A fuzzy record-to-record travel algorithm for solving rough set attribute reduction

Article *in* International Journal of Systems Science · April 2013

DOI: 10.1080/00207721.2013.791000

2 authors:

Majdi Mafarja
Birzeit University

Salwani Abdullah
National University of Malaysia

International Journal of Systems Science, 2015
Vol. 46, No. 3, 503–512, http://dx.doi.org/10.1080/00207721.2013.791000

Taylor & Francis
Taylor & Francis Group

A fuzzy record-to-record travel algorithm for solving rough set attribute reduction

Majdi Mafarja[a,b] and Salwani Abdullah[a,*]

[a]*Data Mining and Optimization Research Group, Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Bangi Selangor, Malaysia;* [b]*Department of Computer Science, Faculty of Information Technology, Birzeit University, Birzeit, Palestine*

(*Received 15 March 2012; final version received 17 March 2013*)

Attribute reduction can be defined as the process of determining a minimal subset of attributes from an original set of attributes. This paper proposes a new attribute reduction method that is based on a record-to-record travel algorithm for solving rough set attribute reduction problems. This algorithm has a solitary parameter called the *DEVIATION*, which plays a pivotal role in controlling the acceptance of the worse solutions, after it becomes pre-tuned. In this paper, we focus on a fuzzy-based record-to-record travel algorithm for attribute reduction (FuzzyRRTAR). This algorithm employs an intelligent fuzzy logic controller mechanism to control the value of *DEVIATION*, which is dynamically changed throughout the search process. The proposed method was tested on standard benchmark data sets. The results show that FuzzyRRTAR is efficient in solving attribute reduction problems when compared with other meta-heuristic approaches.

Keywords: rough set theory; attribute reduction; fuzzy logic; record-to-record travel algorithm

Introduction

Attribute Reduction (AR) is an NP-hard problem (Skowron and Rauszer 1992) that can be defined as the problem of determining a minimum reduct (subset) from the original set of attributes. Attribute reduction problems address high dimensional data sets that contain a large number of relevant or irrelevant attributes. The optimal subset is determined by both relevancy and redundancy aspects. An attribute is said to be relevant if a decision attribute depends on it; otherwise, it is termed irrelevant. However, an attribute can be considered to be redundant if it is highly correlated with other attributes. Rough set theory, proposed by Pawlak (1982, 1991), has been used as a simple mechanism to determine a minimum reduct by locating all the possible reducts and selecting the one with the lowest cardinality and the highest dependency. The reduction of attributes can be achieved by comparing the dependency degrees of the generated subsets so that the reduced set has the same dependency degree as the original set. This process is a time-consuming procedure, and it is only effective for small data sets (Jensen and Shen 2004). Hence, for high-dimensional data sets, a large amount of research has used meta-heuristics to determine better solutions for the attribute reduction problem instead of using the reduction method from rough set theory.

We can find many meta-heuristic approaches in the literature that have been applied to solve attribute reduction problems. For example, Jensen and Shen (2003, 2004) have studied meta-heuristic approaches for solving attribute reduction problems. In their work, they presented three methods, the genetic algorithm (GenRSAR), the ant colony-based method (AntRSAR) and the simulated annealing algorithm (SimRSAR). Whereas Hedar, Wang, and Fukushima (2006) considered a memory-based heuristic of a tabu search to solve the attribute reduction problem in rough set theory, Ke, Feng, and Ren (2008) proposed an ant colony-based approach, and a scatter search approach was introduced by Wang, Hedar, Zheng, and Wang (2009). The great deluge algorithm for attribute reduction was presented by Abdullah and Jaddi (2010); later, Jihad and Abdullah (2010) proposed the composite neighbourhood structure, and Arajy and Abdullah (2010) presented a hybrid variable neighbourhood search algorithm for the same problem. For the first time, a constructive hyper-heuristic for solving attribute reduction problems was employed by Abdullah, Sabar, Nazri, Turabieh, and McCollum (2010). Further reading about attribute reduction problems can be found in John, Kohavi, and Pfleger (1994), Koller and Sahami (1996), Kohavi and John (1997), Dong, Zhong, and Ohsuga (1999), Bazan, Nguyen, Nguyen, Synak, and Wróblewski (2000), Jensen and Shen (2008).

In this paper, we propose a new attribute reduction mechanism that investigates how the RRT algorithm can be applied to determine (near) optimal feature subsets or rough set reducts. The contribution of this paper is the combination of the record-to-record travel (RRT) algorithm with a fuzzy logic controller (FLC) to control the solitary parameter in RRT called the *DEVIATION* for accepting a worse solution. A fuzzy logic is embedded here due to its success on

*Corresponding author. Email: salwani@ftsm.ukm.my

other hybrid approaches applied on various domains, such as in Lin and Lee (2010), Economou, Knowles, Tsourdos, and White (2011), and Cheng, Su, and Tsai (2012). The value of the *DEVIATION* is adaptively changed throughout the search process, on the basis of the quality of the trial and the best solutions. Furthermore, in contrast to the available attribute reduction methods that only report the numbers of generated attributes, we also evaluate the quality of the generated subsets of attributes in terms of the number of generated rules (descriptive patterns) and the classification accuracy.

The remainder of this paper is organised as follows: next section briefly discusses the record-to-record travel algorithm, followed by a detailed implementation of the FuzzyRRTAR. Next, the simulation of the proposed algorithm and the discussion of the experimental results are presented. Finally, concluding remarks on the effectiveness of the proposed technique and the potential future research aspects are discussed.

