A Rough Set outlier detection based on Particle Swarm Optimization

Misinem
[1] Faculty of Information Science and Technology,
National University of Malaysia
misinem_kur@ftsm.ukm.my

Azuraliza Abu Bakar, Abdul Razak Hamdan and Mohd Zakree Ahmad Nazri
[2]Center of Artificial Intelligence Technology
Faculty of Information Science and Technology,
National University of Malaysia
{ aab, arh, mzan}@ftsm.ukm.my

Abstract—**Outlier is strange data values that stand out from datasets. In some applications, finding outliers are more interesting than finding inliers in datasets, such as fraud detection, network system, financial and others. In this research, an algorithm is proposed to find minimum non-Reduct based on Rough set using Particle Swarm Optimization (PSO) for outlier detection. Like Genetic Algorithm (GA), PSO is also a type of optimization algorithm based on populations. It requires only simple mathematical operator and computationally inexpensive in terms of both memory and time. The experiment has been carried out to compute the performance between PSO and GA using 10 UCI datasets and 2 data networks. The comparisons shown that PSO has the ability to detect outliers, with inexpensive computation time compared to GA.**

Keywords: outlier detection, rough set, PSO

I. INTRODUCTION

Outlier is an observation that deviates so much from other observation as to arouse suspicion that it was generated by a different mechanism [1]. Outlier detection algorithms are useful in data mining application such as: fraud detection [2], network intrusion detection [3] and others [4]. Many techniques have been proposed in outlier detection, for example, He *et al.*, [5] proposed Local Search Algorithm (LSA) to detect the outlier from feasible solution based on optimization approach. Wroblewski *et al.*, in [6] used GA to produce proper attributes order and heuristic algorithm to find the minimal reduct. Although this method is much better than [5] but it increases the computation complexity. Shaari *et al.*, [7, 8] proposed a non-Reduct approach based on Rough set to detect an outlier. This approach find minimal attributes which are indiscern in attribute values as well as indiscern in decision attributes from the matrix. In this method, the finding minimum non-Reduct computation is based on GA. In this research, PSO is proposed to substitute GA algorithm. Like GA, PSO is also a type of optimization algorithm based on populations. It requires only simple mathematical operator and computationally inexpensive in terms of both memory and time.

The organization of this paper is as follows. In section 2 The Rough set reduction is explained briefly, in the section 3 the PSO algorithm for reduction is presented. In section 4, the experimental result and discussion are provided. Finally conclusions are made in section 5.

II. ROUGH SET REDUCTION

Rough Set Theory (RST) [9-11] provides a mathematical tool that has been introduced by Pawlak in 1980. The rough set philosophy is founded on the assumption that with every of universe of discourse we associate some information (data knowledge). Many applications found to be developed by RST such as: manufacturing [12], economic financial [13], medical [14] and networking [12-15]. Several notation important of RST which involve equivalence class, indicernibility matrix modulo, discernibility object and Reduct [16, 17].

Reduct is determined from the set of prime implicants of the discernibility function. Reducts do not contain redundant attributes. They are usually interesting attributes and are used in attribute selection process. The computation of Reduct can be used to represent Information system (IS) or Decision System (DS). Rough set reduction method has been used for feature selection [18, 19]. It is worth to find minimal reducts, which can generate more general decision rules and better classification quality of new sample. However, the problem of finding a minimal Reduct is NP-hard [20]. Hence, many approaches have been proposed for finding reducts, these includes discernibility matrices, indicernibility matrices, dynamic reducts [7, 20, 21].

Reduct is an important part of an IS which can discern all objects that are discernible by the original IS. The core contains the attribute that are dispensable to the discrimination of objects that is the attributes that are contained in all reducts. Given $\mathcal{A}= (U,A)$, an attribute a is said to be dispensable in $B \subseteq A$ if $IND(B) = IND(B - \{a\})$, otherwise the attribute is indispensable in *B*. Given an IS $\mathcal{A}= (U,A)$, let $B \subseteq A$, then A Reduct of B is a set of attributes $B' \subseteq B$ such that all attributes $a \subseteq B - B'$ are dispensable and $IND(B) = IND(B')$ [22].

Based on Reduct computation, a non-Reduct of B is defined as, there exist a set of attributes $B - B' \subseteq B$, such that all attributes $a \in B'$ are indispensable, and $IND(B - B') \neq IND(B)$. Non-Reduct can be defined as a non interesting set of attributes which is presumed to contain outlier's knowledge

[8]. A simple DS with distribution of equivalence class is shown in Table1. This table contains five classes and the number of objects for each class shown in rightmost column.

