16]. 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 [17].

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. And thus in this study, a simple pheromone-guided mechanism is explored to improve the performance of PSO method for optimization of multimodal continuous functions [16].

5.1 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 [16].

Each particle's movement is the composition of an initial random velocity and two randomly weighted influences; individuality, the tendency to return to the particle's best position P_{best}, and sociality, the tendency to move forwards the best previous position of the neighborhoods G_{best}.

Particles update their positions and velocities as shown below:

$$v_{t+1}^{i} = w_t v_t^i + c_1 r_1 \left(p_t^i - x_t^i\right) + c_2 r_2 (p_t^g - x_t^i), \quad \textbf{(17)}$$

$$x_{t+1}^{i} = x_t^i + v_{t+1}^{i}. \quad \textbf{(18)}$$

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 $[-v_{max}, v_{max}]$ to control excessive roaming of particles outside the search space; p_t^g is the global best-found position among all particles in the swarm up to iteration count t and forms the social contribution to the velocity vector; r_1 and r_2 are random numbers uniformly distributed in the interval (0, 1), while c_1 and c_2 are the cognitive and social scaling parameters, respectively; w_t is the particle inertia, which is reduced dynamically to decrease the search area in a gradual fashion by Shi et al. in [16].

The variable w_t is updated along with the iterations in (10).

$$w_t = (w_{max} - w_{min}) * \frac{t_{max} - t}{t_{max}} + w_{min}, \quad \textbf{(19)}$$

where, w_{max} and w_{min} denote the maximum and minimum of w_t respectively, variying from 1.4 to 0.4; t_{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^g of the swarm. This process is repeated until iteration count $t = t_{max}$ (a user-defined stopping criterion is reached).

Attributes of PSO consists of P denotes the number of particles in the population; $f(x_t^i)$represents the objective function value of particle i at position x and calculated as,

$$f(x_t^i) = \alpha * \gamma_{x_t^i}(D) + \beta * \frac{|C| - |x_t^i|}{|C|} \quad , \quad \textbf{(20)}$$

where $\gamma_{x_t^i}(D)$ is the classification quality of particle condition attribute set x_t^i relative to decision table 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. α and β parameters correspond to importance of classification quality and subset length, $\alpha \in [0,1]$ and $\beta = 1 - \alpha$.

Comparison of PSO with other optimization techniques, such as GA has been discussed in [17] and the result shows that PSO is quicker in locating the optimal solution. In general, it can find the optimal solution within tens of generations and search for every particle space [18] with maximum number of iteration. If exhaustive search is used to find the optimal reduct in the dataset DNA, there will be tens of thousands of candidate subsets, which are impossible to execute [19].

PSO techniques can also generate high-quality solutions within shorter calculation time and stable convergence characteristics than other stochastic methods [20]. PSO has shown fast convergence speed and global search ability.

However, Suganthan in [17] mentioned the major drawback of PSO, like in other heuristic optimization techniques, is that it has a slow fine tuning ability of solution quality. PSO is also a variant of stochastic optimization techniques, which is requiring relatively a longer computation time than mathematical approaches. 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 [21].

Shi and Eberhart proved that PSO converges fast under all cases but will slow its convergence speed down when reaching the optima [17]. This may be due to the use of the linearly decreasing inertia weight. By using the linearly decreasing inertia weight, the PSO is lacking of global search ability at the end of run, even when the global search ability is required to jump out of the local minimum in some cases [22].

5.2 Ant Colony Optimization (ACO)

ACO studies the concept of “the emergent collective intelligence of groups of simple agents”. As [23] discussed ACO algorithm was motivated by ants social behavior. Ants have no sight and are capable of finding the shortest route between a food source and their nest when moving from one place to another. Ants deposit substance, called pheromone on the ground to form a trail while walking from food sources to the nest and vice versa.

When choosing their way, they tend to choose, with high probability, paths marked by strong pheromone concentrations to decide shorter paths by smelling the pheromone [23]. ACO is particularly attractive for feature selection since there is no heuristic information that can guide the search to the optimal minimal subset every time.

