Comparison between Record to Record Travel and Great Deluge Attribute Reduction Algorithms for Classification Problem

Majdi Mafarja[1, 2], Salwani Abdullah[1]

[1]Data Mining and Optimization Research Group (DMO), Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi Selangor, Malaysia.

[2]Department of Computer Science, Faculty of Information Technology
Birzeit University, P.O. Box 14, Birzeit, Palestine

mmafarja@birzeit.edu
salwani@ftsm.ukm.my

Abstract. In this paper, two single-solution-based meta-heuristic methods for attribute reduction are presented. The first one is based on a record-to-record travel algorithm, while the second is based on a Great Deluge algorithm. These two methods are coded as RRT and m-GD, respectively. Both algorithms are deterministic optimisation algorithms, where their structures are inspired by and resemble the Simulated Annealing algorithm, while they differ in the acceptance of worse solutions. Moreover, they belong to the same family of meta-heuristic algorithms that try to avoid stacking in the local optima by accepting non-improving neighbours. The obtained reducts from both algorithms were passed to ROSETTA and the classification accuracy and the number of generated rules are reported. Computational experiments confirm that RRT m-GD are able to select the most informative attributes which leads to a higher classification accuracy.

Keywords: Record to Record Travel algorithm, Great Deluge algorithm, Rough Set Theory, Classification

1 Introduction

Attribute Reduction (AR) (or data reduction) is regarded as an important preprocesing technique in machine learning and in the data mining process [1-2]. It can be defined as the problem of finding a minimum reduct (subset) from the original set [3] .

The attribute reduction process aims to eliminate the irrelevant and redundant attributes from high-dimensional data sets which increase the chances that a data mining algorithm will find spurious patterns that are not valid in general. Furthermore, when dealing with high-dimensional data sets, a longer time is needed to find the desired results. Liu and Motoda [3] indicate that the purposes of AR are to: (a) improve the performance (speed of learning, predictive accuracy, or simplicity of rules);

(b) visualise the data for model selection; and (c) reduce the dimensionality and remove the noise.

Langley [4] divided the attribute reduction methods into two types of model (i.e., filter and wrapper) based on their dependence on the inductive algorithm that will finally use the selected subset. In a filter model, the selection process is performed independently from the induction algorithm. A wrapper model, which is essentially the opposite of a filter model, uses the induction algorithm to directly evaluate the feature subsets.

The rough set theory, proposed by Pawlak [5-6], has been used as a simple mechanism to determine minimal subsets by locating all of the possible reducts and selecting the one with the lowest cardinality and the highest dependency. This process is a time-consuming procedure and it is only effective for small datasets. As a result, for high-dimensional datasets, many researchers now focus on employing meta-heuristics to search for better solutions instead of using the rough set theory's reduction method. Many researchers are focusing on the problem of finding a subset with minimal attributes from an original set of data in an information system [7-16].

In the literature, many meta-heuristic-based methods which were designed to solve the attribute reduction problem can be found, such as the Genetic Algorithm [17-19], Particle Swarm Optimisation [20], Ant Colony algorithm [21-22], Tabu Search [23], Great Deluge algorithm [24], Composite Neighbourhood Structure [25], Hybrid Variable Neighbourhood Search algorithm [26] and Constructive Hyper-Heuristics [27], Bees Algorithm [28].

In this paper, we examined the affect of employing two attribute reduction methods on the classification accuracy based on 13 benchmark datasets. In this work, we investigate how the use of attribute reduction methods will influence the classification accuracy. Two attribute reduction methods were examined, Record to Record Travel algorithm (RRT) and Modified Great Deluge algorithm (m-GD).

The remainder of this paper is organised as follows: section 2 discusses the RRT algorithm, and this is followed by a detailed description of the implementation of m-GD in section 3. Section 4 presents a simulation of the proposed algorithms together with a discussion of the experimental results. Finally, concluding remarks on the effectiveness of the proposed techniques and the potential future research aspects are presented in section 5.

2 Record-to-Record Travel Algorithm

The RRT algorithm was originally proposed by Dueck [29]. It is a variant of the Simulated Annealing algorithm, with a different mechanism for accepting non-improving solutions [30]. This algorithm has a solitary parameter called the DEVIATION, which plays a pivotal role in controlling the acceptance of the worst solutions after it becomes pre-tuned. The significance of this method relates to the ease of its implementation and the required number of parameters, which influences the performance of the algorithm [29]. In this work, the RRT algorithm is applied to tackle the attribute reduction problem.