TABLE I. EXAMPLE OF EQUIVALENCE CLASS FOR A DECISION SYSTEM (DS)

Eq. Class	a	b	c	d	e	f	Decision	Num of objects
E1	3	3	0	0	0	0	0	2
E2	3	3	0	0	0	1	0	3
E3	2	2	0	1	2	1	1	5
E4	1	1	3	2	0	2	2	30
E5	0	1	0	2	2	2	3	100

Table 2 illustrated the computation of Reduct and non-Reduct based on DS. The first column in the table lists the five equivalence classes from *E1* to *E5*, each of which contains a number of objects from universe that are indiscernible by attribute *a* through *f.* A set of Reduct is as shown in third column and a set of non-Reduct is as shown in rightmost column in the table.

III. PSO ALGORITHM FOR NON-REDUCT COMPUTATION

A. *PSO Standard*

PSO is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [23]. This method finds an optimal solution by simulating social behavior of bird flocking. The PSO algorithm consist of a group of individuals named "particles". Each particle is a potential solution to an *n*-dimensional problem. The group can achieve the solution effectively by using the common information of the group (*gbest*) and the information owned by the particle itself (*pbest*). The particles change their state by "flying" around in *n*-dimensional search space based on information until relatively unchanged state has been encountered, or until number of maximum iteration is reached.

PSO is initialized with a population of particles. Each particle is treated as a point in an *n*-dimensional problem. The *i*-th particle is represented as $\boldsymbol{S}_i = (s_{i1}, s_{i2}, \dots s_{in})$. The best previous position of any particles is $\boldsymbol{P}_i = (p_{i1}, p_{i2}, \dots p_{in})$. The index of the global best is represented by $\boldsymbol{gbest}$. The velocity for particle i-th $\boldsymbol{V}_i = (v_{i1}, v_{i2}, \dots v_{in})$. The velocity of each particle is calculated using (Eq. 1):

$$\boldsymbol{V}_i^{k+1} = \boldsymbol{\omega}\, \boldsymbol{V}_i^k + c_1 r_1 \left(\boldsymbol{P}_i^{\,k} - \boldsymbol{s}_i^k\right) + c_2 r_2 \left(\boldsymbol{gbest}^k - \boldsymbol{s}_i^{\,k}\right)$$

where $\boldsymbol{V}_i^k$, $\boldsymbol{V}_i^{k+1}$, and $\boldsymbol{S}_i^{\,k}$ are the velocity vector, modified velocity vector , and positioning vector of particle i at generation *k,* respectively. $\boldsymbol{P}_i^{\,k}$ is the best position found by particle i and $\boldsymbol{gbest}^k$ is the best position found by the particle's neighbourhood or the entire swarm. c_1 And c_2 are the cognitive and social coefficients, respectively. $\boldsymbol{\omega}$ is the inertia weight, which is employed to control the impact of the previous history of velocities on the current velocity of each particle. r_1 and r_2 are two different random parameters.

B. *BinPSO Algorithm*

Signature Binary PSO (BinPSO) algorithm was also introduced by Kennedy and Eberhart to allow the PSO algorithm to operate in binary problem spaces [24]. It uses the concept of velocity as a probability that a bit (position) takes on 0 or 1. With $r_3 \sim U(0,1)$ and *sig*() is a sigmoid function for transforming the velocity to the probability constrained to the interval [0.0, 1.0] as follows (Eq. 2):

$$sig = \frac{1}{1+e^{-v_i^{k+1}}} \qquad (2)$$

In BinPSO, updating a velocity remains the same as the velocity in basic PSO [23]. However, the updating position is redefined by the following rule (Eq. 3):

$$\boldsymbol{S}_i^{k+1} \begin{cases} 1 & if\ r_3 < sig\left(\boldsymbol{V}_i^{k+1}\right) \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

TABLE II. REDUCTS AND NON-REDUCTS OF DS

Eq. Class	Conjunctive Normal Form (CNF)	Reducts	Conjunctive Normal Form (CNF)	Non-Reduct
E1	(avbvdvevf)^(avbvcvdvf)	{a,b,d,e,f}{a,b,c,d,f}	avbvcvdve	{a}{b}{c}{d}{e}
E2	(avbvdve)^(avbvdvf)	{a,b,d,e}{a, b,d,f}	avbvcvdve	{a}{b}{c}{d}{e}
E3	(avbvcvdvevf)^(bvdvf)	{b}{d}{f}	-	
E4	bvdvf	{b}{d}{f}	-	
E5	bvdvf	{b}{d}{f}	-	

with $sig \in [0, 1]$ r_3, is a quasi random number selected from a uniform distribution in [0.0, 1.0]. For velocity of 0, the sigmoid function returns a probability of 0.5, implying that there is a 50% chance for the bit to flip. In figure 1 depict the process for implementing the BinPSO reduction.

