

Ant Colony Optimization based Feature Selection in Rough Set Theory

Majdi Mafarja, Derar Eleyan

Abstract—Feature selection is an important concept in rough set theory; it aims to determine a minimal subset of features that are jointly sufficient for preserving a particular property of the original data. This paper proposes an attribute reduction method that is based on Ant Colony Optimization algorithm and rough set theory as an evaluation measurement. The proposed method was tested on standard benchmark datasets. The results show that this algorithm performs well and competes other attribute reduction approaches in terms of the number of the selected features and the running time.

Keywords—Rough Set Theory, Attribute Reduction, Ant Colony Optimization

I. INTRODUCTION

FEATURE Selection (FS) is a NP-hard problem [1] that can be defined as the problem of determining a minimum reduct (subset) from the original set of features [2]. 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 which was proposed by Pawlak [3-4], has been used as a simple mechanism to determine a minimal reduct by the complete search which is based on locating all of the possible reducts and selecting the one with the lowest cardinality and the highest dependency. This process is a time-consuming procedure, and it is only effective for small datasets [5]. Hence, for the high-dimensional datasets, lots of research used meta-heuristics algorithms to determine better solutions for the attribute reduction problem instead of using the reduction method in rough set theory.

In literature, many meta-heuristic approaches have been applied to solve attribute reduction problem. For example, Jensen and Shen [5-6] 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 et al. [7] considered a memory-based heuristic of a tabu search to solve the attribute reduction problem in rough

set theory, Liangjun Ke et al. [8] proposed an ant colony-based approach, and a scatter search approach was introduced by Wang et al. [9]. The great deluge algorithm for attribute reduction was presented by Abdullah and Jaddi [10]; later, Jihad and Abdullah [11] proposed the composite neighbourhood structure, and Aradjy and Abdullah [12] 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 et al. [13].

In this paper, an attribute reduction mechanism that based on Ant Colony Optimization algorithm and rough set theory called (ACOFS) is proposed.

The remainder of this paper is organised as follows: Next section briefly discusses the main concepts of the rough set theory followed by the explanation of the Ant Colony Optimization for feature selection. 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.

II. THE ROUGH SET THEORY

The starting point of the rough set theory is the concept of indiscernibility [3]. Let an information system be $I = (U, A)$, where U is a non-empty set of finite objects called the universe of discourse and A is a non-empty set of attributes. For every attribute $a \in A$, a set of its values (V_a) is associated. Any subset P of A determines a binary relation $IND(P)$ on U , which is called an indiscernibility relation. If $(x, y) \in IND(P)$, then x and y are P -indiscernible. The relation $IND(P)$ can be defined as follows:

$$IND(P) = \{ (x, y) \mid U^2 \mid \square a \square P, a(x) = a(y) \} \quad (1)$$

The equivalence classes of the P -indiscernibility relation are denoted as $[x]_P$. The indiscernibility relation will be used to define the upper and lower approximations. For a subset $X \subseteq U$, the P -lower and P -upper approximations can be defined as follows:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\} \quad (2)$$

$$\overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\} \quad (3)$$

Majdi. Mafarja is with the Department of Computer Science, Faculty of Information Technology, Birzeit University, P.O. Box 14, Birzeit, Palestine. (phone: 00970-2982357; e-mail: mmafarja@birzeit.edu.

Derar Eleyan, with the Department of Computer Science, Faculty of Information Technology, Birzeit University, P.O. Box 14, Birzeit, Palestine. e-mail: deleyan@birzeit.edu.

TABLE I
PROPERTIES SAMPLE DATASET

| <i>U</i> | <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> |
|----------|----------|----------|----------|----------|
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 2 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 1 | 1 | 1 |
| 6 | 1 | 0 | 2 | 0 |

As in Table I, we can say that when considering a subset $B = \{a, b\}$, the objects 1, 2 and 3 certainly belong to a class in attribute $\{d\}$, which is indiscernible together with objects 4 and 6. Then we have the following:

$$U/IND(P) = \{ \{1,2,3\}, \{4,6\}, \{5\} \}$$

Let P and Q be equivalence relations over U , then the positive region can be defined as:

$$POS_P(Q) = \bigcup_{x \in U/Q} P_x \tag{4}$$

The positive region contains all of the objects of U that can be classified into classes of U/Q using the information in attribute P .

In the example dataset in Table I, let $P = \{a, b\}$ and $Q = \{d\}$. Then we have the following:

$$U/IND(P) = \{ \{1,4,5\}, \{2,3,6\} \}$$

$$POS_P(Q) = POS_{\{a,b\}}(\{d\}) = \bigcup \{ \{5\}, \emptyset \} = \{5\}$$

The positive region defined above can be used later to determine the rough set degree of dependency of a set of attributes Q on a set of attributes P , which is defined in the following way:

For $P, Q \subset A$, it is said that Q depends on P in a degree k ($0 \leq k \leq 1$), denoted as $P \Rightarrow_k Q$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \tag{5}$$

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

If $k = 1$, then we say that Q depends totally on P . If $0 < k < 1$, then we claim that Q depends partially on P , and if $k = 0$, then we say that Q does not depend on P .

