Optimized Class Association Rule Mining using Genetic Network Programming with Automatic Termination

Eloy Gonzales, Bun Theang Ong, Koji Zettsu
Information Services Platform Laboratory
Universal Communication Research Institute
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
Tel: +81-774-98-6866, Fax: +81-774-98-6960
e-mail: {egonzales, bt_ong, zettsu}@nict.go.jp

***Abstract*—Association rule mining is one of the tasks of data mining and it has been extensively studied during the last years. As a consequence, recently, several methods for extracting association rules have been developed. Some methods use Evolutionary Algorithms to extract association rules. Among them, a relatively new method using Genetic Network Programming (GNP) has been developed and its effectiveness has been shown, which outperforms other conventional algorithms. However, there still remain some issues mainly focused on performance.**

To improve the conventional GNP data mining algorithmic efficiency without loss of reliability, a GNP enhanced with an automatic termination criteria named AT-GNP is proposed in this paper. Indeed, in an effort to save computational resources, the objective is to stop the search right before unnecessary function evaluations are performed. The concept of Gene Matrix (GM) is used to direct the search and to stop it at a proper instant.

An extensive comparison between the conventional GNP-based association rule mining and AT-GNP is performed in the simulations to evaluate the performance. Finally, the association rules extracted using both methods are applied to the classification problems and the prediction accuracies of them are compared with other conventional approaches.

***Keywords*-Association rule mining; classification; evolutionary computation; genetic network programming; termination criteria.**

I. Introduction

Among several methods of extracting association rules that have been reported, a relatively new Evolutionary Computation (EC) method named Genetic Network Programming (GNP) has also been developed recently and the effectiveness of applying it to the data mining is shown by several authors such as Gonzales et. al. [1][2][3] for diverse types of datasets.

However, it still suffers of performance issues especially concerned to the processing time. This is mainly due to the termination criteria which is basically after a fixed number of generation in the evolution of GNP. Empirically have been demonstrated by Shimada et. al. [4][5] that most of the association rules are extracted during the initial generations of the GNP, but the problem is to determine exactly when the algorithm has to terminate without loss of reliability.

There is not much work yet in the research of EAs dealing with the question of the termination criteria. Nevertheless, it is recognized that in many real world applications, saving computational resources is extremely important. There are only a few recent works on termination criteria for EAs. Giggs et. al. [6], empirically studied the problem characteristics in an attempt to determine the maximum number of generations. Kwok et. al. [8] used statistics to terminate the search when it is estimated that no further improvement in terms of solution quality can be expected. Jain et. al. [7] studied eight termination criteria with clustering techniques that examine the distribution of individuals in the search space at a given generation. Ong and Fukushima [9][10] introduced the concept of the Gene Matrix (GM). The GM is a matrix that represents subranges of the possible values of each variable. It gives an indication on the distributions of the variables over the search range. This information is used to provide the search with new diverse solutions and to let the search know how far the exploration process has been performed in order to terminate it. Our proposed method also takes advantage of the GM. The particularity of using GM when compared to the existing methods that deal with the question of the termination criteria (see above) is that the algorithm is expected to terminate without a priori knowledge of any desirable or available solution range, and of any specific number of iterations or function evaluations. Actually, the termination instant after completion of adequate exploration and exploitation is determined by the algorithm itself.

The aim of this paper is to extend the conventional GNP-based mining method [4][5] by using a variation of the GM to guide the search and the evolution of the GNP individuals. That is, a mechanism similar to the GM is applied to GNP. This mechanism ensures that all judgment and decision nodes have been mutually joined to each other at least a given number of times before terminating the search. Consequently, the diversity of the solutions is favored. Concurrently, the depth of the graph structure of GNP is not altered, thus preserving the quality of the final

Copyright (c) IARIA, 2012.   ISBN: 978-1-61208-227-1

solutions.

