# A NEW HEURISTIC FUNCTION IN ANT-MINER APPROACH

Urszula Boryczka and Jan Kozak
*Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland*

Keywords: Ant Miner, Ant Colony Optimization, Data Mining.

Abstract: In this paper, a novel rule discovery system that utilizes the Ant Colony Optimization (ACO) is presented. The ACO is a metaheuristic inspired by the behavior of real ants, where they search optimal solutions by considering both local heuristic and previous knowledge, observed by pheromone changes. In our approach we want to ensure the good performance of Ant-Miner by applying the new versions of heuristic functions in a main rule. We want to emphasize the role of the heuristic function by analyzing the influence of different propositions of these functions to the performance of Ant-Miner. The comparative study will be done using the 5 data sets from the UCI Machine Learning repository.

## 1 INTRODUCTION

Data mining is a process of extracting useful knowledge from real-world data. Among several data mining tasks – such as clustering and classification – this paper focuses on classification. The aim of the classification algorithm is to discover a set of classification rules. One algorithm for solving this task is Ant-Miner, proposed by Parpinelli and colleagues (Parpinelli et al., 2004), which employs ant colony optimization techniques (Corne et al., 1999; Dorigo and Stützle, 2004) to discover classification rules. Ant Colony Optimization is a branch of a newly developed form of artificial intelligence called swarm intelligence. Swarm intelligence is a form of emergent collective intelligence of groups of simple individuals: ants, termits or bees in which a form of indirect communication via pheromone we observe. Pheromone values encourage the following ants to build good solutions of analyzed problem and the learning process occurring in this situation is called positive feedback or auto catalysis.

The application of ant colony algorithms to rule induction and classification is a research area still not very good explored and tested. The appeal of this approach similarly to the evolutionary techniques are that they provide an effective mechanism for conducting a more global search. These approaches have based on a collection of attribute-value terms, then it can be expected that these approaches will also cope better with attribute interaction than greedy induction algorithms (Galea, 2002). What is more, these applications require minimum understanding of the problem domain; the main components are: the heuristic function and an evaluation function, both of which may be employed in ACO approach in the same shapes as in existing literature, concerning deterministic rule induction algorithms.

Ant-Miner is an ant-based system and it is more flexible and robust than traditional approaches. The application of ant colony algorithms to rule induction and classification is a research area still not very good explored and tested. This method incorporates a simple ant system in which a heuristic value based on entropy measure is calculated. Ant-Miner has produced good results when compared with more conventional data mining algorithms, such as C4.5 (Quinlan, 1993), ID3 and CN2 (Clark and Boswell, 1991; Clark and Niblett, 1989), and it is still a relatively recent algorithm, which motivates us trying to amend it. This work proposes some modifications to the Ant-Miner to improve it. In the original Ant-Miner, the goal of the algorithm was to produce an ordered list of rules, which was then applied to test data in order in which they were discovered. Original Ant-Miner was compared to CN2 (Clark and Boswell, 1991; Clark and Niblett, 1989), a classification rule discovery algorithm that uses a strategy for generating rule sets similar to that of heuristic function used in main rule of ants' strategy in Ant-Miner. The comparison was done using 6 data sets from the UCI Machine Learning repository that is accessible

Boryczka U. and Kozak J. (2009).
A NEW HEURISTIC FUNCTION IN ANT-MINER APPROACH.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 33-38
DOI: 10.5220/0001857700330038

at www.ics.uci.edu/˜mlearn/MLRepository.html. The results were analyzed according to the predictive accuracy of the rule sets and the simplicity of the discovered rule set, which is measured by the number of terms per rule. While Ant-Miner had a better predictive accuracy than CN2 on 4 of the data sets and a worse one on only one of the data sets, the most interesting result is that Ant-Miner returned much simpler rules than CN2. Similar conclusions could also be drawn from a comparison of Ant-Miner to C4.5, a well-known decision tree algorithm (Quinlan, 1993).

Outline. This article is organized as follows. Section 1 comprises an introduction to the subject of this article. In section 2, Ant Colony Optimization in Rule Induction is presented. Section 3 describes the modifications and extensions of original Ant-Miner. In section 4 our proposed modifications are shown. Then the computational results performed in five tests are reported. Finally, we conclude with general remarks on this work and further directions for future research are pointed out.

## 2 ANT COLONY OPTIMIZATION IN RULE INDUCTION

The adaptation of ant colony optimization to rule induction and classification is a research area still not good explored and examined. Ant-Miner is a sequential covering algorithm that merged concepts and principles of ACO and rule induction. Starting from a training set, Ant-Miner generates a set of ordered rules through iteratively finding an appropriate rule, that covers a subset of the training data, adds the formulated rule to the induced rule list, and then removes the examples covered by this rule as long as the stopping criteria is reached.