Given a decision table $T = (U, C, D, V, f)$ the set of condition attributes, C, consists of n attributes [25]. A space of n-dimension for reduction problem is set up. Accordingly, each particle's position is represented as a binary bit and mapped one condition attribute. The domain for each dimension is limited to 0 or 1. The value '1' means the corresponding attribute is selected, while '0' not selected. Each position can be decoded to a potential reduction solution, a subset of C.

During the search procedure, each particle is evaluated using the fitness function and then the new position of particles are calculated. According to the definition of Rough set Reduct, the reduction solution must ensure the decision ability is the same as the primary decision table and the number of attribute in the feasible solution is kept as low as possible. Hence, the heuristic algorithm [6] is applied. If the new particle's position is the feasible solution, then the number of '1' in it is calculated. The solution with the lowest number of '1' would be selected.

IV. EXPERIMENT RESULTS

Among results of the proposed approach are compared with GA on ten discrete UCI datasets and two real network datasets. For each comparison, 100 independent run have been perform by both algorithms and average performance is exhibited in terms of the mean value and the standard deviation. The parameter controls values for BinPSO are shown in Table 3 [26]. In this study, a decreasing inertia weight is used, based on Eq. (4):

```
Begin
  Initialize swarm; initialize the size of
    particles and other parameters
    Evaluate fitness function for each
       particle
  Loop
    For each particle
      Update pbest
      Update gbest
      Update velocity using (Eq.1 and
          Eq.2) and position Using (Eq.3)
    End for
      Applies Heuristic Algorithm
    Evaluate fitness function
  Until End_Condition (no. of iterations)
End
```

Figure 1. Algorithms based on PSO for non-Reduct

$$\omega^{k'} = \omega_{\max} - \left(\frac{\omega_{\max}-\omega_{\min}}{k_{\max}} \times k\right) \quad (4)$$

A large starting value of ω is used to initially accommodate more exploration, and is dynamically reduced to speed the convergence to the global optimum at the end of the search process [27].

Data characteristic for all data sets are shown in Table 4 that consists of number of attribute, record, and class. All data sets are sorted based on the number of attributes in ascending order. Number of class formed on the equivalence class for every data set.

The comparison results between GA and PSO for 12 data sets are reported in Table 5 and Figure 2. Two measurements to identify and detect outliers are Top ratio and Coverage ratio.

TABLE III. PARAMATER CONTROL BINPSO

PSO Parameter	Value
Cognitive factor, c_1	1.4
Social factor, c_2	1.4
Inertia weight, ω	0.4-0.9
Random values: r_1,r_2	[0, 1]
No. of particles	No. of implicants

TABLE IV. DATA CHARACTERISTIC

Data Sets	Attribute	No. of Record	No. of Class
IRP	5	150	19
COL	6	160	154
KDD1	7	306	162
ECO	8	335	222
BCE	10	699	459
GLS	10	213	149
HDE	14	270	242
CLV	14	302	292
ACC	15	690	690
ZOO	18	101	59
LYM	19	148	148
KDD2	42	2000	1060

TABLE V. COVERAGE RATIO, TOP RATIO, AND COMPUTATION TIMES FOR NON-REDUCT.

