Performance Analysis of Hybrid Swarm Intelligence Rule Induction Algorithm

C. NALINI[1]
P. BALASUBRAMNAIE[2]

[1]Department of Information Technology
[2] Departament of Computer Science and Engineering
Kongu Engineering College, Perundurai, India

[1]nalinikec@gmail.com
[2]p_balu@kongu.ac.in

Abstract. Data mining is used to extract potential information from data base. Rule induction is used to extract information from data base and display it in IF-THEN rule format. First the classification algorithm builds a predictive model from the training data set and then measure the accuracy of the model by using test data set.This work proposes a hybrid rule induction algorithm using Cooperative Particle Swarm (PSO) with Tabu search (TS), and Ant Colony Optimization (ACO). Real world data base consist of both nominal and continuous attributes. ACO based classification algorithms perform well in nominal data base. PSO based classification algorithms perform well in continuous data base where it converts nominal attributes into numerical values. In conventional PSO, there is no guarantee for local optimal solution. So, the proposed algorithm use tabu search in PSO to improve the search capability and integrate pheromone concept of ACO to handle real world classification problems. It uses cooperative concurrent PSO model to implement the algorithm and run two tasks simultaneously in parallel machines. The output of the work compares with the existing algorithm performance in several public domain data sets. The comparison results provide a evidence that: (a) The proposed algorithm is competitive with existing algorithm with respect to predictive accuracy; and the rule lists discovered by the algorithm are considerably simpler (smaller) than those discovered by the existing algorithm and (b) Reduce the execution time of the algorithm.

Keywords: Data mining, Ant colony Optimization, Coopertive model, Concurrent PSO.

(Received June 03, 2009 / Accepted August 11, 2009)

1 Introduction

Data mining is a multi-disciplinary field which aims to extract knowledge from databases. The work addressed the classification task of data mining. The aim of classification learning algorithm is to build a classifier model from a training set which help to predict the new records The discovered knowledge[19] is often represented in the form of IF (conditions) THEN (class) which has the advantage of representing a comprehensible model to the user. The predictive ability of the classification algorithm is typically measured by its predictive accuracy on the testing examples. However, these rules need to be simple and comprehensive. Otherwise, a human is unable to comprehend them.

2 Related work

2.1 Ant Colony Optimization

ACO based classification algorithm called Ant-Miner was developed by [13], the result shown to be it is very

competitive with the well-known C4.5 [15] and CN2 [2] classification-rule discovery algorithms. But Ant-Miner support only nominal attributes. It uses C4.5 discretization technique to discretize the continuous attribute into nominal attribute. A potential disadvantage of this approach is that less information will be available to the classifier since the discrete intervals have a coarser granularity which can have a negative impact on the accuracy of the discovered knowledge. It generates ordered rule set. In [8] the authors extended the Ant-miner algorithm for mixed variables. It uses Gaussian Probability Density Function (PDF) to create the range for numerical values instead of converting into nominal values. It generates an unordered rule set. It has a limitation. It takes time to normalize the pheromone values between continuous attributes and nominal attributes.

2.2 Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm was developed by [5].It is attracted by more researchers due to its simple concept. The algorithm has been successfully applied to several minimization optimization problems and neural network training. Particle swarm based classification data mining algorithms [17] are competitive with other evolutionary techniques and industry standard algorithm such as the J48, a Java implementation of C4.5 A standard Binary/Discrete PSO developed by [6], does not deal with categorical attributes in a natural fashion when compared to ACO. In standard PSO, binary bit string representation is used to encode the categorical attributes such as true or false.It was designed to cope with binary valued attributes.

It was designed to cope with binary valued attributes. It does not support multi valued categorical attributes. The authors [17] extended the standard binary PSO to cope with multi valued categorical attributes. The algorithm assign an index number to each value of a categorical attribute and then the index value is converted into a binary string. An extra bit is added to each attribute to decide if that attribute is to be included in the resulting rule or not. Once a categorical attribute has been converted into a binary string, the standard binary PSO can then be applied. This encoding approach introduces some problems. The interactions between the bits create an extra layer of complexity and confusion for the algorithm, because PSO algorithm finds the optimal value of each bit individually. The numerical index assigned to a categorical value, and the subsequent binary encoding scheme will affect the result of the particle interaction. There is no ordering in categorical attribute. In [3] the authors said that PSO is suitably challenged for two class problems, no clear conclusions can be drawn for problems with more than two classes.