Besides, if features are represented as a graph, ants can discover the best feature combinations as they traverse the graph [18]. Dre´o, et al. and Socha in [22] found that ants behave as social insects that directly more toward the survival of the colony as a whole than that of a single individual of the colony. Indirect co-operative foraging process of ants is very interesting behavior to be adopted in searching problem of PSO.

In the first stage, it follows the PSO algorithm for global optimization. ACO algorithm was implemented in the second stage. ACO works as a local search to attain rapidly feasible solution space, where the ants apply pheromone-guided mechanism to update the positions found by the particles in the first stage. The ants also use their capability to locate their food resources found by their mates and proven those behaviors to stimulate the optimization of ant foraging behavior in ACO.

The implementation of ACO in the second stage of the proposed PSO/ACO approach was based on the studies by Angeline in [22]. It has been proven that PSO was able to discover reasonable quality solutions much faster than other evolutionary algorithms. However, PSO does not possess the ability to improve upon the quality of the solutions as the number of generations is increased.

6 ANT SWARM OPTIMIZATION FOR ROUGH REDUCTS (ASORR)

Consider a large feature space full of feature subsets. Each feature subset can be seen as a point or position in such a space. If there are N total features, then there will be 2^N kinds of subset, different from each other in the length and features contained in each subset. The optimal position is the subset with least length and highest classification quality.

Now a particle swarm is put into this feature space, each particle takes one position. The particles fly in this space, their goal is to fly to the best position.
Over time, they change their position, communicate with each other, and search around the local best and global best position. Eventually, they should converge on good, possibly optimal, positions. It is this exploration ability of particle swarms that should better equip it to perform feature selection and discover optimal subsets.

6.1 Representation of position

The particle's position is represented as binary bit strings of length N, where N is the total number of attributes. Every bit represents an attribute, the value '1' means the corresponding attribute is selected while '0' not selected. Each position is an attribute subset.

6.2 Representation of velocity

The velocity of each particle is represented as a positive integer, varying between 1 and V_{max}. It implies how many of the particle's bits (features) should be changed, at a particular moment in time, to be the same as that of the global best position, i.e. the velocity of the particle flying toward the best position. The number of different bits between two particles relates to the difference between their positions.

For example, $P_{gbest} = [1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1]$, $p_t^i = [0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$. The difference between $Gbest_t$ and the particle's current position is $P_{gbest} - p_t^i = [1 - 1\ 1\ 1\ 0 - 1\ 1\ -1\ 0\ 0]$. A value of 1 indicates that compared with the best position, this bit (feature) should be selected but is not, which will decrease classification quality and lead to a lower fitness value. Assume that the number of 1's is a.
On the other hand, a value of -1 indicates that, compared with the best position, this bit should not be selected, but is selected. Redundant features will make the length of the subset longer and lead to a lower fitness value. The number of -1's is b. The value of $(a - b)$ is used to express the distance between two positions; $(a - b)$ may be

positive or negative.Such variation makes particles exhibit an exploration ability within the solution space. In this example, $(a-b)=4-3=1$, so $p^g - p_t^i = 1$.

6.3 Representation of PDF and CDF in ASSORR

In the application of Ant Swarm approach, a simple pheromone-guided search 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 [11] and [23] faster. The proposed ACO algorithm from the previous research [11] handles P ants as equal to the number of particles in PSO.

Each i ant generates a solution, z_t^i around the global best-found position among all particles in the swarm, p_t^g up to the iteration count, t as

$$z_t^i = \mathcal{N}(p_t^g, \sigma). \qquad \textbf{(21)}$$

In (20), the algorithm generates components of solution vector z_t^i, which satisfy Gaussian distributions as according to (10). This distribution function has properties to determine the probabilities of z_t^i, which are mean $\mu = p_t^g$ and standard deviation σ, where, initially at t = 1 value of σ = 1 and is updated at the end of each iteration as $\sigma = \sigma \times d$, where, d is a parameter in (0.25, 0.997) and if σ < σ_{min} then σ = σ_{min}, where, σ_{min} is a parameter in (10^{-2}, 10^{-4}).

