ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/260184685

Investigating memetic algorithm in solving rough set attribute reduction

Article *in* International Journal of Computer Applications in Technology · January 2013

DOI: 10.1504/IJCAT.2013.056915

CITATIONS
2

READS
8

2 authors:

Majdi Mafarja
Birzeit University
6 PUBLICATIONS **20** CITATIONS

SEE PROFILE

Salwani Abdullah
National University of Malaysia
88 PUBLICATIONS **1,178** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Using Meta-heuristics algorithms to solve systems of nonlinear equations View project

All content following this page was uploaded by Majdi Mafarja on 11 September 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Investigating memetic algorithm in solving rough set attribute reduction

Majdi Mafarja

Data Mining and Optimisation Research Group (DMO),
Center for Artificial Intelligence Technology,
Universiti Kebangsaan Malaysia,
43600 UKM, Bangi Selangor, Malaysia
and
Department of Computer Science,
Faculty of Information Technology,
Birzeit University,
P.O. Box 14, Birzeit, Palestine
Email: majdi@ftsm.ukm.my

Salwani Abdullah*

Data Mining and Optimisation Research Group (DMO),
Center for Artificial Intelligence Technology,
Universiti Kebangsaan Malaysia,
43600 UKM, Bangi Selangor, Malaysia
Email: salwani@ftsm.ukm.my
*Corresponding author

Abstract: Attribute reduction is the problem of selecting a minimal subset from the original set of attributes. Rough set theory has been used for attribute reduction with much success. Since it is well known that finding a minimal subset is a NP-hard problem; therefore, it is necessary to develop efficient algorithms to solve this problem. In this work, we propose a memetic algorithm-based approach inside the rough set theory which is a hybridisation of genetic algorithm and simulated annealing. The proposed method has been tested on UCI data sets. Experimental results demonstrate the effectiveness of this memetic approach when compared with previous available methods. Possible extensions upon this simple approach are also discussed.

Keywords: rough set theory; attribute reduction; memetic algorithm; genetic algorithm; simulated annealing.

Reference to this paper should be made as follows: Mafarja, M. and Abdullah, S. (2013) 'Investigating memetic algorithm in solving rough set attribute reduction', *Int. J. Computer Applications in Technology*, Vol. 48, No. 3, pp.195–202.

Biographical notes: Majdi Mafarja is currently an Assistant Professor at the Department of Computer Science at Birzeit University. He received his BSc in Software Engineering and MSc 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 PhD in Computer Science at the Universiti Kebangsaan Malaysia (UKM). His research interests include evolutionary computation, meta-heuristics and data mining.

Salwani Abdullah is an Associate Professor and Chairperson of School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. She obtained her BSc in Computer Science from Universiti Teknologi Malaysia and her Master's degree in Computer Science from Universiti Kebangsaan Malaysia (UKM). She did her PhD in Computer Science at University of Nottingham, UK. Her research interest falls under artificial intelligence and operation research, particularly in meta-heuristic algorithms in optimisation areas that involve different real-world applications and optimisation problems, such as university timetabling, job shop scheduling, nurse rostering, space allocation and data mining tasks.

Copyright © 2013 Inderscience Enterprises Ltd.

1 Introduction

Attribute reduction, which is a NP-hard problem (Ke et al., 2008), aims to determine a minimal subset of attributes from a problem domain with a high accuracy in representing the original attributes. The optimal subset is determined by both relevancy and redundancy aspects. An attribute is said to be relevant if a decision is depending on it, otherwise it is irrelevant. An attribute can be considered to be redundant if it is highly correlated with other attributes. Removing redundant and misleading attributes can improve the performance and efficiency of a learning process.