Data sets	Coverage Ratio (%)		Top Ratio (%)		Average Time (sec)		Standard Deviation	
	GA	BinPSO	GA	BinPSO	GA	BinPSO	GA	BinPSO
IRP	89.47%	89.47%	55.03%	55.03%	0.000	0.000	0.000	0.000
COL	8.86%	8.86%	4.38%	4.38%	0.604	0.015	0.009	0.039
KDD1	86.42%	86.42%	48.69%	48.69%	1.687	0.184	0.019	0.157
ECO	77.93%	77.93%	51.94%	51.94%	1.755	0.412	0.015	0.156
BCE	16.00%	16.00%	8.00%	8.00%	2.247	0.985	0.025	0.396
GLS	91.95%	91.95%	79.34%	79.34%	1.008	0.184	0.008	0.119
HDE	98.35%	98.35%	94.80%	94.80%	24.301	3.898	0.323	0.852
CLV	4.45%	4.45%	4.30%	4.30%	24.315	4.975	0.154	0.863
ACC	100.00%	100.00%	100.00%	100.00%	4.026	0.172	0.021	0.226
ZOO	6.78%	6.78%	3.96%	3.96%	0.132	0.082	0.007	0.156
LYM	4.05%	4.05%	4.05%	4.05%	8.525	3.275	0.053	0.127
KDD2	75.23%	75.23%	50.04%	50.04%	65.012	25.245	0.012	0.215

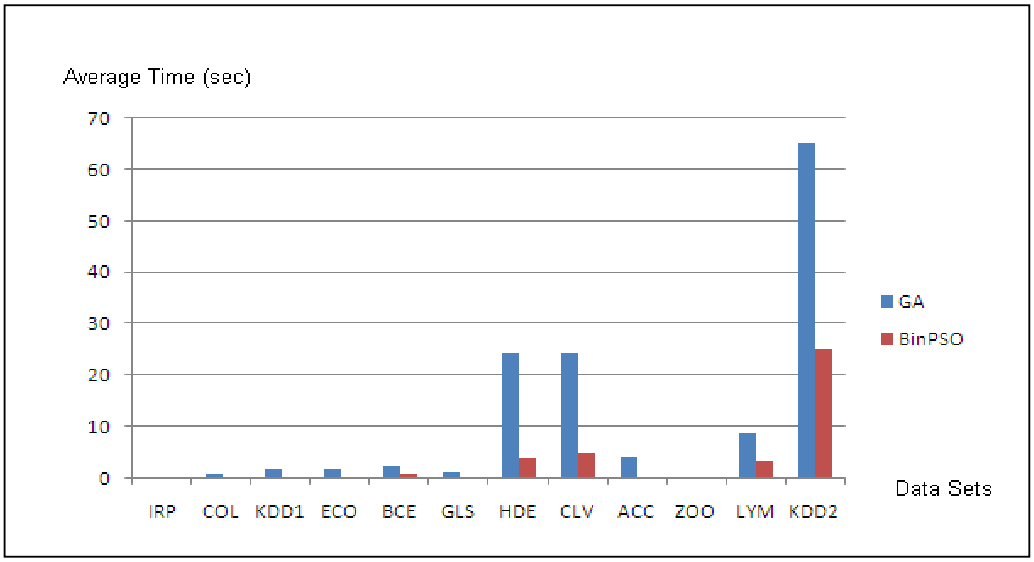

Figure 2. Computation times for non-Reduct

The Top Ratio is the one of measurement is applied during process of searching for outliers in dataset. The searching process began by identifying the top-*n* outliers from all objects in dataset, where *n* is equal to 1, 2, …, total number of instances in an IS. The top-*n* ratio determines the cut-off point in searching of outliers in dataset. The small value of top-*n* ratio indicates the shorter time in searching of outliers; hence indicate fast speed in detection of outliers.

The Coverage ratio is used to calculate the number of outliers that belong to a rare class. In searching outliers in dataset using top-*n* ratio, the coverage ratio checks if the object detected belong to a rare class. If it is true, then the number of outlier belonging to the rare class at the top-*n* ratio which is uncovered is the coverage ratio.

Although the experiment is conducted based on the different algorithms the results shown the same performance in Coverage ratio and Top ratio. That is, both algorithms used the same outlier detection method which is based on Rough set outlier factor (*RSetOF*) [8]. The result shown the BinPSO reached lower values in computation time for all 12 data sets, but has higher values for standard deviation. The lower values in computation time means the BinPSO is more efficient and faster in the process of finding minimum non-Reduct than GA. For example, LYM data set; need 3.75 sec to finding minimum non-Reduct for BinPSO algorithm and 8.525 sec for GA algorithm. The lower standard deviation for GA means that the GA algorithm consistently produces similar results in terms of computation time compared to the BinPSO.

