I.J. Modern Education and Computer Science, **2015, 2, 16-23**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.02.03

An Efficient Algorithm for Finding a Fuzzy Rough Set Reduct Using an Improved Harmony Search

Essam Al Daoud
Essam Al Daoud, Computer Science Department, Zarqa University, Zarqa, Jordan.
Email: essamdz@zu.edu.jo

Abstract—To increase learning accuracy, it is important to remove misleading, redundant, and irrelevant features. Fuzzy rough set offers formal mathematical tools to reduce the number of attributes and determine the minimal subset. Unfortunately, using the formal approach is time consuming, particularly if a large dataset is used. In this paper, an efficient algorithm for finding a reduct is introduced. Several techniques are proposed and combined with the harmony search, such as using a balanced fitness function, fusing the classical ranking methods with the fuzzy-rough method, and applying binary operations to speed up implementation. Comprehensive experiments on 18 datasets demonstrate the efficiency of using the suggested algorithm and show that the new algorithm outperforms several well-known algorithms**.**

Index Terms—Discernibility matrix, Feature selection, Fuzzy rough set, Harmony search, Optimization.

I. Introduction

Several methods have been used in recent decades to reduce the number of attributes in machine learning and data mining applications. However, the major drawback of the classical methods is that the optimal subset is not guaranteed to be found by either a theoretical or practical approach. Therefore, fuzzy rough sets have become a popular tool for discovering the optimal or near-optimal subset [1]. Fuzzy rough set is advocated for handling real attributes, discrete attributes, or mixtures of both. It is a suitable tool for dealing with noisy, vague, uncertain, or inexact information. Furthermore, additional information about the data or the source of the data, such as the probability distribution, is not needed [2, 3]. The most successful application of fuzzy rough sets is finding the optimal subset of attributes, which are equivalent to the complete set of attributes in terms of classification accuracy or similar tasks [4, 5]. There are several advantages to using the optimal subset of attributes instead of the complete set of attributes. These include increased classification accuracy, saved computation time and storage space, removal of irrelevant attributes, reduced dimensionality, facilitation of extraction of the rules, and interpretation of the results [6].

Finding the optimal subset using fuzzy rough set techniques is an NP-complete problem; thus, many heuristic, greedy, and dynamic algorithms have been suggested in the literature to overcome this obstacle and reduce the time required to find a suitable subset [7]. Two main fitness functions are generally used. The first is based on the degree of dependency, and the second is based on a discernibility matrix. Chen et al. constructed a reduct by using minimal elements in the discernibility matrix [8]. Zhang et al. used a greedy technique in which priority was given to the highest-appearing frequency attribute in the discernibility matrix [9]. Jensen and Shen modified the original rough set algorithm by defining a new entropy equation as a fitness function [10]. Wang et al. used particle swarm optimization to find a reduct in which the position of the best particle (the reduct) was updated after calculating the classification quality [11]. Diao and Chen modified the harmony search by treating the musicians independently; a feature is included in the subset if one musician votes for it. They called the suggested model vertical harmony search (VHS) [12]. Tsang et al. developed an algorithm using a discernibility matrix to compute all of the attributes' reductions [13]. Another direction of rough set research focuses on enhancing the accuracy of special cases, such as imbalanced or noisy data. Liu et al. introduced three algorithms based on rough set to deal with imbalanced data: weighted attribute reduction, weighted rule extraction, and weighted decision algorithm [14]. Chen et al. developed a kernel-based rough theory and used kernels as fuzzy similarity relations [15-17]. Hu et al. suggested a new dependence function inspired by a soft margin support vector machine, and they showed that the new model could be used to reduce the influence of noise [18]. In this paper, contrary to previous studies, the fitness functions of the harmony search utilize classical ranking techniques, a discernibility matrix, and the degree of dependency of each individual attribute. Moreover the suggested operations can easily be speeded up by converting them to binary operations.

The rest of this paper is organized as follows: Section 2 introduces the basics of the rough set theory and the reduct extraction algorithms. Section 3 discusses the fuzzy rough sets and the related notation, and Section 4 provides a short introduction to the harmony search. Section 5 describes the suggested fitness function, the

probability distribution of the attributes, the proposed binary operations, and the modified harmony search for reduct finding. Section 6 compares the suggested algorithm with previous studies, and the conclusion is provided in Section 7.

