Fuzzy Modified Great Deluge Algorithm for Attribute Reduction

Majdi Mafarja[1], Salwani Abdullah [2]

[1]Department of Computer Science, Faculty of Information Technology
Birzeit University, Palestine
mmafarja@birzeit.edu

[2]Data Mining and Optimization Research Group (DMO),
Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia

salwani@ftsm.ukm.my

Abstract. This paper proposes a local search meta-heuristic free of parameter tuning to solve the attribute reduction problem. Attribute reduction can be defined as the process of finding minimal subset of attributes from an original set with minimum loss of information. Rough set theory has been used for attribute reduction with much success. However, the reduction method inside rough set theory is applicable only to small datasets, since finding all possible reducts is a time consuming process. This motivates many researchers to find alternative approaches to solve the attribute reduction problem. The proposed method, Fuzzy Modified Great Deluge algorithm (Fuzzy-mGD), has one generic parameter which is controlled throughout the search process by using a fuzzy logic controller. Computational experiments confirmed that the Fuzzy-mGD algorithm produces good results, with greater efficiency for attribute reduction, when compared with other meta-heuristic approaches from the literature.

Keywords: Great Deluge, Fuzzy Logic, Attribute Reduction.

1 Introduction

Attribute Reduction (AR) which is a NP-hard problem [1] can be defined as the problem of finding minimal attributes (subset) from the original set of features. It has become a necessary pre-processing step to reduce the complexity of data mining process by removing the irrelevant and/or redundant attributes. Recently, many researchers tried to implement the stochastic methods to solve attribute reduction problem such as tabu search [2, 3], ant colony optimisation (AntRSAR) [4], genetic algorithm (GenRSAR) [5, 6], simulated annealing (SimRSAR) [6], ant colony optimisation (ACOAR) [7-11], scatter search (SSAR) [12, 13]), great deluge algorithm (GD-RSAR) [14], composite neighbourhood structure (IS-CNS) [15], hybrid variable neighbourhood search algorithm (HVNS-AR) [16], and a constructive hyper-heuristics (CHH_RSAR) [17].

Great Deluge algorithm (GD) [18] is one of the more recent meta-heuristics originally developed as a variant of simulated annealing algorithm. It is a local search procedure that allows worse solutions to be accepted based on some given lower boundary or "*level*". A modified great deluge algorithm (called m-GD), proposed by Mafarja and Abdullah [19] , uses an intelligent mechanism to control the increasing rate (β) of the *"level"* instead of using the linear mechanism used in the original great deluge algorithm. In m-GD the search space is divided into three regions of equal size. The *level* is updated using different increasing rate β according to the region that the level belongs to. This paper proposed an enhancement on the former approach, where a fuzzy logic controller is used to control the value of the single parameter in the algorithm in order to achieve the best possible performance of the algorithm. This approach is called fuzzy modified great deluge for attribute reduction (Fuzzy-mGD).

The paper is structured as follows: Section 2 introduces the proposed fuzzy modified great deluge approach for attribute reduction problem. Section 3 reports the experimental results on the attribute reduction problem. This paper ends with a conclusion and a short summary of our results in Section 4.

2 GREAT DELUDE ALGORITHM (GD)

Great Deluge algorithm (GD) which was originally proposed by Dueck [18] is a generic algorithm applied to optimization problems. It is a local search procedure that allows worse solutions to be accepted based on some given lower boundary or "*level*". The general pseudo code for the great deluge algorithm is shown in Fig. 1. GD is a variant of simulated annealing algorithm (SA) with a different acceptance mechanism for accepting non-improving solution. It depends only on one parameter which is the increasing rate (β) of the water level [18].

Input: *level* **L**.
s = s$_0$; /∗ Generation of the initial solution ∗/
Choose the rain speed $\boldsymbol{\beta}$; /∗ $\boldsymbol{\beta} > 0$ ∗/
Choose the initial water level ***level***;
Repeat
    Generate a random neighbor **s**$'$;
    If $\boldsymbol{f}(s') <$ ***level*** **Then s = s′** /∗ Accept the neighbor solution ∗/
    level **=** ***level*** **−** $\boldsymbol{\beta}$; /∗ update the water level ∗/
Until Stopping criteria satisfied
Output: Best solution found

Fig. 1. A general GD algorithm pseudo code adopted from [20]

GD algorithm always accept a better solution, a worse solution is accepted if the quality of the solution is less than (for minimisation problems) or equal to some given upper boundary value which is called a "*level*". The "*level*" is initially set to be the objective function value of the initial solution, and is iteratively increased by a constant β (where β is referred as an increasing rate in this work) during its run.

3 FUZZY LOGIC CONTROLLER

Fuzzy Logic has been widely used with many real world applications since being introduced by Zadeh in 1965 [21]. For example, Jensen and Shen [22] have proposed three new techniques for fuzzy rough set feature selection based on the use of fuzzy T-transitive similarity relations. Also in scheduling and timetabling applications, fuzzy evaluation functions have been utilised in a number of different applications.

The fuzzy systems are generally consist of four components; an input fuzzifier, a knowledge base (rule base), an interfaces engine and defuzzification inference (see Fig. 2). The rules have a main role of linking the input and output variables (in `IF - THEN' form) are utilised to depict the response of the system relatively in terms of linguistic variables (words) than the mathematical formulae (see Table 1).

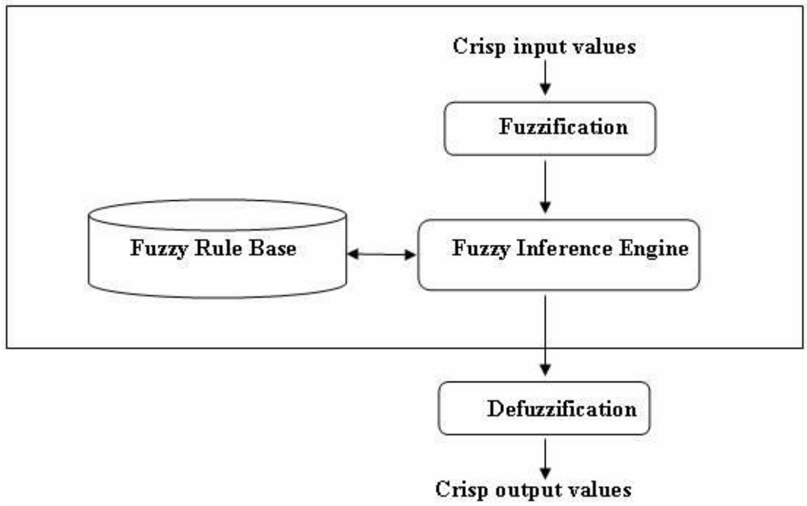

Fig. 2. Structure of a Fuzzy Logic Model

The `IF' part of the rule is mentioned as the `antecedent' and the `THEN' part is mentioned as the `consequent'. The number of inputs and outputs and as well as the desired behaviour of the system have direct impact on the number of rules. After the rules are generated, the system can be seen as a non-linear mapping from inputs to outputs. More details about simple treatment can be found in Cox [23] and complete treatment in Zimmerman [24].

4 Fuzzy Modified Great Deluge for Attribute Reduction (Fuzzy-mGD)

A fuzzy logic controller is used to control the increasing rate (β) parameter value intelligently, based on the quality of the produced solutions during the searching processes.

4.1 Solution Representation and Initial Solution Generation:

In this work, a solution is represented in one dimensional vector, where the length of the vector is based on the number of attributes of the original dataset. Each value in the vector (cell) is represented by "1" or "0". Value "1" shows that the corresponding attribute is selected; otherwise the value is set to "0". Fig. 3 shows the subset of the solution where 4 attributes are selected.

0	1	0	0	0	1	0	0	1	0	1

Fige 3. Representation of the solution.

4.2 Neighbourhood Structure

In this work, the neighbourhood of a trial solution (called Sol_{trial}) is generated by a random flip-flop where three cells are selected at random from the current solution (*Sol*). 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 added by changing "0" to "1". The cardinality of the generated trial solution must not exceed the cardinality of the best solution so far.

4.3 Quality Measure

The quality of the solutions is measured based on the dependency degree (calculated based on a rough set theory (RST) [25]. Two given solutions: best solution, Sol_{best}, and trial solution, Sol_{trial}. The trial solution Sol_{trial} is accepted if there is an improvement in the dependency degree i.e. ($f(Sol_{trial}) > f(Sol_{best})$). However if the dependency degree for both solutions is the same, the solution with the less cardinality is accepted.

4.4 The Algorithm

In this work, we consider the Fuzzy-mGD as a MISO (Multi Input Single Output) dynamical system; by sampling the Fuzzy-mGD outputs and acting on its inputs according to the fuzzy rules. By using the fuzzy controller, the *level* is updated by applying different values through the search process instead of using one increasing rate (as in the original GD), or three increasing rates (as in [19]). The search space is divided into three equalled areas; each one represents a fuzzy set (*low*, *medium* and *high*) in the fuzzy logic system as shown in Fig. 4. The controller takes two inputs, the trial solution (Sol_{trial}) and the best solution (Sol_{best}), that are connected to the general terms: *low*, *medium* and *high* (corresponding to fuzzy sets meanings). A set rules that links the input variables (Sol_{trial} and Sol_{best}) with the single output variable (β), is built according to the fuzzy rules in Table 1.

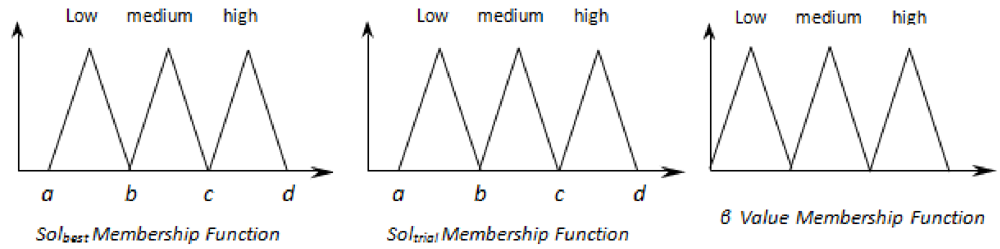

Fig 4. Graphical representation of the membership functions of Fuzzy-mGD

For example, when $f(Sol_{best})$ is "Low" and $f(Sol_{trial})$ is "Low", it means that both solutions fall in the low fuzzy set, and the *level* will be updated according to the degree of membership of an input value to the fuzzy set.

Table 1: The membership functions distribution

$f(Sol_{trial})$ \ $f(Sol_{best})$	**Low**	**Medium**	**High**
Low	Low	Low	Medium
Medium	Low	Medium	Medium
High	Medium	High	High

Listed below are the typical control rules that are used to exemplify the performance of this fuzzy system.

R_1 IF (Sol_{trial} is *low*) AND (Sol_{best} is *low*) THEN (β is *low*)

R_2 IF (Sol_{trial} is *low*) AND (Sol_{best} is *medium*) THEN (β is *low*)

R_3 IF (Sol_{trial} is *low*) AND (Sol_{best} is *high*) THEN (β is *medium*)

R_4 IF (Sol_{trial} is *medium*) AND (Sol_{best} is *low*) THEN (β is *low*)

R_5	IF (Sol_{trial} is *medium*) AND (Sol_{best} is *medium*) THEN (β is *medium*)
R_6	IF (Sol_{trial} is *medium*) AND (Sol_{best} is *high*) THEN (β is *medium*)
R_7	IF (Sol_{trial} is *high*) AND (Sol_{best} is *low*) THEN (β is *medium*)
R_8	IF (Sol_{trial} is *high*) AND (Sol_{best} is *medium*) THEN (β is *high*)
R_9	IF (Sol_{trial} is *high*) AND (Sol_{best} is *high*) THEN (β is *high*)

For each of these inputs and output, three symmetric and triangular-shaped membership functions are defined and evenly distributed on the appropriate universe of discourse. A membership function gives the degree of membership of an input value to every fuzzy set as in Fig. 2, where a is the quality of the initial solution and d equal 1 (the maximum dependency degree). The input may belong to more than one fuzzy set. Depending on the membership functions, the `fuzzifier' calculates the grade of membership of each input variable for every rule. For example, in R_2, the membership grade is calculated for the Sol_{trial} in the fuzzy set *low* and for the Sol_{best} in the *medium* fuzzy set. The result represents β value that is used as the input values for Sol_{best} and Sol_{trial}.

5 Experimental Results

This section presents the results of the experimental studies using the proposed approach. The proposed algorithm was programmed using J2EE Java and performed on an Intel Pentium 4, 2.33 GHz computer, and tested on 13 well-known UCI datasets [6]. For every dataset, the algorithm was executed for 20 times. The comparisons are carried out in terms of the minimal attributes. The purpose of this comparison is to evaluate the effectiveness of using the fuzzy logic controller (as an intelligent mechanism to control the value of the parameter in each algorithm) in obtaining the minimal attributes.The superscripts in parentheses represent the number of runs that achieved this number of attributes, while the number of attributes without superscripts means that the method could obtain that number of attributes in all of the runs.

Table 2 and Table 3 show the minimal reducts that were obtained by Fuzzy-mGD and the state-of-art approaches. The methods in comparison are as follows:

- Simulated annealing (SimRSAR) by Jensen and Shen [6]
- Tabu search (TSAR) by Hedar et al. [2]
- Great deluge algorithm (GD-RSAR) by Abdullah and Jaddi [14]
- Composite neighbourhood structure (IS-CNS) by Jihad and Abdullah [15]
- Hybrid variable neighbourhood searchalgorithm (HVNS-AR) by Arajy and Abdullah [16]
- Constructive hyper-heuristics (CHH_RSAR) by Abdullah et al. [17].

- Ant colony optimisation (AntRSAR) by Jensen and Shen [4, 6]
- Genetic algorithm (GenRSAR) by Jensen and Shen [4, 6]
- Ant colony optimisation (ACOAR) by Ke et al. [8]
- Scatter search (SSAR) by Jue et al. [26]

Table 2: Results of the experiments compared with those in literature 1.

Datasets	Fuzzy-mGD	GD-RSAR	TSAR	SimRSAR	AntRSAR	ACOAR
M-of-N	6	$6^{(10)}\ 7^{(10)}$	6	6	6	6
Exactly	6	$6^{(7)}\ 7^{(10)}8^{(3)}$	6	6	6	6
Exactly2	10	$10^{(14)}11^{(6)}$	10	10	10	10
Heart	$6^{(9)}\ 7^{(11)}$	$9^{(4)}10^{(16)}$	6	$6^{(29)}\ 7^{(1)}$	$6^{(18)}\ 7^{(2)}$	6
Vote	8	$9^{(17)}10^{(3)}$	8	$8^{(15)}\ 9^{(15)}$	8	8
Credit	$8^{(18)}\ 9^{(2)}$	$11^{(11)}12^{(9)}$	$8^{(13)}\ 9^{(5)}\ 10^{(2)}$	$8^{(18)}\ 9^{(1)}\ 11^{(1)}$	$8^{(12)}\ 9^{(4)}\ 10^{(4)}$	$8^{(16)}9^{(4)}$
Mushroom	4	$4^{(8)}\ 5^{(9)}6^{(3)}$	$4^{(17)}\ 5^{(3)}$	4	4	4
LED	5	$8^{(14)}9^{(6)}$	5	5	$5^{(12)}\ 6^{(4)}\ 7^{(3)}$	5
Letters	8	$8^{(7)}9^{(13)}$	$8^{(17)}\ 9^{(3)}$	8	8	8
Derm	$6^{(19)}\ 8^{(1)}$	$12^{(14)}13^{(6)}$	$6^{(14)}\ 7^{(6)}$	$6^{(12)}\ 7^{(8)}$	$6^{(17)}\ 7^{(3)}$	6
Derm2	$8^{(7)}\ 9^{(13)}$	$11^{(14)}12^{(6)}$	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(3)}\ 9^{(7)}$	$8^{(3)}\ 9^{(17)}$	$8^{(4)}9^{(16)}$
WQ	$12^{(5)}\ 13^{(14)}\ 14^{(1)}$	$15^{(14)}16^{(6)}$	$12^{(1)}\ 13^{(13)}\ 14^{(6)}$	$13^{(16)}\ 14^{(4)}$	$12^{(2)}\ 13^{(7)}\ 14^{(11)}$	$12^{(4)}13^{(12)}14^{(4)}$
Lung	$4^{(15)}\ 5^{(5)}$	$4^{(5)}\ 5^{(2)}\ 6^{(13)}$	$4^{(6)}\ 5^{(13)}\ 6^{(1)}$	$4^{(7)}\ 5^{(12)}\ 6^{(1)}$	4	4

Table 3: Results of the experiments compared with those in literature 2.

Datasets	Fuzzy-mGD	IS-CNS	HVNS-AR	GenRSAR	CHH_RSAR	SSAR
M-of-N	6	6	6	$6^{(6)}7^{(12)}$	$6^{(11)}7^{(9)}$	6
Exactly	6	6	6	$6^{(10)}7^{(10)}$	$6^{(13)}7^{(7)}$	6
Exactly2	10	10	10	$10^{(9)}11^{(11)}$	10	10
Heart	$6^{(9)}\ 7^{(11)}$	6	6	$6^{(18)}7^{(2)}$	6	6
Vote	8	8	8	$8^{(2)}9^{(18)}$	8	8
Credit	$8^{(18)}\ 9^{(2)}$	$8^{(10)}9^{(9)}\ 10^{(1)}$	$8^{(7)}9^{(6)}\ 10^{(7)}$	$10^{(6)}11^{(14)}$	$8^{(10)}9^{(7)}\ 10^{(3)}$	$8^{(9)}\ 9^{(8)}\ 10^{(3)}$
Mushroom	4	4	4	$5^{(1)}6^{(5)}7^{(14)}$	4	$4^{(12)}\ 5^{(8)}$
LED	5	5	5	$6^{(1)}7^{(3)}8^{(16)}$	5	5
Letters	8	8	8	$8^{(8)}9^{(12)}$	8	$8^{(5)}\ 9^{(15)}$
Derm	$6^{(19)}\ 8^{(1)}$	$6^{(18)}\ 7^{(2)}$	$6^{(16)}\ 7^{(4)}$	$10^{(6)}11^{(14)}$	6	6
Derm2	$8^{(7)}\ 9^{(13)}$	$8^{(4)}9^{(16)}$	$8^{(5)}9^{(12)}10^{(3)}$	$10^{(4)}11^{(16)}$	$8^{(5)}9^{(5)}10^{(10)}$	$8^{(2)}\ 9^{(18)}$
WQ	$12^{(5)}\ 13^{(14)}\ 14^{(1)}$	$12^{(2)}13^{(8)}14^{(10)}$	$12^{(3)}13^{(6)}14^{(8)}\ 15^{(3)}$	16	$12^{(13)}14^{(7)}$	$13^{(4)}\ 14^{(16)}$
Lung	$4^{(15)}\ 5^{(5)}$	$4^{(17)}\ 5^{(3)}$	$4^{(16)}\ 5^{(4)}$	$6^{(8)}7^{(12)}$	$4^{(10)}\ 5^{(7)}\ 6^{(3)}$	4

Based on the results presented in Table 2 and Table 3, it can be seen that Fuzzy-mGD is comparable with the other approaches since it performs better than most of them. It is better than AntRSAR on five datasets, and better than SSAR on six datasets (ties on five datasets). Our approach is able to produce better results in all datasets when compared with GenRSAR method. Fuzzy-mGD too, has obtained better results than SimRSAR in six datasets and TSAR in eight datasets. The proposed Fuzzy-mGD is able to obtain better results on all datasets when compared with the GD-RSAR. It can produce better results than IS-CNS, HVNS-AR, CHH_RSAR in 6, 5, and 7 instances, respectively. Fuzzy-mGD is able to obtain two results better than ACOAR. In general, we can summarise that our approach is better than most of the approaches introduced. Fuzzy-mGD demonstrates highly promising performance when compared with other available methods. We believe that the strength of the method comes from the improvement of the new modification on the GD algorithm that embeds the fuzzy logic controller to control the parameter β which further enhanced the performance of the proposed approach through a better exploitation during the search process.

6 Conclusions

The work described in this paper proposed a fuzzy modified great deluge algorithm, called Fuzzy-mGD, to solve the attribute reduction problem in the rough set theory. Great Deluge algorithm has only one generic parameter which is controlled throughout the search process using a fuzzy logic controller by taking into account the quality of the produced solutions. Several benchmark UCI datasets are used to evaluate the utilisation efficiency of the proposed method. The experimental results showed that our approach provides qualified solutions to the well-known benchmark datasets from the attribute reduction literature. Employing a fuzzy logic controller positively influences the performance of the original algorithm by producing a lower number of minimal attributes. As a result, we can say that controlling the parameter values affects the behaviour of the Fuzzy-mGD method in searching for the most informative attributes and that the selected subset of attributes is a better representation of the original data.

References

1. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R. (ed.) Intelligent decision support– Handbook of applications and advances of the rough set theory, pp. 311–362. Kluwer Academic Publishers, Poland (1992)
2. Hedar, A.-R., Wang, J., Fukushima, M.: Tabu search for attribute reduction in rough set theory. Soft Comput. 12, 909-918 (2008)

3. Wang, J., Guo, K., Wang, S.: Rough set and Tabu search based feature selection for credit scoring. Procedia Computer Science 1, 2425-2432 (2010)
4. Jensen, R., Shen, Q.: Finding Rough Set Reducts with Ant Colony Optimization. Proceedings of the 2003 UK Workshop on Computational Intelligence, pp. 15-22 (2003)
5. BingXiang, L., Feng, L., Xiang, C.: An adaptive genetic algorithm based on rough set attribute reduction. In: Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, pp. 2880-2883.
6. Jensen, R., Shen, Q.: Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches. IEEE Trans. on Knowl. and Data Eng. 16, 1457-1471 (2004)
7. Wu, J., Qiu, T., Wang, L., Huang, H.: An Approach to Feature Selection Based on Ant Colony Optimization and Rough Set. In: Chen, R. (ed.) Intelligent Computing and Information Science, vol. 134, pp. 466-471. Springer Berlin Heidelberg (2011)
8. Ke, L., Feng, Z., Ren, Z.: An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recog. Lett. 29, 1351-1357 (2008)
9. Chen, Y., Miao, D., Wang, R.: A rough set approach to feature selection based on ant colony optimization. Pattern Recog. Lett. 31, 226-233 (2010)
10. Ming, H.: Feature Selection Based on Ant Colony Optimization and Rough Set Theory. Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on, vol. 1, pp. 247-250 (2008)
11. Wang, G., Wang, S.-J., Shi, L., Huang, D., Chen, H., Liu, Y., Peng, X.: Study of adaptive parameter control for ant colony optimization applied to feature selection problem. Advanced Science Letters. in press, (2012)
12. Jue, W., Hedar, A.R., Shouyang, W.: Scatter Search for Rough Set Attribute Reduction. In: Bio-Inspired Computing: Theories and Applications, 2007. BIC-TA 2007. Second International Conference on, pp. 236-240. (2007)
13. Wang, J., Hedar, A.-R., Wang, S., Ma, J.: Rough set and scatter search metaheuristic based feature selection for credit scoring. Expert Systems with Applications 39, 6123-6128 (2012)
14. Abdullah, S., Jaddi, N.S.: 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, pp. 189-197. Springer Berlin Heidelberg (2010)
15. Jihad, S.K., Abdullah, S.: Investigating composite neighbourhood structure for attribute reduction in rough set theory. Intelligent Systems Design and Applications (ISDA), 10th International Conference, pp. 1015-1020 (2010)
16. Arajy, Y.Z., Abdullah, S.: Hybrid variable neighbourhood search algorithm for attribute reduction in Rough Set Theory. Intelligent Systems Design and Applications (ISDA), pp. 1015 -1020 (2010)

17. Abdullah, S., Sabar, N.R., Nazri, M.Z.A., Turabieh, H., McCollum, B.: A constructive hyper-heuristics for rough set attribute reduction. Intelligent Systems Design and Applications (ISDA), pp. 1032-1035 (2010)
18. Dueck, G.: New Optimization Heuristics The Great Deluge Algorithm and the Record-to-Record Travel. Journal of Computational Physics, 104, 86-92 (1993)
19. Mafarja, M., Abdullah, S.: Modified great deluge for attribute reduction in rough set theory. Fuzzy Systems and Knowledge Discovery (FSKD), vol. 3, pp. 1464-1469. IEEE (2011)
20. Talbi, E.G.: Metaheuristics From design to implementation. Wiley Online Library (2009)
21. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338-353 (1965)
22. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. Fuzzy Systems, IEEE Transactions on 17, 824-838 (2009)
23. Cox, E.: The fuzzy systems handbook: a practitioner's guide to building, using, and maintaining fuzzy systems. Academic Press Professional, Inc. (1994)
24. Zimmermann, H.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Boston (1996)
25. Pawlak, Z.: Rough Sets. International Journal of Information and Computer Sciences 11, 341-356 (1982)
26. Wang, J., Hedar, A.-R., Zheng, G., Wang, S.: Scatter Search for Rough Set Attribute Reduction. pp. 531-535. IEEE (2009)