In this stage, PDF and CDF were applied and compared using the same components as above. Density function will adapt the probabilities associated areas with random variables in (9). Otherwise, distribution function will calculate the accumulative probabilities up to the same random variables of PDF in (13). And then according to each of function computation of z_t^i in (8), evaluate objective function value $f(z_t^i)$ using z_t^i in (4) and replace position x_t^i the current position of particle i in the swarm if $f(z_t^i) < f(x_t^i)$ as $x_t^i = z_t^i$ and $f(x_t^i) = f(z_t^i)$ [10].

This simple pheromone-guided mechanism considers, there is highest density or distribution of trails (single pheromone spot) at the global best solution p_t^g of the swarm at any iteration $t+1$ in each stage of ACO implementation and all ants P search for better solutions in the neighborhood of the global best solution [11] and this process is repeated until iteration $t = t_{max}$.

In the beginning of the search process, ants explore larger search area in the neighborhood of p_t^g due to the high value of standard deviation $\sigma$and intensify the search around p_t^g. Thus, ACO not only helps PSO to efficiently perform global exploration for rapidly attaining the feasible solution space but also to effectively reach the optimal fitness value to gain better reducts, as the algorithm progresses as the following algorithms [15]:

```
Step 1: Initialize Optimization
1.1  Initialize algorithm
     constants $t_{max}$, $P$, and $\{0,1\}^m$
     is the $m$-dimensional
     Boolean particle space.
1.2  Calculate the inertia weight
     of each particle space in
     (19).
```

```
1.3  Initialize randomly all
     particle positions $x_t^i$ and
     velocities $v_t^i$.
1.4  Initialize the positive
     acceleration constants $c_1, c_2$
     and $MaxFit$ as the maximum
     fitness value.
```

Step 2: Perform Optimization (Initialization)

```
2.1  While (t < maxIteration)
  2.1.1 For each particle
    2.1.1.1 Evaluate objective
            function value $f(x_t^i)$ in
            (20)
    2.1.1.2 Assign $p_t^i = x_t^i$ with
            $Pbest_t^i = f(x_t^i)$, $i = 1, \cdots, P$
    2.1.1.3 Evaluate $f_t^{best}(p_t^{best}) = max\{Pbest_t^1, Pbest_t^2, \cdots, Pbest_t^P\}$
    2.1.1.4 If $Pbest_t^i$ > the best
            fitness value
            $f_t^{best}(p_t^{best})$ in history
            then $Gbest_t = f_t^{best}(p_t^{best})$
    2.1.1.5 Assign $p_t^g = p_t^{best}$
  2.1.2 End of For
2.2  End of While
```

Step 3: Perform Optimization (Update the positions and velocities)

```
3.1  While (t < maxIteration)
  3.1.1 For each particle
    3.1.1.1 Update particle
            position $x_t^i$ and
            velocity $v_t^i$ according
            (18) and (19) to all $P$
            particles
    3.1.1.2 Evaluate objective
            function value $f(x_t^i)$ in
            (20)
    3.1.1.3 Generate $P$ solutions
            $z_t^i$ using (21)
    3.1.1.4 Evaluate objective
            function value $f(z_t^i)$ in
            (20)
    3.1.1.5 If $f(z_t^i) > f(x_t^i)$ then
            $f(z_t^i) = f(x_t^i)$ and $z_t^i = x_t^i$
    3.1.1.6 If $Pbest_t^i < f(x_t^i)$ then
            update $Pbest_t^i = f(x_t^i)$ and
            $p_t^i = x_t^i$
    3.1.1.7 Evaluate $f_t^{best}(p_t^{best}) = max\{Pbest_t^1, Pbest_t^2, \cdots, Pbest_t^P\}$
    3.1.1.8 If $Pbest_t^i$ > the best
            fitness value
            $f_t^{best}(p_t^{best})$ in history
            then $Gbest_t = f_t^{best}(p_t^{best})$ and $p_t^g = p_t^{best}$
  3.1.2 End of For
3.2  End of While
```

Step 4: Report best solution p^g as P_{gbest} global best position of the swarm with objective function value $f(p^g)$

7 EXPERIMENTAL RESULTS

The performance of the proposed enhanced Ant Swarm algorithm for global optimization function has been tested on several well-unknown benchmark multimodal problems [15]. All the test functions are multimodal in nature. Because of the characteristics, it is difficult to seek for the global minima. Particle Swarm Ant Colony Optimization (PSACO) algorithm parameter settings used in all the simulations is given as: number of particles, P = 10; cognitive and social scaling parameters, $c1$ = 2, $c2$ = 2; maximum and minimum values of inertia weights, w_{max} = 0.7, w_{min} = 0.4; maximum number of iterations, t_{max} = 100 * n, n is the size of solution vector. Implementation of ASORR has been tested on14 datasets.

