The Ant Colony Algorithms for Rule Induction
Janina Anna Jakubczyz
University of Economics, Wroclaw, Poland
Department of Artificial Intelligence Systems
Janina.Jakubczyz@ae.wroc.pl

Abstract
The data mining is still an open problem with many unsolved tasks. One of them is greediness of existing rule induction algorithms. Therefore there appear attempts enriching them with some global criterion that would control the search. One of such attempt is applying the mechanism of ant colony algorithms. The analysis of this approach is the subject of this paper.

Keywords: Machine Learning, Rule Induction, Ant algorithm.

1. Introduction
The existing rule induction algorithms are not perfect; therefore a search for better solutions is highly motivated. Generally speaking, these imperfections concern the computational complexity, and the quality of the results. One of the interesting research directions is the use of ant colony algorithms. These algorithms have been proven to be successful in solving complex computational problems in many domains (see e.g. [16], [14], [11], [10], [13]). Our discussion is based on the first works by Rafael Perpinelli, Lopes and Freitas ([19], [20], [21]) as well as the works by Michelle Galea [9].

2. The idea of ant colony algorithms
The ant colony algorithms constitute one of the two artificial intelligence techniques inspired by collective behavior. The confirmation of social insects’ success can be found almost everywhere [24]. The success is due to the organization of insect’s community, which is able to execute, among others: work division, specialization, collective control and construction of complex structures [18]. Theories of self-organization, created in the context of physics, chemistry and biology ([17], [2], [23], [9]) are the basis for describing social behavior of insects. They allow claiming that a complex behavior can result from interaction between specimen performing simple actions, which means that a single specimen solving complex tasks (problems) does not have to perform complex actions.

The rules governing insects’ communities may be different. The self-organization of social insects can be direct (e.g. maridibular; antennation, visual or chemical contact) or indirect, e.g. the environment of a nest directs ants’ reactions and the way they leave a trial of pheromone (see, among others, [6], [16]). The pheromone trial which is a way of environment modification is also a communication mechanism that forms a basis for models of collective intelligence. This pheromone tracing mechanism is a way of communication between ants following the trials. It turns out that after performing a task by some of the community members (e.g. nest cleaning – which can be understood as environment modification) lowers the need for performing that task (the environment modification), and the nest mates respond by not engaging in nest cleaning (they don’t clean the nest).

The ant colony algorithms use communication both on local and global levels. Locally – each ant leaves a trial, while globally – the trials leading to the best solution are reinforced.

The remainder of the paper is organized as follows: section 3 focus on the general characteristics of the rule induction algorithms, the next section gives overview of the application of the ant colony algorithms to the rule induction, section 5 includes the analysis of four basic techniques for rule induction, what makes the basis for the proposed EXS algorithm in section 7, and final section 8 includes conclusions and suggestions for future work.

3. The characteristics of the rule induction algorithms
Rule induction is a typical classification task ([12], [1], [3]): given the classes, the cost of incorrect classification and labeled examples, the question is to find a rule or rules that would enable to classify new examples correctly. The above is a complete definition of the classification task. The cost of incorrect classification is the only factor that can be omitted, and in most of the cases it does, also in the case being analyzed. The classification task can be named in different ways: image recognition, supervised learning, and discriminate analysis. In this particular case of rule induction, the solution is a classification rule or a set of such rules.

Most of the existing algorithms follow the schemas shown in Table 1. The differences lie in the way of creating the algorithm and in the choice element that can be either an attribute or a condition.
The CN2 and AQ algorithms by Michalski use the sequential covering approach, which refers to the way the classification rules are created ([3] p. 212). Generating the set of classification rules consists in building single rules sequentially, when each of the rules covers a subset of labeled examples from the training set. The covered examples are discarded from training set. The process is continued until every example in the training set is covered. The premise of a single rule should cover training examples from a single or a dominating category. The conditions for every next rule are chosen using an evaluation function based on the entropy of the training examples distribution in categories (CN2) or on the number of examples covered by a given condition (AQ).