In PSO, the particles are âflownâ through the search space by updating the position of the i^{th} particle at time stamp t according to the following equation:

$$X_i(t+1) = X_i(t) + v_i(t+1) \quad (1)$$

Where $X_i(t)$ and $V_i(t+1)$ are vectors to representing the current position and velocity respectively. The velocity updates are governed by the following equation:

$$v_i(t+1) = wv_i(t) + C_1 r_1(p_i - x_i(t)) + c_2 r_2(p_g - x_i(t)) \quad (2)$$

where $0w < 1$ is an interia weight determining how much of the particleâs previous velocity is preserved, and $c1$,$c2$ are two positive acceleration constants, $r1$, $r2$ are two uniform random sequences sampled from $U(0,1)$, p_i is the personal best position found by ith particle and pg is the best position found by the entire swarm so far. After some iterations, the personal best position of the particle will gradually move closer to the global best position. The particle will eventually converge on position of the global best particle. At this point, the particle will not be able to explore the search space, canât improve its solution. There is no guarantee that the position on which the particle has converged is a local minimum or global minimum.

2.3 Cooperative coevolution model

Coevolution can be classified into two group namely competitive coevolution and cooperative coevolution. While the former tries to make individuals more competitive through evolution, the latter aims to find individuals from which better systems can be constructed. Coevolutionary frameworks increase the efficiency of traditional evolutionary algorithms. The basic approach of cooperative coevolution is to divide a large system into many modules and evolve the modules separately. These modules are then combined again to form the whole system. The cooperative coevolutionary algorithms involve a number of independently evolving species that together form a complex structure for solving difficult problems. The fitness of an individual depends on its ability to collaborate with individuals from other species. The proposed algorithm handles multiple instances and run in parallel machines.

The particle swarm optimizer (PSO) is a stochastic, population-based optimization technique that can be applied to a wide range of problems, including neural network training. Most stochastic optimization algorithms including particle swarm optimizers (PSOs) and genetic

algorithms (GAs) suffer from the âcurse of dimensionality,â which simply put, implies that their performance deteriorates as the dimensionality of the search space increases. In [14] the authors partition the search space by splitting the solution vectors into smaller vectors. Each of these smaller search spaces is then searched by a separate GA; the fitness function is evaluated by combining solutions found by each of the GAs representing the smaller subspaces. Potter found that this decomposition lead to a significant improvement in performance over the basic GA. The main difference between the CPSO and implementation of the cooperative GA [14] is that the optimization process of a PSO is driven by the social interaction of the individuals within that swarm; no exchange of genetic information takes place. In contrast, the cooperative GA is driven by changes in genetic or behavioral traits within individuals of the populations.

The authors present a cooperative coevolutionary algorithm with improved performance on many benchmark functions in [14]. The approach was also successfully applied to applications like string matching and neural network design. In [7] the authors design a cooperative coevolution model to speed up the convergence of a fast evolutionary programming for solving large-scale problems with dimensions ranging from 100 to 1000. In evolutionary optimization the computational cost is measured in terms of time and hardware, it increases as the size and complexity of the problem increase. One approach to overcome such a limitation is to exploit the inherent parallel nature of EA(Evolutionary Algorithm) by formulating the problem into a distributed computing structure suitable for parallel processing i.e., to divide a task into subtasks and to solve the subtasks simultaneously using multiple processors. In [16] the authors defined four possible strategies to parallelize EAs, i.e., global parallelization, fine-grained parallelization, coarse-grained parallelization, and hybrid parallelization.

In global parallelization, only the fitness evaluations of individuals are parallelized by assigning a fraction of the population to each processor. The genetic operators are often performed in the same manner as traditional EAâs since these operators are not as time-consuming as the fitness evaluation. This strategy preserves the behavior of traditional EA and is particularly effective for problems with complicated fitness evaluations. The fine-grained parallelization is often implemented on massively parallel machines, which assigns one individual to each processor and the interactions between individuals are restricted into some neighborhoods. In coarse-grained parallelization, the entire population is partitioned into subpopulations. This strategy is often complex since it consists of multiple subpopulations and different subpopulations may exchange individuals occasionally (migration).

3 Problem definition

In PSO, when a particle discovers a good solution, other particles gather around the solution (gbest) too. Therefore they cannot escape from a local optimal solution. Consequently PSO cannot achieve global searches. In [12] the authors proposed a hybrid PSO/ACO Algorithm for discovering classification rules in Data Mining. The algorithm first generates a nominal rule and then adds continuous attributes with that rule in sequential order. The sequential approach takes more time and there is no interaction between rules. The proposed provide a solution for the above problem. It integrates Tabu search in PSO to explore the search space efficiently, and use cooperative coevolution computational model to create the interaction between swarms. It uses multiple swarms (or sub-swarms) for searching a solution. The swarms exchange information after some iteration. The proposed algorithm use parallel cooperative coevolution computational model.