Record-to-record travel (RRT) algorithm

The RRT algorithm was originally proposed by Dueck (1993). It is a variant of the simulated annealing algorithm with a different mechanism for accepting non-improving solutions. The application of the RRT algorithm was done on the problem of balancing hydraulic turbine runners by Sinclair (1993), team orienteering problem (Chao, Golden, and Wasil 1996), fleet vehicle routing problem (Li, Golden, and Wasil 2007), handwritten classification problems (Radtke, Sabourin, and Wong 2008). Recently, Kharrat, Dhouib, and Chabchoub (2010) used RRT to solve lexicographic goal programming models, and Dhouib, Aissa, and Chabchoub (2010) employed RRT to minimise the manufacturing batch dispersion in order to optimise traceability in food industry. These papers show that the RRT algorithm is able to produce good solutions. To our knowledge, RRT has not been tested on the problem discussed in this paper.

This algorithm has a solitary parameter called the *DEVIATION*, which plays a pivotal role in controlling the acceptance of the worse solutions, after it becomes pretuned. The significance of this method relates to the ease of its implementation and the required number of parameters, which influences the performance (Dueck 1993). The algorithm starts from a randomly-generated initial solution; then, it improves the initial solution by searching its neighbourhood for better solutions on the basis of their evaluation (the dependency degree, in this work). If it is better than the best value so far (the *RECORD*) or slightly worse than the *DEVIATION*, then the solution is accepted. Note that, the initial value of the *RECORD* is equal to the initial objective function. During the search process, the *RECORD* value is updated with an objective value of the best solution so far. More formally, in the case of maximisation, if (*Sol*) is the best solution so far and (Sol_{trial}) is the newly generated solution, then (Sol_{trial}) is accepted as the next solution if the condition mentioned below is satisfied:

$$f(Sol_{best}) - f(Sol_{trial}) < DEVIATION, \qquad (7)$$

where $DEVIATION >= 0$ is the maximum allowed deviation that determines how many worse values than the current record are accepted. This process is repeated till the stopping condition is met (in this work, the number of iterations).

For the RRT algorithm, it is necessary to choose the suitable value of its single parameter (*DEVIATION*). If the value of *DEVIATION* is set to be high, the algorithm provides good results with an extended computational time, whereas a small value of *DEVIATION* makes the algorithm produce poor results in a short span of time (Dueck 1993). To help the algorithm better exploit the search space during the searching process, an FLC is used to intelligently adjust the value of *DEVIATION*.

Fuzzy record-to-record travel algorithm

Here, we propose a new method, called the fuzzy record-to-record travel algorithm for attribute reduction (FuzzyRRTAR).

Fuzzy logic controller

Decision-making requires many factors to be concurrently addressed. The key for generating a better decision lies in the priority order of these factor(s). Apart from this consideration, swapping between the conflicting factors must also be conducted to yield the expected result.

The fuzzy reasoning framework handles uncertainty as follows: a fuzzy set A of a universe of discourse X (the initial range of the input variables) is characterised by a membership function $\mu_A : X \rightarrow [0, 1]$, where each element x in X is associated with a value $\mu_A(x)$ that represents a *grade of membership* of x in A.

The fuzzy systems are generally deployed for representing vague, tentative and undependable knowledge. These systems consist of four components: an input fuzzifier, a knowledge base (rule base), an interface engine and defuzzification inference (see Figure 1). The rules have the

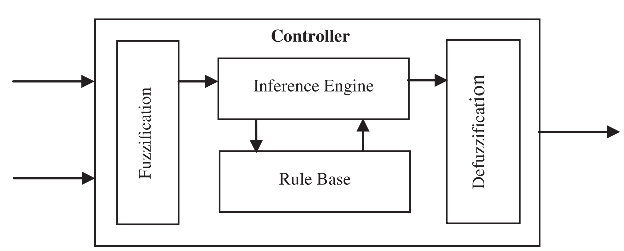

Figure 1. Fuzzy logic controller.

main role of linking the input and output variables, and 'IF -THEN' forms are utilised to depict the response of the system relatively in terms of *linguistic* variables (words) rather than mathematical formulae (see Table 1).

The 'IF' part of the rule is considered to be the 'antecedent', and the 'THEN' part is considered to be the 'consequent'. The number of inputs and outputs, as well as the desired behaviour of the system, has a direct impact on the number of rules. After the rules are generated, the system can be seen as a non-linear mapping from the inputs to the outputs. More details of this simple treatment can be found in Cox (1994), and a complete treatment can be found in Zimmerman (1996).

A system's fuzzy inference process is demonstrated depending on the Sol_{trial} and Sol_{best} input variables, which are connected to the following general terms: *low*, *medium* and *high* (corresponding to fuzzy set meanings). These membership functions are randomly selected to cross the universe of discourse of the variable. A rule set linking the input variables (Sol_{trial} and Sol_{best}) with the single output variable, *DEVIATION*, is built.

Typical control rules are used to exemplify the performance of this fuzzy system.

Rule 1: IF (Sol_{best} is *low*) AND (Sol_{trial} is *medium*) THEN (*DEVIATION* is *medium*)
Rule 2: IF (Sol_{best} is *medium*) AND (Sol_{trial} is *medium*) THEN (*DEVIATION* is *medium*)
Rule 3: IF (Sol_{best} is *medium*) AND (Sol_{trial} is *high*) THEN (*DEVIATION* is *low*)

The normalisation of the input values within the range [0, 1] is performed in the first stage. This conversion is accomplished as follows:

$$v' = \frac{(v \rightarrow \min A)}{(\max A \rightarrow \min A)}, \tag{8}$$

where v is the value of the input variable in the initial range [min$A$, max$A$] and *minA* represents the lowest possible value for the variable A; in this work, the outcome will be different according to the dependency degree of the initial solution. For example, if $v = 0.55$ in [0.35, 1], the normalised value v' is 0.3 in the new range [0, 1].

