Optimisation of Thresholds in Probabilistic Rough Sets with Artificial Bee Colony Algorithm

T. V. Soumya and M. K. Sabu

Department of Computer Applications, Cochin University of Science and Technology, Kochi, India

ABSTRACT
The Probabilistic Rough Sets (PRS) theory determines the certainty of an object’s inclusion into a class, resulting in the division of the entire data set into three regions under a concept. These regions, namely the positive, negative and boundary regions, are generated using an evaluation function and threshold values. The threshold optimisation and the construction and interpretation of an evaluation function offer various methods in the background. Even though most of the methods in the PRS follow an iterative strategy, they lack a common framework, usually affecting the comparison and overall performance evaluation among these methods. This proposed work aims to minimise the uncertainty in three regions via optimising the thresholds using the Artificial Bee Colony (ABC) algorithm. The ABC algorithm is adapted to generate a common framework that results in different optimal pairs of thresholds with a minimum number of iterations. By considering the probabilistic information about an equivalence class structure, we compare the results obtained from the proposed approach with the state-of-the-art methods like Information-Theoretic Rough Sets, Game-Theoretic Rough Sets and Genetic Algorithm-based optimisation. The results reveal that the proposed algorithm outperforms existing techniques and leads to a superior method for threshold optimisation in the PRS.

ARTICLE HISTORY
Received 5 April 2021
Revised 21 October 2021
Accepted 2 November 2021

KEYWORDS
Probabilistic Rough Sets; Optimisation; Entropy; Three-way Decision; Artificial Bee Colony Algorithm

1. Introduction
The PRS is a significant theory under the Three-Way Decisions (TWD) for handling uncertainty in the data from a probabilistic perspective [1–4]. The optimal pair of thresholds in the area of PRS is the predominantly handled subject with heterogeneous algorithms that work iteratively for optimising our objective to find the best. Mostly, the algorithms in PRS deal with cost, uncertainty, impurity, correlation and variance as the objectives that are minimising or maximising as the case may be [5–10]. The theories, namely Decision-Theoretic Rough Sets (DTRS), Information-Theoretic Rough Sets (ITRS) and Game-Theoretic Rough Sets (GTRS), are the major contributions under the PRS for the optimisation of various objectives [11–13]. Methods on the basis of Chi-Square Statistic, Variance and Gini-Index are also contributed to the optimisation of threshold pairs [9,10,14,15]. Other existing models employed classical methods such as Gradient-Descent and Genetic Algorithm (GA) to get
an optimal trisection by considering entropy as the objective function [5,12]. Various methods are contributed to play a major role in the optimisation of objectives in the area of PRS [5,9,10,12,14,15].

Mostly, the methods in the PRS for threshold selection follow an iterative strategy, and thus the models obtain better results comparatively [5]. However, most of the methods in the PRS used a set of threshold pairs initially and searched for the best threshold pair from this set using an iterative strategy [6,9,16]. This tedious exhaustive search process takes time in delivering the desired results. Various optimisation techniques are introduced to optimise the thresholds as a solution to this issue. In the existing methods, the genetic algorithm, gradient-descent algorithm and GTRS-based learning method generate optimised results [5,11–13]. The comparison of the efficiency among these models is important as there exist various models for determining the thresholds. However, Babar Majeed et al. notify that a model with the iterative mechanism in PRS is difficult to compare directly with the other methods in PRS [5]. This is because it is difficult to distinguish the obtained performance is due to the iterative mechanism employed or the capability of the model, i.e. it affects the direct comparison and performance evaluation among these models [5]. GA-based optimisation provides a general framework for optimising the objectives; even so, our proposed work outperforms this method.

The proposed work put forward a new approach based on the ABC algorithm to optimise the thresholds via minimising the uncertainty that exists in three regions. Therefore, we apply the classical ABC method, which includes various functional phases such as food source initialisation, employed bee phase, onlooker bee phase and scout bee phase [17,18]. The food source initialisation phase is employed to generate an initial set of threshold pairs. The employed bee phase assists in exploring a new set of threshold pairs from the initial set to find out the effective thresholds. The onlooker bee phase focuses on the exploitation of new thresholds from the effective thresholds to find the optimal one. Finally, the scout bee phase replaces the non-updated threshold pairs with the new pair. These phases are working iteratively to obtain the optimal threshold pairs.

Using the ABC algorithm, the neighbourhood exploration of the threshold pairs is employed for optimising the objectives. To be specific, the ABC algorithm for optimising the PRS provides a space for the betterment of current results with minimum iterations. Moreover, this algorithm provides a common framework such that it is applicable for different objective functions. This enables us to compare and evaluate the performance of different PRS models directly by using various fitness functions. This approach also extends and enhances the ITRS model with capabilities of the ABC algorithm and produces different possible optimum thresholds. These results are generated from the extended search space by the neighbourhood search mechanism of the ABC algorithm apart from the initial search space. Like [6], we use the probabilistic information of equivalence classes based on a concept to attest our argument, and the experiment gives promising results. The obtained results are then compared with the ITRS, GTRS and GA-based optimisation [5,6]. This work demonstrates that the ABC-based optimisation in PRS is the finest and effective method as it provides better results compared to the existing methods. Moreover, the work gives optimised results with a minimum number of iterations.

The remaining part of this paper includes Section 2, which presents the background theory of our proposed work. The traditional ABC algorithm for threshold optimisation and the functional flow of different phases are detailed thoroughly, assisting with a flowchart
in Section 3. The experimental results are displayed in Section 4 and the corresponding discussion is explained in Section 5. Finally, the paper concludes with Section 6.

2. Background

First, this section explains the theories associated with the PRS and various methods for threshold selection in a nutshell. Then, the theories like ITRS, GTRS, GA-based optimisation and ABC algorithm are explained in detail.