The experimental results are reported based on the number of reducts and iterations, fitness values, and the classification accuracy for performance analysis are shown in Table 1. Both PDF and CDF algorithms implementations use Naïve Bayes to extract rules from the data for rule induction in classification. Ten-fold cross validation was applied to estimate the classification accuracy.

The two types of enhanced Ant Swarm algorithms are compared and the best solutions of both algorithms found are presented. Experimental results have shown that PDF has more optimal results than CDF in most of datasets. Table 1 reports the result produced by PDF in a number of reducts evaluation is smaller than CDF but in average, both function yield the same results. However, a number of reducts is not able to determine which function devises better result, in terms of finding the optimal fitness value and gain higher classification accuracy.

The fitness values of each function reported in Table 1 has shown that PDF can achieve better values and prove that a smaller number of reducts will increase the fitness value. As shown in Table 1, PDF achieved better results than CDF by reducing 5.19 iterations for number of iteration results analysis in average for 10 independent runs. Table 1 shows the results in terms of classification accuracy where PDF takes more significant optimal solution and has the same analysis results with its fitness value performances, and gain better accuracy than CDF.

Thus, based on the results obtained and presented, the implementation of ASORR using PDF is having better performances, both in gaining higher fitness value and better quality of reducts as compared to CDF. However, some previous studies [8, 20] also have explored Ant Swarm algorithm in other implementation field and all of them applied PDF as solution for the optimal results.