For every rule, the function of the fuzzy system, the 'fuzzifier', calculates the grade of membership in each input variable, depending on the membership functions. For example, in *Rule 1*, the membership grade is calculated for the Sol_{best} in the fuzzy set *low* and for the Sol_{trial} in the *medium* fuzzy set. Later, with these fuzzified values, the inference engine calculates the antecedents of *Rule 1* by employing the suitable fuzzy operators consequent to the *AND* or *OR* connectives. Later, the implication operator is applied by the inference engine to the rule, to obtain the fuzzy to be mounted over the output variable. Here, the inference is implemented by shortening the output membership function at a level equivalent to the calculated level of truth in the rule's antecedent. In the end, all of the truncated output membership functions are added up to constitute a solitary fuzzy subset by traversing the maximum over all of the subsequent sets. Then, a defuzzification step is executed, to translate the final fuzzy output into a crisp value. A common form of this process is called centre of area (COA) because it depends on the concept of identifying the area under a scaled membership function, as follows:

$$\int_{\min A}^{\max A} \mu(x_i) \cdot x_i \Bigg/ \int_{\min A}^{\max A} \mu(x_i). \tag{9}$$

The result from this equation represents the value of the *DEVIATION* parameter for the provided input values of Sol_{best} and Sol_{trial}.

Table 1. Fuzzy rule set for FuzzyRRTAR.

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

Fuzzy record-to-record model

In RRT, the deviation (*DEVIATION*) parameter plays a vital role, to assure the competence of the search process and to steer it towards a proper region in the search space. The FLC is employed to manage the value of *DEVIATION* intelligently, depending on the quality of the produced solution, to enhance the performance of the proposed algorithm and to boost its diversification.

According to Jang (1992), and Kropp and Baitinger (1993), defining suitable membership functions and control rules is a crucial and extremely demanding task for designing fuzzy systems. In this work, for all of the inputs and outputs, three symmetric and triangle-shaped membership functions are defined and equally dispersed on the proper space of discourse. The degree of membership of the input value in the fuzzy set is provided by the membership function as shown in Figure 2.

The fuzzified input variables are input to the rule base. Table 1 shows the fuzzy rules that are employed in this research.

The most commonly used defuzzification method, which is the centre of gravity (COG), was used in our system (Jensen and Shen 2008).

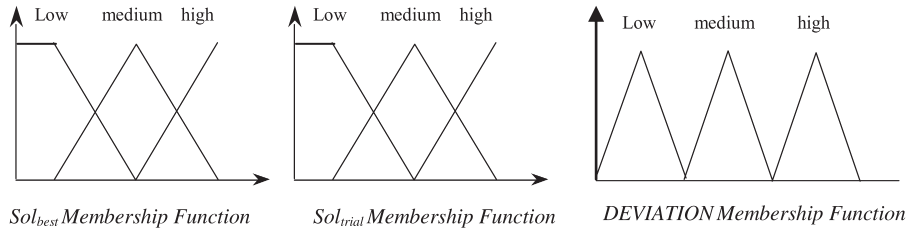

Figure 2. Membership functions of FuzzyRRTAR.

The algorithm

The pseudo code of the FuzzyRRTAR, which was applied in this work, is presented in Figure 3.

The algorithm begins with a randomly generated initial solution. Initially, the best solution, Sol_{best}, is set as *Sol*, and the *Record* is set as the fitness value of the best solution. In the *while* loop, a trial solution is generated randomly (Sol_{trial}). According to the rules in Table 4, the best and the trial solutions are considered to be inputs to the FLC in order to compute the *DEVIATION* value. If the Sol_{trial} is better than the best solution in hand (Sol_{best}), it will be accepted. Then, the Sol_{best} will be updated to Sol_{trial}, and the record will be $f(Sol_{trial})$. If the Sol_{trial} and the Sol_{best} qualities are equal, then the number of attributes for both the solutions would be checked by the algorithm. If the number of attributes in Sol_{best} is greater than that of Sol_{trial}, then Sol_{trial} will be accepted, and Sol_{best} and the *RECORD* will be updated. Nevertheless, the objective function of this trial solution will be compared with the best solution as follows: ($\Delta = f(Sol_{best}) - f(Sol_{trial})$). If Δ is smaller than the deviation parameter *DEVIATION*, then it will be accepted. In this case, the FCS will update the *DEVIATION* value on the basis of the given inputs in every iteration. The algorithm stops when it reaches the termination criterion (the number of generations is set as the termination criterion in this work).

Experimental results

The proposed algorithms were programmed using Java, and simulations were performed on a 2.2 GHz computer with 2 GB of RAM and were tested on 13 well-known UCI data sets, as shown in Table 2. In this work, we ran each algorithm for 20 runs; the stopping conditions were the number of iterations that exceeded *NumOfIte*. On the basis of some preliminary results, we found that RRTAR provides good results (in terms of the number of selected attribute) when using $DEVIATION = 0.09$. In FuzzyRRTAR, FLC will set the value of *DEVIATION* depending on the objective values of Sol_{trial} and Sol_{best} that were obtained in each of the iterations.

The classification accuracy and the number of rules are determined on the basis of the obtained reducts. Classification was performed using the Standard Voter algorithm found in the ROSETTA library (Øhrn 1999). The independent tests were performed with the Voting parameter set to Simple. All the other parameters were set to default values.