II. Rough Sets

An approximate space or information system is [19]:

$$IS = (U, A, V, f) \tag{1}$$

where $U=\{x_1, x_2, x_3, \ldots, x_N\}$ is a set of N objects called the universe, A is a set of features (or attributes) such that

$$V = \bigcup_{a \in A} V_a \tag{2}$$

for every $a\,HMCRrand\,A$, and $f : U \times A \rightarrow V$ is the information function (also called the total decision function) such that $f(x, a) \in V_a$ and $\forall x \in U$. The attributes can be classified into two subsets, i.e., decision attributes D and condition attributes C, such that $A = C \cup D$ and $C \cap D = \varphi$. Thus, the decision table is

$$IS = (U, C, D, V, f) \tag{3}$$

The subset $P \subseteq A$ generates an indiscernibility relation as follows:

$$IND(P) = \{(x, y) \in U^2 : \forall a \in P, f(y, a) = f(x, a)\} \tag{4}$$

and the partition of U by P is

$$U / IND(P) = \{p_1, p_2, \ldots p_k\} \tag{5}$$

where p_i is an equivalence class. Let $X \subseteq U$, then the lower approximation of X with respect to P is defined as:

$$P_*(X) = \bigcup \{p_i \mid p_i \in U / IND(P), p_i \subseteq X\} \tag{6}$$

and the upper approximation of X with respect to P is defined as:

$$P^*(X) = \bigcup \{p_i \mid p_i \in U / IND(P), p_i \cap X = \varphi\} \tag{7}$$

The positive, negative, and boundary regions of D on P can be defined as follows:

$$POS_P(D) = \bigcup_{X \in U/D} P_*(X) \tag{8}$$

$$NEG_P(D) = U - \bigcup_{X \in U/D} P^*(X) \tag{9}$$

$$BND_P(D) = \bigcup_{X \in U/D} P^*(X) - \bigcup_{X \in U/D} P_*(X) \tag{10}$$

A reduct *RED(IS)* is the minimal subset of attributes that is equivalent to the whole set of attributes and can be used to classify the objects in the universe set efficiently, while the core is the intersection of all reducts: $CORE(IS) = \bigcap RED(IS)$. The accuracy of the approximation is defined as:

$$\alpha_P(D) = \frac{|POS_P(D)|}{\sum_{X \in U/D} P^*(X)} \tag{11}$$

The degree of dependency of D on P, or the quality of the classification, is

$$\gamma_P(D) = \frac{|POS_P(D)|}{|U|} \tag{12}$$

if $\gamma_P(D) < 1$, then D depends partially on P, while if $\gamma_P(D) = 1$, then D depends totally on P. A discernibility matrix is a symmetric $U \times U$ matrix and can be defined as follows:

$$d_{ij} = \{c \in C \mid f(x_i, c) \neq f(x_j, c)\} \tag{13}$$

The core and the reduct can be redefined by using the discernibility matrix such that the core is the union of the single entries, while the reduct is a minimal subset M where $M \cap d_{ij} \neq \phi$ for all entries d_{ij} in the discernibility matrix. Two main methods are used to find a reduct (the minimal subset of attributes). The first is by using the degree of dependency, such as QuickReduct, which is described in Algorithm 1, and the second is by using the discernibility matrix [20].

Algorithm 1: QuickReduct
Input: C, the set of all the conditional attributes. D, the set of decision attributes
Output: A Reduct RED
$RED = \varphi$
While $\gamma_{RED}(D) \neq \gamma_C(D)$
  $T = RED$
  $\forall x \in (C - RED)$
    If $\gamma_{RED \cup \{x\}}(D) > \gamma_T(D)$
      $T = RED \cup \{x\}$
  $RED = T$
Return RED

Another technique for finding a minimal subset is by using an entropy-based reduction as follows:

$$E(A) = -\sum_{i=1}^{m} p(a_i) \sum_{j=1}^{n} p(c_j \mid a_i) \lg(p(c_j \mid a_i)) \tag{14}$$

where a_i are the attributes and c_j are the targets. The entropy-based algorithm replaces the increment condition in QuickReduct by $E(RED \cup \{x\}) < E(T)$.