In the example, the degree of dependency of attribute $\{d\}$ on attributes $\{a, b\}$ is:

$$\gamma_{\{a,b\}}(\{d\}) = \frac{|POS_{\{a,b\}}(\{d\})|}{|U|} = \frac{1}{6}$$

III. ANT COLONY OPTIMIZATION (ACO) ALGORITHM

The Ant Colony Optimisation (ACO) algorithm is a meta-heuristic algorithm that was initially proposed by Dorigo et al. [14]. The algorithm simulates the behaviour of real ants which, when searching for the shortest path to a food source, deposit pheromone as they travel; each ant prefers to follow the path that is rich in this pheromone. Ant Colony Optimisation mimics this pattern of behaviour by applying a

simple communication mechanism to enable the ant to find the shortest path between two points. In this paper, an Ant Colony Optimization for Feature Selection (called ACOFS) deals with each feature as a node, then it can be modelled as the problem of finding the path such that the selected features represent a reduct with minimized cardinality. The pheromone trail represents the weight of each edge between two features. The heuristic information we used is basis on the dependency degree.

Fig. 1 shows the pseudo-code of ACOFS. In each iteration (cycle) of the loop, a set of preliminary solutions are generated (each ant represents one solution). Then the pheromone levels are updated, which then results in further actions being taken (the solution construction process). Each solution is constructed by selecting the first feature randomly, and then the next feature is selected from those not selected features depending on a probability that given by Dorigo et al. [14].

$$p_{ij}^k(t) = \frac{\tau_{ij}^z \eta_{ij}^\beta(t)}{\sum_{l \in allowed_k} \tau_{il}^z \eta_{il}^\beta(t)}, \quad j \in allowed_k, \tag{6}$$

where k denotes the number of ants and t denotes the number of iterations, $allowed_k$ represents the set of not selected conditional features, $\tau(i,j)$ and $\eta(i,j)$ are the pheromone value and heuristic information of choosing feature j when at feature i . The construction process is terminated either if the cardinality of the solution is larger than cardinality of the minimal reduct obtained so far or if the fitness of the minimal reduct equals that for all conditional attributes.

After constructing each solution by an ant, the pheromone trails on each edge are updated according the following equation:

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}(t), \tag{7}$$

where $\tau_{ij}(t)$ represents the pheromone amount on a given edge (i,j) at iteration t . ρ ($0 < \rho < 1$) simulates the evaporation of pheromone, and $\Delta \tau_{ij}(t)$ is the amount of the deposited pheromone.

When a termination condition is reached the algorithm stops iterating. The termination condition may be a maximum number of iterations or a given time limit.

Algorithm: ACOFS

Begin

Population := generate initial solutions

initiate the pheromone trials s and calculate heuristic information

while (*the terminated condition is not reached*) do

for (each ant) do

construct a solution

update pheromone trails

end for

end while

end

Fig. 1 ACOFS Algorithm

IV. 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 nine UCI datasets, as shown in Table II. The effectiveness of the proposed approach is assessed through the number of attributes in each reduct and the run time from program start to termination. All parameters are considered as reported in [8].

TABLE II
UCI DATASETS

| Dataset | Objects | Attributes |
|---------------|---------|------------|
| Monk1 | 124 | 6 |
| Tic-tac-toe | 958 | 9 |
| Zoo | 101 | 16 |
| Vote | 435 | 16 |
| Lung | 32 | 56 |
| Soybean small | 47 | 35 |
| Dermatology | 366 | 34 |
| kr-vs-kp | 3196 | 36 |
| Audiology | 200 | 69 |

The performance of the proposed approach is evaluated by comparing its results with the results of IDRSFS [15], RSFSACO [16] and ARWSO [17]. The results are shown in Table III. The entries in this table represent the number of attributes in the minimal reducts that are obtained by each method. Each dataset is test for 20 times and the number in parentheses denotes the times of tests to achieve such an attribute reduct.

From Table III, we can see that ACOFS outperforms other approaches in Vote dataset, and outperforms IDRSFS in the Dermatology dataset. Overall, we can say that all algorithms have similar efficiency when they handle datasets with less than 16 features. However, for the datasets with more than 16 attributes, IDRSFS fails to find the minimal reducts as the other algorithms. This is likely due to IDRSFS's drawback not having heuristic information to search through the feature space for optimal solutions and premature convergence to a local optimum in the space.

The running time for each algorithm is also represented in Table III. It is clear that ACOFS is the fastest algorithm in finding the final results between the reported algorithms, despite the fact that ARWSO and RSFSACO initiate the solution based on the core attributes. This proves the efficiency of ACOFS.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a rough set feature selection approach based on Ant Colony Optimization (called ACOFS) is introduced. The performance of the proposed algorithm was tested on

standard benchmark datasets and comparison results were presented. The results showed that ACOFS performs better than the report approaches from the literature in some datasets and not worse in any dataset. The average running time of the 20 runs for each dataset is reported and it was clear that ACOFS could obtain the results faster than the other approaches. As a future work, the ACO will be enhanced by changing the solution construction mechanism by employing a heuristic algorithm and considering the core features as a start point.

