Reinitiate Great Deluge with Composite Neighbourhood Structures for Rough Set Attribute Reduction

Fayyad Banihani, Salwani Abdullah, Nor Samsiah Sani and Zalinda Othman

Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia

Summary
Attribute reduction problem can be defined as a process of eliminating redundant attributes, while avoiding information loss. It is known to be an NP-hard optimization problem, which deals with finding the minimal attribute from a large set of attributes. Many heuristic and meta-heuristic approaches have been widely used by researchers. However, this study will focus on the use of a reinitiate great deluge algorithm with a composite neighbourhood structure (rGD-cNbs), proposed for the rough set of attribute reduction problem. rGD-cNbs is a meta-heuristic approach, which is based on a basic great deluge. Its difference is that the level of the great deluge is reinitiate if there is no improvement for a certain number of consecutive non-improving iterations. Improved solutions are accepted, as well as worse solutions based on the current level of the rGD-cNbs. Furthermore, a composite neighbourhood structure is employed within the rGD-cNbs in order to help the algorithm to better explore the search space. This study involves the evaluation of approaches on 18 benchmark datasets that are available in UCI machine learning repository. Experimental results show that the rGD-cNbs is able to achieve competitive results in comparison with other available meta-heuristic approaches in the literature in terms of the minimal reduct.

Key words:
Artificial intelligence, attribute reduction; rough set theory; great deluge; composite neighbourhood structures

1. Introduction

The vast development of information technology as well as the massive, and growing challenges in information management have resulted in the familiarity of the term ‘big or complex data.’ It is, in fact, one of the major disruptors in the enterprises as it exists in almost all industries and sectors, such as banking, economics, medi-care, manufacturing and even government. However, big data needs a large storage space with sufficient speed and accuracy when accessing information, else it will fail to deliver knowledge directly to the company. Hence, it has become a critical issue for enterprises.

Conversely, the data themselves need to be transformed into meaningful knowledge and information before being useful in any organizations. Such knowledge is essential for analysts and managers to assist them in making decisions. As such, data mining techniques, algorithms, and data mining software become critical in solving the above mentioned problems.

Research has shown that attribute reduction plays a vital role in scientific development, and has been considered as an NP-hard improvement issue [1]. Its functional work is by getting the least attribute set derived out of a huge dataset of attributes, as well as taking out unrelated and repeated parameters. This process aims at modifying data superiority by managing any abnormality and nebulousness which may exist. One of the most popular theories in this field is the Rough Set Theory introduced initially by Pawlak [2]. This theory highlights the provision of the estimation of a confusing approach set up by dual sophisticated techniques, named as the lower and upper approximations.

Lately, there have been a number of scholars proposing meta-heuristic procedures to sort out the attribute reduction problems. Example 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 [8], constructive hyper-heuristics [9], exponential monte carlo [10], and fuzzy record-to-record [11]. Example of population-based approaches are ant colony [12], [13], and scatter search [14], [15], a whale optimization approach [16], and binary ant lion optimizer [17]. Also, hybrid approaches have been investigated on attribute reduction problems, such as the hybridization between fuzzy logic and record-to-record travel algorithm [18], a hybrid genetic algorithm with great deluge [8], and memetic algorithm [19]. Other approaches on attribute reduction can be found in [20], [21], [16], [17], [22]-[29].

This study focuses on investigating the performance of the reinitiate level great deluge algorithm called (RLGD_RSAR) in comparison with other available meta-heuristic approaches for attribute reduction problem. RLGD-RSAR is an intelligent mechanism to manage and reinitiate the value of the ‘level’ by sensing the lack of improvement for a certain number of repetitions, plus the use of three different composite neighbourhood structures. In this study, the test on RLGD-RSAR was carried out on 18 public domain datasets available in UCI machine learning repository.

Manuscript received October 5, 2019
Manuscript revised October 20, 2019

In achieving the aims of this study, the paper is structured as follows; section material and methods present the description on the attribute reduction problems and elaborate on the details of the proposed algorithm, while experimental study and discussions are presented in the results and discussion section. Finally, the main conclusions drawn from this study are also presented.