2.1. PRS

This theory was introduced by Yiyu Yao et al. to generate the three regions quantitatively [19,20]. That means the PRS generates three regions elicited from the threshold values along with the probabilistic strategy [21]. An equivalence class is classified into a region based on the following equations:

\[
P_\alpha(C) = \bigcup \{[X]|Pr(C|[X]) \geq \alpha\},
\]

\[
N_\beta(C) = \bigcup \{[X]|Pr(C|[X]) \leq \beta\},
\]

\[
B_{(\alpha,\beta)}(C) = \bigcup \{[X]|\beta < Pr(C|[X]) < \alpha\}.
\]

where \([X]\) is an equivalence class in which each object follows an indiscernibility relation I [21–23]. The conditional probability (\(Pr(C|[X])\)) is expressed in terms of confidence level of an object which belongs to \([X]\) for a concept \(C\) [13]. This confidence level is compared with the \(\alpha\) and \(\beta\) values and the three regions namely, positive region (\(P_\alpha\)), negative region (\(N_\beta\)) and boundary region (\(B_{(\alpha,\beta)}\)) are formed using Equation (1) [24,25].

2.2. Threshold Optimisation Methods in the PRS

The first theory, the DTRS in the PRS, was introduced by Yiyu Yao et al. to find the thresholds based on the cost or risk associated with the actions of deciding an object \(x \in C\) to be mapped onto the positive region, negative region or boundary region where \(C\) is a concept [7,19,26]. This theory provides an efficient way of calculating threshold values with optimum cost [11]. Then, Xiaofei Deng et al. proposed ITRS theory using an exhaustive strategy that chooses the thresholds from a set of thresholds given by the experimenter and affixes the threshold values at which they attain minimum uncertainty. This theory follows the Shannon entropy as the objective function to minimise the uncertainty [6,12,27]. ITRS was introduced with a gradient descent optimisation process in order to avoid this tedious selection process [12]. Since this is a general approach, the correctness and efficiency of this framework were not tested by the author [20]. Later, ITRS was used in a different perspective to optimise impurity by using the Gini index as the objective function and executing it iteratively [27]. It selects the best pair of thresholds at which the impurity is minimum [8,16]. Other existing model includes GTRS which was proposed by Joseph P Herbert et al. to optimise multiple objectives simultaneously in a trade-off perspective [13,28]. By varying the threshold values, both objectives are examined and stopped at a Nash equilibrium [13]. In [29], an evolutionary algorithm called Genetic algorithm was effectively utilised for the
optimisation of uncertainty. The basic operations like selection, mutation and crossover are employed for the purpose of optimisation [5]. Optimisation of other qualities such as correlation and variance is introduced in the later stage [9,10]. By employing the Chi-Square Statistic as the objective function, Cong Gao et al. maximise the correlation between the trisection and classification in order to find the optimal pair of thresholds [9]. Also, the ratio of the variance of the inter-regions and intra-regions was optimised by Nouman Azam at a certain threshold level in [10].

2.3. ITRS

This model was introduced by XiaoFei Deng et al. to generate optimal trisection by deriving the information in terms of uncertainty from the three regions which is generated at each threshold pair. Information entropy is a measure of uncertainty that was proposed by Claude Elwood Shannon in 1948 [30]. This measure is accustomed to finding the uncertainty in a concept with respect to the three regions [25,31]. The partition of the universe with respect to the concept C is defined by the following equation:

\[ H(\pi_C) = -\sum_{C \in \{C, C^C\}} \pi_C \log \pi_C \]

In Equation (2), the conditional probabilities, \(Pr(C|P_\alpha(C))\), \(Pr(C|N_\beta(C))\) and \(Pr(C|B(\alpha,\beta)(C))\) help to determine the amount of correctly classified objects in each region based on the concept C [6,20]. Suppose \(\Delta\) is a region, then the conditional probability of the region \(Pr(C|\Delta(\alpha,\beta)(C))\) is defined by

\[ Pr(C|\Delta(\alpha,\beta)(C)) = \frac{|C \cap \Delta(\alpha,\beta)(C)|}{|\Delta(\alpha,\beta)(C)|} \]

From Equation (2), \(H_P\), \(H_N\) and \(H_B\) represent the entropy of the positive region, negative region and boundary region respectively [5,13]. Then, the overall conditional entropy (OCE) is defined by the following equation:

\[ H(\pi_C|\pi(\alpha,\beta)) = W_1H_P + W_2H_N + W_3H_B \]

In Equation (4), \(W_1\), \(W_2\) and \(W_3\) are the weights associated with the trisection [5,13]. In this regard, they are the probabilities of each region such as \(Pr(P_\alpha(C))\), \(Pr(N_\beta(C))\) and \(Pr(B(\alpha,\beta)(C))\) [20]. The probability of the region \(\Delta(\alpha,\beta)(C)\) is generated by:

\[ Pr(\Delta(\alpha,\beta)(C)) = \frac{|\Delta(\alpha,\beta)(C)|}{|U|} \]

In ITRS, the OCE is minimised to obtain the optimal pair of thresholds [6,20]. Like [6,9], the following equation represents the process of generating the optimal pair of thresholds, say...
\((\alpha^*, \beta^*)\).

\[ (\alpha^*, \beta^*) = \arg\min_{(\alpha, \beta)} H(\pi_C | \pi_{(\alpha, \beta)}(C)) \]  

(6)

### 2.4. GTRS

The pioneer of game theory is John von Neumann, and this is an interesting theory that is applicable in various fields [32–34]. Herbert et al. proposed this theory in rough sets (GTRS) to find optimal pairs of thresholds by optimising multiple criteria simultaneously [14,28,35–37]. For game formulation, factors like players, strategies and payoff functions are considered by the GTRS [13,35]. The players involving in the game with a trade-off point of view represent multiple criteria. Each player follows a strategy to execute the game. The function defined for individual players is called the utility or payoff function. The players, accuracy and generality are the typical examples for multiple criteria competing with each other to attain a balanced state. If we try to improve the accuracy of the model, then its generality becomes less. The model with more generality having less accuracy [5,13,38]. This kind of relationship is followed by the properties of the immediate region and the deferred region (boundary region). The positive and negative regions are collectively called the immediate region. In the GTRS, the uncertainties of the immediate and deferred regions are the players with certain strategies [5,13,38]. For strategy, the effect of change of thresholds on the players is employed [13]. Hence, decreasing \(\alpha\), increasing \(\beta\), decreasing \(\alpha\) and increasing \(\beta\), i.e. \(S_1 = \alpha \downarrow\), \(S_2 = \beta \uparrow\) and \(S_3 = (\alpha \downarrow, \beta \uparrow)\) are the three strategies. The strategy starts from the Pawlak’s threshold pair setting \((\alpha, \beta) = (1, 0)\). Later, payoff functions are defined on the basis of each criterion. From Equation (4), the payoff functions are defined. To compute the possible gain, instead of uncertainty, the certainty values are calculated using Equation (7) [13,15].

