Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1737 – 1750

doi:10.3906/elk-1202-113

Hybrid of genetic algorithm and great deluge algorithm for rough set attribute reduction

Najmeh Sadat JADDI,* Salwani ABDULLAH
Data Mining and Optimization Research Group, Centre for Artificial Intelligence Technology,
National University of Malaysia, Bangi, Selangor, Malaysia

Received: 28.02.2012 • **Accepted:** 24.05.2012 • **Published Online:** 02.10.2013 • **Printed:** 28.10.2013

Abstract: The attribute reduction problem is the process of reducing unimportant attributes from a decision system to decrease the difficulty of data mining or knowledge discovery tasks. Many algorithms have been used to optimize this problem in rough set theory. The genetic algorithm (GA) is one of the algorithms that has already been applied to optimize this problem. This paper proposes 2 kinds of memetic algorithms, which are a hybridization of the GA, with 2 versions (linear and nonlinear) of the great deluge (GD) algorithm. The purpose of this hybridization is to investigate the ability of this local search algorithm to improve the performance of the GA. In both of the methods, the local search (the GD algorithm) is employed to each generation of the GA. The only difference of these methods is the rate of increase in the 'level' in the GD algorithm. The level is increased by a fixed value in the linear GD algorithm, while the nonlinear GD algorithm uses the quality of the current solution to calculate the increase rate of the level in each iteration. The 13 datasets taken from the University of California - Irvine machine learning repository are used to test the methods and compare the results with the on-hand results in the literature, especially with the original GA. The classification accuracies of each dataset using the obtained reducts are examined and compared with other approaches using ROSETTA software. The promising results show the potential of the algorithm to solve the attribute reduction problem.

Key words: Great deluge algorithm, genetic algorithm, rough set theory, attribute reduction, classification

1. Introduction

Nowadays, with the large number of attributes in most decision systems, attribute reduction is a necessary task in the preprocessing step to simplify the process of any learning algorithm (e.g., knowledge discovery, machine learning, image processing). Reducing the number of attributes reduces the complexity of any data mining task or learning algorithm. In the process of attribute reduction, a minimal subset of the original attribute set, which contains compulsory and important attributes, is looked for. The minimal subset should represent the original attribute set without losing the information [1]. Finding this minimal subset is known as the nondeterministic polynomial time (NP)-hard problem [2].

An effective mathematical tool to solve this problem is the rough set theory [3,4]. The rough set theory has been widely applied in many areas and its successes have shown the ability of this tool. Rough set theory extracts the relation of decision attributes with conditional attributes, and then, with use of this relation, the dependency degree of the attributes is calculated. This value is used to evaluate the quality of the subset.

*Correspondence: najmehjaddi@yahoo.com

The NP-hard problem is normally solved using approximate algorithms. An approximate algorithm is an algorithm that is used to find an optimum solution, but there is no guarantee that the solution is the best one. The approximate algorithms are categorized into 2 classes: single-based search and population-based search. In this paper, a special highlight to solve the attribute reduction problem has been given to a memetic algorithm, which is a hybridization of the genetic algorithm (GA) as a population-based algorithm and the great deluge (GD) algorithm as a single-based algorithm.

The hybridization is a well-known and effective way to get advantages from different algorithms to reach a better performance of either the convergence speed or the quality of the solutions. The rate of this improvement may be different when different local searches are applied. Therefore, there is additional room to investigate the performance of the different local searches to see the ability of each to increase the quality of the GA. In this paper, we examine the potential of the GD algorithm to better the action of the GA in order to solve the attribute reduction problem.

One of the various algorithms applied to this problem (e.g., simulated annealing (SimRSAR) [5], tabu search (TSAR) [6], ant colony (AntRSAR and ACORA) [7,8], scatter search (SSAR) [9]) is the GA. Many versions of GA have been presented to solve this problem (e.g., [5,10–12]). The GA is an optimization algorithm [13] that has been simulated based on the natural selection and the evolution process. The basic idea of the GA contains the encoding method, fitness function, and selection, crossover, and mutation operations.

Dueck introduced the standard GD algorithm in 1993 [14], which is an enhanced method of the simulated annealing algorithm. The GD algorithm controls the search space using a boundary 'level' and it accepts 2 kinds of solutions: the 1st is the best solution, when there is an improvement in the quality, and the 2nd is a worse solution, when the quality has a better standing compared to the current level. The GD algorithm has shown its performance in solving many optimization problems (e.g., [15–17]). The GD algorithm for rough set attribute reduction (GD-RSAR) was presented in 2010 [18]. In the GD-RSAR, the GD algorithm selects the solution when it improves the quality of the solution. If the qualities of the solutions are the same, the solution with less cardinality is accepted. It also accepts the worse solutions when its quality is better than the boundary level. A modified version of the GD algorithm (MGDAR) [19] for RSAR has shown superior results compared to the GD-RSAR.