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 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 $0 \leq k \leq 1$, 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 $\gamma\{1\} =1/8$; $\gamma\{2\} =0$; $\gamma\{3\} =0$; $\gamma\{4\} =2/8$; $\gamma\{1,2\}=1/8$; $\gamma\{1,3\} =5/8$; $\gamma\{1,4\} =1$; $\gamma\{2,3\}=0$; $\gamma\{2,4\} =2/8$; $\gamma\{3,4\} =3/8$; $\gamma\{1,2,3\} =5/8$; $\gamma\{1,2,4\}=1$; $\gamma\{1,3,4\}=1$; $\gamma\{2,3,4\}=3/8$;

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. Materials and Method

A. Problem Description

In this section, the attribute reduction problem is described, as well as solution representation and the objective function.

1) Attribute Reduction Problems: This is a pre-processing task in data mining and can be represented by a pair of (A, c), where A represents the original set of attributes and c is the objective function that 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 [24], [28].
2) Solution Representation: This is represented as a one-dimensional array, where the array size is equal to the attribute number plus two additional cells representing a number of selected attribute and dependency value (calculated from the rough set theory). The cell with the value of zero indicates that the attribute is not selected, while the cell with the value of one means the attribute is selected. Fig. 1 demonstrates the sample of initial solution with fifteen attributes (A1 to A8), where 4 attributes are selected (A2, A3, A7, A8) and the value of the dependency degree is equal to 0.4.

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	No of selected attributes	Dependency degree
0	1	1	0	0	0	1	1	4	0.4

Fig. 1 Initial solution representation.

3) Objective Function: In this study, the generated subset of the attribute is evaluated based on the dependency degree of rough set theory [2], [30] as its objective function. The dependency degree calculates data dependencies and returns a value between zero and one, where the value of one means that the generated subset of the attribute is informative. Furthermore, the dependency degree that is equal to one in all the generated subset of attributes is maintained in this study by deleting or adding attributes from a given subset. Given two subsets of attribute, the subset with the lowest number of attributes will be accepted.

B. Reinitiate Great Deluge with Composite Neighbourhood Structures

This study proposes the use of a reinitiate great deluge with composite neighbourhood structures to deal with the attribute reduction problem (coded as rGD-Nbs). The following subsections cover the constructive heuristic method, neighbourhood structures, and the rGD-Nbs algorithm.

1) Constructive Heuristic: The initial solution is constructed randomly by distributing zeros and ones into each cell in the one-dimensional binary vector.
2) Composite Neighbourhood Structures: Three neighbourhood structures are employed in this study. Firstly, randomly flip one point (coded as 1Flip-Neig) is employed, where one point is selected. If the selected point is "1", then it will be changed to "0". Secondly, randomly flip two points (coded as 2Flip-Neig), where two points are selected, thus the operation is as in the first. Thirdly, randomly flip three points (coded as 3Flip-Neig), where three points are selected, and the operation is as in the first respectively.
3) The Reinitiate Great Deluge with Composite Neighbourhood Structures Algorithm: Fig. 2 illustrates the pseudo code that represents our approach that is presented in 3 stages (parameter initialization, initial solution generation and solution improvement). In the parameter initialization, a number of iterations are defined

as NumOfIte, an estimated quality of the final solution as EstimatedQuality and non-improving counter as NonImproveCounter. The parameter values are obtained from the preliminary experiments as presented in Table 3.

Table 3: Parameter Initialization

Parameter	Value	Reference
Estimated Quality	Depends on datasets	Mafarja and Abdullah (2014)
NumOfIte	250	Ke et al. (2008)
NonImproveCounter	10	Lenin et al. (2014)

A decreasing rate, β which is calculated:

$$\beta = (f(Sol) - EstimatedQuality) / (NumOfIte) \qquad (11)$$

Based on the Equation (11) *Sol* is an initial solution that is constructed using a constructive heuristic. The *level* is equal to the initial solution, *f*(*Sol*) at the start and will decrease by the value β. The do-while loop neighbour solutions are defined by employing three neighbourhood structures that is coded as Sol_1, Sol_2 and Sol_3 respectively. The best among three neighbour solutions is then selected and referred to as *Sol**. However, the best solution will always be accepted, while the worse solution is accepted if the objective function value of the new solution, *f*(*Sol**) is lower than the *level* and the current solution is updated. Otherwise, non-improving counter, *counter* is updated. The level is updated as *level*=*level*-β.

```
Parameters Initialization
Set estimated quality of the final solution (minimum number of
attributes), EstimatedQuality;
Set number of iteration, NumOfIte;
Set non improvement counter, NonImproveCounter;

Initial Solution Generation
Set initial solution as Sol, obtained from constructive heuristic;
Set best solution, Solbest ← Sol;
Calculate the initial and best objective function, f(Sol) and
f(Solbest);
Set level= f(Sol);
Set decreasing rate, β;
Set counter= 0;

Solution Improvement
do while (NumOfIte!=0)
        Obtain a candidate solution, Sol1 with flip one point
on Sol;
        Obtain a candidate solution, Sol2 with flip two
points on Sol;
        Obtain a candidate solution, Sol3 with flip three
points on Sol;
        Evaluate candidate solutions, f(Sol1, Sol2, Sol3);
        Select best solution, Sol* from So11, So12, So13;
        if (f(Sol*) < f(Sol))
                Sol←Sol*; f(Sol) ←f(Sol*);
                Solbest ←Sol*; f(Solbest) ← f(Sol*);
        else
                if (f(Sol*) ≤ level)
                    Sol ←Sol*; f(Sol) ← f(Sol*);
                    Solbest ← Solbest; f(Solbest) ←
        f(Solbest);
                else
                        Sol ←Sol; f(Sol) ← f(Sol);
                        counter++;
                level = level - β;
                endif
        endif
    if(counter==NonImproveCounter)
        level = f(Sol*);
    endif
    NumOfIte--;
end do;
return Solbest, f(Solbest);
```

Fig. 2 Pseudocode of rGD-cNbs

4. Results and Discussions

The rGD-cNbs programmed in C#, and the 13 well-known UCI datasets (Blake & Merz, 1998) were used to test the performance of the rGD-cNbs. For each dataset, the algorithm was run 20 times, and the performance of the rGD-cNbs is compared with other available approaches in the literature. The results derived in this study are then compared with other algorithms available in previous literature. Note that the chosen methods for comparison is based on the methods that employed the idea of using rough set theory to measure the dependency between attributes, that are categorized 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 Header, Wang and Fukushima [15]
- 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 et al. [9]
- Great Deluge Algorithm (GD-RSAR) by Abdullah and Jaddi [7]
- An Exponential Monte-Carlo algorithm (EMC-FS) by Abdullah et al. [10]

Population-based meta-heuristic methods:

- Ant Colony Optimisation (AntRSAR) by Jensen and Shen [12]
- Genetic Algorithm (GenRSAR) by Jensen and Shen [12]
- Ant Colony Optimisation (ACOAR) by Ke et al. [13]
- Scatter search (SSAR) by Wang et al. [31]

The comparison results with other available approaches are given in Table 4 and Table 5 respectively. A comparison with single solution-based metaheuristic methods shows that rGD- The comparison results with other available approaches are given in Table 4 and Table 5 respectively. A comparison with single solution-based metaheuristic methods shows that rGD-Nbs is able to attain minimal attributes on 10 out of 13 datasets from the 20 runs. It also shows that rGD-Nbs is generally comparable to its closest competitor (EMC-FS) as presented in Table II, except on 2 datasets (Heart and Credit). The comparison with the other six single solution-based meta-heuristic methods also shows that the results obtained by rGD-Nbs are on par. On the other hand, the comparisons with population-based meta-heuristic methods show that the rGD-Nbs outperforms GenRSAR, AntRSAR and SSAR on 7 (Credit, Mushroom, LED, Derm, Derm2, WQ and Lung), 6 (Heart, Vote, Credit, Derm, WQ and Lung) and 4 (Credit, Mushroom, LED and WQ) datasets, respectively. Furthermore, rGD-Nbs obtained the same results as GenRSAR, AntRSAR, ACOAR and SSAR for M-of-N, Exactly, Exactly2 and Letters datasets (see Table 4).

Table 4: Comparison with Single Solution-based Meta-Heuristic Methods in the Literature

Dataset	rGD-Nbs	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	$8^{(16)}9^{(4)}$	$8^{(18)}\ 9^{(1)}11^{(1)}$	$8^{(13)}\ 9(^{5)}\ 10^{(2)}$	$8^{(10)}\ 9^{(9)}10^{(1)}$	$8^{(7)}\ 9^{(6)}\ 10^{(7)}$	$8^{(10)}\ 9^{(7)}\ 10^{(3)}$	$11^{(11)}\ 12^{(9)}$	8
Mushroom	4	4	$4^{(17)}\ 5^{(3)}$	4	4	4	$4^{(8)}\ 5^{(9)}\ 6^{(3)}$	4
Letters	5	5	5	5	5	5	$8^{(14)}\ 9^{(6)}$	5
LED	8	8	$8^{(17)}\ 9^{(3)}$	8	8	8	$8^{(7)}\ 9^{(13)}$	8
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	$8^{(4)}9^{(16)}$	$8^{(3)}\ 9^{(7)}$	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(4)}\ 9^{(16)}$	$8^{(5)}\ 9^{(12)}\ 10^{(3)}$	$8^{(5)}\ 9^{(5)}\ 10^{(10)}$	$11^{(14)}\ 12^{(6)}$	$8^{(19)}\ 9^{(1)}$
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
Zoo	5	-	-	-	-	-	-	-
WineEw	$5^{(12)}6^{(8)}$	-	-	-	-	-	-	-
Lymphography	$8^{(16)}9^{(4)}$	-	-	-	-	-	-	-
Tic-tac-toe	3	-	-	-	-	-	-	-
Breastcancer	$4^{(10)}5^{(10)}$	-	-	-	-	-	-	-

This study also includes 5 new datasets (Zoo, WineEw, Lymphography, Tic-tac-toe and Breastcancer) to be tested on rGD-cNbs. Experimental results show that the rGD-cNbs is able to generate a minimum number of attributes on 2 datasets (Zoo, Tic-tac-toe) from all 20 runs. While for other datasets (WineEw, Lymphography, Breastcancer), the rGD-cNbs shows promising results. Conversely, the results that were obtained using these 5 datasets are not comparable to other methods available in previous literature as they are newly available datasets. Both results presented in Table 4 and Table 5 respectively, clearly show that rGD-Nbs outperformed both single solution-based and population-based methods on certain datasets. This performance shows the applicability of the proposed approach as an alternative problem solver in attribute reduction problems. Principally, it can be stated that our proposed algorithm is competitive, efficient and functions well across all over datasets. This is due to the reinitiate level process with composite neighbourhood structures that have helped to improve exploration of the search space, while looking for more near-optimal solutions.

Table 5: Comparison with Population-based Meta-Heuristic Methods in the Literature

Dataset	rGD-Nbs	AntRSAR	GenRSAR	ACOAR	SSAR
M-of-N	6	6	6	6	6
Exactly	6	6	6	6	6
Exactly2	10	10	10	10	10
Heart	6	$6^{(29)}\ 7^{(1)}$	6	6	6
Vote	8	$8^{(15)}\ 9(^{15)}$	8	8	8
Credit	$8^{(16)}9^{(4)}$	$8^{(18)}\ 9^{(1)}\ 11^{(1)}$	$8^{(13)}\ 9^{(5)}\ 10^{(2)}$	$8^{(16)}\ 9^{(4)}$	$8^{(9)}\ 9^{(8)}10^{(3)}$
Mushroom	4	4	$4^{(17)}\ 5^{(3)}$	4	$4^{(12)}\ 5^{(8)}$
Letters	5	5	5	5	5
LED	8	8	$8^{(17)}\ 9^{(3)}$	8	$8^{(5)}9^{(15)}$
Derm	6	$6^{(12)}\ 7^{(8)}$	$6^{(14)}\ 7^{(6)}$	6	6
Derm2	$8^{(4)}9^{(16)}$	$8^{(3)}\ 9^{(7)}$	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(4)}\ 9^{(16)}$	$8^{(2)}9^{(18)}$
WQ	$12^{(3)}13^{(17}$	$13^{(16)}\ 14^{(4)}$	$12^{(1)}13^{(13)}\ 14^{(6)}$	$12^{(4)}\ 13^{(12)}14^{(4)}$	$13^{(4)}14^{(16)}$
Lung	4	$4^{(7)}\ 5^{(12)}\ 6^{(1)}$	$4^{(6)}\ 5^{(13)}\ 6^{(1)}$	4	4
Zoo	5	-	-	-	-
WineEw	$5^{(12)}6^{(8)}$	-	-	-	-
Lymphography	$8^{(16)}9^{(4)}$	-	-	-	-
Tic-tac-toe	3	-	-	-	-
Breastcancer	$4^{(10)}5^{(10)}$	-	-	-	-

