Genetic Algorithm Selection Strategies based Rough Set for Attribute Reduction

Gadeer Mahmood Alathamneh[1], Salwani Abdullah[1], and Nor Samsiah Sani[1]

[1]Center for Artificial Intelligence Technology (CAIT), Universiti Kebangsaan Malaysia 43600, Bangi, Selangor, Malaysia

Summary
Attribute reduction is considered a vital topic for studies that consider actual data intricacy. The attribute reduction problem aims to find a minimum attribute set from a large set of attributes while avoiding information loss. The problem is denoted as an NP-hard, which is the non-deterministic polynomial time optimization problem. Researchers have widely used many heuristic and meta-heuristic approaches to optimize this problem in rough set theory. Numerous studies have utilized meta-heuristic methods to address the attribute reduction problems, prompting this research to suggest an improved one-population meta-heuristic method. This paper presents the implementation of the genetic algorithm on an attribute reduction-based rough set utilizing different selection strategies: roulette wheel, tournament and rank-based selections. An experiment was performed on 13 datasets from the public domain available in the UCI repository. The results demonstrated that the tournament selection strategy performed better than the roulette wheel and rank-based selection strategies and other published meta-heuristic algorithms.

Key words:
Genetic algorithm selection strategies, Meta-heuristics, Rough set attribute reduction

1. Introduction

Attribute reduction is a pre-processing task to simplify the process of any learning algorithm. Reducing the number of attributes lessens the complexity of any data-mining task or learning algorithm. Attribute reduction loosely refers to a minimal subset of an original attribute set which still contains important attributes like the original. The minimal subset should still represent the original attribute set without losing information [1-2]. The search for this minimal subset is known as the non-deterministic polynomial time (NP)-hard problem. An effective mathematical tool to solve this problem is the rough set theory. Rough set theory extracts the relation of decision attributes with conditional attributes. The dependency degree of the attributes is calculated based on this relation. This value is used to evaluate the quality of the subset.
Meta-heuristic approaches utilized to solve attribute reduction problems can be classified into single-based and population-based approaches. Some examples of single-based approaches on attribute reduction are simulated annealing [3], tabu search [4], variable neighbourhood search [5], iterative algorithm with composite neighbourhood structure [6], great deluge algorithm [7], nonlinear great deluge [33], constructive hyper-heuristics [7], exponential monte carlo [9] and fuzzy record-to-record [10]. Some population-based approaches are genetic algorithms [11], ant colony [11-12], scatter search [13-14], whale optimization approach [15] and binary ant lion optimer [16]. Hybrid approaches on attribute reduction problems, such as the hybridization between fuzzy logic and record-to-record travel algorithm [17], hybrid genetic algorithm with great deluge [8] and memetic algorithm [18], have also been investigated.
Other approaches on attribute reduction can be found in [15-16], [19-28]. In this work, we investigate the effect of selection strategies within the genetic algorithm for solving attribute reduction problems. Three selection strategies are investigated: roulette wheel, rank and tournament-based selection. The proposed method has been tested on UCI datasets [29].We use the rough set theory to evaluate the obtained subset of features [1-2].
The rest of this paper is organized as follows: Section 2 provides a review on a rough set theory. The problem description is presented in Section 3. Section 4 explains the proposed methods in details. Section 5 reports the and discusses the experimental results. The conclusion of this work is provided in last section.

2. Literature Review

A. Rough Set Theory

Rough set theory (RST) is a mathematical method used to analyse ambiguity, uncertainty and vagueness in a big dataset. During the decision-making process, RST uses sets' approximation, called upper and lower set approximation [1-2].
An information system consists of a pair S = (U, F), where a non-empty finite set of objects U is denoted as the universe and F is a non-empty finite set of attributes, such that $f: U \rightarrow Vf$, for every $f \in F$. The set Vf is called the domain. An information system in RST is like a dataset in clustering and unsupervised machine learning. An information system of the form S = (U, F, d), where d is

Manuscript received September 5, 2019
Manuscript revised September 20, 2019

the decision attribute is called a decision system. In supervised learning and classification, a dataset can be deemed as a decision system, where the instances are the objects of the universe and attributes are the elements of F and labels that represent values of the decision attribute [30].

Table 1: Example of data

x ∈ U	f1	f2	f3	f4	D
1	2	1	1	0	3
2	0	1	1	0	4
3	0	1	1	0	4
4	0	1	0	0	4
5	0	1	0	4	5
6	0	1	0	4	5
7	0	1	1	0	4
8	0	1	1	0	4

For any set $B \subseteq F \cup \{d\}$, we define the B-indiscernibility relation as:

$$INDIS(B = \{(x, y) \in U \times U | \forall f \in B, f(x) = f(y)\} \quad (1)$$

For Table 1's dataset, if B = {*f3*, *f4*}, then objects 4 are objects 1,2,3,7,8 and 5,6 are indiscernible. *U/B* is as follows: *U/B* = {{4}, {1,2,3,7,8}, {5,6}}.