The following sections of this paper are organized as follows: In Section II, a brief description of association rules is presented; the outline of GNP is briefly reviewed in Section III, where also the enhanced method for rule extraction using GM is presented. Simulation results are described in Section IV, and finally, conclusion and future work are given in Section V.

II. Association Rules

Zhang et. al. [11] introduced a formal statement of the problem of mining association rules. Let $I = \{A_1, A_2, \ldots, A_l\}$ be a set of l distinct attributes. Let T be a transaction which contains a set of attributes such that $T \subseteq I$. D be a database with different transaction records T. A transaction T contains X, a set of some attributes in I, if $X \subseteq T$.

An association rule is an implication of the form of $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. X is called antecedent and Y is called consequent of the rule. In general, the set of attributes X and Y are called itemsets.

There are two important basic measures for association rules, support and confidence. Support of an association rule $X \Rightarrow Y$ is defined as the percentage of records that contain $X \cup Y$ to the total number of records in the database.

Confidence of an association rule $X \Rightarrow Y$ is defined as the percentage of the number of transactions that contain $X \cup Y$ to the total number of records that contain X.

This measure indicates the relative frequency of the rule, that is, the frequency with which the consequent is fulfilled when the antecedent is also fulfilled .

However, the support-confidence framework has been shown not enough to extract interesting association rules, therefore, in this paper, the cosine correlation measure is used in addition to the conventional measurements of support and confidence.

Given two itemsets X and Y, the cosine measure is defined as:

$$cosine(X,Y) = \frac{P(X \cup Y)}{\sqrt{P(X)\ P(Y)}} = \frac{supp(X \cup Y)}{\sqrt{supp(X)\ supp(Y)}} \tag{1}$$

Cosine is a number between 0 and 1. A value close to 1 indicates positive correlation between X and Y. Cosine measure is also a null-invariant measure.

Therefore, the problem of mining class association rules is to find all rules that are highly likely to be interesting, that is, satisfying the minimum support, confidence and cosine thresholds.

$$\begin{gathered} support(X \Rightarrow Y) \geq min_{supp}, \\ confidence(X \Rightarrow Y) \geq min_{conf} \text{ , and} \\ cosine(X \Rightarrow Y) \geq min_{cosine} \end{gathered} \tag{2}$$

III. Genetic Network Programming

Genetic Network Programming (GNP), introduced by Hirasawa et. al. [12], [13], [14], is one of the evolutionary optimization algorithms, which evolves directed graph structures as solutions instead of strings (Genetic Algorithms) or trees (Genetic Programming). The main aim of developing GNP was to deal with dynamic environments efficiently by using the higher expression ability of graph structures.

The basic structure of GNP is shown in Fig. 1. The graph structure is composed of three types of nodes that are connected on a network structure: a start node, judgment nodes (diamonds), and processing nodes (circles). Judgment nodes are the set of J_1, J_2, …, J_p, which work as *if-then* conditional decision functions and they return judgment results for assigned inputs and determine the next node to be executed. Processing nodes are the set of P_1, P_2, …, P_q, which work as action/processing functions. The start node determines the first node to be executed. The nodes transition begins from the start node, however there are no terminal nodes. After the start node is executed, the next node is determined according to the node's connections and judgment results.

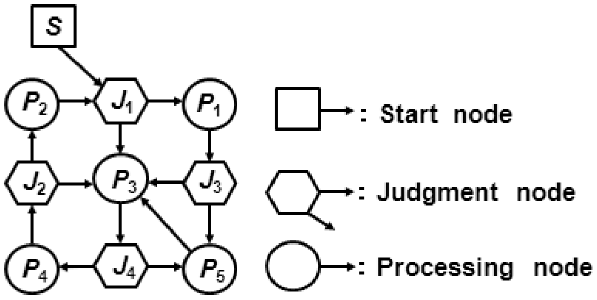
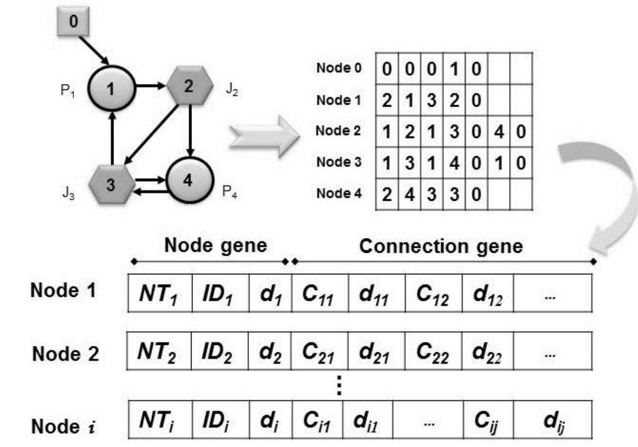


Figure 1. Basic structure of GNP

The gene structure of GNP (node i) is shown in Fig. 2. The set of these genes represents the genotype of GNP-individuals. NT_i describes the node type, $NT_i = 0$ when node i is the start node, $NT_i = 1$ when node i is a judgment node and $NT_i = 2$ when node i is a processing node. ID_i is an identification number, for example, $NT_i = 1$ and $ID_i = 1$ mean node i is J_1. C_{i1}, C_{i2}, …, denote the nodes, which are connected from node i firstly, secondly, …, and so on depending on the arguments of node i. d_i and d_{ij} are the delay time, which are the time required to execute the judgment or processing of node i and the delay time of transition from node i to node j, respectively. In this paper, the execution time delay d_i and the transition time delay d_{ij} are not considered.

A. Class Association Rule Mining using GNP

When GNP is applied to class association rule mining [4] [5], attributes of the dataset and their values correspond to the functions of judgment nodes in GNP-individuals. Association rules are represented as the connections of nodes.


0
P1
1
2
J2
J3
3
4
P4
Node 0 | 0 0 0 1 0
Node 1 | 2 1 3 2 0
Node 2 | 1 2 1 3 0 4 0
Node 3 | 1 3 1 4 0 1 0
Node 4 | 2 4 3 3 0
Node gene
Connection gene
Node 1 | NT1 | ID1 | d1 | C11 | d11 | C12 | d12 | ...
Node 2 | NT2 | ID2 | d2 | C21 | d21 | C22 | d22 | ...
Node i | NTi | IDi | di | Ci1 | di1 | ... | Cij | dij

***NT_i* : node type (Start node=0; Judgment node=1; Processing node=2)**
***ID_i* : identification number; *d_i, d_{ij}* : delay time; *C_{ij}* : connected node**

Figure 2. Gene structure of GNP (node i)

Candidate rules are obtained by genetic operations. Rule extraction using GNP is done without identifying frequent itemsets used in Apriori-like methods such as Agrawal et. al. [15] and stored in a pool through generations. The fundamental difference with other evolutionary methods is that GNP evolves in order to store new interesting rules in the pool, not to obtain the individual with the highest fitness value.

Let A_i be an attribute in a database and its value be 1 or 0, and C be the set of class labels. The method extracts the following association rules:

$(A_m = 1) \wedge \cdots \wedge (A_n = 1) \Rightarrow (C = k)$,
$(C = 0, 1, 2, \ldots, K)$

1) Genetic Operations: Changing an attribute to another one or adding some attributes in the rules would be considered as candidates of important rules. These rules can be obtained effectively by GNP genetic operations, because mutation and crossover will change the connections or contents of the nodes.

Three kinds of genetic operators are used for judgment nodes: GNP-crossover, GNP-mutation-1 (change the connections) and GNP-mutation-2 (change the function of nodes).

- GNP-Crossover: uniform crossover is used. Judgment nodes are selected as the crossover nodes with the probability of P_c. Two parents exchange the gene of the corresponding crossover nodes.
- GNP-Mutation-1: Mutation-1 operator affects one individual. The connection of the judgment nodes is changed randomly by mutation rate of P_{m1}.
- GNP-Mutation-2: Mutation-2 operator also affects one individual. This operator changes the functions of the judgment nodes by a given mutation rate P_{m2}.

On the other hand, all the connections of the processing nodes are changed randomly. At each generation, all GNP-individuals are replaced with the new ones by the following criteria: The GNP-individuals are ranked by their fitness values and the best one-third GNP-individuals are selected. After that, these GNP-individuals are reproduced three times for the next generation using the genetic operators described before.

If the probabilities of crossover (P_c) and mutation (P_{m1}, P_{m2}) are set at small values, then the same rules in the pool may be extracted repeatedly and GNP tends to converge prematurely at an early stage. If the probability of mutation is set at high values, then some genetic characteristics of the individuals might be lost. These parameter values are chosen experimentally avoiding these issues.

2) Fitness of GNP: The number of processing nodes and judgment nodes in each GNP-individual is determined based on experimentation depending on the number of attributes processed. The connections of the nodes and the functions of the judgment nodes at an initial generation are determined randomly for each GNP-individual. Fitness of GNP is defined by:

$$F = \sum_{r \in R} \{\mathit{cosine(r)} + \alpha_{new}(r) + \beta(n(r) - 1)\} \qquad (3)$$

The terms in Eq. (3) are as follows:

R: set of suffixes of extracted important association rules satisfying the minimum support-confidence-correlation measure in a GNP individual
cosine(r): value of cosine correlation of rule r,
$\alpha_{new}(r)$: additional constant defined by

$$\alpha_{new}(r) = \begin{cases} \alpha_{new} & \text{(rule } r \text{ is new)} \\ 0 & \text{(rule } r \text{ has been already extracted)} \end{cases} \qquad (4)$$

β: coefficient for the number of attributes.
$n(r)$: the number of attributes in the antecedent of rule r.

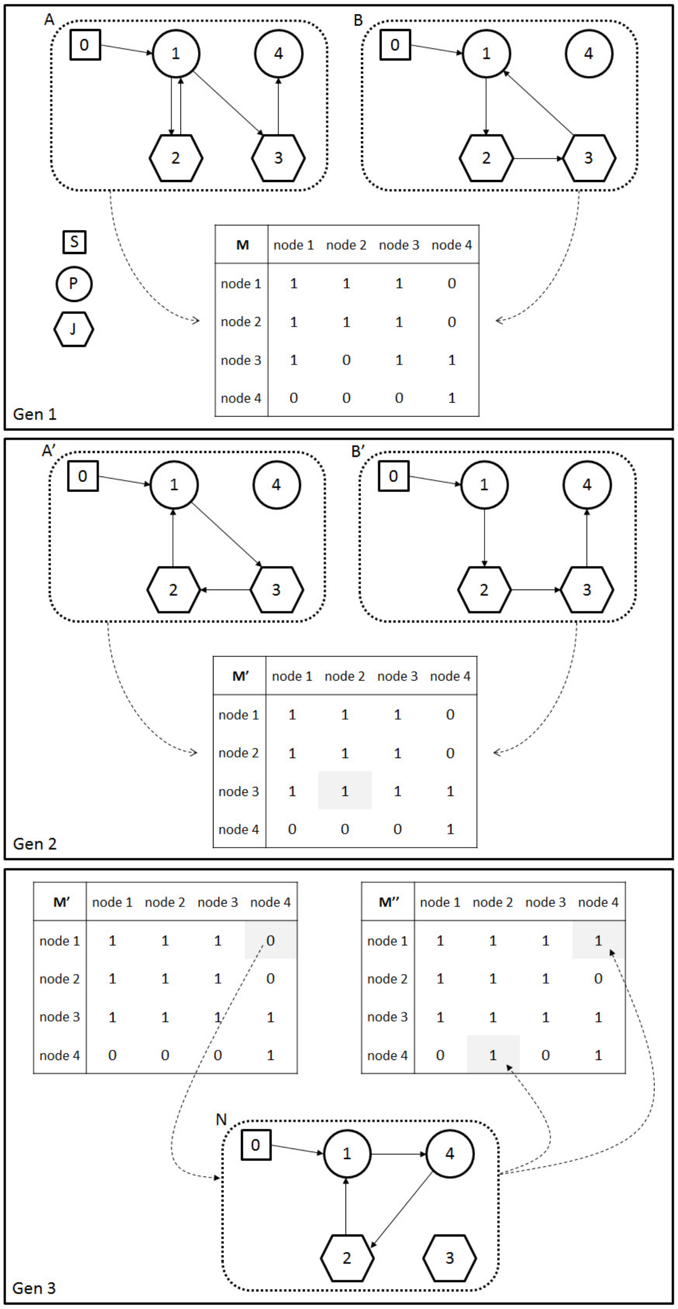
cosine, $n(r)$ and $\alpha_{new}(r)$ are concerned with the importance, complexity and novelty of rule r, respectively.

The fitness represents the potential to extract new rules.

B. Termination Mechanism: Gene Matrix and Mutagenesis

In AT-GNP, the termination instant is determined based on the GM, that works mutually with a special mutation operator called "mutagenesis". Mutagenesis is a more artificial mutation operation that allows some characteristic children to improve themselves by modifying their genes in accordance with the status of the GM.

During the search, the information related to the connections between each node is stored within a matrix M. M is initialized as a square diagonal zero matrix of order $p+q$, where p is the number of judgment nodes and q is the number of processing nodes. While the nodes are being connected during the search process, the corresponding entries in M are updated with a non-null value. This information is used in two ways. First, areas of the search space being



M	node 1	node 2	node 3	node 4
node 1	1	1	1	0
node 2	1	1	1	0
node 3	1	0	1	1
node 4	0	0	0	1

M′	node 1	node 2	node 3	node 4
node 1	1	1	1	0
node 2	1	1	1	0
node 3	1	1	1	1
node 4	0	0	0	1

M′	node 1	node 2	node 3	node 4
node 1	1	1	1	0
node 2	1	1	1	0
node 3	1	1	1	1
node 4	0	0	0	1

M″	node 1	node 2	node 3	node 4
node 1	1	1	1	1
node 2	1	1	1	0
node 3	1	1	1	1
node 4	0	1	0	1

Figure 3. An example of the termination mechanism.

unexplored are revealed during the search. Thus, at each generation, some individuals are mutated in such a way that unexplored regions are visited: The so-called mutagenesis. Second, when M does not contain zero entries anymore, the search is considered to have achieved an advanced exploration process and is stopped.

Mutagenesis operates in two ways in combination with the GM. First, some of the worst individuals that have been selected by the survivor operator to figure in the next generation are altered. To avoid premature convergence, the number of selected individuals should be less than the number of individuals altered by the normal mutation operator. Preliminary tests show that altering the worst or the two worst individuals lead to good performance in most cases. By doing so, we keep genetic diversity and accelerate the exploration process. The difference with the normal mutation operator is that it is not completely random. Indeed, mutagenesis is guided by the status of the GM. Specifically, a zero-position in GM is randomly chosen, say the position (i, j). Then the considered individual sees one of its connections altered with a link between nodes i and j. Hence, there is a chance for the crossover operation to explore different combinations of solutions containing this setting. Afterward, the GM is updated since a new connection has been created.

Figure 3 shows an example of this mechanism with three generations, namely, *Gen 1*, *Gen 2* and *Gen 3*. For the sake of simplicity, the population is reduced to two individuals. At generation *Gen 1*, individuals A and B are represented, along with the matrix M. An individual can be formed using two processing nodes and two judgment nodes. Hence, M is a square matrix of order 4. One can see that many entries are still equal to zero: Entry $(3, 2)$ for instance, meaning that node 3 is not followed by node 2 in any individual, although node 2 leads to node 3 in individual B. In generation *Gen 2*, A and B has been evolved to become individuals A' and B', respectively. Although they represent completely different solutions, with hopefully higher fitness, the matrix M' associated with the second generation reveals that the contribution of A' and B' in increasing the diversity of the population is very poor. Indeed, from the first to the second generation, a comparison between M and M' shows that only entry $(3, 2)$ turned to a non-null value. Consequently, we use this information to accelerate the search by specifically generating new solutions so that their composition contains connections between nodes that have not been explored. Let us consider entry $(1, 4)$ for instance. In our example of Figure 3, an original mutation operator modifies during the third generation (normally, at each generation) a candidate solution such that the resulting solution, here referred to as N, will explicitly contain nodes 1 and 4 connected to each other. In this way, M evolves over the generations such that all its entries become non-null. When that point is reached, then the search is terminated.

IV. Methodology

Fig. 4 shows the schema of the proposed method to evaluate the classification accuracy. 10-fold cross validation procedure was performed on the dataset and the results will be given by their average.

The training set and test set are generated randomly from the dataset. Using the training set, the proposed AT-GNP mining method is applied to obtain a pool of class association rules for each class. Two classes are shown in Fig. 4 as an example, that is, Class A and Class B. Finally,

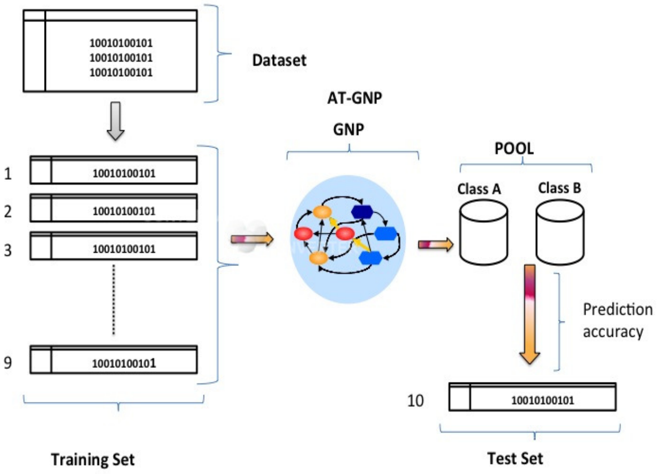


Figure 4. Schema for evaluating the classification accuracy. [2]

these pools are used to evaluate the prediction accuracy of the test set. All algorithms were coded in Java language. Experiments were performed on a 3.2 GHz Pentium PC with 12GB main memory, running Microsoft Windows 7 Ultimate.

A. Numerical Experiments

To assess the performance of the automatically terminated GNP (AT-GNP) against the conventional GNP, the results from 4 widely used databases in the field of DM [16] are compared. Both AT-GNP and GNP were confronted to a classification problem and 10 independent runs were performed in the same experimental conditions for each database. Table I reports the obtained averaged results in terms of number of generations, number of extracted rules and running time for each considered class as well as in terms of accuracy level, for databases “labor”, “crx”, “hepatitis” and “vehicle”, respectively.

Table II shows the comparison of the classification accuracy using several conventional methods. The results of C4.5. [17], Ripper [18], CBA [19], CMAR [20] and CPAR are taken from the Yin and Han paper [21].

For the four considered databases, it can be seen that for an equivalent or slightly improved accuracy, the number of generations obtained by AT-GNP right before automatic termination in all classes is on average 55% less expensive than what is required by GNP. The computation time is directly proportional to the number of generations and sees a reduction of 60% on average. The amount of extracted rules by AT-GNP is in almost all cases lower or of same order than of GNP. However, as indicated by the accuracy level, this does not have a negative impact on the quality of the final solution. For the last database, it is very interesting to notice that where the number of generations obtained by automatic termination is higher than of GNP, the number of extracted rules was particularly low. This may be explained by the fact that AT-GNP is automatically adjusting the effort in an attempt to extract more rules.

During our experiments, we have also considered allowing artificially much more and much less number of generations to assess whether or not AT-GNP suffered from premature convergence or did not terminate without unnecessary computation. However, as partially indicated by the comparison with GNP and a doubled number of generations, it is clear that AT-GNP did dot suffer from any of them.

V. Conclusion and Future Work

By equipping GNP with an automatic termination, we could alleviate the need to specify a given number of generations before running the algorithm. The numerical experiments demonstrated that our mechanisms could determine a proper termination instant without prior knowledge of the database to be handled. For an equivalent or slightly superior accuracy level, AT-GNP requires on average half the number of generations needed by GNP and is thus also reducing the computing time by half.

For future work, the method will be extended to deal with large and heterogeneous scientific databases combined with web data. Also, the authors will study the circumstances under which our termination technique is accurate. It involves the distribution of the data statistics of the databases.

References

[1] E. Gonzales, T. Nakanishi and K. Zettsu, “Large-Scale Association Rule Discovery from Heterogeneous Databases with Missing Values using Genetic Network Programming”, in *Proc. of the First International Conference on Advances in Information Mining and Management (IMMM2011)*, pp. 113-120, Barcelona, Spain, October 2011.

[2] E. Gonzales and K. Zettsu, “Association Rule Mining from Large and Heterogeneous Databases with Uncertain Data using Genetic Network Programming”, in *Proc. of the Fourth International Conference on Advances in Databases, Knowledge and Data Applications (DBKDA2012)*, pp. 74-80, Saint Gilles, Reunion Island, March 2012.

[3] E. Gonzales, K. Taboada, K. Shimada, S. Mabu, and K. Hirasawa, “Combination of Two Evolutionary Methods for Mining Association Rules in Large and Dense Databases”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 13, No. 5, pp. 561-572, 2009.

[4] K. Shimada, R. Wang, K. Hirasawa and T. Furuzuki, “Medical Association Rule Mining Using Genetic Network Programming”, *IEEJ Trans. EIS*, Vol. 126, No. 7, pp. 849-856, 2006.

[5] K. Shimada, K. Hirasawa, and T. Furuzuki, “Genetic Network Programming with Acquisition Mechanisms of Association Rules”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 10, No. 1, pp. 102-111, 2006.

[6] M. S. Giggs, H. R. Maier, G. C. Dandy, and J. B. Nixon, *Minimum number of generations required for convergence of genetic algorithms*, in “Proceedings of 2006 IEEE Congress on Evolutionary Computation”, Vancouver, BC, Canada, pp. 2580-2587, 2006.

Table I
COMPARISON RESULTS BETWEEN GNP AND AT-GNP FOR SEVERAL DATASETS.

labor	Class “Good”			Class “Bad”		
	Generations	Num. of Rules	Time (s)	Generations	Num. of Rules	Time (s)
GNP	101 (fixed)	12363	103.4	101 (fixed)	2991.2	21
AT-GNP	43 (auto)	6347.6	32.4	43.4 (auto)	2237	10.2

CRX	Class “+”			Class “–”		
	Generations	Num. of Rules	Time (s)	Generations	Num. of Rules	Time (s)
GNP	101 (fixed)	562	13	101 (fixed)	1317.6	17.2
AT-GNP	48.4 (auto)	444.6	7	47.6 (auto)	854	8.6

hepatitis	Class “live”			Class “die”		
	Generations	Num. of Rules	Time (s)	Generations	Num. of Rules	Time (s)
GNP	101 (fixed)	34983.4	428	101 (fixed)	2819.6	16.6
AT-GNP	42.6 (auto)	15727.4	97.2	45.2 (auto)	1847.8	8.2

vehicle	Class “bus”			Class “saab”			Class “opel”			Class “van”		
	Gen.	Num. Rules	T(s)	Gen.	Num. Rules	T(s)	Gen.	Num. Rules	T(s)	Gen.	Num. Rules	T(s)
GNP	101	596	11.2	101	4.6	7.8	101	1	7.8	101	174	9.6
AT-GNP	68.4	486.4	8	173.8	5	13	186.2	1	15.4	65.6	159.4	6.6

Table II
COMPARISON OF CLASSIFICATION ACCURACY WITH OTHER METHODS

Dataset	AT-GNP	GNP	C4.5	Ripper	CBA	CMAR	CPAR
Labor	90.66%	89.66%	79.30%	84.00%	86.30%	89.70%	84.70%
CRX	84.01%	83.44%	84.90%	84.90%	84.70%	84.90%	85.70%
Hepatitis	86.15%	86.11%	80.60%	76.70%	81.80%	80.50%	79.40%
Vehicle	76.12%	76.12%	72.60%	62.70%	68.70%	68.80%	69.50%
AVERAGE	84.23%	83.83%	79.35%	77.07%	80.37%	80.97%	79.82%

[7] B. J. Jain, H. Pohlheim, and J. Wegener, *On termination criteria of evolutionary algorithms*, in “Proceedings of the Genetic and Evolutionary Computation Conference”, Morgan Kaufmann Publishers, pp. 768, 2001.

[8] N. M. Kwok, Q. P. Ha, D. K. Liu, G. Fang, and K. C. Tan, *Efficient particle swarm optimization: a termination condition based on the decision-making approach*, in “Proceedings of the IEEE Congress on Evolutionary Computation”, Singapore, pp. 25-28, 2007.

[9] B. T. Ong and M. Fukushima, *Genetic algorithm with automatic termination and search space rotation*, Memetic Computing, **3**, pp. 111-127, 2011.

[10] B. T. Ong and M. Fukushima, *Global optimization via differential evolution with automatic termination*, Numerical Algebra, Control and Optimization, **2**, pp. 57-67, 2012.

[11] C. Zhang and S. Zhang, *Association Rule Mining: models and algorithms*, Springer, 2002.

[12] S. Mabu, K. Hirasawa, and J. Hu, “A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning“, Evolutionary Computation, *MIT Press* , Vol 15, No. 3, pp. 369-398, 2007.

[13] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu, J. Hu, and S. Markon, “A Double-deck Elevator Group Supervisory Control System using Genetic Network Programming”,*IEEE Trans. on System, Man and Cybernetics, Part C*, Vol. 38, No. 4, pp. 535-550, 2008.

[14] T. Eguchi, K. Hirasawa, J. Hu, and N. Ota, “A study of Evolutionary Multiagent Models Based on Symbiosis”,*IEEE Trans. on System, Man and Cybernetics, Part B*, Vol. 36, No. 1, pp. 179-193, 2006.

[15] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, in *Proc. of the 20th VLDB Conf.*, pp. 487-499, 1994.

[16] Frank, A. Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [Last Access: May 14th, 2012]

[17] J. R. Quinlan, “4.5: Programs for Machine Learning”, Morgan Kaufmann, 1993.

[18] W. Cohen, “Fast effective rule induction”, In Proc. of the ICML’95, pp. 115-123, Tahoe City, CA, July 1995.

[19] B. Liu, W. Hsu and Y. Ma, “Integrating Classification and Association Rule Mining”, In Proc. of the KDD’98, pp. 80-86, New York, NY, Aug. 1998.

[20] W. Li, J. Han and J. Pei, “CMAR: Accurate and efficient classification based on multiple class-association rules”, In Proc. of the ICDM’01, pp. 369-376, San Jose, CA, Nov. 2001.

[21] X. Yin and J. Han, “CPAR:Classification based on Predictive Association Rules”, In Proc. of the 2003 SIAM International Conference on Data Mining (SDM’03), 2003.