\[
CE_{P\alpha} = 1 - W_1 H_P, \\
CE_{N\beta} = 1 - W_2 H_N, \\
CE_{B(\alpha, \beta)} = 1 - W_3 H_B.
\]  

(7)

where \(CE_{P\alpha}\), \(CE_{N\beta}\) and \(CE_{B(\alpha, \beta)}\) calculate the certainty for each region \(P_{\alpha}\), \(N_{\beta}\) and \(B_{(\alpha, \beta)}\), respectively [13,15]. Then, the payoff function for immediate region (M) and deferred region (D) is defined [13].

\[
u_M = (CE_{P\alpha} + CE_{N\beta})/2, \quad u_D = CE_{B(\alpha, \beta)}.
\]  

(8)

In Equation (8), \(u_M\) and \(u_D\) determine the certainties of immediate region and deferred region, respectively. The payoff values, \(u_M\) and \(u_D\), are observed based on strategies \(S_1\), \(S_2\) and \(S_3\), respectively, and a particular pair of thresholds at the Nash equilibrium state is returned as optimum [13,35]. No criteria get ready to change their strategies at the Nash equilibrium state, knowing the strategies of other criteria.

### 2.5. GA-based Optimisation

GA is an evolutionary algorithm that mimics the process of natural selection of fittest candidates through selection, crossover and mutation phases for producing the next
The simplified GA algorithm was introduced by John Henry Holland in [41]. GA-based optimisation of PRS was proposed by Babar Majeed et al. with various functional stages such as follows [5]:

1. Producing initial population
2. Population evaluation
3. Termination criteria
4. Selection of new population
5. Crossover and mutation operations.

At first, the binary encoding mechanism is employed for generating binary strings to represent the \( (\alpha, \beta) \) pairs [5]. The \( \alpha \) values 0.7, 0.8, 0.9 and 1.0 are replaced with the 00, 01, 10 and 11 respectively. Similarly, the \( \beta \) values 0.0, 0.1, 0.2 and 0.3 are replaced with 00, 01, 10 and 11 respectively [5]. Then, a chromosome is generated by combining the encoded \( \alpha \) and \( \beta \) values. Consequently, the chromosome is the four bits of a binary string, i.e. (1.0,0.3) is represented by chromosome 1111 [5]. In the population evaluation, the OCE is the fitness function, and the fitness value for each chromosome is determined. The chromosome which generates the best OCE value is chosen for the further process to generate effective thresholds. The process proceeds until a termination criterion is either at a certain level of fitness or no remarkable improvement in the successive iterations. The selection of the new population focuses on the chromosomes with better OCE value to get the chance for further processing. The roulette wheel mechanism helps to find out the chromosomes with higher OCE probabilities [5]. Therefore, the chromosomes are selected on the basis of higher order of their OCE probabilities. Later, the new chromosomes are generated through the crossover operation. Also, by inverting bits in the chromosome, mutation operation avoids unnecessary iterations when the newly generated chromosomes are similar to the first and last chromosomes. Stages 3, 4 and 5 are repeated until the termination condition is satisfied [5].

2.6. ABC Algorithm

In the era of Swarm Intelligence, numerous algorithms are available to solve optimisation problems efficiently [42–44]. In 2005, Dervis Karaboga introduced the Artificial Bee Colony Algorithm, which is one of the basic algorithms that imitate the food searching strategy of honey bees [18,45–47]. This algorithm follows a division of labour among various types of bees, like employed bees, onlooker bees and scout bees, where each of them has a significant role in food search [18]. The first phase starts with a collection of food sources which is initialised randomly. From these initialised food sources, each employed bee selects the corresponding food source and explores the quality of the food source to share the information with other bees in the hive. A fitness function is used by the employed bees in order to measure the quality of the food sources. The best quality food source is preserved as the global best food source. The fitness probability of each food source is measured after the employed bee phase [18,46]. Then the onlooker bees are assigned to these food sources based on the computed probability. The outcome of the onlooker bee phase is the exploitation of the assigned food sources by considering their neighbourhood. After each exploitation, the quality of the resulting food source is compared with the global best. The global best is updated if the generated food source is a better quality food source. This
process is continued for all assigned onlooker bees [18,46,47]. Scout bees are assigned to replace the food source with a new one if any of the food sources are exhausted during the exploration as well as the exploitation phase. The entire process of exploration and exploitation is repeated a certain number of times, and the optimal solution is available in the global best [18]. Algorithm 1 explains the successive operations of various functional stages of the ABC algorithm [18,46,47]. The employed bee phase, onlooker bee phase and scout bee phase are working iteratively until the termination condition is satisfied.

Algorithm 1: Classical ABC algorithm

Result: Best food sources
Initial population;
Initialization;
Find the fitness value for each food source;
Find the global best;
while Termination condition is not reached do
  for Each employed bee do
    Exploration of neighbourhood of food source;
    Find the fitness value;
    if Update the new food source then
      Counter value is zero;
    else
      Counter is incremented;
    Update the global best;
  for Each onlooker bee do
    Find the probability of fitness values;
    Assign onlooker bee regarding the order of higher probability;
    Exploitation of neighbourhood of food source;
    Find the fitness value;
    if Update the new food source then
      Counter value is zero;
    else
      Counter is incremented;
    Update the global best;
    if counter ≥ limit then
      Initialize the food source;
      Find the fitness value;
      Update the global best;
    else
      Go for the next onlooker bee;

3. Experimental Framework

First, the optimisation of PRS using the ABC algorithm is explained briefly with the help of a framework in this section. Then, in Section 3.1, each phase is described with the support of mathematical formulae.
As mentioned above, it explains the process in each phase for finding the best food source and how that is effectively mapped to the optimisation of thresholds. The framework mainly contains various functional phases such as food source initialisation, employed bee phase, onlooker bee phase and scout bee phase [18]. In the first phase, the initial population carries a collection of threshold pairs; then, these pairs undergo the process of initialisation. These initialised values are used to calculate the fitness function. We consider the OCE of three regions as our objective function and calculate the corresponding fitness value in our scenario. Then, we select the threshold pair having maximum fitness value and memorise it as the global best.