This paper presents 2 versions of the hybrid GA with the GD algorithm, which are different in increasing the rate of the lower boundary level. In the linear GD algorithm, the level is increased by a fixed value, whereas in the nonlinear version, the level is increased based on the quality of the current solution in each iteration. These methods are examined with 13 available datasets in the University of California - Irvine (UCI) machine learning repository [20].

The rest of this paper is organized as follows: Section 2 first provides basic definitions of RSAR and then describes the GA and GD algorithms and their usage for attribute reduction. Section 3 explains the details of the proposed methods. The experimental results are discussed in Section 4, and in Section 5, conclusions and suggestions for future work of this research are provided.

2. Preliminaries

2.1. Basic definitions of RSAR

Definition 1 *(Decision system):*

Let $I = (U, A)$ *represent an information system, where* U *is a nonempty set of a finite set of objects and*A *is a nonempty finite set of attributes where* $\alpha: U \to V_\alpha$ *for each* $\alpha \in A$. *If* $A = C \cup D$ *and* $C \cap D =$

Ø, where C is a conditional attribute set and D is a decision attribute set, the information system is called a decision system.

Definition 2 *(Indiscernible relation):*

In decision system I, with any $P \subset A$, the definition of equivalence relation IND (P) is as in Eq. (1). If (x, y) IND (P), then x and y are indiscernible by their attributes from P. The set of all of the equivalence classes of IND (P) is represented by $[x]_P$.

$$IND(P) = \{(x,y) \in U^2 | \forall \alpha \in P, \alpha(x) = \alpha(y)\} \tag{1}$$

Definition 3 *(Approximations):*

In decision system$I$, if X U, the P-lower approximation $\underline{P}X$ of set X is defined as $\underline{P}X = \{x | [x]_p \subseteq X\}$ and the P-upper approximation $\bar{P}X$ is $\bar{P}X = \{x | [x]_p \cap X_{\neq}\emptyset\}$.

Definition 4 *(Positive region and dependency degree):*

In decision system$I$, if P and Q have equivalence relations on U, then the positive region can be defined by Eq. (2) and the dependency degree between P and Q is computed by Eq. (3). $|X|$ means the number of attributes that are contained in set X.

$$POS_p(Q) = \bigcup_{X \in U/Q} \underline{P}X \tag{2}$$

$$k = \gamma p(Q) = |POS_P(Q)|/|U| \tag{3}$$

A reduct is defined as R subset of the conditional attribute set C with decision attribute D, if γR $(D) = \gamma C$ (D). In this case, in order to find any dependency degree γR (D) of the reduct, we need to calculate the positive region with use of P-lower approximation.

In any decision system, there may be more than one reduct, but normally, in attribute reduction, the reduct with less cardinality where no attribute can be removed is searched to perform a reduction with high quality.

2.2. The GA and its usage for RSAR

The GA that was presented by Holland [14] uses a population of solutions, which is demonstrated by encoding. Each participant in the population contains a number of genes, which represents a unit of information. The algorithm starts by generating an initial population (can be generated by random). For an optimization problem, a solution is commonly encoded in a chromosome. Next, the quality of the population participants is evaluated by calculating the value of their fitness. The selection process is performed. The process of selection affects the search direction close to the areas that are more promising. Genetic operation symbols such as crossover (combination of 2 parents) and mutation (small random changes) are accustomed to generate new populations. The algorithm stops when the termination criterion is met. The overall process of the GA is shown in Figure 1.

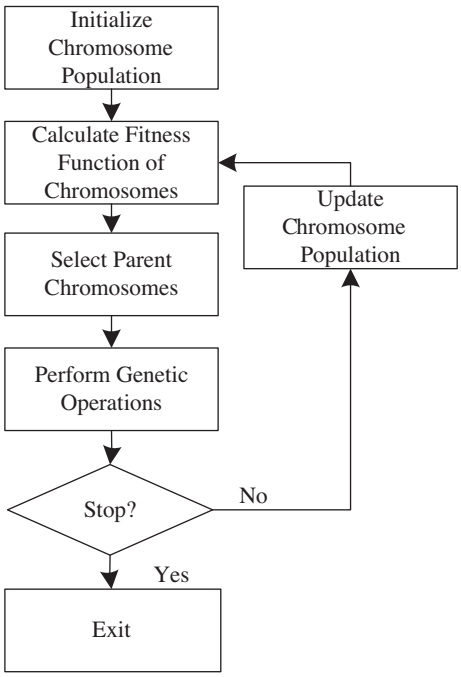

Figure 1. The overall process of the GA.