Rough set theory (Pawlak, 1982; Pawlak, 1991) has been used as a reduction method with much success by purely structural methods. The simplest way to find a minimal reduct is to locate all possible reducts and select the one with the lowest cardinality. This is, obviously, a time-consuming procedure and it is only practical for small data sets. For the high-dimensionality data sets, many research efforts have concentrated on meta-heuristics to find the optimal reduct instead of using the reduction method in rough set theory. For example, Jensen and Shen (2003, 2004) studied three meta-heuristic approaches: the genetic algorithm (GenRSAR), the ant colony-based method (AntRSAR) and the simulated annealing algorithm (SimRSAR). Hedar et al. (2008) considered a memory-based heuristic of a tabu search to solve the attribute reduction problem in the rough set theory; ant colony-based approaches are proposed in the work of Chen et al. (2010), Ke et al. (2008), Ming (2008), Wang et al. (2012b) and Wu et al. (2011). The fire fly algorithm (Banati and Bajaj, 2011), artificial bee colony (Hu et al., 2012), bee colony optimisation (Suguna and Thanushkodi, 2010), scatter search (SSAR) (Wang et al., 2007a, Wang et al., 2012a) and Particle Swarm Optimisation (PSO) (Bae et al., 2010; Bello et al., 2007; Fan and Zhong, 2012; Wang et al., 2007b) are also proposed. 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 to solve attribute reduction problems was employed by Abdullah et al. (2010). Further reading about attribute reduction problems can be found in the work of Bazan et al. (2000), Dong et al. (1999), Jensen and Shen (2008), John et al. (1994), Kohavi and John (1997) and Koller and Sahami (1996). Wu et al. (2012) explored some feature selection methods like gain ratio method, the correlation-based feature selection method and the decision tree-based method.

In this work, we proposed an attribute reduction mechanism, which investigates how the memetic algorithm can be applied to find optimal (or near optimal) feature subsets or rough set reducts. Our approach based on the hybridisation of Genetic Algorithm (GA) with the Simulated Annealing algorithm (SA), in order to have a benefit of the exploration and exploitation mechanism offered by a population-based (i.e. GA) and single-based (i.e. SA) approaches, respectively (Kogilavani and Balasubramanie, 2011). The effectiveness of using the memetic approach is evaluated and tested on 13 UCI (University California Irvine Machine Learning Repository) data sets (Jensen and Shen (2003, 2004).

2 Rough set theory

Rough set theory is a mathematical approach to solve vagueness, imprecision and uncertainty problems (Pawlak, 1982; Pawlak, 1991). The rough set itself is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximation since the vague concepts cannot be categorised in terms of information about their objects. The lower approximation is a description of all objects which surely belong to the subset of interest, whereas the upper approximation is a description of all objects which may belong to the subset. To illustrate the operation of rough set for attribute reduction, we will use an example data set as shown in Table 1. Data are often presented in Table 1 which consists of four conditional attributes (a, b, c, d) and one decision attribute (e), rows represents objects and entries of the table are called attribute values. The task of attribute reduction here is to find the minimal subset from the conditional attributes while retaining a consistent data set with respect to the decision attribute.

Table 1 An example data set

$x \in U$	a	b	c	d	$\Rightarrow e$
0	1	0	2	2	0
1	0	1	1	1	2
2	2	0	0	1	1
3	1	1	0	2	2
4	1	0	2	0	1
5	2	2	0	1	1
6	2	1	1	1	2
7	0	1	1	0	1

The starting point of rough set theory is the concept of indiscernibility. Suppose that two finite, non-empty sets U and A are given, where U is the universe and A is a set of attributes with every attribute $a \in A$, we associate a set Va of its values, called the domain of a.

The pair $I = (U, A)$ is an information system. Any subset B of A determines a binary relation $I(B)$ on U, which is called an indiscernibility relation, and is defined as follows:

$$xI(B)y \text{ if and only if } a(x) = a(y) \textit{ for every } a \in B \quad (1)$$

where $a(x)$ denotes the value of attribute a for element x.