ACO owns a number of features that are important to computational problem solving (Freitas and Johnson, 2003):

- it is relatively simple and easy to understand and then to implement
- it offers emergent complexity to deal with other optimization techniques
- it is compatible with the current trend towards greater decentralization in computing
- it is adaptive and robust and it is enable to cope with noisy data.

There are many other characteristics of ACO which are really important in data mining applications. ACO in contrary to deterministic decision trees or rule induction algorithms during rule induction, tries to extenuate this problem of premature convergence to local optima because of stochastic element which prefers a global search in the problem's search space. Secondly, ACO metaheuristics is a population–based one. It permits the system to search in many independently determined points in the search space concurrently and to use the positive feedback between ants as a search mechanism (Parpinelli et al., 2002).

Ant-Miner was invented by Parpinelli et al. (Parpinelli et al., 2004; Parpinelli et al., 2002). It was the first Ant algorithm for rule induction and it has been shown to be robust and comparable with CN2 (Clark and Boswell, 1991) and C4.5 (Quinlan, 1993) algorithms for classification. Ant-Miner generates solutions in the form of classification rules. Original Ant–Miner has a limitation that it can only process discrete values of attributes.

**Algorithm 1:** Algorithm Ant-Miner.

```
TrainingSet = {all training examples};
DiscoveredRuleList = [ ]; /* rule list is initialized with an empty list */
while (TrainingSet > MaxUncoveredExamples)
    t = 1; /* ant index */
    j = 1; /* convergence test index */
    Initialize all trails with the same amount
    of pheromone;
    repeat
        Ant_t starts with an empty rule and
        incrementally constructs a
        classification rule R_t by adding one
        term at a time to the current rule;
        Prune rule R-t;
        Update the pheromone amount of all
        trails by increasing pheromone in the
        trail followed by Ant_t (proportional to
        the quality of R_t) and decreasing
        pheromone amount in the other trails
        (simulating pheromone evaporation);
        /* update convergence test */
        if (R_t is equal to R_t - 1)
            then j = j + 1;
            else j = 1;
        end if
        t = t + 1;
    until (t ≥ No_of_ants) OR
          (j ≥ No_rules_converg);
    Choose the best rule R_best among all rules
    R_t constructed by all the ants;
    Add rule R_best to DiscoveredRuleList;
    TrainingSet = TrainingSet - (set of
    examples correctly covered by R_best);
end while
```

A short review of the main aspects of the rule discovery process by Ant-Miner is run parallel to the description of Ant-Miner algorithm. Ant-Miner produces a sequential covering approach to discover a list of classification rules, by discovering one rule at a time until all or almost all the examples in the training set are covered by the discovered rules.

All cells in the pheromone table are initialized equally to the following value:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i}$$

where:

- $a$ – the total number of attributes,
- $b_i$ – the number of values in the domain of attribute $i$.

The probability is calculated for all of the attribute–value pairs, and the one with the highest probability is added to the rule. The transition rule in Ant-Miner is given by the following equation:

$$p_{ij} = \frac{\tau_{ij}(t)\cdot\eta_{ij}^{\beta}}{\sum_i^a \sum_j^{b_i} \tau_{ij}(t)\cdot\eta_{ij}^{\beta}}, \forall i \in I$$

where:

- $\eta_{ij}$ is a problem-dependent heuristic value for each term,
- $\tau_{ij}$ is the amount of pheromone currently available at time $t$ on the connection between attribute $i$ and value $j$,
- $I$ is the set of attributes that are not yet used by the ant,
- Parameter $\beta$ is equal to 1.

In Ant-Miner, the heuristic value is supposed to be an information theoretic measure for the quality of the term to be added to the rule. For preferring the quality is measured in terms of entropy this term to the others, and the measure is given as follows:

$$\eta_{ij} = \frac{log_2(k) - InfoT_{ij}}{\sum_i^a \sum_j^{b_j} (log_2(k) - InfoT_{ij})}$$

where the function $Info$ is similar to another function employed in C4.5 approach:

$$InfoT_{ij} = -\sum_{w=1}^{k} \left[\frac{freqT_{ij}^{w}}{|T_{ij}|}\right] log_2 \left[\frac{freqT_{ij}^{w}}{|T_{ij}|}\right] \quad (1)$$