III. Fuzzy Rough Sets

To improve the attributes selection, the previous rough set algorithms must be extended to fuzzy rough sets. The main reasons are that first, most datasets contain real-valued attributes, and second, rough set algorithms cannot handle noisy data. Fuzzy equivalence classes are the central concept of fuzzy rough sets and can be defined as follows [15, 21]:

$$\mu_{[x]R}(y) = \mu_R(x, y) \quad \forall y \in X \tag{15}$$

where $\mu_R(x, y)$ is a fuzzy similarity relation and can be any distance function or kernel. In this paper the Gaussian function is used:

$$\mu_R(x, y) = \exp\left(\frac{-\| x - y \|^2}{\delta}\right) \tag{16}$$

Therefore, the fuzzy-rough lower and upper approximations can be redefined as follows:

$$\mu_{P_*}(E_k) = \inf_x \max\{1 - \mu_{E_k}(x), \mu_X(x)\} \quad \forall k \tag{17}$$

$$\mu_{P^*}(E_k) = \sup_x \min\{\mu_{E_k}(x), \mu_X(x)\} \quad \forall k \tag{18}$$

where E_k is a fuzzy equivalence class. The fuzzy positive region can be defined by

$$\mu_{POS_p}(D) = \sup_{X \in U/D} \mu_{P_*}(X) \tag{19}$$

and the fuzzy-rough dependency function is

$$\beta'_P(D) = \frac{\sum_{x \in U} \mu_{POS_p}(x)}{|U|} \tag{20}$$

The extended version of QuickReduct is described in Algorithm 2 [22].

Algorithm 2: Fuzzy-Rough QuickReduct
Input: C, the set of all the conditional attributes. D, the set of decision attributes
Output: A Reduct RED
RED= φ, $\beta_{best} = 0$, $\beta_{prev} = 0$
Do
  T=RED
  $\beta_{prev} = \beta_{best}$
  $\forall x \in (C - RED)$
    If $\beta_{RED \cup \{x\}}(D) > \beta_T(D)$
      $T = RED \cup \{x\}$
      $\beta_{best} = \beta_T(D)$
  RED=T
Until $\beta_{prev}(D) == \beta_{best}(D)$
Return RED

IV. Harmony Search

Harmony search (HS) was introduced by Geem et al. in 2001 [23- 25]. The basic idea of HS is to create a new vector from the previous vectors in the harmony memory, and if the new one is better than the worst vector, to add it to the harmony memory. Algorithm 3 describes the main steps involved in HS.

Algorithm 3: Harmony search
Input: The bandwidth (bw), pitch adjusting rate (PAR), and considering rate (HMCR)
Output: The best vector
Repeat until the termination condition is held
  for each component i do
    if $HMCR \geq rand$
      $\alpha^i_{new} = \alpha^i_j$
      if $PAR \geq rand$
        $\alpha^i_{new} = \alpha^i_{new} \pm rand \times bw$
    else
      $\alpha^i_{new} = rand$
  If the new vector is better than the worst, replace the worst vector

where $j \in l$ is the memory size. To improve the harmony search, Mahdavi et al. updated the *bw* and the *PAR* as follows [26]:

$$bw(t) = bw_{\max} e^{\left(\frac{h}{MaxIter} \; iter\right)} \tag{21}$$

where

$$h = \ln\left(\frac{bw_{\min}}{bw_{\max}}\right) \tag{22}$$

andl

$$PAR(t) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{MaxIter} iter \tag{23}$$

where *min* and *max* are the minimum and maximum values. other meta-heuristic techniques can be used such as genetic algorithms or Tabu search [27, 28].

V. A New Reduct Algorithm

Several techniques are imbedded in the standard harmony search to find a reduct efficiently in Algorithm 4. The suggested techniques include using a balanced fitness function, fusing the classical ranking methods with the fuzzy-rough method, and applying binary operations to speed up implementation. Furthermore a new vector is added to the harmony memory if it is better than the best vector, and the pitch adjusting step is removed.

A suitable fitness function must maximize the covered subsets (a subset is covered if at least one attribute from

this subset is selected) in the discernibility matrix and minimize the number of selected attributes. Thus the proposed fitness function is:

$$fit(v)= (1-\alpha)\, c+ \alpha\, z \qquad (24)$$

where v is a vector that represents the selected attributes (one at position i if the i^{th} attribute is selected, otherwise zero), c is the percentage of covered subsets by the vector v, z is the percentage of unselected attributes, and $\alpha \in [0\ ,0.5]$ is a dynamic constant (in this paper α starts with 0.25*)*. In this way greater priority is given to the vectors with fewer attributes. However after enough time spent searching, α must be decreased in order to try another vector.