Figure 1 shows the pseudo-code of the proposed method. The algorithm starts from a randomly generated initial solution (*Sol*). The best solution (Sol_{best}) is set as *Sol,* and the *RECORD* is set as the fitness value of the best solution $f(Sol_{best})$. The initial solution is improved by searching its neighbourhood for a better solution (called Sol_{trial}). The neighbourhood of a solution (Sol_{trial}) is generated by a random flip-flop, where three cells are selected at random from the current solution (*Sol*) as in [18]. For each selected cell, if its value is '1' then it is changed to '0', which means that the feature is deleted from the current solution. Otherwise, it is changed from '0' to '1'. The cardinality of the generated trial solution should be less than the cardinality of the best solution so far, because we are trying to generate a solution with a lower cardinality and a higher quality.

Later, Sol_{trial} is evaluated (in this work, evaluation is on the dependency degree in RST). If the quality of Sol_{trial} is better or slightly worse (not more than the *DEVIATION* value) than the best value so far (the $f(Sol_{best})$), then the solution is accepted. Note that the initial value of the *RECORD* is equal to the initial fitness function. During the search process, the *RECORD* is updated with the fitness value of the best solution so far $f(Sol_{best})$. More formally, in the case of maximisation, if (Sol_{best}) is the best solution so far, and (Sol_{trial}) is the newly generated solution, then (Sol_{trial}) is accepted as the next solution if $f(Sol_{best}) - f(Sol_{trial}) < DEVIATION$, where $DEVIATION >= 0$ (is the maximum allowed *DEVIATION* that determines how much worse values than the *RECORD* will be accepted). This process is repeated until the stopping condition is met (in this work, the number of iterations is set as a stopping criterion).

Record-to-Record Travel Algorithm for Attribute Reduction (RRT)

```
Generate a random initial solution Sol;
Set Sol_best = Sol;
Set RECORD  = f(Sol_best);
Set DEVIATION = 0.09;
while (stopping-criterion is not satisfied)
    Generate at random a new solution Sol_trial from Sol;
    Calculate f(Sol_trial);
    if ( f(Sol_trial)> f(Sol_best))
            Sol ← Sol_trial ; Sol_best ← Sol_trial;
            f(Sol) = f(Sol_trial); f(Sol_best) = f(Sol_trial);
    else if (f(Sol_trial) ==  f(Sol_best));
                Calculate cardinality of trial solution, | Sol_trial |;
                Calculate cardinality of best solution, | Sol_best |;
                if (|Sol_trial | < |Sol_best|)
                        Sol ← Sol_trial ; Sol_best ← Sol_trial;
                        f(Sol) = f(Sol_trial); f(Sol_best) = f(Sol_trial);
                end if
    else if (f(Sol_trial)> RECORD - DEVIATION)
                Sol ← Sol_trial ; f(Sol) ← f(Sol_trial);
    end if
    if (f(Sol_trial)> RECORD)
            RECORD = f(Sol_trial);
    end if
end while
Return best solution;
```

Fig 1. Pseudo-code of RRT for attribute reduction

3 Modified Great Deluge Algorithm

The original great deluge algorithm was applied to attribute reduction problems by Abdullah and Jaddi [24] and achieved comparable results with other methods in the literature. Mafarja and Abdullah [31] examined in their paper the ability of improving the performance of this method by changing the increasing rate (β) intelligently. They proposed a mechanism called modified great deluge for attribute reduction (m-GD).

The main idea of m-GD here is that three equaled regions are established between the quality of the initial solution ($f(Sol)$) and the maximum dependency degree which is 1 (by using RST). Based on the interval value, which is calculated as shown in Fig. 2 (a), we define three levels as follows:

interval = *estimated_quality* – $f(Sol)$
region1 = *region2* = *region3* = *interval* /3
level1 = *level*
level2 = *level1*+ *interval*
level3 = *level2*+ *interval*