where: $k$ is the number of classes, $|T_{ij}|$ is the total number of cases in partition $T_{ij}$ (the partition containing the cases, where attribute $A_i$ has the value $V_{ij}$), $freqT_{ij}^{w}$ is the number of cases in partition $T_{ij}$ with class $w$, $b_i$ is a number of values in the domain of attribute $A_i$ ($a$ is the total number of attributes). The higher the value of $InfoT_{ij}$ is, the less likely is that the ant will choose $term_{ij}$ to add to its partial rule. Please note that this heuristic function is a local method and it is sensitive to attribute interaction. The pheromone values assigned to the term have a more global nature. The pheromone updates depend on the evaluation of a rule as a whole, i.e. we must take into account interaction among attributes appearing in the rule. The heuristic function employed here comes from the decision tree world and it is similar to the method used in algorithm C4.5. There are many other heuristic functions that may can adapted and used in Ant-Miner. We can derive them from information theory, distance measures or dependence measures.

The rule pruning procedure iteratively removes the term whose removal will cause the maximum increase in the quality of the rule. The quality of a rule is measured using the following formula:

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg}\right)\cdot\left(\frac{TrueNeg}{FalsePos + TrueNeg}\right)$$

where:

- TruePos - the number of cases covered by the rule and having the same class as the one predicted by the rule,
- FalsePos - the number of cases covered by the rule and having a different class from the one predicted by the rule,
- FalseNeg - the number of cases that are not covered by the rule while having a class predicted by the rule,
- TrueNeg - the number of cases that are not covered by the rule which have a different class from the class predicted by the rule.

The quality measure of a rule is determined by:

$$Q = sensitivity \cdot specificity.$$

We can say that accuracy among positive instances determines sensitivity, and the accuracy among negative instances determines specificity. Now we take into account only the rule accuracy, but it can be changed to analyze the rule length and interestingness.

Once each ant completes the construction of the rule, pheromone updating is carried out as follows:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t)\cdot Q, \forall term_{ij} \in \text{the\_rule}$$

The amount of the pheromones of terms belonging to the constructed rule $R$ are increased in proportion to the quality of $Q$. To simulate pheromone evaporation $\tau_{ij}$, the amount of pheromone associated with each

$term_{ij}$ which does not occur in the constructed rule must be decreased. The reduction of pheromone of an unused term is performed by dividing the value of each $\tau_{ij}$ by the summation of all $\tau_{ij}$. The pheromone levels of all terms are then normalized.

# 3 FIRST MODIFICATIONS

The authors of Ant-Miner (Parpinelli et al., 2004; Parpinelli et al., 2002) suggested two directions for future research:

1. Extension of Ant-Miner to cope with continuous attributes;
2. The investigation of the effects of changes in the main transition rule:
   (a) the local heuristic function,
   (b) the pheromone updating strategies.

Recently, Galea (Galea and Shen, 2006) proposed a few modifications in Ant-Miner. Another modifications (Oakes, 2004; Martens et al., 2006) cope with the problem of attributes having ordered categorical values, some of them improve the flexibility of the rule representation language. Finally, more sophisticated modifications have been proposed to discover multi-label classification rules (Chan and Freitas, 2006) and to investigate fuzzy classification rules (Galea and Shen, 2006). Certainly there are still many problems and open questions for future research.

## 3.1 Data Sets used in our Experiments

The evaluation of the performance behavior of different modifications of Ant-Miner was performed using 5 public-domain data sets from the UCI.Please note that Ant-Miner cannot cope directly with continuous attributes (i.e. continuous attributes have to be discretized in a preprocessing step, using the RSES program *(logic.mimuw.edu.pl/˜rses/)*). In the original Ant-Miner and Galea implementation (Galea, 2002), the discretization was carried out using a method called C4.5-Disc (Kohavi and Sahami, 1996). C4.5-Disc is an entropy-based method that applies the decision-tree algorithm C4.5 to obtain discretization of the continuous attributes.

Both the original Ant-Miner and our proposal have some parameters. The first one – the number of ants will be examined during the experiments.

# 4 PROPOSED MODIFICATIONS

An Ant Colony Optimization technique is in essence, a system based on agents which simulate the natural behavior of ants, incorporating a mechanism of cooperation and adaptation, especially via pheromone updates. When solving different problems with the ACO algorithm we have to analyze three major functions. Choosing these functions appropriately helps to create better results and prevents stacking in local optima of the search space.

The first function is a problem-dependent heuristic function ($\eta$) which measures the quality of terms that can be added to the current partial rule. The heuristic function stays unchanged during the algorithm run in the classical approach. We want to investigate whether the heuristic function depends on the previous well-known approaches in the data-mining area (C4.5, CART, CN2) and can influence the behavior of the whole colony, or not. According to the proposition concerning the heuristic function (Liu et al., 2004), we also analyze the simplicity of this part of a main transition rule in Ant-Miner. The motivation is as follows: in ACO approaches we do not need sophisticated information in the heuristic function, because of the pheromone value, which compensates some mistakes in term selections. Our intention is to explore the effect of using a simpler heuristic function instead of a complex one, originally proposed by Parpinelli (Parpinelli et al., 2004), so we change the formula presented in the formula 1.