V. CONCLUSION

This research proposed a new approach to reduce computation time in finding the minimal non-Reduct in Rough set theory by using PSO. The results from BinPSO algorithm were compared to results from GA algorithms. Although the results from both of algorithms have the same performance for Coverage ratio and Top ratio, but the BinPSO can generate the results with less computation time than GA. However, the proposed approach has to be improved and explored further. Future research will include a new outlier detection method based on minimum support classes.

REFERENCES

1. Hawkins, D.M.: Identification of outliers, Monograph on Applied Probability and Statistic, Reading. Chapman & Hall ,1980.
2. Bolton, R.J., Hand, D.J.: Unsupervised profiling methods for fraud detection. Credit Scoring and Credit Control VII, 1999.
3. Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.N.: Data mining for network intrusion detection. Cite seer , 2002, pp. 21-30.
4. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22**,** 2004, pp.85-126.
5. He, Z., Deng, S., Xu, X.: An optimization model for outlier detection in categorical data. Advances in Intelligent Computing ,2001, pp. 400-409.
6. Wroblewski, J.: Finding minimal reducts using genetic algorithms. second Annual Join conference on Information Sciences, Wrightsville Beach , NC, 1995, pp. 186-189.
7. Shaari, F., Abu Bakar, A., Hamdan, A.: On New Concept in Computation of Reduct in Rough Sets Theory. Rough Sets and Knowledge Technology ,2009, pp. 136-143.
8. Shaari, F., Bakar, A.A., Hamdan, A.R.: Fast outlier detection using rough sets theory. WIT Transactions on Information and Communication Technologies Vol 20, 2008.
9. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. Rough fuzzy hybridization: A new trend in decision-making ,1999, pp. 3–98.
10. Pawlak, Z.: Rough sets. International Journal of Parallel Programming 11**,** 1982, pp. 341-356.
11. Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics and systems 29, 1998, pp. 661-688.
12. Kusiak, A.: Rough set theory: a data mining tool for semiconductor manufacturing. IEEE Transactions on Electronics Packaging Manufacturing 24, 2001, pp. 44-50.
13. Tay, F.E.H., Shen, L.: Economic and financial prediction using rough sets model. European Journal of Operational Research 141**,** 2002, pp. 641-659.
14. Shaari, F., Bakar, A., Hamdan, A.: A Predictive Analysis on Medical Data Based on Outlier Detection Method Using Non-Reduct Computation. Advanced Data Mining and Applications , 2007, pp.603-610.
15. Cai, Z., Guan, X., Shao, P., Peng, Q., Sun, G.: A rough set theory based method for anomaly intrusion detection in computer network systems. Expert Systems 20**,** 2003, pp. 251-259.
16. Skowron, A.: Rough sets in KDD. In Proc.16th World Computer Congress IFIP, 2000, pp. 1–14.
17. Weiss, G.: Mining with rare cases. Data Mining and Knowledge Discovery Handbook , 2005, pp. 765-776.
18. Gupta, K., Aha, D., Moore, P.: Rough set feature selection algorithms for textual case-based classification. Advances in Case-Based Reasoning , 2006, pp.166-181.
19. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognition Letters 24 **,** 2003, pp.833-849.
20. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. Intelligent Decision Support ,1992, pp.311-362.
21. Zhang, J., Wang, J., Li, D., He, H., Sun, J.: A new heuristic reduct algorithm base on rough sets theory. Advances in Web-Age Information Management , 2003, pp.247-253.
22. Mollestad, T.: A Rough Set Approach to Data Mining Extacting a Logic of Default rules from Data. Department of Computer Science. The Norwegian University of Science and Technology Norwegia ,1997,
23. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks Vol. 4, IEEE Service Center, Piscataway ,1995, pp.1942-1948.
24. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm optimization. In: Proc.Conf. System, Piscataway,NJ ,1997, pp. 4104-4108.
25. Yue, B., Yao, W., Abraham, A., Liu, H.: A New Rough Set Reduct Algorithm Based on Particle Swarm Optimization. Bio-inspired Modeling of Cognitive Tasks, 2007, pp. 397-406.
26. Khalid, NK, Ibrahim, Z., Kurniawan, T.B., Khalid, M., and Engelbrecht. A.P., “Implementation of Binary Particle Swarm Optimization for DNA Sequence Design”, International Symposium on Distributed Computing and Artificial Intelligence (DCAI’09), 2009, pp. 450-457.
27. R.C. Eberhart, Y. Shi, Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, Proceedings of IEEE congress evolutionary computation, San Diego, CA, 2000, pp. 84–88.