Jensen and Shen in 2003 [7] and 2004 [5] attempted to find a solution for the attribute reduction problem. One of their study areas was the GA for attribute reduction. Genetic RSAR (GenRSAR) is a GA-based method and its function and activity considers the size of the subset and its quality. The initial population consists of 100 randomly generated attribute subsets, the probabilities of mutation and crossover are respectively set to 0.4 and 0.6, and the number of generations is set to 100. The fitness function thinks about both the size of the subset and its evaluated quality.

2.3. Standard GD algorithm and its usage for RSAR

The GD algorithm [14] is one of the local search procedures in metaheuristic approaches that accept the solution when it improves the quality. It also accepts the worse solutions if the quality is better than that at the boundary level. In the initialization part, the level is set to the quality of initial solution and is then increased or decreased (based on maximization or minimization approaches) by a fixed rate, which is initialized as β. The search will be continued until the quality value reaches the estimated quality function or the number of iterations passes the specified number of iterations that has been initialized in the initialization part. The overall process of the GD algorithm for the maximization approach is shown in Figure 2.

The GD-RSAR [18] follows the standard GD algorithm. The quality function is the dependency degree of the solution; the estimated quality function (*EstimatedQuality*) is set to the highest option of the dependency degree value, the number of iterations (*NumOfIte*) is set to 250, and the level is set to the quality of the initial solution in the initialization part. The GDList that keeps the sequence of the best solutions found so far assists the GD algorithm to improve the quality of the solution during the search process.

3. Hybrid GA with the GD algorithm for RSAR

In this section, we present a combination of the GA with the GD algorithm. The purpose of combining these 2 algorithms is to examine the capability of the GD algorithm to improve the performance of the GA. Figure

3 shows a simple illustration of the hybridization of the GA with the GD algorithm, where it can be seen that the GD algorithm is embedded inside the GA to improve the quality of the solution in each generation.

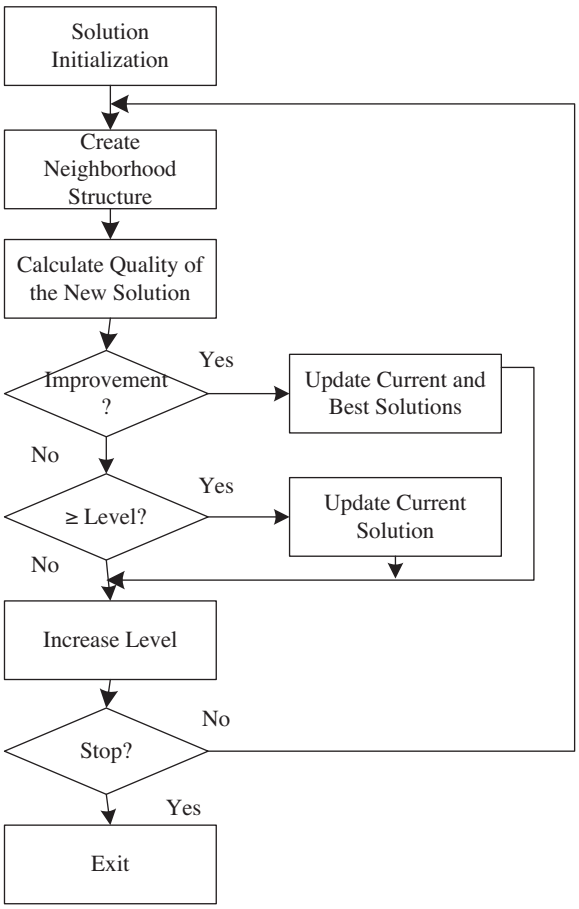

Figure 2. The overall process of the GD algorithm.

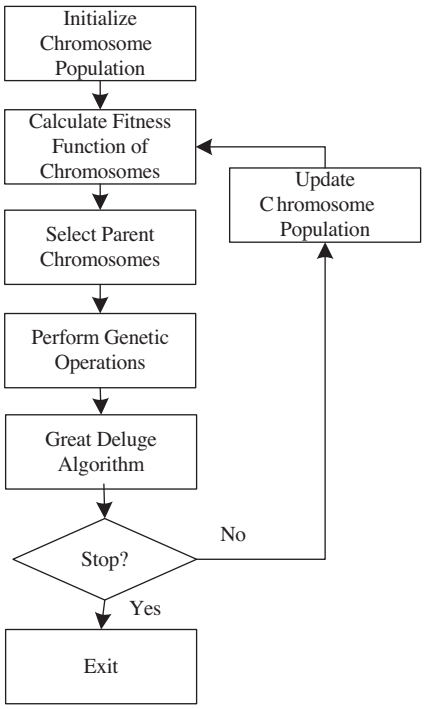

Figure 3. Hybrid of the GA and GD algorithms.