A decision tree is built similarly – in an incremental way, by choosing an attribute for a tree node instead of single attribute/value conditions. A decision tree can be easily converted into a rule set ([3], [22]). Rules can be represented in three forms: decision trees, ordered decision lists, unordered decision lists.

Aside from the basic rule generation algorithm, the ancillary procedures are used. First of them concerns the overfitting of the rules created, which is due to the fact that algorithms are sensitive to errors in training examples. If such situation occurs, it is necessary to generalize rules in some way. The most often, rules are generalized according to the following principle: a shorter rule is accepted if it has a better or identical quality as the longer one. This process is called a global optimization of a rule set. Rules can be pruned also incrementally that is just after they are created [5].

The next ancillary procedure may concern attribute selection. It is possible to take into account all attributes available in a data set, to choose a subset of attributes according to a given criterion, or to create new, synthetic attributes by transforming or merging original ones [3].

The generated (acquired) rule set is evaluated using two measures. First of them is the predictive accuracy, which shows, whether the generated rule set classifies new examples correctly and to what extent. The second comprehensibility measure shows whether the generated rule set is intelligible by a human user, with reference to the number and to the complexity of rules.

The imperfection of rule induction algorithms discussed above concerns the nature of the searching process, which is called greedy or myopic. The process is determined by an incremental principle of rule construction, by the analysis of only one attribute at a time, by the choice – in each step – of the best condition that can be added to the rule’s premise. Therefore it would be advisable to find a mechanism that would allow for a global evaluation of choices and of a rule being created. The mechanism should also allow to make use of experience gained during the process of rule generating, which would govern the next condition’ choices. One of the proposals is the “leave a message” mechanism in ant colony algorithms.

### 4. The ant colony algorithms for rule induction

During the rule induction process, the ant colony algorithms combine two mechanisms. First of them is an inductive generation of classification rules set, second – a mechanism of sharing knowledge about the quality of solutions and about search progress. The CN2 algorithm has been modified (see Table 1). The aim of the modification was to find a balance between CN2’s local greedy search and ant colony algorithms’ global control. An ant colony algorithm of rule induction contains two kinds of procedures:

- a procedure of building/modifying solutions for the rule set creation problem,
- a procedure of modifying the pheromone trial.

The fundamental change lies in the nature of rule set building (modifying) process: instead of being deterministic, the process is stochastic. The process is characterized by a probability ($P_i$ described by formula (1)) of adding new condition to the rule currently being built. The probability comes from a domain-dependent heuristics ($\eta$ described by formula (2)) and the quantity of pheromone ($\tau$ described by formula (4)) left on the path by ants during previous actions. The modification of the pheromone trial is determined by a function, the value of which depends on aeration indicator and on the quality of solution created. Therefore, to implement an ant colony algorithm, one has to define, among others:

- an adequate representation of rule set creation, thanks to which an ant will be able to create (modify) a rule in an incremental way, using a transition function based on the quantity of pheromone on the path, and on a local heuristics;
- a heuristic function ($\eta$ given by formula (2)), that measures the quality of elements (terms-conditions) which can be added to current partial rule;
- a way of enforcing the building process of correct rules;
- a rule for pheromone ($\tau$ given by formula (4)) modifying;

### Table 1. The types of rule induction algorithms

<table>
<thead>
<tr>
<th>Creating the set of rules according to the “one rule a time” principle</th>
<th>Creating a rule according to the “top-down” principle</th>
<th>Creating a rule according to the “bottom-up” principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule=∅ Do: /* rule specialisation*/ Add the best condition to a rule, according to attribute evaluation Repeat until stop criterion is fulfilled</td>
<td>Rule = a conjunction of conditions Do: /<em>rule specialisation</em>/ Add the best condition to a rule, according to term evaluation Repeat until stop criterion is fulfilled</td>
<td>Rule = a bottom-up principle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision tree</th>
<th>Ordered list</th>
<th>Unordered list</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5 Quillian</td>
<td>CN2</td>
<td>AQ by Michalski</td>
</tr>
</tbody>
</table>

Source: based on [7], p. 11
a probabilistic transition principle \((P_{ij} \text{ given by formula } (1))\), that is based on the value of heuristic function \(\eta\) and the quantity of pheromone \(\tau\) on the path.