5. Conclusions

This study has evaluated the attribute reduction problem, and a reinitiate level great deluge algorithm with composite neighbourhood structures (rGD-Nbs) which has been proposed as a potential solution. Numerical experiments on 13 well-known datasets have also been presented to show the effectiveness of the rGD-Nbs in producing the smallest subset of features when compared to current existing approaches. The promising results demonstrate the effectiveness of the proposed attribute reduction method, motivating further study in applying an adaptive mechanism to control the parameters of the proposed algorithm, subject to future study.

Acknowledgements

This study is supported by the Universiti Kebangsaan Malaysia (DIP-2016-024, and PP-FTSM-2019).

References

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

[2] Pawlak, Z. (1982) Rough sets. International journal of computer & information sciences, 11(5) (pp.341-356).

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

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

[5] Arajy, Y. Z and Abdullah, S. (2010) Hybrid variable neighbourhood search algorithm for attribute reduction in Rough Set Theory, In Intelligent Systems Design and Applications ISDA, 2010 10th International Conference (pp.1015-1020).

[6] Jihad, S and Abdullah, S. (2010) Investigating composite neighbourhood structure for attribute reduction in rough set theory, In Intelligent Systems Design and Applications ISDA, 2010 10th International Conference (pp.1183-1188).

[7] Abdullah, S and Jaddi, N. S. (2010) Great deluge algorithm for rough set attribute reduction, In Database theory and application, bio-science and bio-technology (pp.189-197).

[8] Jaddi, N. S and Abdullah, S. (2013) Hybrid of genetic algorithm and great deluge algorithm for rough set attribute reduction, Turkish Journal of Electrical Engineering and Computer Sciences, vol. 216 (pp.1737-1750).

[9] Abdullah, S., Sabar, N. R., Nazri, M. Z. A., Turabieh, H., and McCollum, B. (2010) A constructive hyper-heuristics for rough set attribute reduction, In Intelligent Systems Design and Applications (ISDA) (pp.1032-1035), IEEE.

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

[11] Mafarja, M., and Abdullah, S. (2015) Fuzzy record to record travel algorithm in solving rough set attribute reduction, International Journal of Systems Science, vol. 46: (pp.503-512).

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

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

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

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

[16] Mafarja, M and Mirjalili, S. (2018a) Whale optimization approaches for wrapper feature selection. Applied Soft Computing, vol. 62 (pp. 441-453).

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

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

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

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

[21] Mafarja, M. M and Mirjalili, S. (2017) Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, Neurocomputing, vol. 260 (pp. 302-312).

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

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

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

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

[26] Chen, D., Yang, Y and Dong, Z. (2016) An incremental algorithm for attribute reduction with variable precision rough sets, Applied Soft Computing, vol. 45 (pp. 129-149).

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

[28] Jing, S. Y. (2014) A hybrid genetic algorithm for feature subset selection in rough set theory, Soft Computing, vol. 187 (pp. 1373-1382).
[29] Jing, S. Y. (2014) A hybrid genetic algorithm for feature subset selection in rough set theory, Soft Computing, vol. 187 (pp. 1373-1382).
[30] Pawlak, Z. (1991) Imprecise categories, approximations and rough sets. In Rough sets (pp. 9-32).
[31] Wang, J., Hedar, A. R., Zheng, G and Wang, S. (2009) Scatter search for rough set attribute reduction, In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference, vol. 1 (pp. 531-535) IEEE.