Following the example in Fig. 2, if the fitness value of the initial solution $f(Sol)$ is 0.34 and the maximum dependency degree is 1, then:

interval = 1 - 0.34 = 0.66 (as shown in Fig. 2 (a))
region1 = *region2* = *region3* = 0.66/3 = 0.22
level1 = *level* = 0.34
level2 = *level1*+ *region1* = 0.34+0.22 = 0.56
level3 = *level2*+ *region2* = 0.56+0.22 = 0.78

Each level represents the beginning of a new region in the search space, i.e., the 1st region starts from level1, the 2nd region, starts from level2 and the 3rd region starts from level3. In this method, three values for the increasing rate (β) are introduced (coded as β1, β2 and β3) to be used in updating the level in the three different regions (see Fig. 2 (b)). These values are calculated as follows:

$\beta_1 = (estimated\_quality - f(level1))/ NumOfIte\_GD$
$\beta_2 = (estimated\_quality - f(level2))/ NumOfIte\_GD$
$\beta_3 = (estimated\_quality - f(level3))/ NumOfIte\_GD$

In m-GD the level is updated depending on the region that the trial solution (Sol-trial) belongs to (i.e., if the trial solution falls in region2, then the level is updated using β2 and so on). For more details refer to the pseudo code of the algorithm as represented in Fig 3.

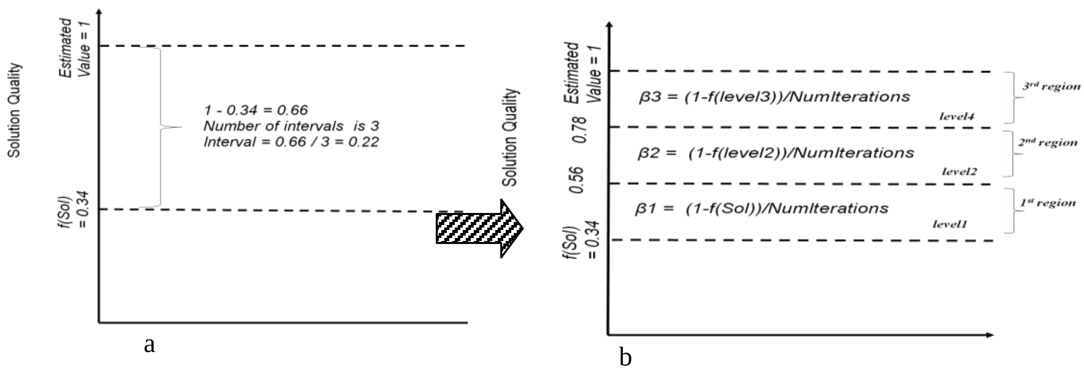

Fig 2. Search space regions in m-GD

```
Modified Great Deluge for Attribute Reduction, m-GD
Generate a random initial solution Sol;
Set Sol_best = Sol;
Set level = f(Sol_best);
Set level1 = level;
Set number of iterations, NumOfIte_GD;
Set estimated_quality =1 (as the maximum value of dependency degree in RST)
Divide the search space into 3 regions (region=( estimated_quality - f(level1))/3);
Calculate the second level, level2=level1+region1;
Calculate the third level, level3=level2+ region2;
Calculate β1 = (estimated_quality -level1))/ NumOfIte_GD;
Calculate β2 = (estimated_quality -level2)/ NumOfIte_GD;
Calculate β3 = (estimated_quality -level3)/ NumOfIte_GD;
Set iteration ← 0;
while (iteration < NumOfIte_GD)
        Generate at random a new solution Sol_trial from Sol;
        Calculate f(Sol_trial);
        if ( f(Sol_trial)> f(Sol_best)) // accepting improving solutions
              Sol ← Sol_trial ; Sol_best ← Sol_trial;
              f(Sol) = f(Sol_trial); f(Sol_best) = f(Sol_trial);
        else if (f(Sol_trial) == f(Sol_best))
                 Calculate cardinality of trial solution, | Sol_trial |;
                 Calculate cardinality of best solution, | Sol_best |;
                 if (|Sol_trial | < |Sol_best|)
                      Sol ← Sol_trial ; Sol_best ← Sol_trial;
                      f(Sol) = f(Sol_trial); f(Sol_best) = f(Sol_trial);
                 end if
        else if ( f(Sol_trial)> level)                      // accepting non-improving solutions
                 Sol ← Sol_trial ; f(Sol) ← f(Sol_trial);
        end if
        if (f(Sol_trial)>= level1 & f(Sol_trial) < level2) // updating the level according to the region
              level = level + β1;
        else if (f(Sol_trial)>= level2 & f(Sol_trial) < level3)
              level = level + β2;
        else if (f(Sol_trial)>= level3)
              level = level + β3;
        end if
        iteration++;
end while
Calculate cardinality of best solution, | Sol_best |;
Return best solution;
```