Two essential concepts of rough sets are the upper and lower set approximations. Let $X \subseteq U$ and $B \subseteq F$, the *B*-upper and *B*-lower approximations of *X* are defined as follows:

$$B_{lower} X = \{x | [x]B \subseteq X\} \quad (2)$$

$$B_{lower} X = x | [x]B \cap X = \emptyset\} \quad (3)$$

The $B_{upper}X$ and $B_{lower}X$ approximations define information contained in B. If $x \in BX$, it particularly belongs to *X* but if $x \in BX$, it may or may not belong to *X*. For example, let *B* = {*f3*, *f4*} and *X*= {1, 2, 5, 4, 6}, then

$$B_{lower} X = \{4, 5, 6\}$$

$$B_{upper} X = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

By the definition of *BX* and *BX*, the objects in *U* can be compartmentalized into three parts, called the negative and positive regions.

$$POSB(X) = B_{lower} X \quad (4)$$

$$NEGB(X) = U - BX \quad (5)$$

In the example, the two regions for *B = {f3, f4}* and *X= {1, 2, 5, 4, and 6}* are as follows:

$$POSB(X) = \{4,5,6\}$$

$$NEGB(X) = \{1,2,3,7,8\}$$

In data analysis, discovering dependencies among attributes is an important issue. Let *D* and *C* be subsets of $F \cup \{d\}$. For $0 \leqslant k \leqslant 1$, it is said that *D* depends on *C* in the k_{th} degree (denoted $C \Rightarrow kD$), if

$$k = \gamma(C, D) + \frac{|POSC\ (D)|}{|U|} \quad (6)$$

where *POSC (D) =U* $\underline{C}X$

$$X \in \frac{U}{D} \quad (7)$$

Equation 7 calls a positive region of the partition *U/D*, with regard to *C*. This region is the set of all elements of *U* that can be uniquely classified into blocks of the partition *U/D*, by means of *C*. In the example, if C = {*f 3*, *f 4*} then: *POSC* (*d*) = *U* ($\underline{C}$ {1, 2, 3, 8, 7}, $\underline{C}$ {4}, $\underline{C}$ {5,6}) = {4, 5, 6}.

The degree of dependency of attribute *d* on attributes {*f 3*, *f 4*} is:

$$\gamma(\{f3, f4\}, d) = \frac{|POS\{f3, f4\}(d)|}{|U|} = \frac{3}{8}$$

The functional dependency of *D* and *C* ($C \Rightarrow D$) is a special case of dependency, where γ (*C*, *D*) =1. In this case, it is said that all attributes' values from *D* are uniquely specified by the values of attributes from *C*.

A reduct is defined as a subset of minimum cardinality of the conditional attribute set C, such that $\gamma R(D) = \gamma C(D)$

$$R = \{X : X \subseteq C, \gamma x(D) = \gamma C(D)\} \quad (8)$$

$$R_{min} = \{X : X \in R, \forall Y \in R, |X| \leq |Y|\} \quad (9)$$

The core is defined as an intersection of all the sets in R_{min}

$$Core(R) = \cap X_{X \in R} \quad (10)$$

The core elements are attributes that are impossible to omit without introducing more contradictions to the dataset.

Utilizing the dataset in Table 1 and the degree of dependency *D* = {*d*} on all possible subsets of *C* can be calculated as γ{1} =1/8; γ{2} =0; γ{3} =0; γ{4} =2/8; γ{1,2}=1/8; γ{1,3} =5/8; γ{1,4} =1; γ{2,3}=0; γ{2,4} =2/8; γ{3,4} =3/8; γ{1,2,3} =5/8; γ{1,2,4}=1; γ{1,3,4}=1; γ{2,3,4}=3/8;

The minimal results obtained in this example are: R_{min} = {*f1*, *f4*}.

The process to find minimum reducts is labelled as an NP-hard problem. Calculating all the potential reducts (*Core*(*R*)) is a time-consuming process. Therefore, the researchers have attempted to utilise heuristic algorithms to find approximate solutions to this problem. Table 2 shows the dataset after reduction, where the dependency value of attributes equals 1.

Table 2: Dataset after reduction

x ∈ U	f1	f4	D
1	2	0	3
2	0	0	4
3	0	0	4
4	0	0	4
5	0	4	5
6	0	4	5
7	0	0	4
8	0	0	4

3. Problem Description

In this section, we describe the attribute reduction problem, solution representation and fitness function.