Instead of randomly testing and selecting a new attribute as suggested by Fuzzy-Rough QuickReduct and previous studies, the proposed algorithm uses the filtering and ranking methods as a recommender such that a new probability distribution is constructed and used according to the following equation:

$$Dist(i) = \left(\sum_{k=1}^{s} rank_k(att_i)/s * m \right) * d \qquad (25)$$

where i indicates the i^{th} attribute, att_i is the value of the i^{th} attribute, s is the number of ranking techniques, m is the number of attributes, and $d \in [0,1]$ is the constant that is used to reduce the probability of selecting an attribute (in this paper d is 0.75). The aim of this constant is to consistently prevent the attributes that have high ranks from being selected most of the time. Two ranking methods were used in this study. The first was the T-test, and the second was the fuzzy-rough dependency function for each individual attribute. Both methods can be implemented linearly. The T-test can be described as follows:

$$T(C,D) = \frac{\mu_+ - \mu_-}{\sqrt{\sigma_+^2/m_+ + \sigma_-^2/m_-}} \qquad (26)$$

where μ is the mean, σ is the standard deviation, and m is the number of samples. The positive and negative signs indicate the positive and negative regions.

Algorithm 4: A modified harmony search algorithm for a reduct

Input: C, the set of all the conditional attributes. D, the set of decision attributes

Output: A reduct

1- Calculate the discernibility matrix by (8).

2- Find the CORE from the discernibility matrix (the union of all the single entries).

3- For each attribute, $att_i \notin CORE$

$$find\ \ Dist(i) = \left(\sum_{k=1}^{s} rank_k(att_i)/s * m \right) * d$$

4- Generate t vectors. Each vector contains attributes at each position corresponding to the CORE attributes and whenever dist(i)>rand(0,1), where t is the length of the harmony memory.

5- Find the fitness for each vector by (18) and let v_{best} be the best fitness.

6- Repeat until the discernibility matrix is covered or the number of iterations is fulfilled.

Let v_{new} contain one at each position corresponding to the CORE attributes

for each component i do.

if $v_{new}^i \neq 1$

if $HMCR \geq rand$

$v_{new}^i = v_j^i$

else

if $Dist(i) > rand(0,1)$

$v_{new}^i = 1$

else

$v_{new}^i = 0$

if $fit(v_{new}) > fit(v_{best})$

$v_{best} = v_{new}$

Replace a random vector from the harmony memory.

Add the new vector to the harmony memory.

7- Return v_{best} .

Most of the operations in Algorithm 4 can be implemented using binary operations. For example, consider the following discernibility matrix:

$$DM = \begin{bmatrix} & & & \\ \{a_2,a_5\} & & & \\ \{a_3\} & \{a_1,a_3,a_4\} & & \\ \{a_1,a_3,a_4\} & \{a_4,a_5\} & \{a_3\} & \\ \{a_2,a_4\} & \{a_3\} & \{a_1,a_3\} & \{a_1,a_5\} \end{bmatrix}$$

During implementation, *DM* is re-represented as the following

DM = [01001, 00100, 10110, 01010, 10110, 00011, 00100, 00100, 10100, 10001]

Let, for example,

$$v_{best} = [10100].$$

Therefore the corresponding cover is

$$CV_{best} = [0\ 1\ 2\ 0\ 2\ 0\ 1\ 1\ 2\ 1]$$

The numbers in this vector indicate the number of attributes in a covered subset, while the zeros indicate that the subset is not yet covered. Thus all the subsets are covered if all the entries of the previous vector are greater than zero. To update CV_{best} based on a new vector, the difference between v_{best} and v_{new} is calculated, XNOR ($\otimes$) operation is applied on *DM*, and incrementing or decrementing the CV_{best} elements. To illustrate this point, consider $v_{new} = [10101]$, then

$$df = v_{new} - v_{best} = [00001]$$

and

$$R = DM \otimes df$$

therefore

$$R = [00001, 00000, 00000, 00000, 00000, 00001, 00000, 00000, 00000, 00001]$$

For each non-zero element in *R*, the vector CV_{best} will be increased by one and stored in the temporary vector T. Thus

$$T = [1\ 1\ 2\ 0\ 2\ 1\ 1\ 1\ 2\ 2]$$

In this case, v_{new} is better than v_{best} because T contains fewer zeros than CV_{best}, therefore

$$v_{best} = v_{new}$$

and

$$CV_{best} = T.$$

In the event that *df* contains negative values, all the entries in CV_{best} corresponding to the non-zeros in *R* will be decreased by one.