## 4.1 CART Influences

In the case of a method CART proposed by (Breiman et al., 1984), the value of $\mathrm{InfoT}_{ij}$ is determined according to the following formula 2.

$$InfoT_{ij} = 2 \cdot P_L \cdot P_P \cdot \sum_{w=1}^{k} |Pw_L - Pw_P| \qquad (2)$$

where:

- $P_L$ – a ratio of a number of objects in which the specific attribute $i$ has a value $j$ to all objects in a testable data set,
- $P_P$ – a ratio of a number of objects in which the specific attribute $i$ has not an analyzed value $j$ to all objects in a testable data set,
- $Pw_L$ – a ratio of a subset of objects belonging to the decision class $w$ in which the specific attribute $i$ has a value $j$ to all objects having the value $j$,
- $Pw_P$ – the ratio of a subset of objects belonging to the decision class $w$ in which the specific attribute $i$ has not a value $j$ to all objects having the value $j$.

## 4.2 CN2 Influences

In the case of a method CN2 proposed by (Clark and Boswell, 1991; Clark and Niblett, 1989), the value of $\mathrm{InfoT}_{ij}$ is calculated in the formula 3 (according the Laplace error estimate):

$$InfoT_{ij} = argmax\left(\frac{fregT_{ij}^{w}+1}{|T_{ij}|+k}\right) \tag{3}$$

where $w$ is a specific decision class, range from 1 to $k$.

## 4.3 Mixture of Modifications

Mixed methods proposed to determine the $\mathrm{InfoT}_{ij}$ values make use of early presented rules (early proposed C4.5, CART and CN2 influences). These mechanisms are used after the rule construction, alternately.

Experiments in this part of experimental study will be performed with a combination of following modifications: C4.5 + CART, C4.5 + CART + CN2, CART + CN2, C4.5 + CN2.

## 4.4 Results

For each experimental study the number of ants was established experimentally, separately for each of the testable data sets. For each data set we execute 100 times per experiment. Seven different modifications are analyzed separately for each testable data set.

### 4.4.1 Breast Cancer Data Set

In this experimental study we want to see whether the changeable method of calculation the $\mathrm{InfoT}_{ij}$ has an effect on the better performance in this case. Table 1 shows the results for 5 ants employed in this experiment with Breast cancer data set. The better results concerning the predictive accuracy (1,73%) and smaller values in the case of standard deviations.

### 4.4.2 Wisconsin Breast Cancer Data Set

From the study for Wisconsin breast cancer we observe the similar results as in a classical approach. We consider only one ant as a population size. The better results we can find in the case of separate C4.5 modification and the mixture of C4.5, CART and CN2 (the smaller value of standard deviations).

### 4.4.3 Dermatology Data Set

According to the proposition concerning different heuristic functions, we also analyzed the same effect in Dermatology data set as in previous one. Our investigation performed 40 ants and the predictive accuracy has the higher value for C4.5, CN2 and CART mixture of modifications. Slightly worth results we obtain in case of C4.5 approach (0,04%). In general, this data set is more resistant than the others in the context of effectiveness of our approaches. It can be seen that these modifications are similar to the original Ant-Miner approach in the context of accuracy.

### 4.4.4 Hepatitis Data Set

In the case of Hepatitis data set the analyzed modifications are slightly different from the Ant-Miner implementation. The standard deviations are higher than in other experiments. It is especially interesting that for Breast cancer and Hepatitis, the algorithms achieved the worse effectiveness. On contrary, in the case of Wisconsin data set we observe the good performance for all analyzed modifications. The best performance we observe in the mixture of C4.5 and CART approaches (only 0,27%).

### 4.4.5 Tic-tac-toe Data Set

We also observe a diminishing value of the accuracy in the case of Tic-tac-toe data set. In this situation, the question arises as to whether the loss of accuracy is due to the incorrect methodology or to the specific difficulty in the process of classification.

We also observed not very promising performance in this experimental study.

Table 1 shows the accuracy and standard deviations for rule sets produced in different approaches. It can be seen that in general these modifications are similar to the original Ant-Miner in the context of effectiveness. It can be intriguing aspect for future research to adjust specific features of data sets to the nature of methods used as a heuristic functions.