In this research, we consider 2 versions of the GD algorithm to hybridize with the GA: 1) the linear GD algorithm, which uses a fixed increasing rate for the value of the level, and 2) the nonlinear GD algorithm, where the level is increased with a value that is calculated based on the quality of the current solution. Both methods are explained in detail in this section.

3.1. Encoding candidate solutions and population construction

The proposed method employs a binary representation for each candidate solution (in GA this is usually called a chromosome). A trial solution Sol^* is a 0–1 vector, where the length of the vector is equal to the number of conditional attributes $|C|$. If $x_i = 1$ is an element of the solution $\{x_1, x_2, x_3, ...x_{|c|}\}$, then the ith attribute from the conditional attribute set is contained in the subset. If $x_i = 0$, the subset does not contain the ith attribute. The structure of a sample chromosome is illustrated in Figure 4.

In the GA, a number of chromosomes generate a population. The structure for each population can be represented as in Figure 5.

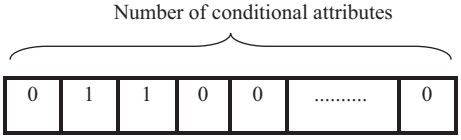

Figure 4. The structure of the solution.

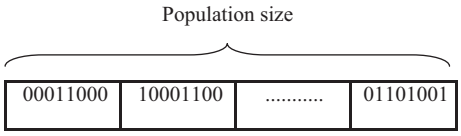

Figure 5. The structure of the population.

3.2. Fitness function

A fitness function is employed for evaluating the quality of the chromosome. In the GA, the fitness value is generally determined based on 2 issues: the number of attributes that a solution is contained in and the quality of the solution. The fitness function used in this work (as employed in [5]) is shown in Eq. (4), where R is a reduct or subset, C is a conditional attribute set, and D is a decision attribute.

$$F(R) = \gamma_R(D) * \frac{|C| - |R|}{|C|} \tag{4}$$

3.3. Selection mechanism

The selection method, which is called a roulette wheel, is used in this research. Let us consider the population $\{x_1, x_2, \ldots x_m\}$, where m is the size of the population and the fitness value of solution xi is $F(xi)$, and then the probability of x_i to be selected is calculated by Eq. (5). The bigger fitness value has the bigger probability to be selected.

$$P_F(x_i) = \frac{F(x_i)}{\sum_{i=1}^{m} F(x_i)} \text{ where } i = 1, 2, 3, \ldots m \tag{5}$$

3.4. Crossover operation

The 1-point crossover method is used to reproduce the solution with a probability of $P_C = 60\%$ (as employed in [5]). In this method, a random cut-point is selected, and then an offspring is generated. For the offspring, the segment of 1 parent from the left of the cut-point is combined with the segment of the other parents from the right of the cut-point. Figure 6 shows a sample 1-point crossover used in this work.

3.5. Mutation operation

Mutation is an operation that produces random changes in solutions. Mutation provides a chance for the solutions that have been lost from the populations. In this work, a random alteration of the gene is considered by changing from 0 to 1 or from 1 to 0, with probability $P_m = 40\%$ (as employed in [5]). In Figure 7, a sample mutation operation is shown.

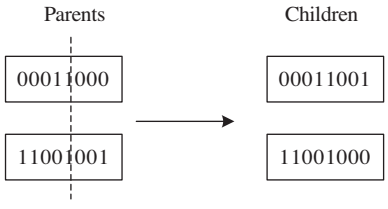

Figure 6. Sample 1-point crossover operation.

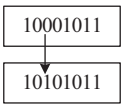

Figure 7. Sample mutation operation.

3.6. Applying the GD algorithm

The standard GD algorithm for the maximization approach uses a lower boundary level to control the search space and it increases the level with a fixed positive value. The solutions that improve the quality are always accepted. The solutions with a lower quality are accepted as the current solutions if only the quality is greater than or equal to the level. In the GD-RSAR, the solution with a higher quality is always accepted, but the solution with a quality equal to the quality of the best solution is accepted under the condition of having a lower number of attributes. The worse solution is accepted if the quality is greater than the level.

In this paper, we investigate hybridization of the GA with 2 kinds of the GD algorithm, which are different only in the way that the level is increased. The linear GD algorithm increases the level with a fixed value, while in the nonlinear GD algorithm, the level is increased by a value that is calculated based on the quality of the current solution in each iteration. Both methods are shown in Figure 8 and explained in this section:

- Linear GD

  In the initialization part, the level is set to the quality of the initial solution, γ(Sol), and the increasing rate 'β' is calculated by Eq. (6).

$$\beta = (EstimatedQuality - \gamma(Sol))/NumOfGen \tag{6}$$

- Nonlinear GD

  The level is initialized by the quality of the initial solution, γ(Sol). The increasing rate β is set to 0 in the initialization stage and then in the nonlinear level, the speed of increasing the level is controlled by an exponential function. As shown in Eq. (7), this increasing rate is calculated based on the quality of the current solution, $\gamma(CurrentSol)$, in each iteration. $\gamma(CurrentSol)$ in Eq. (7) is presented by *CurrentQuality*. According to Eq. (7), if the quality of the current solution is higher, the speed of the increasing rate is faster.

$$\beta = (\exp^{CurrentQuality} - EstimatedQuality)/NumOfGen \tag{7}$$

3.7. Elitist strategy

The fitness of the solutions in the current population is compared with the fitness of Sol_{best}. The worst solution is replaced with Sol_{best} if the fitness of Sol_{best} is greater.

4. Results and discussion

The hybrid GA models, with both the linear and nonlinear GD algorithms, are programmed in Java. The models are examined on 13 standard datasets that are available from UCI. The programs are run 20 times for each dataset. In this section, the results of these methods are compared with the other available approaches in the literature.

4.1. Number of attributes in reducts found

Table 1 presents the results from 20 runs with the number of their successes (superscripts) for all of the compared algorithms. The GAGD-RSAR and GANLGD-RSAR could not achieve the same number of attributes for all runs (the numbers without superscript). The results of the proposed methods are inferior compared to the MGDAR, TSAR, SimRSAR, AntRSAR, ACOAR, and SSAR. However, the results show better standing when

```
Set number of generation, NumOfGen ←100;
Set population size, PopSize ←100;
Generate Population;
Set a random solution from Population as initial solution, Sol;
Set best solution, Sol_best ← Sol;
Calculate the initial and best cost function, γ(Sol) and γ(Sol_best);
Set estimated quality of final solution , EstimatedQuality ← 1;
Set initial level: level ← γ(Sol);

Set increasing rate, { β = (EstimatedQuality - γ(Sol)) / NumOfGen;   → in (GAGD-RSAR)
                     { β =0;                                        → in (GANLGD-RSAR)
Set iteration ← 0;
do while (iteration < NumOfGen)
        Select parents;
          Crossover operation;
        Mutation operation;
        Evaluate the children;
        For i=1 to number of children
        Sol*= Sol_child[i];
            if (γ(Sol*) > γ(Sol_best))
                Sol ← Sol*; Sol_best ← Sol*;
                γ(Sol) ← γ(Sol*); γ(Sol_best) ← γ(Sol*);
            else
                if (γ(Sol*) == γ(Sol_best))
                    Calculate cardinality of trial solution, |Sol*|;
                    Calculate cardinality of best solution, |Sol_best|;
                    If ( |Sol*| < |Sol_best| )
                        Sol ← Sol*; Sol_best ← Sol*;
                        γ(Sol) ← γ(Sol*); γ(Sol_best) ← γ(Sol*);
                    else
                      if (γ(Sol*) ≥ level)
                            Sol ← Sol*; γ(Sol) ← γ(Sol*);
                endif
            endif
            β = (exp^CurrentQuality - EstimatedQuality) / NumOfGen;   → in (GANLGD-RSAR)
            level = level + β;
            iteration++;
        endfor
        Elitist strategy;
end do
Calculate cardinality of best solution, |Sol_best|;
return |Sol_best|, γ(Sol_best), Sol_best;
```

Figure 8. Pseudocode for the hybrid of the GA and GD algorithms.

the number of achievements is compared with GenRSAR. Only in the Heart dataset do the proposed methods show inferior results in terms of the number of achievements compared to the GA. Even when they are compared with GD-RSAR, the proposed methods show improvement in many of the datasets.

In general, the GD algorithm inside the GA shows a positive effect on the number of attributes compared

Table 1. Comparison of the number of attributes in the obtained reducts.