VI. Experimental Results

In this section the proposed algorithm is tested using 18 datasets from UCI [29]. The selected datasets have mixed features (discrete and continuous). The number of features, samples, and classes are summarized in Table 1. All experiments were carried out using Matlab 9 on a dual-core CPU with 2.3 GHz and 1.8 GB of RAM. Table 2 compares the length of the reduct with four algorithms: Fuzzy-Rough QuickReduct (FRQR), vertical harmony search (VHS) from [18], particle swarm optimization (PSO) from [15], and matrix from [24]. It is important to note that the best reduct is not the shortest one the one that is closest to the optimal; thus, the matrix algorithm and the proposed algorithm are better than the other algorithms in term of reduct length. As shown in Tables 3 and 4, the support vector machine (SVM) and Neural networks, respectively, are applied to the selected reduct for each algorithm. In both methods, ten-fold is used to estimate the classification accuracy. The results indicate that the matrix algorithm and the proposed algorithm have almost the same classification rate, outperform the other algorithms, and are even better than the complete set of features. Table 5 compares the required times to find the reduct using each algorithm. It is clear that the proposed algorithm is faster than the other tested algorithms for most of the tested datasets. The efficiency of the new algorithm becomes even more obvious for larger datasets, such as German, car, and wdbc.

VII. Conclusion

In this paper, we presented a new reduct algorithm based on a modified harmony search. The proposed fitness function integrates the advantages of several techniques, classical ranking methods, discernibility matrix, and degree of dependency. In contrast to previous work, the suggested algorithm can find the minimum subset of attributes without sacrificing accuracy or computation time. Moreover, the superiority of the suggested algorithm becomes clearer when larger datasets are used. A future investigation will focus on extending the suggested algorithm to deal with imbalanced and very noisy data. This can be done by using another kernel as a membership function or by integrating soft margin with the suggested algorithm.

Table 1. Description of the datasets

No	Data	Samples	Features	Class
1	Pima	768	9	4
2	Monk1	124	7	3
3	Bridges	108	13	2
4	Breast	286	9	2
5	Horse	368	22	2
6	Votes	435	16	2
7	Credit	690	15	2
8	Tic	958	9	2
9	German	1000	24	2
10	Zoo	101	16	7
11	Wine	178	13	3
12	Glass	214	9	6
13	Heart	303	13	5
14	Solar	323	10	3
15	iono	351	34	2
16	wdbc	569	31	2
17	Car	1728	7	6
18	Hepatitis	155	19	2

Table 2. Comparison of reduct lengths using different algorithms for each dataset

No	Data	FRQR	VHS	PSO	Matrix	New
1	Pima	7	5	6	4	5
2	Monk1	5	3	5	3	3
3	Bridges	4	3	4	2	2
4	Breast	6	5	5	4	5
5	Horse	8	8	8	4	4
6	Votes	11	9	9	8	8
7	Credit	10	8	9	8	8
8	Tic	8	8	8	8	8
9	German	15	10	12	10	11
10	Zoo	8	7	8	5	5
11	Wine	9	5	7	6	6
12	Glass	7	5	7	3	3
13	Heart	12	8	10	6	6
14	Solar	8	7	7	7	7
15	iono	25	7	10	18	18
16	wdbc	23	19	21	19	19
17	Car	7	6	7	6	6
18	Hepatitis	9	6	9	4	4

Table 3. Comparison of SVM classification accuracy using different algorithms for each dataset