The partition of U generated by $IND(B)$ will be denoted by $U/IND(B)$, or simple U/B and can be calculated as follows:

$$U/B = \{a \in B : U/IND(\{a\})\}, \quad (2)$$

where

$$A \otimes G = \{X \cap Y : \forall X \in A, \forall Y \in G, X \cap Y \neq \varnothing\} \quad (3)$$

If $(x, y) \in I(B)$ then we will say that x and y are B-indiscernible.

To depict above definitions, let us refer to the example in Table 1. If $B = \{b, c\}$, then objects 1, 6 and 7 are indiscernible as objects 0 and 4. $I(B)$ creates the following partition of U:

$$\begin{aligned} U/B &= U/I(b) \otimes U/I(c) \\ &= \{\{0,2,4\},\{1,3,6,7\},\{5\}\} \\ &\otimes \{\{2,3,5\},\{1,6,7\},\{0,4\}\} \\ &= \{\{2\},\{0,4\},\{3\},\{1,6,7\},\{5\}\} \end{aligned}$$

The indiscernibility relation will be used next to define approximations; the basic concepts of rough set theory.

Let $X \subseteq U$, approximations; $B(X)$ and $\overline{B}(X)$ called the B-lower and the B-upper approximation of X, respectively, can be defined as follows:

$$\underline{B}(X) = \{x \in U : B(x) \subseteq X\} \quad (4)$$

$$\overline{B}(X) = \{x \in U : B(x) \cap X\} \quad (5)$$

Let D and C be subsets of A, then the positive, negative and boundary regions can be defined as follows:

The positive region of the partition U/D with respect to C contains all objects of U that can be uniquely classified to blocks of the partition U/D using the knowledge in attributes C.

$$POS_B(D) = \bigcup_{X \in U/I(D)} B_*(X) \quad (6)$$

For example, let $B = \{b, c\}$ and $D = \{e\}$, then

$$POS_B(\mathrm{D}) = \bigcup \{2,5,\}\{3\} = \{2,3.5\}$$

We can say that when considering attributes b and c, objects 1, 3 and 5 can certainly be classified as belonging to a class in attribute e.

An often applied measure is dependency degree between attributes (Jensen and Shen, 2004). Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C. If there exists a functional dependency between values of D and C, then D depends totally on C. Dependency can be defined in as follows:

For D, $C \subset A$, it is said that C depends on D in a degree of k $(0 \leq k \leq 1)$ denoted $C \Rightarrow k\ D$, if

$$k = \gamma_D(C) \frac{|POS_D(C)|}{|U|} \quad (7)$$

where $|F|$ denotes the cardinality of set F.

If $k = 1$, we say that D depends totally on C, if $k < 1$, we say that D depends partially on C, and if $k = 0$, we say that D does not depend on C. In the example data set shown in Table 1, let $C = \{b,c\}$ and $D = \{e\}$, then the degree of dependency is

$$\begin{aligned} \gamma_{\{b,c\}}(\{e\}) &= \frac{|POS_{\{b,c\}}(\{e\})|}{|U|} \\ &= \frac{|\{2,3,5\}|}{|\{0,1,2,3,4,5,6,7\}|} = \frac{3}{8} \end{aligned}$$

2.1 Reduction method

One of the major applications of rough set theory is to find the minimal reducts by eliminating the redundant attributes from original sets, without any information loss (Pawlak, 1982; Pawlak, 1991). 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 of the original set (Jensen and Shen, 2004). A reduct is formally defined as a subset R of minimal cardinality of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$ where D is a decision system

$$\gamma_R(D) = \gamma_C(D) \quad (8)$$

$$R_{\min} = \{X : X \in R, \forall Y \in R, |X| \leq |Y|\} \quad (9)$$

The intersection of all reduced subsets is called the core which contains all those attributes that cannot be removed from the data set without introducing more contradictions.