Datasets	No attributes	GAGD-RSAR	GANLG-RSAR	GD-RSAR	MGDAR	TSAR	SimRSAR	AntRSAR	GenRSAR	ACOAR	SSAR
M-of-N	13	$6^{(12)}7^{(8)}$	$6^{(13)}7^{(7)}$	$6^{(10)}7^{(10)}$	6	6	6	6	$6^{(6)}7^{(12)}$	6	6
Exactly	13	$6^{(11)}7^{(9)}$	$6^{(13)}7^{(7)}$	$6^{(7)}7^{(10)}8^{(3)}$	6	6	6	6	$6^{(10)}7^{(10)}$	6	6
Exactly2	13	$10^{(15)}11^{(5)}$	$10^{(17)}11^{(3)}$	$10^{(14)}11^{(6)}$	10	10	10	10	$10^{(9)}11^{(11)}$	10	10
Heart	13	$6^{(15)}7^{(5)}$	$6^{(12)}7^{(8)}$	$9^{(4)}10^{(16)}$	$6^{(14)}7^{(6)}$	6	$6^{(29)}\ 7^{(1)}$	$6^{(18)}\ 7^{(2)}$	$6^{(18)}7^{(2)}$	6	6
Vote	16	$8^{(18)}9^{(2)}$	$8^{(11)}9^{(9)}$	$9^{(17)}10^{(3)}$	8	8	$8^{(15)}\ 9^{(15)}$	8	$8^{(2)}9^{(18)}$	8	8
Credit	20	$10^{(10)}11^{(10)}$	$10^{(14)}11^{(6)}$	$11^{(11)}12^{(9)}$	$8^{(13)}9^{(3)}10^{(4)}$	$8^{(13)}\ 9^{(5)}\ 10^{(2)}$	$8^{(18)}9^{(1)}11^{(1)}$	$8^{(12)}\ 9^{(4)}\ 10^{(4)}$	$10^{(6)}11^{(14)}$	$8^{(16)}9^{(4)}$	$8^{(9)}9^{(8)}10^{(3)}$
Mush-room	22	$5^{(6)}6^{(7)}7^{(7)}$	$5^{(8)}6^{(8)}7^{(4)}$	$4^{(8)}5^{(9)}6^{(3)}$	$4^{(7)}5^{(13)}$	$4^{(17)}\ 5^{(3)}$	4	4	$5^{(1)}6^{(5)}7^{(14)}$	4	$4^{(12)}\ 5^{(8)}$
LED	24	$6^{(13)}7^{(7)}$	$6^{(7)}7^{(13)}$	$8^{(14)}9^{(6)}$	5	5	5	$5^{(12)}\ 6^{(4)}\ 7^{(3)}$	$6^{(1)}7^{(3)}8^{(16)}$	5	5
Letters	25	$8^{(11)}9^{(9)}$	$8^{(13)}9^{(7)}$	$8^{(7)}9^{(13)}$	$8^{(18)}9^{(2)}$	$8^{(17)}\ 9^{(3)}$	8	8	$8^{(8)}9^{(12)}$	8	$8^{(5)}\ 9^{(15)}$
Derm	34	$10^{(15)}11^{(5)}$	$10^{(13)}11^{(7)}$	$12^{(14)}13^{(6)}$	$6^{(11)}7^{(9)}$	$6^{(14)}\ 7^{(6)}$	$6^{(12)}\ 7^{(8)}$	$6^{(17)}\ 7^{(3)}$	$10^{(6)}11^{(14)}$	6	6
Derm2	34	$10^{(11)}11^{(9)}$	$10^{(10)}11^{(10)}$	$11^{(14)}12^{(6)}$	$8^{(4)}9^{(12)}10^{(4)}$	$8^{(2)}9^{(14)}10^{(4)}$	$8^{(3)}\ 9^{(7)}$	$8^{(3)}\ 9^{(17)}$	$10^{(4)}11^{(16)}$	$8^{(4)}9^{(16)}$	$8^{(2)}\ 9^{(18)}$
WQ	38	$15^{(12)}16^{(8)}$	$15^{(14)}16^{(6)}$	$15^{(14)}16^{(6)}$	$12^{(1)}13^{(11)}14^{(8)}$	$12^{(1)}13^{(13)}14^{(6)}$	$13^{(16)}\ 14^{(4)}$	$12^{(2)}13^{(7)}14^{(11)}$	16	$12^{(4)}13^{(12)}14^{(4)}$	$13^{(4)}\ 14^{(16)}$
Lung	56	$5^{(1)}6^{(16)}7^{(3)}$	$5^{(5)}6^{(10)}7^{(5)}$	$4^{(5)}5^{(2)}6^{(13)}$	$4^{(6)}5^{(11)}6^{(3)}$	$4^{(6)}\ 5^{(13)}\ 6^{(1)}$	$4^{(7)}5^{(12)}\ 6^{(1)}$	4	$6^{(8)}7^{(12)}$	4	4

to the performance of each algorithm alone. To have an easier comparison, Table 2 reports the average number of attributes in the best results. Table 2 also shows that the results are comparable with other available approaches in the literature.

4.2. Running time and number of calculating dependency degrees

The time taken for finding the reducts by AntRSAR, SimRSAR, and GenRSAR was reported in [5]. The analysis of the time taken appears to be unfair in that it compares times from earlier works with times from current methods. Such a comparison is not useful and the methods should be run in the same environment as the proposed methods. In this case, the number of calculating dependency degrees, which is a time-consuming process in our study, was investigated for all of the methods reported in [6] and also for GD-RSAR and our proposed approach in Figure 9.