No	Data	All Data	FRQR	VHS	PSO	Matrix	New
1	**Pima**	70.1 (5.3)	70.1(3.7)	71.6(8.0)	71.7(6.7)	72.1(4.2)	72.4(5.3)
2	**Monk1**	94.7(2.1)	94.6(3.0)	95.0(9.4)	95.4(2.5)	98.4(4.1)	98.2(4.6)
3	**Bridges**	81.4(5.2)	81.5(4.2)	83.5(5.1)	82.2(4.4)	86.1(5.3)	85.8(5.1)
4	**Breast**	81.4(3.0)	80.6(5.2)	81.9(4.7)	87.8(4.9)	87.7(3.6)	87.5(3.9)
5	**Horse**	85.4(3.8)	85.6(2.0)	88.3(2.9)	89.2(5.5)	91.8(4.9)	91.8(5.0)
6	**Votes**	91.0 (2.5)	91.8(2.2)	95.2(2.1)	94.3(3.3)	96.6(2.3)	96.2(2.1)
7	**Credit**	83.0(6.9)	82.7(3.6)	84.3(5.5)	83.9(3.4)	85.4(7.7)	85.2(7.5)
8	**Tic**	94.8(1.1)	95.3(1.9)	97.7(0.6)	97.1(1.9)	97.2(2.4)	97.7(1.1)
9	**German**	60.7 (8.9)	60.5 (8.0)	62.1(8.3)	60.6(5.7)	70.3(6.0)	69.3(5.1)
10	**Zoo**	85.4(3.5)	83.3(6.0)	91.3(4.0)	91.2(3.8)	98.7(0.7)	98.8(0.8)
11	**Wine**	93.8(1.3)	94.7(1.7)	97.5(1.9)	97.8(1.1)	97.8(1.1)	97.4(1.1)
12	**Glass**	60.2 (8.4)	60.1(7.5)	63.3(5.4)	63.6(4.1)	65.7(5.8)	65.8(5.3)
13	**Heart**	82.5 (3.9)	82.9(5.5)	81.1(3.7)	85.0(2.1)	85.0(3.2)	85.1(2.7)
14	**Solar**	83.2 (6.3)	83.5(4.4)	83.0(3.9)	84.1(5.2)	83.8(7.6)	82.4(7.2)
15	**iono**	93.2 (1.6)	93.2(3.9)	94.7(1.3)	92.1(3.3)	94.9(2.7)	94.2(3.4)
16	**wdbc**	96.4 (2.0)	96.4 (1.7)	97.2(1.7)	97.2(1.6)	97.3(1.3)	96.6(1.0)
17	**Car**	95.6(1.5)	95.1 (2.5)	96.2(1.4)	96.7(1.2)	97.1(0.6)	98.2(0.7)
18	**Hepatitis**	86.2(3.2)	86.0(5.5)	81.3(7.8)	83.5(2.0)	90.9(3.4)	91.2(2.6)
	Average	84.3	84.3	85.8	86.3	88.7	88.5

Table 4. Comparison of Neural network classification accuracy using different algorithms for each dataset

No	Data	All Data	FRQR	VHS	PSO	Matrix	New
1	**Pima**	72.3 (6.2)	71.5(7.2)	72.4(7.5)	72.8(6.3)	74.4(3.5)	74.3(2.3)
2	**Monk1**	95.3(3.2)	95.0(3.3)	95.6(9.4)	95.8(2.7)	98.7(3.4)	98.6(3.2)
3	**Bridges**	79.2(3.6)	79.3(2.7)	80.7(4.3)	80.8(4.2)	85.7(4.1)	85.8(4.3)
4	**Breast**	80.6(4.1)	80.4(3.6)	83.7(2.6)	86.6(2.8)	87.5(4.1)	87.2(3.5)
5	**Horse**	82.5(4.3)	83.4(3.5)	83.8(3.2)	83.9(4.0)	89.5(3.5)	88.9(4.1)
6	**Votes**	86.7 (4.2)	85.5(3.9)	87.5(3.7)	88.6(3.5)	90.8(3.3)	92.2(3.0)
7	**Credit**	83.2(2.3)	81.9(2.5)	84.5(3.2)	83.2(2.7)	86.7(4.2)	86.3(5.0)
8	**Tic**	93.3(2.5)	93.5(2.1)	96.1(1.3)	95.9(2.0)	96.9(3.2)	96.6(2.4)
9	**German**	62.2 (5.5)	61.2 (6.2)	65.3(4.8)	62.4(4.5)	68.5(5.3)	68.7(4.9)
10	**Zoo**	90.2(4.6)	88.1(3.5)	94.2(3.9)	93.4(4.0)	98.3(0.4)	99.0(0.5)
11	**Wine**	95.6(1.6)	95.5(1.2)	97.1(1.7)	97.2(2.1)	98.0(1.2)	98.1(1.0)
12	**Glass**	62.3 (6.7)	62.3(5.3)	65.5(1.9)	66.1(5.2)	66.8(7.0)	67.2(4.7)
13	**Heart**	83.2 (2.5)	83.1(3.7)	82.2(2.8)	84.2(3.4)	85.8(2.5)	86.3(2.3)
14	**Solar**	86.2 (3.8)	86.1(4.3)	86.2(4.1)	88.1(3.2)	87.5(6.0)	88.1(3.8)
15	**iono**	89.9 (3.1)	91.2(4.2)	91.6(2.1)	92.2(4.2)	93.0(3.2)	92.1(2.8)
16	**wdbc**	92.1 (3.2)	93.2 (3.4)	93.5(2.2)	94.1(2.5)	94.2(3.3)	94.2(3.1)
17	**Car**	95.9(2.1)	95.8 (2.3)	96.3(1.3)	97.2(1.0)	97.9(1.1)	98.5(0.5)
18	**Hepatitis**	85.7(5.2)	85.2(5.7)	80.2(4.8)	83.2(2.6)	88.8(3.5)	89.1(3.1)
	Average	84.2	84.0	85.4	85.9	88.3	88.4