The employed bee phase focuses on the exploration of these initialised food sources based on their neighbourhood. This phase updates the food source assigned to it if the fitness value of the newly generated food source is better than the current one [18,46,47]. In our scenario, each threshold pair assigned to the employed bees undergo this process of updating if the fitness value of the generated candidate is better than the existing one. At the end of this phase, the maximum fitness value, along with the threshold pair, is stored as the global best.

In the onlooker bee phase, we quantify the quality of the food sources in terms of the probability of the corresponding fitness values. The onlooker bees are assigned based on the higher-order probability of the fitness values. Again, the food sources are exploited on the basis of their neighbourhood. Like employed bees, the same process is followed by the onlooker bees to exploit the new food source and updating the global best [18,46,47]. In the proposed method, the candidate with the higher fitness probability is selected one by one and update the threshold pair by considering the neighbourhood. Then, we select the maximum fitness value at a particular threshold pair and update the global best if the generated candidate is better than the existing one.

The exhausted food sources are gone through the initialisation process, and it is replaced with the new food sources by the scout bees [18,46,47]. Suppose the candidate is not updated continuously in the employed bee phase and onlooker bee phase to a predefined number of searches; the corresponding candidate faces the initialisation process to identify a new threshold pair. The non-updated threshold pair is replaced with the new one. This entire process in three phases is continued up to satisfy the termination condition.

This framework is capable of using different objective functions and makes them comparable. The ABC-based framework follows different functional phases in each cycle, which works collectively to obtain the optimal pair of thresholds. The workflow of the proposed architecture is depicted in Figure 1.

3.1. **ABC Algorithm for Threshold Optimisation**

3.1.1. **Initialisation**

Before starting the execution, we select an initial population consisting of five (α, β) pairs, say (α_s, β_s) based on the limits \(0.5 \leq \alpha_s \leq 1.0\) and \(0 \leq \beta_s < 0.5\), respectively, where \(s\) represents the index of each candidate which starts from 1 to 5. The following two formulae initialise each pair of (α_s, β_s) values as food sources [18]. The initialised (α_s, β_s) pairs are denoted by (α_n, β_n) where \(n = 1, 2, ..., 5\).

\[
\alpha_n = |\min(\alpha_s) + \phi_1(\max(\alpha_s) - \min(\alpha_s))|, \\
\beta_n = |\min(\beta_s) + \phi_2(\max(\beta_s) - \min(\beta_s))|, \tag{9}
\]
In order to generate the $\alpha_n$ and $\beta_n$ values within the specific limits ($0.5 \leq \alpha_n \leq 1.0$, $0 \leq \beta_n < 0.5$), $\phi_1$ and $\phi_2$ are randomly chosen between the limits $0.0 \leq \phi_1 \leq 1.0$ and $-1.0 \leq \phi_2 \leq 0.0$, respectively. For each newly generated pair, we calculate the OCE value using Equation (4). Then, we calculate the fitness value for each $(\alpha_n, \beta_n)$ pair regarding the
following equation [18,46,47]:

\[
f_n = \begin{cases} 
\frac{1}{1 + OCE_n}, & \text{if } OCE_n \geq 0, \\
\frac{1}{1 + |OCE_n|}, & \text{if } OCE_n < 0.
\end{cases}
\] (10)

Our aim is to obtain the optimal \((\alpha, \beta)\) pairs by maximising the fitness value. In this phase, we determine the maximum fitness value based on the initialised threshold pairs \((\alpha_n, \beta_n)\). This fitness value and the corresponding threshold pair are stored as the global best and move to the next phase.

3.1.2. Employed Bee Phase

This phase enrolls each \((\alpha_n, \beta_n)\) as the threshold pairs in the employed bee phase. The new threshold pairs \((\alpha_{new}, \beta_{new})\) are generated based on the \(k\)th neighbourhood using Equation (11) where \(k = 1, 2 \cdots, 5\) [18,46,47].

\[
\begin{align*}
\alpha_{new} &= |\alpha_n + \phi_3 (\alpha_n - \alpha_k)|, \\
\beta_{new} &= |\beta_n + \phi_4 (\beta_n - \beta_k)|,
\end{align*}
\] (11)

where \(\phi_3\) is a random value belongs to \(-1.0 \leq \phi_3 \leq 0.0\) and \(\phi_4\) belongs to \(-1.0 \leq \phi_4 \leq 0.3\). Thus each threshold pair \((\alpha_n, \beta_n)\) is updated only when \((\alpha_{new}, \beta_{new})\) results in better fitness value. Finally, the updated \((\alpha, \beta)\) pair with best fitness value is stored as the global best. The resultant threshold pairs in employed bee phase are denoted by \((\alpha_e, \beta_e)\), which are selected for the onlooker bee phase.

3.1.3. Onlooker Bee Phase

The threshold pairs obtained from the previous step become the threshold pairs to be exploited by the onlooker bees. The selection of these threshold pairs to exploit is based on the fitness probability \(Pr(f_i)\), computed using Equation (12) [18,46,47].

\[
Pr(f_i) = \frac{f_i}{\sum_{i=0}^{m} f_i}
\] (12)

where \(m\) is the total number of threshold pairs. The candidate with the highest fitness probability is the threshold pair selected by the first onlooker bee to exploit. At this phase, as a result of exploitation, each candidate generates new threshold pairs \((\alpha_{new}, \beta_{new})\) using Equation (11), but with a different set of random numbers \(\phi_5\) and \(\phi_6\), generated within the limits \(-1.0 \leq \phi_5 \leq 0.0\) and \(-1.0 \leq \phi_6 \leq 0.3\), respectively. This process is continued for all the neighbours, and the better fitness value thus obtained is compared with the global best. If the generated fitness value is better than the global best, it is replaced with the new fitness value together with \((\alpha_{new}, \beta_{new})\). This process is repeated for all the onlookers, and as an outcome of this phase, we get a possible solution to the problem.