8 CONCLUSIONS

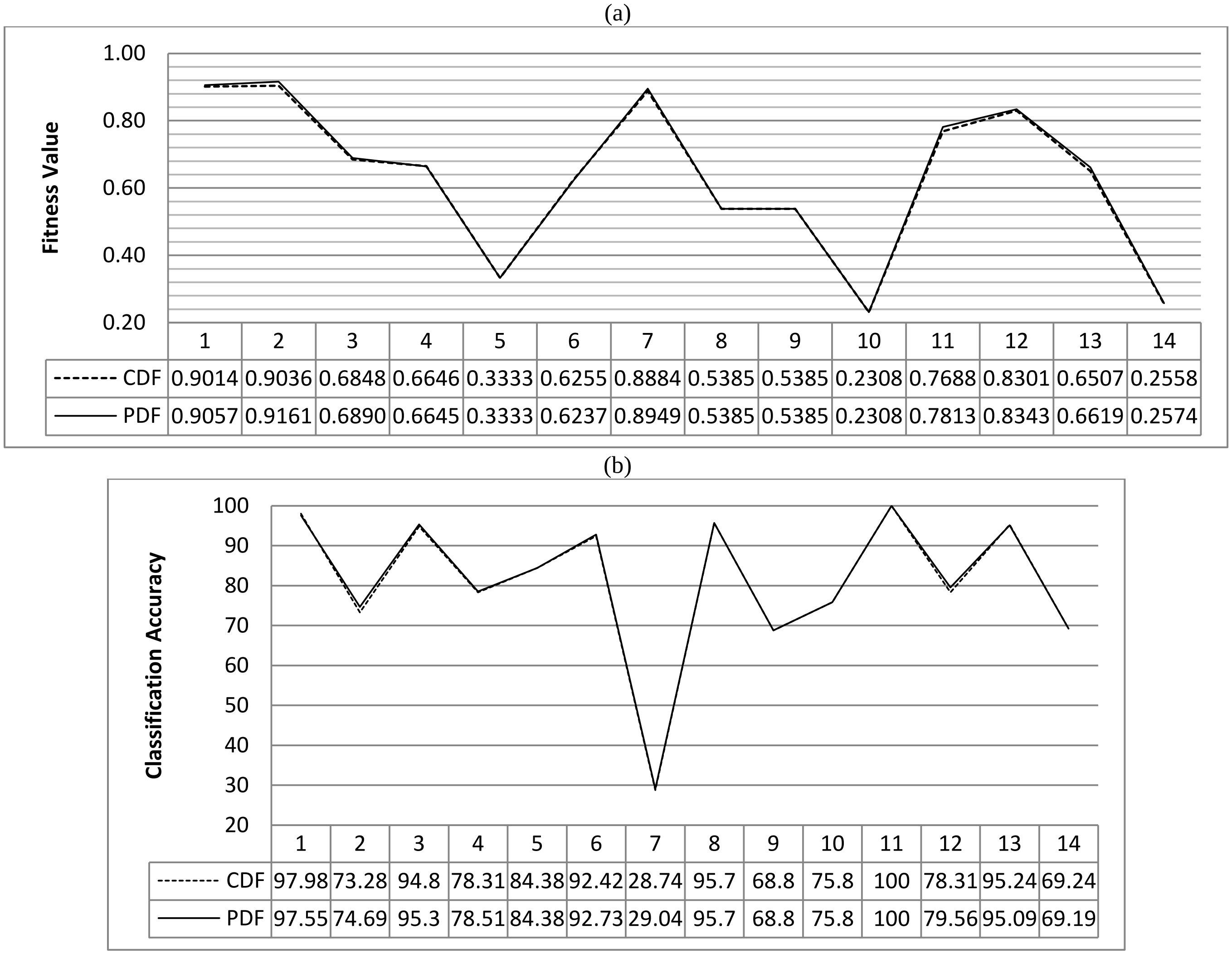
An extensive comparative study on rough reducts optimization has been presented. This paper compared the merits of PDF and CDF focusing on PSO/ACO enhanced rough reducts. A simple-pheromone-guided mechanism is implemented as local search by using Gaussian distribution functions, the PDF and the CDF, to improve the performance of PSO algorithm. The experiments have shown that PDF is better than CDF in terms of generating smaller number of reducts, improved fitness value, lower number of iterations, and higher classification accuracy. Thus, the experimental results have also provided further justification on previous studies which have implemented PDF instead of CDF into Ant Swarm algorithm in various domains. The initial results of PDF are promising in most of the tested datasets. Hence, future works are to test on the enhanced ASORR algorithm with PDF in various domains to validate its performance in yielding the optimal reducts set.

9 ACKNOWLEDGEMENT

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.

10 REFERENCES

1. Wang, Xuren, Xu, Rongsheng and Wang, Wansen. Rough Set Theory: Application in Electronic Commerce Data Mining. In: IEEE Computer Society. Proc.s of the IEEE/WIC/ACM Intl. Conf. on Web Intelligence (2004)
2. Pawlak, Zdzislaw.: Some Issues on Rough Sets. In: Peters, J.F., et al. (Eds.) Transactions on Rough Sets I. LNCS, 3100, pp. 1—58 Springer, Heidelberg (2004)
3. Choo, Yun-Huoy; Bakar, Azuraliza Abu and Hamdan, Abdul Razak: The fitness-rough: A new attribute reduction method based on statistical and rough set theory. Int. J. of Intelligent Data Analysis. 12, 73--87 (2008)
4. Wang, Jianguo, Meng, Guoyan and Zheng, Xiaolong.: The Attribute Reduce Based on Rough Sets and SAT Algorithm. Shanghai : IEEE. Intelligent Information Technology Application. Vol. 1 (2008)
5. Barceló, A. Juan.: Computational Intelligence in Archaeology. In : Idea Group Inc (IGI), 1599044897 (2008)
6. Pawlak, Zdzisław, and Skowron, Andrzej.: Rudiments of Rough Sets. Information Science. 177, 3--27 (2007)
7. Pawlak, Z.: Rough Sets. University of Information Technology and Management, Poland (2004)
8. Pawlak, Z.: Rough Sets. Int. J. of Information and Computer Sciences. 11, 341--356 (1982)
9. Pawlak, Z., and Skowron, A.: Rough Membership Function. In: Yeager, R.E., Fedrizzi, M., and Kacprzyk, J. (Eds.), Advances in the Dempster-Schafer of Evidence, pp. 251--271. New York (1994)
10. Pawlak, Z. and Skowron, A.: Rough Sets and Boolean Reasoning. Elsevier, 177, 41--73 (2007)
11. Socha, K. and Dorigo M. Ant colony optimization for continuous domains, The European Journal of Operational Research, 185, 1155--73 (2008)
12. Pratiwi, L., et al.: A Framework of Rough Reducts Optimization Based on PSO/ACO Hybridized Algorithm. In: 3rd Data Mining and Optimization Conference, pp. 153--159. IEEE, Putrajaya (2011)
13. Freiwald, Ron. The Cumulative Distribution Function for a Random Variable X. Washington University, www.math.wustl.edu/~freiwald/Math132/cdf.pdf. Retrieved on March 15, 2011
14. The Probability Density and Cumulative Distribution Functions. ReliaSoft, http://www.weibull.com/LifeDataWeb/the_probability_density_and_cumulative_distribution_functions.htm. Retrieved on March 14, 2011
15. Ye, Dongyi, et al.: A New Fitness Function for Solving Minimum Attribute Reduction Problem. In: RSKT'10. LNCS, 6401, pp. 118--125. Springer, Heidelberg (2010)
16. Shelokar, P.S., et al.: Particle Swarm and Ant Colony Algorithms Hybridized for Improved Continues Optimization. 1, In : Elsevier Inc., Applied Mathematics and Computation, Vol. 188, pp. 129--142 (2007)
17. Kadirgama, K., Noor, M.M., and Alla, A.N.A.,: Response Ant Colony Optimization of End Milling Surface Roughness. J.s of Sensors. 10, pp. 2054--2063 (2010)
18. Shi, Y and Eberhart, R. C.: Empirical study of particle swarm optimization. In : IEEE. Int. Congr. Evolutionary Computation. vol. 1, pp. 101--106 (2001)
19. Wang, Xiangyang, et al.: Feature Selection Based on Rough Sets and Particel Swarm Optimization., 4, New York : Elsevier Science Inc., vol. 28 (2007)
20. Lee, K.Y., and Park, J. B.: Application of Particle Swarm Optimization to Economic Dispatch Problem: Advantages and Disadvantages. 1, In : IEEE, PSCE (2006)
21. Yang, Qin and Wang, Danyang.: An Improved Particle Swarm Optimization. In : IEEE, BMEI (2009)
22. Salehizadeh, S.M.A., et al.: Local Optima Avoidable Particle Swarm Optimization. In: Swarm Intelligence Symposium, pp. 16--21. IEEE, Nashville (2009)
23. Chen, Yumin, Miao, Duoqian and Wang, Ruizhi.: A Rough Set Approach to Feature Selection Based on Ant Colony Optimization. 3, In : Elsevier, vol. 31 (2010) Springer (2007)