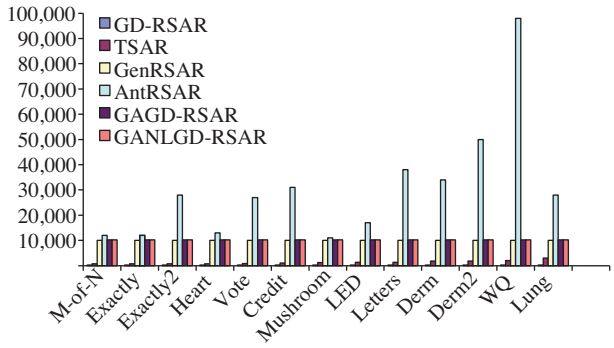

Figure 9. Comparison of the number of calculating dependency degrees.

The number of calculating dependency degrees of the proposed methods and the GenRSAR are mostly the same. In this case, we can conclude that the proposed methods with the same number of calculating dependency degrees show a better ability of reducing the number of attributes compared to GenRSAR. AntRSAR needs a large number of calculations compared to the other approaches. TSAR and GD-RSAR require a lower number of calculating dependency degrees to achieve the reducts.

4.3. Accuracy of the classification

The classification accuracy is investigated to evaluate the quality of the reducts found. The rules and the classification accuracies are generated using the ROSETTA software [21]. The 10-fold cross validation is used to predict the accuracy of the classification. The results that are presented in Table 3 are compared with the GD-RSAR [18] and GA, and Johnson's algorithm and Holte's 1R algorithm, which are available in ROSETTA.

The number of rules and the classification accuracy of our methods show competitive results. Many cases have even better achievement compared to Johnson's algorithm and Holte's 1R algorithm. The number of rules generated by the GA is extensively greater than by the proposed hybrid algorithms and GD-RSAR. The reducts found by the proposed methods were able to predict promising classification accuracy compared to the original GA, with a lower number of rules. The proposed methods are even able to reach a 100% accuracy in the M-of-N, Exactly, and LED datasets, which is a better achievement compared to the other algorithms. The outcome of this experience shows that the hybridization of the GA with the GD algorithm can be an effective method in the area of the attribute reduction problem.

Table 2. Comparison of the average number of attributes in the best reduct.

Datasets	No. attributes	GAGD-RSAR	GANLGD-RSAR	GD-RSAR	MGDAR	TSAR	SimRSAR	AntRSAR	GenRSAR	ACOAR	SSAR
M-of-N	13	6.4	6.3	6.5	6	6	6	6	6.7	6	6
Exactly	13	6.45	6.3	6.8	6	6	6	6	6.5	6	6
Exactly2	13	10.25	10.15	10.3	10	10	10	10	10.55	10	10
Heart	13	6.25	6.4	9.8	6.3	6	6.03	6.1	6.1	6	6
Vote	16	8.1	8.45	9.15	8	8	8.5	8	8.9	8	8
Credit	20	10.5	10.3	11.45	8.55	8.45	8.2	8.6	10.7	8.2	8.7
Mush-room	22	6.05	5.8	4.75	4.65	4.15	4	4	6.65	4	4.4
LED	24	6.35	6.65	8.3	5	5	5	5.6	7.75	5	5
Letters	25	8.45	8.35	8.65	8.1	8.15	8	8	8.6	8	8.75
Derm	34	10.25	10.35	12.3	6.45	6.3	6.4	6.15	10.7	6	6
Derm2	34	10.45	10.5	11.3	9	8.9	8.7	8.85	10.8	8.8	8.9
WQ	38	15.4	15.3	15.3	13.35	13.25	13.2	13.45	16	13	13.8
Lung	56	6.1	6	5.4	4.85	4.75	4.7	4	6.6	4	4

Table 3. Comparison of the number of rules and classification accuracies.

Datasets	GAGD-RSAR		GANLGD-RSAR		GD-RSAR		GA		Johnson's algorithm		Holte's 1R algorithm	
	Average number of rules	Accuracy %	Average number of rules	Accuracy %	Average number of rules	Accuracy %	Average number of rules	Accuracy %	Average number of rules	Accuracy %	Average number of rules	Accuracy %
M-of-N	64	100	64	100	64	100	8658	95	45	99	26	63
Exactly	64	100	64	100	64	100	28350	73	271	94	26	68
Exactly2	559	68	559	68	559	68	27167	74	441	79	26	73
Heart	195	50	195	50	195	50	1984	81	133	68	50	64
Vote	70	92	70	92	71	92	2517	95	54	95	48	90
Credit	527	58	527	60	527	58	47450	72	560	67	81	69
Mushroom	52	100	52	100	52	100	8872	100	85	100	112	89
LED	38	100	38	100	40	100	41908	98	316	99	48	63
Letters	19	0	19	0	19	0	1139	0	20	0	50	0
Derm	127	86	127	86	127	86	7485	94	92	89	129	48
Derm2	151	67	189	68	196	64	8378	92	98	88	138	48
WQ	135	54	130	54	136	55	48687	69	329	58	94	51
Lung	18	66	19	70	18	66	1387	71	12	69	156	73