Table 2. UCI data sets.

Data sets	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

Comparison between RRTAR and FuzzyRRTAR

We are interested in comparing the proposed approaches, to check the effectiveness of using the FLC on the number of minimal reduct, classification accuracy and the number of generated rules. This section reveals the comparative analysis that was performed on the two approaches by identifying the effectiveness of each approach in terms of the minimal reducts, classification accuracy and number of rules. Table 3 shows the results that were obtained by RRTAR and FuzzyRRTAR. The superscripts in parentheses represent the number of runs that achieved the minimal reducts, while the number of attributes without superscripts means that the method could obtain only that number of attributes in all of the runs. Note that, the classification is performed on the reduced data sets, which are obtained using the RRTAR and the FuzzyRRTAR dimensionality reduction techniques.

On the basis of the results in Table 3, in terms of the minimal reducts, the FuzzyRRTAR outperformed RRTAR on 8 data sets out of 13, and they obtained the same results on 5 data sets, and no worst result was obtained by FuzzyRRTAR. The results show that using the FLC

Fuzzy Record-to-Record Travel Algorithm (FuzzyRRTAR)
Generate a random initial solution *Sol*
Set $Sol_{best} = Sol$
Set $RECORD = f(Sol_{best})$
while (stopping-criterion is not satisfied)
Generate at random a new solution Sol_{trial} *in the neighbour of Sol*
Calculate the quality of Sol_{trial}, $f(Sol_{trial})$
$DEVIATION = fuzzy\ (f\ (Sol_{trial}),\ f\ (Sol_{best}))$ // employing fuzzy to find the *DEVIATION* value
if ($f(Sol_{trial}) > f(Sol_{best})$)
$Sol \leftarrow Sol_{trial}$; $Sol_{best} \leftarrow Sol_{trial}$
$f(Sol) = f(Sol_{trial})$; $f(Sol_{best}) = f(Sol_{trial})$
else if ($f(Sol_{trial}) == f(Sol_{best})$)
Calculate cardinality of trial solution, $\| Sol_{trial} \|$;
Calculate cardinality of best solution, $\| Sol_{best} \|$;
if ($\|Sol_{trial}\| < \|Solbest\|$)
$Sol \leftarrow Sol_{trial}$; $Sol_{best} \leftarrow Sol_{trial}$
$f(Sol) = f(Sol_{trial})$; $f(Sol_{best}) = f(Sol_{trial})$
end if
else if ($f(Sol_{trial}) > RECORD - DEVIATION$)
$Sol \leftarrow Sol_{trial}$; $f(Sol) \leftarrow f(Sol_{trial})$
end if
if ($f(Sol_{trial}) > RECORD$)
$RECORD = f(Sol_{trial})$;
end if
end while
Return best solution found

Figure 3. Pseudo code for FuzzyRRTAR.

to control the value of the single parameter in RRT helps the algorithm to better exploit the search space, to obtain the minimal results and to improve the performance of the algorithm. This improvement may result from selecting different features for each subset, which affects the classification accuracy and the number of generated rules. The results of FuzzyRRTAR clearly demonstrate that the intelligent control of the RRT algorithm supports the performance of the algorithm in determining the combinations of the minimal reducts.

Table 3. Results obtained from RRTAR and FuzzyRRTAR.

Data sets	RRTAR			FuzzyRRTAR		
	Reducts	Accuracy (%)	No. of rules	Reducts	Accuracy (%)	No. of rules
M-of-N	6	100	64	6	100	64
Exactly	6	100	64	6	100	64
Exactly2	10	57	606	10	57	606
Heart	$6^{(9)}\ 7^{(11)}$	10	1037	6	0	261
Vote	$8^{(13)}\ 9^{(7)}$	63	1050	$8^{(16)}\ 9^{(4)}$	63	586
Credit	$8^{(18)}\ 9^{(2)}$	10	1755	$8^{(18)}\ 9^{(2)}$	8	1756
Mushroom	$4^{(6)}\ 5^{(14)}$	100	1244	4	100	601
LED	$5^{(18)}\ 6^{(2)}$	100	10	5	100	10
Letters	8	0	459	8	0	413
Derm	$7^{(1)}\ 8^{(16)}\ 9^{(3)}$	19	6411	6	19	5984
Derm2	$9^{(2)}\ 10^{(18)}$	78	5992	$8^{(5)}\ 9^{(15)}$	80	4994
WQ	$13^{(2)}\ 14^{(13)}\ 15^{(5)}$	21	9236	13	27	8301
Lung	$6^{(14)}\ 7^{(6)}$	100	491	$4^{(9)}\ 5^{(11)}$	100	491

The classification results show that the FuzzyRRTAR approach performs well and shows improvement in the classification accuracy for some of the data sets in which there was a corresponding decrease in dimensionality (e.g. WQ, Derm2), for example, on the WQ data set, FuzzyRRTAR shows an increase of up to 23% in classification accuracy while simultaneously demonstrating a reduction in dimensionality. Sometimes, there may be an insignificant decrease in the classification accuracy for some data sets (e.g. Credit) where RRTAR and FuzzyRRTAR are able to obtain the same number of attributes. On the other hand, FuzzyRRTAR may sometimes determine subsets of similar size to those determined by RRTAR and demonstrates an increase in classification accuracy. This may be due to the different selected attributes.

The results in Table 3 show that RRTAR demonstrates a small increase in the classification accuracy on 2 out of 13 data sets (i.e. Credit and Heart) when compared with FuzzyRRTAR. However, FuzzyRRTAR significantly outperforms RRTAR in some cases.

In terms of the number of generated rules, from Table 6, FuzzyRRTAR outperforms RRTAR significantly in some data sets, i.e. in the Heart, Vote, Mushroom, Derm2 and WQ data sets, for which 261, 586, 601, 4994 and 8301 rules were generated, respectively. A total of 1037, 1050, 1244, 5992 and 9236 rules were generated when using RRTAR on these five data sets. In some data sets, the difference in the number of generated rules is notable but not significant (i.e. Letters and Derm). In these cases, the number of generated rules is decreased from 7% to 11%. In other cases, RRTAR obtains the same number of rules.

Comparison with state-of-the-art methods

We compared our approaches with the other available approaches in the literature to assess the behaviour of our proposed approaches among different rough set attribute reduction and heuristic optimisation methods. For a comparative study, we used the following methods:

- Tabu Search (TSAR) by Hedar, Wang, and Fukushima (2008)
- Ant Colony Optimisation (AntRSAR) by Jensen and Shen (2003)
- Genetic Algorithm (GenRSAR) by Jensen and Shen (2003)
- Simulated Annealing (SimRSAR) by Jensen and Shen (2004)
- Ant Colony Optimisation (ACOAR) by Ke et al. (2008)
- Scatter Search (SSAR) by Wang et al. (2009)
- Great Deluge Algorithm (GD-RSAR) by Abdullah and Jaddi (2010)
- Composite Neighbourhood Structure for Attribute Reduction (IS-CNS) by Jihad and Abdullah (2010)
- Hybrid variable neighbourhood search algorithm (HVNS-AR) by Arajy and Abdullah (2010)
- Constructive Hyper-Heuristics (CHH_RSAR) by Abdullah et al. (2010)

Number of minimal attributes (reducts)

Note that since FuzzyRRTAR is able to outperform RRTAR for most of the data sets, we are interested in comparing FuzzyRRTAR with the other approaches that are available in the literature. The results of our approach and state-of-the art methods are reported in Tables 4 and 5. The entries in these tables represent the number of attributes in the minimal reducts that are obtained by each method. The results demonstrate that it is better than AntRSAR on six data sets and better than SSAR on five data sets (it ties on six data sets). Our approach produced better results in all the data sets when compared with the GenRSAR method. However, FuzzyRRTAR obtained better results than SimRSAR on seven data sets and better than TSAR on eight data sets. FuzzyRRTAR can produce better results than IS-CNS, HVNS-AR and CHH_RSAR in three, five and six instances, respectively. FuzzyRRTAR obtains two results better than ACOAR. In general, we can conclude that our approach is comparable with the previous methods on many data sets and that it can sometimes outperform some of them.

Classification accuracy and number of rules

The data presented in Table 6 showthe percentage of classification accuracy and the number of generated rules. These results were compared with results that were obtained from the methods that exist in the ROSETTA system: a genetic algorithm, Johnson's algorithm and Holt's 1R algorithm.

From the results in Table 6, we can observe that FuzzyRRTAR can produce 100% classification accuracy for five data sets (i.e. M-of-N, Exactly, Mushroom, LED and lung). The accuracy that is produced when using FuzzyRRTAR is between 55% and 80% for three of the data sets, and four of the data sets have accuracy between 8% and 27% (excluding 0% on Heart and Letters data sets). FuzzyRRTAR is able to obtain the best result in one data set (i.e. Derm2 data set) and it is comparable with the other approaches in the other data sets. For nearly the same size data sets (M-of-N and Exactly), it can obtain the same results as the other approaches.

Holt's algorithm shows good results when compared with the other two methods in the literature. In comparison with this approach, we found that it had outperformed our approach in some data sets, but at the same time, our approach was able to outperform it in other data sets. In the Exactly2 data set, FuzzyRRTAR obtained an accuracy of 57% while Holt's obtained 71%; in the Heart and Credit data sets, our approach obtained 0% and 8% while Holt's obtained 69% for both. On the other hand, FuzzyRRTAR in

Table 4. Comparison with the literature-1.

Data sets	FuzzyRRTAR	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	$9^{(4)}\ 10^{(16)}$	6	$6^{(29)}\ 7^{(1)}$	$6^{(18)}\ 7^{(2)}$	6
Vote	$8^{(16)}\ 9^{(4)}$	$9^{(17)}\ 10^{(3)}$	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	$12^{(14)}\ 13^{(6)}$	$6^{(14)}\ 7^{(6)}$	$6^{(12)}\ 7^{(8)}$	$6^{(17)}\ 7^{(3)}$	6
Derm2	$8^{(5)}\ 9^{(15)}$	$11^{(14)}\ 12^{(6)}$	$8^{(2)}\ 9^{(14)}\ 10^{(4)}$	$8^{(3)}\ 9^{(7)}$	$8^{(3)}\ 9^{(17)}$	$8^{(4)}\ 9^{(16)}$
WQ	13	$15^{(14)}\ 16^{(6)}$	$12^{(1)}\ 13^{(13)}\ 14^{(6)}$	$13^{(16)}\ 14^{(4)}$	$12^{(2)}\ 13^{(7)}\ 14^{(11)}$	$12^{(4)}\ 13^{(12)}\ 14^{(4)}$
Lung	$4^{(9)}\ 5^{(11)}$	$4^{(5)}\ 5^{(2)}\ 6^{(13)}$	$4^{(6)}\ 5^{(13)}\ 6^{(1)}$	$4^{(7)}\ 5^{(12)}\ 6^{(1)}$	4	4

Table 5. Comparison with the literature-2.

Datasets	FuzzyRRTAR	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	6	6	$6^{(18)}\ 7^{(2)}$	6	6
Vote	$8^{(16)}\ 9^{(4)}$	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	$6^{(18)}\ 7^{(2)}$	$6^{(16)}\ 7^{(4)}$	$10^{(6)}\ 11^{(14)}$	6	6
Derm2	$8^{(5)}\ 9^{(15)}$	$8^{(4)}\ 9^{(16)}$	$8^{(5)}\ 9^{(12)}\ 10^{(3)}$	$10^{(4)}\ 11^{(16)}$	$8^{(5)}\ 9^{(5)}\ 10^{(10)}$	$8^{(2)}\ 9^{(18)}$
WQ	13	$12^{(2)}\ 13^{(8)}\ 14^{(10)}$	$12^{(3)}\ 13^{(6)}\ 14^{(8)}\ 15^{(3)}$	16	$12^{(13)}\ 14^{(7)}$	$13^{(4)}\ 14^{(16)}$
Lung	$4^{(9)}\ 5^{(11)}$	$4^{(17)}\ 5^{(3)}$	$4^{(16)}\ 5^{(4)}$	$6^{(8)}\ 7^{(12)}$	$4^{(10)}\ 5^{(7)}\ 6^{(3)}$	4

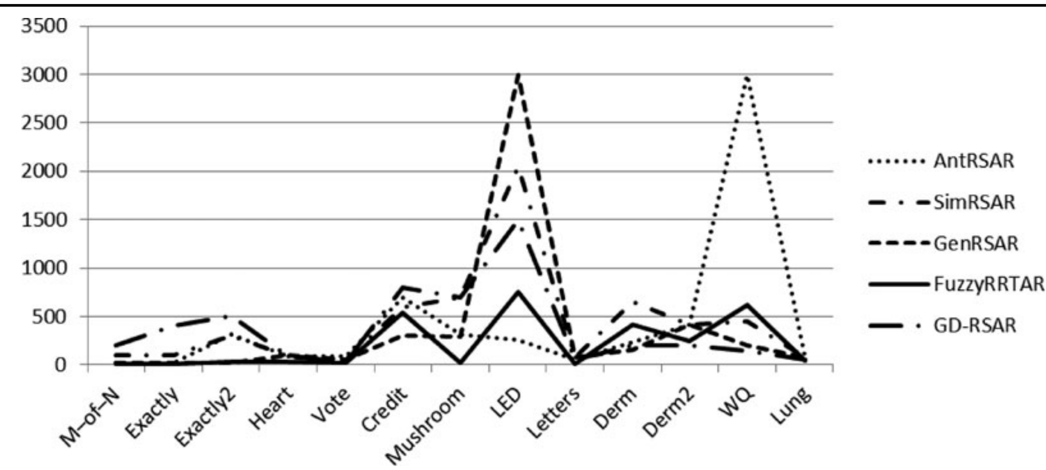

Figure 4. Comparison of running time.

the LED and Derm2 data sets obtained 100% and 80% accuracy at the same time that Holt's obtained 76% and 61%, respectively. Overall, FuzzyRRTAR outperforms Holt's algorithm in five data sets, and Holt's algorithm outperforms FuzzyRRTAR in six data sets. FuzzyRRTAR is comparable with the GA and Johnson algorithms in most of the data sets.

Clearly, the above results show how the attributes that are selected by each method affect the performance of the classification accuracy and the number of generated rules,

Table 6. Comparison with the literature in terms of classification accuracy and the number of rules.

Data sets	FuzzyRRTAR		GA		Johnson		Holt's	
	No. of rules	Accuracy (%)	No. of rules	Accuracy (%)	No. of rules	Accuracy (%)	No. of rules	Accuracy (%)
M-of-N	64	100	64	100	64	100	26	63
Exactly	64	100	64	100	64	100	26	66
Exactly2	606	57	606	57	606	57	26	71
Heart	261	0	7485	17	261	0	66	69
Vote	586	63	494	63	134	63	48	87
Credit	1756	8	151,133	21	887	2	83	69
Mushroom	601	100	1	100	1	100	119	100
LED	10	100	1788	100	10	100	27	76
Letters	413	0	4278	0	23	0	27	0
Derm	5984	19	57,336	38	320	0	189	57
Derm2	4994	80	63,254	75	310	14	135	61
WQ	8301	27	99,903	25	465	2	132	61
Lung	491	100	3986	100	26	100	158	100

because each subset may contain different attributes. We can see that, if the selected attributes from the FuzzyRRTAR method are used, it causes a different accuracy percentage and a different number of generated rules, compared with those that are obtained using the attributes that are selected from other approaches. This scenario improves the efficiency of FuzzyRRTAR in selecting the most related attributes that enhance the performance of the learning algorithm.

Running time

Figure 4 represents the running time of AntRSAR (Jensen and Shen 2003), SimRSAR (Jensen and Shen 2004), GenRSAR (2003), GD-RSAR (Abdullah and Jaddi 2010) and FuzzyRRTAR algorithms. This graph shows that the running time for FuzzyRRTAR is consistently faster than the other methods on most of the tested data sets except SimRSAR and GD-RSAR where they took less running time on one and three data sets, respectively. It is clear that the performance of FuzzyRRTAR methods is improved by fine-tuning the parameters to each individual data set by employing the FLC.