3.1.4. Scout Bee Phase

If a threshold pair is not updated by any of its neighbourhoods in both the employed bee phase and onlooker bee phase, then the counter value is incremented by one during each comparison with its neighbourhood. If a threshold pair is updated, then the counter
Table 1. Probabilistic information of equivalence classes [5,6].

| | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 | X_7 | X_8 |
|---|---|---|---|---|---|---|---|---|
| Pr(X_I) | 0.0177 | 0.1285 | 0.0137 | 0.1352 | 0.0580 | 0.0069 | 0.0498 | 0.1070 |
| Pr(C|X_I) | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.9 | 0.8 | 0.6 |
| | X_9 | X_10 | X_11 | X_12 | X_13 | X_14 | X_15 |
| Pr(X_I) | 0.1155 | 0.0792 | 0.0998 | 0.1299 | 0.0080 | 0.0441 | 0.0067 |
| Pr(C|X_I) | 0.5 | 0.4 | 0.4 | 0.2 | 0.1 | 0.0 | 0.0 |

Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Rough Sets and Knowledge Technology [6] An information-theoretic interpretation of thresholds in probabilistic rough sets, Xiaofei Deng and Yiyu Yao, © 2012 https://doi.org/10.1007/978-3-642-31900-6_46

value is zero. If the counter exceeds the limit, which is calculated by Equation (13), then the threshold pair is replaced with a new one generated using Equation (9) [18,46,47].

\[
\text{limit} = \frac{\text{Total number of threshold pairs} \times \text{Dimension}}{2} 
\]

In Equation (13), the Dimension is the number of parameters to be optimised. As usual, the fitness value is calculated for each threshold pair, and the updated fitness value is compared with the global best. The best fitness value and the corresponding threshold pair are stored as the global best.

All these phases are sequentially executing a certain number of cycles, and finally, the optimal pair of thresholds satisfying the minimum uncertainty of the trisection is generated.

4. Experimental Results

This section gives a brief description of the experimental environment, data set and experimental parameters. Also, Section 4.1 provides the comprehensive results of each cycle. The experiment focuses on the optimisation of entropy in probabilistic regions using the ABC algorithm, as explained. It is conducted in a computer with 19.4 GiB memory, Intel Core i7-9750H CPU @ 2.60 GHz × 12 processor, TU117M graphics, 1.5 TB disk capacity and Ubuntu 20.04.2 LTS operating system. The outcome of this experiment offers a better way of optimisation of quantified uncertainty in terms of Shannon entropy. To prove this argument, we make use of probabilistic details about an equivalence class structure on the basis of class C, since the existing methods use the same data in their works [5,6,10]. Table 1 shows the information about the equivalence class structure which includes 15 equivalence classes in total. The data displayed in this table is generated randomly in which it quantifies the probability of each equivalence class and the corresponding conditional probability (Pr(C|X_I)). In order to compute the probabilistic regions conveniently, they are arranged in the descending order of their conditional probability. These conditional probabilities are compared with the (\(\alpha, \beta\)) pairs and three regions are generated where \(\alpha\) and \(\beta\) values lies within the limit \(0 \leq \beta < 0.5 \leq \alpha \leq 1\). Based on these three regions, the uncertainty is calculated using Shannon entropy. In the ABC algorithm, new (\(\alpha, \beta\)) pairs are generated in four phases, i.e. Initialisation Phase, Employed Bee Phase, Onlooker Bee Phase and Scout Bee Phase [18]. In each phase, entropy is calculated and updated, and the entropy is optimised after a certain number of cycles.

Our experiment starts with an initial population of five candidates. Therefore, for each run, the algorithm returns five optimal pairs of \(\alpha\) and \(\beta\). For input, Table 2 lists the primary
Table 2. Initial population of threshold pairs.

| Candidate | Alpha | Beta |
|-----------|-------|------|
| 1         | 1.0   | 0.4  |
| 2         | 0.9   | 0.3  |
| 3         | 0.8   | 0.2  |
| 4         | 0.7   | 0.1  |
| 5         | 0.5   | 0.0  |

food sources in terms of the initial population of threshold pairs. Each candidate contains \( \alpha \) and \( \beta \) values as the threshold pair. Based on these threshold pairs, all phases are working sequentially in a cyclic manner. We need only two cycles to converge the proposed algorithm for this particular problem since the subsequent cycles have the least role in further improvement of the results.

4.1. Experimental Results in Each Cycle

The following sections explain the results obtained during the execution of our proposed algorithm. Section 4.1.1 explains the initialisation phase. Section 4.1.2 gives the result of the sequential execution of the remaining phases in the first cycle. Similarly, Section 4.1.3 explains the result of the second cycle. In this execution, the algorithm converges after completing two cycles. Therefore, the results obtained at the end of the two cycles are shown in the corresponding tables.

4.1.1. Initialisation

The entire candidates in the initial population go through the initialisation process using Equation (9), and we determine the OCE and the fitness value for each candidate using Equation (4) and Equation (10), respectively. The maximum fitness value 0.6206 is stored as the global best together with (0.86, 0.31) as the corresponding threshold pair from these results. In Table 3, the initialised threshold pairs are displayed.

4.1.2. Cycle 1

In this cycle, the employed bee phase generates new candidates based on their neighbourhood. Later, it determines the fitness value for each candidate. If the new fitness value is better than the existing one, it is updated along with the threshold pair. This cycle returns 0.6206 as the best fitness value. To assign the onlooker, the fitness probabilities corresponding to these solutions are computed next. All onlookers are assigned to these food sources in the higher order of their fitness probabilities. The algorithm selects a food source randomly if a tie occurs. In the onlooker phase, these threshold pairs are exploited and tried.

Table 3. Initialised threshold pairs.

| Candidate | Alpha | Beta | OCE   | Fitness value |
|-----------|-------|------|-------|---------------|
| 1         | 0.86  | 0.31 | 0.6115| 0.6206        |
| 2         | 0.59  | 0.28 | 0.6701| 0.5988        |
| 3         | 0.70  | 0.08 | 0.629 | 0.6139        |
| 4         | 0.68  | 0.16 | 0.6286| 0.6140        |
| 5         | 0.92  | 0.28 | 0.6213| 0.6168        |
to find the optimal pair of thresholds. This phase returns the optimal results at \((0.87, 0.20)\), \((0.88, 0.26)\), \((0.88, 0.26)\) and \((0.88, 0.25)\) except the third candidate. The optimal result at these different threshold pairs is displayed in Table 4. The scout bee phase involves the execution only when the counter value for any candidates exceeds the limit. This phase is not involved in this cycle since the counter vector for the candidates is \([0, 0, 2, 0, 0]\), where the limit for each candidate is 5.