Fig 3. Pseudo code for m-GD adopted from[31]

4 Experimental Results

The proposed algorithms were programmed using J2EE Java, and the simulations were performed on an Intel Pentium 4 2.2 GHz computer with 2 GB of RAM and tested on 13 well-known UCI datasets, as shown in Table 1. For every dataset, the algorithm was run 20 times; the stopping conditions were the number of iterations that exceeded *NumOfIte* for both RRT and m-GD algorithms.

Table 1. UCI Datasets

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

The experiments in this paper are carried out to determine the classification accuracy and the number of generated rules for all data sets based on the obtained reducts from RRT and m-GD. As in Table 2, the classification accuracy assessment is performed using the Standard Voter algorithm found in the ROSETTA library [32]. Independent tests are performed with the Voting parameter set to Simple. Table 2 show the details of the classification accuracy and the number of generated rules along with some details of the obtained minimal attributes using RRT and m-GD algorithms.

Table 2 shows the comparison of using RRT and m-GD in terms of the classification accuracy (in percentage) and the number of generated rules. The results without any attribute reduction are also presented. The results for the classification accuracy show that the m-GD method is slightly better (in the case of three data sets) than the RRT method, whereas RRT is better than m-GD in the case of two data sets. RRT and m-GD tie in the case of eight data sets. Based on the results presented for the classification accuracy, it cannot be claimed that the m-GD is consistently better than RRT. However, in relation to the minimal attributes, it can be seen that even though the classification accuracies are almost similar, the number of attributes obtained by the m-GD is fewer than that obtained by the RRT method. For example, m-GD shows an increment in the classification accuracy for the Heart, Vote, and Derm2 data sets while it simultaneously demonstrates a reduction in the dimensionality. In the case of the Derm and WQ data sets, although the classification accuracy obtained by m-GD is lower than that obtained by RRT, m-GD is better in terms of the minimal attributes.

Moreover, comparison in terms of the classification accuracy between ‘without attribute reduction’ and ‘with attribute reduction’ shows that both RRT and m-GD (that are classified under ‘with attribute reduction’) are able to give higher classification accuracy compared to ‘without attribute reduction’. The use of all attributes does not guarantee 100% accuracy. This is most likely because the data sets with all attributes contain noise such as irrelevant and redundant attributes. Although the attributes have been reduced in the RRT and m-GD methods, good results are still able to be obtained for most of the data sets except Mushroom, Letters and Lung. These results show that attribute selection is important in producing a good quality of attribute for classification.

In terms of the number of generated rules, from Table 2, it can be seen that m-GD outperforms RRT on the case of five out of 13 data sets (i.e., Mushroom, Derm, Derm2, WQ and Lung) and ties in the case of six data sets (i.e., M-of-N, Exactly, Exactly2, Vote, Credit and LED). However, RRT manages to obtain a lower number of rules compared to m-GD in the case of two data sets (i.e., Heart and Letters). In general, m-GD produces a lower number of generated rules simultaneously with higher classification accuracy.

Table 2. Comparison between RRT and m-GD in terms of minimal attributes, classification accuracy and number of rules