4 Implementation

It uses concurrent PSO (CONPSO) to implement the algorithm. It uses two swarms concurrently for search a solution in the solution space. The swarms exchanged their global best solution. After every exchange point, the two swarms were to track the better global best found. The swarm1 handles nominal attributes and swarm2 handles continuous attributes. The work implements CONPSO in parallel machines to improve the performance of the existing algorithm and reduce the execution time.

4.1 Cooperative coevloution approach

First, vertically partition the given data into two groups (i.e) nominal and continuous attributes and then evaluate simultaneously by using individual swarms. The proposed algorithm uses two swarms. Swarm1 handles nominal attributes and swarm2 handles continuous attributes.

Initialize ruleset $RS = \Phi$
For each class C
PS=Training examples belonging to class C
Vertically partition PS into two (nominal attributes and continuous attributes)
Initialize the two swarms
iter_number=1, rule_index=1;
Repeat

Run the two swarms concurrently i.e
PSO/ACO & PSO
Send the best rule to Cooperative_module
IF the current trule is not same as previous rule
THEN
IF (Laplace corrected confidence of rule R > Rule threshold)
THEN
Add rule R to the rule set RS.
END IF
ELSE
increment rule_index value by one.
END IF
Iter_number=iter_number+1
UNTIL (rule_index $\leq userdefinedvalue \|$ iter_number $\leq$ Max Iterations
End For.
Order rule in RS by descending order.
Prune RS removing unnecessary terms/rules.
Cooperative_module( ) Begin Receive rules from two swarms and generate a
single rule R.
Prune the rule R based on Q value
Return (R)
End

4.1.1 Hybrid PSO and ACO algorithm

It uses pheromone updating in PSO for handling nominal attributes. Each particle in the population has a collection of n pheromone matrices (each matrix encodes a set of probabilities) where n is the number of nominal attributes in a data set. Each particle can be decoded probabilistically into a rule with a predefined consequent class. Each matrix has two entries stand for an off state and on state. If the off state is (probabilistically) selected then the corresponding (seeding) attribute-value pair will not be included in the decoded rule. If the on state is selected then the corresponding (seeding) attribute-value pair will be decoded in the rule. The attribute-value pair (term) is dependant on the seeding values. The particles are initialized by randomly selected examples. From that the particle decode a rule with attribute-values equal to the seeding terms, or to a rule without some or all those terms.
FOR $i = 1$ to no_of_iterations
REPEAT for Maxiterations
FOR every particle x in swarm1
Set Rule R_x = âIF Φ **THEN** Câ
FOR every dimension d in x
Use roulette selection to choose the state
END FOR
Calculate Quality Q_x of R_x
P = x's past best state
Q_p = P's quality
IF $Q_x > Q_p$ **THEN**
$Q_p = Q_x$
P = x
END IF
END FOR
FOR every particle x
P = x's past best state
N = the best state ever held by a neighbour of x according to Nâs quality QN
FOR every dimension d in x
IF ($P_d = N_d$) **THEN** Q_p increases pheromone entry corresponding to the value of N_d in the current x_d
ELSE IF ($P_d = off$ AND seeding term for x_d is not equal to N_d) **THEN** Q_p increases pheromone entry for the off state in x_d
ELSE Q_p decreases pheromone entry Corresponding to the value of Nd in the current x_d
END IF
Normalize pheromone entries
END FOR
END FOR
END REPEAT
RETURN best rule discovered R_1
END FOR

The output of this algorithm is like

IF A_{nom1} **AND** A_{nom2} **THEN** class C

where A_{nom1} and A_{nom2} are nominal attributes and C is a predicting class.

4.2 Pheromone updating

In [12] the authors said that the pheromone updating procedure is influenced by two factors, the best state of a particle x has ever held (P) and the best state ever held by a neighbor particle N. After update pheromone value, the algorithm will normalize the entries in pheromone matrix. If any entry has a value less than a predefined minimum pheromone value, then it is set to the minimum amount (0.01).

Next, check the amount of pheromone in the matrix becomes less than 1.0 as long as both entries have greater than zero amount of pheromone. In this case the algorithm normalizes both the entries. It is similar to Min-Max ant system. Table 1 shows the pheromone updating strategies.