$$Core(R) = \bigcap_{X \in R} X$$

Using the example shown in Table 1, the minimal reduct sets of C are:

$$R = \{\{a,b\},\{b,c\},\{a,c\},\{a\},\{b\},\{c\}\}$$

The dependency degree of $D = \{d\}$ on all possible reducts of C can be calculated as follows:

$$\gamma_C(D) = 1,$$

$$\gamma_{\{a,b\}}(D) = \frac{1}{6}, \gamma_{\{b,c\}}(D) = 1, \gamma_{\{a,c\}}(D) = 1,$$

$$\gamma_{\{a\}}(D) = 0, \gamma_{\{b\}}(D) = \frac{1}{6}, \gamma_{\{c\}}(D) = \frac{2}{3},$$

From these sets, the minimal reduct is

$$R_{\min} = \{\{b,c\},\{b,c\}\}$$

If the minimal reduct $\{a, c\}$ is selected, then the example data set presented in Table 1 can be reduced as shown in Table 2. On the other hand, Table 3 shows a reduced data set corresponding to the reduct $\{b, c\}$.

It is obvious that finding all possible reducts is a time-consuming process, and moreover it is applicable only with small data sets. It is meaningless to calculate all reducts

aiming to find only one minimal. To improve the performance of the above method, an alternative strategy is required for large data sets.

Table 2 A reduced data set

U	*a*	*c*	*d*
1	0	0	1
2	0	1	0
3	0	2	0
4	1	0	1
5	1	1	1
6	1	2	0

Table 3 A reduced data set corresponding {b, c}

U	*b*	*c*	*d*
1	0	0	1
2	0	1	0
3	0	2	0
4	0	0	1
5	1	1	1
6	0	2	0

3 Memetic algorithm for attribute reduction (MA-RSAR)

In this section, we present our memetic approach called (MA-RSAR). The term Memetic Algorithm (MA) was originally proposed by Moscato (1989), it is an approach for heuristic search and optimisation problems that combine the population-based global search (e.g. genetic algorithm) and the single-based search (e.g. Simulated annealing). MAs have shown that they are more sophisticated versions of the traditional genetic algorithms for some problem domains (Hart et al., 2005).

3.1 Genetic algorithm

Genetic Algorithms (GAs) have been developed by Holand in the 1970s to achieve the goal of understanding the adaptive process of the natural systems (Holland, 1992). The traditional GA uses a population of solutions in solving a given problem; each solution is represented by a chromosome with a length of m where m is the number of attributes in the data set (Yang and Honavar, 1998). Usually a binary representation is used to represent the solution. In the binary representation, the bit '1' implies a selected attribute while the bit '0' implies an excluded attribute. GA has been applied in many fields in machine learning (Kogilavani and Balasubramanie, 2011).

In the general template of GA, firstly the initial population is randomly generated, where a random scheme is operated to decide the number of 1-bits in each chromosome and the places where those 1-bits will be located inside the chromosome, after the initialisation step; the quality of the chromosome will be calculated by a fitness function. Then two parents will be selected to apply the GAs operators: crossover and mutation. The selection process can be either randomly or using a mechanism (e.g. Roulette Wheel Selection (RWS), tournament selection) (Michalewicz, 1996). The crossover operator will be applied on the selected parents in order to allow the search to look in diverse directions for attractive solutions to be combined in a single child to generate a new population. Then mutation operator will be applied to alter one or more components of the new child. This process (selection, crossover, and mutation) continues until the stopping criterion is satisfied.

One of the main properties of GA is that it tends to locate the local optimum in a region of convergence which may cost high computational time for the big size data sets and which may sometimes not find the optimum with sufficient precision due to its inherent nature (Zhu et al., 2010). To solve this problem, the combination (hybridisation) between GAs and some local search operations (Moscato, 1989) (e.g. SA) is proposed because of the capability of fine-tuning and improving the solutions generated by GA to make them more accurate and efficient.

3.2 The local search: simulated annealing algorithm

Simulated annealing, proposed by Kirkpatrick et al. (1983) is a hill climbing-based method that uses probability to escape from local optima. The process starts by evaluating the randomly generated neighbour (move) of the best solution. The improving move with respect to the objective function is always be accepted, whilst worse solution is accepted with a certain probability determined by the Boltzmann probability, $P = e^{-\theta/T}$ where θ is the difference of the evaluation of the objective function between the best solution (Sol_{best}) and the trial solution (Sol_{trial}); T is a parameter (called the temperature) which periodically decreases during the search process according to some cooling schedule. In this work, the initial temperature is set to $2*|C|$, where $|C|$ represents the number of attributes for each data set, and the cooling schedule is calculated as $T = 0.93 * T$ [as adopted in the work of Jensen and Shen (2003)].

3.3 Solution quality measure

Dependency degree, which is calculated based on rough set theory, is used to measure the quality of the solutions, it is denoted as δ. Two given solutions are: 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. ($\delta(Sol_{trial}) > \delta(Sol_{best})$). However, if the dependency degree for both solutions is same, the solution with the less cardinality is accepted.

3.4 The algorithm

The pseudo code of the memetic algorithm (MA-RSAR) applied in this work is presented in Figure 1.

Figure 1 Pseudo code for MA-RSAR

```
Set number of generation, NumOfGen;
Set population size popSize,
for i = 1 to popSize Do
    Generate random solutions Sol_i
end for
Calculate the fitness of the solutions in the population
Select Sol_best // the best solution among the population
Set iteration ← 0;
while (iteration < NumOfGen)
    select two parents (S1 and S2) using RWS
    apply crossover and mutation operators on S1 and
        S2 to produce offspring S1' and S2'
    Evaluate S1' and S2' // Using RST
    for i =1 to number of offspring DO
        Call SA (Figure 2)
    end for
    iteration++;
    update Population
end while
return |Sol_best|, δ(Sol_best), Sol_best;
```

Figure 2 Pseudo code for simulated annealing algorithm

```
T0 = 2*|C| where |C| is the number of attributes for each
dataset
Sol_bestSA ← Si'
δ (Sol_bestSA) ← δ (Si) // δ indicates the quality of the
solution
while T > T0
    generate at random a new solution Sol_trial in the
    neighbour of Si'
    calculate δ(Sol_trial)
    if (δ(Sol_trial) > δ(Sol_bestSA))
        Si' ← Sol_trial; Sol_bestSA ← Sol_trial;
        δ(Si') ← δ(Sol_trial); δ(Sol_bestSA) ← δ(Sol_trial);
    else if ((δ(Sol_trial) == δ(Sol_bestSA))
        Calculate |Sol_trial| and | Sol_bestSA |;
        if (|Sol_trial| < | Sol_bestSA |)
            Si' ← Sol_trial; Sol_bestSA ← Sol_trial;
            δ(Si') ← δ(Sol_trial);
            δ(Sol_bestSA) ← δ(Sol_trial);
        end if
    else // accepting the worse solution
            Calculate θ = δ(Sol_trial) – δ(Sol_bestSA))
            Generate a random number, P = [0,1];
            if (P ≤ e^(-θ/T))
                Si' ← Sol_trial; δ(Si) ← δ(Sol_trial);
            end if
    end if
    T = 0.93 * T; // update temperature
end while
if δ(Sol_bestSA) > δ(Sol_best)
    Sol_best ← Sol_bestSA
end if
```

The genetic algorithm starts by generating population of solutions, which consists of 100 randomly generated solutions. The population is evaluated then (by calculating the dependency degree of each solution (fitness value)) and the best solution among the population is assigned to Sol_{best}. Then, two solutions (which called parents) are selected based on RWS. In RWS, The fitness value is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i in the population, its probability of being selected is

$$p_i = f_i \Big/ \sum_{j=1}^{N} f_i \tag{10}$$

where N is the number of individuals in the population. After that, these two parents undergo through genetic operators (one point crossover and mutation) to generate two new offspring.