4.1.3. Cycle 2
This cycle also returns 0.6206 as the best fitness value. There is no further improvement shows in the results of Cycle 2 in terms of fitness value. Table 5 gives the results of Cycle 2 where each candidate converges to the best fitness value (0.6206) at the threshold \((0.88, 0.24)\). The scout bee phase is not engaged in this cycle since the counter vector for five candidates is \([0, 0, 0, 0, 0]\).

5. Discussion
First, this section discusses the benefits and the limitations associated with our proposed method. Then, the obtained results are compared with the existing methods in Section 5.1. When summarising the work, the effective threshold pairs are generated from the initialised threshold pairs in the employed bee phase. Similarly, from the effective threshold pairs, the optimised threshold pairs are generated in the onlooker bee phase, i.e. the process of exploration and exploitation of threshold pairs is well executed in this proposed method. Also, non-updated threshold pairs are replaced with the help of the scout bee phase. At each run, all the phases are executed iteratively and produced the optimised results. Since the population size is too small, the algorithm converges very quickly. From the results obtained, it is clear that the proposed algorithm generates results apart from the initial threshold pairs. Also, the fitness function is designed based on the objective function representing the quality we have to optimise in this proposed approach. This enables us to incorporate different objective functions for the optimisation, providing a common framework for this optimisation problem.

Moreover, at each run, the algorithm produces five pairs of thresholds and their corresponding entropy values. To make it clear, refer to Table 6, which describes the results by running the algorithm 10 times. From these results, it is effectively proven that the optimum

| Candidate | Alpha | Beta | OCE   | Fitness value |
|-----------|-------|------|-------|---------------|
| 1         | 0.87  | 0.20 | 0.6115| 0.6206        |
| 2         | 0.88  | 0.26 | 0.6115| 0.6206        |
| 3         | 0.80  | 0.10 | 0.6286| 0.6140        |
| 4         | 0.88  | 0.26 | 0.6115| 0.6206        |
| 5         | 0.88  | 0.25 | 0.6115| 0.6206        |

| Candidate | Alpha | Beta | OCE   | Fitness value |
|-----------|-------|------|-------|---------------|
| 1, 2, 3, 4, 5 | 0.88  | 0.24 | 0.6115| 0.6206        |
results are obtained at different pairs of thresholds. The algorithm outperforms the other existing models in terms of various optimal pairs of thresholds. The algorithm converges to the minimum entropy, 0.6115. However, the classical ABC algorithm has a specific drawback associated with the local minimum. In Table 6, the 8th row describes the problem of local minimum where the algorithm converges to 0.615 as the optimal entropy. The global optimisation methods in ABC algorithm are proposed in [17,48–51] and they avoid the problem of local optimum in the classical ABC algorithm. For the sake of brevity, this paper focuses on the advantages of the classical ABC algorithm in the optimisation of thresholds.

5.1. Comparison with the Existing Models

The proposed algorithm provides variant threshold pairs with minimum uncertainty. For briefness, we explain the thresholds in a range, $0.8 < \alpha \leq 0.9$ and $0.2 \leq \beta < 0.4$ which provides a minimum uncertainty value (0.6115) with minimum iteration. Therefore, our argument is that the ABC-based optimisation of PRS is a perfect method to derive all possible optimised results by running the algorithm at different times. To attest to our argument, we compare it with different PRS models like ITRS, GTRS and GA-based optimisation. The comparison results are displayed in Table 7.

The ITRS returns 0.6115 as the minimum uncertainty for this particular problem [6]. This algorithm determines the best pair of thresholds directly from the initial set of threshold pairs. However, this problem is avoided in the ABC-based optimisation by the neighbourhood exploration of each candidate. Though the result is the same for both ITRS and ABC-based optimisation for this particular problem, the optimisation algorithm is the right choice for the optimal pair of thresholds.

In GTRS, our ultimate aim is to balance the uncertainties between the immediate region and boundary region at a Nash equilibrium state. The objectives are balanced at $(0.8, 0.1)$ with the certainty values (0.946, 0.480) as the results in [5]. The results obtained in our

| Table 6. Optimised results when execute the algorithm ten times. |
|-----------------|-----------------|-----------------|
| Index | Five pairs of optimised $(\alpha, \beta)$ | Minimum Entropy |
| 1     | $(0.83,0.28), (0.83,0.24), (0.83,0.31), (0.83,0.30), (0.83,0.28)$ | 0.6115 |
| 2     | $(0.81,0.20), (0.81,0.23), (0.81,0.20), (0.81,0.22), (0.81,0.23)$ | 0.6115 |
| 3     | $(0.85,0.25), (0.85,0.25), (0.85,0.36), (0.85,0.25), (0.85,0.36)$ | 0.6115 |
| 4     | $(0.89,0.27), (0.89,0.27), (0.89,0.27), (0.89,0.27), (0.89,0.27)$ | 0.6115 |
| 5     | $(0.87,0.26), (0.87,0.26), (0.87,0.24), (0.87,0.26), (0.87,0.24)$ | 0.6115 |
| 6     | $(0.83,0.31), (0.84,0.21), (0.83,0.29), (0.83,0.31), (0.83,0.32)$ | 0.6115 |
| 7     | $(0.83,0.34), (0.84,0.24), (0.83,0.27), (0.84,0.30), (0.84,0.31)$ | 0.6115 |
| 8     | $(0.71,0.24), (0.71,0.24), (0.71,0.24), (0.71,0.24), (0.71,0.24)$ | 0.615 |
| 9     | $(0.85,0.29), (0.86,0.29), (0.85,0.29), (0.86,0.29), (0.86,0.30)$ | 0.6115 |
| 10    | $(0.85,0.34), (0.87,0.38), (0.82,0.29), (0.85,0.29), (0.84,0.36)$ | 0.6115 |

| Table 7. Experimental results of different PRS models. |
|--------|----------|----------|------|
| Model  | Method   | OCE      | Threshold pair |
| ABC-based Model | Optimisation | 0.6115 | $0.8 < \alpha \leq 0.9$, $0.2 \leq \beta < 0.4$ |
| ITRS Model    | Selection   | 0.6115 | $(0.9, 0.2)$ |
| GTRS Model    | Trade-off   | 0.6286 | $(0.8, 0.1)$ |
| GA-based Model | Optimisation | 0.6150 | $(0.8, 0.2)$ |
Table 8. Time complexity of different PRS models.