Table 1: Different pheromone updating scenarios.

Term Seeding for x_d	P_d	N_d	Outcome for entries in x_d
value=w	(on) value=w	(on) value=w	on pheromone increased off off pheromone decreased
value=w	(on) value=w	(on) value<>w	on pheromone increased off pheromone increased
value=w	(on) value=w	off	On heromone decreased off pheromone increased
value=w	off	(on) value=w	On pheromone increased off pheromone decreased
value=w	off	(on) value<>w	On pheromone decreased off pheromone increased
value=w	off	off	On pheromone increased off pheromone decreased

4.3 Rule pruning procedure

Rule pruning procedure improves the quality of a rule by removing irrelevant terms from the rule antecedent and use to solve overfitting problem. The Laplace-corrected confidence is used to measure the quality of a rule.

Laplace corrected Confidence= $\frac{|term_{ij},k|+1}{|term_{ij}|+no-of-classes}$ (3)

RuleConfidenceThreshol= $\frac{MAX(0.4,|K|)}{|trainingset|}$ (4)

Where $|k|$ is the no of training cases belong to C and 0.4 is user-defined value.

The discovered rules are evaluated by using the concept:

If Laplace-corrected confidence is greater than Rule Confidence Threshold then the rule will be added to the rule set otherwise ignore the rule.

The rule set cleaning routine is used to clean the rule set. It follows the rule:

- There is a previous rule in the rule set that has a subset of the ruleâs attribute.
- If it predicts the same class as the default rule and is located just before it.

4.4 PSO with Tabu search â Continuous attribute

The output of the rule contains only continuous attributes. TS is used to improve the search capability of PSO. Tabu list prevents the cycling search. Here, Tabu search is used to improve global search solution.

The number of particles is defined by N and the number of dimension in the solution space is D. The variables count_P and count_G, tabu_P and tabu_G used to check the updated value of pbest and gbest. At initial stage, count_P and count_G are set to 0.

If pbest_ij and gbest_ij are not updated. Then the value of count_P and count_G will be incremented by one. Otherwise set to zero.

The update equation of velocity is different by the value of tabu_Pij and tabu_Gij:

(i) If tabu_Pij and tabu_Gij=0 then equ.(1) will be applied to update the velocity.

(ii) If tabu_Pij=1 and tabu_Gij =0 then equ.(5) will be applied to update the velocity.
$v_i(t+1) = vw_i(t) - c_3(p_i - x_i(t)) - c_2 rand(p_g - x_i(t))$ (5)

(iii) If tabu_Pij =0 and tabu_Gij =1 then equ.(6) will be applied to update the velocity.
$v_i(t+1) = vw_i(t) + C_1 rand(p_i - x_i(t)) - c_4 rand(p_g - x_i(t))$ (6)

(iv) If tabu_Pij =1and tabu_Gij =1 then equ. (7) will be applied to update the velocity.

$v_i(t+1) = vw_i(t) - C_3(p_i - x_i(t)) - c_4(p_g - x_i(t))$ (7)

$X_i(t+1) = X_i(t) + v_i(t+1)$ (8)

$c_3 = V_{max} exp\left(\frac{p_i - x_i(t)}{x_s pan}\right)$ (9)

$c_4 = V_{max} exp\left(\frac{p_s - x_i(t)}{x_s pan}\right)$ (10)

Step 1: Initialize each particle in Swarm2
Step 2: For each particle x in the swarm
FOR each continuous attribute d of x
Find the lowest and highest value of each
d from the training examples.
Find the range of each d
Set initial position
(i.e) Upper bound = the value of a randomly chosen seed
exampleâs d +range for that attribute
Lower bound= the value of a randomly Chosen seed

exampleâs d-range for that attribute
IF (seeding position is outside the range of the values seen in the data set)**THEN**
Set upper bound= Highest value
Set lower bound=Lowest value
END IF **END FOR Step 3**: Generate a rule (Rx) and calculate quality Q
Step 4: Find best position of a particle (Pi) and global best(Pg)
Step 5: Check the updating value of Pi and Pg .Based on the value apply equations (1)-(2) (5)-(9).
Step 6: Repeat step 1 to 5 until it reach the maximum iteration value.
Step 7:Return Rule(R2).

The output of this algorithm is like:
IF A_{cont1} **AND** A_{cont2} **THEN class C**

where A_{cont1} and A_{cont2} are continuous attributes and C is a predicting class.