Since this work proposes a memetic algorithm, the next step is applying a local search algorithm (SA in this work) on the two generated offspring which will be used as initial solution for the SA process (individually). Then it improves the initial solution by searching its neighbourhood for better solutions based on their evaluation, the generated solution is called the trial solution (Sol_{trial}). If it is better than the best value so far (Sol_{bestSA}), then the Sol_{trial} is accepted. If the quality of Sol_{trial} is equal to the quality of the Sol_{bestSA}, then the algorithm checks the number of attribute (cardinality) for both solutions. If the cardinality of Sol_{trial} is less than the cardinality of $Sol_{bestSAs}$, then Sol_{trial} is accepted. Otherwise, a random number P (between 0 and 1) is generated in order to accept a worse solution. Sol_{trial} is accepted if P is less than the probability which is computed by $e^{-\theta/T}$. The process is repeated and the value of the temperature is updated until exceeding the maximum temperature, then the process is stopped. Here, the best solution found from the SA process (Sol_{bestSA}) is compared to best one in the population (Sol_{best}), if it has a higher dependency, then Sol_{best} is assigned the value of Sol_{bestSA}. Then same SA process is repeated for the second time to improve the second offspring.

Once local search is finished, the population is updated. The algorithm stops when the termination criterion is met (in this work, the termination criterion is set as the number of generations).

4 Experimental results

The proposed algorithm was programmed using Java and simulations which were performed on the Intel Pentium 4 2.0 GHz. Table 4 shows the parameters used in the GAs tested in this paper as taken from work of Jensen and Shen (2004).

Table 4 Parameter settings for GAs

Parameters	*Value*
Population size	100
Number of generation	100
Crossover rate	0.6
Mutation rate	0.4

We considered 13 well-known UCI data sets that can be taken from work of Blake and Merz (1998), which is shown in Table 5. For every data set, the algorithm was run 20 times as suggested by Jensen and Shen (2003, 2004).

The comparison of our results with other results in the literature for these benchmark data sets is presented in Tables 6 and 7. The approaches compared here are tabu search (TSAR) by Hedar et al. (2008), ant colony optimisation (AntRSAR) by Jensen and Shen (2003), genetic algorithm (GenRSAR) by Jensen and Shen (2004), 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), and a constructive hyper-heuristics (CHH_RSAR) by Abdullah et al. (2010).

Table 5 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

Table 6 Comparison results 1

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

Table 7 Comparison results 2

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

The entries in Tables 6 and 7 represent the number of attributes in the minimal reducts obtained by each method. The superscripts in parentheses represent the number of runs that achieved the minimal reducts, i.e. $8^{(16)}9^{(4)}$ means that the algorithm is able to obtain 8 attributes in 16 runs, while 9 attributes are selected in 4 runs. The number of attribute without superscripts means that the method could obtain this number of attribute for all runs. Our approach is able to produce two best-known results on Derm2 and WQ data sets. It is better than TSAR on seven data sets and ties on six data sets; when compared with AntRSAR, it is able to outperform it on six data sets (ties on six data sets); also better than ACOAR on two data sets (ties on ten data sets); at the same time it perform better than SSAR on five data sets. However, MA-RSAR has obtained better results than GD-RSAR in all data sets. It outperforms IS-CNS, HVNS-AR, CHH_RSAR on 4, 4, and 5 instances, respectively.

We are interested to compare our approach with GenRSAR and SimRSAR (Jensen and Shen, 2004). GenRSAR employs a genetic search strategy in order to determine rough set reducts. The initial population consists of 100 randomly generated feature subsets, the probability of mutation and crossover is set to 0.4 and 0.6, respectively, and the number of generations is set to 100. On the other hand, SimRSAR employs a simulated annealing-based feature selection mechanism. The initial temperature of the system is estimated as $2*|C|$, where C is the number of features in the data set, and the cooling schedule is $T = 0.93*T$ which is adopted from the work of Jensen and Shen (2004).

These two methods are selected here because we want to see how the hybridisation of genetic operators and simulated annealing as a local search applied in this work outperforms other methods in isolation.

According to the experimental results, it can be seen that MA-RSAR is able to obtain better results on all data sets compared to GenRSAR, whereas MA-RSAR works better than SimRSAR on 7 data sets out of 13 data sets (ties on six data sets). This is clearly shown that the local search (i.e. simulated annealing in this case) embedded inside genetic operators is able to enhance the performance of the algorithm when compared to genetic operators or simulated annealing alone. On the whole, our hybridisation algorithm (memetic algorithm) works reasonably well across all problem instances and it does not perform worst in any of the comparisons due to its ability to balance between the global search (by GA) and the intensification (by simulated annealing) during the search process.

5 Conclusion and future work

The memetic algorithm for attribute reduction problem in rough set theory has been studied in this paper. To our knowledge, this is the first such algorithm aimed at this problem domain. The performance of the proposed algorithm is tested on standard benchmark data sets and comparison results are presented. Experimental results show that our approach is able to produce three best-known results on the literature and is comparable with other approaches in the literature on the rest of the data sets. Our future work will concentrate on incorporating a fuzzy logical principle in identifying the acceptance of the generated trial solution based on certain rules generated from an intelligent fuzzy membership function and also try to reduce the number of parameter setting. Again, this is subject to our future work.

References

Abdullah, S. and Jaddi, N.S. (2010) 'Great deluge algorithm for rough set attribute reduction', Zhang, Y., Cuzzocrea, A., Ma, J., Chung, K-I., Arslan, T. and Song, X. (Eds): *Database Theory and Application, Bio-Science and Bio-Technology*, Springer, Berlin Heidelberg, 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', *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', *Intelligent Systems Design and Applications (ISDA)*, pp.1015–1020.

Bae, C., Yeh, W-C., Chung, Y.Y. and Liu, S-L. (2010) 'Feature selection with intelligent dynamic swarm and rough set', *Expert Systems with Applications*, Vol. 37, No. 10, pp.7026–7032.

Banati, H. and Bajaj, M. (2011) 'Fire fly based feature selection approach', *International Journal of Computer Science*, Vol. 8, No. 4, pp.473–480.

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*. Physica Verlag, Heidelberg, New York, pp.49–88.

Bello, R., Gomez, Y., Nowe, A. and Garcia, M.M. (2007) 'Two-step particle swarm optimization to solve the feature selection problem', *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications*, IEEE Computer Society, pp.691–696.

Blake, C.L. and Merz, C.J. (1998) *UCI Repository of Machine Learning Databases*. Available online at: http://www.ics.uci.edu/~mlearn/ (accessed on 1 February 2010).

Chen, Y., Miao, D. and Wang, R. (2010) 'A rough set approach to feature selection based on ant colony optimization', *Pattern Recognition Letters*, Vol. 31, No. 3, pp.226–233.

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

Fan, H. and Zhong, Y. (2012) 'A rough set approach to feature selection based on wasp swarm optimization', *Journal of Computational Information Systems*, Vol. 8, No. 3, pp.1037–1045.

Hart, W.E., Krasnogor, N. and Smith, J.E. (2005) *Recent Advances in Memetic Algorithms*, Springer-Verlag, Berlin Heidelberg.

Hedar, A-R., Wang, J. and Fukushima, M. (2008) 'Tabu search for attribute reduction in rough set theory', *Soft Computing*, Vol. 12, No. 9, pp.909–918.

Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems*, MIT Press.

Hu, Y., Ding, L., Xie, D. and Wang, S. (2012) 'A novel discrete artificial bee colony algorithm for rough set-based feature selection', *International Journal of Advancements in Computing Technology*, Vol. 4, No. 6, pp.295–305.