According to the algorithm described above, creating the classification rules set consists of the incremental creation of rule set and rules themselves. The choice of consecutive rule premises is determined by a probability of choosing a term \(i,j\), given by formula:

\[
P_{ij} = \frac{\tau_{ij}(t)\eta_{ij}}{\sum_{i=1}^{a} \sum_{j=1}^{b} \tau_{ij}(t)\eta_{ij} \forall i \in I}, \tag{1}\]

where:

\(\eta_{ij}\) – the value of heuristic function for term \(i,j\),
\(\tau_{ij}(t)\) – the amount of pheromone currently available at time \(t\) in the position \(i,j\) of the trial being followed by the ant
\(a\) – the total number of attributes,
\(b\) – the total number of values on the domain of attribute \(i\)
\(I\) – the attributes \(i\) not yet used by the ant

In contrary to the classic algorithm, the choice of the next premise’s condition is not deterministic, but depends on a probability. Another change lies in the fact, that the quality of terms is augmented by the experience gained during classification rules creation process. This experience is measured by the quantity of pheromone \(\tau_{ij}(t)\). The measure of predictive quality of a term is a well known definition based on entropy, given by formula:

\[
\eta_{ij} = \frac{\log_2 k - H(W | A_i = V_{ij})}{\sum_{i=1}^{a} \sum_{j=1}^{b} \log_2 k - H(W | A_i = V_{ij})}, \tag{2}\]

where:

\(x_{ij}\) – is set to 1, if the attribute \(A_i\) is available and 0 otherwise,
\(a\) – the total number of attributes,
\(b\) – the total number of values on the domain of attribute \(i\)
\(k\) – the number of classes
\(V_{ij}\) – j-th value belonging to the domain of \(A_i\)
\(A_i\) – i-th attribute
\(H\) is the heuristic function based on the entropy of a term which is defined by:

\[
H(W | A_i = V_{ij}) = -\sum_{w} (P(w | A_i = V_{ij}) \log_2 P(w | A_i = V_{ij})), \tag{2.1}\]

where:

\(W\) – the class attribute
\(k\) – the number of class labels
\(w\) – the value of the class attribute
\(P(w | A_i = V_{ij})\) – the empirical probability

The next step is to evaluate rule’s quality; it is needed for modifying the pheromone level (the knowledge acquired). The quality is represented by a symbol \(Q\) and is given by formula:

\[
Q = \text{sensitivity} \times \text{specificity} = \frac{TP}{TP + FN} \times \frac{TN}{FP + TN}, \tag{3}\]

where:

\(TP\) - (True Positives) the number of cases covered by the rule that have the class predicted by the rule;
\(FP\) - (False Positive) the number of cases covered by the rule that have a class different from the class predicted by the rule;
\(FN\) - (False Negative) the number of cases that are not covered by the rule but that have the class predicted by the rule;
\(TN\) - (True Negatives) the number of cases that are not covered by the rule and do not have the class predicted by the rule.

Now we have defined all the elements necessary to define the last one, namely the pheromone modification strategy, given by the following formula:

\[
\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}Q, \forall i,j \in R \tag{4}\]

The level of pheromone of all terms can be normalized by dividing it by a total amount of pheromone. The amount of pheromone increases for terms that appeared in the rule \(R\), and decreases for the others.