The algorithm first finds the maximum and minimum value of each continuous attribute. Next, finds the range of each attribute. From that initialize the upper bound and lower bound of all attributes. Based on particles best position, add attributes to the rule. If the particle has same value for global best and local best then it apply TS to explore the search base.

5 Computational Results and Discussion

The performance of the algorithm was evaluated using eleven public-domain data sets from the UCI (University of California at Irvine) repository. The performance of the parallel implementation is evaluated against sequential approach. The existing algorithm PSO/ACO uses the excellence of Particle Swarm Optimization and Ant Colony Optimization to discover classification rules form data set. It first finds rules based on nominal attributes and then add continuous attributes in sequential order. But,the proposed algorithm use Cooperative frame work to implement the algorithm and also integrates tabu search in PSO to improve the search quality. It vertically partitions the attributes into two groups and then swarms are initialized.

The partitioning is used to solve cure dimensionality problem. Another enhancement is swarms are divided into two groups. Each group performs a separate task. The two swarms simultaneously execute the algorithm and exchanging their global best. From the result the work identified that the concurrent implementation reduces the execution time by 10%-25% , improves the accuracy of the algorithm, and generates comprehensible rule set. The algorithm adopted message passing interface technique to pass information between processors. Table 1, Tables 2, 3 and 4 illustrates the predictive accuracy and simplicity of the algorithm.

Studentâs t-test is used to compare the performance of the algorithm. Each entry in the Table 2 shows the average value of the accuracy obtained via the cross-validation procedure followed by the standard deviation. In all tables the bold entry indicates that the accuracy of the proposed algorithm was significantly greater than the accuracy of the existing algorithm for that dataset with the significance level $\hat{I}\pm = 5\%.The proposed algorithm generates s$

Table 2: Predictive Accuracy.

S.No	Data Set	PSO	HPSACO With coevolution
1	BreastCancer	72.62 ± 6.84	94.27 ± 2.05
2	Wiscosin Breast Cancer	93.42 ± 3.79	98.25 ± 1.01
3	Diabetes	72.67 ± 4.98	80.64 ± 1.95
4	Heart-c	77.38 ± 5.45	88.25 ± 5.75
5	Tic-Tac-Toe	100 ± 0	98.09 ± 0.97
6	Zoo	97.18 ± 6.25	98.25 ± 1.23
7	Iris	94.67 ± 5.26	99.25 ± 1.6
8	Mushroom	99.9 ± 0.11	98.75 ± 1.01
9	Crx	85.6 ± 2.84	92.65 ± 1.02
10	Ionosphere	88.06 ± 4.91	97.62 ± 1.05
11	Glass	70.95 ± 7.5	76.02 ± 3.21

Table 3: Rule Set Size.

S.No	Data Set	PSO	HPSACO With coevolution
1	BreastCancer	12.4 ± 2.27	10.2 ± 3.25
2	Wiscosin Breast Cancer	9.9 ± 1.6	6 ± 1.2
3	Diabetes	33.4 ± 1.43	12.67 ± 1.23
4	Heart-c	12.6 ± 0.84	8.6 ± 0.45
5	Tic-Tac-Toe	9.0 ± 0	11.7 ± 2.78
6	Zoo	7.1 ± 0.325	7.6 ± 1.37
7	Iris	3.0 ± 0	6.5 ± 1.65
8	Mushroom	8.7 ± 0.48	8.25 ± 1.78
9	Crx	22.5 ± 3.1	13.67 ± 1.43
10	Ionosphere	3.6 ± 0.97	5.8 ± 1.5
11	Glass	20.4 ± 1.35	15.35 ± 1.97

6 Conclusion

Classification is a predictive task of data mining. Classification algorithms build a model to predict the characteristics of new examples in the data base. The current work developed a new algorithm for generate an unordered rule set for mixed variables. The performance

Table 4: Rule Size.

S.No	Data Set	PSO	HPSACO With coevolution
1	BreastCancer	1.73 ± 0.26	1.5 ± 1.25
2	Wiscosin Breast Cancer	1.17 ± 0.09	1.02 ± 0.03
3	Diabetes	3.88 ± 0.29	1.54 ± 0.65
4	Heart-c	3.33 ± 0.19	2.7 ± 0.23
5	Tic-Tac-Toe	2.67 ± 0.0	2.5 ± 0.11
6	Zoo	1.14 ± 0.18	1.45 ± 0.12
7	Iris	0.93 ± 0.14	1.05 ± 0.06
8	Mushroom	1.86 ± 0.18	2.35 ± 0.43
9	Crx	2.94 ± 0.28	2.42 ± 0.63
10	Ionosphere	3.33 ± 0.79	1.4 ± 0.17
11	Glass	3.11 ± 0.18	1.97 ± 0.06

of the proposed algorithm is evaluated based on predictive accuracy, rule set generation, and speed.