Table 5. Comparison of running times using different algorithms for each dataset and svm for classification

No	Data	VHS	PSO	Matrix	New
1	Pima	198	150	205	**80.4**
2	Monk1	27.1	13.6	**4.6**	6.1
3	Bridges	32.0	17.1	**5.1**	6.8
4	Breast	17.2	12.3	8.3	**8.1**
5	Horse	249	221	289	**123**
6	Votes	198	192	215	**115**
7	Credit	374	256	318	**136**
8	Tic	280	317	277	**82.8**
9	German	811	723	998	**316**
10	Zoo	22.0	18.9	**5.2**	7.1
11	Wine	11.4	11.5	12.7	**9.9**
12	Glass	29.3	27.2	24.7	**12.0**
13	Heart	33.9	46.6	55.1	**20.1**
14	Solar	36.2	55.0	53.6	**15.8**
15	iono	153	284	367	**76.7**
16	wdbc	982	1204	1873	**336**
17	Car	301	275	345	**129**
18	Hepatitis	24.4	19.2	7.6	**7.0**
	Average	210	214	281	**82.65**

REFERENCES

[1] S.Y. Zhao, E.C. Tsang and D.G. Chen, "The model of fuzzy variable precision rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 2, 2009, pp 451–467.

[2] S.Y. Zhao, E.C. Tsang, D.G. Chen and X. Z. Wang, "Building a rule-based classifier—A fuzzy-rough set approach," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, 2010, pp. 624–638.

[3] K. G. Saharidis, G. Kolomvos, and G. Liberopoulos, "Modeling and Solution Approach for the Environmental Traveling Salesman Problem," *Engineering Letters*, vol. 22, no. 2, 2014, pp. 70-74.

[4] A. Soleimani, and Z. Kobti, "Toward a Fuzzy Approach for Emotion Generation Dynamics Based on OCC Emotion Model," *IAENG International Journal of Computer Science*, vol. 41, no. 1, 2014, pp. 48-61.

[5] H.H. Huang, and Y. H. Kuo, "Cross-lingual document representation and semantic similarity measure: A fuzzy set and rough set based approach," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 6, 2010, pp. 1098–1111.

[6] T.J. Li, and W.X. Zhang, "Rough fuzzy approximations on two universes of discourse," *Inform. Sci.*, vol. 178, pp. 892–906, 2008.

[7] Q. Hu, S. An, X. Yu, and D. Yu, "Robust fuzzy rough classifiers," *Fuzzy Sets Syst.*, vol. 183, 2011, pp. 26–43.

[8] D. Chen, L. Zhang, S. Zhao, Q. Hu, and P. Zhu, "A Novel Algorithm for Finding Reducts With Fuzzy Rough Sets ," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, 2012, pp. 385-389.

[9] J. Zhang, J. Wang, D. Li, H. He, and J. Sun, "A New Heuristic Reduct Algorithm Base on Rough Sets Theory," In *Proceedings of The 4th International Conference of WAIM, Springer Berlin / Heidelberg, Advances in Web-Age Information Management, LNCS*, vol. 2762, 2003, pp. 247-253.

[10] R. Jensen and Q. Shen, "Finding rough set reducts with ant colony optimization," In *Proceeding of 2003 UK Workshop Computational Intelligence*, 2004, pp.15-22.

[11] X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen, "Feature selection based on Rough Sets and Particle Swarm Optimization," *Pattern Recognition Letters*, vol. 28, no. 4, 2007, pp. 459–471.