Datasets	Without attribute reduction			*RRT*								*m-GD*							
				|#A|		Acc				|#R|		|#A|		Acc				|#R|	
	|#A|	Acc	|#R|	*Min*	*Max*	*Min*	*Max*	*Avg*	*Std*	*Min*	*Max*	*Min*	*Max*	*Min*	*Max*	*Avg*	*Std*	*Min*	*Max*
M-of-N	13	59.00	853	6	6	100.00	100.00	**100.00**	0.00	64	64	6	6	100.00	100.00	**100.00**	0.00	64	64
Exactly	13	36.00	839	6	6	100.00	100.00	**100.00**	0.00	64	64	6	6	100.00	10.000	**100.00**	0.00	64	64
Exactly2	13	35.00	855	10	10	71.00	71.00	**71.00**	0.00	606	606	10	10	71.00	71.00	**71.00**	0.00	606	606
Heart	13	31.00	263	6	7	31.00	41.00	33.00	4.10	234	256	6	7	69.00	69.00	**69.00**	0.00	244	258
Vote	16	40.00	229	8	9	53.30	70.00	66.49	6.27	135	157	8	8	70.00	70.00	**70.00**	0.00	135	135
Credit	20	69.00	896	8	9	70.00	73.00	**72.70**	0.92	725	789	8	10	66.00	73.00	71.75	2.02	725	864
Mushroom	22	100.00	7312	4	5	100.00	100.00	**100.00**	0.00	36	142	4	5	100.00	100.00	**100.00**	0.00	30	165
LED	24	7.50	1800	5	6	100.00	100.00	**100.00**	0.00	10	20	5	6	100.00	100.00	**100.00**	0.00	10	20
Letters	25	0.00	23	8	8	0.00	0.00	0.00	0.00	19	23	8	9	0.00	0.00	**0.00**	0.00	23	23
Derm	34	48.60	319	7	9	40.50	62.20	**53.00**	7.71	88	196	6	7	32.40	73.00	52.71	9.96	41	163
Derm2	34	38.90	322	9	10	41.70	52.80	47.21	3.62	260	316	8	10	41.70	61.10	**52.97**	5.73	253	305
WQ	38	61.50	470	13	15	61.50	63.50	62.30	1.01	453	467	12	14	61.50	65.40	61.99	1.25	449	469
Lung	56	100.00	29	6	7	100.00	100.00	**100.00**	0.00	21	29	4	6	100.00	100.00	**100.00**	0.00	15	29

Note: Minimum (*Min*); Maximum (*Max*); Average (*Avg*); Standard deviation (*Std*); Classification accuracy (Acc); Number of attributes (|#A|); Number of rules (|#R|)

5 Conclusion

In this paper, two single-solution-based meta-heuristic approaches for attribute reduction problems in RST, namely, RRT, and m-GD were presented. In order to address the efficiency of the proposed methods, 13 UCI benchmark data sets were used. The results showed that there was a difference between the proposed methods in terms of minimal attributes, classification accuracy and number of generated rules, where m-GD was better than RRT in some cases and m-GD produces better results in other cases. This indicates the beneficial influence of attribute reduction algorithm on the classification accuracy and the number of generated rules. As a future work, we may change the classification algorithm to study the performance of the presented attribute reduction algorithms.

References

1. D. Zhang, S. Chen and Z. H. Zhou: "Constraint Score: A new filter method for feature selection with pairwise constraints," *Pattern Recognition,* Vol. 41, 2008, pp. 1440-1451
2. Q. Hu, P. Zhu, J. Liu, Y. Yang and D. Yu: "Feature Selection via Maximizing Fuzzy Dependency," *Fundamenta Informaticae,* Vol. 98, 2010, pp. 167-181
3. H. Liu and H. Motoda: Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers Boston, 1998.
4. P. Langley, "Selection of relevant features in machine learning," In: Proceedings of the AAAI Fall Symposium on Relevance,1994, pp. 1-5.
5. Z. Pawlak: "Rough Sets," *International Journal of Information and Computer Sciences,* Vol. 11, 1982, pp. 341-356
6. Z. Pawlak: Rough sets: Theoretical aspects of reasoning about data, Springer, 1991.
7. R. W. Swiniarski and A. Skowron: "Rough set methods in feature selection and recognition," *Pattern Recog. Lett.,* Vol. 24, 2003, pp. 833-849
8. H. Liu, H. Motoda and L. Yu, "Feature selection with selective sampling," In,2002, pp. 395-402.
9. J. Liang, F. Wang, C. Dang and Y. Qian: "An efficient rough feature selection algorithm with a multi-granulation view," *International Journal of Approximate Reasoning,* Vol. 53, 2012, pp. 912–926
10. M. M. Kabir, M. Shahjahan and K. Murase: "A new hybrid ant colony optimization algorithm for feature selection," *Expert Syst. Appl.,* Vol. 39, 2012, pp. 3747-3763