From the results the work observed that the algorithm produce better accuracy results in six data sets. In the remaining data set, it produces the similar predictive accuracy. Even though it produces similar predictive accuracy, it generates comprehensible rule sets for more data sets. Moreover, execution time has been reduced since it handles both attributes simultaneously in parallel machines.

The work concluded that the proposed algorithm is comparative to existing algorithm with respect to predictive accuracy , comprehensibility and execution time. The concurrent implementation reduces the execution time by 10%-25%.It is suitable for all data sets. The performance of the proposed work enhanced the existing algorithm performance by using tabu search . In future, we are going to implement the same concept for other data mining tasks like association rule, clustering etc.

7 References

[1].Baskar.S, Suganthan. P. N.,A novel concurrent particle swarm optimization in Proc IEEE Congress on Evolutionary Computation, vol.1,pp.792â 796,2004.

[2].Clark, P.,Niblett,T, The CN2 rule induction algorithm , Machine Learning 3(4) pp 261â283, 1989.

[3].De Falco Della Cioppa. A, E.Tarantino,Facing classification problems with particle swarm optimization, Applied Soft Computing, vol. 7, no. 3, pp.652â658,2007.

[4].Dorigo, M., Stutzle, T.,Ant Colony Optimization. MIT Press, .2004.

[5].Eberhart.R.C. Kennedy.J, A New Optimizer using Particle Swarm Theory.In Proc.6th Symp.Micro Machine and Human Science, Nagoya,Japan pp39-43,1995.

[6].Kennedy.J Eberhart.R.C., Y. Shi,Swarm Intelligence, Morgan Kaufman Academic Press, San Francisco, CA, USA,2001.

[7].Liu .H, Yen.G.G,,Rank-density-based multi objective genetic algorithm and benchmark test function study, 1 IEEE Trans. Evol. Comput.,vol.7, pp.325â 343,2003.

[8].Nalini.C,Balasubramanie.P,Discovering unordered rule sets for mixed variables using an Ant-Miner algorithm,Data Science Journal,vol7,pp 76-77, 2008.

[9].Nalini.CBalasubramanie.P,Discovering unordered rule sets for mixed variables using Ant Colony Optimization, Journal of Computational Intelligence in Bioinformatics , Vol -1 No 2-3, pp 131-142,2008.

[10].Nalini.C Balasubramanie.P,Performance analysis of swarm intelligence classification algorithm, in the proceeding Genetic Evolutionary and Methods , USA,July 14-16,2009.

[11].Newman D.J., Hettich . S, Blake C.L.and Merz C.J. , UCI Repository of machine learning databases http://www.ics.uci.edu/ mlearn/MLRepository.html .
Irvine, CA: University of California, Department of Information and Computer Science,1998.

[12] Nicholas Holden Alex.A.F,A Hybrid PSO/ACO Algorithm for Discovering classification rules in Data Mining,Journal of Artificial Evolution and applications, Article ID 316145 doi:10.1155/2008/316145,2008.

[13].Parpinelli, R., Lopes, H., Freitas, A,Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation , pp 321,332, 2002.

[14]. Potter .M.A, De Jong.K.A, Acooperative coevolutionary approach to function optimization, in Proc. Parallel Problem Solving from Nature III Conf.,pp. 249â257, 1994.

[15]. Quinlan, J,: C4.5: Programs for Machine Learning. Morgan Kaufmann San Mateo, CA,1993.

[16].Rivera.W,Scalable parallel genetic algorithms, Artificial Intelligence Rev., vol.16, pp. 153â16,2001.

[17].Sousa.T Silva.A Neves.A ,Particle swarm based data mining algorithms for classification tasks,Journal on Parallel Computing vol.30, No. 5-6, pp.767â783,2004.

[18].Van den Bergh .F,An analysis of Particle Swarm Optimizers,Ph.D thesis University of Pretoria,2000.

[19].Witten, H ,Frank, E., Data Mining: Practical Machine Learning Tools and Techniques 2nd edn. Morgan Kaufman Academic Press, San Francisco, CA,USA, 2005.