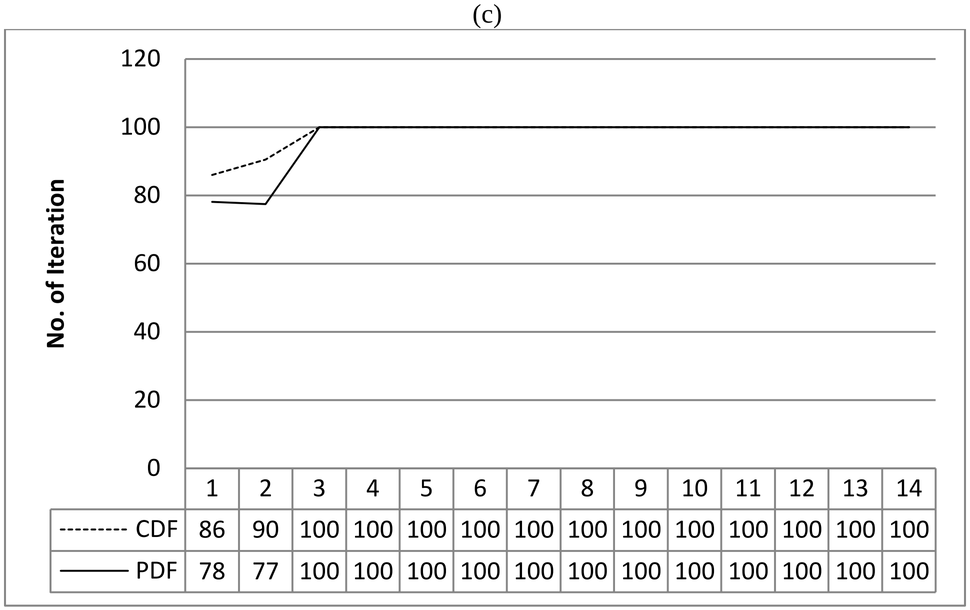


(a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
------- CDF	0.9014	0.9036	0.6848	0.6646	0.3333	0.6255	0.8884	0.5385	0.5385	0.2308	0.7688	0.8301	0.6507	0.2558
——— PDF	0.9057	0.9161	0.6890	0.6645	0.3333	0.6237	0.8949	0.5385	0.5385	0.2308	0.7813	0.8343	0.6619	0.2574

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
--------- CDF	97.98	73.28	94.8	78.31	84.38	92.42	28.74	95.7	68.8	75.8	100	78.31	95.24	69.24
——— PDF	97.55	74.69	95.3	78.51	84.38	92.73	29.04	95.7	68.8	75.8	100	79.56	95.09	69.19



(c)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
--------- CDF	86	90	100	100	100	100	100	100	100	100	100	100	100	100
——— PDF	78	77	100	100	100	100	100	100	100	100	100	100	100	100

Fig. 5. ASORR Experimental Results on Line Graph based on Fitness Value (a), Classification Accuracy (%) (b), and No. of Iteration (c)

Table 1. ASORR Experimental Results on No. of Reducts, Fitness Value, No. of Iteration, and Classification Accuracy (%)

				Criteria										
No.	Dataset	Features	Instances	No. of Reducts		Fitness Value			No. of Iteration			Classification Accuracy		
				ASORSAR		ASORSAR		Best Result	ASORSAR		Best Result	ASORSAR		Best Result
				CDF	PDF	CDF	PDF		CDF	PDF		CDF	PDF	
1	Soybean-small	35	47	3	3	0.9014	0.9057	PDF	86	78	PDF	97.98	97.55	CDF
2	Lung	56	32	5	5	0.9036	0.9161	PDF	90	77	PDF	73.28	74.69	PDF
3	Zoo	16	101	4	4	0.6848	0.6890	PDF	100	100	None	94.8	95.3	PDF
4	Lymphography	18	148	5	5	0.6646	0.6645	CDF	100	100	None	78.31	78.51	PDF
5	Corral	6	64	4	4	0.3333	0.3333	None	100	100	None	84.38	84.38	None
6	Vote	16	300	5	5	0.6255	0.6237	CDF	100	100	None	92.42	92.73	PDF
7	DNA	57	318	6	6	0.8884	0.8949	PDF	100	100	None	28.74	29.04	PDF
8	M-of-N	13	1000	6	6	0.5385	0.5385	None	100	100	None	95.7	95.7	None
9	Exactly	13	1000	6	6	0.5385	0.5385	None	100	100	None	68.8	68.8	None
10	Exactly2	13	1000	10	10	0.2308	0.2308	None	100	100	None	75.8	75.8	None
11	Led	24	2000	6	5	0.7688	0.7813	PDF	100	100	None	100	100	None
12	Mushroom	22	8124	3	3	0.8301	0.8343	PDF	100	100	None	78.31	79.56	PDF
13	Breastcancer	9	699	3	2	0.6507	0.6619	PDF	100	100	None	95.24	95.09	CDF
14	Tic-Tac-Toe	9	958	6	6	0.2558	0.2574	PDF	100	100	None	69.24	69.19	CDF
	Average			5	5	0.6296	0.6336	PDF	98	97	PDF	80.93	81.17	PDF