Jensen, R. and Shen, Q. (2003) 'Finding rough set reducts with ant colony optimization', *Proceedings of the 2003 UK Workshop on Computational Intelligence*, pp.15–22.

Jensen, R. and Shen, Q. (2004) 'Fuzzy-rough attribute reduction with application to web categorization', *Fuzzy Sets System*, Vol. 141, No. 3, pp.469–485.

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

Jihad, S.K. and Abdullah, S. (2010) 'Investigating composite neighbourhood structure for attribute reduction in rough set theory', *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', *Machine Learning: Proceedings of the 11th International*, 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*, Vol. 29, No. 9, pp.1351–1357.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, pp.671–680.

Kogilavani, A. and Balasubramanie, P. (2011) 'Multi-document summarisation using genetic algorithm-based sentence extraction', *International Journal of Computer Applications in Technology*, Vol. 40, No. 4, pp.246–253.

Kohavi, R. and John, G.H. (1997) 'Wrappers for feature subset selection', *Artificial Intelligence*, Vol. 97, Nos. 1/2, pp.273–324.

Koller, D. and Sahami, M. (1996) 'Toward optimal feature selection', *International Conference on Machine Learning*, pp.284–292.

Michalewicz, Z. (1996) *Genetic Algorithms+ Data Structures*, Springer, New York.

Ming, H. (2008) 'Feature selection based on ant colony optimization and rough set theory', *International Symposium on Computer Science and Computational Technology (ISCSCT'08)*, pp.247–250.

Moscato, P. (1989) *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms*, Report, California Institute of Technology.

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

Pawlak, Z. (1991) *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Boston, MA.

Suguna, N. and Thanushkodi, K. (2010) 'A novel rough set reduct algorithm for medical domain based on bee colony optimization', *Journal of Computing*, Vol. 2, No. 6, pp.49–54.

Wang, J., Hedar, A-R. and Wang, S. (2007a) 'Scatter search for rough set attribute reduction', *2nd International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2007)*, 14–17 September 2007, pp.236–240.

Wang, J., Hedar, A-R., Wang, S. and Ma, J. (2012a) 'Rough set and scatter search metaheuristic based feature selection for credit scoring', *Expert Systems with Applications*, Vol. 39, No. 6, pp.6123–6128.

Wang, J., Hedar, A-R., Zheng, G. and Wang, S. (2009) 'Scatter search for rough set attribute reduction', *International Joint Conference on Computational Sciences and Optimization (CSO 2009)*, pp.531–535.

Wang, G., Wang, S-J., Shi, L., Huang, D., Chen, H., Liu, Y. and Peng, X. (2012b) 'Study of adaptive parameter control for ant colony optimization applied to feature selection problem', *Advanced Science Letters*, in press.

Wang, X., Yang, J., Teng, X., Xia, W. and Jensen, R. (2007b) 'Feature selection based on rough sets and particle swarm optimization', *Pattern Recognition Letters*, Vol. 28, No. 4, pp.459–471.

Wu, J., Cai, Z. and Ao, S. (2012) 'Hybrid dynamic k-nearest-neighbour and distance and attribute weighted method for classification', *International Journal of Computer Applications in Technology*, Vol. 43, No. 4, pp.378–384.

Wu, J., Qiu, T., Wang, L. and Huang, H. (2011) 'An approach to feature selection based on ant colony optimization and rough set', in Chen, R. (Ed.): *Intelligent Computing and Information Science*, Springer, Berlin, Heidelberg, pp.466–471.

Yang, J. and Honavar, V.G. (1998) 'Feature subset selection using a genetic algorithm', *IEEE Intelligent Systems*, Vol. 13, No. 2, pp.44–49.

Zhu, Z., Jia, S. and Ji, Z. (2010) 'Towards a memetic feature selection paradigm [Application Notes]', *Computational Intelligence Magazine, IEEE*, Vol. 5, No. 2, pp.41–53.

View publication stats