# 5 CONCLUSIONS

In this paper we examined different modifications concerning the heuristic function in Ant-Miner approach. The proposed modifications were simulated and compared for different data sets. The results showed that the proposed modifications were similar to the classical approach and they can preserve high value of predictive accuracy.

Finally, a lot of research works is still remaining in order to find a good strategy for matching the special heuristic function to the specific features of a data structure. We plan in the future to evaluate

Table 1: Comparative study. Accuracy of classification and standard deviation %.

| Dataset | Standard (C4.5) | CART | CN2 | C4.5 + CART | C4.5 + CART + CN2 | CART + CN2 | C4.5 + CN2 |
|---|---|---|---|---|---|---|---|
| Breast cancer | 72.07 (± 3.41) | 71.98 (± 2.19) | 72.11 (± 2.62) | **73.80 (± 2.34)** | 73.59 (± 2.27) | 71.78 (± 2.52) | 73.60 (± 2.41) |
| Wisconsin breast cancer | 92.15 (± 1.11) | 91.80 (± 0.90) | 91.78 (± 1.11) | 91.69 (± 1.38) | **92.16 (± 0.88)** | 91.69 (± 1.06) | 91.71 (± 1.07) |
| Dermatology | 93.75 (± 1.29) | 93.86 (± 1.32) | 93.62 (± 1.70) | 93.37 (± 1.67) | **93.90 (± 1.43)** | 93.79 (± 1.41) | 93.83 (± 1.76) |
| Hepatitis | 77.98 (± 2.97) | 77.34 (± 2.86) | 76.09 (± 3.45) | **78.25 (± 2.79)** | 77.90 (± 2.93) | 76.58 (± 2.72) | 77.81 (± 2.66) |
| Tic-tac-toe | 73.90 (± 1.81) | 72.45 (± 1.91) | 71.69 (± 1.81) | 73.77 (± 1.87) | **74.06 (± 1.89)** | 72.02 (± 1.77) | 73.55 (± 1.78) |

Ant-Miner modifications with large data sets and with several new modifications to better validation of our approach.

## REFERENCES


Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Belmont C.A., Wadsworth.

Chan, A. and Freitas, A. A. (2006). A new ant colony algorithm for multi-label alssification with applications in bioinformatics. In *Proceedings of Genetic and Evolutionary Computation Conf. (GECCO' 2006)*, pages 27–34, San Francisco.

Clark, P. and Boswell, R. (1991). Rule induction with CN2: some recent improvements. In *Proc. European Working Session on Learning (EWSL-91)*, pages 151–163, Berlin. Springer Verlag, LNAI 482.

Clark, P. and Niblett, T. (1989). The CN2 rule induction algorithm. *Machine Learning*, 3(4):261–283.

Corne, D., Dorigo, M., and Glover, F. (1999). *New Ideas in Optimization*. Mc Graw–Hill, Cambridge.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge.

Freitas, A. A. and Johnson, C. G. (2003). Research cluster in swarm intelligence. Technical Report EPSRC Research Proposal GR/S63274/01 — Case for Support, Computing Laboratory, Laboratory of Kent, Kent.

Galea, M. (2002). Applying swarm intelligence to rule induction. Master's thesis, MS thesis, University of Edingbourgh.

Galea, M. and Shen, Q. (2006). Simultaneous ant colony optimization algorithms for learning linguistic fuzzy rules. In Agraham, A., Grosan, C., and Ramos, V., editors, *Swarm Intelligence in Data Mining*. Springer, Berlin.

Kohavi, R. and Sahami, M. (1996). Error-based and entropy-based discretization of continuous features. In *Proc. 2nd Intern. Conference Knowledge Discovery and Data Mining*, pages 114–119.

Liu, B., Abbas, H. A., and Kay, B. M. (2004). Classification rule discovery with ant colony optimization. *IEEE Computational Intelligence Bulletin*, 1(3):31–35.

Martens, D., Backer, M. D., Haesen, R., Baesens, B., and Holvoet, T. (2006). Ants constructing rule-based classifiers. In Agraham, A., Grosan, C., and Ramos, V., editors, *Swarm Intelligence in Data Mining*. Springer, Berlin.

Oakes, M. P. (2004). Ant colony optimization for stylometry: the federalist papers. In *Proceedings of Recent Advances in Soft Computing (RASC — 2004)*, pages 86–91.

Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. (2002). An ant colony algorithm for classification rule discovery. In Abbas, H., Sarker, R., and Newton, C., editors, *Data Mining: a Heuristic Approach*. Idea Group Publishing, London.

Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. (2004). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation, Special issue on Ant Colony Algorithms*, 6(4):321–332.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco.