

Pace University

DigitalCommons@Pace

---

CSIS Technical Reports

Ivan G. Seidenberg School of Computer Science  
and Information Systems

---

9-1-2006

## Discovering effective, comprehensible data classification rules using hybrid evolutionary / Swarm models for data mining.

Bora Uran

Follow this and additional works at: [https://digitalcommons.pace.edu/csis\\_tech\\_reports](https://digitalcommons.pace.edu/csis_tech_reports)

---

### Recommended Citation

Uran, Bora, "Discovering effective, comprehensible data classification rules using hybrid evolutionary / Swarm models for data mining." (2006). *CSIS Technical Reports*. 142.  
[https://digitalcommons.pace.edu/csis\\_tech\\_reports/142](https://digitalcommons.pace.edu/csis_tech_reports/142)

This Thesis is brought to you for free and open access by the Ivan G. Seidenberg School of Computer Science and Information Systems at DigitalCommons@Pace. It has been accepted for inclusion in CSIS Technical Reports by an authorized administrator of DigitalCommons@Pace. For more information, please contact [nmcguire@pace.edu](mailto:nmcguire@pace.edu).

**TECHNICAL REPORT**

---

Number 230, September, 2006

Discovering Effective, Comprehensible Data Classification Rules  
Using Hybrid Evolutionary / Swarm Models for Data Mining

Bora Uran  
Michael L. Gargano

The present paper is reprinted from  
Congressus Numerantium  
Volume 176 (2005), pages 49-63

***Bora Uran*** (Gokbora Uran) received his Doctorate of Professional Studies in Computing from Pace University in 2005. His dissertation was titled “A Hybrid Heuristic Model for Classification Rule Discovery.” Professor Michael L. Gargano was his dissertation chairperson.

***Michael L. Gargano*** is Professor and quondam Chair of Computer Science at Pace University's New York City campus. His interests include artificial intelligence (in particular, neural networks and genetic algorithms), discrete mathematics (in particular, graph theory), computer theory, and business applications of quantitative computer models. Mike's Ph.D. is in combinatorial mathematics.

Professor Gargano has over ten years of real-world business experience and over twenty-five years of teaching experience (including computer science, mathematics, and management science). Mike enjoys working with students doing research, giving talks, presenting research results, and writing papers. Prominent among his research interests are applications of genetic algorithms and neural networks, societies of agents, logic graphs, and cryptology.

# **Discovering Effective, Comprehensible Data Classification Rules Using Hybrid Evolutionary / Swarm Models for Data Mining**

**Bora Uran      Michael L. Gargano**

**DPS/CSIS Pace University  
1 Martine Ave.  
White Plains, New York 10606-1909  
1 212 346 1687**

## **Abstract**

Classification rules reflect information that can be extracted from a database using data mining. We began by considering a hybrid (i.e., particle swarm, genetic algorithm, hill climber) model to evolve the rules. This paper studies hybrid heuristic models in the context of classification rule discovery. Nature inspired search algorithms such as Genetic Algorithms, Ant Colonies and Particle Swarm Optimization have been previously applied to data mining tasks, in particular, classification rule discovery. We extend this work by applying hybrid models that combine GA, PSO and/or hill climbers to the same type of classification tasks. Such models have already been tested and proved to be better than individual standalone search algorithms in various combinatorial optimization problems. Our research focused on investigating the same kind of potential performance enhancements in classification rule discovery tasks. We developed a model for a hybrid heuristic based classifier and implemented different variations of it in Java. These algorithms have been benchmarked against the well-known decision tree induction algorithm C4.5 using previously studied data sets in the literature. Results have been compared in terms of prediction accuracy, speed and comprehensibility. Our results showed that, heuristic based classifiers compete with C4.5 in terms of prediction accuracy on certain data sets and outperform C4.5 in general in terms of

comprehensibility. C4.5 always outperformed heuristic based classifiers in terms of speed due to the relative inefficiency inherent in heuristic based classification models. We also showed that hybridization of heuristics could bring improvements in terms of execution speed in comparison to plain standalone heuristic based classifiers.

**Keywords:** data classification rules, hybrid evolutionary models, life-cycle model

## 1. Heuristics and Data Classification Rules

Many real-world problems, like creating rules by data mining, are complex problems with a very high number of possible solutions. The size of such a search space prohibits an exhaustive search. Since searching for an exact solution using brute force for these types of problems will lead to computation times too high for practical purposes, approximate methods or heuristics are used. There is a practical trade off made between guaranteeing a best solution and finding a good solution in a significantly reduced time.

A meta-heuristic is a general-purpose heuristic method designed to guide the search for a good solution towards more promising regions of the search space. Effective and efficient exploration of the search space is the primary objective of meta-heuristics.

- Meta-heuristics are strategies that “guide” the search process.
- Their goal is to efficiently explore the search space in order to find optimal or near optimal solutions.
- Techniques that constitute meta-heuristic algorithms range from simple local search procedures to more complex learning processes.
- Meta-heuristic algorithms are approximate and usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- Meta-heuristics are not problem-specific.
- Meta-heuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by an upper level strategy.
- Advanced meta-heuristics use search experience (embodied in some form of memory) to guide the search.

The strategies of diversification and intensification are used in meta-heuristic search. Diversification refers to exploration of the different regions of the search space, whereas intensification limits the search based on accumulated search experience. The balance between these two is the key to the success of a meta-heuristic. Meta-heuristics are classified based on different properties of diversification and intensification strategies they used.

Local search based algorithms working on a single solution are called trajectory methods. They all share the property of describing a trajectory in the search space during the search

process. They start with an initial solution and follow a path based on the strategy defined in the algorithm, problem representation, and neighborhood structure. In the iterative local search algorithm a simple strategy of moving towards another potential solution in the neighborhood, provided that it's better than current solution is used. This approach is simple but ineffective in avoiding the local optima that may be far worse than other local optima in the search space. The main reason for getting stuck in local optima in a simple trajectory method is the strategy that always moves toward a better solution. So a simple technique to escape local optima would be moving to a solution even though it's not better than the current solution. This will lead to exploration of new regions of the search space, which would otherwise not be visited at all. This is exactly what the Stochastic Hill Climber algorithm does. Here, the decision to accept a potential solution from a neighborhood is based on a probability function. This makes moving to a solution that is worse than the current solution possible. One of the factors this probability depends on is the difference between the fitness values of the current solution and new solution. Thus, if the new solution is worse than the current solution, then there is a lower probability of accepting the new solution

As opposed to trajectory based methods that rely on a single solution in order to explore search space, population based methods manipulate a set of candidate solutions at every iteration. Improvement towards better solutions comes as result of competition and/or cooperation between the members of the population. Two population-based meta-heuristics will be reviewed below. Evolutionary Computation and Computational Swarm Intelligence.

Evolutionary Computation encompasses a set of algorithms that regenerates a population of solutions each iteration by applying operators inspired from the natural evolution process. Analogous to natural selection, a selection scheme that favors individuals with better fitness is used to guide the population towards more promising regions of the search. In addition to selection, the other two operators applied in different forms are crossover (also known as recombination or simply mating) and mutation.

According to introducers of the concept, Kennedy and Eberhard [15], Particle Swarm Optimization has its roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming. Kennedy and Eberhard link the PSO to the Adaptive Culture model in which the behavior of individuals is shaped by individual learning (that is, their own experience) and cultural transmission (that is, performance and information of individuals around them) [3]. The socio-cognitive theory that explains the behavior of an individual during the process of cultural adaptation can be explained in terms of three principles:

- Evaluate: the tendency to rate anything as positive or negative
- Compare: the tendency to use others as a standard for measuring themselves
- Imitate: the realization of the purpose of others' behaviors and the adoption of them (if appropriate.).

Based on the cultural adaptation model above, this movement is a function of current position and velocity, location of individual best success and best position found by a neighbor. Each iteration particle swarm optimization makes use of the velocity vector to

update the current position of each particle in the swarm. In this way, the position of each particle is updated based on the success of other individuals in their neighborhood. As an individual gets closer to its neighbor's best position, it may in turn perform better and influence its neighbors. This simulates the social behavior of a population of individuals, or a swarm, adapting to its environment. Alternatively, it is the "formation of a culture in a computational population" as it's suggested in [3] by Kennedy & Eberhart. The process is stochastic in nature and makes use of the memory of each particle as well as the knowledge gained by the swarm as a whole.

Similar to evolutionary algorithms, in PSO, a population of individuals that represent solutions moves through the search space trying to reach areas with good fitness by using probabilistic transition rules. But more than the similarities, the differences contribute to better understanding of PSO:

<u>EA</u>	<u>PSO</u>	<u>Hill Climber</u>
Individuals replaced	Individuals move	One individual moves
Position changes	Velocity changes	Position changes
No memory of direction	Velocity as memory of direction	No memory of direction
Natural selection	Self-organization	Local landscape
Interaction by crossover throughout the population	Interaction by cultural transmission within a neighborhood	No Interaction

Each of the meta-heuristics mentioned above, has relative strengths and weaknesses. For example population-based methods are better in exploration of the search space but intensification can be more effective in hill-climbers. Over the years, researchers have developed new algorithms by hybridizing different meta-heuristics in order to get a combination of the benefits of their relative strengths. Talbi[13] proposed a taxonomy of the hybrid meta-heuristics based on algorithm design.

In a low level hybridization a particular feature of one meta-heuristic is replaced by a feature of another meta-heuristic. Whereas in a high-level hybrid model each individual meta-heuristic operates in a self-contained manner. Relay versus teamwork distinction addresses the type of interaction between the meta-heuristics. In a relay hybrid model, individual heuristics run one after another by using output from previous one. In the case of teamwork hybridization, parallel agents performing different meta-heuristics run at the same time in the search space.

#### Four Classes of Hybrid Meta-heuristics

Low Level Relay	Low Level Team Work	High Level Relay	High Level Team Work
--------------------	------------------------	---------------------	-------------------------

The hybrid model applied to classification rule discovery in this thesis is inspired from life cycle model developed by Løvbjerg. Based on classification provided by Talbi above, the Life Cycle model is a high-level, relay hybrid of particle swarm optimization, genetic algorithms and stochastic hill climbers. The model uses a simple self-adaptive transition relay method between heuristics in order to improve performance. It has been applied to various combinatorial optimization problems and delivered better results in certain cases.

```

program LifeCycle Model
begin
    initialize
    while (not terminate-condition) do
    begin
        for (all individuals)
        evaluate fitness
    switch LifeCycle stage if no recent improvement
    for (PSO particles)
        calculate new velocity vectors
        move
    for (GA individuals)
        select new population
        recombine population
        mutate population
    for (HillClimbers)
        find possible new neighboring solution
        evaluate fitness for the new solution
        shift to new solution with probability p
    end
end

```

Data mining is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from the data in large databases. With the advent of technologies which facilitates collection and processing of large amounts of data by businesses, scientific institutions and governmental agencies, data mining has emerged as a field with a lot of practical applications in different areas such as marketing, fraud detection, biomedical research, banking and so on. As a part of a larger framework, Knowledge Discovery in Databases (KDD), data mining relates to fields of machine learning, statistics, visualization and databases.

Classification, clustering and association are the typical and widely used data mining tasks. Classification is the process of assigning each item of a data set, or records of a database, to a class from a predefined set of classes. The first step of classification is the learning the rules from a data set which contains instances with known classes. This type of learning is called supervised learning. In contrast, learning process in clustering is unsupervised. Clustering is the process of discovering classes in the data by partitioning records into clusters. Association rules find relations between different attributes of data records. In order to perform these and other data mining tasks, a large number of



algorithms have been developed in different research fields such as machine learning, statistics and neural and fuzzy computing.

First we will provide details of classification rule discovery and a well-known classification algorithm, C4.5 decision tree induction. Algorithms developed as part of this study have been benchmarked against C4.5. Then a brief survey of research on heuristics based classification rule discovery algorithms has been provided. The output of classification rule discovery can be represented in the forms of decision tables, decision trees or classification rules constructed as IF-THEN propositions. Decision tables are the most rudimentary form of the output representation. It simply lists different attribute values and the corresponding class in the rows of a table.

Many different mechanisms for classification rule discovery have been developed by researchers. Decision tree induction, Bayesian classification and neural networks are a few of them. These methods have relative advantages or disadvantages for different type of problems in terms of following common measures used in evaluating classification methods:

- Predictive accuracy
- Speed and Scalability - Time to construct / use the model
- Robustness - Handling noise and missing values
- Interestingness
- Goodness of the rules - Decision tree size / Compactness of the rules

Meta-heuristics have been used to construct efficient models that are capable to find high-quality solutions for many problems with large search spaces. This, combined with the fact that classification rule discovery is a search problem, the solution to which is a set of combinations of logical conditions constructed by values of attributes and a prediction of class value, that needs to be sought in a large search space [23], motivated researchers to apply heuristic methods from evolutionary algorithms and swarm intelligence to classification rule discovery. In [24] a detailed analysis of the benefits and the limitations of genetic algorithms (applicable to other population based heuristics as well) is given and compared to greedy local search based and rule induction algorithms such as C4.5.

The two primary benefits of genetic algorithms as a rule discovery method are their ability to perform a very thorough search of the space and flexibility in using a fitness function which can be fine tuned to problem specific aspect of the search. The second advantage is related to better handling of attribute interaction. Execution speed is the primary disadvantage of genetic algorithms in classification rule discovery. This is due to fact that population based heuristics evaluates large number of rules which constitute the population for the heuristic at each iteration which alters the population via small changes to its members. The other two disadvantages suggested in [24] are the randomness during initialization of the population as well as search process and tendency to focus on good solution once it is found.

Genetic Algorithms, in relation to classification rule discovery or supervised concept learning, have been studied by many researchers and applied to many different classification problems. [25] provides a comprehensive survey of the research on evolutionary algorithms, in particular Genetic algorithms and Genetic Programming, for data mining and knowledge discovery. Ant Colony Optimization has also been studied in the context of classification [28] and clustering [29]. [19] is the only Particle Swarm Optimization application of classification rule discovery that we are aware of so far.

[25] also discusses various aspect of implementation of evolutionary algorithms to classification rule discovery. Among them, individual representation and fitness function are the most critical from the perspective this thesis and for this reason they will be the subjects of the next two sections of this chapter. The first question that needs to be answered in relation to individual representation is how many rules should be encoded into a single individual in the population. There are two approaches for this. In the Michigan approach each individual consist of a single rule whereas in the Pittsburgh approach multiple rules make up an individual. The Pittsburgh approach is more suitable for classification tasks due to its ability to evaluate entire rule set together and as a result of this to take rule interaction into consideration. However it leads to longer individuals and, because of this, higher cost of computation. It also needs modification to operators of the heuristic due to complex and an in some cases variable-length individuals of the population. The Michigan approach has the advantage of lower computational cost due less complex individuals but evolution of one rule at time and disregard of rule interaction due to this is a major drawback, Another issue with this approach is that heuristic is converges to a single individual even though a set of rules is needed. One way to deal with this is running the algorithm multiple times to discover different rules, but this increases the computational cost.

The fitness function for the classification rule discovery is usually chosen based on the objectives of the classification process and form of the rule representation used. Maximizing predictive accuracy is usually the primary objective and predictive accuracy can be simply calculated as the ratio of correctly classified instances to total number of instances in the training set. [26], for example, suggests the following as a fitness function which provides a non-linear bias toward correctly classifying instances:

$$\text{Fitness}(\text{individual } i) = (\text{percentage of correctly classified instances})^2$$

However this function may not be sufficiently penalizing the incorrect classification caused by the individual rules and this may cause a performance problem especially when the rules are represented with Michigan approach, in which rule interaction can not be addressed within individuals. In order to address this, different variations of fitness functions which consider not only true positives and true negatives (correctly classified instances) but also false positives and false negatives.

Predictive Accuracy is not the only factor in the evaluation of classification rules. Comprehensibility and interestingness can also be factored into the fitness function. Comprehensibility or simplicity can be measured in different ways depending on the problem domain and user preferences. Number of attribute tests in the antecedent of the rule is a common measure. Measuring interestingness is usually more complex. [27], for example, suggest a method that calculates information gain from each attribute in the rule

antecedent based on weights assigned by users. As a part of the research, a number of variations of a hybrid heuristic based classification algorithm have been implemented and benchmarked on various data sets.

The experiment phase consisted of implementation of algorithms within the framework of Weka data mining software and benchmarking them with data sets used in previous data mining studies. The primary factor that determines the quality of a rule set is number of instances correctly classified (true positives). But in the Michigan approach individual rules are evaluated independently rather than entire rule set. This makes the number of false positives (instances classified incorrectly) another important factor. As the individual rules of the rule set are applied in the order they are added to the rule set, a high number of false positives will negatively affect the quality of the subsequent rules. [19] suggested the following fitness function which takes, not only true positives, but false positives and false negatives into consideration :

$$\text{Fitness} = TP / (TP + FN) * TN / (FP + FN)$$

TP – True Positives: number of instances the rule classified correctly.

FP – False Positives: number of instances the rule classified incorrectly.

TN – True Negatives: number of instances the rule “not classified” correctly.

FN – False Negatives: number of instances the rule “not classified” incorrectly.

This fitness function has been used in the heuristics of all algorithms tested. Increasing the penalization of false positives has also been experimented by simply increasing weight of false positives.

The core heuristic model used in all algorithms is inspired from the Lifecycle model. Based on the taxonomy provided by Talbi [13], they are mainly high-level, relay hybrids of Particle Swarm Optimization, Genetic Algorithms and stochastic hill climbers. Different forms of this basic model were tested by changing the relay sequence or by leaving out one of the heuristics. Rule representation was another key factor that led to testing of more variations. Benchmarking has been done between hybrid models, plain standalone heuristic based classifier and C4.5 decision tree induction algorithm. The core rule discovery function of the algorithms tested, used combinations of three heuristics:

- Discrete PSO, a variant of particle swarm optimization defined as a “model of binary decision” [3]
- Genetic algorithm – a basic model [1]
- Stochastic Hill Climber as defined in [1]

Stochastic hill climber had been included in number of tests, but after observing its contribution to be minimal, it has been excluded from most of the cases and primary focus has been on PSO/ GA combinations.

The core discovery algorithm starts with initialization of population for one of the heuristics. Then this population is altered with operators of that heuristic. After a certain number of iterations without any improvement in the best fitness value, the population is converted to the population of the next heuristic. And alteration of population continues for another set of iterations until a certain number of iteration without any improvement is reached. This cycle goes on until a termination condition is met. This may be an overall

limit on number of iterations or number of cycles of the heuristic sequence. In most of cases tested here, the loop has been terminated after the first cycle.

At the end, the global best, which was updated each iteration throughout the execution of the heuristics, is returned as the rule.

Procedure core rule discovery

Begin

Get number of PSO particles / Initialize each PSO particle

Get number of GA members / Initialize each GA members

Calculate fitness values for particles and GA members

Identify the particle and member with best fitness

repeat

repeat

for each particle

calculate velocity of the particle

move the particle its new location

end

Calculate fitness values for each particle

Identify the particle with best fitness

Until best fitness is not improved for N iterations

Convert particles to GA members

repeat

perform Selection on GA population

perform Crossover on GA population

perform Mutation on GA population

Calculate fitness values for each member of GA population

Identify the member with best fitness

Until best fitness is not improved for N iterations

Convert each member of GA population into a hill climber

Perform the hill-climbing algorithm on each hill climber

Replace best fitness if a hill-climber reaches to better fitness

Convert the hill climbers to PSO particles

until a certain number of iterations

return the member or particle or hill climber with the best fitness as the rule

End

## 2. Results of Experiments

The results given in this section provide accuracy and runtime comparisons between J4.8 and four different heuristics based classifiers on four different data sets. All heuristics were run long enough to converge to an optima. These optima and times have been identified as a result of multiple experimental runs with a different number of iterations each time. In the case of hybrid heuristics, the results provided below reflect the best time and accuracy combinations reached on each data set. All results for heuristic based classifiers are the averages of 5 runs.

Results for Zoo data set

	Accuracy/Execution Time		Size of Rule Set
J4.8	92%	0.08 seconds	17/9
GA	90 %	0.62 seconds	8/6
PSO	89%	0.71 seconds	7/6
PSO + GA relay Hybrid	88%	0.80 seconds	7/6
GA + PSO relay Hybrid	88%	0.68 seconds	7/6

For the artificially created zoo data set, smallest data set tested, J4.8 outperforms heuristic based classifiers. GA classifier is the closest to J4.8 both in terms of both speed and accuracy. Although time differences for this very small and simple data set are very insignificant, it still reflects behavior which is much more apparent in Breast-cancer and waveform data sets: GA approaches to better regions of the search space faster than PSO.

Results for Breast Cancer data set

	Accuracy/Execution Time		Size of Rule set
J4.8	76 %	0.21 seconds	4/6
GA	75 %	1.78 seconds	4/5
PSO	75%	2.21 seconds	4/6
PSO + GA relay Hybrid	75%	4.21 seconds	4/6
GA + PSO relay Hybrid	76%	2.18 seconds	4/6

The results from the breast-cancer data also shows that the GA converges faster but to less than optimal solutions. Here GA+PSO hybrid provides the best result in terms of accuracy and time among the heuristics.

Results for Wisconsin Breast Cancer data set

	Accuracy/Execution Time		Size of Rule set
J4.8	95 %	0.17 seconds	27/14
GA	94 %	3.40 seconds	7/7
PSO	95%	3.5 seconds	7/7
PSO + GA relay Hybrid	94%	3.86 seconds	7/9
GA + PSO relay Hybrid	95%	3.63 seconds	7/7

Some observation can be done on results from waveform data set. GA is by far converges faster and GA+PSO hybrid also converges to better solutions better or at least as good as other heuristic based classifiers.

In terms of rule comprehensibility, in most of the tests, heuristic based classifiers outperformed J4.8. This is especially clear with the more complex waveform data set.

Results of Waveform data set

	Accuracy/Execution Time		Size of Rule set
J4.8	75 %	6.99 seconds	330/659
GA	57 %	77.00 seconds	6/18
PSO	71%	96.20 seconds	7/31
PSO + GA relay Hybrid	71%	90.60 seconds	7/32
GA + PSO relay Hybrid	72%	78.00 seconds	7/32

### 3. Discussion of the Results

The results from our experiments fall into two different categories. We can study the performance of hybrid heuristic classifiers in comparison to plain heuristic classifiers or we can compare between heuristic based classifiers improved with hybridization to decision tree induction algorithms.

When we looked at the results in terms of predictive accuracy, no significant differences between hybrid classifiers and other heuristic classifiers were observed. In most cases all heuristics, using the rule representation based on the Michigan approach, converge to or very close to a global optimum. One exception to this is the GA classifier's significantly worse performance in comparison to both the PSO and hybrid classifier with the waveform data set. Since this is a data set with difficult to learn concepts and with a higher number of attributes and instances, it is more complex than the other data sets. In the tests with this data set, the GA classifier converged faster but to sub-optimal solutions.

More generally, this tendency of GA based classifiers, converging to a solution faster, can be observed with other data sets as well. In all tests the GA classifier has been the fastest among the heuristic based classifiers. PSO takes longer to converge. This can be explained by the fact that PSO brings smaller improvements to the existing members, and unlike GA, no members are displaced from the population. While GA's faster convergence does not create a significant disadvantage in the less complex search spaces, it is definitely penalized in more challenging ones, as observed in the tests with waveform data set.

This brings us to the most significant contribution of the hybridization observed during this study. Since the GA reaches to more promising regions of the search space faster and

the PSO intensifies subsequently, the GA+PSO hybrid reaches the predictive accuracy of PSO faster than the plain PSO based classifier. This explanation is consistent with the fact that the same type of improvement is not observed with the PSO+GA hybrid. In this case, the PSO is slower to make the initial exploration and the GA cannot effectively intensify to improve the results of the PSO, if the PSO iteration is terminated before it converges to the optimum solution.

When we look at comparison of the heuristic based classifiers with the C4.5 tree induction, rule comprehensibility arises as the most striking advantage of heuristic classifiers. The fact that C4.5 is a greedy algorithm that searches through the complete search space, leads to the generation of very large decision trees. Whereas the heuristic based classifiers intensifies on more promising regions with the ability of converging to simpler rules with similar accuracy.

In terms of predictive accuracy, C4.5 and heuristic based classifiers usually return comparable results. While in the smaller data sets the heuristic based classifiers perform as good as C4.5, in the waveform data set performance of C4.5 was better. This can be due to the fact that the heuristic based classifiers cannot beat the greedy C4.5 in this complex search space. Another factor that should be considered is the relative disadvantage of the simple, user defined discretization technique we used, as opposed to C4.5's built-in mathematical information gain based discretization technique, which is a form of a supervised discretization. In addition, the Michigan approach, the rule representation technique used in our algorithms, has an inherent deficiency in handling interaction between rules. This also works against the heuristic classifiers in terms of predictive accuracy. Discussion of this has been presented in Chapter 3. Longer execution time is the most significant disadvantage of non-hybrid heuristic based classifiers as observed in our tests as well as in previous studies [24]. The main reason for this is the high number of iterations needed to converge to a good solution and the computationally expensive fitness calculation that requires the processing of all instances in the data set each iteration. So we consider the speed improvement brought by the hybridization an important contribution of this study.

#### **4. Summary of the Results & Future Work**

- Heuristic-based classifiers' prediction performance is comparable to C4.5 decision tree induction algorithm. Speed is the major drawback when using this classifiers. Rule comprehensibility, however, which is measured via rule set size, can be considered an advantage in some situations.

- GA-PSO high-level relay hybrid provides faster execution times than the plain standalone heuristic based algorithms, and closes the gap in favor of hybrid classifiers.

- Teamwork hybrids did not show any improvement over relay hybrids and slightly longer execution times have been observed.

- Hybridization of hill-climbers does not help in terms of predictive accuracy and negatively affects the speed.

During this study, we focused on rule representation with the Michigan approach (that is, the representation of single rules in every individual of the population) mainly due to its simplicity. Drawbacks of this approach have been discussed earlier. The Pittsburg approach (that is, representing multiple rules in one individual) can bring improvements in terms of predictive accuracy. This will require handling more complex individuals and making significant modifications to the operators used in the heuristics. Another rule representation related opportunity for future study is, the analysis of usage of different encoding techniques, such as real values, instead of binary encoding. Finally, benchmarks on different data sets and domain specific modification of operators can also be included in any future work.

## 5. Acknowledgements

We wish to thank Pace University's School of Computer Science and Information Systems for partially supporting this research. We also wish to thank Dr. Maheswara Prasad Kasinadhuni for his invaluable help.

## 6. References

- [1] Michalewicz, Z., Fogel, D. B.: *How to Solve It: Modern Heuristics*, Springer, 2002.
- [2] Fortnow, L., Homer, S. *A Short History of Computational Complexity*, Bulletin of the European Association for Theoretical Computer Science - Computational Complexity Column (<http://theorie.informatik.uni-ulm.de/Personen/toran/beatcs/>) Article 80 6 / 2003.
- [3] Kennedy, J. and Eberhart, R.C., *Swarm Intelligence* Morgan Kaufmann Publishers 2001.
- [4] Blum, C., Roli, A., *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison* ACM Computing Surveys, Vol. 35, No. 3, September 2003, pp. 268–308.
- [5] Russell, S.J., Norvig, P., *Artificial Intelligence A Modern Approach* 2nd Edition Prentice Hall, 2003.
- [6] Hoos, H.H., N., Stutzle, T., *Stochastic Search Algorithms*  
[www.intellektik.informatik.tu-darmstadt.de/~klausvpp/GAILS/L/S/SLS.pdf](http://www.intellektik.informatik.tu-darmstadt.de/~klausvpp/GAILS/L/S/SLS.pdf)
- [7] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial Intelligence through Simulated Evolution*, Wiley, New York, 1966.
- [8] Rechenberg, I. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, 1973.
- [9] Holland, J. H., *Adoption in Natural and Artificial Systems*, The University of Michigan Press, Ann Harbor, MI, 1975.



- [10] Koza, J.R., Bennett, F.H., III, Andre, D. and Keane, M.A. *Genetic Programming III: Darwinian Invention and Problem Solving*, San Francisco: Morgan Kaufmann, 1999.
- [11] Whitley, D. *A Genetic Algorithm Tutorial*, Statistics and Computing, vol. 4, pp. 65–85, 1994.
- [12] Bonabeau, E., Dorigo, M. And Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999
- [13] Talbi, E.G., *A Taxonomy of Hybrid Metaheuristics* Journal of Heuristics, 8:541-564 2002 Kluwer Academic Publishers 2002.
- [14] Krink, T. and Løvbjerg, M. *The LifeCycle Model: Combining Particle Swarm Optimization, Genetic Algorithms and Hill Climbers*, In: Proceedings of Parallel Problem Solving from Nature VII (PPSN-2002), p. 621-630
- [15] Kennedy, J. and Eberhart, R. *Particle Swarm Optimization* from Proc. IEEE Int'l Conf. on Neural Network, Perth, Australia, 1998.
- [16] Banzhaf, W. and Feldkamp, U., *Computational Intelligence – Introductory Lecture Series*, University of Dortmund, 2003.
- [17] Written, I.H. and Frank, E. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementation*, Morgan Kaufman Publishers, 2000
- [18] Liu, H., Hussain, F. , Tan,C.L., and Dash,M. *Discretization: An Enabling Technique Data Mining and Knowledge Discovery*, 6, 393–423, Kluwer Academic Publishers, 2002.
- [19] Sousa, T., Neves, A. and Silva, A. *Swarm Optimization as a New Tool for Data Mining*, Proceedings of International Parallel and Distributed Processing Symposium 2003.
- [20] Hunt, J. *Java For Practitioners*, Springer-Verlag 2000
- [21] Quinlan, R., *Induction of Decision Trees*, Machine Learning, April 1986 81-106.
- [22] Shannon, C.E. and Weaver, W. (1949) *The Mathematical Theory of Communication*, University of Illinois Press, Urbana Illinois.
- [23] Parpinelli, S.R., Lopes, H.S. and Freitas, A.A. *An Ant Colony Algorithm for Classification Rule Discovery*, in Data Mining: a Heuristic Approach, pp. 191-208. London: Idea Group Publishing, 2002.
- [24] Dhari, V., Chou, D. and Provost, F. *Discovering Interesting Patterns for Investment Decision Making with GLOWER L - A Genetic Learner Overlaid With Entropy Reduction* Journal of Data Mining and Knowledge Discovery January, 2000
- [25] Freitas, A.A., *A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery* in Advances in Evolutionary Computation. Springer-Verlag, 2002.

[26] Spears, W.M. and De Jong, K.A. *Using Genetic Algorithms for Supervised Concept Learning*, Machine Learning Volume 13, Issue 2-3 Nov./Dec. 1993

[27] Noda, E., Freitas, A.A. and Lopes, H.S. *Discovering Interesting Rules with a Genetic Algorithm*, Proc. Congress on Evolutionary Computation (CEC-99), 1322-1329. Washington D.C., USA, July 1999

[28] Parpinelli, S.R., Lopes, H.S. and Freitas, A.A. *An Ant Colony Algorithm for Classification Rule Discovery* in Data Mining: a Heuristic Approach, pp. 191-208. London: Idea Group Publishing, 2002.

[29] Monmarchè, N. *On Data Clustering with Artificial Ants* in Data Mining with Evolutionary Algorithms, Research Directions – Papers from the AIII workshop (pp. 23-26) Menlo Park, CA, AIII Press.

[30] Lu, H. and Liu, H. *Decision Tables: Scalable Classification Exploring RDBMS Capabilities* Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000