Statistical test

We employed a statistical test to analyse and compare the performance of the designed meta-heuristic algorithms for the attribute reduction problem. The statistical tests are usually performed to estimate the confidence of the results in order to determine the scientific validity of the results (Talbi 2009). The selection of the statistical test depends on the characteristics of the data; under normality conditions, the paired t-test is the most widely used test. In a case in which the data are not normally distributed, a non-parametric test may be used, such as a Wilcoxon test (Good 2000). Table 7 summarises the results of applying the Wilcoxon rank sum test to compare FuzzyRRTAR with the other approaches in the literature with a confidence level of 95%.

The results in Table 7 demonstrate that there is a significant difference between our approach and the other approaches in the literature (denoted in bold). FuzzyRRTAR is significantly different when compared with GDRSAR and GenRSAR. In the rest of the comparisons, FuzzyRRTAR is significantly different between two (with ACOAR) to six (with SSAR) data sets.

From the presented results, we can conclude that the FuzzyRRTAR can be considered as one of the appropriate methodologies to be employed for the attribute reduction problem.

Conclusions and future work

In this paper, two attribute reduction methods, based on the record-to-record travel (RRT) algorithm, are proposed. To the best of our knowledge, this paper presents the first such algorithm that focuses on this problem domain. The first method was the basic RRT algorithm. The FLC was embedded in RRT to control the single parameter that controls the acceptance of the worst solutions. The performance of the proposed algorithm was tested on standard benchmark data sets and comparison results were presented. Preliminary results showed that FuzzyRRTAR outperforms RRT in most cases, which improves the effect of using an intelligent mechanism to control the parameter in the proposed approach.

The experimental results showed that our approach could produce three of the best-known results in the literature and were comparable with other approaches for the remainder of the data sets. We used ROSETTA to determine the classification accuracy and the number of generated rules, using the features that were selected using our

Table 7. *p*-value using Wilcoxon rank sum test.

Data set	Fuzzy RRTAR-GDRSAR	Fuzzy RRTAR-TSAR	Fuzzy RRTAR-SimRSAR	Fuzzy RRTAR-AntRSAR	Fuzzy RRTAR-ACOAR	Fuzzy RRTAR-ISCNS	Fuzzy RRTAR-HVNSAR	Fuzzy RRTAR-GenRSAR	Fuzzy RRTAR-CHHRSAR	Fuzzy RRTAR-SSAR
M-of–N	**.002**	1.000	1.000	1.000	1.000	1.000	1.000	**.000**	**.003**	1.000
Exactly	**.001**	1.000	1.000	1.000	1.000	1.000	1.000	**.002**	**.008**	1.000
Exactly2	**.014**	1.000	1.000	1.000	1.000	1.000	1.000	**.001**	1.000	1.000
Heart	**.000**	1.000	1.000	.157	1.000	1.000	1.000	.157	1.000	1.000
Vote	**.000**	.046	**.014**	**.046**	**.046**	**.046**	**.046**	**.000**	**.046**	**.046**
Credit	**.000**	.008	.317	.008	.157	**.003**	**.001**	**.000**	**.002**	**.001**
Mushroom	**.001**	.083	1.000	1.000	1.000	1.000	1.000	**.000**	1.000	**.005**
LED	**.000**	1.000	1.000	.015	1.000	1.000	1.000	**.000**	1.000	1.000
Letters	**.000**	.083	1.000	1.000	1.000	1.000	1.000	**.001**	1.000	**.000**
Derm	**.000**	**.014**	**.005**	.083	1.000	.157	**.046**	**.000**	1.000	1.000
Derm2	**.000**	**.008**	.083	.157	.317	.317	.083	**.000**	**.002**	.083
WQ	**.000**	.059	**.046**	**.013**	1.000	**.021**	**.022**	**.000**	.180	**.000**
Lung	**.000**	**.046**	.083	**.001**	**.001**	**.005**	**.008**	**.000**	.317	**.001**

approach. In terms of the classification accuracy, it can be said that our approach is comparable with the other methods in ROSETTA, and it can outperform some methods in terms of the number of generated rules as well.

Notes on contributors

Majdi Mafarja received his B.Sc. in software engineering and M.Sc. in computer information systems from Philadelphia University and The Arab Academy for Banking and Financial Sciences, Jordan, in 2005 and 2007, respectively. He did his Ph.D. in computer science at the National University of Malaysia (UKM) in 2012. Now, he works as an assistant professor at the Department of Computer Science at Birzeit University. His research interests include evolutionary computation, meta-heuristics and data mining.

Salwani Abdullah obtained her B.Sc. in computer science from Universiti Teknologi Malaysia and her master's degree specialising in computer science from Universiti Kebangsaan Malaysia (UKM). She did her Ph.D. in computer science at University of Nottingham, United Kingdom. Now, she is an associate professor and chairperson of the School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. Her research interest falls under artificial intelligence and operation research, particularly in meta-heuristic algorithms in the optimisation area that involves different real-world applications and optimisation problems, such as timetabling, scheduling, space allocation and data mining tasks.

References

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*, eds. Y. Zhang, A. Cuzzocrea, J. Ma, K.-I. Chung, T. Arslan, and X. Song, Berlin: Springer , pp. 189–197.

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.

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)*, pp. 1015–1020.

Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., and Wróblewski, J. (2000), 'Rough Set Algorithms in Classification Problem', *Rough Set Methods and Applications*, Hiedelberg: Physica Verlag, pp. 49–88.

Chao, I., Golden, B.L., and Wasil, E.A. (1996), 'A Fast and Effective Heuristic for the Orienteering Problem', *European Journal of Operational Research*, 88, 475–489.

Cheng, M.B., Su, W.C., and Tsai, C.C. (2012), 'Robust Tracking Control of a Unicycle-Type Wheeled Mobile Manipulator Using a Hybrid Sliding Mode Fuzzy Neural Network', *International Journal of Systems Science*, 43(3), 408–425.