[12] R. Diao and Q. Shen, "Two New Approaches to Feature Selection with Harmony Search," *WCCI 2010 IEEE World Congress on Computational Intelligence*, 2010, pp. 18-23.

[13] E.C. Tsang, D. G. Chen, D. S. Yeung, X. Z. Wang, and J. T. Lee, "Attributes reduction using fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, 2008, pp.1130–1141.

[14] J. Liu, Q. Hu, and D. Yu, "A weighted rough set based method developed for class imbalance learning," *Information Sciences*, vol. 178, 2008, pp. 1235–1256.

[15] D. Chen, Q. Hu and Y. Yang, "Parameterized attribute reduction with Gaussian kernel based fuzzy rough sets," *Information Sciences*, vol. 181, 2011, pp. 5169–5179.

[16] Y. V. Bodyanskiy, O. K. Tyshchenko and D. S. Kopaliani, "A Multidimensional Cascade Neuro-Fuzzy System with Neuron Pool Optimization in Each Cascade," *International Journal of Information Technology and Computer Science*, vol 6, no 8, 2014, pp 11-17. DOI: 10.5815/ijitcs.2014.08.02.

[17] M. Barman and J. P. Chaudhury, "A Framework for Selection of Membership Function Using Fuzzy Rule Base System for the Diagnosis of Heart Disease," *International Journal of Information Technology and Computer Science*, vol 5, no 11, 2013, pp 62-70. DOI: 10.5815/ijitcs.2013.11.07.

[18] Q. Hu, S. An and D. Yu, "Soft fuzzy rough sets for robust feature evaluation and selection," *Information Sciences*, vol. 180, 2010, pp. 4384–4400.
[19] Z. Pawlak, "Rough Sets," *Int. J. Compute Inf. Sci.*, vol. 11, 1982, pp. 341–356.
[20] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers. 1991.
[21] X.D. Liu, W. Pedrycz, T.Y. Chai, and M. L. Song, "The development of fuzzy rough sets with the use of structures and algebras of axiomatic fuzzy sets," *IEEE Trans. Knowl. Data Eng.*, vol .21, no. 3, 2009, pp. 443–462.
[22] R. Jensen, and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, 2009, pp. 824–838.
[23] Z.W. Geem, J. H. Kim, and G.V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, 2011, pp. 60–68.
[24] Z.W. Geem, "Music-Inspired Harmony Search Algorithm: Theory and Applications," *Studies in Computational Intelligence, Springer*, vol. 191, 2009, pp. 1-14.
[25] G. Georgoulas, P. Karvelis, G. Iacobellis, V. Boschian, M. P. Fanti, W. Ukovich, and C. D. Stylios, "Harmony Search augmented with Optimal Computing Budget Allocation Capabilities for Noisy Optimization," *IAENG International Journal of Computer Science*, vol 40, no.4, 2013, pp. 285-290.
[26] O. M. Alia and M. Rajeswari, "The variants of the harmony search algorithm: an Overview," *Artif. Intell. Rev.*, vol. 36, 2011, pp. 49–68.
[27] M. Gabli, J. El Miloud, and M. El Bekkaye, " A Genetic Algorithm Approach for an Equitable Treatment of Objective Functions in Multi-objective Optimization Problems," *IAENG International Journal of Computer Science*, vol. 41, no. 2, 2014, pp. 102-111.
[28] K. Tamura, and H.Kitakami, "A New Distributed Modified Extremal Optimization using Tabu Search Mechanism for Reducing Crossovers in Reconciliation Graph and Its Performance Evaluation," *IAENG International Journal of Computer Science*, vol. 41, no. 2, 2014, pp. 131-140.
[29] UCI Machine Learning Repository. (2005). [Online]. http://www.ics.uci.edu/ mlearn /MLRepository.html.

Authors' Profiles

Essam Al Daoud received his BSc from Mu'tah university, MSc from Al Al-Bayt university, and his PhD in computer science from university putra malaysia in 2002. Currently, he is an associate professor in the computer science department at Zarqa university, Jordan. His research interests include machine learning, optimization quantum computation and cryptography**.**

How to cite this paper: Essam Al Daoud,"An Efficient Algorithm for Finding a Fuzzy Rough Set Reduct Using an Improved Harmony Search", IJMECS, vol.7, no.2, pp.16-23, 2015.DOI: 10.5815/ijmecs.2015.02.03