REFERENCES

- [1] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," *Intelligent decision support—Handbook of applications and advances of the rough set theory*, pp. 311–362, 1992.
- [2] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers 1998.
- [3] Z. Pawlak, "Rough Sets," *International Journal of Information and Computer Sciences*, vol. 11, pp. 341-356, 1982.
- [4] Z. Pawlak, *Rough sets: Theoretical aspects of reasoning about data*: Springer, 1991.
- [5] R. Jensen and Q. Shen, "Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, pp. 1457-1471, 2004.
- [6] R. Jensen and Q. Shen, "Finding Rough Set Reducts with Ant Colony Optimization," presented at the Proceedings of the 2003 UK Workshop on Computational Intelligence, 2003.
- [7] A.-R. Hedar, *et al.*, "Tabu search for attribute reduction in rough set theory," *Soft Comput.*, vol. 12, pp. 909-918, 2006.
- [8] L. Ke, *et al.*, "An efficient ant colony optimization approach to attribute reduction in rough set theory," *Pattern Recognition Letters*, vol. 29, pp. 1351-1357, 2008.
- [9] J. Wang, *et al.*, "Scatter Search for Rough Set Attribute Reduction," 2009.
- [10] S. Abdullah and N. S. Jaddi, "Great Deluge Algorithm for Rough Set Attribute Reduction," in *Database Theory and Application, Bio-Science and Bio-Technology*. vol. 118, Y. Zhang, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 189-197.
- [11] S. K. Jihad and S. Abdullah, "Investigating composite neighbourhood structure for attribute reduction in rough set theory," presented at the Intelligent Systems Design and Applications (ISDA), 10th International Conference, 2010.
- [12] Y. Z. Arajy and S. Abdullah, "Hybrid variable neighbourhood search algorithm for attribute reduction in Rough Set Theory," presented at the Intelligent Systems Design and Applications (ISDA), 2010.
- [13] S. Abdullah, *et al.*, "A constructive hyper-heuristics for rough set attribute reduction," presented at the Intelligent Systems Design and Applications (ISDA), 2010.
- [14] M. Dorigo, *et al.*, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, pp. 29-41, 1996.
- [15] C. Bae, *et al.*, "Feature selection with Intelligent Dynamic Swarm and Rough Set," *Expert Systems with Applications*, vol. 37, pp. 7026-7032, 2010.
- [16] Y. Chen, *et al.*, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognition Letters*, vol. 31, pp. 226-233, 2010.
- [17] H. Fan and Y. Zhong, "A Rough Set Approach to Feature Selection Based on Wasp Swarm Optimization," *Journal of Computational Information Systems*, vol. 8, pp. 1037-1045, 2012.

TABLE III
EXPERIMENTAL RESULTS COMPARISON WITH IDSRFSFS, RSFSACO AND ARWSO

| Data | Min | IDSRFSFS | | RSFSACO | | ARWSO | | ACOFS | |
|---------------|-----|--|---------|--------------------------------------|---------|--------------------------------------|---------|--------------------------------------|---------|
| | | Reduct size | AvgTime | Reduct size | AvgTime | Reduct size | AvgTime | Reduct size | AvgTime |
| Monk1 | 3 | 3 | 478 | 3 | 310 | 3 | 302 | 3 | 60 |
| Tic-tac-toe | 8 | 8 | 3315 | 8 | 2917 | 8 | 2821 | 8 | 400 |
| Zoo | 5 | 5 | 997 | 5 | 616 | 5 | 580 | 5 | 89 |
| Vote | 9 | 9 | 10634 | 9 | 9283 | 9 | 8699 | 8 | 204 |
| Lung | 4 | 4 ⁽⁹⁾ 5 ⁽¹¹⁾ | 17282 | 4 | 13027 | 4 | 11983 | 4 | 181 |
| Soybean small | 2 | 2 ⁽⁸⁾ 3 ⁽¹²⁾ | 3941 | 2 | 3529 | 2 | 3216 | 2 | 67 |
| Dermatology | 6 | 8 ⁽³⁾ 9 ⁽⁴⁾ 10 ⁽¹³⁾ | 75166 | 6 | 56477 | 6 | 53132 | 6 | 421 |
| kr-vs-kp | 29 | 31 ⁽³⁾ 32 ⁽⁶⁾ 33 ⁽¹¹⁾ | 951935 | 29 | 852762 | 29 | 800287 | 29 | 3426 |
| Audiology | 13 | 16 ⁽²⁾ 17 ⁽⁵⁾ 18 ⁽¹³⁾ | 261029 | 13 ⁽⁷⁾ 14 ⁽¹³⁾ | 218332 | 13 ⁽¹⁸⁾ 14 ⁽²⁾ | 193796 | 13 ⁽¹⁸⁾ 14 ⁽²⁾ | 50000 |