5. Conclusion

This paper presented the hybrid GA with 2 versions of the GD algorithm. The proposal of this hybridization is to evaluate ability of the GD algorithm to improve the performance of the GA. One of these 2 versions is the linear GD algorithm and the other is the nonlinear GD algorithm. The proposed methods have embedded these 2 kinds of GD algorithm inside the GA. The presented models were tested in 13 datasets available from UCI and the results were compared with the available results in the literature, especially with the original GA. Although the results were inferior compared to other approaches, the reducts found by the proposed methods had better standing compared to those of the original GA, at least in terms of number of achievements with the same number of calculating dependency degrees. These reducts found were employed to classify the datasets using ROSETTA in order to evaluate the reducts.

The promising results show the effectiveness of the proposed attribute reduction methods. This promising result motivates us to continue our study with a weightier version of the GA. This may create changes in the selection procedure, crossover operation (examining other methods of crossover), mutation, or elitist strategy. This is an additional realm in the area of our research that is considered as our future work.

References

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Oxford, Morgan Kaufmann Publishers, 2006.

[2] Z. Pawlak, J. Grzymala, R. Slowinski, "Rough sets", Communications of the Association for Computing Machinery, Vol. 8, pp. 89–95, 1995.

[3] Z. Pawlak, "Rough sets", International Journal of Information and Computer Science. Vol. 11, pp. 341–356, 1982.

[4] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Boston, Kluwer, 1991.

[5] R. Jensen, Q. Shen, "Semantics-preserving dimensionality reduction: rough and fuzzy-rough based approaches", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp. 1457–1471, 2004.

[6] A. Hedar, J. Wang, M. Fukushima, "Tabu search for attribute reduction in rough set theory", Soft Computing, Vol. 12, pp. 909–918, 2006.

[7] R. Jensen, Q. Shen, "Finding rough set reducts with ant colony optimization", Proceedings of the UK Workshop Computational Intelligence, pp. 15–22, 2003.

[8] L. Ke, Z. Feng, Z. Ren, "An efficient ant colony optimization approach to attribute reduction in rough set theory", Pattern Recognition Letters, Vol. 29, pp. 1351–1357, 2008.

[9] J. Wang, A. Hedar, G. Zheng, S. Wang, "Scatter search for rough set attribute reduction", International Joint Conference on Computational Sciences and Optimization, Vol. 1, pp. 531–535, 2009.

[10] Z. Jun, L. Jian-yong, W. Zhen, "Rough set attribute reduction algorithm based on immune genetic algorithm", 2nd International IEEE Conference on Computer Science and Information Technology, pp. 421–424, 2009.

[11] Z. Xu, D. Gu, B. Yang, "Attribute reduction algorithm based on genetic algorithm", International Conference on Intelligent Computation Technology and Automation, pp. 169–172, 2009.

[12] B. Liu, F. Liu, X. Cheng, "An adaptive genetic algorithm based on rough set attribute reduction", International Conference on Biomedical Engineering and Informatics, pp. 2880–2883, 2010.

[13] J. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor, University of Michigan Press, 1975.

[14] G. Dueck, "New optimization heuristics: the great deluge algorithm and the record-to-record travel", Journal of Computational Physics, Vol. 104, pp. 86–92, 1993.

[15] S. Abdullah, E.K. Burke, "A multi-start large neighbourhood search approach with local search methods for examination timetabling", International Conference on Automated Planning and Scheduling, pp. 334–337, 2006.

[16] D. Landa-Silva, J.H. Obit, "Great deluge with nonlinear decay rate for solving course timetabling problems", IEEE Conference on Intelligent Systems, pp. 8.11–8.18, 2008.

[17] D. Landa-Silva, J.H. Obit, "Evolutionary nonlinear great deluge for university course timetabling", International Conference on Hybrid Artificial Intelligence Systems, 2009.

[18] S. Abdullah, N.S. Jaddi, "Great deluge algorithm for rough set attribute reduction", Database Theory and Application, Bio-Science and Bio-Technology, Communications in Computer and Information Science, Vol. 118, pp. 189–197, 2010.

[19] M. Mafarja, S. Abdullah, "Modified great deluge for attribute reduction in rough set theory", Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1464–1469, 2011.

[20] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, University of California at Irvine, http://www.ics.uci.edu/∼mlearn/, 1998.

[21] J. Komorowski, A. Øhrn, A. Skowron, The ROSETTA Rough Set Software System, in Handbook of Data Mining and Knowledge Discovery (Eds. W. Klösgen and J. Zytkow), Oxford University Press, pp. 554–559, 2002.