11. T. Deng, C. Yang and X. Wang: "A Reduct Derived From Feature Selection," *Pattern Recog. Lett.,* Vol. 33, 2012, pp. 1638–1646
12. K. Kaneiwa and Y. Kudo: "A sequential pattern mining algorithm using rough set theory," *International Journal of Approximate Reasoning,* Vol. 52, 2011, pp. 881-893
13. M. M. Kabir, M. Shahjahan and K. Murase: "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing,* Vol. 74, 2011, pp. 2914-2928
14. J. R. Anaraki and M. Eftekhari: Improving fuzzy-rough quick reduct for feature selection. In: Electrical Engineering (ICEE), 2011 19th Iranian Conference on, pp. 1-6.
15. N. Suguna and K. Thanushkodi: "A Novel Rough Set Reduct Algorithm for Medical Domain Based on Bee Colony Optimization," *Journal of Computing,* Vol. 2, 2010, pp. 49-54
16. H. Liu, A. Abraham and Y. Li: "Nature inspired population-based heuristics for rough set reduction," *Rough Set Theory: A True Landmark in Data Analysis*, Vol. 2009, pp. 261-278
17. J. Wroblewski: Finding minimal reducts using genetic algorithms. pp. 186-189.
18. R. Jensen and Q. Shen: "Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches," *IEEE Trans. on Knowl. and Data Eng.,* Vol. 16, 2004, pp. 1457-1471
19. H. Handels, T. Roß, J. Kreusch, H. H. Wolff and S. J. Pöppl: "Feature selection for optimized skin tumor recognition using genetic algorithms " *Artif. Intell. Med.,* Vol. 16, 1999, pp. 283-297
20. X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen: "Feature selection based on rough sets and particle swarm optimization," *Pattern Recog. Lett.,* Vol. 28, 2007, pp. 459-471
21. R. Jensen and Q. Shen, "Finding Rough Set Reducts with Ant Colony Optimization," In: Proceedings of the 2003 UK Workshop on Computational Intelligence,2003, pp. 15-22.
22. L. Ke, Z. Feng and Z. Ren: "An efficient ant colony optimization approach to attribute reduction in rough set theory," *Pattern Recog. Lett.,* Vol. 29, 2008, pp. 1351-1357
23. A.-R. Hedar, J. Wang and M. Fukushima: "Tabu search for attribute reduction in rough set theory," *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* Vol. 12, 2006, pp. 909-918
24. S. Abdullah and N. S. Jaddi: "Great Deluge Algorithm for Rough Set Attribute Reduction." In: Zhang, Y., Cuzzocrea, A., Ma, J., Chung, K.-i., Arslan, T., Song, X. (eds.) Database Theory and Application, Bio-Science and Bio-Technology, vol. 118,Springer Berlin Heidelberg, 2010, pp. 189-197.
25. S. K. Jihad and S. Abdullah, "Investigating composite neighbourhood structure for attribute reduction in rough set theory," In: Intelligent Systems Design and Applications (ISDA), 10th International Conference,2010, pp. 1015-1020.

26. Y. Z. Arajy and S. Abdullah, "Hybrid variable neighbourhood search algorithm for attribute reduction in Rough Set Theory," In: Intelligent Systems Design and Applications (ISDA),2010, pp. 1015 -1020.
27. S. Abdullah, N. R. Sabar, M. Z. A. Nazri, H. Turabieh and B. McCollum, "A constructive hyper-heuristics for rough set attribute reduction," In: Intelligent Systems Design and Applications (ISDA),2010, pp. 1032-1035.
28. O. Alomari and Z. A. Othman: "Bees Algorithm for feature selection in Network Anomaly detection," *Journal of Applied Sciences Research,* Vol. 8, 2012, pp. 1748-1756
29. G. Dueck: "New Optimization Heuristics The Great Deluge Algorithm and the Record-to-Record Travel," *Journal of Computational Physics,,* Vol. 104, 1993, pp. 86-92
30. E. G. Talbi: Metaheuristics From design to implementation, Wiley Online Library, 2009.
31. M. Mafarja and S. Abdullah, "Modified great deluge for attribute reduction in rough set theory," In: Fuzzy Systems and Knowledge Discovery (FSKD),2011, pp. 1464-1469.
32. A. Øhrn: Discernibility and rough sets in medicine: tools and applications. Department of Computer and Information Science, vol. PhD, pp. 239. Norwegian University of Science and Technology, Trondheim, Norway (1999)