A schematic ant colony algorithm for rule induction is as follows:

1. Each ant begins with an empty rule set
   - it adds one condition at a time to a current partial rule\(^2\)

(The current form of a rule determines the path that an ant follows in a set of conditions)

Choosing a condition means choosing the path direction among all possible directions;

- the criterion of condition choice is determined by:
  - problem-dependent heuristic function, and
  - the amount of pheromone linked with the given condition (term)
- rule creation is complete if:
  - adding any other condition decreases a number of cases covered by the rule
  - condition set is empty
  - verification of the rule created

2. The pheromone is updated and a next ant begins work (Point 1), making use of its predecessors’ experience.

Table 2. The characteristics of AntMiner and Thesis algorithms.

<table>
<thead>
<tr>
<th>Element</th>
<th>AntMiner</th>
<th>Thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule quality threshold</td>
<td>Specified by 10</td>
<td>Unspecified</td>
</tr>
<tr>
<td>Stop criterion</td>
<td>Maximum number of uncovered cases - 10</td>
<td>Unspecified</td>
</tr>
<tr>
<td>Rule truncating</td>
<td>When quality after reduction is higher then current one</td>
<td>When quality after reduction is higher or the same as the current one, or even lower</td>
</tr>
<tr>
<td>Pheromone modification</td>
<td>After each rule created by an ant</td>
<td>After each iteration, according to the best rule in the last iteration</td>
</tr>
</tbody>
</table>

Source: Own elaboration based on [19], [20], [21], [8].

\(^2\) The current form of a rule determines the path that an ant follows in a set of conditions
We discuss two ant colony algorithms for classification rule induction: AntMiner by R. S. Parpinelli, H. S. Lopes and Freitas, described e.g. in [20], and an algorithm that we called Thesis, by M. Galea [8]. The above statements apply to both algorithms, and the differences between AntMiner and Thesis are presented in Table 2.

5. Analysis of the results

Below we present the list of the results obtained by using four different techniques for a classification task in standard, experimental databases 3. The first table contains a list of classification accuracy expressed in percents rounded up to integers, and a systematic error rounded up to integers, to make data more comprehensible.

### Table 3. The list of classification accuracy of rule induction algorithms

<table>
<thead>
<tr>
<th>Database</th>
<th>AntMiner</th>
<th>Thesis</th>
<th>C4.5</th>
<th>CN2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T O T O</td>
<td>T O T O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ljubljana Breast Cancer</td>
<td>75  11  73</td>
<td>8  73  3</td>
<td>68  4</td>
<td></td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>96  3  92  3</td>
<td>95  0  95  1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tic-Tac-Toe</td>
<td>73  8  74  4</td>
<td>83  2  97  0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dermatology</td>
<td>87  6  90  6</td>
<td>89  1  90  2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hepatitis</td>
<td>90  9  84  11</td>
<td>86  1  90  2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cleveland Hart Disease</td>
<td>60  8  56  8</td>
<td>58  1  57  2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

T - classification accuracy in %; O – systematic error in %.

<table>
<thead>
<tr>
<th>Database</th>
<th>Algorithm</th>
<th>The best</th>
<th>The worst</th>
<th>Maximal difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ljubljana Breast Cancer</td>
<td>AntMiner</td>
<td>CN2</td>
<td></td>
<td>7 (75)</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>AntMiner</td>
<td>Thesis</td>
<td></td>
<td>4 (96)</td>
</tr>
<tr>
<td>Tic-Tac-Toe</td>
<td>CN2</td>
<td>AntMiner</td>
<td></td>
<td>26 (97)</td>
</tr>
<tr>
<td>Dermatology</td>
<td>CN2</td>
<td>AntMiner</td>
<td></td>
<td>3 (90)</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>CN2</td>
<td>Thesis</td>
<td></td>
<td>6 (90)</td>
</tr>
<tr>
<td>Cleveland Hart Disease</td>
<td>AntMiner</td>
<td>Thesis</td>
<td></td>
<td>4 (60)</td>
</tr>
</tbody>
</table>

Maximal classification accuracy is given in parentheses

Source: Own elaboration based on [19], [20], [21], [8].

The comparison of classification accuracy of the four rule induction algorithms leads to the following conclusions:

- C4.5 is the most resistant and universal algorithm. Its classification accuracy for each of the databases was between the best and the worst results;
- AntMiner is the most unforeseeable algorithm, one can even say that its nature is ‘speculative’;
- Thesis is the algorithm that gives the worst classification accuracy the most often.

Generally speaking we may see the failure of the attempts to strengthen the rule induction algorithms with a mechanism of sharing the knowledge gained during the learning process. We advance a hypothesis that this failure was due to the construction of knowledge sharing mechanism. This construction concerns local choices directed by a global orientation. In our opinion the global choice of a direction is not in fact global, because it is done after a single rule is discovered, that is when the classification problem is solved only partially. This can lead both to the best and to the worst solutions. It will be rather an accidental than an intended operation. In contrary, in the case of genetic algorithms for example, the global evaluation expressed by a fit function concerns the complete solution. So is in the case of travelling salesman problem. Global control should concern the complete solution. Nevertheless, taking into account a sequential nature of the presented rule induction algorithms, it is not possible to apply such global control. While analysing classification accuracy, we can make another important observation: it is possible to improve classification quality. Undoubtedly it results from the fact, that such a measure as entropy and also other, e.g. statistical measures are not perfect. However including a global control of best solution searching into the algorithms, gave completely different results. If we omit the global control, which – in our opinion – is not global, the only difference would lie in the way of choice making that is whether the choice is deterministic or stochastic. Therefore we advance a hypothesis that if a deterministic choice is replaced with a stochastic one, the classification accuracy will not be significantly lower, and there is no need to include the ant colony algorithms mechanism. The influence of ants population size on the classification accuracy level is an interesting observation. It leads to a conclusion, that 10 ants are enough to cause a significant increase in quality (over 1% of accuracy; see [8], p. 64, table 7.3. (a) and (b)). An increase in population size does not result in a monotonic improvement of prediction quality. Prediction values are distributed among populations that differ in size. Four of the six databases gave better results when the population was increased. Therefore the author of the Thesis algorithm claims that it is purposeful to use an ant colony algorithm, but he also suggests that there should be more different local choices and less global control. In our opinion it isn’t necessary, we rather stress imperfections of the measures and the stochastic nature of the process of choosing the next condition in a rule’s premise. The classification accuracy criterion is additionally strengthened by a criterion concerning the intelligibility of a solution. This is determined by the number and the complexity of the rules in the set (the list) of rules acquired. The results of the algorithms run in different databases are presented in Table 5.

---

3 The databases can be found at: http://www.ics.edu/~mlearn/MLRepository.html

4 The worst results can be due to a less restrictive rule truncating criterion, in comparison to the AntMiner algorithm (see [8], p. 59).
The reason for which the Ant Miner algorithm is significantly better than others (a smaller number of rules and a smaller number of conditions in rule’s premises) may be due to the fact, that in the Thesis algorithm the threshold concerning the minimal number of cases covered by a rule is unspecified. Such statement is legitimate as the differences between algorithms are not large. On the other hand, the simplification of rules and of their set may be surprising. We may advance a hypothesis that one of the reasons for this situation is a stochastic choice of consecutive conditions in a premise. The second hypothesis may concern the moment at which verification (truncating) of rules is done. In C4.5 and CN2 verification is done after a rule set is created, while in AntMiner and Thesis it is done after generating each single rule. It is also worthy to examine, using a bigger number of databases, the relations between classification accuracy of acquired rule sets and their complexity.

6. The experiment

In the previous section we advanced a hypothesis, that if deterministic choices are replaced with stochastic ones, the classification accuracy will not worsen, without having to use the mechanism of the ant colony algorithms. The above hypothesis forms a basis for our experiment. We run iteratively the C4.5 algorithm, in which the quality of each element was measured by a probability calculated according to the formula (1), but omitting the pheromone quantity, that is according to a formula:

$$P_i = \frac{\eta_i}{\sum \eta_j \forall j \in I}$$  (5)

The number of iterations has been set by 10 for each rule, therefore to create a single rule ten algorithms’ runs were done. Such assumption comes from the analysis of how population size influences the classification accuracy. The analysis showed that 10 ants are enough to significantly rise the quality of the results. Next, the best rule is chosen. All the best rules together form the final set of classification rules. All the other parameters are the same as in the AntMiner algorithm – the stop criterion (no more conditions to add to a set or less than 10 cases uncovered); the level of rule quality – covering of minimum 10 examples; truncating after creation of every single rule.

The results of the experiment are presented in Table 6.

Table 5. The complexity of the acquired sets of classification rules.

<table>
<thead>
<tr>
<th>Database</th>
<th>AntMiner</th>
<th>Thesis</th>
<th>C4.5</th>
<th>CN2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>T</td>
<td>R</td>
<td>T</td>
</tr>
<tr>
<td>Ljubljana Breast Cancer</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Tic-Tac-Toe</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Dermatology</td>
<td>7</td>
<td>81</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Cleveland Hart Disease</td>
<td>10</td>
<td>16</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 6. The list of classification accuracy results of the rule induction algorithms.

<table>
<thead>
<tr>
<th>Database</th>
<th>AntMiner</th>
<th>Thesis</th>
<th>C4.5</th>
<th>CN2</th>
<th>EKS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>T</td>
<td>R</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Ljubljana Breast Cancer</td>
<td>75</td>
<td>11</td>
<td>73</td>
<td>8</td>
<td>73</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>96</td>
<td>3</td>
<td>92</td>
<td>3</td>
<td>95</td>
</tr>
<tr>
<td>Tic-Tac-Toe</td>
<td>73</td>
<td>8</td>
<td>74</td>
<td>4</td>
<td>83</td>
</tr>
<tr>
<td>Dermatology</td>
<td>87</td>
<td>6</td>
<td>90</td>
<td>6</td>
<td>89</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>90</td>
<td>9</td>
<td>84</td>
<td>11</td>
<td>86</td>
</tr>
<tr>
<td>Cleveland Hart Disease</td>
<td>60</td>
<td>8</td>
<td>56</td>
<td>8</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 7. The complexity of acquired sets of classification rules.

<table>
<thead>
<tr>
<th>Database</th>
<th>AntMiner</th>
<th>Thesis</th>
<th>C4.5</th>
<th>CN2</th>
<th>EKS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>T</td>
<td>R</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Ljubljana Breast Cancer</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Tic-Tac-Toe</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>83</td>
</tr>
<tr>
<td>Dermatology</td>
<td>7</td>
<td>81</td>
<td>7</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Cleveland Hart Disease</td>
<td>10</td>
<td>16</td>
<td>15</td>
<td>2</td>
<td>49</td>
</tr>
</tbody>
</table>

The degree of the rule set and rules’ premises complexity is slightly better than in the case of the C4.5 algorithm. However this improvement is definitely unimportant in comparison to the ant colony algorithms. Therefore we propose a further research that would allow verifying the hypothesis presented in previous section.

7. Conclusions
An ant colony algorithm may be considered a promising tool that could assist the avaricious algorithms for rule induction. Nevertheless, because of the difficulties in defining the global evaluation criterion, the ant colony algorithm cannot compete with the classic approaches. As the influence of a partial solution on the final one is not known, it is hard to accept the definition of the global criterion concerning these partial solutions. The analysis performed in previous sections confirms this opinion. The aim of the experiment described in the paper was to show that it is possible to improve classification accuracy without creating more complex algorithms.

The research carried out is an initial one. Further research will be devoted to the possibilities of defining a global evaluation criterion in sequential rule induction algorithms. We also plan to broaden the experiment, in order to find other, more profound reasons underlying such significant differences between the algorithms with "leave a message" mechanism and the C4.5, CN2 and AQ algorithms, concerning the complexity of rule sets.

### 8. References

8. M. Galea. Applying Swarm Intelligence to Rule Induction, MSc Artificial Intelligence, Division of Informatics, University of Edinburgh, 2002.