| Model          | Time complexity                                    | Time complexity (K=2) |
|----------------|---------------------------------------------------|-----------------------|
| ABC-based Model | $O(N_p + N_qN_p + K(N_qN_p^2 + N_p \log N_p))$   | $O(853.219)$          |
| ITRS Model     | $O(N_qN_p)$                                       | $O(375)$              |
| GTRS Model     | $O(KN_q)$                                         | $O(30)$               |
| GA-based Model | $O(N_p + N_qN_p + K(N_qN_p + N_p \log N_p))$       | $O(253.219)$          |

Population size ($N_p = 5$), Number of equivalence classes ($N_q = 15$), Number of $\alpha$ values ($N_\alpha = 5$), Number of $\beta$ values ($N_\beta = 5$), Number of iterations ($K = 2$).

problem are not optimum at (0.8 0.1) with an uncertainty level of 0.6286. One limitation associated with this theory is the trade-off strategy between the objectives, which suits well with the multi-objective optimisation problems [52]. Another problem is the limitation of the search space by the predefined strategies. There is a chance of missing optimal solutions because of these predefined strategies. In the proposed method, the search space is generated based on the neighbours of each candidate randomly, and also multiple optimal solutions are the advantages of the ABC algorithm over GTRS.

GA-based optimisation algorithm returns 0.6150 as the minimum quantified uncertainty at the threshold (0.8, 0.2) [5]. As per our algorithm, optimal alpha lies in the range $0.8 < \alpha \leq 0.9$ precisely. Therefore, the proposed method offers a slightly better result (0.6115) than GA-based optimisation. Even though GA gives good quality solutions, it does not guarantee the optimum solutions due to the premature convergence [53]. However, GA-based optimisation gives a general framework for the optimisation, and the author suggests incrementing the number of iterations for getting the more accurate result [5]. Moreover, in this method, the search space for the solution directly depends on the initial population. The number of chromosomes is to be incremented in order to increase the search space. Since the search space is vast for ABC-based optimisation by randomisation, the possibilities of getting an accurate result are high. Additionally, ABC-based optimisation returns different optimal results at various thresholds.

Moreover, all of these methods are compared using their time complexity. Here, Big O of each algorithm is explained on the basis of the number of equivalence classes ($N_q$), size of the population ($N_p$), number of $\alpha$ values ($N_\alpha$), number of $\beta$ values ($N_\beta$) and the number of iterations ($K$). The complexity is displayed in Table 8. Also, it shows the complexity at $K = 2$ since our proposed method converges at this level. The complexity of the ABC algorithm is slightly higher than the other algorithms due to the extended search spaces. However, the ABC algorithm outperforms the existing methods when compared to the other results discussed above.

6. Conclusion

Probabilistic Rough Sets offer different models for the optimisation of thresholds which follows the iterative strategy to optimise various objectives. The performance obtained from these models is either due to the iteration mechanism followed by the model or the capability of the model. In order to avoid this ambiguity, a common framework is necessary to compare the performances of these models to identify which one is more efficient. The proposed algorithm follows a common framework capable of generating different optimal pairs of thresholds with the minimum number of iterations. An ABC-based
optimisation strategy is employed in this algorithm to minimise the uncertainty involved in the three regions, which are obtained as a result of the corresponding threshold pair. Hence the algorithm outputs threshold pairs satisfying the optimal criteria. In order to prove the efficiency of the proposed algorithm, it is compared with the three different existing models, and the results prove that the ABC-based optimisation is the finest method for optimising the thresholds. However, this method faces the local optimisation problem. Our future research work mainly focuses on the elimination of this drawback and, at the same time, investigating the performance of the same algorithm by considering other objective functions suitable to measure the quality of the trisection.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

T. V. Soumya was born in Kerala, India. She received the MCA degree from Mahatma Gandhi University, Kottayam, India. She is a research scholar at the Department of Computer Applications, Cochin University of Science and Technology, India, where she specialises in Three-Way Decisions. Her area of interest includes Machine Learning, Data Mining and Rough Set Theory.

M. K. Sabu was born in Kerala, India. He received the MCA degree from Government Engineering College, Thrissur, Kerala, India, and the PhD degree in Computer Science from the School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala, India. He is currently a professor in the Department of Computer Applications, Cochin University of Science and Technology (CUSAT), Kochi, India. His research interests include Data Mining, Rough Set Theory, Optimisation Techniques, Machine Learning and Artificial Intelligence. He has numerous articles in well-known journals and chaired many conferences. He is very active as a resource person, examiner, reviewer, visiting faculty member and member of the board of studies in various universities and several autonomous colleges in Kerala.