A. Attribute Reduction Problem

An attribute reduction problem is a pre-processing task in data mining. An attribute reduction problem can be represented by a pair (*A*, *c*), where *A* represents the original set of attributes and *c* is the fitness function which evaluates how good the selected subset is. The problem is to find the best subset of attributes A in such a way that the generated subsets have a smaller number of attributes compared to the original set *A*, with better accuracy [23, 27].

B. Solution Representation

In this method, a one-dimensional binary vector is used to represent a solution, where the size of the vector is equal to the number of attributes in the original dataset plus two extra cells, i.e. the dependency degree of each individual and the number of selected attributes (number of ones). Each vector cell is represented by “1” or “0”, where “1” shows that the corresponding attribute is selected and “0” means that the corresponding attribute is discarded or known as an unselected attribute. Figure 1 shows the solution representation with six attributes (a_1, a_2, a_3, a_4, a_5, a_6), where three attributes (a_3, a_4 and a_5) are selected; with the dependency degree (based on rough set theory) equal to 1 and the number of ones equal to 3. The other attributes are discarded.

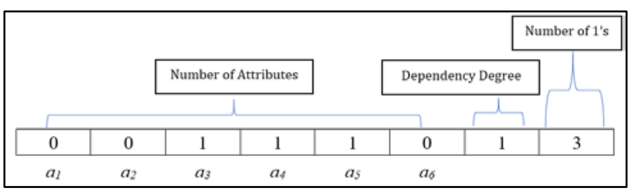

Fig. 1 Initial solution representation.

C. Fitness Function

Here, the dependency degree of the rough set theory [1-2] is used as the fitness function to evaluate the generated subset of attributes. The dependency degree calculates data dependencies and returns a value between zero and one. A dependency degree equal to one means that the generated subset of attributes is informative. In this work, we maintain a dependency degree equal to one on all generated subsets of attributes by adding or deleting attributes from a given subset. Given two subsets of attributes, the subset with the lowest number of attributes will be accepted.

4. Methodology

A. Genetic Algorithm for Rough Set Attribute Reduction (GA-AR)

In this work, we discuss the genetic algorithm with three different selection strategies to deal with the attribute reduction problem (coded as GA-AR). The algorithm aims to investigate the impact of the selection strategies, when solving the attribute reduction problem, compared to other available approaches. The following subsections cover the initial solution generation method and neighbourhood operator as well as the GA-AR algorithm.

1) Initial solution method and the neighbourhood operator: The initial solution is constructed randomly by distributing zeros and ones into each cell ofthe one-dimensional binary vector. We use a systematic operator to generate a neighbourhood solution by starting from the first element of the array and using a flip strategy. If the value of the selected cell is “1”, it will be changed to “0” and vice versa.

2) GA-AR algorithm: The genetic algorithm (GA) is a popular algorithm among the different evolutionary algorithms. The GA begins with the search process for solutions in an initial individuals’ population that isusually randomly generated. Each individual represents a probable solution to the problem. Commonly, individuals are encoded in strings of 1s and 0s. Then, the initial population evolves in generations. In each generation, each individual of the current population is evaluated depending on the fitness function. The pseudocode of the genetic algorithm is shown in Figure 2.

```
Parameter initialization
 Initial solution construction
        While i < Number of generation
                Fitness calculation
               Selection
               Crossover
               Mutation
               Update the population
        End while
Return the best solution
```

Fig. 2 The pseudocode of a GA.

A new population is generatedby applying a selection strategy that usually accepts an improved solution. The algorithm moves from one solution to another through the population until an optimal solution is found or a criterion of termination is met. To generate new offspring individuals, some of the selected individuals are modified by mutating and recombining their parts. Then, the selected individuals are brought forward to the next intact generation and the new population is utilized in the next geneation. The selection strategies supported by the genetic operators (crossover and mutation) are intended to move the population to the optimal solution. Based on the developed GA program, and by following various literatures on GA parameterisations, key numbers of parameters required by the GA are identified and performed as shown in Table 3.

Table 3: Parameter settings

Name	Value
Population size, m	10
Number of generations	200
Tournament size, k	2
Single-point crossover	Random
Mutation rate, Pm	0.4

In the GA process, selection is a significant operation. The selection phase identifies individuals chosen for mating (reproduction) and the number of offspring produced by each chosen individual. The selection strategy's main principle is "the better a solution is, the higher its chance of being a parent" [31]. This process decides which solutions are to be conserved and allowed to reproduce and which ones can be eliminated. The main target of the selection operator is to assure valid solutions and eliminate invalid ones in a population whilst preserving a constant population size.

There are many strategies for selection; however, this research describes the selection strategy required for implementation. The three selection strategies employed in this research are roulette wheel, tournament and rank-based selection. The different selection methods used in this algorithm can be described as follows:

GA-AR with roulette wheel selection (GARW-AR): In the roulette wheel selection strategy, each individual has aselection probability proportionate to its fitness value. In other words, the individual's selection chance corresponds to a segment of the roulette wheel. The probability of selecting a parent can be seen as spinning a roulette wheel with asegment size for each parent proportionate to its fitness. Certainly, those who have the largest fitness (i.e. largest sizes of the segments) have a greater probability of being chosen; the fittest individual occupies the largest segment, while the least fit has a correspondingly smaller segment. The probability of selection of an individual Soli is shown below:

$$P(Sol_i) = \frac{f(Sol_i)}{\sum_{i=1}^{m} f(Sol_i)}, i=1\ldots m \qquad (11)$$

where f (Sol_i) is the fitness value of the individual Sol_i and m is the population size.

This selection scheme, by which the minimisation fitness function ought to be transformed to a maximisation function as in attribute reduction, is difficult to use on minimisation problems. It solves the selection problem to some extent, but makes the problem rather confusing. In attribute reduction, for instance, the best individual will continuously be assigned to the maximum fitness value among all other fitness functions. Consequently, the minimum tour is desired but the fitness maximises the fitness value as a by product. Hence, several other selection methods with a probability not proportionate to the fitness values of individuals are developed to encounter the proportionate selection problem. In general, there are two types of non-proportional selection operators: rank-based and tournament selection, which assign the probability value depending on the order of individuals according to their fitness values.

To demonstrate roulette wheel selection, consider a fitness function calculated for several solutions. The fitness function in this study is based on the number of non-selected attributes (number of 0s) instead of selected attributes (number of 1s). Table 4 illustrates the solutions for a given population, where the fitness value, in turn, generates a probability value of selection for each solution. The solutions can be ranked according to the largest probability value, if required, and a random number between 0 and 1 is generated (total sum of solutions' probability) to highlight the selected solution. The probability of each solution is calculated by Equation 11.

Table 4: Fitness value of each solution in GARW-AR

Solutions	Fitness value (#of selected attribute)	Probability value
Sol1	2	0.11
Sol2	1	0.05
Sol3	3	0.16
Sol4	2	0.11
Sol5	3	0.16
Sol6	1	0.05
Sol7	3	0.16
Sol8	1	0.05
Sol9	1	0.05
Sol10	2	0.11

a) GA-AR with tournament selection (GAT-AR): In tournament selection, k individuals are randomly selected from the populationand compete against each other. The winning individual selected for further processing of GA is the one with the highest fitness value. The tournament size k represents number of individuals taking part in every tournament. Table 5 illustrates the solutions in one population and the fitness value of each solution.

Table 5: Fitness value of each solution in GAT-AR

Solutions	Fitness values (#of selected attributes)
Sol_1	4
Sol_2	5
Sol_3	3
Sol_4	4
Sol_5	3
Sol_6	5
Sol_7	3
Sol_8	5
Sol_9	5
Sol_{10}	4

Figure 3 further highlights an instance where solutions (Sol_1 and Sol_7) based on tournament size k=2 are randomly selected from the population. By then, the fitter solution, Sol_7, has been chosen based on the fitness value. Variations to the tournament size can also be made based on smaller or larger sizes.

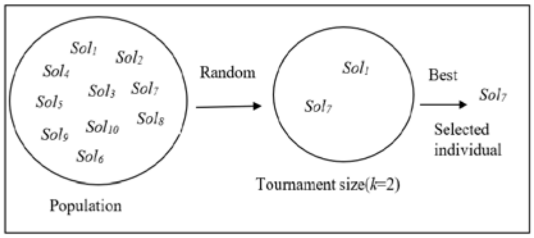

Fig. 3 Tournament selection strategy.

b) GA-AR with rank-based selection (GAR-AR): In this selection method, the individuals' rank is used instead of their corresponding fitness value. The function is biased towards individuals who have high rank (i.e., good fitness). The rank may be linearly scaled using the following formula:

$$P(i) = 2 - S + \left(2.0(S-1).\frac{(r(i)-1)}{m-1}\right) \qquad (12)$$

where S is the pressure of selection (1.0 <S≤ 2.0), m is the population size, and r(i) is the rank associated with the individual i. The greater the selection pressure S, the more important it is to individuals with a better rank. The expected probability P of the best individual is S, while the expected probability of the worst individual is 2−S. The selection pressure of all other population members can be interpreted by linear interpolation of the selection pressure according to rank. Here, the solutions must first be ranked based on their fitness value. The best solution is ranked as (m) and the worst solution attains rank 1. Table 6 illustrates an example of rank-based selection in the same population with selection pressure equal to 2 of fitness value, rank and probability value. The probability of each solution is calculated by Equation 12.

Table 6 shows that P(Sol3) is the solution which has the highest probability value. However, P(Sol8) possesses the smallest probability value, while Sol9 does not have any chance of selection as P(Sol9) is 0. In this example, Sol3 is randomly selected as parent 1 while Sol5 is selected as parent 2.

Table 6: Scaled rank with S

Solutions	**Fitness values (#of selected attributes)**	**Rank**	**Probability values, *P* with *S* =2.0**
Sol_3	3	10	2.0
Sol_5	3	9	1.7
Sol_7	3	8	1.5
Sol_4	4	7	1.3
Sol_1	4	6	1.1
Sol_{10}	4	5	0.8
Sol_6	5	4	0.6
Sol_2	5	3	0.4
Sol_8	5	2	0.2
Sol_9	5	1	0

5. Results and Discussion

The main objective of this study is to compare the performance of the GA in an attribute reduction-based rough set by utilizing different selection strategies – roulette wheel, tournament, and rank selection. The experiments using different selection strategies were performed on 13 datasets with different numbers of attributes and objects. Table 7 shows the description of datasets available in the University of California Irvine (UCI) repository. These datasets contain real-valued attributes and have been split to allow all methods to be fairly compared [11].

Table 7: Dataset specifications

Dataset	**No. of objects**	**No. of attributes**
M-of-N	1000	13
Exactly	1000	13
Exactly2	1000	13
Heart	294	13
Vote	300	16
Credit	1000	20
Mushroom	8124	22
Letters	26	25
LED	2000	24
Derm	366	34
Derm2	358	34
WQ	521	38
Lung	32	56

The performance of the proposed selection strategies (rank based and tournament) is compared with that of the standard selection strategy (roulette wheel). The comparison of results in Table 8 shows the reducts' size found in each strategy (represented in numbers). The number of runs that achieved the minimal reducts obtained by the algorithm from 20 trials is represented by the superscripts in parentheses. The number of attributes without superscripts illustrate that the strategy can obtain the minimal number of attributes for all runs.

Table 8: Comparison between proposed strategies

Dataset	**GA_{RW}-AR**	**GA_R-AR**	**GA_T-AR**
M-of-N	$6^{(15)}7^{(5)}$	$6^{(17)}7^{(3)}$	6
Exactly	$6^{(18)}7^{(2)}$	$6^{(19)}7^{(1)}$	6
Exactly2	$10^{(18)}11^{(2)}$	10	10
Heart	$6^{(14)}7^{(6)}$	$6^{(17)}7^{(3)}$	6
Vote	$8^{(18)}9^{(2)}$	8	8
Credit	$8^{(5)}9^{(11)}10^{(4)}$	$8^{(6)}9^{(11)}10^{(3)}$	$8^{(6)}9^{(13)}10^{(1)}$
Mushroom	4	4	4
LED	5	5	5
Letters	$8^{(15)}9^{(5)}$	8	8
Derm	$6^{(17)}7^{(4)}$	6	6
Derm2	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(8)}\ 9^{(11)}\ 10^{(1)}$	$8^{(8)}\ 9^{(11)}\ 10^{(1)}$
WQ	$12^{(1)}13^{(14)}\ 14^{(5)}$	$12^{(2)}13^{(16)}4^{(2)}$	$12^{(3)}\ 13^{(17)}$
Lung	$4^{(16)}5^{(4)}$	4	4

Table 8 demonstrates that the GA with tournament selection, GA_T-AR, outperformed other selection strategies on almost all datasets, with rank-based selection not too far behind. It is noticed that roulette wheel selection is biased to the best solution during the selection process. Therefore, the diversity of the population is poor, leading to the local optimum.

Results attained in this work are compared with other algorithms available in the literature. Note that the chosen methods for comparison are based on methods that borrowed the idea of using RST to measure the dependency between attributes and were categorised into single-based and population-based methods as follows:

Single solution-based meta-heuristic methods:

- Simulated annealing (SimRSAR) by Jensen and Shen [3]
- Tabu search (TSAR) by Hedar et al. [4]
- Composite neighbourhood structure for attribute reduction (IS-CNS) by Jihad and Abdullah [6]
- Hybrid variable neighbourhood search algorithm (HVNS-AR) by Arajy and Abdullah [5]
- Constructive hyper-heuristics (CHH_RSAR) by Abdullah and Jaddi [7]
- Great deluge algorithm (GD-RSAR) by Abdullah and Jaddi [7]
- Exponential MonteCarlo algorithm (EMC-FS) by Abdullah et al. [9]

Population-based meta-heuristic methods:

- Ant colony optimisation (AntRSAR) by Jensen and Shen [11]
- Genetic algorithm (GenRSAR) by Jensen and Shen [11]
- Ant colony optimisation (ACOAR) by Ke et al. [12]
- Scatter search (SSAR) by Wang et al. [32]

Table 9: Comparison with single solution-based meta-heuristic methods in the literature

Dataset	***GA_T-AR***	***SimRSAR***	***TSAR***	***IS-CNS***	***HVNS-AR***	***CHH_RSAR***	***GD-RSAR***	***EMC-FS***
M-of-N	6	6	6	6	6	$6^{(11)}\ 7^{(9)}$	$6^{(10)}7^{(10)}$	6
Exactly	6	6	6	6	6	$6^{(13)}\ 7^{(7)}$	$6^{(7)}7^{(10)}8^{(3)}$	6
Exactly2	10	10	10	10	10	10	$10^{(14)}11^{(6)}$	10
Heart	6	$6^{(29)}\ 7^{(1)}$	6	6	6	6	$9^{(4)}10^{(16)}$	$5^{(3)}\ 6^{(17)}$
Vote	8	$8^{(15)}$ 9($^{15)}$	8	8	8	8	$9^{(17)}10^{(3)}$	8
Credit	$8^{(6)}9^{(13)}$ 10 (1)	$8^{(18)}\ 9^{(1)}\ 11^{(1)}$	$8^{(13)}$ 9($^{5)}$ $10^{(2)}$	$8^{(10)}\ 9^{(9)}\ 10^{(1)}$	$8^{(7)}9^{(6)}\ 10^{(7)}$	$8^{(10)}\ 9^{(7)}\ 10^{(3)}$	$11^{(11)}12^{(9)}$	8
Mushroom	4	4	$4^{(17)}\ 5^{(3)}$	4	4	4	$4^{(8)}5^{(9)}6^{(3)}$	4
Letters	8	8	$8^{(17)}\ 9^{(3)}$	8	8	8	$8^{(7)}9^{(13)}$	8
LED	5	5	5	5	5	5	$8^{(14)}7^{(6)}$	5
Derm	6	$6^{(12)}7^{(8)}$	$6^{(14)}\ 7^{(6)}$	$6^{(18)}\ 7^{(2)}$	$6^{(16)}\ 7^{(4)}$	6	$12^{(14)}13^{(6)}$	6
Derm2	$8^{(8)}9^{(11)}$ 10 (1)	$8^{(3)}\ 9^{(7)}$	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(4)}9^{(16)}$	$8^{(5)}9^{(12)}10^{(3)}$	$8^{(5)}\ 9^{(5)}\ 10^{(10)}$	$11^{(14)}12^{(6)}$	$8^{(19)}\ 9^{(1)}$
WQ	$12^{(3)}13^{(17)}$	$13^{(16)}\ 14^{(4)}$	$12^{(1)}13^{(13}$ $14^{(6)}$	$12^{(2)}\ 13^{(8)}14^{(10)}$	$12^{(3)}13^{(6)}14^{(8)}\ 15^{(3)}$	$12^{(13)}\ 14^{(7)}$	$15^{(14)}16^{(6)}$	$12^{(17)}14^{(3)}$
Lung	4	$4^{(7)}\ 5^{(12)}\ 6^{(1)}$	$4^{(6)}\ 5^{(13)}$ 6(1)	$4^{(17)}\ 5^{(3)}$	$4^{(16)}\ 5^{(4)}$	$4^{(10)}\ 5^{(7)}\ 6^{(3)}$	$4^{(5)}5^{(2)}6^{(13)}$	4

The comparison results with other available approaches are given in Tables 9 and 10. The comparison with single solution-based meta-heuristic methods shows that GA_T-AR is able to obtain minimal attributes on 10 out of 13 datasets on all 20 runs. It also shows that GA_T-AR is generally comparable with its close competitor, i.e. EMC-FS (see Table 9), except on four datasets (Heart, Credit, Derm2 and WQ). The comparison with six other single solution-based meta-heuristic methods shows that the results obtained by GA_T-AR are better or at par with others. Conversely, the comparison with population-based meta-heuristic methods shows that the GA_T-AR outperformed GenRSAR on six datasets, which is better than AntRSAR and SSAR on seven and four datasets, respectively. The closest competitor is the ACOAR, where GA_T-AR is better on one dataset (Derm2) and worse on two (Credit and WQ). Furthermore, GA_T-AR obtained the same results as GenRSAR, AntRSAR, ACOAR and SSAR for six, six, eight and ten datasets, respectively (see Table 10).

The results presented in Tables 9 and 10 clearly show that GA_T-AR outperformed single solution-based and population-based methods on certain datasets. It is believed that using a tournament selection strategy within GA could avoid a bias selection in comparison to the roulette wheel and rank-based selection strategies. Thus, it helps the GA to have better diversity to explore a bigger search space, leading to better solution quality

Table 10: Comparison with population-based meta-heuristic methods in the literature

Dataset	***GA_T-AR***	***AntRSAR***	***GenRSAR***	***ACOAR***	***SSAR***
M-of-N	6	6	$6^{(6)}7^{(12)}$	6	6
Exactly	6	6	$6^{(10)}7^{(10)}$	6	6
Exactly2	10	10	$10^{(9)}11^{(11)}$	10	10
Heart	6	$6^{(18)}\ 7^{(2)}$	$6^{(18)}\ 7^{(2)}$	6	6
Vote	8	8	$8^{(2)}\ 9^{(18)}$	8	8
Credit	$8^{(6)}9^{(13)}10^{(1)}$	$8^{(18)}9^{(1)}10^{(4)}$	$10^{(6)}\ 11^{(14)}$	$8^{(16)}\ 9^{(4)}$	$8^{(9)}9^{(8)}10^{(3)}$
Mushroom	4	4	$5^{(1)}\ 6^{(5)}\ 7^{(14)}$	4	$4^{(12)}\ 5^{(8)}$
Letters	8	8	$8^{(8)}\ 9^{(12)}$	8	$8^{(5)}9^{(15)}$
LED	5	$5^{(12)}\ 6^{(4)}\ 7^{(3)}$	$6^{(1)}\ 7^{(3)}8^{(16)}$	5	5
Derm	6	$6^{(17)}\ 7^{(3)}$	$10^{(6)}\ 11^{(14)}$	6	6
Derm2	$8^{(8)}9^{(11)}\ 10^{(1)}$	$8^{(3)}\ 9^{(17)}$	$10^{(6)}\ 11^{(16)}$	$8^{(4)}\ 9^{(16)}$	$8^{(2)}9^{(18)}$
WQ	$12^{(3)}13^{(17)}$	$12^{(2)}\ 13^{(7)}\ 14^{(11)}$	16	$12^{(4)}13^{(12)}14^{(4)}$	$13^{(4)}14^{(16)}$
Lung	4	4	$6^{(8)}\ 7^{(12)}$	4	4

6. Conclusion

This study investigated the effect of GA with different selection strategies on the quality of the final solution in the rough set attribute reduction problem. Experiment results showed the weakness of roulette wheel selection in solving the problem; it generally failed to produce acceptable results. Meanwhile, rank-based and tournament selection strategies in GA achieved a good success rate when applied to attribute reduction problems. The findings showed that tournament and rank-based selection strategies outperformed basic GA (with roulette wheel selection). Both selection strategies were proven to produce competitive results in comparison to other results available in the literature.

In the tournament selection strategy, several individuals were selected at random from a larger population, where the selected individuals were made to compete against one another. This gave all individuals the opportunity to be selected, there by preserving diversity while avoiding the risk of premature convergence. The results demonstrated that the GAT-AR performed better than the roulette-wheel and rank-based selection strategies as well as other published meta-heuristic algorithms. Furthermore, it was capable of attaining the best-known results on 10 out of 13 datasets. However, several alternative experimentation methods and testing choices were available, which were not pursued during the study. These suggestions are proposed as further recommendations of the study. Firstly, a diverse initial population could be based on different constructive heuristics instead of being random. Secondly, the diversity of the solution should be maintained within the population. Finally, another recommendation is to apply adaptive changes to parameter values based on the quality of the solution.

Acknowledgements

This work was supported by the Universiti Kebangsaan Malaysia (grant number DIP-2016-024).

References

[1] Pawlak, Z. (1982). Rough sets. International Journal of Computer and Information Sciences, 115, 341–356.

[2] Polkowski, L. & Skowron, A. (1998). Rough sets: A perspective. Rough Sets in Knowledge Discovery, 1, 31–56.

[3] Jensen, R.& Shen, Q. (2004). Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches, IEEE Transactions on Knowledge and Data Engineering, 16(12), 1457–1471.

[4] Hedar, A.R., Wang, J.& Fukushima, M. (2008). Tabu search for attribute reduction in rough set theory, Soft Computing, l. 12(9), 909–918.

[5] Arajy, Y. Z.& Abdullah, S. (2010). Hybrid variable neighbourhood search algorithm for attribute reduction in Rough Set Theory, Intelligent Systems Design and Applications ISDA, 1015–1020.

[6] Jihad, S.& Abdullah, S. (2010). Investigating composite neighbourhood structure for attribute reduction in rough set theory, Intelligent Systems Design and Applications ISDA, 1183–1188.

[7] Abdullah, S., Sabar, N. R., Nazri, M. Z. A., Turabieh, H., & McCollum, B. (2010). A constructive hyper-heuristics for rough set attribute reduction, 2010 10th International Conference on Intelligent Systems Design and Applications, 1032-1035.

[8] Jaddi, N. S.& Abdullah, S.(2013a). Nonlinear great deluge algorithm for rough set attribute reduction, Journal of Information Science and Engineering, 29, 49–62.

[9] Abdullah, S., Sabar, N. R., Nazri, M. Z. A.& Ayob, M. (2014). An Exponential Monte-Carlo algorithm for feature selection problems, Computers & Industrial Engineering, 67, 160–167.

[10] Mafarja, M.& Abdullah, S. (2015). Fuzzy record to record travel algorithm in solving rough set attribute reduction, International Journal of Systems Science, vol. 46(3),503–512.

[11] Jensen, R.& Shen, Q. (2003). Finding rough set reducts with ant colony optimization, Proceedings of the 2003 UK workshop on computational intelligence, 15–22.

[12] Ke, L., Feng, Z.& Ren, Z. (2008). An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recognition Letters, 29(9), 1351–1357.

[13] Jue, W., Hedar, A. R., Guihuan, Z.& Shouyang, W. (2009). Scatter search for rough set attribute reduction, Computational Sciences and Optimization 2009, CSO, 1, 531–535.

[14] Hedar, A. R., Ibrahim, A. M. M., Abdel-Hakim, A. E.& Sewisy, A. A. (2018). Modulated clustering using integrated rough sets and scatter search attribute reduction.Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, 1394–1401.

[15] Mafarja, M.& Mirjalili, S. (2018a). Whale optimization approaches for wrapper feature selection, Applied Soft Computing, 62, 441–453.

[16] Mafarja, M. M.& Mirjalili, S. (2018b). Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection, Soft Computing, 1–17.

[17] Mafarja, M.& Abdullah, S. (2015). A fuzzy record-to-record travel algorithm for solving rough set attribute reduction, International Journal of Systems Science, 463, 503–512.

[18] Mafarja, M.& Abdullah, S. (2013). Investigating memetic algorithm in solving rough set attribute reduction, International Journal of Computer Applications in Technology, 48(3), 195–201.

[19] Tiwari, A. K., Shreevastava, S., Som, T.& Shukla, K. K. (2018). Tolerance-based intuitionistic fuzzy-rough set approach for attribute reduction, Expert Systems with Applications, 101, 205–212.

[20] Mafarja, M. & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing, 260, 302–312.

[21] Gao, C., Lai, Z., Zhou, J., Zhao, C.& Miao, D. (2018). Maximum decision entropy-based attribute reduction in decision-theoretic rough set model, Knowledge-Based Systems, 143, 179–191.

[22] Dai, J., Hu, H., Wu, W. Z., Qian, Y.& Huang, D. (2018). Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets, IEEE Transactions on Fuzzy Systems, 264, 2174–2187.

[23] Jia, X., Shang, L., Zhou, B.& Yao, Y. (2016). Generalized attribute reduct in rough set theory, Knowledge-Based Systems, 91, 204–218.

[24] Chen, H., Li, T., Cai, Y., Luo, C.& Fujita, H. (2016). Parallel attribute reduction in dominance-based neighbourhood rough set, Information Sciences, 373, 351–368.

[25] Chen, D., Yang, Y.& Dong, Z. (2016).An incremental algorithm for attribute reduction with variable precision rough sets, Applied Soft Computing, 45, 129–149.

[26] Wang, C., Shao, M., Sun, B.& Hu, Q. (2015). An improved attribute reduction scheme with covering based rough sets, Applied Soft Computing, 26, 235–243.

[27] Chen, Y., Zhu, Q.& Xu, H. (2015). Finding rough set reducts with fish swarm algorithm, Knowledge-Based Systems, 81, 22–29.

[28] Jing, S. Y. (2014). A hybrid genetic algorithm for feature subset selection in rough set theory, Soft Computing, 187, 1373–1382.

[29] Blake, C. L. & Merz, C. J. (1998).UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science.

[30] Eskandari, S. & Javidi, M. M. (2016). Online streaming feature selection using rough sets, International Journal of Approximate Reasoning, 69, 35–5.

[31] Holliday, J. D., Sani, N. & Willett, P. (2015). Calculation of substructural analysis weights using a genetic algorithm, Journal of Chemical Information and Modeling, 55(2), 214–221.

[32] Wang, J., Hedar, A. R., Zheng, G.& Wang, S. (2009). Scatter search for rough set attribute reduction, Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference, 1, 531–535.

[33] Jaddi, N. S.& Abdullah, S. (2013b). Hybrid of genetic algorithm and great deluge for rough set attribute reduction, Turkish Journal of Electrical Engineering and Computer Sciences, 21,1737–1750.