Cox, E. (1994), *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*, Waltham, MA: Academic Press Professional, Inc.

Dhouib, S., Ben Aissa, M., and Chabchoub, H. (2010), 'A Record to Record Travel Metaheuristic to Minimize Total Dispersion in Food Industry', in *IEEE International Conference on Computer Systems and Applications (AICCSA), 2010 IEEE/ACS*, pp. 1–4.

Dong, J., Zhong, N., and Ohsuga, S. (1999), 'Using Rough Sets with Heuristics for Feature Selection', in *Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, pp. 178–187.

Dueck, G. (1993), 'New Optimization Heuristics. The Great Deluge Algorithm and the Record-to-Record Travel', *Journal of Computational Physics*, 104, 86–92.

Economou, J.T., Knowles, K., Tsourdos, A., and White, B.A. (2011), 'Fuzzy-Hybrid Land Vehicle Driveline Modelling Based on a Moving Window Subtractive Clustering Approach', *International Journal of Systems Science*, 42(2), 303–317.

Good, P.I. (2000), *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, New York: Springer.

Hedar, A.-R., Wang, J., and Fukushima, M. (2006), 'Tabu Search for Attribute Reduction in Rough Set Theory', *Soft Computing*, 12, 909–918.

Hedar, A.-R., Wang, J., and Fukushima, M. (2008), 'Tabu Search for Attribute Reduction in Rough Set Theory', *Soft Computing*, 12, 909–918.

Jang, J.S.R. (1992), 'Self-Learning Fuzzy Controllers Based on Temporal Back Propagation', *IEEE Transactions on Neural Networks*, 3, 714–723.

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.

Jensen, R., and Shen, Q. (2004), 'Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches', *IEEE Transactions on Knowledge and Data Engineering*, 16, 1457–1471.

Jensen, R., and Shen, Q. (2008), *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*, New York: Wiley-IEEE Press.

Jihad, S.K., and Abdullah, S. (2010), 'Investigating Composite Neighbourhood Structure for Attribute Reduction in Rough Set Theory', in *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 1015–1020.

John, G.H., Kohavi, R., and Pfleger, K. (1994), 'Irrelevant Features and the Subset Selection Problem', in *Machine Learning: Proceedings of the Eleventh International Conference,* ed. W.W.C.H. Hirsh, San Francisco, CA: Morgan Kaufmann, pp. 121–129.

Ke, L., Feng, Z., and Ren, Z. (2008), 'An Efficient Ant Colony Optimization Approach to Attribute Reduction in Rough Set Theory', *Pattern Recognition Letters*, 29, 1351–1357.

Kharrat, A., Dhouib, S., and Chabchoub, H. (2010), 'Adaptive Record-to-Record Travel Method to Solve Lexicographic Goal Programming Models', *International Journal of Information and Decision Sciences*, 2, 147–169.

Kohavi, R., and John, G.H. (1997), 'Wrappers for Feature Subset Selection', *Artificial Intelligence*, 97, 273–324.

Koller, D., and Sahami, M. (1996), 'Toward Optimal Feature Selection', in *International Conference on Machine Learning*, pp. 284–292.

Kropp, K., and Baitinger, U. (1993), 'Optimization of Fuzzy Logic Controller Inference Rules Using a Genetic Algorithm', in *EUFIT 93 First European Congress on Fuzzy and Intelligent Technologies, Aachen,* Vol. 2, pp. 1090–1096.

Li, F., Golden, W., and Wasil, E. (2007), 'A Record-to-Record Travel Algorithm for Solving the Heterogeneous Fleet Vehicle Routing Problem', *Computers & Operations Research*, 34(9), 2734–2742

Lin, C.J., and Lee, C.Y. (2010), 'Non-Linear System Control Using a Recurrent Fuzzy Neural Network Based on Improved Particle Swarm Optimisation', *International Journal of Systems Science*, 41(4), 381–395

øhrn, A. (1999), 'Discernibility and Rough Sets in Medicine: Tools and Applications', Ph.D. dissertation, Norwegian University of Science and Technology, Dept. of Computer and Information Science.

Pawlak, Z. (1982), 'Rough Sets', *International Journal of Information and Computer Sciences*, 11, 341–356.

Pawlak, Z. (1991), *Rough Sets: Theoretical Aspects of Reasoning About Data*, New York: Springer.

Radtke, P.V.W., Sabourin, P., and Wong, T. (2008), 'Using the RRT Algorithm to Optimize Classification Systems for Handwritten Digits and Letters', in *Proceedings of the 2008 ACM Symposium on Applied Computing: ACM*, pp. 1748–1752.

Sinclair, M. (1993), 'Comparison of the Performance of Modern Heuristics for Combinatorial Optimization on Real Data', *Computers & Operations Research*, 20, 687–695.

Skowron, A., and Rauszer, C. (1992), 'The Discernibility Matrices and Functions in Information Systems', in *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Set Theory*, ed. R. Slowinski, Poland: Kluwer Academic Publishers, pp. 311–362.

Talbi, E.G. (2009), *Metaheuristics: From Design to Implementation*, Wiley Online Library.

Wang, J., Hedar, A.-R., Zheng, G., and Wang, S. (2009), 'Scatter Search for Rough Set Attribute Reduction', in *Bio-Inspired Computing: Theories and Applications, 2007 (BIC-TA 2007), Second International Conference*, pp. 531–535.

Zimmermann, H. (1996), *Fuzzy Set Theory and Its Applications*, Boston: Kluwer Academic Publishers.