ORCID

T. V. Soumya http://orcid.org/0000-0003-0067-0656
M. K. Sabu http://orcid.org/0000-0002-4958-5033

References

[1] Yao Y. Three-way decisions with probabilistic rough sets. Inf Sci (NY). 2010;180(3):341–353.
[2] Yao Y. The superiority of three-way decisions in probabilistic rough set models. Inf Sci (NY). 2011;181(6):1080–1096.
[3] Yao Y. Probabilistic rough set approximations. Int J Approx Reason. 2008;49(2):255–271.
[4] Yao Y. Two semantic issues in a probabilistic rough set model. Fundam Inform. 2011;108(3–4):249–265.
[5] Majeed B, Azam N, Yao JT. Thresholds determination for probabilistic rough sets with genetic algorithms. Rough Sets and Knowledge Technology; Springer International Publishing; 2014. p. 693–704.
[6] Deng X, Yao Y. An information-theoretic interpretation of thresholds in probabilistic rough sets. International Conference on Rough Sets and Knowledge Technology; Springer; 2012. p. 369–378.
[7] Yao Y. Three-way decision: an interpretation of rules in rough set theory. International Conference on Rough Sets and Knowledge Technology; Springer; 2009. p. 642–649.
[8] Zhang Y, Yao JT. Gini objective functions for three-way classifications. Int J Approx Reason. 2017;81:103–114.
[9] Gao C, Yao Y. Determining thresholds in three-way decisions with chi-square statistic. International Joint Conference on Rough Sets; Springer; 2016. p. 272–281.
[10] Azam N, Yao JT. Variance based determination of three-way decisions using probabilistic rough sets. International Joint Conference on Rough Sets; Springer; 2016. p. 209–218.
[11] Jia X, Tang Z, Liao W, et al. On an optimization representation of decision-theoretic rough set model. Int J Approx Reason. 2014;55(1):156–166.
[12] Deng X, Yao Y. A multifaceted analysis of probabilistic three-way decisions. Fundam Inform. 2014;132(3):291–313.
[13] Azam N, Yao J. Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets. Int J Approx Reason. 2014;55(1):142–155.
[14] Zhang Y. Optimizing gini coefficient of probabilistic rough set regions using game-theoretic rough sets. 2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE); IEEE; 2013. p. 1–4.
[15] Zhang Y, Yao JT. Determining three-way decision regions by combining gini objective functions and gtrs. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Springer; 2015. p. 414–425.
[16] Zhang Y, Yao JT. Determining three-way decision regions with gini coefficients. International Conference on Rough Sets and Current Trends in Computing; Springer; 2014. p. 160–171.
[17] Gao W, Liu S. Improved artificial bee colony algorithm for global optimization. Inf Process Lett. 2011;111(17):871–882.
[18] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J Global Optim. 2007;39(3):459–471.
[19] Yao Y. Decision-theoretic rough set models. International conference on rough sets and knowledge technology; Springer; 2007. p. 1–12.
[20] Deng X. Three-way classification models [PhD thesis]. Faculty of Graduate Studies and Research, University of Regina; 2015.
[21] Yao Y, Greco S, Słowiński R. Probabilistic rough sets. Springer handbook of computational intelligence. Springer; 2015. p. 387–411.
[22] Pawlak Z. Rough sets. Int J Computer & Inform Sci. 1982;11(5):341–356.
[23] Pawlak Z, Skowron A. Rough sets: some extensions. Inf Sci (Ny). 2007;177(1):28–40.
[24] Jia X, Shang L. Three-way decisions versus two-way decisions on filtering spam email. Transactions on Rough Sets XVIII. Springer; 2014. p. 69–91.
[25] Zhou B, Yao Y, Luo J. A three-way decision approach to email spam filtering. Canadian Conference on Artificial Intelligence; Springer; 2010. p. 28–39.
[26] Yao Y. Probabilistic approaches to rough sets. Expert Syst. 2003;20(5):287–297.
[27] Nauman M, Azam N, Yao JT. A three-way decision making approach to malware analysis using probabilistic rough sets. Inf Sci (Ny). 2016;374:193–209.
[28] Herbert JP, Yao JT. Game-theoretic rough sets. Fundam Inform. 2011;108(3-4):267–286.
[29] Wang S-C. Genetic algorithm. Interdisciplinary Computing in Java Programming. Springer; 2003. p. 101–116.
[30] Shannon CE. A mathematical theory of communication. Bell Syst Tech J. 1948;27(3):379–423.
[31] Deng Y. Deng entropy. Chaos Solitons Fractals. 2016;91:549–553.
[32] Von Neumann J, Morgenstern O. Theory of games and economic behavior (commemorative edition). Princeton (NJ): Princeton University Press; 2007.
[33] Pelillo M. What is a cluster? perspectives from game theory. Proc. of the NIPS Workshop on Clustering Theory; Citeseer; 2009.
[34] Shoham Y. Computer science and game theory. Commun ACM. 2008;51(8):74–79.
[35] Liu P. Incorporating Three-Way Email Spam Filtering With Game-Theoretic Rough Sets [PhD thesis]. Faculty of Graduate Studies and Research, University of Regina; 2019.
[36] Azam N. Investigating Decision Making with Game-Theoretic Rough Sets [PhD thesis]. Faculty of Graduate Studies and Research, University of Regina; 2014.

[37] Zhang Y. Game Theoretic Approaches to Three-Way Decisions [PhD thesis]. Faculty of Graduate Studies and Research, University of Regina; 2018.

[38] Zhang Y, Liu PF, Yao JT. Three-way email spam filtering with game-theoretic rough sets. 2019 International Conference on Computing, Networking and Communications (ICNC); IEEE; 2019. p. 552–556.

[39] Wang S-C. Genetic algorithm. Interdisciplinary computing in java programming. Springer; 2003. p. 101–116.

[40] Whitley D. A genetic algorithm tutorial. Stat Comput. 1994;4(2):65–85.

[41] Holland JH. Adaptation in Natural and Artificial systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge: MIT Press; 1992.

[42] Chakraborty A, Kar AK. Swarm intelligence: A review of algorithms. Nature-Inspired Computing and Optimization. 2017. p. 475–494.

[43] Mavrovouniotis M, Li C, Yang S. A survey of swarm intelligence for dynamic optimization: algorithms and applications. Swarm Evol Comput. 2017;33:1–17.

[44] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput. 2011;1(1):3–18.

[45] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical report, Technical Report–tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[46] Karaboga D, Basturk B. On the performance of artificial bee colony (abc) algorithm. Appl Soft Comput. 2008;8(1):687–697.

[47] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput. 2009;214(1):108–132.

[48] Guo P, Cheng W, Liang J. Global artificial bee colony search algorithm for numerical function optimization. 2011 Seventh International Conference on Natural Computation; Vol. 3, IEEE; 2011. p. 1280–1283.

[49] Liu J, Zhu H, Ma Q, et al. An artificial bee colony algorithm with guide of global & local optima and asynchronous scaling factors for numerical optimization. Appl Soft Comput. 2015;37:608–618.

[50] Gao W-G, Liu S-Y. A modified artificial bee colony algorithm. Comput Oper Res. 2012;39(3): 687–697.

[51] Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. J Comput Appl Math. 2012;236(11):2741–2753.

[52] Azam N, Yao JT. Multiple criteria decision analysis with game-theoretic rough sets. International Conference on Rough Sets and Knowledge Technology; Springer; 2012. p. 399–408.

[53] Pandey HM, Chaudhary A, Mehrotra D. A comparative review of approaches to prevent premature convergence in GA. Appl Soft Comput. 2014;24:1047–1077.